ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 7-12

Using a Meta Neural Network for RPROP parameter adaptation

Colin McCormack,
Dept. of Computer Science,
University College Cork,
Cork, Ireland.
E-mail: colin@odyssey.ucc.ie

Abstract. This paper proposes an application independent method of automating learning rule parameter selection using a form of supervisor neural network, known as a Meta Neural Network, to alter the value of a learning rule parameter during training. The Meta Neural Network is trained using data generated by observing the training of a neural network and recording the effects of the selection of various parameter values. Experiments are undertaken to see how this method performs by using it to adapt a global parameter of the RPROP learning rule.

1. Introduction

Despite the development of more efficient learning rules it remains necessary to manually select appropriate learning rule parameter values in order to achieve an acceptable solution. Two of the major problems associated with the selection of suitable parameters are the erratic nature of the quality of the solution (where quality can be defined as the speed of convergence and the accuracy of the resultant network) and the waste of resources used to train an unsatisfactory network. The quality of the solution is heavily dependant on the initial learning rule parameters and training a neural network using general parameter values which are unsuitable for the problem at hand can lead to a waste of computational resources.

The function of this paper is to investigate a method of parameter adaptation which involves the use of a separate neural network (called a Meta Neural Network) to select appropriate values for the η^- parameter of the RPROP learning rule. We look at the results obtained when a standard RPROP rule and a Meta Neural Network are applied to three benchmark problems.

1.1 RPROP

Resilient Backpropagation (RPROP) [1] is a local adaptive learning scheme. In it the size of the derivative is taken to indicate the direction of the weight update. The size of the weight update, $\Delta w_{ij}^{(t)}$, is determined by a weight update value $\Delta_{ij}^{(t)}$, where t is the current epoch and i, j are the nodes adjoining the weight being updated. The weight update value is adjusted as training is carried out.

Currently RPROP appears to be the most effective supervised learning rule [2],[7]. This paper shows that the performance of RPROP can be improved further by adding a supervisor network to make suggestions for parameter values.

The adaptation rules for the RPROP algorithm are:

ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 7-12

$$\Delta w_{ij}^{(t)} = \begin{cases} -\Delta_{ij}^{(t)}, if \frac{\delta E^{(t)}}{\delta w_{ij}} > 0 \\ +\Delta_{ij}^{(t)}, if \frac{\delta E^{(t)}}{\delta w_{ij}} < 0 \\ 0, else \end{cases} \quad \text{where} \quad \Delta_{ij}^{(t)} = \begin{cases} \eta^{+} * \Delta_{ij}^{(t-1)}, if \frac{\delta E^{(t-1)}}{\delta w_{ij}} * \frac{\delta E^{(t)}}{\delta w_{ij}} > 0 \\ \eta^{-} * \Delta_{ij}^{(t-1)}, if \frac{\delta E^{(t-1)}}{\delta w_{ij}} * \frac{\delta E^{(t)}}{\delta w_{ij}} < 0 \\ \Delta_{ij}^{(t-1)}, else \end{cases}$$

Where $\frac{\delta E^{(t)}}{\delta w_{ij}}$ denotes the summed gradient information (slope) over all patterns in the training set and $0 < \eta^{-} < 1 < \eta^{+}$. The value of η^{-} is usually 0.5 and η^{+} is 1.2.

2. Experimental Description

The experiments performed investigate the effectiveness of using a Meta Neural Network to adapt the η^{-} parameter of RPROP. A Meta Neural Network (MNN) is a form of supervisor neural network which makes suggestions to a conventional learning rule on the values of various parameters. The earliest work on MNN's [3] proposed a system which made suggestions for weight and parameter values, previous work in the area of parameter selection used a MNN to adapt the ε parameter for the Quickpropagation [4] learning rule [5]. In this paper three different MNN's are evaluated, one which uses its experience of learning on the same problem domain as the network it is aiding and two which use experience of learning on a different problem domain.

