# Simulation of mental models with recurrent neural networks

**Dissertation**
Zur Erlangung des Akademischen Grades
des Doktors der Naturwissenschaften
an der Fakultät für Biologie
Universität Bielefeld

vorgelegt
von
SIMONE KÜHN

Betreuer
HOLK CRUSE

Januar 2006

# Contents

## Versicherung der Autorenschaft

Ich versichere, dass ich die vorliegende Dissertation selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe. Alle Stellen, die dem Sinn nach anderen Werken entnommen sind, habe ich in jedem einzelnen Fall unter genauer Angabe der Quelle deutlich als Zitat kenntlich gemacht.

Bielefeld, den 23.01.2006

(Simone Kühn)

# Erklärung der Beteiligten der Gruppenarbeit

Prof. Dr. Wolf-Jürgen Beyn (Fakultät für Mathematik, Universität Bielefeld, 33501 Bielefeld) ist Autor des Beweises in Kapitel 3.6.
Simone Kühn (Abteilung für theoretische Biologie und biologische Kybernetik, Fakultät für Biologie, Universität Bielefeld, 33501 Bielefeld) ist Autor aller anderen Teile dieser Arbeit.

Bielefeld, den 23.01.2006

(Wolf-Jürgen Beyn) (Simone Kühn)

# Dank

Danken möchte ich vor allem Holk Cruse, der das Thema stellte, mir zu jeder Zeit mit Rat und Ideen zur Seite stand und ein wirklich toller Doktorvater ist, meinen Eltern, auf die ich mich in jeder Situation verlassen kann und jedem einzelnen aus der Abt4. Durch Euch, all die netten Kaffeegespräche und Diskussionen wird mir die Zeit der Promotion in unvergesslich schöner Erinnerung bleiben.

Gefördert wurde diese Arbeit durch die DFG (GK *Verhaltensstrategien und Verhaltensoptimierung*) und durch die EU (EU-IST, SPARK).

# List of Publications

Kühn S, Cruse H (2005) Mental representation and cognitive behaviour – a recurrent neural network approach. In: *Modeling Language, Cognition and Action: Proceedings of the 9th Neural Computation and Psychology Workshop* (Cangelosi A, Bugmann G, Borisyuk R, eds), Singapore: World Scientific, pp 183-192.

Kühn S, Cruse H (2005) Static mental representations in recurrent neural networks for the control of dynamic behavioural sequences. *Connection Science* 17: 343-360.

**See Chapter 2**

Kühn S, Beyn W-J, Cruse H (2005) Modelling memory function with recurrent neural networks consisting of Input Compensation Units I - Static situations. (submitted)

**See Chapter 3**

Kühn S, Cruse H (2005) Modelling memory function with recurrent neural networks consisting of Input Compensation Units II - Dynamic situations. (submitted)

**See Chapter 4**

# Zusammenfassung

Menschen sind in der Lage, mentale Modelle von Informationen, die sie aus der Umwelt aufnehmen, aufzubauen. Solche mentalen Modelle bilden die Grundlage für die Organisation und Strukturierung von sensorischer Information und Wissen und sind damit wesentlich, um Gedächtnisaufgaben ausführen zu können. Sie können in so genannten *cell assemblies*, also wechselseitig miteinander verschalteten Neuronen, realisiert werden. Für den Aufbau von mentalen Modellen muss das neuronale System adaptive Fähigkeiten besitzen, d.h. die Fähigkeit zu lernen. Eine wesentliche Voraussetzung für diese Lernfähigkeit ist die Anpassung der synaptische Übertragungsstärke in Abhängigkeit von der jeweiligen Situation.

Die Informationen aus der Umwelt, mit denen der Mensch und damit sein neuronales System in einzelnen Situationen konfrontiert ist, können verschiedengestaltig sein: Es kann sich zum Beispiel um statische Muster handeln, wie z.B. einen Baum oder einen Stuhl, aber auch um dynamische Szenen wie etwa ein vorbeifahrendes Auto oder ein auf und ab hüpfender Ball.

In dieser Arbeit werden zwei Ansätze vorgeschlagen, die eine Modellierung mentaler Modelle sowohl statischer als auch dynamischer Informationen mit Hilfe von rekurrenten neuronalen Netzen ermöglichen. Beiden Modellen ist gemein, dass die Lernprozesse selbstorganisiert und nur in Abhängigkeit von lokaler Information ablaufen. Da die Lerndynamik und die rekurrente Dynamik bei Online-Lernverfahren eng miteinander verwoben sind, ist der Erfolg des ersten Modells (Kapitel 2) stark abhängig von der Anpassung der Lernrate. Mit der *Input Compensation* (IC) Struktur wird in Kapitel 3 und 4 eine biologisch inspirierte Neuronenstruktur vorgeschlagen, die Lern- und rekurrente Dynamik voneinander entkoppelt und somit Online-Lernen in sehr effizienter Weise erlaubt. Rekurrente Netze, die mit solchen IC Neuronen ausgestattet sind, können erfolgreich trainiert werden, um sowohl statische als auch dynamische Situationen abzubilden. Somit bietet dieses Modell eine Grundlage für sehr vielfältige Gedächtnisleistungen.

# 1 Introduction

Lucretius derives in *De rerum natura* a cosmology, which postulates that the invisible atoms fall like rain straight downwards but time and again deviate little from their path. Due to this *clinamen*, i.e. little and unpredictable swerve, random collisions between the atoms occur resulting in gradual formation of complex atomic clusters. Thus, yet around 70 B.C. Lucretius describes a hypothesised constructive principle that provides a mechanism for an inevitable process in the complex world: for *self-organisation*. By the way, this unpredictable *clinamen* serves for him also as a physical foundation for human free will (Neubauer, 2003).

## 1.1 Self-Organisation

So already in this very early work as well as in its Greek predecessor Epicur the basic idea of self-organisation can be found: Order emerges spontaneously without any obviously apparent driving force. Fascinating examples of self-organisation can be found throughout various disciplines when looking at pattern formation processes: For example sand grains assembling into ripples or water molecules aggregating to form crystalline snowflakes, high in the clouds with temperatures far below freezing. But also biological systems provide a variety of phenomena which are explained by self-organising processes: The creation of structures by social animals like termite mounds or flocking behaviour like flocks of birds or schools of fish; the stripes of a zebra or patterns on the wings of butterflies (Camazine et al., 2001; 2003); formation of lipid bilayer membranes; emergence of sustained delay activity in memory tasks (Yakovlev et al., 1998; Del Giudice et al., 2003), synchronisation in neuronal firing (Singer, 1999) or homeostatic plasticity in neuronal firing (Turrigiano and Nelson, 2004). This undoubtedly incomplete list of self-organising systems can of course be expanded by many more examples – but although they originate from diverse fields they suffice to put forward the basic ideas of self-organisation: Nobody thinks of a snowflake-maker when seeing snowflakes or of someone painting black and white stripes on zebras. These higher-level properties emerge solely from the interplay of the lower-level components.

Scott Camanzine and colleagues have condensed this in their definition of self-organisation (2001, S.8, my emphases):

> *"Self-organization is a process in which **pattern at the global level of a system** emerges solely from numerous **interactions among the lower-level components** of the system. Moreover, the **rules** specifying interactions among the system's components are executed **using only local information**, without reference to the global pattern."*

Thus, without any ordering influence imposed to the system a global pattern emerges, which the lower-level entities themselves do not display. "Pattern" means here an organised arrangement of entities in space or time. The above-mentioned examples show that the lower-level components the systems consist of can be by themselves on very different levels of complexity like for example animals, cells, neurons or molecules.

This definition given by Camazine and colleagues can serve as a guideline throughout this work: The overall goal is to provide mechanisms by which "***pattern at the global level of a system"*** (Chapter 1.2) can emerge; here the desired patterns are mental representations of situations as they are thought to be the basis of any cognitive and adaptive behaviour (for details see below). The biological substrate of these mental representations is the neural tissue, i.e. the brain. Of course, it is impossible to model a whole brain in detail. Therefore, artificial neural networks often serve as simplified models for brain functions. Here two different types of recurrent neural networks are used: MMC networks (Chapter 2; Kühn and Cruse, 2005) and IC networks (Chapter 3 and 4). The former have been developed primarily to control motor tasks whereas the latter comprise a completely new type of networks.

The ***lower-level components*** (Chapter 1.3) interacting to produce the patterns at higher levels are artificial neurons. The ***rules***, by which the connections between these neurons are built up in a self-organised fashion ***use*** in both cases ***only local information*** (Chapter 1.4), thus are not obliged to an external teacher.

### 1.1.1 History of self-organisation

Even though focussing on self-organisation is a relatively new research area, the idea that material things and dynamics of systems tend to become by themselves in course of time what we observe at present has a long history. After the notion of the *clinamen* by

Lucretius and the Epicureans it was already Descartes who described it in the fifth part of his *Discours de la Méthode* (1637). The naturalists of the 18th century tried to explain the observed appearances of living organisms by understanding the universal laws of form, an idea which fell into disrepute because it was associated with Lamarckism. Only in the beginning of the 20th century was the idea revitalised by D'Arcy Wentworth Thompson in his book *On Growth and Form* (Thompson, 1917) who thought that the growth of form is a dynamic process driven by natural forces and not the endpoint of a teleological process. Since the midst of the 20th century a growing number of publications on self-organisation and emergent properties substantiated those ideas. The term "self-organisation" seems to have been coined first by the engineer and psychiatrist W. Ross Ashby (1947).

From then on self-organisation was studied in the fields of physics, chemistry, biochemistry, developmental biology, systems theory and computer science. Within the latter the primary applications have been in the area of learning, especially unsupervised learning (e.g. Hinton and Sejnowski, 1999), memory (e.g. Kohonen, 1989) and adaptation. For a more detailed description see Shalizi (2001).

## 1.2 Pattern at the global level: Mental representations and cognitive behaviour

As described above the desired patterns at global level which should emerge resulting from the interaction of recurrently connected single neuronal units are mental representations of environmental situation. But what are mental representations and why should they be a matter of particular interest?

Every living organism has to move around in order to find food to survive and mates to reproduce – thus it faces the problem of sensorimotor control. A solution for this problem requires a suitable control mechanism which helps to influence the behaviour "appropriately" with respect to the sensory situation. This problem can be tackled in at least three different ways:

First, the organism can simply move around and when bumping into things modify its behaviour according to these collisions. This is, of course, the most basic way of behaving. An improvement on this solution is to use sensory information directly for the

control of behaviour. Examples are reactive systems like the so-called Braitenberg-vehicles (Braitenberg, 1984): The problem here is that they can show appropriate behaviour only as long as the external stimulus is present. If it disappears, the behaviour disappears, too. Thus, instead of reacting directly to the input, a solution on a higher level is to use sensory information to construct mental representations of the environment as a basis for subsequent behaviour.

Here, the term representation is used in the broad sense of Steels (1995) as being a physical structure (for example electro-chemical states) which has correlations with aspects of the environment. Thus, a representation is meant to be a formal structure relating the information an organism has to cope with. Since the information processing capacity of an organism is limited, it is unable to process all input signals in depth. Hence, it is indispensable for the organism to be equipped with the capability of situating each stimulus within a conceptual system, i.e. to build up concepts. The main advantage of such a capability is to be able to reduce the complexity provided by a variable environment.

One should be aware of the fact that the term representation is connoted differently with respect to the different disciplines making use of it, as for example philosophy, cognitive science or computer science. Philosophers, for example, tried to solve the problem how Mind and Matter might be able to interact. Concerning sensation and thinking René Descartes proposed that our mind does not directly know the objects but only mediated by ideas which represent them. While Descartes was convinced of those representative ideas to be innate empiricists like John Locke, Thomas Hobbes or George Berkeley proposed that the ideas emerge during development.

Additionally, we often encounter the term representation in an ambiguous sense: On the one hand it is used in a formal sense where representation means to give some efficiently manipulable structure to an abstract concept like for example *knowledge* or a *group*, and on the other hand it is used in the concrete sense where representation means to construct some model or image of a concrete phenomenon like an external object or the movement of a falling body. In this work, the latter sense is accounted for.

Internal mental representations may become apparent for example in the form of imagination, dreaming, internal language, and in a very impressive way if we observe what young children from the age of about two years on do: They are nearly obsessed

with representation making. This can be seen for example by looking at their drawings and the language games they play again and again. In the latter the children try to assign meanings to symbols.

An advantage of such internal mental models besides the above mentioned reduction of complexity is that the behaviour can be uncoupled from direct environmental control. This enables the organism for example to respond to features of the world that are not directly present, to use past experiences to shape present behaviour, to plan ahead, to internally manipulate the content, etc. (Cruse, 2003b).

The ability of using mental representations of their bodies independently of the actual sensory input is developed in children at the age of four to five while chimpanzees are not able to acquire this capability. This becomes apparent in an experiment performed by Povinelli et al. (1999). They filmed children of this age as well as chimpanzees and presented them the video tapes three minutes afterwards. After this delay the children are still able to recognize themselves while the animals don't. These results suggest that the children from the age of about five years on can uncouple their mental representation from the sensory input in contrast to younger children and chimpanzees.

Therefore, we can conclude that these mental representations form an essential prerequisite to explain how organisms can behave in a cognitive adaptive way – which is a conclusion contradicting Brooks' (Brooks, 1991) idea of *Intelligence without representation*. Also in philosophical discourse the capacity of organisms to provide neural mechanisms to internally construct and process representations of their body and their environment in order to shape their behaviour is regarded to be a significant branch-point in evolutionary history (O'Brien and Opie, 2004). Thus, a causal role in controlling behaviour is attributed to those mental representations.

What has been said so far should not imply the impression of a strict separation between perception and generation of behaviour in the sense of the '*information processing metaphor*' (Pfeifer and Scheier, 1994), a concept which is going back to the work of Marr (1982). Sensory perception and behaviour rather have to be regarded to be different sides of the same coin: They resemble different aspects of the same neuronal system. This idea is in the line of argument of many approaches like the '*perception through anticipation*' approach (Möller, 1997), the '*Gestaltkreis*' (von Weizsäcker,

1950), the '*action-perception-cycle*' (Arbib, 1981), the '*representation-execution continuum*' (Jeannerod, 1994; Jeannerod, 1997), or the proposal by Prinz (1997) to assume a '*common coding of perception and action*'.

Many experimental results imply such a holistic view of a tight connection between internal representations of sensory information and the respective actions. Recent neurally based theories of action and action understanding suggest that (1) a mental simulation of the action to be performed is routinely generated along with the actual performance of the action (Wolpert et al., 1995; Jeannerod, 1999).

(2) A mental simulation of the action is generated in an observer's motor system when viewing someone else performing an action (or relevant parts of it): In electrophysiological recordings in monkeys so-called '*mirror neurons*' were found which respond to both self-generated action as well as observed actions in others (Di Pellegrino et al., 1992; Gallese et al., 1996; Rizzolatti et al., 1996; for a review see Rizzolatti and Craighero, 2004). This mirror system has also been shown to exist in humans. Neuroimaging studies reveal an activation of motor areas when imitating or observing actions (Hari et al., 1998; Cochin et al., 1999; Iacoboni et al., 1999; Buccino et al., 2001; Grezes et al., 2001). Moreover, mental simulations are even generated when subjects view manipulable tools (Grafton et al., 1997) and understand actions described in sentences (Rizzolatti and Arbib, 1998; Glenberg and Kaschak, 2002). Other examples of neurons that cannot be attributed to be either sensory or motor elements are the so-called '*bimodal neurons*', which code body-centred extra-personal space, described by Iriki et al. (1996; see also Sakata et al., 1997) and the '*decision neurons*' (Kast, 2001).

(3) A mental simulation of the described action appears also to be generated when action sentences are understood. Thus, bodily activity has a significant impact on understanding of language comprehension (e.g. Glenberg and Robertson, 1999; for further literature see Glenberg and Kaschak, 2002). This is what Glenberg calls the embodiment of language comprehension: Language is understood when we are able to simulate sentences using the same neural systems as those used in perception, action, emotion and perhaps other bodily states. The symbols of language are grounded by relating them to bodily processes. This idea is based on Lakoffs concept of the embodied mind (Lakoff, 1987; Lakoff and Johnson, 1999). He is arguing that almost all

of human cognition can only be understood when taking the body into account. There are many well-known proponents of this view of the importance of such an embodiment as Rafael Núñez, Humberto Maturana, Francisco Varela, Vilayanur Ramachandran, Gerald Edelman, Antonio Damasio and others. The embodiment hypothesis is also very close to the phenomenology of mind and the concept of "In-der-Welt-Sein" of Martin Heidegger and other existentialists.

Especially research on text comprehension and language understanding revealed this embodied nature of those internal representations. Mainly based on the work of Johnson-Laird (1983) and van Dijk and Kintsch (1983), linguists and psychologists found out that it is rather the situation described in a text than the text itself which is represented in the mind. This finding has implications on modelling such representations: To represent situations, it is necessary to be able to also simulate the dynamic aspects of the situations, a claim which is supported by studies unveiling these dynamic properties (Freyd, 1993; McIntyre et al., 2001; Glenberg and Kaschak, 2002; Zwaan et al., 2004).

To summarise: internal mental representations are essential for cognitive behaviour and tightly connect sensory information and bodily activity. Based on these considerations we propose two recurrent neural network models here which are suitable to account for these requirements of mental representations. In the approach described in Chapter 2 and Kühn and Cruse (2005) so-called MMC networks, which are primarily developed for generation of action like arm-movements or landmark navigation (e.g. Cruse, 2003a; Steinkühler and Cruse, 1998), are adapted to build up internal mental representations. Thus, the model complies with the findings of many studies revealing a tight connection between the perception and action system (Rizzolatti and Craighero, 2004) which seems to be an essential property for action understanding (Rizzolatti et al., 2001), controlling motor output (Jeannerod, 1999; Cruse, 2003a), but also for language production and understanding (Glenberg and Kaschak, 2002).
Further, in Chapter 3 a completely new structure for artificial neurons (see below) is proposed by which models can be built up that are capable of generating internal

representations of static environmental scenes as well as of the dynamics that might be involved (Chapter 4) and that form the basis for understanding.

## 1.3 Lower-level components: artificial neurons (MMC and IC)

Biological research has accumulated a vast amount of knowledge about structure and function of the brain and the lower-level components brain circuits consist of: The neurons, which are intricately interconnected (see e.g. Kandel et al., 2000).

Theoretical models for the description of neurons exist varying in the levels of abstraction (for an overview see Wilson, 1999; Gerstner and Kistler, 2002). The most detailed level incorporates the diffusion of ionic potentials along the dentritic tree with its complex geometry (Rall, 1989; Segev et al., 1989). Spiking neuron models range from detailed biophysical ones, the so-called compartmental or conductance-based models, to integrate-and-fire models. The former try to describe the generation and shape of each individual action potential as a function of the opening and closing dynamics of the ion channels in dependence of voltage and messenger molecules as exactly as possible by sets of equations. These models originate in the four differential equations by which (Hodgkin and Huxley, 1952) summarised their studies on the giant axon of the squid. The integrate-and-fire models are on a higher level of abstraction. They consider the neuron as a homogeneous unit generating spikes if the total excitation is sufficiently large without concerning the spatial structure of a neuron. The best-known formal spiking neuron model is probably the leaky integrate-and-fire model which has been studied intensively by Stein (1967a; 1967b). In this approach the neuron is modelled as a leaky integrator which is reset when firing occurs.

In contrast to these spiking neuron models rate coding models neglect the pulse structure of the neuronal output – they code the mean firing rate of a neuron which varies between zero and some maximum value. Therefore, they are on the highest level of abstraction. The most abstract level, of course, is to characterise a neuron as a device that is either on or off (1 or 0), a description introduced by McCulloch and Pitts (1943). The pioneering work on the concept of mean firing rates has been performed by Adrian (1928) who defined rates in terms of spike counts, i.e. an average over time. But when comparing the experimental literature there are at least three different notions of rate

referring to different averaging procedures: averaging over time as proposed by Adrian (1928), or averaging over several repetitions of the experiment, or averaging over a population of neurons (see Gerstner and Kistler, 2002).

Whether the information neurons process is rate coded or spike coded is a fundamental and still unsolved question in neuroscience whereas the dividing line is not always clearly drawn.

The level of abstraction used in simulations has to be suitable for the respective purpose. Someone who is interested in the molecular mechanisms of individual neurons has to choose a more detailed level of description than somebody who wants to model aspects of for example motor control, categorisation or short-term memory. As the scope of the work presented here is to model internal mental representations of external situations, i.e. to model short-term memory abilities, rate coding neurons are applied in both approaches, in the MMC networks (Chapter 2) as well as in the IC networks (Chapter 3 and 4). The neurons code the mean firing rate in the classical sense described by Adrian (1928) as the activations of the single neurons are meant to be averaged over time.

## 1.4 Rules using only local information: Training recurrent neural networks

Like neurons in biological neural tissue the artificial neurons are interconnected via synapses or synaptic weights and build neural networks. The single units can be connected in a feedforward manner, i.e. the information flow is oriented in one direction only. But real neurons especially in the brain are thought to be organised in highly interconnected neural networks also comprising many recurrent connections. Therefore, to model a system which is biologically more realistic, a recurrent neural network architecture is chosen.

The patterns, i.e. the representations, should emerge from interactions between the lower-level components in a self-organised process. Thus, learning is required as in the beginning the neurons of the network are connected by synaptic weights having random values or being zero.

In classical neural network theory three different types of learning algorithms are applied: supervised learning, reinforcement learning and unsupervised learning. For supervised learning a teacher is necessary that "tells" the system the desired state. In every time step the output of the network is compared with the desired output, as given by the teacher, and according to the error originating from this comparison the synaptic weights are changed. The problem with any teacher-forced learning procedure is that it is biologically highly implausible. Usually, nature does no provide any external information of how the correct output of a system or action of an agent should be.

Reinforcement learning procedures provide a reward (which can be either positive or negative) according to the quality of the output; the goal here is to maximise the cumulative reward over the course of a task (Kaebling et al., 1996; Sutton and Barto, 1998). Reinforcement learning can be related to supervised learning as far as external information, which goes beyond the simple input, is necessary; but it differs from supervised methods in that no correct input-output pairs are presented. For reinforcement learning biological mechanisms exist for example in children's development; many abilities are acquired due to positive feedback or punishment, i.e. negative rewards.

In unsupervised learning no global external knowledge, neither a teacher nor a rewarding system, exists. The weights are updated using local information only like input correlation as in Hebbian learning methods. Thus, it is assumed that biological systems make use of such learning methods as no global knowledge is necessary.

Both mechanisms used here – the DD rule (Chapter 2; Kühn and Cruse, 2005) and the IC learning procedure (Chapter 3 and 4) – follow this principle of not being dependent on global information. To change the weights of a neuron in both cases only information is needed, which is directly available at each neuronal unit.

When dealing with training of recurrent neural networks in particular another general problem arises: training should take place online, i.e. while the system is working or the agent is behaving. But in this situation, two dynamics are superimposed: the dynamics of the recurrent network as well as the dynamics of the effects of learning (Steil, 1999). To avoid this problem both dynamics are often uncoupled by introducing alternating epochs of phases during which the weights are changed and phases during which the recurrent dynamics are calculated (Baldi and Pineda, 1991; Jaeger and Haas, 2004). But,

of course, in any biological system it is not likely that learning only takes place in separated phases where no behaviour occurs. Thus, it is a big advantage of the two methods introduced here that they are applied online.

To come full circle, the two models presented here – the MMC model and the IC model – account for the requirements of self-organisation as postulated in the definition by Scott Camazine and colleagues: ***patterns at the global level*** – internal representations – *of a* ***system*** – recurrent neural networks – *emerge within from numerous* ***interactions among the lower-level components*** *of the system* – the artificial neurons. *Moreover, the* ***rules*** – the learning rules – *specifying interactions among the system's components are executed* ***using only local information****, without reference to the global pattern.*
Thus, the models proposed can provide a basis for self-organisation which is an inevitable process in the complex world. This organisation process proceeds in a biologically very plausible way, as it really only relies on local information available at each single neuron and thus has a great advantage over other technical learning procedures.

## 1.5 References

Adrian ED (1928) *The basis of sensation*. New York: W. W. Norton.

Arbib MA (1981) Perceptual structures and distributed motor control. In: *Handbook of physiology: Nervous system - Volume 2* (V.B. Brooks, ed), Bethesda, MD: American Physiology Society, pp 1448-1480.

Ashby WR (1947) Principles of self-organizing dynamic systems. *Journal of General Psychology* 37: 125-128.

Baldi P, Pineda F (1991) Contrastive learning and neural oscillator. *Neural Computation* 3: 526-545.

Brooks R (1991) Intelligence without reason. *Proceedings of the International Joint Conference on Artificial Intelligence*: 569-595.

Buccino B, Binkofski F, Fink GR, Fadiga L, Fogassi L, Gallese V, Seitz RJ, Zilles K, Rizzolatti G, Freund H-J (2001) Action observation activates premotor and parietal areas in a somatotopic manner: an fMRI study. *European Journal of Neuroscience* 13: 400-404.

Camazine S (2003) Patterns in nature. *Natural History* 6: 34-41.

Camazine S, Deneubourg J-L, Franks NR, Sneyd J, Theraulaz G, Bonabeau E (2001) *Self-Organization in Biological Systems*. Princton University Press.

Cochin S, Barthelmy C, Roux S, Martineau J (1999) Observation and execution of movement: similarities demonstrated by quantified electroencephalography. *European Journal of Neuroscience* 11: 1839-1842.

Cruse H (2003a) Landmark-based navigation. *Biological Cybernetics* 88: 425-437.

Cruse H (2003b) The evolution of cognition - a hypothesis. *Cognitive Science* 27: 135-155.

Del Giudice P, Fusi S, Mattia M (2003) Modelling the formation of working memory with networks of integrate-and-fire neurons connected by plastic synapses. *Journal of Physiology* 97: 659-681.

Di Pellegrino G, Fadiga L, Fogassi L, Gallese V, and Rizzolatti G (1992) Understanding motor events: a neurophysiological study. *Experimental Brain Research* 91: 176-180.

Freyd JJ (1993) Five hunches about perceptual processes and dynamic representations. In: *Attention and performance XIV: Synergies in experimental psychology, artificial intelligence, and cognitive neuroscience* (D. Meyer, S. Kornblum, eds), Cambridge, MA: MIT Press, pp 99-119.

Gallese V, Fadiga L, Fogassi L, and Rizzolatti G (1996) Action recognition in premotor cortex. *Brain* 119: 593-609.

Gerstner W, Kistler WM (2002) *Spiking neuron models. Single neurons, populations, plasticity*. Cambridge: University Press.

Glenberg AM, Kaschak MP (2002) Grounding language in action. *Psychonomic Bulletin & Review* 9: 558-565.

Glenberg AM, Robertson DA (1999) Indexical understanding of instructions. *Discourse Processes* 28: 1-26.

Grafton ST, Fadiga L, Arbib MA, and Rizzolatti G (1997) Premotor cortex activation during observation and naming familiar tools. *NeuroImage* 6: 231-236.

Grezes J, Fonlupt P, Bertenthal B, Delon-Martin C, Segebarth C, and Decety J (2001) Does perception of biological motion rely on specific brain regions? *NeuroImage* 13: 775-785.

Hari R, Forss N, Avikainen S, Kirveskari E, Salenius S, Rizzolatti G (1998) Activation of human primary motor cortex during action observation: a neuroimaging study. *Proc Natl Acad Sci USA* 95: 15061-15065.

Hinton GE, Sejnowski TJ (1999) *Unsupervised Learning: Foundations of Neural Computation*. Cambridge, MA: MIT Press.

Hodgkin AL, Huxley AF (1952) A quantitative describtion of ion currents and its applications to conduction and excitation in nerve membranes. *Journal of Physiology (London)* 117: 500-544.

Iacoboni M, Woods RP, Brass M, Bekkering H, Mazziotta JC, and Rizzolatti G (1999) Cortical mechanisms of human imitation. *Science* 286: 2526-2528.

Iriki M, Tanaka Y, Iwamura Y (1996) Coding of modified body schema during tool use by macaque postcentral neurons. *Neuroreport* 7: 2325-2330.

Jaeger H, Haas H (2004) Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication. *Science* 304: 78-80.

Jeannerod M (1994) The representing brain: Neural correlates of motor intention and imagery. *Behavioral and Brain Sciences* 17: 187-245.

Jeannerod M (1997) *The cognitive neuroscience of action*. Oxford: Blackwell.

Jeannerod M (1999) To act or not to act: Perspectives on the representation of actions. *Quarterly Journal of Experimental Psychology* 52A: 1-29.

Johnson-Laird PN (1983) *Mental models: towards a cognitive science of language, inference, and consciousness*. Cambridge: Cambridge University Press.

Kaebling L, Littmann ML, Moore AW (1996) Reinforcement learning. *Journal of Artificial Intelligence Research* 4: 237-285.

Kandel ER, Schwartz JH, Jessell TM (2000) *Principles of neural science*. New York: McGraw-Hill.

Kast B (2001) Decisions, decisions... *Nature* 411: 126-128.

Kohonen T (1989) *Self-organization and associative memory*. New York: Springer.

Kühn S, Cruse H (2005) Mental representation and cognitive behaviour – a recurrent neural network approach. In: *Modeling Language, Cognition and Action: Proceedings of the 9th Neural Computation and Psychology Workshop* (Cangelosi A, Bugmann G, Borisyuk R, eds), Singapore: World Scientific, pp 183-192.

Lakoff G (1987) *Women, fire, and dangerous things: What categories reveal about the mind*. Chicago: University of Chicago Press.

Lakoff G, Johnson M (1999) *Philosophy in the flesh: The embodied mind and its challenge to western thought*. New York: Basic Books.

Marr D (1982) *Vision.* San Francisco: Freeman.

McCulloch WS, Pitts W (1943) A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics* 5: 115-133.

McIntyre J, Zago M, Berthoz A, Lacquaniti F (2001) Does the brain model Newton's laws? *Nature Neuroscience* 4: 693-694.

Möller R (1997) Perception through anticipation - An approach to behaviour-based perception. In *Proc. New Trends in Cognitive Science*, Vienna, ASoCs Technical Report 97-0, pp 184-190.

Neubauer J (2003) Reflections on the "convergence" between literature and science. *Modern Language Notes* 118: 740-754.

O'Brien G, Opie J (2004) Notes towards a structuralist theory of mental representation. In: *Representation in mind: New approaches to mental representation* (H. Clapin, P. Staines, P. Slezak, eds), Greenwood Publishers.

Pfeifer R and Scheier Chr (1994) From perception to action: The right direction? In *Proceedings of PerAc 94*, Los Alamitos: IEEE Computer Society Press. pp 1-11.

Povinelli DJ, Landry AM, Theall LA, Clark BR, Castille CM (1999) Development of young children's understanding that the recent past is causally bound to the present. *Developmental Psychology* 35: 1426-1439.

Prinz W (1997) Why Donders has led us astray. Theoretical issues in stimulus-response compatibility. *Advances in Psychology* 118: 247-267.

Rall W (1989) Cable theory for dendritic neurons. In: *Methods in neural modelling: from synapses to networks* (C. Koch, I. Segev, eds), Cambridge, MA.: MIT Press. pp 9-62.

Rizzolatti G, Arbib MA (1998) Language within our grasp. *Trends in Neurosciences* 21: 188-194.

Rizzolatti G, Craighero L (2004) The mirror-neuron system. *Annual Review of Neuroscience* 27: 169-192.

Rizzolatti G, Fadiga L, Fogassi L, and Gallese V (1996) Premotor cortex and the recognition of motor actions. *Cognitive Brain Research* 3: 131-141.

Rizzolatti G, Fogassi L, and Gallese V (2001) Neurophysiological mechanisms underlying the understanding and imitation of action. *Nature Reviews Neuroscience* 2, 661-670.

Sakata H, Taira M, Kusunoki M, Murata A, Tanaka Y (1997) The parietal association cortex in depth perception and visual control of hand action. *Trends in Neurosciences* 20: 350-357.

Segev I, Fleshman JW, Burke RE (1989) Compartmental models of complex neurons. In: *Methods in neural modelling: from synapses to networks* (C. Koch, I. Segev, eds), Cambridge, MA.: MIT Press, pp 63-93.

Shalizi CR (2001) Causal Architecture, Complexity and Self-Organization in Time Series and Cellular Automata. 2001. Thesis: University of Wisconsin, Madison, Physical Department.

Singer W (1999) Neuronal synchrony: A versatile code for the definition of relations? *Neuron* 24: 31-47.

Steels L (1995) Intelligence - Dynamics and Representations. In: *The Biology and Technology of Intelligent Autonomous Agents* (L. Steels, ed), Berlin: Springer, pp 72-89.

Steil JJ (1999) *Input-Output Stability of Recurrent Neural Networks*. Göttingen: Cuvillier Verlag.

