# Vertical Vector Fields and Neural Networks: An Application in Atmospheric Pollution Forecasting

David William Pearson[1] Mireille Batton-Hubert [2] and Gérard Dray[3]

[1]EURISE (Roanne Research Group) Jean Monnet University of Saint-Etienne, I.U.T. de Roanne, 20, avenue de Paris, 42334 Roanne, France. email - david.pearson@univ-st-etienne.fr [2]SITE, Ecole des Mines de Saint-Etienne, cours Fauriel, Saint-Etienne, France. email - batton@ensmse.fr [3]LGI2P, EMA Site-EERIE, Parc Scientifique G. Besse, 30319 Nimes, France. email - gerard.dray@site-eerie.ema.fr

**Abstract**

In this paper we look at the role that vertical fields can play in enhancing the performance of a feedforward neural network. Vertical fields help us to determine zones in the input space that are mapped onto the same output, they act in a similar way to kernels of linear mappings but in a nonlinear setting. In the paper we illustrate our ideas using data from a real application, namely forecasting atmospheric pollution for the town of Saint-Etienne in France.

## 1 Introduction

We have been investigating the problem of forecasting atmospheric pollution for some time [1, 2, 3]. We have worked on data from two towns in France: Montpellier and Saint-Etienne. Although the local conditions are not the same for the two towns and pollution levels are different, the problems facing potential forecasters are the same for these and indeed most towns.

In general we have a large database available, data are collected for many different variables about every 10 minutes or so 24 hours a day throughout the year. We need to plunge into the databases in order to find relationships between variables. These relationships are usually expressed in the form of linear or nonlinear regression formulae, neural networks, fuzzy inference systems, or indeed many other types of mathematical tools. One then uses the relationship in order to forecast pollution levels. We are particularly interested in ozone levels.

One thing which intrigues us is that there are instances where "similar" days produce the same levels of ozone and "dissimilar days" produce the same levels of ozone. Similar and dissimilar being defined by Euclidian distances between data vectors containing meteorological data, actual levels of pollution etc.

For this reason, we looked toward vertical vector fields to see whether they could enlighten us. This article illustrates some of our preliminary trials.

## 2 Vertical vector fields

Given a mapping $\pi : X \to Y$, where the topological spaces $X$ and $Y$ have dimensions $n$ and $m$ respectively with $n > m$, and where we assume that the mapping is at least $C^1$. We denote the differential of this mapping by $\pi_\star$ (the matrix of first partial derivatives) then a vector field $v : X \to X$ is said to be a vertical vector field if

$$\pi_\star v(x) = 0 \tag{1}$$

where the 0 on the right hand side of (1) represents a vector of zeros of dimension $m$. In this article we will be taking the input space to an open subset of $\mathbb{R}^n$ and we will assume that $\pi_\star$ has full rank (i.e. $m$) on this subset. That being the case, there will be $n - m$ independent vertical fields satisfying (1).

The interesting thing about vertical vector fields is that if $\pi(x_0) = y_0$ then

$$\pi(\exp(tv)x_0) = y_0 \text{ for } t > 0 \tag{2}$$

where $\exp(tv)x_0$ denotes the operation of exponentiating the vector field, i.e. solving the differential equation

$$\dot{x} = v(x) \text{ with } x(0) = x_0$$

where $\dot{x} := \frac{dx}{dt}$.

What is more, due to the fact that any trajectory defined by the $n - m$ vertical vector fields is mapped by $\pi$ onto the same value, we can make use of the following construction. Let the mapping $\phi : S \to X$, where $S$ is an open neighbourhood of the origin in $\mathbb{R}^{n-m}$, be defined as follows

$$\phi(s) = \exp(s^{n-m}v_{n-m}) \cdots \exp(s^1 v_1)x_0 \tag{3}$$

where the $s^i$ are the elements of a vector $s \in S$ and $x_0$ is such that $\pi(x_0) = y_0$. The concatenation in (3) means that we start at the point $x_0$ and go in the direction of the vector field $v_1$ for $s^1$ time units, then we go in the direction of vector field $v_2$ for $s^2$ time units etc. The

D. W. Pearson et al. (eds.), *Artificial Neural Nets and Genetic Algorithms*

result is that $\pi(\phi(s)) = y_0$. As $s$ goes through all of $S$ then the trajectories form a hyper-surface in $X$ and this hyper-surface is usually called the leaf above $y_0$ and denoted $L_{y_0}$.

