14
Neural Network Classification and Prior Class Probabilities

Steve Lawrence[1], Ian Burns[2], Andrew Back[3], Ah Chung Tsoi[4], and C. Lee Giles[1⋆]

[1] NEC Research Institute⋆⋆, 4 Independence Way, Princeton, NJ 08540
[2]Open Access Pty Ltd, Level 2, 7–9 Albany St, St. Leonards, NSW 2065, Australia
[3] RIKEN Brain Science Institute, 2-1 Hirosawa, Wako-shi, Saitama, 351-0198, Japan
[4] Faculty of Informatics, University of Wollongong, Northfields Ave, Wollongong, NSW 2522, Australia
{lawrence,giles}@research.nj.nec.com, ian.burns@oa.com.au, back@brain.riken.go.jp, Ah_Chung_Tsoi@uow.edu.au
http://www.neci.nj.nec.com/homepages/lawrence/

Abstract. A commonly encountered problem in MLP (multi-layer perceptron) classification problems is related to the prior probabilities of the individual classes – if the number of training examples that correspond to each class varies significantly between the classes, then it may be harder for the network to learn the rarer classes in some cases. Such practical experience does not match theoretical results which show that MLPs approximate Bayesian *a posteriori* probabilities (independent of the prior class probabilities). Our investigation of the problem shows that the difference between the theoretical and practical results lies with the assumptions made in the theory (accurate estimation of Bayesian *a posteriori* probabilities requires the network to be large enough, training to converge to a global minimum, infinite training data, and the *a priori* class probabilities of the test set to be correctly represented in the training set). Specifically, the problem can often be traced to the fact that efficient MLP training mechanisms lead to sub-optimal solutions for most practical problems. In this chapter, we demonstrate the problem, discuss possible methods for alleviating it, and introduce new heuristics which are shown to perform well on a sample ECG classification problem. The heuristics may also be used as a simple means of adjusting for unequal misclassification costs.

14.1 Introduction

It has been shown theoretically that MLPs approximate Bayesian *a posteriori* probabilities when the desired network outputs are *1 of M* and squared-error

⋆ Lee Giles is also with the Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742.

⋆⋆ `http://www.neci.nj.nec.com`

G.B. Orr, K.–R. Müller (Eds.): Neural Networks: Tricks of the Trade, LNCS 1524, pp. 299–313, 1998.

or cross-entropy cost functions are used [6, 11, 12, 15, 23, 25, 26, 28, 29, 32]. This result relies on a number of assumptions for accurate estimation: the network must be large enough and training must find a global minimum, infinite training data is required, and the *a priori* class probabilities of the test set must be correctly represented in the training set.

In practice, MLPs have also been shown to accurately estimate Bayesian *a posteriori* probabilities for certain experiments [10]. However, a commonly encountered problem in MLP classification is related to the case when the frequency of the classes in the training set varies significantly[1]. If the number of training examples for each class varies significantly between classes then there may be a bias towards predicting the more common classes [3, 4], leading to worse classification performance for the rarer classes. In [5] it was observed that classes with low *a priori* probability in a speech application were "ignored" (no samples were classified as these classes after training). Such problems indicate that either the estimation of Bayesian *a posteriori* probabilities is inaccurate, or that such estimation may not be desired (e.g. due to varying misclassification costs (this is explained further in section 14.4)). Bourlard and Morgan [7] have demonstrated inaccurate estimation of Bayesian *a posteriori* probabilities in speech recognition. This chapter discusses how the problem may occur along with methods of dealing with the problem.

14.2 The Trick

This section describes the tricks for alleviating the aforementioned problem. Motivation for their use and experimental results are provided in the following sections. The methods all consider some kind of scaling which is performed on a class by class basis[2].

14.2.1 Prior Scaling

A method of scaling weight updates on a class by class basis according to the prior class probabilities is proposed in this section. Consider gradient descent weight updates for each pattern: $w^l_{ki}(\text{new}) = w^l_{ki}(\text{old}) + \Delta w^l_{ki}(p)$ where $\Delta w^l_{ki}(p) = -\eta \frac{\partial E(p)}{\partial w^l_{ki}}$, p is the pattern index, and w_{ki} is the weight between neuron k in layer l and neuron i in layer $l-1$. Scaling the weight updates on a pattern by pattern basis is considered such that the total expected update for patterns belonging to each class is equal (i.e. independent of the number of patterns in

[1] For the data in general. Others have considered the case of different class probabilities between the training and test sets, e.g. [23].

[2] Anand et al. [2] have also presented an algorithm related to unequal prior class probabilities. However, their algorithm aims only to improve convergence speed. Additionally, their algorithm is only for two class problems and batch update.