2.1 MNN Methodology

The scheme for creating a MNN to aid a conventional neural network is composed of three stages. In the first stage data for training the MNN is created, in the second stage the MNN is trained and in the third stage the MNN is used to guide a conventional learning rule.

In stage 1 a backtracking system was set up which allowed a learning algorithm to see the results of the selection of the next η^{-} value. Backtracking allows the learning rule to backtrack from a parameter value choice that does not lead to a short term decrease in error value. At each epoch potential η^{-} values are evaluated with the η^{-} value leading to the greatest reduction in error in the training set being retained and the training process continued. The value of η^{-} is limited to six values (in the range 0.3 to 0.8) and at each epoch η^{-} is allowed to increase or decrease by 0.1 or remain at the current value, these three η^{-} values are then evaluated. After the evaluation the results are used to augment a set S with the current network slope, the previous network slope and the action (increment/maintain/decrement) which produced the best value of η^{-} (i.e. the value of η^{-} which caused the largest reduction in error).

In stage 2 the set S is used as a training set for a MNN, where the inputs are: current network slope, previous network slope and the output is a single value which

ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 7-12

indicated whether the value of η^- increased, decreased or remained the same. The learning rule used to train the MNN was RPROP, as described above.

In stage 3 at each epoch the RPROP learning rule passes the value of the current network slope and the previous network slope to the MNN which suggests an increase/decrease or no change in the value of η^-. The upper and lower bounds for η^- are 0.3 and 0.8.

3. Evaluation of the Meta Neural Networks

3.1 Benchmark problem description

Three benchmark problems, the Thyroid problem, the Building problem and the Soybean problem are used to evaluate the effectiveness of MNN's. These problems are also used to produce data for the MNN's which are in turn trained and evaluated. The problems are taken from a comprehensive study of neural network benchmarks [6] and have seen wide use in AI and Neural Network literature. The problems are referred to in [6] as 'Thyroid a', 'Building a' and 'Soybean a'. A description of each problems attributes and the network used with this problem is given in Table 1.

Problem	Nr. of Inputs	Nr. of Outputs	Nr. of Examples	Network Used
Thyroid	21	3	7200	21-7-3
Building	14	3	4208	14-16-3
Soybean	82	19	683	82-32-19

Table 1: Benchmark problem description

3.2 Result Evaluation

The set of available examples is divided into three sets: a training set is used to train the network, a validation set is used to evaluate the quality of the network during training and to measure overfitting, finally a test set is used at the end of training to evaluate the resultant network. In the series of experiments undertaken 50% of the problems total available examples are allocated for the training set, 25% for the validation set and 25% for the test set.

The error measure, E, used was the squared error percentage [6]. This was derived from the normalisation of the mean squared error to reduce its dependence on the number of coefficients in the problem representation and on the range of output values used: $E = 100.\frac{o_{\max} - o_{\min}}{N.P}\sum_{p=1}^{P}\sum_{i=1}^{N}(o_{pi} - tr_{pi})^2$ where o_{min} and o_{max} are the minimum and maximum values of the output coefficients used in the problem, N is the number of output nodes of the network, P is the number of patterns in the data set, o is the network output and tr is the target value.

The Generalisation Loss (GL) at epoch t is defined as the relative increase of the validation error over the minimum so far: $GL(t) = 100.\left(\frac{E_{va}(t)}{E_{opt}(t)} - 1\right)$, where $E_{va}(t)$ is the current validation error and the value $E_{opt}(t)$ is defined as being the lowest validation set error obtained in the epochs up to t: $E_{opt}(t) = \min_{t' \le t} E_{va}(t')$. The GL value represents a measure of the level of overfitting performed by the network [6].