Stein RB (1967a) Some models of neuronal variability. *Biophysical Journal* 7: 37-68.

Stein RB (1967b) The information capacity of nerve cells using a frequency code. *Biophysical Journal* 7: 797-826.

Steinkühler U, Cruse H (1998) A holistic model for an internal representation to control movement of a manipulator with redundant degrees of freedom. *Biological Cybernetics* 79: 457-466.

Sutton RS, Barto AG (1998) *Reinforcement learning - An introduction*. Cambridge, MA: MIT Press.

Thompson DW (1917) *On Growth and Form*. Cambridge: Cambridge University Press.

Turrigiano GG, Nelson SB (2004) Homeostatic plasticity in the developing nervous system. *Nature Reviews Neuroscience* 5: 97-107.

van Dijk TA, Kintsch W (1983) *Strategies in text comprehension*. New York: Academic Press.

von Weizsäcker V (1950) *Gestaltkreis.* Stuttgart: Thieme.

Wilson HR (1999) *Spikes, decisions, and actions*. Oxford: University Press.

Wolpert,DM, Ghahramani Z, and Jordan MI (1995) An internal model for sensorimotor integration. *Science* 269: 1880-1882.

Yakovlev V, Fusi S, Berman E, Zohary E (1998) Inter-trial neuronal activity in inferior temporal cortex: a putative vehicle to generate long-term visual associations. *Nature Neuroscience* 1: 310-317.

Zwaan RA, Madden CL, Yaxley RH, Aveyard ME (2004) Moving words: dynamic representations in language comprehension. *Cognitive Science* 28: 611-619.

# 2 Static mental representations in recurrent neural networks for the control of dynamic behavioural sequences

What enables an organism to perform behaviour we would call cognitive and adaptive, like language? Here, it is argued that an essential prerequisite is the ability to build up mental representations of external situations to uncouple the behaviour from direct environmental control. Such representations can be realised by building up cell assemblies. The recurrent neural network presented to cope with this task has been used for generation of action but can also be utilised as a basis for mental representations due to its attractor characteristics. In this context, a new learning algorithm (*Dynamic Delta Rule*) is proposed, which leads to a self-organised weight distribution yielding stable states on the one hand and which, on the other hand, only activates subpopulations of larger networks that code for the respective situation. In a second step, ways are shown of how the static information of these internal models can be transformed into time-dependent behavioural sequences.

## 2.1 Introduction

'What remains if I subtract the fact my arm went up from the fact that I raised my arm?' (Wittgenstein, 1958: section 621). This question posed by Wittgenstein half a century ago could imply the existence of "something" in the brain beyond the performed action. Recent research on action and action understanding has indeed shown that internal representations of the actions to be performed are built up while acting (e.g. Jeannerod, 1999) – a concept that is also proposed by computer scientists (Wolpert et al., 1995). Additionally, many studies revealed a tight connection between perception of action of others and the motor system. In electrophysiological recordings in monkeys so-called 'mirror neurons' were found that respond to both self-generated action as well as observed actions in others (Di Pellegrino et al., 1992; Gallese et al., 1996; Rizzolatti et al., 1996; for a review see Rizzolatti and Craighero, 2004). Neuroimaging studies in humans also show the activation of motor areas when imitating or observing actions (Iacoboni et al., 1999, Buccino et al., 2001; Grezes et al., 2001). Moreover, mental

simulations are even generated when subjects view manipulable tools (Grafton et al., 1997) and understand actions described in sentences (Rizzolatti and Arbib, 1998; Glenberg and Kaschak, 2002). Some researchers even go a step further by arguing that motor-control is tightly connected with thinking in general: For example Calvin (1996) regards thoughts as movements that have not taken place yet and perhaps never will and Fuster (1995) states that thoughts are a kind of imagined movement.

All these results on 'shared representation' as sometimes called (Jeannerod 1999, Decety and Sommerville, 2003) have focused the researchers' view on the impact of bodily activity in understanding of language comprehension (e.g. Glenberg and Robertson, 1999; for further literature see Glenberg and Kaschak, 2002).

Therefore, in our approach we propose to adapt a model primarily developed for generation of action to build up internal mental representations[*] of the direct environmental situation. This representation can then be used as a basis to produce sequences of behaviour as for example language. Thus, the problem the model has to tackle is a bipartite one:

On the one hand it perceives different pieces of environmental information at the same time, for example different objects or persons, which have to be integrated or bound together into a coherent internal representation as a kind of working memory. Thus, a small number of neuronal components coding for these objects should be activated together for some limited time in order to represent the actual environmental situation – this problem, the so-called binding problem, is widely discussed in systems neurobiology (for a review see Roskies, 1999, and further the other articles published in the special issue Neuron 24).

On the other hand these pieces of information then have to be used to construct appropriate sequences of behaviour like for example a sentence. When producing language which is linear by nature (de Saussure, 1967) the speaker has to decide what to say first, what to say second, and so on from a non-sequential presentation; this is called *linearization problem* (Levelt, 1989) which is the second problem the model has to deal with.

Let us consider as an example the production of a language sequence of a two years old girl. At this age, language consists mainly of simple two- and three-word utterances

[*] In this work the term representation is used in the broad sense of Steels (1995): as being 'physical structures (for example electro-chemical states) which have correlations with aspects of the environment'.

(Mills, 1985). Assume that this girl is in her room, her mother is coming in and a book is lying on the floor. Now she may utter *'Mommy book'* (Tomasello, 1992) to express that she wants her mother to read from the book. All the items that should occur in the sequence – the mother and the book – are present as sensory input simultaneously. Thus, the temporal behavioural chain, in this case the sentence, has to be produced while the order of the sequence is not explicitly determined by the information available from the environment – a task which differs from the tasks performed by models proposed by Jordan (1986) or Elman (1990) for example (see Chapter 2.5).

The idea of the importance of mental representations for cognitive abilities – a widely-used term in cognitive science (von Eckardt, 1993) – has a long tradition (e.g. in the so-called 'picture theory' by Heinrich Hertz dating from 1884 (cited after Heidelberger, 1998). Especially in research on text comprehension, the relevance of mental representations got more and more into the focus in recent years inspired by the theory of situation models (van Dijk and Kintsch, 1983; for a review on situation models see Zwaan et al., 1998) and mental models (Johnson-Laird, 1983). Also Levelt (1989) argues that the construction of mental models in the sense of Johnson-Laird is rather more the rule than the exception as a first step in language production. Thus, he describes forming a *preverbal message* as a kind of input representation as the first processing component in his *blueprint for a speaker*.

What is the advantage of mental representations? By means of these representations the behaviour can be uncoupled from direct environmental control. This enables the organism for example to respond to features of the world that are not directly present, to use past experiences to shape present behaviour, to plan ahead, to internally manipulate the content etc. (Cruse, 2003b). All these instances characterise a special feature of language called 'displacement' (Hockett, 1960). Therefore, we conclude that these mental representations form an essential prerequisite to explain how organisms can behave in a cognitive way.

Environmental stimuli can be represented by activation of so-called cell assemblies, a theory which goes back to the idea of Hebb (1949). Various models to realise such cell assemblies have been proposed like multi-layer perceptrons (MLPs), Hopfield networks (Hopfield, 1982), Kohonen-maps (Kohonen, 1982), Jordan and Elman networks (Jordan, 1986; Elman, 1990), recurrent networks and recurrent experts (Wolpert and

Kawato, 1998; Tani and Nolfi, 1999). Here, we want to refer to a special type of recurrent neural network: the MMC network. These networks can be used on the one hand for the generation of action and on the other hand for building up mental representations of the environment due to their attractor characteristics.

## 2.2 MMC networks

In a nutshell, we want to develop a model here with which mental representations of existing environmental situations can be built up that in a second step can be transformed into sequential behaviour like for example sentences. Thus, the task to be accomplished can be split into two parts:

(1) Generating mental representations. For this task it should be possible to build up mental representations of more than one segmental situation consisting of some known objects simultaneously, i.e. several cell assemblies should be able to coexist. Take as an example an overall situation with a mother and a book building one segmental situation and additionally a boy and a chair building another segmental situation. The goal is to activate a "mother-book" assembly simultaneously with a "boy-chair" assembly within a neural network. To cope with this task we will describe the *Dynamic Delta Rule*, a new learning algorithm based on a combination of Hebb's Rule and the common delta rule to train MMC network (Chapter 2.3).

(2) Generating sequential behaviour. These mental representations of the environment which are static by nature and contain no sequential information should then be used in a second step to produce sequential behaviour as for example the utterance of the little girl *'Mommy book'* (Chapter 2.4). This means that the only input the network receives is the activation caused by the objects present in the environment which does not contain sequential information. Why are MMC network suitable to cope with these tasks?

### 2.2.1 MMC networks for generation of action and as basis for mental representations

MMC networks are fully connected recurrent neural networks. This means each unit (in the following 'unit' and 'neuron' are used synonymously when talking about neural networks) is connected with every other unit via a synaptic weight (open and filled circles and squares in Figure 2.1a). Primarily, this network was invented to solve geometrical tasks characterised by redundant degrees of freedom like the control of arm movements (Cruse et al., 1998; Steinkühler and Cruse, 1998; Steinkühler et al., 2000), six-legged walking (Kindermann and Cruse, 2002) and landmark navigation (Cruse, 2003a). In all these tasks the networks cope with the problem of sensorimotor integration.

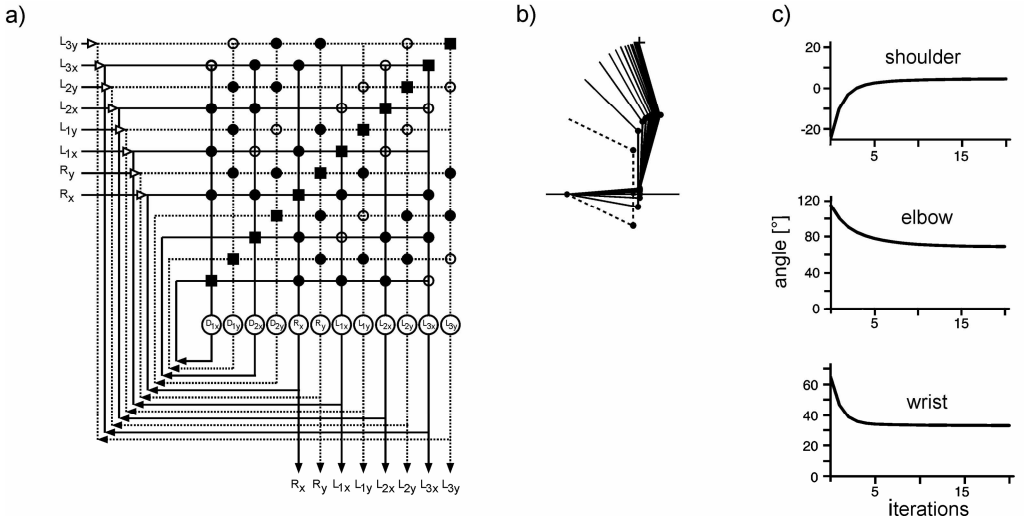


**Figure 2.1: (a) Schematic drawing of an MMC network for the control of a three-joint arm moving in a 2D plane. (b+c) Relaxation of the arm to meet a target point: (b) Movement of the arm from the starting configuration (---) towards the target point (cross). (c) Convergence of the three joints (shoulder, elbow and wrist) to stable states (modified after Cruse *et al.* 1998).**

Figure 2.1b shows an arm with three segments (and three joints) operating in a 2-D plane. If the network is provided with an endpoint to which the arm should point to – here depicted by a cross – it iterates and after some steps the arm points in the direction of the given point taking a geometrically possible configuration of the arm-segments. Thus the network is able to solve the classical problem of inverse kinematics even if the

problem is ill-posed (namely to find three joint angles such that the hand points to a given position in 2D space).

In this form, the network exploits the redundancies within the system: a given value is determined simultaneously in several ways as often found in biological systems. By calculating the Mean value of these Multiple Computations (MMC) the final output is obtained. The main feature of this network is that it converges to stable states corresponding to a geometrically correct solution, even when the input does not fully constrain the solution (Cruse and Steinkühler 1993). This is shown in Figure2,1c. External information is stored within the network by activation of several units. The neurons activate each other via recurrent connections in such a way that the activation is maintained.

These networks have two features making them suitable for our purpose. First, as has been mentioned above, there is evidence supporting a tight connection between perception and motor system. This connection is a characteristic of these "holistic" networks as we could not label the units to be either motor or sensory elements.

Second, if we uncouple the motor output from the network, it can serve as an internal model for example to represent the position of the arm or to simulate the movement of the arm reaching to a target. Hence, this type of network can be taken as a neural basis for mental representations because it provides a possibility to perform mental activities (e.g. the movement of an arm), i.e. uncouple the behaviour from direct control of the sensory input.

As will be shown below, the units cannot only be used to represent geometrical entities, but also abstract entities as for example objects occurring in the world (like a book or a chair). A tight connection between perception and motor system is also necessary for language production and has even been found to play an important role in language understanding (Glenberg and Kaschak, 2002). Thus, we put forward the hypothesis that neural networks like the MMC networks can not only be used for motor tasks but also for dealing with more abstract entities as is, for example, necessary for language.

In all applications of the MMC networks mentioned so far the synaptic weights within the network are fixed according to the equations forming the basis of the network. This is no problem for a body model because this could be regarded to be "innate". However, when an organism faces a new situation comprising different objects, the mental

representation must emerge starting with a "naïve" recurrent network with small random weights before the stimulus appears. Therefore, the main goal in the first step is to find a way to train the weights to stabilise the output of the network, i.e. to obtain stable states in the end which form the mental representations of the current environmental situation.

### 2.2.2 The network

Before explaining a learning algorithm that can be used to train MMC networks, we want to describe the principles of the network structure in more detail starting with a simple task: A mental representation of a situation should be built up that is characterised by two items $x_1$ and $x_2$, for example an environmental situation showing two objects (e.g. *mommy* and *book*). Here we use for a first approach localist-encoding linguistic units as they code for single linguistic entities, namely words (Cangelosi, 2004). Thus, the corresponding neural system consists of two units each of which represents one object (Figure 2.2). The problem of levels of representation will be addressed later (Chapter 2.5).

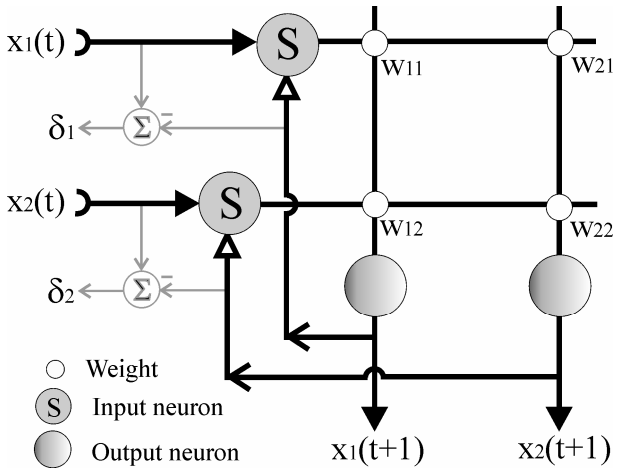


**Figure 2.2: Architecture of a two-unit MMC network: $x_i$(t) are the inputs, $x_i$(t+1) the outputs and $w_{ij}$ the weights. Neurons that calculate the weighted sum are depicted as shaded circles. The input neurons, which are suppression units marked by *S*, are shown as grey circles. The external inputs $x_i$ are suppressed by the recurrent connections symbolised by the open arrowheads. The difference $\delta_i$ between the first inputs $x_i$(1) and the output values $x_i$(t+1) is taken as error signals for training the weights.**

The units form a recurrent neural network in which the input neurons (depicted as grey circles marked by an S in Figure 2.2) are suppression units: The external inputs $x_i$ are

suppressed by the recurrent connections symbolised by the open arrowheads, which means that these external inputs are only effective during the first time step. The output neurons (shaded grey circles) are linear summation neurons, i.e. they calculate the weighted sum of the incoming activity.

The MMC network is a fully connected recurrent neural network which can be described by the following general equation for RNNs:

$$\mathbf{x}(t+1) = \mathbf{W} \cdot \mathbf{x}(t) \tag{1}$$

Here, $\mathbf{x}(t)$ is the recurrent input vector for *n* neurons, $\mathbf{x}(t+1)$ is the output vector, and **W** a $n \times n$ weight matrix. Thus, for a network with two units (Figure 2.2) the system of equations holds:

$$\begin{aligned} x_1(t+1) &= w_{11} \cdot x_1(t) + w_{12} \cdot x_2(t) \\ x_2(t+1) &= w_{21} \cdot x_1(t) + w_{22} \cdot x_2(t) \end{aligned} \tag{2}$$

The next step is to find a distribution of the weights $w_{ij}$ that leads to a stable state $x_1(t+1) \neq 0$ and $x_2(t+1) \neq 0$ after the network has been provided with an external input $x_1(t) \neq 0$ and $x_2(t) \neq 0$. A stable solution is achieved if there is a fixed relation between the parameters $x_1$ and $x_2$: $x_2 = f(x_1)$. The simplest case is given by a linear equation:

$$a \cdot x_1 - b \cdot x_2 = 0 \tag{3}$$

This equation is taken as basic equation for the MMC network leading to the following equations:

$$\begin{aligned} x_1 &= w_1 \cdot x_2; \quad \text{with } w_1 = \frac{b}{a} \\ x_2 &= w_2 \cdot x_1; \quad \text{with } w_2 = \frac{a}{b} \end{aligned} \tag{4}$$

Thus, equations (4) fulfil the basic equation (3) if the weights are constrained according to:

$$w_1 \cdot w_2 = 1 \tag{5}$$

Equations (4) can be transformed into equations (2) by introducing damping factors $d_i \neq 0$ by which the self-activating connections are weighted:

$$\begin{aligned} x_1(t+1) &= \frac{1}{1+d_1} \cdot (d_1 \cdot x_1(t) + w_1 \cdot x_2(t)) \\ x_2(t+1) &= \frac{1}{1+d_2} \cdot (w_2 \cdot x_1(t) + d_2 \cdot x_2(t)) \end{aligned} \tag{6}$$

By means of these equations, the four weights in Figure 2.2 and equations (2) can be specified in more detail:

$$w_{11} = \frac{d_1}{1+d_1},\ w_{12} = \frac{w_1}{1+d_1},\ w_{21} = \frac{w_2}{1+d_2} \text{ and } w_{22} = \frac{d_2}{1+d_2}.$$

The damping factors $d_i$ change the dynamics of the network: The higher they are, the slower the network converges to a stable state. Thus, the system obtains low-pass filter properties (Cruse et al., 1998) and oscillations that might occur during the relaxation process can be avoided successfully.

The principle characterising MMC networks, namely calculating one variable in many different ways, is reduced to a minimum in the case described here, as for each variable there exists only one equation which includes the contribution of the damping term.

## 2.3 Training the weights

### 2.3.1 MMC criterion

As mentioned above, a mechanism for training the weights is necessary to build up a mental representation of a new situation – for example when two objects are presented to the network. To this end, the equations forming the MMC network should fulfil condition (5) $w_1 \cdot w_2 = 1$, as the network is unstable in all other cases. Following condition (5) the MMC network is a neutrally stable system.

To put it in mathematical terms: A matrix of weights is searched for that has one eigenvalue $\lambda_1 = 1$ and a second eigenvalue $\lambda_2 < 1$. The eigenvector $\boldsymbol{v_1}$ and all linearly dependent vectors $\boldsymbol{v'_1}$ of $\boldsymbol{v_1}$ with the eigenvalue $\lambda_1 = 1$ are stable solutions of the networks. The target distribution of the weights described by equation (5), i.e. with given damping factors $d_i$, is obtained by calculating the zeros of the characteristic polynom

$$\begin{aligned} &\det(\mathbf{W}_d - \lambda\mathbf{I}) = 0 \quad \text{with } \lambda = 1 \\ &\Leftrightarrow 1 - w_1 \cdot w_2 = 0 \end{aligned} \tag{7}$$

Here, $\mathbf{W}_d$ is the weight matrix for given damping factors and $\mathbf{I}$ the identity matrix. If the damping factors $d_i$ are not given in advance (equations (2)), we can also derive a condition for stability from the zeros of the characteristic polynom with $\lambda = 1$:

$$\det(\mathbf{W}-\lambda\mathbf{I})=0$$

$$\Leftrightarrow (1-w_{11})\cdot(1-w_{22})-w_{12}\cdot w_{21}=0 \tag{8}$$

In general, the zeros of the characteristic polynom with $\lambda=1$ provide a condition, which yields stability of the network; therefore, it is termed *MMC criterion* in the following. Thus, to evaluate the state of the network during learning, we have to look whether this criterion is fulfilled. To measure how much the weights diverge from this criterion a Harmony function *H* is defined:

$$H=\frac{1}{1+(\det(\mathbf{W}-\lambda\mathbf{I}))^2} \tag{9}$$

Harmony *H* approaches the value 1 the more the better the MMC criterion is fulfilled, i.e. the more the characteristic polynom approaches zero.

### 2.3.2 Learning algorithm: Dynamic Delta Rule

The task is – as described above (Chapter 2.2) – to generate mental representations of more than one segmental situation simultaneously within a network consisting of an arbitrary large number of units. Figure 2.3 shows a section of such a network for the task of building up two mental models each of which describes a segmental situation comprising two objects (e.g. *mommy/book* and *boy/chair*). It is the aim to activate only those four synaptic connections, which combine the features belonging to the respective situation (grey boxes in Figure 2.3). Therefore, only four units of the network are depicted in Figure 2.3. How could these connections be trained accordingly to solve this binding problem?

If the first new segmental situation is presented, the two units coding for the objects of this situation (Figure 2.3: $x_1$, $x_2$) change their activity. This means, because of the recurrent architecture, that there is both a *pre-* and a *post-synaptic change* of activity at those four weights connecting the objects of this situation (Figure 2.3, grey box, upper left). This can be exploited by a learning algorithm to change these four weights (see below). Now assume that these weights have already been trained and the activation of the respective neurons has stabilised at a constant level different from zero. If later the network is provided with a second segmental situation (Figure 2.3: $x_3$, $x_4$), a learning algorithm, which is based on pre- and post-synaptic activity as usual hebbian mechanisms, would not only activate the four weights representing situation 2 (Figure 2.3, grey box, lower right) but all 16 weights. Therefore, the question arises, how it

could be avoided to change those eight weights that connect features of different segmental situations and that would produce 'crosstalk' between both representations (e.g. the weight connecting $x_2$ and $x_3$ marked by an asterisk in Figure 2.3). For this purpose, the following observation can be exploited: At those weights connecting units of different situations there is only a *pre*-synaptic *change* of activity. The activity of the post-synaptic side does not change anymore, because the unit $x_2$ has already stabilised after presentation of the first scene.

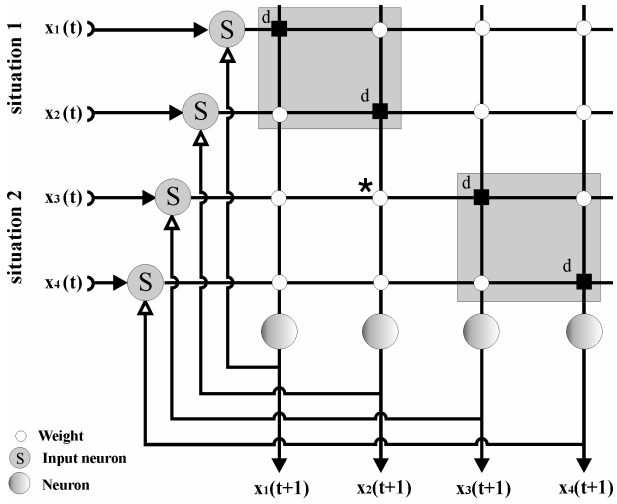


**Figure 2.3: Section of a network for a task where two segmental situations 1 and 2 are presented. Self-activating connections are defined by *d*. The two grey boxes mark the weights which should be activated when the two situations are presented to the network subsequently. The asterisk tags one of the weights that would produce 'crosstalk' between both representations when being different from zero.**

Consequently, we can solve the problem if we find a learning algorithm that meets two requirements: First, only those weights should be activated at which we find both a *pre*- and a *post-synaptic change* of activity and second, the MMC criterion should be fulfilled in the end to obtain a stable solution. How is it possible to change the weights that both requirements are fulfilled? This can be achieved by applying a combination of a hebbian mechanism, that, however, uses the temporal change $\Delta x$ of the activity instead of the activity $x$ itself, and the $\delta$-error as it is used in the delta rule.

We assume that, in the beginning, all the weights within the network have small values (e.g. zero, or small positive random values). Sensory input is presented to any of the input channels, for example to units $x_1$ and $x_2$. As a mental representation of a current external situation, which is represented by the external input vector, should be built up

this input vector can be used as trainer for the network. The difference $\delta$ between the external input vector $x(1)$ and the output vector $x(t+1)$ can then be taken as error signal for training the weights (see Figure 2.2 light grey lines). In doing so the loop through the environment can be closed. As representations are built up of actually present situations, these internal models are directly grounded in the external world (Cangelosi, 2001).

Therefore, we propose to change the weights by applying the following learning algorithm:

$$\Delta w_{ij} = \eta \cdot \Delta x_i \cdot \Delta x_j \cdot \delta_i$$

with $\eta$ being the learning rate, $\Delta x_i = x_i(t+1) - x_i(t)$, $\Delta x_j = x_j(t+1) - x_j(t)$ and $\delta_i = x_i(1) - x_i(t)$.

This learning algorithm can be interpreted as a combination of Hebb's rule, however applied to the temporal derivative of pre- and postsynaptic activity, and the delta rule. Therefore, this algorithm is termed *Dynamic Delta Rule* (DD Rule). The learning algorithm serves for two purposes: (i) it can select and activate specific weights to form a mental representation according to the actual environmental situation and (ii) it stabilises this neutrally stable system against incidental disturbances at those weight values.

Note that it is not the aim here – as in usual applications of the common delta rule – to minimise the $\delta$-error, but to minimise the value of the temporal changes $\Delta x_i$ and $\Delta x_j$. So after learning is finished there could still be a $\delta$-error which is, however, no problem for the concept presented here. Accordingly, the activations of the output units need not to equate the input activations after learning.

### 2.3.3 Results

To investigate the properties of this algorithm we first consider a network with only two units as shown in Figure 2.2. The system can be tested within two different conditions: One with fixed damping factors $d_i$ (see equations (6)) and another with free damping factors (see equations (2)). In the former case only two weights have to be trained, whereas in the latter case all four weights are free to be changed.

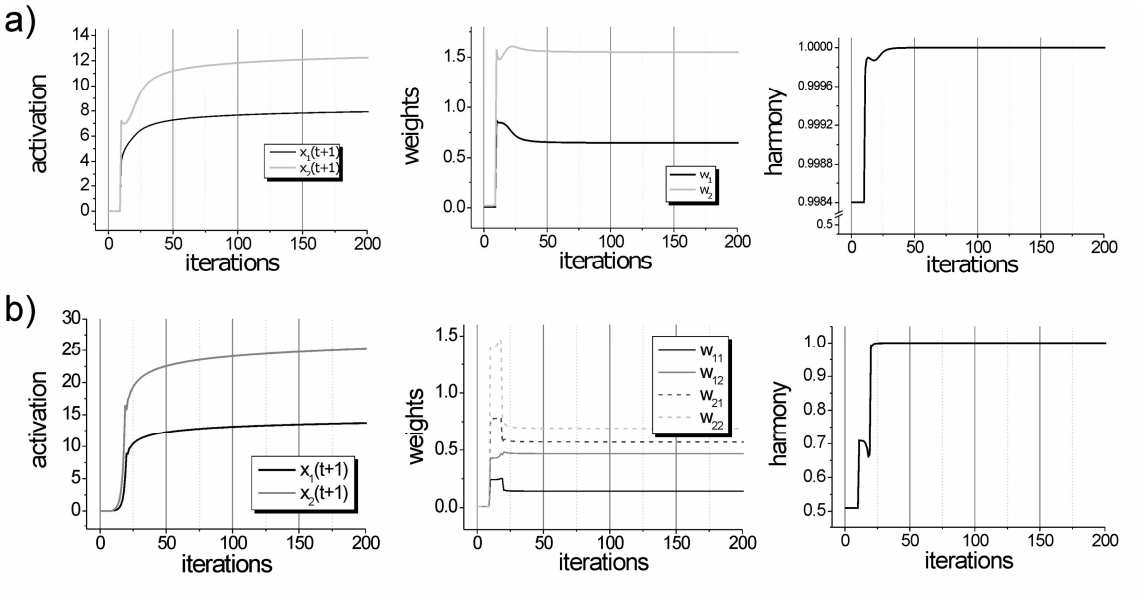


**Figure 2.4: Example of a network trained using the *Dynamic Delta Rule* with two different starting conditions; both simulations were run with the same input vector $\mathbf{x} = (5, 9)$. Upper plots (a): The damping factors $d_i$ are fixed to $d_i = 4$ and the learning rate to $\eta = 0.5$. Lower plots (b): The damping factors $d_i$ are free and thus four weights are variable. Here, a learning rate $\eta = 0.002$ is applied. In both cases the left plot shows the output activation of the network, the middle plot the changing of the weights and the right plot the value of the Harmony function.**

In each of the simulations the weights $w_{ij}$ were initialised with random values $0 < w_{ij} < 0.02$. Input values are chosen in a range of $1 < x_i < 10$ with $x_i \in \mathbb{N}$. These inputs resemble the activation of the respective units caused by the external environmental stimuli. Examples for both cases are given in Figure 2.4 (upper plots: fixed damping factors, lower plots: all four weights free). In both cases the output activation (first column) stabilises at values different from zero and the Harmony function approximates the value of one (last column) which indicates that the MMC criterion is fulfilled. Thus in the end, the external situation is represented within the connections of the network.

The example of Figure 2.4 shows that we need very different learning rates to stabilise the system for the two versions of the task in spite of using the same input vector $\mathbf{x} = (5, 9)$. The simulation shown in Figure 2.4a (two free weights, upper plots) is run with a learning rate $\eta = 0.5$. For this input configuration the network stabilises when choosing $\eta$ in a range of $0.5 \leq \eta \leq 0.7$. The results for a situation with four free weights (Figure 2.4b, lower plots) are obtained by a learning rate of $\eta = 0.002$.

Here we touch a general problem of online-learning in recurrent neural networks: It couples two dynamics, the dynamics of the recurrent network as well as the dynamics of the effects of learning (Steil, 1999). A learning rate that is too large can easily cause the network to diverge. On the other hand, a learning rate that is too small results in output activities decreasing to zero. Therefore, the learning rate $\eta$ is a critical parameter.

A mathematical proof of convergence appears to be a difficult task due to the nonlinearity of the system. Even without applying an online-learning mechanism it is not easy to describe the dynamics of a recurrent neural network analytically even it consists of two units only (Haschke et al., 2001). Nevertheless, based on many numerical tests we can make some general, qualitative remarks on how to adapt the learning rate $\eta$.

The value of an appropriate learning rate does not only depend on the number of free weights but also on the amplitude of the input signals. The larger the inputs $x_1$ and $x_2$ are the smaller the learning rate $\eta$ has to be chosen to yield stability of the network. For the case with fixed damping factors $d_i$ we found that the network stabilises for learning rates between about 0.3 and 1.5 depending on the input vector chosen if at least one input is larger than five. But if both inputs are smaller than five, much larger learning rates have to be applied – for example, for an input vector $\mathbf{x} = (3, 4)$, stabilisation is achieved by using learning rates in a range of $3.2 \le \eta \le 5.6$.

If the difference between both input values is too large, for example when choosing $x_1 = 2$ and $x_2 = 8$, no stable solution can be found if both weights are trained with the same learning rate. But if the learning rate is adapted independently to the input size, stabilisation can be achieved. This is possible, if the learning rate of the weight, by which the larger input is multiplied, is chosen smaller than the learning rate of the weight, by which the smaller input is multiplied. This is due to the fact that the network dynamics depend on the strength of activation of the individual units and that, on the other hand, the training mechanism has to counteract the contribution of the units. In the case of an input vector $\mathbf{x} = (2, 8)$, for example, a possible solution is to set $\eta_1 = 0.9$ to train $w_1$ and $\eta_2 = 6$ to train $w_2$. Qualitatively the product of input activation and learning rate appears to be the relevant factor.

In the situation with four free weights, learning rates leading to stable states have to be about two orders of magnitude smaller than in the case of only two free weights. Additionally, the ranges of appropriate learning rates become very narrow. In the case presented in Figure 2.4b, for example, only then does the network stabilise, if $\eta$ is approximately in the range between 0.002 and 0.003. The task can, however, be simplified considerably, if the weights have upper and lower bounds, which is a physiologically plausible assumption. By applying bounds of $0.001 \leq w_{ij} \leq 3$, the network is prevented from diverging. If the weights reach the bounds during training, oscillations occur, but – for a broad range of learning rates – the network stabilises after some time.