In our case, the mapping $\pi$ is a standard feedforward neural network with analytic activation functions. We assume that the neural network has been trained on some subset of the available training data, taken to be a collection of data pairs $(x_i, y_i)$, $i = 1, \ldots, N$. In particular, we assume that the subset used for training is of very low cardinality when compared to all the data available and that the network has been trained as a sort of classifier. So, for example, if we initially train the network to identify two classes then $\pi(x) = y_1$ if $x \in \mathcal{X}_\infty$ and $\pi(x) = y_2$ if $x \in \mathcal{X}_\in$ where $\mathcal{X}_\infty$ and $\mathcal{X}_\in$ are two disjoint subsets of $X$.

Once the vertical vector fields have been determined for $\pi(x_0) = y_0$ we can determine whether or not any of the other data points $x_i \in L_{y_0}$, if such is the case then $\pi(x_i) = y_0$ of course and so $x_i$ need not be included in the training data.

We believe that determining whether or not data points lie in $L_{y_0}$ (for various $y_0$) will give us a better picture of the structure of the data. For example, we could retrain the neural network using data selected on the basis of whether or not they are related by vertical vector fields. If two points belonged theoretically to the same class but not to the same $L_{y_0}$, then we could consider including them in the training data set but with them mapping to very slightly different output values. In this way, the two data points should belong to two leaves in vertical spaces very close together but not coincident. The separation or nearness of leaves could be used to increase robustness properties of the neural network.

## 3 Estimating the vertical vector fields

We need to find a set of vector fields satisfying (1). This is quite a task because these vector fields could be nonlinear, this is indeed the case with feedforward networks. Not only that, it would be extremely difficult to find a set of vector fields which satisfy the condition everywhere, i.e. $\forall x \in X$. We are therefore confined to making approximations and assumptions about the vector fields.

- The first, and probably the strongest, assumption that we make is that the vertical vector fields are linear so that in (1) $v(x) = Ax$ for some $n \times n$ matrix $A$.
- The principal approximation that we use is to satisfy (1) at a single point $x_0$, by continuity this will mean that the condition is satisfied in a neighbourhood of $x_0$.
- The second assumption that is made about the vector fields is that they form an involutive Lie algebra at the point $x_0$ and thus in a neighbourhood of $x_0$.

To begin the calculation we select a point $x_0$ and evaluate $\pi_\star(x_0)$ by using a second order central difference approximation for the partial derivatives. We then determine a basis for the kernel of $\pi_\star(x_0)$ by using the singular value decomposition [4], that we denote by $\mathrm{Ker}(\pi_\star(x_0))=\{v_1, \ldots v_{n-m}\}$. Making use of the above listed approximation and to satisfy the first of the assumptions we need to find matrices $A_1, \ldots A_{n-m}$ such that

$$A_k x_0 = v_k \text{ for } k = 1, \ldots, n-m \tag{4}$$

Then, for the second of the above assumptions the following conditions have to be satisfied [5]

$$\begin{gathered}[A_i, A_j](x_0) = \sum_{k=1}^{n-m} c_{ij}^k A_k x_0 \\ \text{for } i, j = 1, \ldots n-m \text{ and } i \neq j\end{gathered} \tag{5}$$

where $[A_i, A_j] = A_i A_j - A_j A_i$ is the Lie bracket and the $c_{ij}^k$ are scalars.

In fact, we employ a numerical optimisation method to calculate the elements of the matrices $A_i$ and the scalars $c_{ij}^k$ in order to ensure that the following condition is satisfied.

$$\begin{gathered}\frac{d\pi(x(t))^\nu}{dt^\nu_{|x=x_0}} = 0 \\ \text{for } \nu = 0, 1, 2, \ldots\end{gathered}$$

The complete method is explained in [6].

Having calculated the vertical vector fields, we can then evaluate $\phi$ as in (3) for a fixed $s$.