Training progress P [6] is measured after a training strip of length k, which is a sequence of k epochs numbered n+1,...,n+k where n is divisible by k: $P_k(t) = 1000.\left(\frac{\sum_{t' \in t-k+1..t} E_{tr}(t')}{k.\min_{t' \in t-k+1..t} E_{tr}(t')} - 1\right)$, where E_{tr} is the training set error. In the experiments detailed in this paper k=5. The training progress gives the extent of the difference between the average training error in the strip and the minimum training error and is used to determine when the network has reached a point where no further training is effectively taking place. In the experiments performed in this paper training is halted when the progress P drops below 0.1. This is consistent with the approach taken in [6].

4. Experiments

Each MNN is trained using a set S derived from backtracking on a particular problem. The networks trained using these MNN to suggest parameter values are known as 'Thyroid MNN' (i.e. the MNN was trained using results obtained from training on the Thyroid problem), 'Building MNN' and 'Soybean MNN'.

In the experiments ten networks were trained using the standard RPROP method (η^- = 0.5) and RPROP using η^- values that have been determined as optimal from previous experiments, these values are: η^- = 0.3 for the Thyroid problem, 0.5 for the building problem and 0.7 for the Soybean problem. A similar number of networks which used different MNN were trained for comparison. All initial learning rule and network architecture parameters were fixed apart from the initial weight set which was random for each network. The average of the results and the standard deviation is presented in tables 2, 3 and 4.

5. Results

Tables 2, 3 and 4 provide the results for the experiments, each table gives the result for a particular benchmark problem. The columns in the tables contain the average results for the training, validation and test sets at the cessation of training, the standard deviation for the results is included in brackets. The 'Nr. of Epochs' is the number of epochs at the cessation of training. The 'Last GL' column provides the final generalisation loss measurement, this indicates the extent to which the network is overfitting. The rows give the result for 'normal' RPROP (i.e. η^- = 0.5), 'optimal' RPROP (i.e. using optimal values for η^-) and the results for the MNN aided networks.

	Training	Validation	Test	Nr. of Epochs	Last GL
Normal RPROP	1.17 (0.99)	1.59 (0.93)	1.58 (0.90)	7258 (2786)	112 (137)
Optimal RPROP	0.6 (0.22)	1.01 (0.2)	1.12 (0.25)	4344 (2963)	24 (22)
Thyroid MNN	0.25 (0.07)	0.78 (0.13)	0.86 (0.15)	967 (311)	14 (5)
Building MNN	0.18 (0.04)	0.83 (0.08)	0.83 (0.05)	3616 (1624)	22 (11)
Soybean MNN	0.27 (0.08)	0.85 (0.13)	0.81 (0.11)	1305 (641)	20 (12)

Table 2: Results for experiments on the Thyroid problem.

For the Thyroid benchmark (table 2) the best performing network is one which uses a Building MNN to suggest alterations of η^-, the second best is one which uses a Thyroid MNN and third is a network which uses a Soybean MNN. The errors for all three MNN are 30% lower than the optimal RPROP and are 50% lower than the errors obtained by normal RPROP. The standard deviations are also lower for the MNN. Each MNN also takes less epochs on average to reach the end of training and when training is halted the MNN trained networks overfit less.

	Training	Validation	Test	Nr. of Epochs	Last GL
Normal/ Optimal RPROP	0.069(0.006)	0.72 (0.09)	0.53 (0.02)	662 (319)	31.2 (87)
Thyroid MNN	0.069(0.002)	0.68 (0.07)	0.53 (0.02)	897 (212)	2.5 (2.6)
Building MNN	0.070(0.003)	0.66 (0.07)	0.52 (0.02)	877 (278)	4.9 (4.6)
Soybean MNN	0.071(0.003)	0.65 (0.06)	0.52 (0.01)	832 (294)	2.8 (3.2)

Table 3: Results for experiments on the Building problem.

In the table of the results for the Building problem (table 3) the results for the optimal RPROP in are the same as those for normal RPROP since the optimal value of η^- was found to be 0.5. The best result is obtained using a Soybean MNN, the second best uses a Building MNN and the third uses a Thyroid MNN. The MNN errors are on average 10% lower than using a conventional RPROP method.