If the dimension of the input vector is increased, it becomes more and more difficult to find appropriate learning rates that stabilise the output of the network due to the above-mentioned high non-linearity of the system. In general, at least for low dimensional input vectors the DD Rule is able to stabilise the system at values different from zero even in spite of little noise in the range of $[-0.01; 0.01]$ given to the input, if some constraints are fulfilled. But with respect to the task this low-dimensional case is sufficient: A mental model of an overall environmental situation consisting of segmental situations each of which comprises only a small number of objects can successfully be built up within the network even if the overall situation contains a large number of such segmental situations.

## 2.4 Transformation of static into sequential information

To accomplish the second part of the goal as specified above, these static mental representations have to be utilised to generate sequences of behaviour, i.e. the *linearization problem* (Levelt, 1989) has to be solved. There are many different examples of behavioural chains varying in complexity like, of course, language. Think of our example of the girl uttering *'Mommy book'* mentioned in the Introduction. But also sequences of movements like the grooming behaviour of rats and other rodents can have syntax-like properties (e.g. Berridge et al., 1987); thus they also face a kind of *linearization problem* as different behavioural subparts – in this case these are not

spoken words but movements of limbs – have to be ordered into a coherent sequence. Yet in 1951 Lashley widened this view in saying that in almost every cerebral activity the problem of temporal sequences, i.e. syntax, can be found.

### 2.4.1 Accessibility

The sequence might be either hardwired within the network as it may be the case in a quite stereotyped behaviour like grooming or, in more variable behavioural sequences, the decision on the temporal order of the items may not only depend on prescribed rules but also on the actual context. If an organism is, for example, faced with a food source and a mate it should decide on the context – its own state and the environmental situation – if it would be better to eat or to reproduce first. Here we want to focus on a more variable context-dependent behaviour, namely the production of language – like in the example of the young girl.

In linguistic literature different factors are discussed which have an impact on word order. For example, animate entities tend to occur earlier in sentences than inanimate entities (e.g. Harris, 1978) and there seems to exist a bias towards the order direct object – indirect object (Bock and Brewer, 1974) that already appears in early language acquisition (Osgood and Zehler 1981). These assumptions can be summed up in a more general hypothesis, the *focus-of-attention hypothesis* (Johnson-Laird 1968a, b): The more an item is in the focus of attention, i.e. the more salient it is for a speaker, the earlier it is produced within the sentence. This idea appears to directly contradict the empirical evidence for a *given – new* ordering: A new piece of information generally is placed second in a sentence (e.g. Smith, 1971; Clark and Clark, 1977; concerning *given – new* ordering in spatial reasoning tasks see Hörnig et al., 2005). As new information is supposed to be more important, this seems to reverse the focus-of-attention hypothesis.

Different attempts have been made to cope with this contradiction (for a review, see Bock, 1982). A simple way to overcome this difficulty is to regard the accessibility of the items to be a main factor which influences the word order in sentences. *Accessibility* is used here in the sense of cognitive psychology (Tulving and Pearlstone, 1966): It refers to the ease of recall by which information could be activated from memory. In this sense, highly accessible elements tend to come first (Levelt, 1999). If the recall of a certain element of memory is facilitated either by the focus of attention or by preceding activation due to the context of discourse, it can be processed earlier. If we assume the

influence of the activation of previous situations to be larger than the focus of attention, a given memory element can – whenever a corresponding context exists – be produced earlier while new and thus important information moves more to the back. But if there is no contextual influence, the focussed information can be the first in sentence production. Many different hints coming from psychological literature confirm the effect of accessibility on word ordering (Bock, 1982) as for example the advantage of the First Mention, i.e. the element mentioned first in a sentence is more accessible than the second one (Gernsbacher and Hargreaves, 1988; Gernsbacher, 1989).

Thus, the principle of accessibility provides a very simple criterion for generating sequences. Needless to say, that this approach cannot explain the complete syntax of any human language. But this principle seems to be a reasonable first approximation to solve the problem as to how sequences could be produced from the items represented within a mental representation.

### 2.4.2 Accessibility in MMC networks

How can the information concerning the accessibility of certain items be coded within the network? The simplest way is to represent it in the form of the activation levels of the single units representing the items of the given situation: The higher the units coding for a special part of the behavioural chain is activated the earlier it will be produced. Bock (1982), for example, proposes in her model that information, which is more accessible and thus more activated, can be processed faster (see also Zwaan et al., 2000). As explained above, the lexical entities vary with respect to the accessibility. Therefore, we assume a separate, internal system to exist which influences the accessibility of the lexical entities and thereby determines their activation. The information passes this system before it is processed within the MMC network. Some units can, for example, be pre-activated because of the given context and thus set to a higher level of activity by this accessibility system. Figure 2.5 shows how the MMC network building the mental representation can be expanded by such a system.

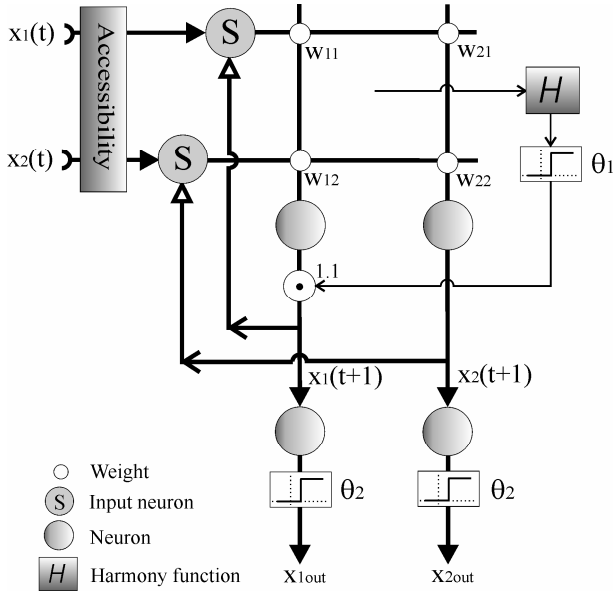


**Figure 2.5: Architecture of a model serving for the task of transforming the static information of the mental representation into time-dependent sequences. The internal system, which determines the activation level of the units according to their accessibility, is symbolised by the shaded grey box on the left ("accessibility"). In the model shown in this figure, the activation of unit $x_1$ is amplified (by the factor 1.1 in this example), when the MMC criterion is fulfilled, i.e. after the Harmony *H* exceeds a given threshold $\theta_1$. If the amplified activation of a unit exceeds a threshold $\theta_2$, the corresponding output neuron elicits an action. For further information see Figure 2.2.**

### 2.4.3 From static to sequential information

One possibility of how the static information represented by the activation levels of the units could be converted into time-dependent sequences is to use a WTA (Winner-takes-all) network, which is attached to the output of the MMC network and selects always the highest activated unit to trigger an action after the MMC criterion is fulfilled, i.e. after the Harmony function *H* has approximated the value one to a sufficient degree.

Since, however, as for this solution a second network is needed, we propose a simpler solution that is depicted in Figure 2.5. After the MMC criterion has been accomplished (Figure 2.5 $\theta_1$), at least one unit is chosen randomly; in Figure 2.5 this is the unit $x_1$. The activation of this unit is continuously increased by an arousal signal that is used to elicit the behaviour. Due to the recurrent connections of the network this influence increases the activation of all other units, too. Now we can define a threshold $\theta_2$ in the subsequent motor units. The unit with the strongest activation reaches this threshold $\theta_2$ first and therefore triggers the corresponding action first. Further application of the arousal signal then drives the unit with the second strongest activation above the threshold. After all

units having exceeded the threshold, the arousal signal can be stopped. Thus, the motor units can elicit actions sequentially.

An example of a simulation showing such behaviour is given in Figure 2.6. Here, the first part of the figure up to the vertical dotted line (iteration step 250) shows the process of building up the internal representation (cf. Figure 2.4). The two units represent for example the words *book* ($x_1$) and *Mommy* ($x_2$). Due to the amplification the activation of unit $x_2$ exceeds the threshold $\theta_2$ (horizontal dashed line) first and therefore the word *Mommy* can be produced first (left arrow). Some iterations later also the second unit ($x_1$) exceeds the threshold $\theta_2$ and the word *book* is produced (right arrow). In this way it is possible to produce sequences like *Mommy book*.

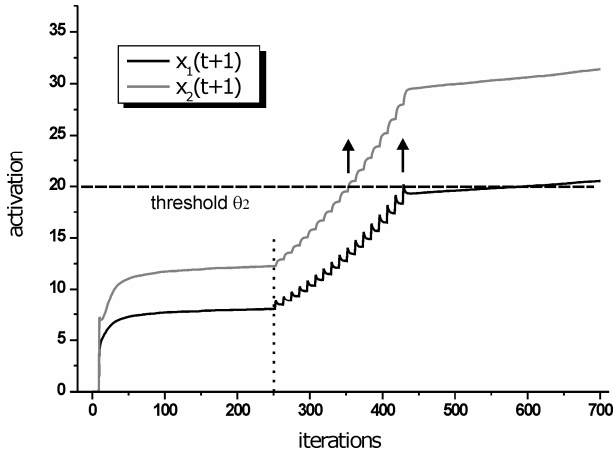


**Figure 2.6: Example of a simulation using the model shown in Figure 2.5. The initial conditions are the same as in Figure 2.4(a). When the MMC criterion is fulfilled (indicated by the vertical dotted line, iteration step 250) the output of unit $x_1$ (black line) is amplified by a factor of 1.1 each tenth iteration. If the activation of a unit exceeds the threshold $\theta_2$ (horizontal dashed line, in this example set to $\theta_2$ = 20), its corresponding output neuron elicits an action (arrow). In this way, unit $x_2$ triggers an action at iteration step 354 and correspondingly unit $x_1$ at iteration step 429.**

This solution has several advantages. First, no second network is needed as in the case when using the WTA network. Second, the motor output can be disconnected from the mental representation by simply inhibiting the arousal signal or by inhibiting the connections to the motor units. Such inhibitory influences during imagined or observed movements have been observed in some studies (Rizzolatti and Arbib, 1998; Jeannerod, 1999). Thus, this model realises the capability of disconnecting the internal mental representation from the motor system, which is considered to be a precondition for the occurrence of cognitive abilities.

## 2.5 Conclusion and Discussion

The main thrust of this chapter was to present a model which can serve as a basis for sequence production, especially for language production based on information received directly from the environment. As we have argued in the Introduction (Chapter 2.1), the first step to cope with this problem is to build up a mental representation as a kind of working memory of the objects present in the environment which in a second step can be used for producing sequential behaviour like, for example, spoken sentences.

Many studies revealed a tight connection between the perception and action system (Rizzolatti and Craighero, 2004) which seems to be an essential property for action understanding (Rizzolatti et al., 2001), controlling motor output (Jeannerod, 1999; Cruse, 2003a), but also for language production and understanding (Glenberg and Kaschak, 2002). Therefore, to account for these findings and to fulfil the bipartite task described, we adapted a model, namely the MMC network, which was primarily invented for action tasks like arm-movements or landmark navigation and which is able to cope with the problem of sensorimotor integration.

### 2.5.1 Mental representations and the linearization problem

The main feature of these networks – converging to stable states – can be exploited to generate mental representations. These attractor characteristics of MMC networks are similar to the attractors of Hopfield type networks (Hopfield, 1982; 1984). But, in contrast to Hopfield networks which only have a limited number of discrete attractors, MMC networks show a smooth attractor space. Referring to the example of the arm movement, every geometrically possible solution can be adopted. Additionally, neither a symmetrical weight distribution nor a bounded activation function is needed.

By means of the Dynamic Delta Rule the weights can be trained in a way that only sub-populations of a larger network are activated according to the respective environmental situation in an online mode; here it is not necessary to disconnect the learning dynamics from the dynamics of the recurrent connections like in other approaches. Thus, application of this learning algorithm serves to cope with the task of generating a kind of *preverbal message* (Levelt, 1989), i.e. a mental representation in which the content of the following language production is bound together. This first step can also be compared with the first process of the so-called *Indexical Hypothesis* – a model for

sentence comprehension – proposed by Glenberg and Robertson (1999): during this *indexing* process the content of the language to be comprehended is verified to determine what or who is being talked about. Hence, the idea of first building up mental representations of the situation is well-established in both language comprehension as well as language production; our model can provide an explanation on neural level for this supposed first step.

Starting from this mental representation the speaker has to solve the *linearization problem* (Levelt, 1989) as he/she has to decide about the order in which the single elements should be produced. Thus, according to Levelt we take here a totally different approach in generating sequences than others dealing with the processing of temporal sequences (e.g. Elman, 1990, see also Porr and Wörgötter, 2003). In these studies networks are trained with sequential input patterns, i.e. the sequence to be stored is given in advance (for a recent physiological study of sequence encoding see for example Jensen and Lisman, 2005). This however is not our goal. In our approach in contrast a behavioural sequence can be produced from a nonlinear internal representation without knowing the temporal order before starting the behaviour – which seems to be reasonable especially when dealing with language production. Training like in Elman networks could however be used in our model to learn grammatical information that may later on be applied by the accessibility system.

### 2.5.2 Scaling the network

As other researchers like Cangelosi and Parisi (1998: 84), who examined the 'evolutionary emergence of a very limited "language" made up of just two one-word utterances', we started to learn and represent situations consisting of a small number of items within our framework and to transform them into sequential actions. Therefore, the system provides a basis for the control of simple chains of behaviour as can be observed for example in language production of young children at the age of about two years (see Chapter 2.1).

If we try to enlarge the representation up to more than three or four different objects which means adding more units to the model, stability problems arise. Although this seems to be a disadvantage at first glance at least from a neurocomputational point of view, the model could provide an explanation for the limited capacity of working memory (Baddeley, 1986): various experiments show that only a relatively small

number of objects of around four can be held in working memory concurrently (Luck and Vogel, 1997; for a review see Cowan, 2000).

Nevertheless it has to be possible to scale the system to problems that require a higher dimensionality as for example longer sentences. How could this be achieved? The studies cited above concerning the capacity of working memory point at a solution: Though the number of objects to be stored within one model is limited, the objects themselves can be integrated, i.e. consist of different features. A similar solution could arise when considering the construction of a somewhat more complex MMC network, which has been used to represent the kinematics of a six-legged insect body with a total of 18 degrees of freedom. The complete network would require a matrix with $276 \times 276$ weights (Kindermann, 2003). However, the network could be simplified dramatically by dividing it into six different nested subnetworks ("chunks"). Each leg is represented within the body module simply by a leg-vector pointing from the body to the foot point. Such a vector could be called a "symbol" of a leg because it comprises the main information about each leg, namely the position of the foot point in relation to the body. The details concerning the leg joints are represented within the leg subnetworks.

A similar approach seems to be reasonable when understanding or producing language: A possible solution to cope with more complex situations could be to subdivide larger structures into many small modules containing chunks or subparts of sentences as the segmental situations we described above. Taking a top-down view, such nested structures could be applied to represent lower level information down to morphemes and phonemes. In this way, the limitation arising from using localist-encoding linguistic units could be overcome. A similar approach has also been proposed by Haschke et al. (2001) to solve the problem of controlling the dynamics of large recurrent neural networks by breaking them down in small modules using a small number of (input) parameters.

Taking a bottom-up view, nested structures could be used to address higher level symbols. Different approaches exist to analyse the structure of sentences. One of these approaches is the so-called *Construction Grammar* (Fillmore, 1988; Goldberg, 1995). Here, constructions are considered to be the basic units of language. No difference is made between lexicon and syntax, as lexical and syntactical constructions mainly vary with respect to their internal complexity while representing the same kind of declarative

data: Both combine form and meaning. Thus, strictly speaking lexical items are also constructions (Goldberg, 1995). As an example let's take the use of the English made-up verb *to floos*: In different expressions like *'X floosed Y the Z'* or *'X floosed Y'* most native speakers of English assign different meanings to these sentences. As the verb itself has no meaning, the construction seems to carry any meaning inherently. Kaschak and Glenberg (2000) provide a test to verify Goldbergs notion of construction; they have shown that not only children – as has been demonstrated in language acquisition studies (e.g. Pinker, 1989) – but also adults are sensitive to the meanings associated with particular constructions (see also Fisher, 1994; Naigles and Terrazas, 1998). This implies that the abstract structure becomes a kind of symbol that is at least in some way independent from the words the construction consists of (Tomasello, 1999). Tomasello (1999) gives a possible explanation, why construction could have become a special form of internally complex language symbols: They could be suitable to react to recurring communicational functions.

Hence we can argue that – like the above-mentioned leg-vectors which could be interpreted as symbols for the legs – it is also reasonable to think of higher symbolic levels like constructions. All these considerations point into a direction how the problem of scaling could be solved: We have to subdivide more complex environmental situations into smaller modules containing subsets of information, as we have already done by activating small segmental situations, which then in turn have to be combined to larger structures.

Two major problems have to be addressed in future work: First, reasonable ways have to be proposed of how many small network modules could be combined to cope with more complex behavioural tasks. The second issue concerns the kind of information that should be represented within neural models to "understand" the world outside. Up to now we only have treated situation models that are static by nature because they are simply built up from objects existing in the environment. Glenberg among others points out that with regard to real understanding it is important to be able to internally simulate not only the objects by themselves but also the *affordances* of objects, i.e. the actions that could be done with objects (Kaschak and Glenberg, 2000). According to Gibson (1966; 1979), who first coined the notion of affordances, for example a chair is a chair

because it affords sitting for adult humans. Therefore, it is necessary to find ways to combine the static representations of external situations with dynamic representations of actions or events.

## 2.6 References

Baddeley A (1986), *Working memory* (Oxford: Clarendon Press).

Berridge KC, Fentress JC and Parr H (1987) Natural syntax rules control action sequence of rats, *Behavioural Brain Research*, 23: 59-68.

Bock JK, (1982) Toward a Cognitive Psychology of Syntax: Information Processing Contributions to Sentence Formulation, *Psychological Review*, 89: 1-47.

Bock JK, and Brewer WF (1974) econstructive recall in sentences with alternative surface structure, *Journal of Experimental Psychology*, 103: 837-843.

Buccino B, Binkofski F, Fink GR, Fadiga L, Fogassi L, Gallese V, Seitz RJ,Zilles K, Rizzolatti G, and Freund H-J (2001) Action observation activates premotor and parietal areas in a somatotopic manner: an fMRI study, *European Journal of Neuroscience*, 13:. 400-404.

Calvin WH (1996) *How brains think*, New York: Basic Books.

Cangelosi A (2001) Evolution of communication and language using signals, symbols and words, *IEEE Transactions in Evolution Computation*, 5: 93-101.

Cangelosi A (2004) The sensorimotor bases of linguistic structure: Experiments with grounded adaptive agents. In *Proceedings of the eighth International Conference on Simulation of Adaptive Behavior* (S. Schaal, A. Ijspeert, A. Billard, S. Vijayakumar, J. Hallam, J.-A. Meyer, eds) Cambridge, MA: MIT Press, pp. 487-496.

Cangelosi A and Parisi D (1998) The emergence of a "language" in an evolving population of neural networks, *Connection Science*, 10: 83-97.

Clark HH and Clark EV (1977) *Psychology and Language*, New York: Harcourt Brace Jovanovich.

Cowan N (2000) The magical number 4 in short-term memory: A reconsideration of mental storage capacity, *Behavioral and Brain Sciences*, 2: 87-114.

Cruse H (2003a) Landmark-based navigation, *Biological Cybernetics*, 88: 425-437.

Cruse H (2003b) The evolution of cognition - a hypothesis, *Cognitive Science*, 27: 135-155.

Cruse H and Steinkühler U (1993) Solution of the direct and inverse kinematic problem by a unique algorithm using the mean of multiple computation method, *Biological Cybernetics*, 69: 345-351.

Cruse H, Steinkühler U and Burkamp C (1998) MMC - a recurrent neural network which can be used as manipulable body model. In *Proceedings of the fifth International Conference on Simulation of Adaptive Behavior,* Cambridge, MA: MIT Press, pp. 381-389.

de Saussure F (1967), *Grundfragen der allgemeinen Sprachwissenschaft,* (C. Bally, ed.), Berlin: De Gruyter, 1967 (original work published 1916).

Decety J, and Sommerville JA (2003) Shared representation between self and other: a social cognitive neuroscience view, *Trends in Cognitive Sciences*, 7: 527-533.

Di Pellegrino G, Fadiga L, Fogassi L, Gallese V and Rizzolatti G (1992) Understanding motor events: a neurophysiological study, *Experimental Brain Research* , 91: 176-180.

Elman JL (1990) Finding Structure in Time, *Cognitive Science*, 14: 179-211.

Fillmore C (1988) The mechanics of "Construction Grammar", *Berkeley Linguistics Society*, 14: 35-55.

Fisher C (1994) Structure and meaning in the verb lexicon: Input from a syntax-aided verb learning procedure, *Language and Cognitive Processes*, 9: 473-518.

Fuster JM (1995) *Memory in the cerebral cortex,* Cambridge, MA: MIT Press.

Gallese V, Fadiga L, Fogassi L and Rizzolatti G (1996) Action recognition in premotor cortex, *Brain*, 119: 593-609.

Gernsbacher MA (1989) Mechanisms that improve referential access", *Cognition*, 32: 99-156.

Gernsbacher MA and Hargreaves DJ (1988) Accessing sentence participants: The advantage of the First Mention, *Journal of Memory and Language*, 27: 699-717.

Gibson JJ (1966) *The senses considered as perceptual systems,* Boston: Houghton-Mifflin.

Gibson JJ (1979) *The ecological approach to visual perception*, Boston: Hougthon-Mifflin.

Glenberg AM and Kaschak MP (2002) Grounding language in action, *Psychonomic Bulletin & Review*, 9: 558-565.

Glenberg AM and Robertson DA (1999) Indexical understanding of instructions", *Discourse Processes*, 2: 1-26.

Goldberg AE (1995) *A construction Grammar approach to argument structure,* Chicago: The University of Chicago Press.

Grafton ST, Fadiga L, Arbib MA and Rizzolatti G (1997) Premotor cortex activation during observation and naming familiar tools", *NeuroImage*, 6: 231-236.

Grezes J, Fonlupt P, Bertenthal B, Delon-Martin C, Segebarth C and Decety J (2001) Does perception of biological motion rely on specific brain regions?, *NeuroImage*, 13: 775-785.

Harris M (1978) Noun animacy and the passive voice: A developmental approach, *Quarterly Journal of Experimental Psychology*, 30: 495-501.

Haschke R, Steil JJ and Ritter H (2001) Controlling oscillatory behaviour of a two neuron recurrent neural network using inputs. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN),* Heidelberg: Springer, pp. 1109-1114.

Hebb DO (1949) *The Organization of Behavior - A Neuropsychologial Theory,* New York: John Wiley & Sons.

Heidelberger M (1998) From Helmholtz's philosophy of science to Hertz's picture-theory. In *Heinrich Hertz: Classical physicist, modern philosopher* (R.S. Cohen and M.W. Wartofsky, eds.), Dordrecht: Kluwer, pp. 9-25.

Hockett C (1960) The origin of speech, *Scientific American*, 203: 88-96.

Hopfield JJ (1982) Neural networks and physical systems with emergent collective computational abilities, *Proceedings of National Academics Sciences USA*, 79: 2554-2558.

Hopfield JJ (1984) Neurons with graded response have collective computational properties like those of two-state neurons, *Proceedings of National Academics Sciences USA*, 81: 3088-3092.

Hörnig R, Oberbauer K and Weidenfeld A (2005) Two principles of premise integration in spatial reasoning", *Memory and Cognition,* 33: 131-139.

Iacoboni M, Woods RP, Brass M, Bekkering H, Mazziotta JC and Rizzolatti G (1999) Cortical mechanisms of human imitation, *Science,* 286: 2526-2528.

Jeannerod M (1999) To act or not to act: Perspectives on the representation of actions, *Quarterly Journal of Experimental Psychology*, 52A: 1-29.

Jensen O and Lisman JE (2005) Hippocampal sequence-encoding driven by a cortical mulit-item working memory buffer, *Trends in Neurosciences*, 28: 67-72.

Johnson-Laird PN (1968a) The choice of the passive voice in a communicative task, *British Journal of Psychology*, 59: 7-15.

Johnson-Laird PN (1968b) The interpretation of the passive voice, *Quarterly Journal of Experimental Psychology*, 20: 69-73.

Johnson-Laird PN (1983) *Mental models: towards a cognitive science of language, inference, and consciousness,* Cambridge: Cambridge University Press.

Jordan MI (1986) Attractor dynamics and parallelism in a connectionist sequential machine. In *Proceedings of the eighth annual conference of the cognitive science society,* Hillsdale: Earlbaum, pp. 531-546.

Kaschak MP and Glenberg AM (2000) Constructing meaning: The role of affordances and grammatical constructions in sentence comprehension, *Journal of Memory and Language*, 43: 508-529.

Kindermann T (2003) *Positive Rückkopplung zur Kontrolle komplexer Kinematiken des hexapoden Laufens: Experimente und Simulation.* PhD thesis, University of Bielefeld.

Kindermann T and Cruse H (2002) MMC - a new numerical approach to the kinematics of complex manipulators" *Mechanism and Machine Theory*, 37: 375-394.

Kohonen T (1982) Self-organized formation of topologically correct feature maps", *Biological Cybernetics*, 43: 59-69.

Lashley KS (1951) The problem of serial order in behaviour. In *Cerebral mechanisms in behaviour* (L.A. Jeffress, ed.), New York: Wiley, pp. 112-146.

Levelt W (1989) *Speaking - From Intention to Articulation,* Cambridge, MA: MIT Press.

Levelt W (1999) Producing spoken language: a blueprint of the speaker. In *The Neurocognition of Language,* (C.M. Brown and P. Hagoort, eds.), Oxford: University Press, pp. 83-122.

Luck SJ and Vogel EK (1997) The capacity of visual memory for features and conjunctions, *Nature*, 390: 279-281.

Mills, AE (1985) The acquisition of German. In *The cross-linguistic study of language acquisition: The Data. Vol. 1,* (D.I. Slobin, ed.), Hillsdale, N.J.: Erlbaum, pp. 141-254.

Naigles, LR and Terrazas P (1998) Motion verb generalization in English and Spanish: Influences in language and syntax, *Psychological Science*, 9: 363-369.

Osgood C and Zehler A (1981) Acquisition of bi-transitive sentences: Prelinguistic determinants of language acquisition, *Journal of Child Language*, 8: 367-383.

Pinker, S (1989) *Learnability and cognition: The acquisition of argument structure,* Cambridge, MA: MIT Press.

Porr B and Wörgötter F (2003) Isotropic sequence order learning in a closed-loop behavioural system, *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, 361: 2225-2244.

Rizzolatti G and Arbib MA (1998) Language within our grasp, *Trends in Neurosciences*, 21: 188-194.

Rizzolatti G and Craighero L (2004) The mirror-neuron system, *Annual Review of Neuroscience*, 27: 169-192.

Rizzolatti, G, Fadiga L, Fogassi L and Gallese V (1996) Premotor cortex and the recognition of motor actions, *Cognitive Brain Research*, 3: 131-141.

Rizzolatti G, Fogassi L and Gallese V (2001) Neurophysiological mechanisms underlying the understanding and imitation of action, *Nature Reviews Neuroscience*, 2: 661-670.

Roskies AL (1999) The Binding Problem, *Neuron*, 24: 7-9.

Smith C (1971) Sentences in discourse, *Journal of Linguistics*, 7: 213-235.

Steels L (1995) Intelligence - dynamics and representations. In *The biology and technology of intelligent autonomous agents*, (L. Steels, ed.), Berlin: Springer, pp. 72-89.

Steil JJ (1999) *Input-Output Stability of Recurrent Neural Networks*, Göttingen: Cuvillier Verlag.

Steinkühler U, Burkamp C and. Cruse H (2000) MMC - a holistic system for a nonsymbolic internal body representation. In *Prerational Intelligence: Adaptive Behavior and Intelligent Systems Without Symbols and Logic* (H. Ritter, H. Cruse and J. Dean, eds.), Dordrecht: Kluwer, pp. 121-140.

Steinkühler U and Cruse H (1998) A holistic model for an internal representation to control movement of a manipulator with redundant degrees of freedom, *Biological Cybernetics*, 79: 457-466.

Tani J and Nolfi S (1999) Learning to perceive the world articulated: an approach for hierarchical learning in sensory-motor systems, *Neural Networks*, 12: 1131-1141.

Tomasello M (1992) *First verbs - A case study of early grammatical development*, Cambridge: University Press.

Tomasello M (1999) *The cultural origins of human cognition*, Cambridge, MA: Harvard University Press.

Tulving E and Pearlstone Z (1966) Availability versus accessibility of information in memory of words, *Journal of Verbal Learning and Verbal Behavior*, 5: 381-391.

van Dijk TA and. Kintsch W (1983) *Strategies in text comprehension*, New York: Academic Press.

Von Eckard B (1983) *What is cognitive science?*, Cambridge, MA: MIT Press.

Wittgenstein L (1958) *Philosophical Investigations*, G.E.M. Anscombe, Transl., New York, NY: MacMillan Publishing Co.

Wolpert DM and Kawato M (1998) Multiple paired forward and inverse models for motor control, *Neural Networks*, 11: 1317-1329.

Wolpert DM, Ghahramani Z and Jordan MI (1995) An internal model for sensorimotor integration, *Science*, 269: 1880-1882.

Zwaan RA, Madden CL and Whitten SN (2000) The presence of an event in the narrated situation affects its availability to the comprehender, *Memory & Cognition*, 28: 1022-1028.

Zwaan RA and Radvansky GA (1998) Situation models in language comprehension and memory, *Psychological Bulletin*, 123: 162-185.

# 3 Modelling Memory Functions with Recurrent Neural Networks consisting of Input Compensation Units: I. Static Situations

Humans are able to form internal representations of the information they process – a capability which enables them to perform many different memory tasks. Therefore, the neural system has to learn somehow to represent aspects of the environmental situation; this process is assumed to be based on synaptic changes. The situations to be represented are various as for example different types of static patterns but also dynamic scenes. How are neural networks consisting of mutually connected neurons capable of performing such tasks?

Here we propose a new neuronal structure for artificial neurons. This structure allows to disentangle the dynamics of the recurrent connectivity from the dynamics induced by synaptic changes due to the learning processes. The error signal is computed locally within the individual neuron. Thus, online learning is possible without any additional structures. Recurrent neural networks equipped with these computational units are able to cope with different memory tasks. Examples illustrate how information is extracted from environmental situations comprising fixed patterns to produce sustained activity and to deal with simple algebraic relations.

## 3.1 Introduction

From early childhood on humans brains have a fundamental ability: they build up representations. Brains and their constituents, the neurons, are specialised to represent aspects of the environment which means that these neurons or groups of neurons "stand for" those aspects. This information coded within neural circuits can be multifaceted. Information of objects like a tree or a chair can as well be represented as rules, for example underlying grammar in language, or dynamic events like the movement of one person towards another. To start with we want to focus on the two first examples: We propose a new neuronal architecture that is able to deal with these problems. Its ability to represent dynamic situations is treated in Chapter 4.

A basic function of our brain is to provide some kind of working memory (Baddeley, 1986, 1992). It allows us to hold representations of external information actively in memory, at least for a short time, to be able to act within and react to the world. In various experiments the properties of working memory have been investigated applying so-called delayed response tasks. The pioneer work has been done by Fuster and Niki (Fuster and Alexander, 1971; Fuster, 1973; Niki, 1974a, 1974b). In continuing this work many studies using electrophysiological recordings show a stimulus-specific, enhanced delay activity in several brain areas (for reviews see Fuster, 1995; Miyashita and Hayashi, 2000; Wang, 2001). This sustained internal activity in the absence of the external stimulus is argued to be the neural substrate of working memory.

Another important capability human brains have is representing rules. This becomes apparent when regarding language learning. Marcus et al. (1999) have shown that statistical learning mechanism – which are, of course, not called into question to exist – do not exhaust the child's repertoire of learning mechanisms. They performed experiments showing that already 7 month old babies are able to extract simple algebraic relations from acoustic input. The babies were able to distinguish between three word sentences consisting of made-up words and following either the condition "ABA" or "ABB". As the test words were totally new and the sentences were of the same length the babies could not distinguish them based on transitional probabilities or statistical properties.

Representing such algebraic relations means representing "open-ended abstract relationships for which we can substitute arbitrary items. For instance, we can substitute any value of $x$ into the equation $y = x + 2$." (Marcus et al., 1999; see also Chomsky, 1980; Pinker and Prince, 1988; Pinker, 1991; Marcus et al., 1995; Marcus, 2001). The point made in the study is that it is not only the capability of generalising due to statistical learning mechanisms which enables us – just like the young babies – to build correct sentences as described but especially the capability of representing the underlying general rule: It is important to be able to represent such rules.

