An Investigation of the Generalisation Performance of Neural Networks Applied to Lofargram Classification

Gary D. Kendall¹, Trevor J. Hall¹ and Timothy J. Newton²

¹Department of Physics, King's College London, Strand, London WC2R 2LS, UK,
²Smith System Engineering Ltd., Surrey Research Park, Guildford, Surrey GU2 5YP, UK.

There exists a substantial problem in obtaining good generalisation performance in the application of artificial neural network technology where training data is limited. Generalisation ability is analysed for a number of computational paradigms which attempt to alleviate this for the multilayer perceptron. The problem of line detection in a time/frequency sonar image or 'lofargram' is adopted as a case study on which to assess these techniques. The effect on neural network generalisation performance is studied for (a) heuristically changing the number of hidden nodes, (b) weight decay, (c) soft weight-sharing, and (d) Ockham's networks. These techniques are introduced from the perspective of the Minimum Description Length principle. Results show that the use of weight decay and Ockham's networks are able to improve generalisation beyond that available by simply altering the number of hidden nodes. It is shown that line detection in lofargram images is possible at a success rate of 85% for data outside of the training set.

Keywords: Neural network; Non-linear modelling; Generalisation; Minimum Description Length; Ockham's Razor; Weight decay; Soft weight-sharing; Ockham's network; Sonar; Lofargram

1. Introduction

1.1. General

Artificial neural networks are a class of non-linear models which have received significant attention in the literature in recent years. They are typically characterised by a dense interconnection of non-linear summation and thresholding devices, commonly arranged in layers, with designations for input and output connections. Although much work has been conducted on the hardware implementation of artificial neural networks, most research has been performed on computational models on conventional computers.

During the learning phase of an artificial neural network it is normal to minimise the output error over a set of example input/output pairs using a gradient based optimisation technique. This ensures good modelling of the training set given a sufficiently large neural network if no local minima are encountered during optimisation. However, given a trained neural network, it is not possible to guarantee a particular level of performance on unseen input/output pairs, i.e. generalisation performance. It is in fact common for the generalisation performance of a neural network to become sub-optimal if training is allowed to continue indefinitely as the model overfits the training data. One approach to this problem is to restrict the size or interconnection density of the neural network, thus restricting performance on the training set.

1.2. Objectives

It is the main objective of this study to consider the generalisation problem associated with neural
networks as a case study with respect to a particular application. The application chosen is the detection of lines in a lofargram* given a set of labelled data. It is studied by the implementation of four of the available techniques to restrict neural network complexity during training.

Section 2 presents a broad overview of the basic concepts involved in the application of artificial neural networks. This includes a discussion of techniques in the recent literature specifically designed to improve generalisation by reducing network complexity. Section 3 presents the case study of the application of neural networks to the problem of lofargram classification. This includes experimental results for techniques discussed in the previous section as well as other issues raised during the assessment of techniques. Section 4 is an analysis of both the application studied as well as the techniques applied to it during the case study.

2. Neural Network Generalisation

2.1. Multilayer Perceptron

One of the most common neural network models in current use is the multilayer perceptron. In the basic multilayer perceptron the thresholding summation elements (nodes) are arranged in layers with full interconnection. The inputs of each node in one layer are connected to the outputs of all nodes in the previous layer, with the first and last layers in the network corresponding to inputs and outputs respectively. The network is thus described as having a feed forward architecture. This is shown schematically in Figure 1 for a multilayer perceptron with a single hidden layer.

The multilayer perceptron with a single hidden layer has \( n_0 \) inputs, \( n_1 \) hidden nodes and \( n_2 \) outputs. Each weight \( w_{ijk} \) connects output \( k \) in layer \( i-1 \) to input \( j \) in layer \( i \). The output of any neuron can thus be defined as:

\[
\sigma_j = \Theta \left( w_{ij0} + \sum_{k=1}^{n_1} \sigma_{i-1k} w_{ijk} \right) \tag{1}
\]

where \( \Theta(x) \) is the thresholding function and \( w_{ij0} \) is a biasing connection. It is useful to define a bounded monotonically increasing thresholding function such that \( \Theta'(x) \) exists. A function in common use is a sigmoidal function of the form:

\[
\Theta(x) = \frac{1}{1 + e^{-x}} \tag{2}
\]

Adopting an appropriate cost function for the neural network is straightforward to optimise the \( w_{ijk} \) using a gradient-based technique. One of the most commonly used techniques in use for this is simple gradient descent with momentum. This is often referred to as back propagation [1], though this actually refers to the technique used to calculate the derivative of the cost with respect to the weights. Optimisation by steepest descent, even with momentum terms, is inefficient. Some attempts have been made to include an adaptive step size in the gradient descent algorithm to improve performance. However, a simple improvement to learning performance for the multilayer perceptron is to use a conjugate gradient algorithm rather than steepest descent. This is becoming increasingly common and typically reduces training times by an order of magnitude [2].

2.2. Importance of Network Size

Overtraining and Generalisation. One of the most significant difficulties encountered in the application of neural networks to real world data is the problem of generalisation, i.e. to perform the optimum modelling of a set of training data that will provide the best possible performance on unseen data. This problem arises because the neural network is a general model and is therefore unable to provide a perfect mapping of the underlying statistical process from a set of incomplete training data. It is particularly problematic when training data is noisy and the neural network has too many weights, as it is common for an overtrained neural network to classify outliers in the training set correctly.

The most commonly used technique to avoid overtraining is to monitor generalisation perform-