	Training	Validation	Test	Nr. of Epochs	Last GL
Normal RPROP	0.01 (0.01)	1.09 (0.15)	1.19 (0.17)	3988 (2380)	113 (66)
Optimal RPROP	0.01 (0.01)	1.09 (0.18)	1.21 (0.13)	3666 (1279)	144 (77)
Thyroid MNN	0 (0)	0.39 (0.03)	0.61 (0.06)	3361 (870)	7.7 (10.7)
Building MNN	0 (0)	0.39 (0.04)	0.63 (0.03)	4046 (1415)	16 (9.6)
Soybean MNN	0 (0)	0.41 (0.05)	0.61 (0.05)	3861 (1225)	7.6 (8.6)

Table 4: Results for experiments on the Soybean problem.

For the Soybean problem (table 4) the best result is obtained using a Thyroid MNN, second is a network using Soybean MNN, and third is one using a Building MNN. The errors for all three MNNs are only half those obtained using a conventional RPROP algorithm.

For the three benchmarks used the top three performers are always MNNs even though the RPROP algorithm is one of the best learning rules and the η^- parameter has been tuned for each particular problem. The standard deviation is also lower for all the MNN trained networks indicating that the MNNs are more consistent performers.

6. Conclusion

A Meta Neural Network is a way of acquiring and using information about the learning mechanism of a neural network. In this paper we have seen that a MNN trained using knowledge derived from an unrelated problem domain can outperform

a learning rule which has had the optimal η^- parameter for that problem domain preselected by trial and error. The advantage of a MNN scheme is that a MNN's effectiveness on a problem is independent of the MNN's original training problem and its associated architecture. MNN's can thus be trained once and used successfully on different problem domains. While a MNN may not always significantly outperform a conventional learning method it fares no worse and when it does perform well the benefits are significant. This approach compares favourably with the more usual approach of developing more general learning rules which may not be optimal for every type of problem, developing specialised learning rules which are only suitable for specific types of problem and using ad-hoc methods of tuning parameter values.

This paper shows a successful approach to adaptation of the η^- parameter of the RPROP learning rule and complements previous work [5] which has shown success with the Quickpropagation learning rule. To improve the capabilities of neural networks it may be worthwhile considering the incorporation of parameter adapting MNN's into learning rules so they can be more flexible, more generally applicable and produce a better quality solution.

References

1. Riedmiller, M. and Braun, H. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. Procs of the IEEE International Conference on Neural Networks. 586-591, San Francisco. (1993)
2. Riedmiller, M. Advanced supervised learning in multi-layered perceptrons: From Backpropagation to adaptive learning algorithms. *Computer Standards and Interfaces.* vol 16 part 3. 265-278. (1994)
3. Naik, D.K. Meta-Neural Networks that learn by learning. *IJCNN '90.* I 437-442. (1990)
4. Fahlman, S. Faster-Learning variations on Back-Propagation: An Empirical Study. *Proceedings of the 1988 Connectionist Models Summer School.* Morgan Kaufmann. (1988)
5. McCormack, C. Parameter Adaptation using a Meta Neural Network. Proceedings of the World Conference on Neural Networks, Washington D.C., Vol 3, 147-150. (1995)
6. Prechelt, L. PROBEN1: A set of neural network benchmark problems and benchmarking rules. *Technical Report, Dept. of Informatics, University of Karlsruhe, Germany.* ftp://ftp.ira.uka.de/pub/neuron/proben1.tar.gz. (1994)
7. Schiffmann, W. Optimisation of the Backpropagation Algorithm for training multilayer perceptrons. *Technical report, Dept of Physics, University of Kobelnz, Germany.*ftp://archive.cis.ohiostate.edu/pub/neuroprose/schiff.bp_speedup.ps.Z (1993)