For many of the different abilities of brains computational models have been proposed. The most promising among them are models with recurrently coupled neurons because they seem to resemble natural neuronal assemblies best. As the tasks mentioned require an internal representation of the current external situation, some form of learning is

necessary. To model example-based learning different forms of error backpropagation (Rumelhart et al., 1986; Hertz et al., 1991) are widely used training procedures for both feed-forward and recurrent neural networks (RNNs). But backpropagation is often considered to be biologically implausible because the error signal has to be provided externally and a specific additional network is required that is able to propagate these error signals.

Additionally, most artificial recurrent networks exposed to learning situations suffer from two severe problems. On the one hand, training is particularly difficult in RNNs because two different dynamics are intertwined: There is the dynamics of the RNN itself, the properties of which depend on distribution and size of the weights. If, on the other hand, these weights are changed additionally due to the learning procedure, a second dynamic process is introduced that interacts with the first one. Therefore, neural and synaptic dynamics are coupled in a very intricate way (Del Giudice et al., 2003) making the control of the network a hard problem (Steil, 1999). This difficulty is often solved by application of off-line training procedures, that separate the dynamics of the network from the dynamics of the training procedure like in Contrastive Hebbian Learning (Movellan, 1990, Baldi and Pineda, 1991; Xie and Seung, 2003) or training echo state networks (Jaeger and Haas, 2004), or by hand-tuning the parameters (e.g. Seung et al., 2000). But neither a cut-off of the feedback loop nor hand-tuning seems to be biologically plausible. Online learning algorithms like real-time recurrent learning (e.g. Williams and Zipser, 1989b), in contrast, are often very slow and computationally very expensive concerning storage capacity and computation time (see (Williams and Zipser, 1989b; Schmidhuber, 1992; Doya, 1995). Furthermore, they are non-local and would require a large additional network structure when being applied to biological systems.

In this chapter, we propose a new biologically inspired computational circuit of a neuronal unit called *Input Compensation Unit* (IC Unit) which disconnects the dynamics of the recurrent network from the dynamics due to the learning procedure and therefore allows for an easy training of RNNs in an online mode to model the two tasks mentioned above – i.e. holding an item in memory which means learning the representation of static patterns, and representing simple algebraic relations.

Additionally it is possible that a network equipped with those units is also able to learn dynamic situations. This is described in Chapter 4.

The circuit acts within a neuronal unit and incorporates a learning rule that formally corresponds to the delta rule (Widrow and Hoff, 1960), but does not require a separate network for backpropagating the error. Each neuron only needs local information directly available via its synaptic connections. The error is determined within each neuron. Therefore, the training procedure is unsupervised as no global trainer is necessary and each neuron relies on local information only. Consequently, the computational costs are very low. Thus, our model overcomes the main objections against traditional approaches in training recurrent neural networks. A very similar rule has been proposed by (Kalveram, 2000) for training feedforward networks. The difference to our approach is discussed below (Chapter 3.5).

The final goal behind this approach is to design a memory system that contains the representation of many different situations. Such situations may comprise static or moving objects or describe connections between a sensory input and a motor output, analogue to so-called motor primitives as proposed by Wolpert and Kawato (1998), for example. The view, that different situations are stored by specific networks, is supported by physiological findings (Fogassi et al., 2005). Studying mirror neurons, i.e. neurons which likewise represent sensory as well as motor aspects, Fogassi and colleagues (2005) have shown that different neurons are activated when movements are either observed or performed that are similar but of different meaning (e.g. eating or placing). In this chapter we do not deal with the question of how cooperation or selection of different situation models may be organised, but first concentrate on the basic structure of such situation models.

In the following (Chapter 3.2) we want to specify the tasks in more detail the network should be able to deal with. The structure of the circuit proposed is described in Chapter 3.3. After having presented the results (Chapter 3.4) the chapter concludes with a discussion of the networks' properties including some biological interpretations (Chapter 5).

## 3.2 The tasks

### 3.2.1 Learning a static pattern to produce sustained activity

The first task the network should cope with is to represent a fixed static pattern consisting of analogue values that is given as input to produce sustained activity even if the input pattern disappears. Specifically, the task is as follows: The recurrent network consists of at least *n* units. As an example a network for $n = 3$ is depicted in Figure 3.1a. Any *n*-dimensional input vector is provided to the network. The learning algorithm should change the weights in a way that all units of the network adopt activations that correspond to the input and maintain their activation even after the external input is switched off.

Which values should the weights take if a fixed input vector is presented? Assume that we have a network with *n* units with output values $x_1$, $x_2$, … $x_n$ and the input vector consists of the components $\mathbf{a} = (a_1, a_2, ...a_n)^T$. The task is then to find a weight matrix **W** with $\mathbf{a} = \mathbf{W} \cdot \mathbf{a}$. This means that the weights of the recurrent network should form a matrix that has the vector $(a_1, a_2, ...a_n)^T$ as an eigenvector corresponding to the eigenvalue $\lambda = 1$, while all other eigenvalues satisfy $|\lambda| < 1$. As we have $n^2$ weights there is a manifold of matrices that fulfil this condition. *n* equations determine *n* degrees of freedom. Therefore, $(n^2 - n)$ of the $n^2$ weights can be chosen arbitrarily. For $n = 3$ one possible solution is given by matrix W1:

$$1/(a_1 + a_2 + a_3) \cdot \begin{pmatrix} a_1 & a_1 & a_1 \\ a_2 & a_2 & a_2 \\ a_3 & a_3 & a_3 \end{pmatrix} \qquad \text{(W1)}$$

With $\mathbf{1} = (1,1,1)^T$, W1 can be rewritten as $(1/(\mathbf{1}^T \cdot \mathbf{a})) \cdot \mathbf{a} \cdot \mathbf{1}^T$. W1 is a skew projector. It projects onto $span\{\mathbf{a}\}$ along the space that is orthogonal to **1**. Such a network does not only stabilise an input situation given by vector $(a_1, a_2, a_3)^T$, but any multiple of this vector. If the initial activations of the units are set to values that deviate from this condition, the network relaxes to a vector that obeys this relation, i.e., to a multiple of $(a_1, a_2, a_3)^T$. The network can therefore be described as forming an attractor consisting

of a two-dimensional subspace that is described by the plane $a_1x_1 + a_2x_2 + a_3x_3 = 0$. This network is only neutrally stable. Neutral stability means that if any weight is changed arbitrarily, the activations of the units increase to infinity or may decrease to zero. Therefore, a learning mechanism is needed that automatically stabilises the weights against disturbances as for example disturbances due to synaptic noise.

### 3.2.2 Representing simple algebraic relations

As a further task, the network should be able to store simple algebraic relations. Here, we deal with two examples of such relations: First, the results obtained by Marcus et al. (1999) should be simulated with the network proposed here. Marcus and colleagues found, that the infants tested were able to extract abstract algebra-like rules that represent the relationship between variables such as "the first item X is the same as the third item Y". Two experiments have to be performed: In the first one the network has to be trained with external input of structure "ABA" and in the second one with external input of structure "ABB". The network can be tested afterwards (just like the babies) with consistent input, i.e. input resembling the structure of the training phase, or with inconsistent input. The test input has to consist of variables not yet presented during the training phase to prevent learning based on transitional probabilities. The babies in the experiments described above paid attention to the inconsistent sentences for a longer period of time (for details see Marcus et al., 1999).

The second task to be learnt by the network is more general by nature: It should be able to represent simple linear equations. The network should be able to sum up two variables, i.e. to represent all possible configurations of $x_1$ and $x_2$ that result in a value $x_3 = x_1 + x_2$. If we do not wish to apply a 3D look-up table for all possible cases, the mechanism, i.e. the underlying rule or equation, should be represented which can then be applied to any given values. For this specific example, an easy solution is to use two input units $x_1$ and $x_2$, the output of which is fed in as input to a third unit, with weights of unity. However, there are two tasks related tightly: The task $x_3 = x_1 + x_2$ also implies that $x_1 = x_3 - x_2$ and $x_2 = x_3 - x_1$. Of course, two further independent networks could be constructed that can solve these additional tasks. This solution would require a kind of selector network that decides which of the three networks should be used depending on the task given.

A simpler solution is to form one "holistic" network that represents the complete situation and can solve all three tasks. This recurrent network is given by the equation $\mathbf{x}(t+1) = \mathbf{W} \cdot \mathbf{x}(t)$ or, for $n = 3$, by:

$$x_1(t+1) = w_{11} \cdot x_1(t) + w_{12} \cdot x_2(t) + w_{13} \cdot x_3(t)$$
$$x_2(t+1) = w_{21} \cdot x_1(t) + w_{22} \cdot x_2(t) + w_{23} \cdot x_3(t)$$
$$x_3(t+1) = w_{31} \cdot x_1(t) + w_{32} \cdot x_2(t) + w_{33} \cdot x_3(t)$$

Here, $\mathbf{x}(t)$ is the vector describing the actual activation of the $n$ units ($n = 3$ in our case) and $\mathbf{x}(t+1)$ the vector describing the activation in the following time step. **W** describes the $n^2$ weights $w_{ij}$ ($i = 1 \text{ to } n$, $j = 1 \text{ to } n$). If the weights are chosen appropriately, this system has stable solutions that fulfil the equation $x_1 + x_2 - x_3 = 0$. An appropriate weight matrix is given by matrix W2:

$$\begin{pmatrix} 0 & -1 & 1 \\ -1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \qquad \text{(W2)}$$

The tasks regarded here can be understood as pattern completion tasks: Given any two values as input, after relaxation the network will provide all three values $x_1$, $x_2$, and $x_3$ at the output, i.e., a correct solution in any case. Therefore, depending on the input variables chosen, any of the three subtasks can be solved by this network. A correct solution is even found if only one input value is defined. As this latter task is underconstrained, different solutions are possible. The solution actually chosen by the network depends on its earlier state.

## 3.3 The model: A recurrent neural network with IC Units

### 3.3.1 Structure of IC Units

In this section we explain the architecture of a network that can cope with both tasks specified above and can, as will be shown in Chapter 4, also treat dynamic situations. To explain the structure of the network and to explicate its individual units let us

consider a network that consists of $n$ recurrently connected units. An example of a three-unit network is shown in Figure 3.1a.

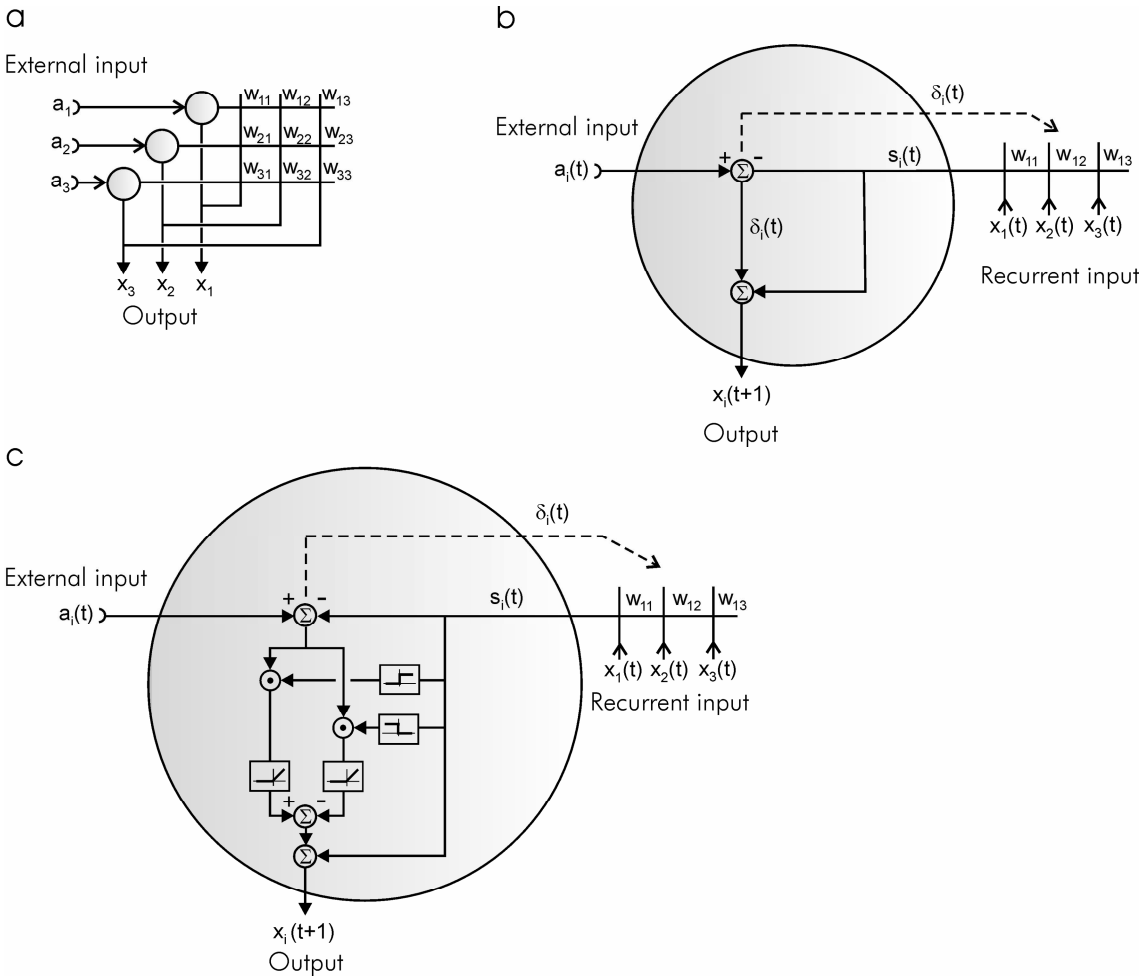


**Figure 3.1: (a) Schematic drawing of a three-unit recurrent network; $a_i$ is the external input, $x_i$ the recurrent input and $w_{ij}$ are the weights. (b) Architecture of one linear IC Unit; $s_i(t)$ is the weighted sum of the recurrent inputs and $\delta_i(t)$ the difference between the external input $a_i(t)$ and $s_i(t)$. (c) Architecture of one IC Unit with the nonlinear extension (see text for explanation).**

Each individual neuron $x_i$ $(i = 1 \text{ to } n)$ is equipped with a special internal structure (Figure 3.1b) described in the following. The dendritic tree is partitioned into two regions: One region with fixed synapses, whose presynaptic neurons belong to sensory neurons transmitting the external input $a_i$. To simplify matters each neuron can only be stimulated by one external stimulus. As the synaptic weight is fixed it is not specified in Figure 3.1b and 1c. The second dendritic region is characterised by active synapses $w_{ij}$, whose presynaptic neurons are components of the recurrent network (Figure 3.1a) and

are recurrently connected to neuron $x_i$. *Active synapses* are synapses which can be either potentiated or depressed (Montgomery and Madison, 2004) and thus are exposed to learning. Therefore, the activation of a single neuron is determined by an external component $a_i$ and an internal component, the weighted sum of the internal recurrent inputs $s_i$. The weighted sum of the internal recurrent inputs of neuron $x_i$ is given by

$s_i(t) = \sum_{j=1}^{n} w_{ij}(t) \cdot x_j(t)$ or, for the complete network, $\mathbf{s}(t) = \mathbf{W}(t) \cdot \mathbf{x}(t)$.

Such a splitting in an external and a recurrent component can also be found in the model described by Del Giudice et al. (2003).

### 3.3.2 Training the synaptic weights

The overall goal in both tasks mentioned above is to represent the external situation **a** (a static pattern or several examples following an algebraic relation) perceived via the sense organs within the network. 'Representing the external situation' can be defined as follows: If the weighted sum of the internal recurrent inputs $s_i$ of neuron $x_i$ equals the external input $a_i$, this stimulus is represented within the network because then the external input is no longer needed to elicit the activation characterising the stimulus $a_i$. In order to reach this goal the synaptic weights $w_{ij}$ have to be adapted in a learning process.

As has been mentioned above, a major problem with training RNNs is that the dynamics of the network are superimposed by the dynamics due to the learning procedure. Both dynamics could however be separated, if, during training, the overall output $x_i$ would always equal the external input (i.e. $x_i = a_i$) independent of the actual learning state, i.e., independent of the actual values of the weights $w_{ij}$. This can be achieved if we determine the output $x_i$ by

$$x_i(t+1) = a_i(t) = s_i(t) + a_i(t) - s_i(t) = s_i(t) + \delta_i(t) \qquad (1)$$

with $\delta_i(t) = a_i(t) - s_i(t)$. The corresponding circuit is shown in Figure 3.1b (solid lines).

To attain the overall goal, the weights $w_{ij}$ have to be changed such that $x_i(t+1) = s_i(t)$ or, in other terms, $\delta_i(t) = 0$. This can be obtained by application of the learning algorithm

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij} \text{ with } \Delta w_{ij} = \varepsilon \cdot x_j(t) \cdot \delta_i(t) \qquad (2)$$

with $\varepsilon > 0$ being the learning rate (for more detailed information about the choice of $\varepsilon$ see Appendix in Chapter 3.6). This learning algorithm formally corresponds to the delta rule. However, in contrast to the traditional approach, the delta error is here assumed to be determined and propagated locally within each neuron (Figure 3.1b, dashed arrows) as has been proposed by Kalveram (2000) for feedforward networks or Jaeger and Haas (2004) for echo state networks. Application of the rule depicted in equation (2) leads to a weight change until $\delta_i(t) = 0$, i.e., until the sum $s_i$ of the weighted recurrent inputs equals the external input $a_i$. We call units with this internal structure *Input Compensation Units* (IC Units), because this circuit compensates the effect of the external input, independent of the actual state of the recurrent weights.

To be able to address this memory content later, it is necessary to prevent the network to automatically adapt to each new input situation. Thus, once the synaptic connections have learnt the specific input situation, further learning is stopped. A simple solution is to finish learning after the error $\delta_i$ has fallen below a given threshold because then external situation is represented within the network. To simplify matters, in the simulations shown here further learning is stopped, if the summed squared error $E(t) = \sum_{i=1}^{n} \delta_i(t)^2$ of the entire network has fallen below a given threshold.

### 3.3.3 Extension of the neuronal structure

To account for working memory capabilities, it should also be possible to sustain the activation once induced by a stimulus. As explained above the overall output of an IC Unit as shown in Figure 3.1b will, however, decay to zero after the external stimulus vanishes. This is due to the property of the IC Units, that the output always equals the input. Thus, the network cannot remain active to act as working memory.

In order to be able to sustain the activation, the architecture requires an extension. If $a_i$ is smaller than $s_i$ in a unit shown in Figure 3.1b the output activation $x_i$ decreases, because then negative $\delta$-values (recall that $\delta_i(t) = a_i(t) - s_i(t)$) are added to $s_i$. This effect can, however, be avoided if we rewrite equation (1) by using rectifiers, which means that only the positive part of the function is transmitted. The rectifier is marked by a $_+$ in the following equations.

For an explanation, we will first consider only positive input values ($a_i(t) \geq 0$). If the weights are small at the beginning of training, for example zero, which means that $s_i(t_0) = 0$, we can assume that during training the condition $0 \leq s_i(t) \leq a_i(t)$ is fulfilled which is biologically plausible. With this assumption, the condition $a_i(t) \geq 0$ can be replaced by $s_i(t) \geq 0$ and equation (1) can be rewritten:

$$x_i(t+1) = s_i(t) + [a_i(t) - s_i(t)]_+, \qquad \text{for } s_i(t) \geq 0 \tag{3.1}$$

Following (3.1), $x_i$ still corresponds to $s_i$, even if $a_i(t) < s_i(t)$. Therefore, using this rectifier, the external input can indeed be switched off after training is finished, i.e. $a_i(t) = 0$, and no changes occur to the output (if training has not yet been finished completely, the activation of the units will slowly decrease to zero, see Discussion in Chapter 3.5). Note, that the rectifiers do not influence the $\delta$-value used for learning.

Furthermore, we can generalise this condition for negative input values ($a_i(t) \leq 0$): If we again assume that the weights are small at the beginning of learning, for example zero, we can state $0 \leq |s_i(t)| \leq |a_i(t)|$, because during learning $s_i$ will approach $a_i$ starting from zero also for negative input values $a_i$. Correspondingly, we can now replace the condition $a_i(t) \leq 0$ by $s_i(t) \leq 0$. This leads to the second equation

$$x_i(t+1) = s_i(t) - [-a_i(t) + s_i(t)]_+, \qquad \text{for } s_i(t) \leq 0 \tag{3.2}$$

Both equations (3.1) and (3.2) are depicted in the circuit diagram in Figure 3.1c. The condition $s_i(t) \geq 0$ and $s_i(t) \leq 0$ are represented by the clipping functions. The two

rectifiers used in equations (3.1) and (3.2) are depicted in the lower part of the circuit (Figure 3.1c). This circuit fulfils three requirements:

(i) It allows to apply both positive and negative input values $a_i$.

(ii) After training is finished, it maintains its activation after the external input has been switched off.

(iii) It shows the same training properties as the linear version (Figure 3.1b), if the condition $0 \le |s_i(t)| \le |a_i(t)|$ is fulfilled.

The results shown in the following were obtained by using this expanded network. Note that the nonlinear expansions applied are only necessary for being able to use the network after learning is finished, i.e. in the testing mode. The learning procedure as such can still be described by a linear approach. As before and during training the activations of the neurons are only determined by the external input values $a_i$ due to their *input compensation* property, the dynamics resulting from the weight changes do not affect the dynamics of the complete network and therefore do not cause stability problems.

## 3.4 Results

### 3.4.1 Learning a static pattern to produce sustained activity

***Training the network.*** Let us first consider the case of a network consisting of three units that receives an external, fixed input vector $(a_1, a_2, a_3)^{\mathrm{T}}$. Numerical investigations reveal the following results which can also be proven to hold generally (see Appendix in Chapter 3.6).

If all nine weights including the diagonal weights, by which each neuron influences itself directly, are allowed to be learnt and all weights are set to zero at the beginning, the IC learning procedure (Figure 3.1, Eq. (2)) provides the solution shown by matrix W3

$$1/\left(a_1{}^2 + a_2{}^2 + a_3{}^2\right) \cdot \begin{pmatrix} a_1a_1 & a_1a_2 & a_1a_3 \\ a_2a_1 & a_2a_2 & a_2a_3 \\ a_3a_1 & a_3a_2 & a_3a_3 \end{pmatrix} = \left(1/\left(\mathbf{a}^{\mathrm{T}}\mathbf{a}\right)\right) \cdot \mathbf{a}\mathbf{a}^{\mathrm{T}} \qquad \text{(W3)}$$

Matrix W3 is the orthogonal projector onto $span\{\mathbf{a}\}$. In geometrical terms, the behaviour of an individual unit *k* can be described as follows: Assume the network consists of *n* units and is trained with a vector **a**. The output of unit *k* is determined by

$$x_k(t+1) = w_{k1}x_1(t) + w_{k2}x_2(t) + ... + w_{kn}x_n(t),$$

which describes a linear function in an $(n+1)$-dimensional space. This function corresponds to an *n*-dimensional hyperplane that contains the origin and, after training, the $(n+1)$-dimensional vector $(a_1, a_2, ...a_n, a'_k)^T$. $a'_k$ and $x'_k$ describe the additional dimension given by the output value $x_k(t+1)$. This hyperplane also contains the $(n-1)$-dimensional subspace that is contained in the *n*-dimensional space ($x_1$ to $x_n$). This subspace is orthogonal to vector $(a_1, a_2, ...a_n)^T$. In other words, this hyperplane could be constructed in the following way: The hyperplane defined by $x'_k = 0$ is rotated around the vector orthogonal to $(a_1, a_2, ...a_n)^T$ until it contains the vector $(a_1, a_2, ...a_n, a'_k)^T$. For $n = 2$ and $k = 2$, this process is schematised in Figure 3.2.

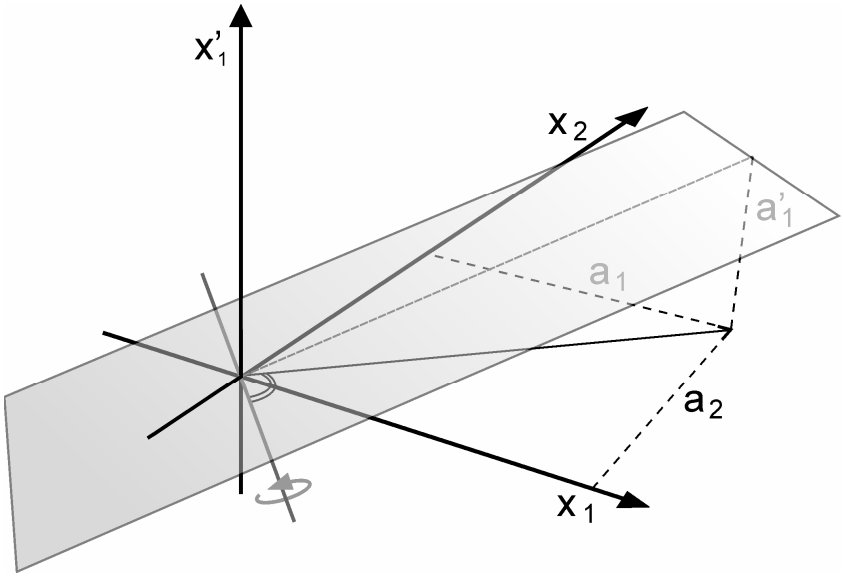


**Figure 3.2: Geometrical illustration for the process of training a two-unit network. The axis around which the plane is rotated is denoted by the grey arrow.**

The network adopts solution W4 (for a proof see Appendix in Chapter 3.6), if during training all diagonal weights are constantly set to zero:

$$\begin{pmatrix} 0 & a_1a_2/\left(a_2^{\,2}+a_3^{\,2}\right) & a_1a_3/\left(a_2^{\,2}+a_3^{\,2}\right) \\ a_1a_2/\left(a_1^{\,2}+a_3^{\,2}\right) & 0 & a_2a_3/\left(a_1^{\,2}+a_3^{\,2}\right) \\ a_1a_3/\left(a_1^{\,2}+a_2^{\,2}\right) & a_2a_3/\left(a_1^{\,2}+a_2^{\,2}\right) & 0 \end{pmatrix} \qquad \text{(W4)}$$

In general, matrix W4 is asymmetric. The *n*-dimensional hyperplane described by unit *k* contains the origin and the vector $(a_1, a_2, ...a_n, a_k)^{\mathrm{T}}$, but now contains the $k^{\mathrm{th}}$ coordinate axis instead of the vector orthogonal to $(a_1, a_2, ...a_n)^{\mathrm{T}}$ as was the case for (W3).

Solution (W4) is of practical interest, because starting from this solution, a manifold of solutions can be constructed by replacing the diagonal weights by arbitrary positive values $d_i$ first and then normalising all weights of unit *i* by multiplication with $1/(1+d_i)$. Parameters $d_i$ can be interpreted as damping factors: The larger $d_i$, the slower the network approaches to a stable solution. A special treatment of the diagonal weights is plausible in biological systems, because these weights correspond to the only synapses by which the neurons are connected to themselves.

***Addressing the memory content.*** After having trained the network with a certain input vector **a** this external input can be switched off without changing the output; thus, due to the internal connections built up during learning the network keeps the activity induced by the external stimuli even if the stimuli are no longer present.

How does the network react to incorrect input? If for a limited period of time an input vector is provided to the network that does not correspond to its stored vector, the network relaxes to a stable state that corresponds to its stored vector or a multiple thereof, after having switched off the input. Therefore, the network has the ability of pattern completion. For a network characterised by matrix W3, the stable state is reached immediately. For matrix W4 the relaxation takes some time depending upon value $d_i$ ($d_i > 0$). A given ε-neighbourhood of the stable state is reached the faster, the more similar input vector and stored vector (or its nearest multiple) are.

### 3.4.2 Representing simple algebraic relations

***Training the network.*** The second task addressed in the Introduction (Chapter 3.1) and Chapter 3.2 was to learn algebraic rules, as given in the condition ABA or ABB on the one hand and equations like $x_3 = x_1 + x_2$ on the other hand. Such tasks require that not only one vector is learnt, but a solution for all vectors is found that fulfil the respective condition.

Providing a network consisting of IC Units with input vectors following the former condition ABA (e.g. $(5,1,5)$, $(2,3,2)$) leads to weight matrix (W5):

$$\begin{pmatrix} 0.5 & 0 & 0.5 \\ 0 & 1 & 0 \\ 0.5 & 0 & 0.5 \end{pmatrix} \qquad \text{(W5).}$$

Training the network with the second condition applied by (Marcus et al., 1999), namely ABB (e.g. $(5,1,1)$, $(2,3,3)$) another weight matrix is obtained:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0.5 & 0.5 \\ 0 & 0.5 & 0.5 \end{pmatrix} \qquad \text{(W6).}$$

The second task mentioned in Section 2.2 requires a solution for all vectors fulfilling the equation $\sum_{i=1}^{n} c_i x_i = 0$ for given $c_i \in \mathbb{R}$, i.e. all vectors of an $(n-1)$-dimensional hyperplane containing the origin. Geometrically, for $n = 3$, the solution is given by a plane in the 3D coordinate system that contains all points given by the coordinates that fulfil the equation $c_1 x_1 + c_2 x_2 + c_3 x_3 = 0$. Therefore, the solution is completely defined if three points are given. As $(0,0,0)$ is already a solution, only two further examples (not collinear with $(0,0,0)$) are sufficient to specify the solution. Generally, the solution for any task described by $\sum_{i=1}^{n} c_i x_i = 0$ with fixed coefficients $c_i \in \mathbb{R}$ is uniquely defined if

$n-1$ examples are presented to the network that form an $(n-1)$-dimensional subspace and do not contain the origin.

Application of a network with IC Units actually leads to this solution. To illustrate this ability, we again use a three unit network. The task to be trained is $x_3 = x_1 + x_2$. Any two training examples fulfilling the equation could be used (e.g. $(5,-1,4)$, $(-1,4,3)$). The same solution (W7) is obtained whether the training examples are presented in periodic epochs or in random order:

$$\begin{pmatrix} 2/3 & -1/3 & 1/3 \\ -1/3 & 2/3 & 1/3 \\ 1/3 & 1/3 & 2/3 \end{pmatrix} \qquad \text{(W7)}$$

Here, all nine weights were allowed to learn. Matrix W7 can be interpreted to be a special case of matrix W2 that is expanded by application of a damping factor $d_i = 2$ as explained above. If, however, the diagonal weights are constrained and always set to zero, we obtain a solution that corresponds to matrix W2 (see Chapter 3.2). These results are based on numerical investigations; a general proof is still pending.

***Addressing the memory content.*** If a network trained on either of the two conditions ABA or ABB is provided with a consistent input (e.g. $(7,1,7)$ for the first condition ABA), it immediately stabilises at this values even if the values have not been presented to the network before, i.e. are totally new. If, in turn, inconsistent input is presented to the network (e.g. $(5,1,1)$ for the first condition ABA), the activation of the unit not matching the condition asymptotically approaches the correct value.

To address the memory content after having trained all the weights according to the task of representing the summation (or, based on matrix W4, after the application of any positive damping factors), the network is provided with a vector **a** the first component $a_1$ and the second component $a_2$ of which are fixed to certain values while the third $a_3$ is set to zero. In the end, the third component should be the sum of the other components.

In each case, the network provides a solution that fulfils $x_3 = x_1 + x_2$. But it is not necessarily the case that $x_1 = a_1$ and $x_2 = a_2$. This condition is fulfilled in two cases:

(*i*) $x_1 = a_1$ and $x_2 = a_2$, if $|a_1| > |a_2|$ and $|a_2| \geq \frac{|a_1|}{2}$

(*ii*) $x_1 = a_1$ and $x_2 = a_2$, if $|a_2| > |a_1|$ and $|a_1| \geq \frac{|a_2|}{2}$.

If, however, $|a_1| > |a_2|$ and $|a_2| < \frac{|a_1|}{2}$, we obtain $x_1 = a_1$ and $x_2 = a_2 - \frac{a_1 + a_2}{2}$; and if $|a_2| > |a_1|$ and $|a_1| \geq \frac{|a_2|}{2}$, we obtain $x_1 = a_1 - \frac{a_1 + a_2}{2}$ and $x_2 = a_2$.

Therefore, if all $a_i \in (i) \vee (ii)$ unit $x_3$ approaches asymptotically the value $x_3 = a_1 + a_2$. Nevertheless, in the other cases as mentioned the network still stabilises at a value $x_3$ following the summation task $x_3 = x_1 + x_2$. Thus, the trained network is able to cope also with this pattern completion task. Correspondingly, solving the equation for the other variables is possible, too.

