Ideas about a Regularized MLP Classifier by Means of Weight Decay Stepping

Paavo Nieminen and Tommi Kärkkäinen

Department of Mathematical Information Technology,
University of Jyväskylä, Finland
{nieminen,tka}@jyu.fi
http://www.mit.jyu.fi/
http://users.jyu.fi/~nieminen/

Abstract. The generalization capability of a multilayer perceptron can be adjusted by adding a penalty (weight decay) term to the cost function used in the training process. In this paper we present a possible heuristic method for finding a good coefficient for this regularization term while, at the same time, looking for a well-regularized MLP model. The simple heuristic is based on validation error, but not strictly in the sense of early stopping; instead, we compare different coefficients using a subdivision of the training data for quality evaluation, and in this way we try to find a coefficient that yields good generalization even after a training run that ends up in full convergence to a cost minimum, given a certain accuracy goal. At the time of writing, we are still working on benchmarking and improving the heuristic, published here for the first time.

Keywords: classification, neural networks, MLP, regularization, heuristic.

1 Introduction

This paper deals with the task of pattern classification. We begin by going through the necessary definitions and equations for those who want to implement a similar system. Later on, we shall turn to less formalized, heuristic ideas.

Let $\{\mathbf{x}_i\}_{i=1}^N, \mathbf{x}_i \in \mathbb{R}^D$ be a set of N vectors of dimension D. Each vector $\mathbf{x}_i$ is known to belong to a class $c_i \in \{1, \ldots, C\}$, coded as a binary vector $\mathbf{y}_i \in \mathbb{R}^C$ such that the components of the i:th vector are $(\mathbf{y}_i)_k = 1$ for $k = c_i$ and $(\mathbf{y}_i)_k = -1$ for $k \neq c_i$. Using the known pairs $(\mathbf{x}_i, \mathbf{y}_i)$ as examples, we aim to train a computer system to associate vectors with a corresponding class. For our current purposes, any vector $\mathbf{y} \in \mathbb{R}^C$ is converted to an exact class representative by choosing c equal to smallest k for which $(\mathbf{y})_k \geq (\mathbf{y})_j$ for all $j \in \{1, \ldots, C\}$, i.e., selection of the index of the largest component, or, in the case of equality, the one with the smallest index. This conversion works fine for vectors that are already approximately close to the encoded class prototype.

We especially want the machine to be able to generalize what it has learned by example, and apply the knowledge to new vectors $\mathbf{x}_q \notin \{\mathbf{x}_i\}_{i=1}^N$ for which nobody

M. Kolehmainen et al. (Eds.): ICANNGA 2009, LNCS 5495, pp. 32–41, 2009.

knows the proper class before the machine makes its advisory guess. In this way, we can automatically identify images, sounds, industrial measurements, and other useful things that can be represented as vectors. Artificial neural networks (ANN) are a widely used system for such automatic classification tasks. For more knowledge about ANNs, we refer to the textbook [1].

Of special interest in this study is the multilayer perceptron (MLP), a feed-forward ANN with sigmoidal activation. Such an ANN can be easily (and efficiently) implemented directly from the matrix representation described in prior works [2,3,4]. For any $\mathbf{x} \in \mathbb{R}^D$ we set

$$\mathbf{o}^0 = \mathbf{x}, \quad \mathbf{o}^l = \mathcal{F}^l(\mathbf{W}^l \hat{\mathbf{o}}^{(l-1)}) \quad \text{for} \quad l = 1, \ldots, L\,. \tag{1}$$

By the notation $\mathbf{o}^0$ we mean the vector on the "zeroth" layer, which is considered to be the input vector $\mathbf{x}$ presented to the network. Iterative computation yields the output vectors $\mathbf{o}^l$ for all the other L layers of the network, numbered $l = 1, \ldots, L$. The values depend on the selection of layer-wise neural weight matrices $\mathbf{W}^l$. The hat notation $\hat{\mathbf{o}}^{(l-1)}$ means that the output vector of the previous layer $\mathbf{o}^{(l-1)}$ is extended by an initial coordinate of value one, facilitating the bias mechanism so that the first column of $\mathbf{W}^l$ contains the biases of neurons on layer l. Finally, the notation $\mathcal{F}^l(\cdot)$ denotes the application of a function matrix to a vector. For this study, the function matrices are diagonal, and they consist solely of the traditional hyperbolic tangent activation function on hidden layers and the identity mapping on the output layer. The reasoning behind this choice is presented in [3]. The final output of the MLP resides in the output vector $\mathbf{o}^L$. We use the notation $\mathcal{N}(\{\mathbf{W}^l\})(\mathbf{x}) = \mathbf{o}^L$ to denote the output of the above iterative computation.

A model such as (1) is to be trained somehow. A usual way is to minimize an error function, or *cost function*, computed over the available training data set. The present work is based on the following cost function formulation:

$$J(\{\mathbf{W}^l\}) = \frac{1}{2N} \sum_{i=1}^{N} \left\| \mathcal{N}(\{\mathbf{W}^l\})(\mathbf{x}_i) - \mathbf{y}_i \right\|^2 + \beta \sum_{l=1}^{L} \sum_{(i,j) \in I_l} \frac{1}{2S_l} |\mathbf{W}^l_{i,j}|^2 \tag{2}$$

for $\beta \geq 0$. Here, the index set I_l contains all other indices of the weight matrices except the ones corresponding to the bias-vector (i.e., first column) of $\mathbf{W}^L$ as suggested by the test results in [3] (see also [5], Chapter 9). S_l is the number of elements in the index set I_l. This averaging divisor was not present in the earlier works [3,4].

If we were to use only the first mean-squared-error term, and find a model near the global optimum of that cost function, it is quite likely that the model would *overfit* individual quirks of the training examples, and not be able to generalize into unforeseen vectors. As seen from the formula, we have decided to make experiments with the well-known method of adding a *regularization term*, with a weight coefficient β called the *regularization parameter* [5]. The choice of the weight decay formulation with a single coefficient is more thoroughly explained and contrasted with early stopping in [3].