## 3.5 Discussion

In this chapter we propose *Input Compensation Units* (IC Units) as a new internal structure for artificial neurons that can be used as a basic building block of recurrent neural networks and allows for an efficient training of the synaptic weights. RNNs consisting of these IC Units and being trained in the described way have two main advantages over traditional approaches in training recurrent neural networks making them biologically more realistic:

First, the learning algorithm can be applied online, i.e. without cutting the recurrent connections, because the learning dynamics are disentangled from the dynamics of the recurrent network as such. This is possible due to the following properties: As the sum of the weighted internal inputs is subtracted from the external input, the output of the neuron always equals the size of the external input and is therefore independent of its learning state (Eq. 3.1 and 3.2). In other words, as the built-in compensation mechanism always replaces that part of the input signal that corresponds to the sum of the recurrent signals, the global dynamics of the network is protected from the learning dynamics. Therefore, no stability problems arise here due to weight changes. During the training procedure, the weights stabilise at values guaranteeing that in the end the summed recurrent input $s_i$ equals the external input $a_i$. After learning is completed, and the

summed internal input equals the external input, the latter can be switched off without changing the activation of the network.

Second, the synaptic weights of each neuron are adapted using local information only. The single neuron does not rely on information about the activation of whole network but only to information directly available at its synaptic connections just like real neurons. Consequently, the computational costs are very low – in contrast to many other training procedures (e.g. Williams and Zipser, 1989a; Schmidhuber, 1992) as no specific network for determination of the error and for its backpropagation is needed.

### 3.5.1 Biological plausibility

To implement the mechanism described the neuron has to distinguish between external input and input supplied by the recurrent connections of the network. How is this possible in a biological network? It is known (e.g. Kandel et al., 2000) that different types of synapses exist; the strength of one type does not easily change whereas other synapses show variation depending on activity. Additionally, physiological findings show, that the dendritic tree of a neuron is subdivided into different computational subunits for chemical signals such as changes in concentration of ions or other second messengers; this compartmentalisation is considered to be the basis of local modifications of the dendritic properties to achieve, for example, input-specific changes of synaptic weights (Helmchen, 1999) and it is also important from a computational perspective (Mel, 1999). Therefore, a different treatment of sensory input to the neuron and the recurrent internal input might well be possible.

Furthermore, some speculations concerning potential molecular mechanisms underlying the internal structure of the IC Units are possible; basic building blocks necessary to realise the algorithm proposed here can be found in real neurons (e.g. Kandel et al., 2000): Several pathways are known that increase and others that decrease the concentration of substances that influence the insertion of AMPA receptors in the synaptic membrane, for example. It is widely assumed, that the kinetics and magnitude of NMDA receptor mediated $Ca^{2+}$ signal determine the sign of synaptic modification (Kirkwood et al., 1993; Cummings et al., 1996). A large increase of $Ca^{2+}$ favours the activation of kinases which results in a phosphorylation of AMPA receptors; a lower increase in contrast favours the activation of phosphatases which results in a dephosphorylation of AMPA receptors (e.g. Lisman, 1989; Cormier et al., 2001).

### 3.5.2 Capabilities of the network

***Representing static patterns.*** Using these units it is possible to solve several memory tasks. First, static input patterns can be applied; due to the built-in learning mechanism the weights adapt in a way that the activations of the units remain fixed even after the external input signal has been switched off, thus producing sustained activity in the network.

It has been suggested to use attractor dynamics of coupled neurons provided with strong feedback for modelling these states of enhanced activity (Wilson and Cowan, 1973; Amari, 1977; Hopfield, 1982; Zipser et al., 1993; Amit, 1995; see also Chapter 2; for reviews on neurocomputational models see Durstewitz et al., 2000; Del Giudice et al., 2003). However, the performance of many of the proposed models is highly dependent on fine tuning the network parameters such as synaptic strength. If parameters only deviate slightly from the tuned values, the networks tend to diverge (Wang, 2001). In contrast, our model does not require fine-tuning of the weights as it automatically adapts to the current input situation.

When providing the network with a vector different from the stored one, the stored vector or a multiple of it is reproduced. This property can be interpreted as an error correction mechanism (or the capability to generalise) as it has been described for Hopfield networks (Hopfield, 1984; for a more detailed comparison with other recurrent neural networks see below).

Additionally, if a part of the vector is not specified by the input, i.e. a component of the input vector is set to zero, the network shows the ability of pattern completion: It finds an appropriate activation for the unspecified units.

***Representing algebraic relations.*** There has been a heated debate on the claim made by Marcus et al. (1999) that it is not possible to replicate their results with simple recurrent neural networks (see Seidenberg and Elman, 1999). The problem with connectionist-like models is that they are not able to generalise the abstract patterns to new words and are thus dependent on the input choice. They cannot abstract the underlying rule as it is necessary for the task described in the Introduction (Chapter 3.1) and in Chapter 3.2. The model presented here does not represent any word explicitly but

only the rule of an open-ended abstract relationship, in this case a simple algebraic relation. If the network is provided with consistent input it immediately stabilises on these activation values, whereas it needs some time to relax on the inconsistent condition. This matches with the results of the experiments performed by Marcus et al. (1999). The time the network needs to relax when provided with inconsistent input can be interpreted as to correspond to the longer time of attention the infants paid to sentences being inconsistent with the trained ones in the experiments carried out by Marcus et al. (1999). Therefore it is possible to simulate the experimental results obtained by Marcus et al. (1999) with networks consisting of IC Units.

Similarly, such algebraic rules may also underlie other grammatical phenomena as for example building English sentences with plural agreement from an arbitrary set of noun and verb phrases. In this sense humans know for example that a correct English sentence can be formed by combining any plural noun phrase with any verb phrase with plural agreement: From the two phrases “Bart and Lisa”, which is a plural noun phrase, and “played in the garden”, which is a verb phrase with plural agreement, we can infer that “Bart and Lisa played in the garden” is a correct English sentence. Here as well, networks that rather represent the abstract relations between the items than the single words may underlie the ability to build correct sentences.

The network can also be trained to represent any linear task $\sum_{i=1}^{n} c_i x_i = 0$ when only some (at least $n-1$) correct training examples are presented. The network forms a holistic representation of this algebraic relation implying the capability of pattern completion also in this task: If $n-1$ variables are given, the remaining variable is determined by the network. If, during recall, fewer variables are given and the task is therefore underdetermined, the network still provides a correct solution. The task is not solved by using a look-up table, but by representing the underlying mechanism.

The tasks described in Chapter 3.4.2 are characterised by homogeneous equations $\sum_{i=1}^{n} c_i x_i = 0$. However, this network can also be applied to tasks that require non homogeneous equations $\sum_{i=1}^{n} c_i x_i = h_i$ with constant values $h_i$. This corresponds to the

introduction of a 'bias unit' often used in neural networks. The network can simply be extended by such a bias unit by adding a unit, which is assumed to have a constant activation of 1. The weight of this bias unit corresponds to the value $h_i$ and can be trained using the same algorithm explained above.

Human's internal representations are not necessarily static by nature. As already mentioned by Johnson-Laird (1983) internal representations could be dynamic, i.e. they show time-dependent behaviour. This claim is underpinned by recall experiments showing that memory can be influenced by the observed movement (e.g. direction and speed) of an object (Freyd and Finke, 1984). Such dynamical systems can also be modelled by a network consisting of Input Compensation Units as will be explained in Chapter 4.

### 3.5.3 Comparison with other recurrent neural networks

The underlying idea of the Input Compensation Units corresponds to the clamped phase in Contrastive Learning (CL) procedures (Movellan, 1990; Baldi and Pineda, 1991). The advantages of CL are the possibility to train networks with hidden units on the on hand and to use nonlinear activation functions on the other hand. Up to now it has not been tested how the IC approach could deal with nonlinearities and hidden units. These are certainly the next problems to be tackled.

But there are three main differences between the two approaches: First, in all examples of the CL approach the weights of the feedback connections are assumed to be symmetric with the feedforward connections. In networks consisting of IC Units the weights are not constraint.

Second, in contrast to CL only one phase is applied and no oscillations between a phase with a teacher signal and one without a teacher signal are necessary.

Third, in CL the dynamics of the network are separated from the dynamics due to the learning procedure by definition as the dynamical equations are first run until convergence to a fixed point and then the weights are updated (Xie and Seung, 2003). In doing so, the problem of intertwining two interacting dynamics does not arise. But it is biologically not plausible that the synapses only then change, after the dynamics of the network has settled. For biological systems this "waxing and waning" of the synapses is assumed to not be explicitly uncoupled from the networks activity but on the contrary

explicitly *dependent* on the networks activity. The latter is the case in the neuronal units presented here: the updating of the weights is performed online, i.e. in each single time step and there is no necessity to decouple it explicitly from the network. Therefore, the IC approach appears to be nearer to biological reality.

The training procedure used here is based on the principle of teacher forcing (e.g. Williams and Zipser, 1989a; Doya, 1995; Jaeger and Haas, 2004): the actual output of a unit is replaced by the teacher signal in the subsequent computation. This principle permits online learning and has been applied by other approaches like real-time recurrent learning for RNNs (e.g. Williams and Zipser, 1989b). The problem with real-time recurrent learning is that it is computationally very intensive concerning storage and time and – moreover – the algorithm is non-local because each weight needs the knowledge of the complete recurrent weight matrix and the error vector. RNNs consisting of IC Units are trained using local information only and therefore the computational costs are very low.

To alleviate the problem of computational costs, a number of approaches have been put forward like, for example, the modification of the real-time recurrent learning algorithm by Schmidhuber (1992) which reduces at least the computational time but still needs quite large storage capacities.

Kalveram (2000) also proposed a learning algorithm formally corresponding to the delta rule like the IC approach incorporated on the level of the individual neuron. This has been applied to feedforward networks. The weights of external inputs are trained by providing the unit with the desired output. This input corresponds to the fixed external input used here but has to be switched off after training. In contrast our networks comprise memory units that are activated via the external input (see also below).

Other examples trying to reduce computational costs are the echo-state networks (Jaeger and Haas, 2004) and, quite similar besides using spiking neurons, the liquid-state machines (Maass et al., 2002). These types of networks need more units to equip the reservoir but are able to learn complex dynamic behaviour. Storing static patterns has not been addressed within these approaches. It will be shown in Chapter 4 that learning dynamic patterns is also possible with RNNs consisting of IC Units.

Similarities could be figured out, too, between the IC networks and Hopfield (Hopfield, 1982; 1984) networks on the one hand and MSBE networks (Cruse, 2005) on the other hand. What is the difference between the weight matrices resulting from the training procedure presented here to that of those other types of recurrent neural networks? The former are defined by symmetric weights and bounded activation functions. The units used here do not have bounded activation functions. Symmetric weights could, but do not necessarily result from application of the IC algorithm. Symmetric weights arise in matrices W2, W5, W6 and W7, but not in W3 and W4. Therefore, application of IC Units does generally not lead to Hopfield type networks.

MSBE networks are derived in the following way. If an equation with $n$ variables $\sum_{i=1}^{n} v_i \cdot x_i = 0$ is solved for each variable $x_i$, a set of equations is obtained. If each of theses n equations is considered to represent the computation performed by the corresponding neuron $i$, the network represents *M*ultiple *S*olutions for the *B*asic *E*quation $\sum_{i=1}^{n} v_i \cdot x_i = 0$ and is termed therefore MSBE network. For $n = 3$, for example, the basic equation $v_1 x_1 + v_2 x_2 + v_3 x_3 = 0$ being resolved for $x_1$, $x_2$ and $x_3$ leads to a weight matrix

$$\begin{pmatrix} 0 & v_2/v_1 & v_3/v_1 \\ v_1/v_2 & 0 & v_3/v_2 \\ v_1/v_3 & v_2/v_3 & 0 \end{pmatrix} \qquad \text{(W8)}$$

MSBE networks, like Hopfield networks, can be considered as autoassociators that have the property of pattern completion. Unlike Hopfield networks, that show discrete attractors, the attractor points of MSBE networks form a smooth, bounded space.

The weights follow the condition $w_{ij} \cdot w_{ji} = 1$. So the MSBE network is symmetric only for $v_1 = v_2 = v_3$. As described above for (W4), the weight matrix W8 can be extended by the introduction of damping factors $d_1$, $d_2$, and $d_3$.

Inspection of the different weight matrices obtained by the learning procedure applied to the IC Units reveals that some, but not all matrices fulfil the condition $w_{ij} = 1/w_{ji}$.

Matrix W2 fulfils the condition, matrix W3 only when applying a damping factor $d_i = \frac{a_i^2}{a_1^2 + a_2^2 + a_3^2} - 1$ and W4 when applying $d_i = \frac{a_i^2}{a_1^2 + a_2^2 + a_3^2} - 1$. This means that the IC algorithm can but does not necessarily produce weight distributions typical for MSBE networks. The latter is the case in particular, when in contrast to all examples used here, the weights are not all set to zero at the beginning of training.

### 3.5.4 Working memory and long term memory functions

In various experiments properties of the working memory have been investigated (Del Giudice et al., 2003). In electrophysical recordings stimulus-specific, enhanced activity can be observed which is assigned to be a feature of active working memory and enables animals to hold items in memory for some time. If no further attention is applied to the content of memory, it vanishes after a short time.

This property can also be found in our model: After presenting a static stimulus the activation of the artificial neurons is enhanced. During learning the weights approach the final values characterising the neutrally stable state only asymptotically. Therefore, in more natural situations, training is finished with non-ideal weight values. Hence, after an input has been presented to the network and later switched off, the activation of the network does not remain constant, but decreases to zero with a velocity depending on how closely the ideal values have been approximated during training (note that the weights maintain their values). This property may be considered as corresponding to the function of working memory, the content of which disappears if no specific attention is applied to maintain this content for a longer time. The velocity of this decrease of activation depends on the quality of learning, i.e., on learning time.

At the same time, the network can be considered to represent a passive memory (Fuster, 1995). If, after an activated network has been returned to zero activation, the input $a_1$, $a_2$, $a_3$ is presented again later, it would immediately activate the network.

As described above the weight values are only changed by means of the learning algorithm (Eq. 2), i.e., only when an external input is given. However, weights may also decay spontaneously (as do synapses), but with a long time constant (e.g. hours or days). Under this condition, the IC Units alone were not sufficient to explain long term memory. The following additional mechanism could, however, be applied: If the excitation has been strong enough, or has been repeated sufficiently often, a special

mechanism may come into action that prohibits synaptic decay and weights may stay fixed. In other words, the network forms a long term memory only after this fixation process has been performed (for a review of observations concerning switches between discrete states of synapses see Montgomery and Madison, 2004). In contrast to the architecture explained above, this additional mechanism would imply that not every input is maintained in the long term memory. Rather the system would be able to select frequent or salient information, and only such information is stored permanently.

## 3.6 Appendix: Learning a static pattern to produce sustained activity

### 3.6.1 Proof of convergence – training all the weights

During the training phase the $n \times n$ weight matrix $\mathbf{W}(t)$ is updated according to (2) as follows

(A1) $\mathbf{W}(t+1) = \mathbf{W}(t) + \varepsilon \cdot \delta(t) \cdot \mathbf{a}^{\mathrm{T}} = \mathbf{W}(t) + \varepsilon \cdot (\mathbf{I} - \mathbf{W}(t)) \cdot \mathbf{a} \cdot \mathbf{a}^{\mathrm{T}} \quad t = 0,1,2,\ldots.$

We denote by $\mathbf{P}_{\mathrm{a}} = \dfrac{1}{\mathbf{a}^{\mathrm{T}} \cdot \mathbf{a}} \cdot \mathbf{a} \cdot \mathbf{a}^{\mathrm{T}}$ the orthogonal projector onto span{a}.

**Theorem 1** *Under the assumption*

(A2) $0 < \varepsilon < \dfrac{2}{\mathbf{a}^{\mathrm{T}} \cdot \mathbf{a}}$

*the iteration (A1) converges for any* $\mathbf{W}(0) = \mathbf{W}_0$ *to the weight matrix*

(A3) $\mathbf{W}_\infty = \mathbf{W}_0 \cdot (\mathbf{I} - \mathbf{P}_{\mathrm{a}}) + \mathbf{P}_{\mathrm{a}}$.

*In particular if* $\mathbf{W}(0) = 0$ *we obtain* $\mathbf{W}_\infty = \mathbf{P}_{\mathrm{a}}$ *as in* (**W3**).

**Proof :** We use the following well-known result.

**Theorem 2** *Let* $\mathbf{X}$ *be a finite dimensional linear space and let* $\mathbf{X}$ *be the direct sum of two of its subspaces* $\mathbf{X}_1$ *and* $\mathbf{X}_2$*, i.e. every* $\mathbf{W} \in \mathbf{X}$ *can be written in a unique way as* $\mathbf{W} = \mathbf{W}_1 + \mathbf{W}_2$ *where* $\mathbf{W}_1 \in \mathbf{X}_1, \mathbf{W}_2 \in \mathbf{X}_2$.

*Let* $\mathbf{L} : \mathbf{X} \to \mathbf{X}$ *be a linear map such that*

(i) $\mathbf{L} \cdot \mathbf{W} = \mathbf{W}$ *for all* $\mathbf{W} \in \mathbf{X}_1$

(ii) $\mathbf{L}$ *maps* $\mathbf{X}_2$ *into itself and* $|\lambda| < 1$ *for all eigenvalues* $\lambda$ *of* $\mathbf{L}$ *that belong to eigenvectors in* $\mathbf{X}_2$.

*Then the iteration*

(A4) $\mathbf{W}(t+1) = \mathbf{L} \cdot \mathbf{W}(t) + \mathbf{R}_2, \ \mathbf{W}(0) = \mathbf{W}_0$

*converges for any* $\mathbf{W}_0 \in \mathbf{X}$ *and any* $\mathbf{R}_2 \in \mathbf{X}_2$ *to*

$$\mathbf{W}_\infty = (\mathbf{W}_0)_1 + \mathbf{W}_2$$

*where* $\mathbf{W}_0 = (\mathbf{W}_0)_1 + (\mathbf{W}_0)_2$ *is the decomposition of* $\mathbf{W}_0$ *and* $\mathbf{W}_2 \in \mathbf{X}_2$ *is the unique solution in* $\mathbf{X}_2$ *of the equation*

(A5) $\mathbf{W}_2 = \mathbf{L} \cdot \mathbf{W}_2 + \mathbf{R}_2$.

We apply this Theorem 1 to (A1) with $\mathbf{X}$ the space of $n \times n$ matrices and

$$\mathbf{X}_1 = \{\mathbf{W} \in \mathbf{X} : \mathbf{W} \cdot \mathbf{a} = 0\}, \dim \mathbf{X}_1 = n^2 - n$$

$$\mathbf{X}_2 = \{\mathbf{b} \cdot \mathbf{a}^{\mathrm{T}} : \mathbf{b} \in \mathbb{R}^n\}, \dim \mathbf{X}_2 = n.$$

The decomposition of $\mathbf{W} \in \mathbf{X}$ is given by

$$\mathbf{W} = \mathbf{W}(\mathbf{I} - \mathbf{P}_{\mathrm{a}}) + \mathbf{W} \cdot \mathbf{P}_{\mathrm{a}} = \mathbf{W}_1 + \mathbf{W}_2$$

since $\mathbf{W}(\mathbf{I} - \mathbf{P}_{\mathrm{a}}) \cdot \mathbf{a} = 0$ and $\mathbf{W} \cdot \mathbf{P}_{\mathrm{a}} = \mathbf{b} \cdot \mathbf{a}^{\mathrm{T}}$ with $\mathbf{b} = \dfrac{1}{\mathbf{a}^{\mathrm{T}} \cdot \mathbf{a}} \cdot \mathbf{W} \cdot \mathbf{a}$.

The iteration (A1) has the form (A4) if we define

(A6) $\mathbf{L} \cdot \mathbf{W} = \mathbf{W} \cdot (\mathbf{I} - \varepsilon \cdot \mathbf{a} \cdot \mathbf{a}^{\mathrm{T}}), \mathbf{R}_2 = \varepsilon \cdot \mathbf{a} \cdot \mathbf{a}^{\mathrm{T}}$.

Note that (i) follows from $\mathbf{L} \cdot \mathbf{W} = \mathbf{W} \cdot (\mathbf{I} - \varepsilon \cdot \mathbf{a} \cdot \mathbf{a}^{\mathrm{T}}) = \mathbf{W}$ for $\mathbf{W} \in \mathbf{X}_1$.

If $\mathbf{W} = \mathbf{b} \cdot \mathbf{a}^{\mathrm{T}} \in \mathbf{X}_2$ then we have

$$\mathbf{L} \cdot \mathbf{W} = \mathbf{b} \cdot \mathbf{a}^{\mathrm{T}} \cdot (\mathbf{I} - \varepsilon \cdot \mathbf{a} \cdot \mathbf{a}^{\mathrm{T}}) = \mathbf{b} \cdot (\mathbf{a}^{\mathrm{T}} - \varepsilon \cdot (\mathbf{a}^{\mathrm{T}} \cdot \mathbf{a}) \cdot \mathbf{a}^{\mathrm{T}}) = (1 - \varepsilon \cdot \mathbf{a}^{\mathrm{T}} \cdot \mathbf{a}) \cdot \mathbf{b} \cdot \mathbf{a}^{\mathrm{T}},$$

therefore $\lambda = 1 - \varepsilon \cdot \mathbf{a}^{\mathrm{T}} \cdot \mathbf{a}$ is an $n$-fold eigenvalue of $\mathbf{L}$ and $|\lambda| < 1$ holds if and only if (A2) is satisfied.

Then Theorem 1 is applicable and yields (A3) if we show that (A5) holds for $\mathbf{W}_2 = \mathbf{P}_{\mathrm{a}}$.

In fact, $\mathbf{P}_{\mathrm{a}} \in \mathbf{X}_2$ and

$$\mathbf{P}_{\mathrm{a}} - \mathbf{L} \cdot \mathbf{P}_{\mathrm{a}} = \mathbf{P}_{\mathrm{a}} - \mathbf{P}_{\mathrm{a}}(\mathbf{I} - \varepsilon \cdot \mathbf{a} \cdot \mathbf{a}^{\mathrm{T}}) = \varepsilon \cdot \mathbf{P}_{\mathrm{a}} \cdot \mathbf{a} \cdot \mathbf{a}^{\mathrm{T}} = \varepsilon \cdot \mathbf{a} \cdot \mathbf{a}^{\mathrm{T}} = \mathbf{R}_2.$$

❒

### 3.6.2 Proof of convergence – training with constraints

Now we consider the learning rule (A1) where only certain entries of the weight matrix are updated. We write this as follows:

(A7) $\mathbf{W}(t+1) = \mathbf{W}(t) + \varepsilon \cdot \mathbf{E} \circ (\mathbf{I} - \mathbf{W}(t)) \cdot \mathbf{a} \cdot \mathbf{a}^{\mathrm{T}}$

where E is an $n \times n$ matrix with entries 0 or 1 and where we used the Hadamard product $\mathbf{E} \circ \mathbf{B}$ of $n \times n$-matrices given by

(A8) $(\mathbf{E} \circ \mathbf{B})_{ij} = \mathbf{E}_{ij} \cdot \mathbf{B}_{ij}$.

The entries $\mathbf{W}_{ij}$ with $\mathbf{E}_{ij} = 1$ are updated while those with $\mathbf{E}_{ij} = 0$ are kept constant. In particular, for the choice

(A9) $$\mathbf{E} = \begin{pmatrix} 0 & 1 & \cdots & 1 \\ 1 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 1 & \cdots & 1 & 0 \end{pmatrix}$$

only the weights $\mathbf{W}_{ij}$ with $i \neq j$ are updated.

**Theorem 3** *Assume that the matrix* **E** *has no zero row and let* $\varepsilon > 0$ *satisfy for all* $i = 1,...,n$

(A10) $\varepsilon \cdot d_i < 2$, *where* $d_i = \sum_{E_{ij}=1} a_j^{\,2}$.

*Then the learning rule* (A7) *converges for any* $\mathbf{W}(0) = \mathbf{W}_0$ *to some limit matrix* $\mathbf{W}_\infty$. *In case* $\mathbf{W}_0 = 0$ *the entries of* $\mathbf{W}_\infty$ *are given by*

(A11) $(\mathbf{W}_\infty)_{ij} = \dfrac{a_i \cdot a_j}{d_i}$ *if* $\mathbf{E}_{ij} = 1$

*and* $(\mathbf{W}_\infty)_{ij} = 0$ *otherwise.*

*In particular, for the pattern matrix E from* (A9) *with* $n = 3$ *we obtain exactly the matrix* (**W4**). *In case* $\mathbf{E}_{ij} = 1$ *for all i,j we recover the results from Theorem 1.*

**Proof**: We apply Theorem 1 again with the setting

$$\mathbf{X} = \{\mathbf{W} : \mathbf{E} \circ \mathbf{W} = \mathbf{W}\} = \{\mathbf{W} : \mathbf{W}_{ij} = 0 \text{ if } \mathbf{E}_{ij} = 0\}$$

and

(A12) $\mathbf{L} \cdot \mathbf{W} = \mathbf{W} - \varepsilon \cdot \mathbf{E} \circ [\mathbf{W} \cdot \mathbf{a} \cdot \mathbf{a}^{\mathrm{T}}], \mathbf{R}_2 = \varepsilon \cdot \mathbf{E} \circ [\mathbf{a} \cdot \mathbf{a}^{\mathrm{T}}].$

The spaces $\mathbf{X}_1$ and $\mathbf{X}_2$ are given by

$$\mathbf{X}_1 = \{\mathbf{W} \in \mathbf{X} : \mathbf{W} \cdot \mathbf{a} = 0\}$$

$$\mathbf{X}_2 = \{\mathbf{W} = \mathbf{E} \circ (\mathbf{b} \cdot \mathbf{a}^{\mathrm{T}}) : \mathbf{b} \in \mathbb{R}^n\}.$$

First note that $\mathbf{L} \cdot \mathbf{W} = \mathbf{W}$ is obvious for $\mathbf{W} \in \mathbf{X}_1$.

In $\mathbf{X}_2$ we choose basis vectors

(A13) $\mathbf{V}_i = \mathbf{E} \circ (\mathbf{e}^i \cdot \mathbf{a}^{\mathrm{T}}),\ i = 1,...,n$

where $\mathbf{e}^i = (0,...,1,...0)$ is the i-th Cartesian basis vector. Note that

$$(\mathbf{V}_i \cdot \mathbf{a})_k = \sum \mathbf{E}_{ij} \cdot (\mathbf{e}^i)_k \cdot \mathbf{a}_j^2 = (\mathbf{e}^i)_k \cdot \sum_{\mathbf{E}_{ij}=1} \mathbf{a}_j^2 = d_i (\mathbf{e}^i)_k$$

holds and therefore

(A14) $\mathbf{V}_i \cdot \mathbf{a} = d_i \cdot \mathbf{e}^i$.

Since $\mathbf{E}$ has no zero row we have $d_i > 0$ for all $i$. Equation (A14) then implies that the vectors $\mathbf{V}_i$ are linearly independent and moreover we find that the vectors $\mathbf{V}_i$ are eigenvectors of $\mathbf{L}$

(A15) $\mathbf{L} \cdot \mathbf{V}_i = \mathbf{V}_i - \varepsilon \cdot d_i \cdot \mathbf{E} \circ (\mathbf{e}^i \cdot \mathbf{a}^{\mathrm{T}}) = \lambda_i \cdot \mathbf{V}_i,\ \lambda_i = 1 - \varepsilon \cdot d_i$

Condition (A10) guarantees that $|\lambda_i| < 1$ holds for all eigenvalues.

The decomposition $\mathbf{W} = \mathbf{W}_1 + \mathbf{W}_2$, $\mathbf{W}_1 \in \mathbf{X}_1, \mathbf{W}_2 \in \mathbf{X}_2$ is given by

(A16) $\mathbf{W}_2 = \sum_{i=1}^{n} b_i \cdot \mathbf{V}_i,\ b_i = \dfrac{(\mathbf{W} \cdot \mathbf{a})_i}{d_i},\ \mathbf{W}_1 := \mathbf{W} - \mathbf{W}_2$.

Note that $\mathbf{W}_1 = \mathbf{W} - \mathbf{W}_2 \in \mathbf{X}$ satisfies by (A14)

$$\mathbf{W}_1 \cdot \mathbf{a} = \mathbf{W} \cdot \mathbf{a} - \sum_{i=1}^{n} b_i \cdot \mathbf{V}_i \cdot \mathbf{a} = \mathbf{W} \cdot \mathbf{a} - \sum_{i=1}^{n} (\mathbf{W} \cdot \mathbf{a})_i \cdot \mathbf{e}^i = 0.$$

The decomposition is unique since $\mathbf{W} \in \mathbf{X}_1$ and $\mathbf{W} = \sum_i b_i \cdot \mathbf{V}_i \in \mathbf{X}_2$ implies

$$0 = \mathbf{W} \cdot \mathbf{a} = \sum_{i=1}^{n} b_i \cdot \mathbf{V}_i \cdot \mathbf{a} = \sum_{i=1}^{n} b_i \cdot \mathbf{e}^i = \mathbf{b} .$$

We have now verified the assumptions of Theorem 1.

In order to determine the limit matrix $\mathbf{W}_\infty$ we need to solve (A5) with $\mathbf{R}_2$ given in (A12). The solution is

$$\mathbf{W}_2 = \sum_{i=1}^{n} \frac{a_i}{d_i} \cdot \mathbf{V}_i$$

since by (A15)

$$\mathbf{W}_2 - \mathbf{L} \cdot \mathbf{W}_2 = \sum_{i=1}^{n} \frac{a_i}{d_i} \cdot \left( \mathbf{V}_i - \mathbf{L} \cdot \mathbf{V}_i \right) = \sum_{i=1}^{n} \frac{a_i}{d_i} \cdot \left( 1 - \lambda_i \right) \cdot \mathbf{V}_i = \varepsilon \cdot \sum_{i=1}^{n} a_i \cdot \mathbf{V}_i = \mathbf{R}_2 .$$

Combining this with (A16) Theorem 1 leads to the limit matrix $\mathbf{W}_\infty$ given for $\mathbf{E}_{ij} = 1$ by

(A17) $$\left( \mathbf{W}_\infty \right)_{ij} = \left( \mathbf{W}_0 \right)_{ij} + \frac{1}{d_i} \cdot \left( a_i - \left( \mathbf{W}_0 \cdot \mathbf{a} \right)_i \right) \cdot a_j .$$

In the case $\mathbf{W}_0 = 0$ this leads to formula (A11).

❒

## 3.7 References

Amari S (1977) Dynamics of pattern formation in lateral-inhibition type neural fields. *Biol Cybern* 27: 77-87.

Amit DJ (1995) The hebbian paradigm reintegrated: local reverberations as internal representations. *Behavioral and Brain Sciences* 18: 617-626.

Baddeley A (1986) *Working memory*. Oxford: Clarendon Press.

Baddeley A (1992) Working memory. *Science* 255: 556-559.

Baldi P, Pineda F (1991) Contrastive learning and neural oscillator. *Neural Computation* 3: 526-545.

Chomsky NA (1980) *Rules and Representation*. New York: Columbia University Press.

Cormier RJ, Greenwood AC, Connor JA (2001) Bidirectional Synaptic Plasticity Correlated With the Magnitude of Dendritic Calcium Transients Above a Threshold. *Journal of Neurophysiology* 85: 399-406.

Cruse, H (2005) Neural Networks as Cybernetic Systems. http://193.30.112.98/brain/archive/cruse [2].

Cummings JA, Mulkey RM, Nicoll RA, Malenka RC (1996) $Ca^{2+}$ signaling requirements for long-term depression in the hippocampus. *Neuron* 16: 825-833.

Del Giudice P, Fusi S, Mattia M (2003) Modelling the formation of working memory with networks of integrate-and-fire neurons connected by plastic synapses. *Journal of Physiology* 97: 659-681.

Doya K (1995) Recurrent networks: Supervised learning. In: *The Handbook of Brain Theory and Neural Networks* (M.Arbib, ed), Cambridge, MA: MIT Press, pp 796-800.

Durstewitz D, Seamans JK, Sejnowski TJ (2000) Neurocomputational models of working memory. *Nature Neuroscience Supplement* 3: 1184-1191.

Fogassi L, Ferrari PF, Gesierich B, Rozzi S, Chersi F, Rizzolatti G (2005) Parietal lobe: From action organization to intention understanding. *Science* 308: 662-666.

Freyd JJ, Finke RA (1984) Representational momentum. Journal of Experimental Psychology: *Learning, Memory, and Cognition* 10: 126-132.

Fuster JM (1973) Unit activity in the prefrontal cortex during delayed response performance: neuronal correlates of transient memory. *Journal of Neurophysiology* 36: 61-78.

Fuster JM (1995) *Memory in the Cerebral Cortex: An Empirical Approach to Neural networks in the Human and Nonhuman Primate.* Cambridge, MA: MIT Press.

Fuster JM, Alexander GE (1971) Neuron activity related to short-term memory. *Science* 173: 652-654.

Helmchen F (1999) Dendrites as biochemical compartments. In: *Dendrites* (Greg Stuart, Nelson Spruston, Michael Häusser, eds), Oxford: University Press, pp 161-192.

Hertz J, Krogh A, Palmer RG (1991) *Introduction to the theory of neural computation.* Redwood City: Addison-Wesley Pub.

Hopfield JJ (1982) Neural networks and physical systems with emergent collective computational abilities. *Proc Natl Acad Sci USA* 79: 2554-2558.

Hopfield JJ (1984) Neurons with graded response have collective computational properties like those of two-state neurons. *Proc Natl Acad Sci USA* 81: 3088-3092.

Jaeger H, Haas H (2004) Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication. *Science* 2: 78-80.

Johnson-Laird PN (1983) *Mental models: towards a cognitive science of language, inference, and consciousness.* Cambridge: Cambridge University Press.

Kalveram KT (2000) Sensorimotor sequential learning by a neural network based on redifined Hebbian learning. In: *Perspectives in Neural Computing - Proceedings of the ANNIMAB-1 Conference* (H.Malgrem, M.Borga, L.Niklasson, eds), London: Springer, pp 271-276.

Kandel ER, Schwartz JH, Jessell TM (2000) *Principles of neural science*. New York: McGraw-Hill.

Kirkwood A, Dudek SM, Gold JT, Aizenman CD, Bear MF (1993) Common forms of synaptic plasticity in the hippocampus and neocortex in vitro. *Science* 260: 1518-1521.

Lisman JE (1989) A mechanism for the Hebb and the anti-Hebb processes underlying learning and memory. *Proceedings of the National Academy of Sciences of the USA* 86: 9574-9578.

Maass W, Natschläger T, Markram H (2002) Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation* 14: 2531-2560.

Marcus G, Vijayan S, Bandi Rao S, Vishton PM (1999) Rule learning by seven-month-old infants. *Science* 283: 77-80.

Marcus GF, Brinkman U, Clahsen H, Wiese R, Pinker S (1995) German inflection: The exception that proves the rule. *Cognitive Psychology* 29: 189-256.

Marcus GF (2001) *The algebraic mind: Integrating connectionism and cognitive science*. Cambridge, MA: MIT Press.

Mel BW (1999) Why have dendrites? A computational perspective. In: *Dendrites* (Greg Stuart, Nelson Spruston, Michael Häusser, eds), Oxford: University Press, pp 271-289.

Miyashita Y, Hayashi T (2000) Neural representation of visual objects: encoding and top-down activation. *Current Opinion in Neurobiology* 10: 187-194.

Montgomery JM, Madison DV (2004) Discrete synaptic states define a major mechanism of synaptic plasticity. *Trends in Neurosciences* 27: 744-750.

Movellan J (1990) Contrastive Hebbian learning in the continuous Hopfield model. In: *Proceedings of the 1990 Connectionist Models Summer School* (D.Touretzky, J.Elman, T.Sejnowski, G.Hinton, eds), San Mateo, CA: Morgan Kaufmann, pp 10-17.

Niki H (1974a) Prefrontal unit activity during delayed alternation in the monkey: I. Relation to direction of response. *Brain Research* 68: 185-196.

Niki H (1974b) Prefrontal unit activity during delayed alternation in the monkey: II. Relation to absolute versus relative direction of response. *Brain Research* 68: 197-204.

Pinker S (1991) Rules of language. *Science* 253: 530-535.

Pinker S, Prince A (1988) On language and connectionism - analysis of a parallel distributed-processing model of language-acquisition. *Cognition* 28: 73-193.

Rumelhart DE, Hinton GE, Williams RJ (1986) Learning internal representations by error propagation. In: *Parallel Distributed Processing, volume 1: Foundations* (D.E.Rumelhart, J.L.McClelland, eds.), Cambridge, MA: MIT Press, pp 318-362.

Schmidhuber J (1992) A fixed size storage $O(n^3)$ time complexity learning algorithm for fully recurrent continually running networks. *Neural Computation* 4: 243-248.

Seidenberg MS, Elman JL (1999) Networks are not 'hidden rules'. *Trends in Cognitive Sciences* 3: 288-289.

Seung HS, Lee DD, Reis BY, Tank DW (2000) Stability of the memory of eye position in a recurrent network of conductance-based model neurons. *Neuron* 26: 259-271.

Steil JJ (1999) Input-Output Stability of Recurrent Neural Networks. Göttingen: Cuvillier Verlag.

Wang X-J (2001) Synaptic reverberation underlying mnemonic persistent activity. *Trends in Neurosciences* 24: 455-463.

Widrow B, Hoff ME (1960) Adaptive switching circuits. In: *1960 WESCON Convention record Part IV*, New York: Institute of Radio Engineers, pp 96-104.

Williams RJ, Zipser D (1989a) A learning algorithm for continually running fully recurrent neural networks. *Neural Computation* 1: 270-280.

Williams RJ, Zipser D (1989b) Experimental analysis of the real-time recurrent learning algorithm. *Connection Science* 1: 87-111.

Wilson H, Cowan J (1973) A mathematical theory of the functional dynamics of cortical and thalamic nervous tissue. *Kybernetik* 13: 55-80.

Wolpert DM, Kawato M (1998) Multiple paired forward and inverse models for motor control. *Neural Networks* 11: 1317-1329.

Xie X, Seung HS (2003) Equivalence of Backpropagation and Contrastive Hebbian Learning in layered networks. *Neural Computation* 15: 441-454.

Zipser D, Kehoe B, Littlewort G, Fuster J (1993) A spiking network model of short-term active memory. *Journal of Neuroscience* 13: 3406-3420.

# 4 Modelling Memory Functions with Recurrent Neural Networks consisting of Input Compensation Units: II. Dynamic Situations

Modelling cognitive abilities of humans or animals or building agents that are supposed to behave cognitively requires modelling a memory system that is able to store and retrieve various contents. The content to be stored is assumed to comprise information about more or less invariant environmental objects as well as information about movements. A combination of both information about objects and movements may be called situation model.

Here, we focus on the one hand on models storing dynamic patterns. Particularly, two abilities of humans in representing dynamical systems have been concentrated on: the capability of representing acceleration of objects as can be found in the movement of a pendulum or freely falling objects and representing actions of transfer, i.e. motion from one point to another, has been modelled using recurrent networks consisting of IC Units.

On the other hand, possibilities of combining static and dynamic properties within a single model have been studied.

## 4.1 Introduction

To account for various aspects of human and animal cognition cognitive scientists have put forward the theoretical notion of "mental representation" (see von Eckardt, 1993). Especially in research on text comprehension much work has been done concerning the content of these representations. The traditional view changed with two books published independently in 1983 (Johnson-Laird, 1983; van Dijk and Kintsch, 1983): linguistic and psychological studies revealed that it is rather the situation described within a text which is represented than the text itself. These "mental representation of verbally described situations" (Zwaan et al., 1998) have become known as mental models (Johnson-Laird, 1983) or situation models (van Dijk and Kintsch, 1983).

What is the nature of such mental representations? In order to describe humans' cognitive capacities von Eckardt (1999) lists a number of features mental representations must have, among which are that it must be possible to "represent specific objects; to represent many different kind of objects – concrete objects, sets, properties, events, and state of affairs in this world, in possible worlds, and in fictional worlds as well as abstract objects such as universals and numbers." In Chapter 2 and 3 we have already dealt with the representation of static objects. But the representations humans build up are not necessarily static by nature. They can also comprise "events" as von Eckard calls it or, in other words, be dynamic, i.e. show time-dependent behaviour. Already Johnson-Laird postulated that "many mental representations are kinematic or dynamic" (Johnson-Laird, 1983). Up to now much research has been performed showing that humans tend to realise things as dynamic structures and additionally anticipate suspected changes already in the mental representation. Continuous changes are likely to be represented dynamically, i.e. these changes are simulated mentally by a respective change within the representation (Freyd, 1993). Therefore, here the focus is on this special feature of mental representations. The term representation is used here like defined above (Chapter 2) in the broad sense of Steels (1995) as being "physical structures (for example electro-chemical states) which have correlations with aspects of the environment".

The dynamic nature of representations has already been considered in early studies dealing with the so-called *representational momentum* (e.g. Freyd and Finke, 1984, for further literature see Freyd, 1993). The term representational momentum describes the finding that memory can be influenced by the observed movement (e.g. direction and speed) of an object: Memory failures can fall along the direction of implied motion. In the classical experiment (Freyd and Finke, 1984) test persons were provided with a sequence of pictures showing a rectangle at different orientations along a possible path of rotation. The pictures were separated by an interstimulus interval of 250-500 *ms*. Test persons were instructed to remember the orientation of the last object. In this setup the subject's memory tends to be displaced forward in the direction of the implied motion. This indicates that the test person's internal mental representation of the external situation presented comprises dynamic properties as for example inertia like real physical objects.

Research on language comprehension also provides evidence for the existence of such dynamic representations (e.g. Glenberg and Kaschak 2002; Zwaan et al., 2004). The hypothesis put forward by Zwaan et al. (2004) starts from the considerations on the representational momentum and the theories that language comprehension involves perceptual simulation (see above), but goes a step further: These authors assume that dynamic mental representations are "perceptual traces that are stored as temporal patterns of activation that unfold over time" corresponding to the respective perceptual experience. Along this line of argumentation they predict that the perception of a visual motion event is facilitated by preceding comprehension of a sentence describing this motion event.

In their experiments test persons heard a sentence describing the motion of a ball towards or away from the observer. A short time after the sentence a picture of a ball is presented followed by a second picture of a ball. The ball in the second picture was slightly smaller or larger than in the first one, suggesting movements of the ball towards or away from the observer. Subjects should judge whether the two sequentially presented visual objects were the same. Zwaan et al. (2004) found that test persons responded faster when the implied movement of the ball matched the movement described in the sentence. Thus, their results support the view that during language comprehension dynamic perceptual simulations are involved.

Also Glenberg and others (e.g. Glenberg and Kaschak, 2002) affirm the view that understanding a sentence describing for example actions of transfer seems to require the ability to internally simulate the motion of the object towards or away from the body even using the same neural system as in actually producing transfer. Thus the symbols of language are grounded by relating them to bodily processes (e.g. Lakoff, 1987; Glenberg and Robertson, 1999; Barsalou, 1999; Glenberg and Robertson, 2000; Fincher-Kiefer, 2001; for further literature see Glenberg & Kaschak 2002), because only then can real understanding be achieved. For example, we understand what a chair is, because we always derive the affordances from this object when seeing one. Affordances are potential interactions between bodies and objects (Glenberg and Kaschak, 2002); according to Gibson (1979), who coined the notion of affordances, a chair is a chair because it affords sitting for adult humans. So the idea is that language is made meaningful by cognitively simulating the actions implied by a sentence.

Glenberg and Kaschak (2002) were able to corroborate this hypothesis by experiments showing that actions in one direction (e.g. “close the drawer”) implied by a sentence interfere with really performed actions in the opposite direction (e.g. movement towards the body). If there is a mismatch between the action described in the sentence and the action, that should be performed, reaction time goes up. Thus, the study of Glenberg and Kaschak demonstrates humans’ ability to dynamically represent actions of transfer.

Furthermore people are not only capable of representing such transfer actions but are also capable of predicting the motion of accelerating objects as for example a ball falling down or a swinging pendulum. How do we manage to perceive such motions given the fact that the visual system is only poorly sensitive to acceleration ( Todd, 19 81; Lisberger and Movshon, 1999; Brouwer et al., 2002) as for example caused by gravity? When catching objects under normal gravity conditions, the movements are well synchronised with the arrival of the objects (McIntyre et al., 2001). In contrast, experiments with astronauts in a space-shuttle under reduced gravity revealed that the peak of anticipatory muscle activation as well as forearm movements occurred earlier relative to impact (McIntyre et al., 2001). Only after a few days the astronauts adapt to the new gravity conditions. These findings imply the existence of an internal model within the brain which calculates the effects of gravity usually experienced on earth to provide an estimate of the time-to-contact with accelerated objects.

In a nutshell, theses studies (for further literature see Freyd and Finke, 1984) show that humans do build up dynamic representations as an essential basis for understanding environmental situations as for example actions of transfer and properties of accelerating objects. The latter is especially important for survival as it a prerequisite of estimating time of collision with objects. But to our knowledge, none (or only a few) attempts have been made so far to provide models that explain of how such representations could be realised and learnt in a biologically plausible way.

Here, we demonstrate that a recurrent neural network consisting of IC Units, which has successfully be used to model sustained activity to represent static objects (Chapter 3) and which will be described briefly in Chapter 4.2, can also be utilised for online-learning and for representing dynamic situations. Such a network can be trained to represent the dynamics of physical systems like a pendulum or free-fall as well as the dynamics of low- and high-pass filters (Chapter 4.3 and 4.4). The former provides a

neural realisation of the model of gravity humans seem to have as shown by Lacquaniti and colleagues (Indovina et al., 2005), while the dynamics of low-pass filters can resemble actions of transfer as found by Glenberg and coworkers. In a second step (Chapter 4.5) a way is shown, how it is possible to merge the results of former studies concerning the representation of static objects (Chapter 3) with these dynamic representations. Here, for a first approach we focus on combing the model of static objects with the dynamics of a low-pass filter to model the content of sentences describing actions of transfer like for example "Homer walks to Marge".

We will not deal with questions concerning ways how to arrange a number of different such models within a larger memory system (see also Chapter 2.5 for further discussion).

## 4.2 The Model

The recurrent neural network used here to model representations of dynamic situations consists of so-called *Input Compensation Units* (IC Units) (for a detailed description see Chapter 3). The essential property of these units is to disentangle the dynamics of the recurrent network from the dynamics due to the learning procedure. This would be possible, if the output of the network always equated the input regardless of the actual size of the weights.

In order to achieve this, two input types are distinguished: One external input $a_i$, the weight of which is fixed to 1, and $n$ internal, i.e., recurrent input connections $x_j$, the weights $w_{ij}$ of which can be changed by learning. The overall output of one neuron is calculated in the following way:

$$x_i(t+1) = \begin{cases} s_i(t) + (a_i(t) - s_i(t))_+, & s_i(t) > 0 \\ s_i(t) - (-a_i(t) + s_i(t))_+, & s_i(t) \le 0 \end{cases} \qquad (1)$$

with $s_i(t) = \sum_{j=1}^{n} w_{ij} \cdot x_j(t)$, for $i = 1$ to $n$.

The $_+$ means that only the positive part of the expression within the brackets is transmitted. The structure of an Input Compensation Unit is schematically shown in Figure 4.1.

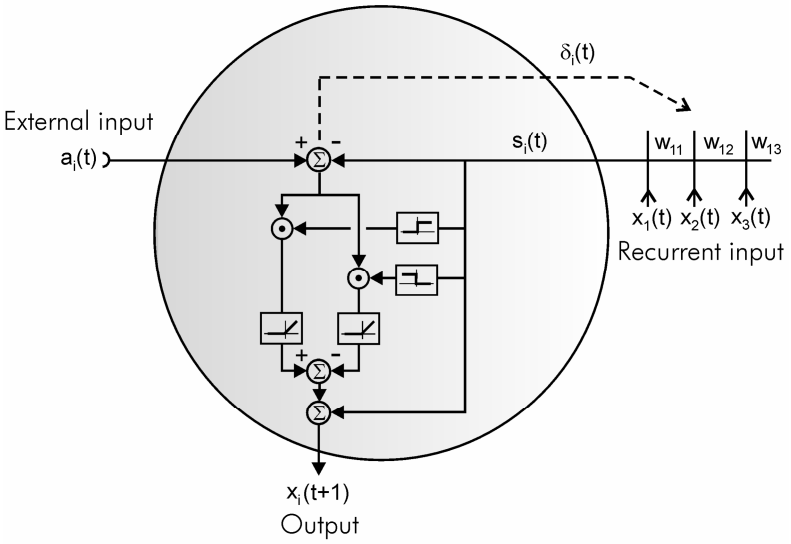


**Figure 4.1: Schematic drawing of the internal structure of an IC Unit: $a_i$ is the external input, $s_i$(t) is the weighted sum of the recurrent inputs and $\delta_i$(t) the difference between the external input $a_i$(t) and $s_i$(t) (for further explanation see text and Chapter 3.**

From equation (1) it is immediately clear that the output $x_j$ of the network always equals the external input independent of the actual weight size because the sum of the weighted internal inputs $s_i$ is added and subtracted instantly again. Therefore, this built-in compensation mechanism protects the global dynamics of the network from the learning dynamics. This enables us to train a recurrent neural network consisting of IC Units online, i.e. without cutting the feedback connections during training which is a big advantage over traditional approaches in training recurrent neural networks (for discussion of related approaches see Chapter 3).

The weights are changed according to the following algorithm:

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij} \text{ with } \Delta w_{ij} = \varepsilon \cdot x_j(t) \cdot \delta_i(t) \qquad (2)$$

with $\varepsilon > 0$ being the learning rate and $\delta_i(t) = a_i(t) - s_i(t)$ being the local error. By applying this learning rule the weights will change until $\delta_i(t) = 0$. This learning algorithm formally corresponds to the delta rule (Widrow and Hoff, 1960), but does not

require a separate network for calculating and backpropagating the error as it is computed locally within each single neuron. During the training procedure, the weights stabilise at values that in the end the summed recurrent input $s_i$ equals the external input $a_i$, i.e. $\delta_i(t) = 0$. After learning is completed, and the summed internal input equals the external input, the latter can be switched off without changing the activation of the network. In fact, learning has to be terminated to prevent the network from learning the zero input vector.

## 4.3 Methods

Before we explain how a dynamic situation as brought up in the Introduction (Chapter 4.1) could be learnt by a neuronal system, we demonstrate the existence and possible form of a solution. The dynamical systems mentioned can be described by linear differential equations. In case of the harmonic oscillations of a pendulum and the dynamics of a freely falling object these are 2nd order differential equations. To construct a recurrent neural network, we have to rewrite these 2nd order differential equation as a system of two coupled 1st order differential equation by introducing the velocity $\dot{x}$ as auxiliary parameter. In general, any explicit linear differential equation can be represented within a recurrent neural network by transferring a given explicit differential equation of order *n* to *n* coupled differential equations of order 1 (Nauck et al., 2003).

### 4.3.1 Pendulum

The dynamics of a mass-spring pendulum, i.e., its position and its velocity changing over time, are given by the differential equation $\ddot{x} = -\omega \cdot x - r \cdot \dot{x}$ with $\omega = \sqrt{D/m}$ representing the frequency (*D* is the spring constant and *m* the mass) and *r* being a measure of the friction (friction is zero for $r = \omega$). As explained, we substitute $\dot{x} = v$ and obtain the two equations:

$\dot{x} = v$ and $\dot{v} = -\omega \cdot x - r \cdot v$

By looking at the discrete derivative the following difference quotient holds:

$\frac{\Delta x}{\Delta t} = v_t$ and $\frac{\Delta v}{\Delta t} = -\omega \cdot x_t - r \cdot v_t$.

For $\Delta t = 1$ we obtain:

$x_{t+1} = x_t + v_t$ and

$v_{t+1} = -\omega \cdot x_t + (1-r) \cdot v_t$.

These equations can be described by matrix W1:

$$\begin{pmatrix} 1 & 1 \\ -\omega & 1-r \end{pmatrix} \qquad \text{(W1)}$$

In the form

$$\begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{pmatrix}$$

the values can be used as weights of a recurrent network (Figure 4.2, in this case $w_{13} = w_{23} = 0$).

### 4.3.2 Free-fall

The same method can be applied to simulate the dynamics of a freely falling object. When considering the case without friction, i.e. in vacuum, the system is described by the differential equation $m \cdot \ddot{x} = -g \cdot m$ with $m$ being the mass of the object and $g$ the acceleration. The negative sign indicates that the objects fall downwards. Again we use the velocity $\dot{x} = v$ as auxiliary variable and obtain the following equations:

$\dot{x} = v$ and $\dot{v} = -g$

The difference quotients of the two equations with $\Delta t = 1$ again provide the basis for the weight matrix of the recurrent neural network:

$x_{t+1} = x_t + v_t$ and

$v_{t+1} = v_t - g$

The resulting matrix is given by W2:

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & -g \end{pmatrix} \qquad \text{(W2)}$$

This can be used as a weight matrix for the recurrent network in the form

$$\begin{pmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \end{pmatrix}$$

Thus, in contrast to the case of the oscillating system we here need a bias unit which has a constant activation of one (Figure 4.2). It has been shown (Chapter 3) that the weights of such a bias unit can also be trained using IC Units. The weight $w_{23}$ in Figure 4.2 describes the acceleration *g*.

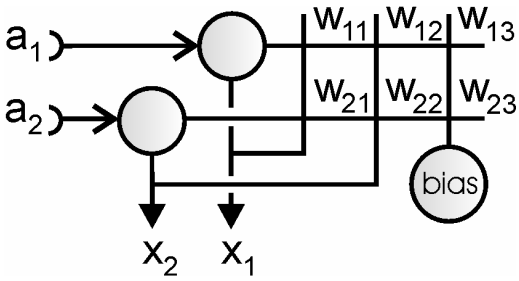


**Figure 4.2: Schematic drawing of a two unit recurrent neural network with a bias unit.**

Under realistic physical conditions though, friction decelerates the fall of an object. As friction is proportional to the velocity, it can simply be introduced in the system by adding a friction term to the second order differential equation: $\ddot{x} = -g - r \cdot \dot{x}$ with $r = \frac{k}{m}$ (*k* is constant and depends on the medium, *m* is the mass). The weight matrix W2 now changes to:

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & (1-r) & -g \end{pmatrix} \qquad \text{(W3)}$$

### 4.3.3 Low-pass & high-pass filter

It is also possible to build a recurrent neural network with low-pass filter properties defined by $\tau \cdot \dot{x} = -x + I$ with $\tau$ being the time constant and *I* the external input. This differential equation can be replaced by the two equations

$x_{t+1} = x_t + v_t / \tau$ and

$v_{t+1} = -x_t + I_t$

which leads to the weight matrix W4:

$$\begin{pmatrix} 1 & k & 0 \\ -1 & 0 & 1 \end{pmatrix} \qquad \text{(W4).}$$

Here $k$ determines the time constant of the system: $\tau = 1/k$. The output of the first unit shows the response of a low-pass filter and that of the second unit the response of a high-pass filter. This task requires a sensory input I.

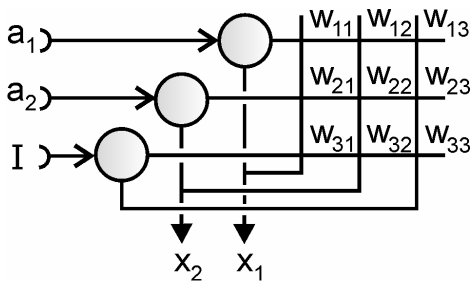


**Figure 4.3: Schematic drawing of a two unit recurrent neural network with an additional sensory input unit I.**

This third unit could manually be added to the network. However, it is also possible to add a third recurrently connected IC Unit (Figure 4.3) to the network the weights of which are trained in the same way as the other weights. This leads to the following weight matrix:

$$\begin{pmatrix} 1 & k & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{(W4a)}$$

This matrix shows that the external input does not depend on the activation of the other units.

Hence, in all of these cases a solution exists, but the question arises whether it is possible to automatically stabilise an appropriate weight matrix online. To achieve this goal, the network architecture described in Chapters 3 and 4.2 is used to cope with these tasks. In general, a network consisting of at least two IC Units is needed plus one bias

unit, but any network consisting of more units is also suitable for the tasks (see Chapter 4.5.1). In the simulation, the training data are provided by separate trainer networks with fixed values for frequency, friction or the time constant as described above. These networks are used to represent the real world situation. The network to be trained receives as input only the activation of position $x$ and of velocity $v$ from the trainer network. To avoid artefacts due to time discretisation of a computer simulation, $k$ should be chosen small enough (e.g. $k < 0.3$).

## 4.4 Results

When training a network which should be able to internally simulate dynamical systems like a pendulum, free fall or low-pass filters, the results will be evaluated according to the quality of the internal simulation on the one hand and the overall weight error on the other hand.

The quality of the internal simulation can be judged by calculating the accumulated local $\delta$-error $\delta_{acc}$ of the position unit ($\delta_{acc} = \sum_{j=1}^{n} |\delta_j|$ with $n$ describing the number of iteration steps of a single learning epoch) during each learning epoch. The $\delta$-error describes the difference between the external input and the sum of the weighted internal inputs (see Chapter 4.2). Thus, smaller $\delta$-errors correspond to better internal simulations. To be able to compare the results of the different dynamic simulations the accumulated $\delta$-error $\delta_{acc}$ is normalised to an epoch with a length of 100 iteration steps. Learning is switched off after this accumulated $\delta$-error $\delta_{acc}$ has fallen below a given threshold: the internal simulation of the process now resembles the external dynamics and the quality of the internal simulation is said to be high. This threshold was fixed to 0.01.

Additionally, the weight matrix learnt during training can be compared with the matrix derived from the differential equations (Chapter 4.3). After some learning epochs (the number of which depends on the learning rate $\varepsilon$) a weight matrix is learnt that approximates the desired matrix respectively (W1 up to W4). This can be seen by

looking at the overall error $E$ determined by $E = \sum_{j=1}^{n} \left( w_{ij} - v_{ij} \right)^2$ for $i = 1$ to $n$. Here, $w_{ij}$ are the weights of the IC network which are trained and $v_{ij}$ are the weights of the trainer network which are fixed according to the respective desired matrix. The weights $v_{ij}$ are, in this case, given a priori. They are, however, only used here for the calculation of the error value which is merely needed for descriptive purposes. The overall error signal $E$ does not influence the learning process itself. Learning depends on the local error only (see Eq. 2).

### 4.4.1 Pendulum

To train the properties of dynamic systems, as a first example a spring-mass pendulum is considered. A network consisting of two IC Units is provided with a temporal sequence of position and velocity values several epochs each lasting 190 iterations (Figure 4.4a shows 4 epochs).

Figure 4.4 shows an example of an IC network which was trained applying a learning rate of $\varepsilon = 0.8$ to internally simulate a system with a frequency $\omega = 0.05$ and a friction $r = 0.06$; presentation of the external dynamic system for four learning epochs is necessary for the model to be able to internally simulate the dynamics with high quality. The arrow in Figure 4.4b marks the iteration step when the accumulated $\delta$-error $\delta_{acc}$ has fallen below the threshold.

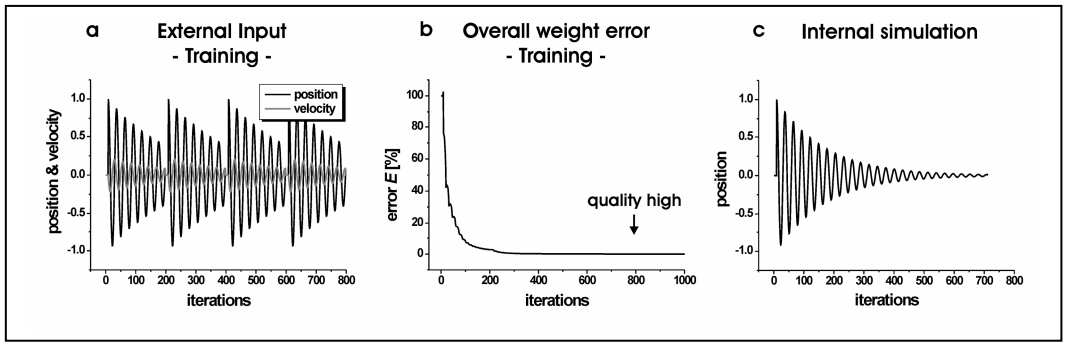


**Figure 4.4: An example of an IC network which was trained to simulate the dynamics of a pendulum. Frequency was chosen to be $\omega = 0.05$ and friction $r = 0.06$; a learning rate of $\varepsilon = 0.8$ was applied. (a) External training input. (b) Overall weight error *E;* the arrow indicates the time step when the accumulated $\delta$-error $\delta_{acc}$ has fallen below the threshold defined. (c) Internal simulation of the dynamics of a pendulum.**

After having stopped training the weights, the model is able to internally simulate the oscillations of the pendulum. Presentation of any position input to the network for one iteration step leads to damped oscillating behaviour similar to those of the training network (Figure 4.4c). Learning rates, which are larger than $\varepsilon = 1.7$, cause the network to diverge.

### 4.4.2 Free-fall

To learn the dynamics of freely falling objects, again two input values, one for position and one for velocity, have to be presented to the IC network. The training network is provided with a starting position value from which an object is supposed to fall down; in the example shown in Figure 4.5a (in a world without friction) and 5d (in a world with friction) the object is lifted up to the starting position of 1000*m*. Each time the object touches the ground a short interval of four iteration steps is inserted during which the neurons do not get excited. After this interval the object is lifted up again to its starting position.

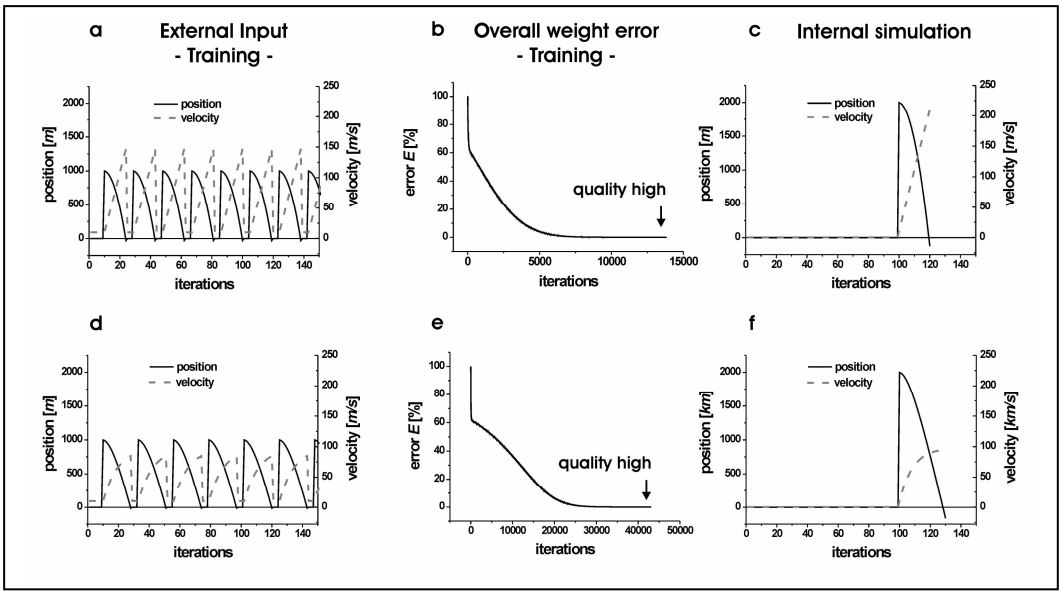


**Figure 4.5: Example of an IC network trained to represent the dynamics of freely falling objects in a world without friction (a-c) and in a world exposed to friction of $r = 0.1$ (b-f). In both cases learning rate was chosen to be $\varepsilon = 0.3$. (a) + (d) Position and velocity of the training input. (b) + (e) Overall weight error *E;* the arrow indicates the time step when the accumulated $\delta$-error $\delta_{acc}$ has fallen below the threshold defined. (c) + (f) Position and velocity of the internal simulation of freely falling objects not exposed to friction (c) and exposed to friction (f).**

Training a network to represent the dynamics of a freely falling object in a world without friction with an acceleration of $g = -9{,}81\frac{m}{s^2}$ (which corresponds to the acceleration due to gravity on earth's surface) and a starting position value of 1000*m* requires about 10000 to 13000 learning steps depending on the learning rate ($\varepsilon$ was chosen in the range of $[0.1, 0.3]$). Learning rates larger than 0.3 lead to divergence of the network. Under these conditions the object touches the ground after 14.14 *s*. In the simulation the process requires 15 iteration steps. Therefore, one iteration step approximately corresponds to one second real time.

When no friction is imposed, about 500 to 1200 examples of falling object are necessary for the model to be able to internally simulate the dynamics, i.e. the accumulated $\delta$-error of the position unit has fallen below the threshold (arrow in Figure 4.5b). If the object is dropped from a higher position, which results in an elongation of the time until the object hits the ground, fewer examples are necessary to train the network.

If, after having trained the weights, an arbitrary starting position is presented to the network (in the example shown in Figure 4.5c the height is set to 2000*m*), it is able to internally simulate the dynamics of free-fall because the resulting weight matrix almost equals matrix W2 ( $E = 3.24^{-3}\%$ ).

Similar results are obtained when the network is trained in a world that is additionally exposed to friction (Figure 4.5d-f). With a friction of $r = 0.1$ and an acceleration of $g = -9{,}81\frac{m}{s^2}$, learning rates of $\varepsilon = [0.1, 0.3]$ lead within about 45000 to 80000 iteration steps to solutions which allow for internal simulations of the dynamics. Due to the friction one learning epoch here lasts 19 iteration steps, i.e. approximately 19 *s*. Figure 4.5 d-f shows the results for training a system with a friction of $r = 0.1$ and a learning rate of $\varepsilon = 0.2$.

If friction is decreased also learning time decreases as long as the learning rate is kept constant: for a system with a friction of $r = 0.01$ it takes about 13000 to 30000 iteration steps until the accumulated $\delta$-error has fallen below the threshold when applying learning rates in the range of $[0.1, 0.3]$. Again, when using higher learning rates the network diverges.

### 4.4.3 Low-pass & high-pass filter

It is also possible to represent the dynamics of a low-pass filter (Chapter 4.3.3). For this example step responses (position $x$ and velocity $\dot{x}$) of a low-pass filter as well as the input function are presented to an untrained network consisting of three IC Units or consisting of two IC Units plus an additional bias unit. The IC network is provided with the inputs in periodically alternating increasing and decreasing step functions each lasting for 30 iterations. After the accumulated $\delta$-error of the position has fallen below 0.01, the model is able to internally simulate the dynamics of a low-pass filter. For a system the time constant of which is determined by $k = 0.03$ and using a learning rate of $\varepsilon = 0.1$, for example, this $\delta$-error has reached the threshold after about 41 learning epochs. Learning rates higher than 0.2 lead to instable results here. Corresponding results have been received when training a high-pass filter.

In all these cases the weight matrices emerging resemble those given by the differential equations, i.e. W1 to W4. As was mentioned above, any linear differential equation can be transformed to a recurrent neural network. It has been shown earlier that, for the static case, any such matrix can be learnt when IC Units are used (Chapter 3). However, a general proof for dynamic situations is still pending; such a proof additionally may help to define the exact parameter boundaries.

## 4.5 Combination of static and dynamic representation

As pointed out in the Introduction (Chapter 4.1), to account for cognitive capabilities of humans or animals it is necessary to be able to build internal mental representations which either can be static or dynamic. As an example we will treat the situation, of perceiving a scene, which can be described by a sentence like “Homer walks to Marge” in human language. Following Glenberg and others (e.g. Glenberg and Kaschak, 2002) the ability to internally simulate the motion of the moving object – in this case the walking person – is mandatory to really understand such a situation. Thus, the task is to represent static objects that may also be able to show dynamic properties.
In the above-mentioned examples of learning dynamic situations, we implicitly assumed the existence of some kind of “perception system”. This perception system contains

sensors and a network that is able to detect objects and properties of objects as for example their color, their position, or their velocity. We take this network for granted and do not deal with the question how it is constructed. To simplify matters, we further assume that the objects and each of their properties, when occurring in the world, activate one unit of this perception network; thus, the units are localist-encoding units not only for linguistic entities as used in former models (Chapter 2; Cangelosi, 2004), but also for properties of the objects as their position and velocity, for example. Each of these perception units projects to an IC Unit. These connections are also considered to be given. The perception network will not be illustrated in the figures.

Such a system is now confronted with an external situation as described in the sentence "Homer walks to Marge". This scene can be subdivided into two tasks: In the beginning the two persons, Homer and Marge, stand apart at a certain distance. Thus, the starting situation is a static one. Then one person, say Homer, starts to move. Therefore, both static and dynamic representations have to be combined to cope with the task of representing the scene.

In order to represent the static starting situation described in the sentence, i.e. Homer and Marge standing apart, three units are needed. The activation of neuron $x_i$ is assumed to represent the position of *Homer*, the activation of neuron $x_j$ *Marge*, and the activation of neuron $x_{dist}$ the distance between both. An IC network for this static situation can be described by the following equations:

$$x_i(t+1) = w_{11} \cdot x_i(t) + w_{12} \cdot x_{dist}(t) + w_{13} \cdot x_j(t)$$
$$x_{dist}(t+1) = w_{21} \cdot x_i(t) + w_{22} \cdot x_{dist}(t) + w_{23} \cdot x_j(t)$$
$$x_j(t+1) = w_{31} \cdot x_i(t) + w_{32} \cdot x_{dist}(t) + w_{33} \cdot x_j(t)$$

These equations are derived from the basic equation $x_i + x_{dist} = x_j$ and the situation can be trained using IC Units as has been shown in Chapter 3. A possible solution is given by the matrix

$$\begin{pmatrix} 2/3 & -1/3 & 1/3 \\ -1/3 & 2/3 & 1/3 \\ 1/3 & 1/3 & 2/3 \end{pmatrix} \quad \text{(W5)}.$$

Matrix W6 shows another solution:

$$\begin{pmatrix} 0 & -1 & 1 \\ -1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \qquad \text{(W6)}.$$

These are solutions for the static starting situation. But how could we cope with the problem of representing the motion from one point to another? One simple possibility is to assume Homer moving with a constant velocity. This movement then could be described by a simple integrator. However, we will deal with a somewhat more complex task: The movement of Homer should stop when he meets Marge. This is done here by assuming that Homer's movement corresponds to an exponential function as described by the step response of a low-pass filter (which might, for example, result from the movement being controlled by a negative feedback controller).

At first sight, there are two possible ways to solve the problem of representing this situation:

(i) Either there are two networks, one for the static situation and the other for the dynamic one, or

(ii) there is one unified network for representing both aspects. However, as will be explained in the following, the former could also be interpreted as representing one network. Therefore, only the latter will be considered in the simulation.

Taking the first possibility of using two separate networks to simulate the situation, six units are required: three units for the network representing the static starting situation (e.g. matrix W5 or W6) and an additional three unit network for the low-pass filter as described above (Section 3.3, matrix W4). The position unit of the low-pass filter codes for the position of the moving person, the velocity unit for its velocity, and its input unit represents the goal, i.e. the position of the person who is being approached. In neural terms this *move event* (Steels, 2003) could be represented by the decrease of the activation of the position unit until it equals the activation of the input unit. To start the

simulation of the movement, the position unit of Marge $x_j$ has to be connected to the input unit of the low-pass network. Furthermore, the position unit of the low-pass network and unit $x_i$, the position of Homer in the static network, describe the same value and therefore have to be connected, too. Therefore, the two connected networks could be regarded as one network.

Consequently, units $x_i$ and $x_j$ occurring in both networks can be merged respectively. Therefore, only one additional unit is necessary: the velocity unit of the dynamic network. This unit is termed *dynamic unit* $x_{dyn}$ in the combined network. In this way we come up with a network consisting of four units for the task (Figure 4.6).

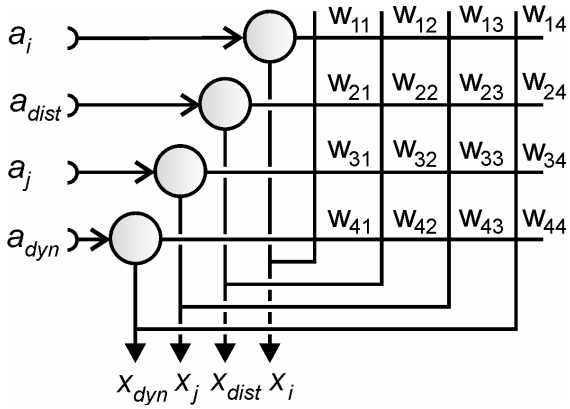


**Figure 4.6: A four unit recurrent neural network suitable for the task of concatenating a static and a dynamic situation to represent a *move event*. The units $x_i$ and $x_j$ represent the position of the respective person, $x_{dist}$ the distance between them and $x_{dyn}$ the velocity of the moving person.**

In the following it is described how this simpler version, a network consisting of four units, can be trained to represent both, the static and the dynamic part of a *move event*. Two different procedures to train this network are presented:

(i) The network will be trained in two phases, i.e. in the first step only the static starting situation is presented (*static phase*) and in the second step only the dynamic situation is presented (*dynamic phase*) (Chapter 4.5.1) and

(ii) training will be performed in a single phase where only the dynamic input is presented to the four unit network (Chapter 4.5.2).

## 4.5.1 Training the network in two phases

### 4.5.1.1 Static phase: Representing the static situation

Training the network can proceed in two steps. In the first step the static starting situation $a_i + a_{dist} = a_j$ is presented to the network depicted in Figure 4.6. Two possible solutions have been mentioned above (matrices W5 and W6). After this first training step the weight matrix of the network has the following structure (for details about learning simple algebraic relations see Chapter 3):

$$\begin{pmatrix} 2/3 & -1/3 & 1/3 & 0 \\ -1/3 & 2/3 & 1/3 & 0 \\ 1/3 & 1/3 & 2/3 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \qquad \text{(W7)}$$

The weights, which are non-zero, correspond to the solution W5.

For this task, unit $x_{dyn}$ is not yet necessary. However, the example illustrates that any additional unit belonging to the network does not change its weights as long as the unit does not receive an input different from zero.

### 4.5.1.2 Dynamic phase: Representing the dynamic situation

After having trained the network with this static situation, the second training step comprises the presentation of the dynamics of a low-pass filter. Thus, training the weights in the second phase does not start with a completely naïve network with all the weights being zero but with matrix W7.

How is the training performed? As in the earlier cases, in principle, the four unit network shown in Figure 4.7 receives its input via the perception network that, in turn, observes the outside world. The position $x_j$ of Marge is fixed to the value $a_j$ according to the actual training situation. Unit $x_{dist}$ is provided with the distance seen between Homer and Marge $a_{dist}$ in each iteration. For the simulation of the movement, we use a low-pass filter as dynamic training network which is connected to the IC network as shown in Figure 4.7. The dashed lines represent the function of the perception network. To train the movement of Homer from his starting position to Marge, unit $x_i$ (position of Homer) receives the position input ($x_1$ in Figure 4.7) of the low-pass filter network. The input

value of this network represents the desired position of Homer which is the position of Marge. The additional dynamic unit $x_{dyn}$ is provided with the velocity-signal ($x_2$ in Figure 4.7) of the dynamic training network.

During this second training phase different positions of Marge are presented to the dynamic training network in periodical epochs each lasting for 30 iteration steps. As in the earlier examples, the output of the training network provides the input for the network to be trained (Figure 4.7). Recall that this training network is only necessary for simulating the outside world: In a real environment neurons of the perception network (see above) will get activated by the objects, their position, and their velocity.

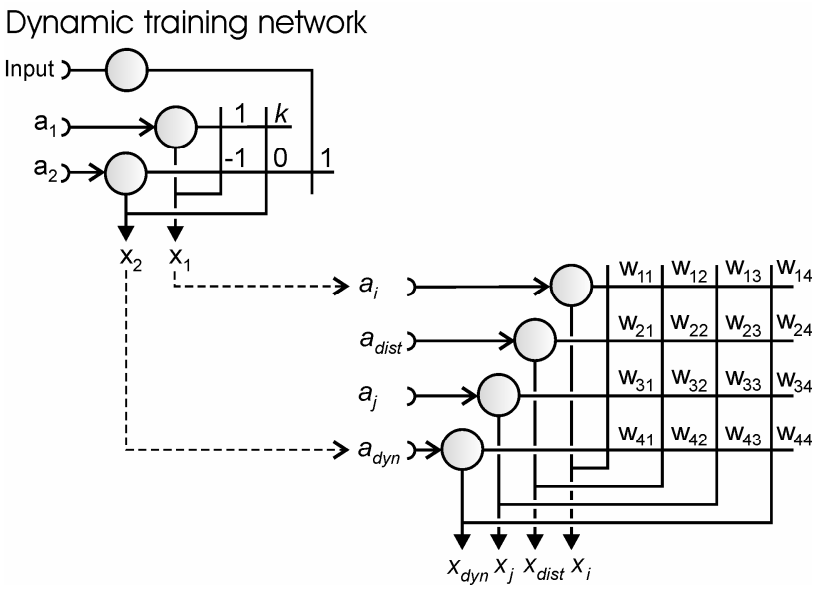


**Figure 4.7: Schematic drawing of training a network to represent a move event. The movement in the external world is simulated by dynamic training network representing the dynamics of a low-pass filter (upper left). The dashed arrows symbolise the perception network. Unit $a_i$ receives as input the positional value $x_2$ from the dynamic training network, unit $a_{dyn}$ the velocity value $x_2$.**

When a network that already represents the static situation is trained with the dynamic situation, the weights, which represent the starting situation, can either be locked so that no further changes are possible at these weights (see *dynamic phase A*) or they can be left plastic, i.e. they can further be changed in the second dynamic training phase (see *dynamic phase B*). As will be shown, both training procedures are successful but result in different weight matrices.

*Dynamic phase A: static weights are locked*

If all the weights having been learnt with the static training protocol (resulting in matrix W7) are locked, i.e. kept constant throughout the dynamic training phase, the following weight matrix is approximated:

$$\begin{pmatrix} 2/3 & -1/3 & 1/3 & w_{14} \\ -1/3 & 2/3 & 1/3 & w_{24} \\ 1/3 & 1/3 & 2/3 & w_{34} \\ -1 & 0 & 1 & 0 \end{pmatrix} \qquad \text{(W8)}$$

with $w_{14} = -k \cdot w_{12} + k$, $w_{24} = -k \cdot w_{22}$ and $w_{34} = -k \cdot w_{32}$; as described above, $k$ is a measure for the time constant of the low pass filter ($k = 1/\tau$). The overall weight error $E$ is calculated according to $E = \sum_{j=1}^{n} \left( w_{ij} - v_{ij} \right)^2$ for $i = 1$ to $n$ (see Chapter 4.4.1). Here $w_{ij}$ are the weights of the IC network which are trained and $v_{ij}$ are the weights of the final matrix W8.

To judge whether or not the internal simulation of the network is good, again the accumulated $\delta$-error $\delta_{acc} = \sum_{j=1}^{n} \left| \delta_j \right|$ is monitored. In this case, the accumulated $\delta$-error of the additional dynamical unit $x_{dyn}$ is a good measure for the learning process as the weights of this unit are only started to be learnt within this dynamic phase. Therefore, after this value as fallen below the threshold of 0.01 learning is stopped (marked by an arrow in Figure 4.8a).

In the example shown in Figure 4.8 the network was trained applying a learning rate of $\varepsilon = 0.2$ and using a dynamic training network having a time constant determined by $k = 0.1$. With this parameter configuration it takes about 660 iteration steps, i.e. 22 learning epochs, to train the network until the normalised accumulated $\delta$-error has fallen below the threshold of 0.01 leading to an overall weight error of $E = 2.42^{-3}\%$. If the learning rate is larger than 0.3 the network diverges.

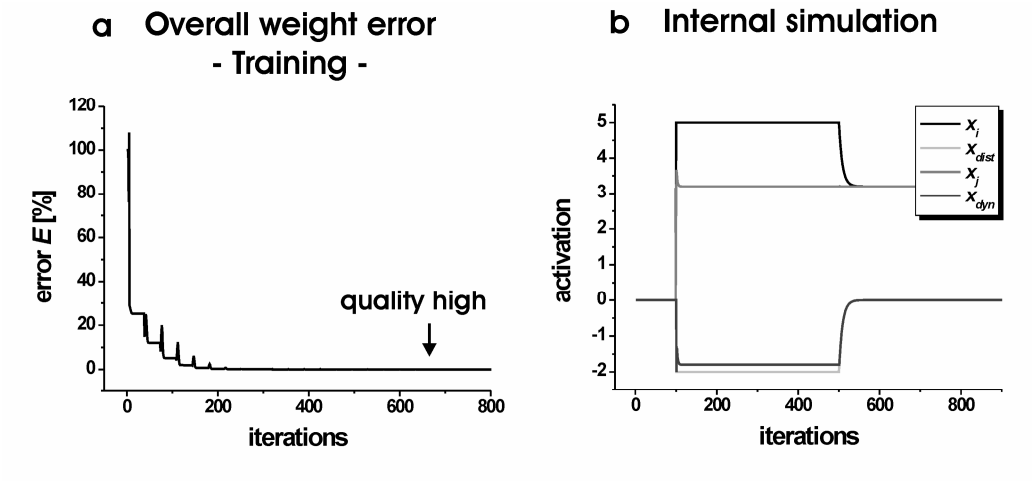


**Figure 4.8: Example of the networks' performance when locking the static weights. The time constant of the dynamic training network was determined by $k = 0.1$ and the learning rate set to $\varepsilon = 0.2$. (a) Overall weight error during training. (b) Internal simulation of the move event (for further explanation see text).**

The task for the network after training is to internally simulate the event "Homer walks to Marge". To this end the network first is provided with an external input mirroring the situation of two persons standing apart, i.e. a vector the components of which follow the equation $a_i + a_{dist} = a_j$. After some time one person should start moving towards the other; hence, the activations of the units $x_i$ and $x_{dist}$ should change according to the situation to be mentally simulated while only the activation of unit $x_j$ has to stay at a constant level. Therefore, units $x_i$ and $x_{dist}$ do not receive an external input any longer and the activation of $x_j$ is kept constant. An example of such a dynamical process is illustrated in Figure 4.8b: The activation of unit $x_i$ decreases until it equals the activation of $x_j$.

### *Dynamic phase B: changing all the weights*

Instead of locking some of the weights during the second phase of the training procedure all the weights can be allowed to be changed in this second phase, i.e. also the weights of the static part learnt in the first phase. Applying this procedure the weights converge to a weight distribution shown in matrix W9:

$$\begin{pmatrix} 1 & w_{12} & 0 & w_{14} \\ -1 & 1/3 & 1 & -1/3 \\ 0 & 1/6 & 1 & -1/6 \\ -1 & 0 & 1 & 0 \end{pmatrix} \qquad \text{(W9)}$$

with $w_{12} = (k + w_{12stat})/2$ and $w_{14} = (k - w_{12stat})/2$; $w_{12stat}$ is the value of the weight $w_{12}$ adopted in the static training phase which is $w_{12stat} = -1/3$. Thus, the weights $w_{12}$ and $w_{14}$ add up to the value of *k*, which determines the time constant of the respective dynamic training network. The time in which this solution is reached during the second training phase depends on the one hand, of course, on the learning rate $\varepsilon$, but also on the value of *k*. The higher *k* is chosen the faster matrix W9 is attained.

Figure 4.9 shows an example of the behaviour of a network trained by using the same time constant ($k = 0.1$) and learning rate ($\varepsilon = 0.2$) as in the example of Figure 4.8. Here, also 22 learning epochs are necessary until training can be stopped. The overall weight error has decreased to $E = 1.36^{-5}\%$.

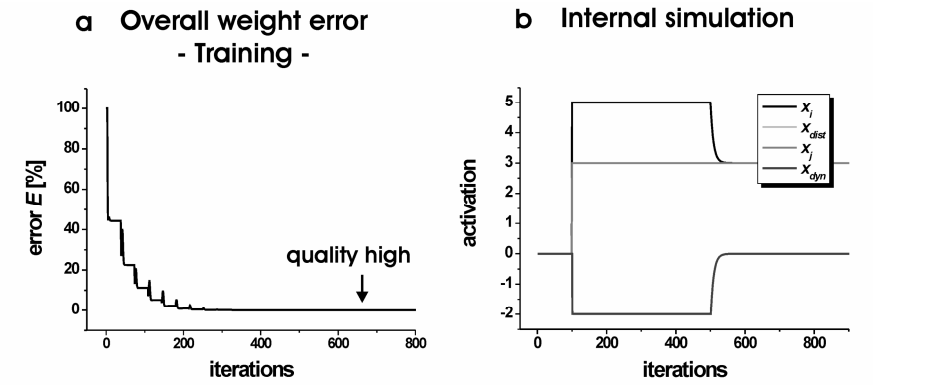


**Figure 4.9: Example of the networks' performance when changing all the weights. The time constant of the dynamic training network was determined by $k = 0.1$ and the learning rate set to $\varepsilon = 0.2$. (a) Overall weight error during training. (b) Internal simulation of the move event (for further explanation see text).**

After having trained the network it has the ability to internally simulate the situation of one person moving towards the other (Figure 4.9b).

### 4.5.2 Training the network in one step

In contrast to these types of training procedures separated into two steps, training in one step is also possible. In order to perform this, simply the first phase of learning to

represent the static input situation is skipped. The dynamic training of the second phase is executed in the same way as described above. Applying this shortened training method results in weight matrix W10:

$$\begin{pmatrix} 1 & k/2 & 0 & k/2 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 0 & 1 & 0 \end{pmatrix} \quad \text{(W10)}$$

Like all other training procedures this method leads to a weight matrix only dependent on the time constant of the dynamic training network. Numerical results show that the smaller $k$ is, the longer the network has to be trained in order to be able to internally simulate the situation.

Thus, different possibilities exists how the content of a sentence like “Homer walks to Marge” can be represented. Both the information about the objects as well as the information about the motion can be merged within a single representation.

## 4.6 Discussion

Internal mental representations are an important prerequisite for cognitive behaviour. Therefore, when trying to model cognitive abilities of humans or animals or when building agents that are supposed to behave cognitively, the goal is to model a general memory system that is able to store and retrieve various contents. This information is assumed to be stored in the form of situation models which in general connect perception to action. Studies of mirror neurons indicate that situation models consist of separate neural assemblies (e.g. Fogassi et al., 2005) supporting earlier assumptions (Wolpert and Kawato, 1998).

Addressing this goal raises several questions: how are such situation models learnt? Are these models ordered in a parallel or some hierarchical structure? How is it possible to serially connect different models to produce sequential behaviour? And, of course, how are these models selected and activated? Here, we concentrate on the question how such

situation models could be realised in a biologically plausible neuronal structure and how individual models could be learnt. Storing static patterns has been addressed in Chapter 3. Here, we concentrate on models storing dynamic patterns, and as a first step, try to combine static and dynamic properties within one model.

Two abilities of humans in representing dynamical systems have been focussed within this chapter: the capability of representing acceleration of objects as can be found in the movement of a pendulum or freely falling objects and representing actions of transfer, i.e. motion from one point to another. The recurrent networks consisting of IC Units used here are well suited to simulate those various aspects of dynamic internal representations. When provided with the respective dynamical input they adapt to represent the situation existing in the outside world.

Studying the combination of static and dynamic models has shown that storing dynamic properties together with the static information appears to be more parsimonious than application of a separate, multipurpose dynamic model that is activated together with a static model on request. Learning is possible in a sequential order, first the static information and later the dynamic part. However, it is also easily possible and faster to learn both aspects at the same time. Furthermore, this procedure has led, at least in our examples, to simpler weight matrices.

The results presented are particularly interesting because the IC Units used here apply a simple, biologically plausible and local learning algorithm. The individual units in this recurrent network are not specified as to be of entirely sensory or motor type. Such a separation is generally not possible in this 'holistic' network. Rather, the situation model may be used for perception as well as for control of action (Cruse, 2003).

### 4.6.1 Combination of static and dynamic representations

The model we proposed here is able to simulate diverse aspects of dynamic internal representations and has already been shown to perform well when representing static situations (Chapter 3). Here, we were also able to show that both the static and the dynamic representations can be merged within one unified network.

The different training procedures lead to different weight matrices. This is, as discussed in Chapter 3, due to the fact that the task is underdetermined. The results depend on the values of the weights at the beginning of training (to simplify matters, in all experiments the values of all weights are set to zero at the beginning) and on side conditions, as for

example holding some of the weight at fixed values or using different training examples. Already for the static task two different solutions were presented, one with diagonal weights (responsible for the self-excitation of the unit) being zero (W6), the other with positive values for the diagonal weights (W5). The latter produces a network having some kind of inertia (for details see Chapter 3). The identity matrix with all weights being zero except for the diagonal weights is, of course, also a solution, even though a trivial one.

As for the matrices found in the dynamic training phases no mathematical proof is available yet, we will give a phenomenological description of the results. Interestingly, the weights of unit $x_{dyn}$ are the same in all cases investigated and correspond to the weights occurring in the lower line of matrix W4 determining the velocity unit of the low-pass filter. Therefore, these weights appear to represent the only solution possible.

What about the weights of the other units of the matrices? The most perspicuous solution is adopted in matrix W10. The weights of unit $x_i$ can be interpreted as follows: $w_{11} = 1$ provides the actual position of the moving person corresponding to the upper line of matrix W4. In contrast to matrix W4 the positional change is distributed over $w_{12} = k/2$ and $w_{14} = k/2$, namely the contribution of $x_{dist}$ and $x_{dyn}$. As both units provide the same information, the effect of the positional change is equally distributed to both weights. The weights of unit $x_{dist}$ (second line) approach the solution of matrix W6 with the diagonal weight $w_{22} = 0$. The weights of unit $x_j$ (third line) stabilise on the trivial solution with only the self-exciting weight $w_{33}$ being one and all other weights being zero. Apparently this simple solution is found when the value is always constant within a training epoch and not dependent on the other units.

Matrix W9 results from a similar training situation like W10 as all weights are free to change. The difference, however, is that training of the dynamic situation in case of W9 starts with weight values different from zero. Correspondingly, the results are similar but not identical. The weights $w_{12}$ and $w_{14}$ of unit $x_i$ add up to the value of $k$ in both cases, W9 and W10. In the case of W10 both weights contribute in exactly the same ratio, i.e. $w_{12} = w_{14} = k/2$, whereas the distribution in the case of matrix W9 also depends on the weight values $w_{ijstat}$ adopted in the static training phase. In this case the weights are $w_{12} = k/2 + w_{12\,stat}/2$ and $w_{14} = k/2 - w_{12\,stat}/2$. The second terms depending

on $w_{12stat}$ respectively balance each other. In the second line of matrix W9, weights $w_{21}$ and $w_{23}$ correspond to that of the static solution in W5. However weight $w_{22}$ is not zero in W9, but attains the value $w_{22} = w_{22\,stat}/2$ which is the mean value between $w_{22stat}$ and zero. The contribution of this weight has therefore be counteracted by weight $w_{24} = -w_{22}$. In the third line weights $w_{31} = 0$ and $w_{33} = 1$ selected the trivial solution (see W10). For weights $w_{32}$ and $w_{34}$ the explanation corresponds to that given above for $w_{22}$ and $w_{24}$: $w_{34} = -w_{32}$ balances the influence of $w_{32} = w_{32\,stat}/2$.

Matrix W8 was obtained when the static weights were kept constant while learning the dynamic situation. As units $x_{dist}$ and $x_{dyn}$ provide the same information, the second and the fourth column of matrix W8 can be rewritten for explanation purposes:

$$\begin{pmatrix} 2/3 & -1/3\cdot(1-k) & 1/3 & k \\ -1/3 & 2/3\cdot(1-k) & 1/3 & 0 \\ 1/3 & 1/3\cdot(1-k) & 2/3 & 0 \\ -1 & 0 & 1 & 0 \end{pmatrix} \qquad \text{(W8a)}$$

This illustrates that the positional change of the moving person $x_i$ is mediated mainly by the weight $w_{14}$ and depends on the time constant of the low-pass filter ($k$). Matrix W8a shows, that the influence of the dynamical units $x_{dist}$ and $x_{dyn}$ is not compensated totally as in the results discussed above (W9 and W10); therefore, the actual value of the velocity (here given by $x_{dist}$ and $x_{dyn}$) has to be taken into account: this is the value from the previous iteration step which decreases according to the factor $1-k$ when the low-pass filter increases.

Thus, different possibilities exists how the content of a sentence like “Homer walks to Marge” can be represented. The individual results can be explained, but a general proof does not exist up to now and it is not clear why the different solutions are found in the cases considered. Nevertheless, with the model presented here the dynamics of situations can internally be represented. Therefore, they provide a possible neural basis for mental simulations of such situations. They also can serve for recognition purposes; if a model of a special dynamical situation has been learnt it will get highly activated

when the situation occurs in the outside world again. This activation can be interpreted as recognition of the situation learnt before. Furthermore, the networks could be used to control motor output in order to produce movements according to the dynamics of the respective representation. This means that the same type of networks can be used for imagination, perception and the control of action (Cruse, 2003). Accordingly, these networks might form a basis to explain findings from psychological studies as mentioned above as well as neurophysiological results concerning the so-call mirror neurons (for a review see Rizzolatti and Craighero, 2004).

The results show how both the information concerning static objects and the information concerning motion can be merged within a single representation. An alternative and at first sight more parsimonious solution could be that within the brain the dynamical properties, for example low-pass filter properties, are stored in a specific, separate network and not in combination with the respective situation. This separate network might then be connected to any actual static situation if necessary. As, however, our results have shown that only one additional unit is sufficient to cope with this combined task, the latter solution might finally be the simpler one, in particular when the problem of establishing the connections between the static network and the respective dynamic one is taken into account.
Our examples for dynamic properties are based on differential equations of first or second order. Movements of constant speed can be simulated simply by using an integrator (e.g. first line in matrix W1). Actually, we investigate nonlinear versions of IC Units which might allow for modelling more complex dynamic properties.

### 4.6.2 Representation of dynamical systems

Perception and representation of motion is a fundamental property of the neural system. It is crucial for survival for example to estimate the time of contact with accelerated objects to react properly. The simulations of two accelerated objects, a pendulum and freely falling objects, show that the IC model is able to calculate the effects of external forces as for example gravity. Thus, our model provides a neural solution for the finding that humans seem to have such an internal model of acceleration which has been postulated to exist by Indovina et al. (2005) based on their experiments performed in a spacelab (McIntyre et al., 2001). In general, we assume that any dynamical system

which can be represented by a linear differential equation of order *n* can be learnt by a recurrent neural network of the type used here.

Additionally, experiments have revealed that understanding of situations is aided by internally simulating the dynamical aspects of perceived situations by grounding them in bodily activity (Glenberg and Kaschak, 2002), a fact which has often been ignored up to now. The ability of representing object motion by means of the dynamics of a low-pass filter can conceptually be applied to concrete actions. Thus, sentences describing physical motions as "Homer walks to Marge" can be represented by the described network (Chapter 4.5). But following Glenberg and Kaschak (2002) this idea can also be generalised to abstract situations when taking into account the considerations of *Construction Grammar* (Goldberg, 1995; Fillmore, 1988). The construction grammarians argue that also constructions themselves carry a general meaning independent of the single lexical items the sentences consist of. Kaschak and Glenberg (2000) provide a test to verify Goldbergs notion of construction by using made-up denomial verbs, i.e. invented verbs generated from nouns like "to crutch". They have shown that not only children – as has been demonstrated in language acquisition studies (e.g. Pinker, 1989) – but also adults are sensitive to the meanings associated with particular constructions (see also Naigles and Terrazas, 1998; Fisher, 1994).

Glenberg and Kaschak (2002) focus in their paper on double-object constructions. These constructions consist of "subject – verb – $object_1$ – $object_2$" and carry the meaning that the subject transfers $object_2$ to $object_1$ as "You give Liz the toy". This can also be applied to the double-object constructions when not a physical object but a piece of information is transferred, as in "You told Liz the story." Glenberg and Kaschak state (2002: 563):

> "That is, we come to understand the sentence as a physical movement from "you" to "Liz." To say it differently, over the course of learning the English double-object construction, we learn to treat the construction as an instruction to simulate a literal transfer of an object from one entity to another even when the object being transferred is not a physical object. This simulation is consistent with the claim that people understand communication as a type of transfer in which words act as containers of information (Lakoff, 1987)."

They argue that almost all language expressions, even abstract ones like the notion of cause, can be explained by their embodied analysis the core of which is that humans

tend to conceptualise most expressions by means of action involving bodies. (An extreme example of explicit bodily identification is reported by Ochs et al. (1996): They observed physicists while discussing new hypotheses: "When I come down I'm in the domain state." With "I" temperature is meant here. Ochs et al. noted that explicit identification with the body like this was often used when difficult hypotheses had to be understood.) Thus, the considerations made regarding actions of transfer can be generalised– at least to some extent – and are also applicable to more abstract actions where no real physical objects are involved.

The idea that internal representations comprise also the dynamical aspects of situations has already been put forward within the so-called scanpath-theory by Noton and Stark (1971). They found that each object is memorised and stored in memory as an alternating sequence of object features and eye movements required in reaching the next feature of the object. This dynamical aspect is supported by the results of brain imaging studies, too. A recent fMRI study located the internal model of gravity humans are supposed to have in the vestibular cortex as these networks are selectively engaged when acceleration is consistent with natural gravity (Indovina et al., 2005). Another study using event-related fMRI in humans has shown that reading action verbs activates classical language area, i.e. left inferior frontal and superior temporal cortex (Broca's and Wernicke's areas) as well as frontocerebral motor regions, including motor and premotor cortex (Hauk et al., 2004). Even static objects conveying motion activate brain regions engaged in the perceptual analysis of visual motion (Kourtzi and Kanwisher, 2000). Hence, these data support the idea resulting from the linguistic and psychological experiments explained above that processing word meanings involves dynamic representations.

### 4.6.3 Recombination of mental elements – Future work

The model introduced allows us up to now to model internal representations of static situations (Chapter 3) and, as shown here, dynamic situations as well as combinations of both. The next step is to ask how it might be possible to integrate a number of individual representations – static or dynamic ones – within larger frameworks. One question is whether activation of different situation models should strictly exclude each other or if parallel activation, i.e. blending of models, is sensible (Wolpert et al., 2003).

More fundamental problems encountered here are how the complete system decides where in the brain the information concerning the actual situation has to be stored, and, if it has already been stored earlier, to recognise this situation as a known one.
These problems include the question whether and how information is organised in any kind of hierarchical structure as it appears to be the case and as it is generally assumed. However, different types of connections may exist. A magpie, a sparrow, and a robin for example belong to the category "birds", a connection, which is described by an *is-a* relation according to the theory of semantic networks (Sowa, 1991). Each of these examples has, for instance, wings and feathers, a connection described by a *has-a* relation. However, such simple tree-like hierarchies are not sufficient. The actually used hierarchy may depend on the context. A bat may be considered to belong to the category of flying animals, together with (many, but not all) birds, or it could be regarded as a mammal and birds do not belong to this category. So, it must be possible to adapt the hierarchy in a dynamical fashion to the actual context. The ability to change the hierarchical order is a prerequisite for the ability to adopt the viewpoint of another person, which, according to Tomasello (2000) develops in human infants at an early age.

Another very important and related aspect is whether and how the integral system is equipped with the capability to find new categories, i.e. to combine stored models to form groups of related items, or chunks. These new self-invented models can be regarded as representing symbols (Steels, 1999) and as such be combined to other categories. This capability would also account for the power of recursion, i.e. the faculty of embedding different items into each other.
This faculty provides humans with a high degree of flexibility which is thought to be a decisive feature of human intelligence (Premack, 2004): both should enable the organism to combine mental elements – motor primitives or more abstract representations – to generate a more or less unlimited repertoire of behaviour in order to be as flexible as possible.
Specifically, human's recursive grammar allows for embedding one instance of an item in another instance of the same item. Owing to recursion humans are able to widely separate words in a sentence which yet depend on one another (Premack, 2004). This is

a key feature of human language in contrast to other animal communication systems which enables humans to create an open-ended and limitless system of communication (Chomsky, 1957; Chomsky, 1959; ;Chomsky, 1965; Hauser et al., 2002).

The third, also tightly related question concerns serial connection of such models. As discussed by Wolpert et al. (2003) this might be possible by using the output of one model as describing a new situation to which other models might react. This would require an internal feedback connection that connects the output of the integrated system to its input. This feedback connection then so to speak replaces the external input. Such an internal loop might allow the production and imagination of longer behavioural chains.

Finally, we want to address briefly a problem only rarely considered in the context of modelling mental representations. Any neural network model proposed comprises a mathematical or physical model describing the hypothetical mechanism. A person performing a mental simulation, however, experiences a phenomenal aspect, a subjective experience, which is a domain of course, not tackled when considering such physical models. It is an open question whether a biological neuronal network after being activated to form a situation model at the same time is sufficient to elicit phenomenal aspects (Cruse, 2003). Here, we do not want to further this philosophical discussion (Cruse, 1999) but, at least point to the fact that when dealing with modelling internal simulations in artificial systems, we tightly approach such philosophical aspects.

## 4.7 References

Barsalou LB (1999) Perceptual symbols systems. *Behavioral and Brain Sciences* 22: 577-660.

Brouwer A-M, Brenner E, Smeets JBJ (2002) Perception of acceleration with short presentation times: Can acceleration be used in interception? *Perception and Psychophysics* 64: 1160-1168.

Cangelosi, A (2004) The sensorimotor bases of linguistic structure: experiments with grounded adaptive agents". In *Proceedings of the 8th International Conference on Simulation of Adaptive Behavior*, Cambridge, MA: MIT Press, pp. 487–496.

Chomsky N (1957) *Syntactic structures*. The Hague: Mouton.

Chomsky N (1959) A Review of Skinner's *Verbal Behavior*. *Language* 35: 26-58.

Chomsky N (1965) *Aspects of the theory of syntax*. Cambridge, MA: MIT Press.

Cruse H (2003) The evolution of cognition - a hypothesis. *Cognitive Science* 27: 135-155.

Cruse H (1999) Feeling our body - the basis of cognition? *Evolution and Cognition* 5: 162-173.

Fillmore C (1988) The mechanics of "Construction Grammar". *Berkeley Linguistics Society* 14: 35-55.

Fincher-Kiefer R (2001) Perceptual components of situation models. *Memory & Cognition* 29: 336-343.

Fisher C (1994) Structure and meaning in the verb lexicon: Input from a syntasx-aided verb learning procedure. *Language and Cognitive Processes* 9: 473-518.

Fogassi L, Ferrari PF, Gesierich B, Rozzi S, Chersi F, Rizzolatti G (2005) Parietal lobe: From action organization to intention understanding. *Science* 308: 662-666.

Freyd JJ (1993) Five hunches about perceptual processes and dynamic representations. In: *Attention and performance XIV: Synergies in experimental psychology, artificial intelligence, and cognitive neuroscience* (D. Meyer, S. Kornblum, eds.), Cambridge, MA: MIT Press, pp 99-119.

Freyd JJ, Finke RA (1984) Representational momentum. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 10: 126-132.

Gibson JJ (1979) *The ecological approach to visual perception*. Boston: Hougthon Mifflin.

Glenberg AM, Kaschak MP (2002) Grounding language in action. *Psychonomic Bulletin & Review* 9: 558-565.

Glenberg, AM, Robertson, DA (1999) Indexical understanding of instructions. *Discourse Processes* 28, 1-26.

Glenberg AM, Robertson DA (2000) Symbol grounding and meaning: A comparison of high-dimensional and embodied theories of meaning. *Journal of Memory and Language* 43: 379-401.

Goldberg AE (1995) *A construction Grammar approach to argument structure*. Chicago: The University of Chicago Press.

Hauk O, Johnsrude I, Pulvermuller F (2004) Somatotopic representation of action words in human motor and premotor cortex. *Neuron* 41: 301-307.

Hauser MD, Chomsky N, Fitch T (2002) The faculty of language: What is it, who has it, and how did it evolve? *Science* 298: 1569-1579.

Indovina I, Maffei V, Bosco G, Zago M, Macaluso E, Lacquaniti F (2005) Representation of visual gravitational motion in the human vestibular cortex. *Science* 308: 416-419.

Johnson-Laird PN (1983) *Mental models: towards a cognitive science of language, inference, and consciousness*. Cambridge: Cambridge University Press.

Kaschak MP, Glenberg AM (2000) Constructing meaning: The role of affordances and grammatical constructions in sentence comprehension. *Journal of Memory and Language* 43: 508-529.

Kourtzi Z, Kanwisher N (2000) Activation in human MT/MST by static images with implied motion. *Journal of Cognitive Neuroscience* 12: 48-55.

Lakoff G (1987) *Women, fire, and dangerous things: What categories reveal about the mind*. Chicago: University of Chicago Press.

Lisberger SG, Movshon JA (1999) Visual motion analysis for pursuit eye movements in area MT of macaque monkeys. *Journal of Neuroscience* 19: 2224-2246.

McIntyre J, Zago M, Berthoz A, Lacquaniti F (2001) Does the brain model Newton's laws? *Nature Neuroscience* 4: 693-694.

Naigles, LR, Terrazas, P (1998) Motion verb generalization in English and Spanish: Influences in language and syntax. *Psychological Science* 9: 363-369.

Nauck D, Klawonn F, Borgelt C, Kruse R (2003) *Neuronale Netze und Fuzzy Systeme*. Braunschweig/ Wiesbaden: Vieweg-Verlag.

Noton D, Stark L (1971) Scanpaths in saccadic eye movements while viewing and recognizing patterns. *Vision Research* 11: 929-942.

Ochs E, Gonzales P, Jacoby S (1996) "When I come down I'm in the domain state": Grammar and graphic representation in the interpretative activity of physicists. In: *Interaction and grammar* (Ochs E, Schelgloff EA, Thompson SA, eds.), New York: Cambridge University Press, pp 328-369.

Pinker S (1989) *Learnability and cognition: The acquisition of argument structure*. Cambridge, MA: MIT Press.

Premack D (2004) Is language the key to human intelligence? *Science* 303: 318-320.

Rizzolatti G, Craighero L (2004) The mirror-neuron system. *Annual Review of Neuroscience* 27: 169-192.

Steels L (1995) Intelligence - Dynamics and Representations. In: *The Biology and Technology of Intelligent Autonomous Agents* (Steels L, ed), Berlin: Springer, pp 72-89.

Steels L (1999) *The talking head experiment*. Antwerpen: Special pre-edition for LABORATORIUM.

Steels L (2002) Simulating the evolution of a grammar for case. In: *Proceedings of the Fourth Evolution of Language Conference*, Harvard University.

Sowa JF (ed.) (1991) Principles of Semantic Networks: Explorations in the Representation of Knowledge. San Mateo, CA: Morgan Kaufmann Publishers.

Todd JT (1981) Visual information about moving objects. Journal of Experimental Psychology: *Human Perception & Performance* 7: 975-810.

van Dijk TA, Kintsch W (1983) *Strategies in text comprehension*. New York: Academic Press.

von Eckardt B (1993) *What is cognitive science?* Cambridge, MA: MIT Press.

von Eckardt B (1999) Mental representation. In*: MIT Encyclopedia of the Cognitive Sciences* (Wilson RA, Keil FC, eds), Cambridge, MA: MIT Press, pp 527-529.

Widrow B, Hoff ME (1960) Adaptive switching circuits. In: *1960 WESCON Convention record Part IV*, New York: Institute of Radio Engineers, pp 96-104.

Wolpert DM, Doya K, Kawato M (2003) A unifying computational framework for motor control and social interaction. *Phil Trans R Soc Lond B* 358: 593-602.

Wolpert DM, Kawato M (1998) Multiple paired forward and inverse models for motor control. *Neural Networks* 11: 1317-1329.

Zwaan RA, Madden CL, Yaxley RH, Aveyard ME (2004) Moving words: dynamic representations in language comprehension. *Cognitive Science* 28: 611-619.

Zwaan RA, Radvansky, Gabriel A. (1998) Situation models in language comprehension and memory. *Psychological Bulletin* 123: 162-185.

# 5 Discussion

## 5.1 Use of models in sciences

Understanding the brain and the various functions it fulfils is an intriguing task. Due to its complexity it is not only intriguing but also very difficult to disclose the brains mysteries. There are different approaches possible to gain insights. Neuroscientists are able to explain the structure of the brain on cellular and molecular level in more and more detail and electrophysiological recordings as well as brain imaging techniques help to elucidate brain functions. But explaining higher cognitive functions is fairly difficult, because the nervous system is structured in many different levels ranging from the molecular level to the systems level each of which has its own important aspects. Some properties might not be found when looking at lower level components but emerge from the interaction between these components on higher levels (Sejnowski et al., 1988). Such emergent properties may possibly only be understood by application of models.

A vast amount of literature originating from quite different fields like philosophy, cybernetics and cognitive sciences, to name only some, exists on the relevance and meaning of models (for a deep discussion on modelling see Webb, 2001) which also mirrors the confusion about what exactly is meant by the term *model* in relation to science (Leatherdale, 1974). Wartofsky (1979) has called this lack of agreement “model muddle”. But there seems to be a general agreement that models are representations of entities of the real world (Webb, 2001).

Of course, the benefit of using models is discussed controversially. Some researchers argue that “developing formalised models for phenomena which are not even understood on an elementary level is a risky venture: what can be gained by casting some quite gratuitous assumptions about particular phenomena in mathematical form?” (Croon and van de Vijver, 1994:4-5). Others in contrast put forward the demand of theoretical frameworks because of the complexity of animal behaviour (Barto, 1991). As, for example, the nervous system is far to complex to be understood experimentally, quantitative approaches provided by modelling are supposed to be necessary (Bower, 1992). In this sense, modelling could help to unveil what the relevant structures or essential features a system is composed of are. Thus, models offer a possibility of better

understanding and probing experimentally obtained results. Obviously, models will not solve the problems by themselves and do not replace experiments but they could amplify one's intuition and could probably reveal new phenomena and thus provide a basis for deeper insights in the working brain.

## 5.2 Models as part of the process of explanation

To explain the use of models Webb (2001) proposed a framework for the role of modelling as part of the process of explanation and prediction of certain target behaviours. A modified version of this framework is shown in Figure 5.1. This framework may be helpful to verify on the one hand the benefits the models presented here actually have and on the other hand to define which directions future work based on these models has to head for.

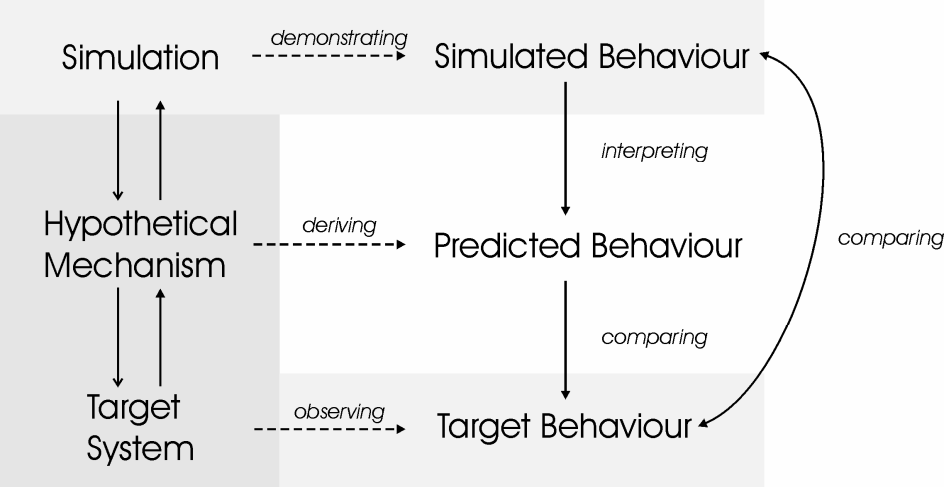


**Figure 5.1: Models as part of the process of explanation (adapted from Webb, 2001). Shaded grey parts are tackled within this work. Parts shaded in lighter grey allow for further research.**

Webb (2001) points out that the term *model* can be applied to different parts of the diagram depending on the viewpoint. Some consider the target system itself to be a model because selecting a system from the world already involves abstraction or simplification (Cartwright 1983). Other approaches like the "semantic" approach to scientific explanation (Giere 1997) regard the hypothesis to be a model because it specifies a hypothetical mechanism or system the target belongs to. The latter use of the

term *model* is quite common in contrast to the former which is only rarely found (for a detailed explanation see Webb, 2001).

In the work presented here the term *model* is taken to correspond to the box labelled *Simulation* just as Webb herself uses the term. In this sense, models are added to the cycle between hypothesis, prediction and observed behaviour. Thus, modelling is understood to support producing predictions from the hypothesis. Here, we do not claim to build a realistic model in the sense of Churchland and Sejnowski (1988: 744), which are "genuinely and strongly predictive of some aspects of nervous systems dynamics or anatomy" ,but so-called simplifying models: These models are models, "which though not so predictive, demonstrate that the nervous system could be governed by specific principles". These specific principles are the learning dynamics and in the case of the IC models the neuronal structure. Nevertheless the models can be explained within the framework proposed by Webb (2001) and shown in Figure 5.1. The parts embraced within this work are marked by shaded grey boxes.

As the goal was to simulate internal models which represent external situations, i.e. situation models, in fact we use the word *model* on two levels: on the one hand model is used in terms of simulation, on the other hand the overall target system itself is a model, namely the internal model consisting of many combined neurons; that means we are modelling models.

### 5.2.1 The target system: The neuron

The target system to be modelled here are neurons connected to build networks, which are thought to be the basic units enabling the brain to fulfil any function, in this case especially building up internal representations. Such internal representations can be multifaceted (Chapter 3 and 4; see also Kühn and Cruse, 2005): they could resemble static scenes and situations characterised by any kind of dynamics like acceleration or movements from one point to another as well as more or less abstract rules. As pointed out in the Introduction (Chapter 1) this is an important capability in order to behave cognitively and adaptively because internal representations allow the organism to predict the distinct consequences in the external world of distinct behavioural options.

### 5.2.2 Hypothetical mechanism: Self-Organisation

Just as snowflakes come into existence without a snowflake-maker, internal representations of the external world emerge from the interactions between the low-level components, the neurons, only without any supervising force, i.e. in a process of self-organisation. The hypothetical mechanisms in both models proposed comprise a local rule instantiated on the level of a single neuron. Therefore, no ordering influence besides the information from the external world is necessary to enable the organism to internally represent the external information.

When implementing the learning mechanism it is important to be cautious not to use the wrong internal activation values for training the weights. It is necessary to use the values of the same and not the subsequent iteration step.

### 5.2.3 Simulation: Entire recurrent neural network

If many neurons equipped with the hypothetical mechanism cooperate due to their synaptic connections the target behaviour should emerge in a self-organised manner. This allows us to compare the behaviour the neural network model produces with the behaviour observed in experiments.

As the training procedure proposed for IC units is more promising we concentrate on applications of networks consisting of IC units for the moment (Chapter 3 and 4). Different possibilities of application are possible. Within this work, the focus was on storing patterns having been known in advance. But it is also possible to use this type of networks for prediction purposes. The ability to predict the output of the next time step is touched when learning the temporal course of dynamical systems (Chapter 4). To reproduce the dynamics of, for example, a pendulum, the network has to be able to predict the respectively next position and velocity value.

These IC networks can also be applied in another context of prediction, namely to learn classical conditioning tasks (Wittmann, 2005). During training the network is presented with an incomplete input vector only. After training the network is able to predict the respective response to a certain stimulus situation. In contrast to the approach described in this work for learning classical conditioning tasks the $\delta$-error is rectified before using it for training the weights. Consequently, on the one hand no negative weights occur during training and on the other hand weight values cannot decrease again. Therefore, Wittmann proposed to normalise the weights in way that the sum of all weights does not

exceed the value of one. Using this normalisation the weights remain flexible even if the situation changes.

The neurons and neural networks used here can be regarded to be a model of the so-called mirror-neurons (for a review see Rizzolatti and Craighero, 2004). These neurons can neither be attributed to represent only sensory aspects nor to represent only motor aspects. Various studies have shown that they are active during both perception and action. The models presented here are also suitable for both and a separation between sensory and motor units in these 'holistic' networks is hardly possible (see also Chapter 4). Therefore, the same type of networks and the same neurons can be used for perception and the control of action (Cruse, 2003).

## 5.3 Future work

Figure 5.1 allows clarifying what still has to be done in future work to broaden the capabilities of the models presented here. Up to now we have focussed on modelling internal representations as target behaviour. Of course, brains have far more capabilities originating from the interplay of their building blocks – the neurons. To illustrate that here still some work has to be done, the part *Target Behaviour* in Figure 5.1 is marked by a box shaded in lighter grey. Thus, other applications can be thought of which in turn has implications on the model itself. Consequently, the parts *Simulation* and *Simulated Behaviour* in Figure 5.1 are also depicted by a box shaded in lighter grey.

### 5.3.1 Other applications

The IC model as proposed in Chapter 3 can account for findings according to the expression of immediate early genes (IEGs) (Huchzermeyer et al., 2005). In the absence of sensory stimuli only a small amount IEGs is expressed within the brain. If neurons are activated, also expression of IEGs increases (Sheng and Greenberg, 1990). These early genes and their proteins like ZENK are thought to play a role in fast learning processes; they are supposed to mediate between synaptic activation and the activation of late response genes. Huchzermeyer and colleagues (2005) have come up with an astonishing result when studying sexual imprinting in young zebra finches: They found a negative correlation between the preference score and the IEG activation; the bigger

the difference between the stored representation of the sexual partner and the female presented in the experiment was the stronger was the ZENK signal. This finding was a bit counterintuitive because previously the idea was prevalent that a brain area coding a stimulus learnt before should be the more activated the better the actual stimulus matches the template. This idea was derived from hierarchical models of stimulus processing as they have been proposed for the visual cortex, for example, by Hubel and Wiesel (1962).

But the dynamics of the ZENK signal are analogue to the $\delta$-error in the model presented in Chapter 3. The more the external input deviates from the current output of a neuron the bigger the $\delta$-error. Up to now, learning is stopped in the model more or less arbitrarily after the overall error has fallen below a given threshold. By doing so, further plasticity is not allowed. But one could also think of a mechanism which decreases the learning rate over the course of time. If the learning rate is very small or zero the weights do not change any longer. This process of decreasing the learning rate can resemble the decrease of synaptic plasticity over time found in experiments (Gan et al., 2003).

If now a new stimulus is presented to the model, which was not learnt before, the $\delta$-error will be large, but would not have any effect on the weights. Therefore, the $\delta$-error could resemble the activity of the ZENK signal. This activity could express the fact that brains principally always have the disposition to learn but that, due to the decrease of plasticity in the course of time, learning does not take place with the same amount than earlier in development.

The next step to be done here is to compare the real data found in the experiments with the simulated results. Could it be possible to adapt the model in a way that the real data could be predicted?

When trying to model and understand more complex behaviour like processing nested sentences or controlling six-legged walking in uneven terrain certainly changes in the simulation, i.e. in the model are necessary. Here, four main considerations are focused on: the capability of dealing with nonlinearities, training classical MMC networks, scaling the network's size, and ordering and connecting individual internal models.

### 5.3.2 Nonlinearities

The problems tackled so far are linear or only mildly nonlinear due to the rectifiers used in the IC units. Especially in the case of the seemingly more powerful IC models it is an outstanding question if it is possible to introduce nonlinearities as for example nonlinear activation functions of the single neurons and still be able to train the networks. Three different possibilities of activation functions are expedient here: the neurons can be equipped with rectifiers, functions without saturation like the square root function or functions with saturation like squashing functions.

### 5.3.3 Scaling the networks

Being able to scale the properties of a network with its size is crucially important for a model in order to serve as a biologically plausible brain model. The architecture of many models has to be additionally constrained to scale it by, for example, restricting the connectivity to local neighbourhoods only (Sejnowski et al., 1988).

The model described in Chapter 2 consists of MSBE networks because it provides *M*ultiple *S*olutions for the *B*asic *E*quation $\sum_{i=1}^{n} v_i \cdot x_i = 0$. In former publications this has been called MMC network (Kühn and Cruse, 2005; 2005a but is now distinguished from what has been called MMC network in earlier papers (Steinkühler and Cruse, 1998; Cruse, Steinkühler, et al.; 1998). The learning rule described in Chapter 2 suffers from the problem of scaling. If representations should be build up consisting of more items, the model's capabilities as such are soon at their limits. But this restriction of only being able to process a limited number of items could also be found in real brains too: Humans working memory has only a limited capacity (Baddeley, 1986). A solution to cope with this scaling problem is proposed below (Chapter 5.3.5) by combining more small subnetworks each of which contains a limited amount of information.

The second model type we used, the building blocks of which are IC Units (Chapter 3 and 4), does not suffer from scaling problems as long as the learning rate $\varepsilon$ is chosen small enough according to $0 < \varepsilon < \frac{2}{\mathbf{a}^{\mathrm{T}} \cdot \mathbf{a}}$ (see Appendix in Chapter 3.6). Thus, the more units the network has, the smaller the learning rate has to be in order to obtain stable solutions. Therefore, this IC model seems to be more promising for further applications

than the model described in Chapter 2 and it principally should be possible to train more realistic networks consisting of a large number of neurons.

### 5.3.4 Training classical MMC networks

Additionally, this could provide a solution for the still unsolved problem of training classical MMC networks as used for control of, for example, arm movements (Steinkühler and Cruse, 1998; Cruse, Steinkühler, et al.; 1998). The position of each vector necessary for calculating the endpoint of the hand symbolised by R is calculated from four Basic Equations, respectively. This is shown in Figure 5.2: Each vector is obtained by calculating the mean $\bar{x}$ of the four composing equations. That means each of these subsystems is an MSBE network. If a way was found to train these subnetworks effectively, for example by combining the ideas of training IC networks with these MSBE networks, it perhaps would be possible to merge the results to obtain a solution for the entire MMC network. Finding ways to train these MMC networks would dramatically improve their adaptability.

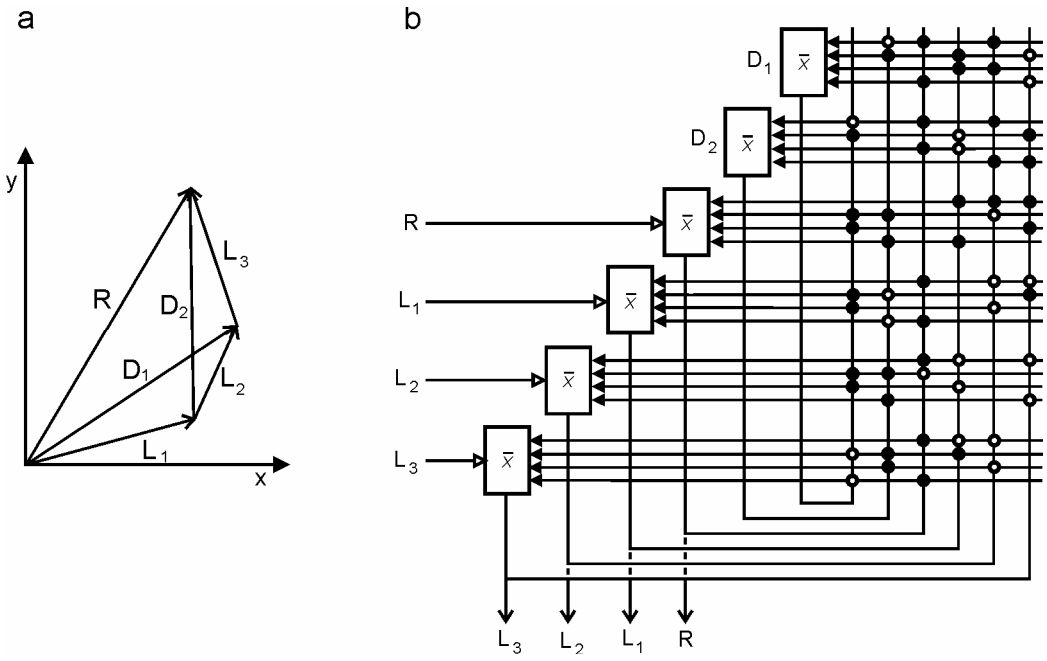


**Figure 5.2: MMC network for a three-joint planar arm. (a) Schematic drawing of a three-segmented arm. (b) MMC network: Each vector is obtained from calculating the mean $\bar{x}$ from four basic equations. The weights are symbolised by closed circles (1) and open circles (-1) (adapted from Cruse et al., 1998).**

In this context another problem is touched: the problem of how to deal with hidden units and how to train networks that are equipped with hidden layers. In the classical MMC networks the units representing the diagonal vectors D1 and D2 (see Figure 5.2) are such hidden units because they do not receive direct external input but only the recurrent input from the other units within the network. If we equipped the single units within the MMC networks with the IC structure for providing better training possibilities for the single elements of the entire network and if we were able to combine them in the end, we probably could kill two birds with one stone: training classical MMC networks as well as training networks with hidden units.

### 5.3.5 Connecting individual internal models

The brain is, of course, able to process and cope with a larger amount of information at the same time; thus, many internal models of external situations coexist. Storing many different individual models raises the question of how these distinct models may be learnt, stored, and retrieved. To solve these problems hints from evolutionary biology might help. Individual models could be equipped with a kind of fitness value as proposed by Steels (1999). This fitness value may depend on spontaneous changes and on successful application in the world. Based on this fitness value different internal models could compete via winner takes all connections.

As a next step, it is indispensable to think of possibilities of how models containing limited information can be combined to build larger frameworks. It is, for example, no problem for us to follow and understand sentences which are long and complicated because of many embedded subordinate clauses. This capability is called recursion and is assumed to be a decisive feature of human intelligence (Hauser et al., 2002; Premack, 2004).

On the one hand, experiments indicate that there appears to be some kind of hierarchical structure: bottom-up attentional mechanisms are much faster than top-down mechanisms implementing our long-term cognitive strategies (Connor et al., 2004). On the other hand, the structure seems to be variable and depends on the context in a dynamical way. Thus, to account for cognitive abilities as, for example, the power of recursion we have to find a way of how combining many small models into hierarchical structures the hierarchy of which depends on the actual context (Chapter 4). This claim is necessary because simple tree-like hierarchies are often not sufficient. For example, a

bat may be considered to belong to the category of flying animals, together with (many, but not all) birds, or it could be regarded as a mammal; but birds do not belong to this category. Thus, it must be possible to adapt the hierarchy dynamically to the actual context. Furthermore, the ability to change the hierarchical order is a prerequisite for the ability to adopt the viewpoint of another person, which, according to Tomasello (2000) develops in human infants at an early age.

Thus, the work presented here provides many interesting opportunities to be investigated in subsequent studies.

## 5.4 References

Baddeley A (1986) *Working memory*. Oxford: Clarendon Press.

Barto AG (1991) Learning and incremental dynamic programming. *Behavioral and Brain Sciences* 14: 94-95.

Bower JM (1992) Modelling the nervous system. *Trends in Neurosciences* 15: 411-412.

Cartwright N (1983) *How the laws of physics lie*. Oxford: Clarendon Press.

Churchland PS, Sejnowski TJ (1988) Perspectives on Cognitive Neuroscience. *Science* 242: 741-745.

Connor CE, Egeth HE and Yantis, S (2004) Visual attention: bottom-up versus top-down. *Current Biology* 14: R850-R852.

Croon MA, van de Vijver FJR (1994) *Viability of mathematical models in the social and behavioural science*. Swets and Zeitlinger.

Cruse H, Steinkühler U and Burkamp C (1998) MMC - a recurrent neural network which can be used as manipulable body model. In *Proceedings of the fifth International Conference on Simulation of Adaptive Behavior,* Cambridge, MA: MIT Press, pp. 381-389.

Elman JL (1990) Finding Structure in Time. *Cognitive Science* 14: 179-211.

Gan W-B, Kwon E, Feng G, sanes JR and Lichtman JW (2003) Synaptic dynamism measured over minutes to month: age-dependent decline in an autonomic ganglion. Nature Neuroscience 6: 956-960.

Giere RN (1997) *Understanding Scientific Reasoning*. Orlando: Harcourt Brace.

Hauser MD, Chomsky N, Fitch T (2002) The faculty of language: What is it, who has it, and how did it evolve? *Science* 298: 1569-1579.

Hubel D, Wiesel T (1962) Receptive fields, binocular interaction and functional architechture in the cat's visual cortex. *Journal of Physiology (London)* 160: 106-154.

Huchzermeyer C, Husemann P, Lieshoff C, Bischof H-J (2005) ZENK expression in a restricted forbrain area correlates negatively with preference score for an imprinted stimulus. submitted.

Kühn S, Cruse H (2005) Mental representation and cognitive behaviour – a recurrent neural network approach. In: *Modeling Language, Cognition and Action: Proceedings of the 9th Neural Computation and Psychology Workshop* (Cangelosi A, Bugmann G, Borisyuk R, eds), Singapore: World Scientific, pp 183-192.

Kühn S, Cruse H (2005a) Static mental representations in recurrent neural networks for the control of dynamic behavioural sequences. *Connection Science* 17: 343-360.

Leatherdale WH (1974) *The role of analogy, model and metaphor in science*. Amsterdam: North-Holland Publ. Co. Elsevier.

Premack D (2004) Is language the key to human intelligence? *Science* 303: 318-320.

Rizzolatti G, Craighero L (2004) The mirror-neuron system. *Annual Review of Neuroscience* 27: 169-192.

Sejnowski TJ, Koch C, Churchland PS (1988) Computational Neuroscience. *Science* 241: 1299-1306.

Sheng M, Greenberg ME (1990) The regulation and function of c-fos and other immediate early genes in the nervous system. *Neuron* 4, 477-485.

Steinkühler U, Cruse H (1998) A holistic model for an internal representation to control movement of a manipulator with redundant degrees of freedom. *Biological Cybernetics* 79: 457-466.

Wartofsky MW (1979) *Models: representation and the scientific understanding*. Dordrecht: Reidel.

Webb B (2001) Can robots make good models of biological behaviour*? Behavioral and Brain Sciences* 24: 1033-1050.

Wittmann J (2005) *Simulation von Situationsmodellen mit Hilfe künstlicher neuronaler Netze zur Überprüfung einer neuen Interpretation klassischer Lernparadigmen*. Diplomarbeit Bielefeld.