Recent Advances in Neural Network Training Using Constrained Optimization Methods

Stavros J. Perantonis, Vassilis Virvilis and Nikolaos Ampazis

Institute of Informatics and Telecommunications, National Center for Scientific Research "Demokritos", 153 10 Aghia Paraskevi, Greece. Tel: 6510 310. Fax: 6532 175. E-mail: sper@iit.nrcps.ariadne-t.gr.

I. Introduction

Methods from the field of Optimization have played an important role in developing training algorithms for connectionist systems. Indeed, the realization that simple gradient descent (back propagation - BP) can be applied with success to the training of multilayered feedforward networks (MFNs) [1] was responsible to a great extent for the resurgence of interest in this kind of networks during the mid 1980s. Most of the methods used for supervised learning originate from unconstrained optimization techniques. Obviously, this is related to the "black box" nature of connectionist systems: Apart from the minimization of a cost function, no other information or knowledge is usually taken into account.

Nevertheless, recent research has shown that it is often beneficial to incorporate additional knowledge in neural network learning rules. Often, the additional knowledge can be encoded in the form of mathematical relations that have to be satisfied simultaneously with the demand for minimization of the cost function. Naturally, methods from the field of constrained optimization are essential for solving these modified neural network learning tasks.

In this paper, we present some recent results from our work on neural network training using constrained optimization techniques. Four examples are presented, in which the additional knowledge incorporated in the learning rule may be either network specific or problem

specific. In the first three examples, additional information about the specific type of the neural network, the nature and characteristics of its cost function landscape is used to facilitate learning in broad classes of problems. In the final example, additional information is used to solve a specific problem (polynomial factorization) and stems from the very nature of the problem itself.

II. Improving learning speed and convergence in MFNs

Conventional unconstrained supervised learning in MFNs involves minimization of a cost function of the form

$$E = \sum_p ||T_p - O_p(\boldsymbol{W})|| \tag{1}$$

with respect to the synaptic weight vector $\boldsymbol{W}$. Here p is an index running over the patterns of the training set, O_p is the network output and T_p is the corresponding target for pattern p.

Learning in feedforward networks is usually hindered by specific characteristics of the cost function landscape. The two most common problems arise because

- of the occurrence of long, deep valleys or troughs that force gradient descent to follow zig-zag paths.
- of the possible existence of temporary minima in the cost function landscape.

In order to avoid zig-zag paths in long deep valleys, it is desirable to align current and previous epoch weight update vectors as much as possible, without compromising the need for a decrease in the cost function. Thus, satisfaction of an additional condition is required, amounting to maximization of the quantity $(\boldsymbol{W} - \boldsymbol{W}_c)(\boldsymbol{W}_c - \boldsymbol{W}_l)$ with respect to the synaptic weight vector $\boldsymbol{W}$ at each epoch of the algorithm. Here $\boldsymbol{W}_c$ and $\boldsymbol{W}_l$ are the values of the weight vectors at the present and immediately preceding epoch respectively. The additional condition can be incorporated in a learning algorithm that uses constrained gradient descent to solve the optimization problem [2]. This algorithm is much faster than BP and some of its well known

variants even in large scale problems. Some examples are shown in Table I.

III. Constrained learning algorithm inspired from a dynamical system model

In earlier work [3], the problem of temporary minima was approached in the framework of constrained learning from a rather heuristic point of view. In more recent work, we have approached the problem from a new angle, using a method that originates from the theory of dynamical systems [4]. It is well known that temporary minima result from the development of internal symmetries and from the subsequent building of redundancy in the hidden layer. In this case, one or more of the hidden nodes perform approximately the same function and the network is trapped in a temporary minimum. Introducing suitable state variables formed by appropriate linear combinations of the synaptic weights, we can derive a dynamical system model which describes the dynamics of the feedforward network in the vicinity of these temporary minima. The corresponding non-linear system can be linearized in the vicinity of temporary minima and the learning behaviour of the feedforward network can then be characterized by the largest eigenvalue of the Jacobian matrix corresponding to the linearized system. It turns out that in the vicinity of the temporary minima, learning is slow because the largest eigenvalue of the Jacobian matrix is very small, and therefore the system evolves very slowly. Moreover, it is possible to get an analytical expression which approximates the largest eigenvalue. Consequently, it is possible to incorporate into the learning algorithm an extra condition requiring maximization of the largest eigenvalue, along with the condition for lowering the cost function at each epoch. The result is significant acceleration of learning in the vicinity of the temporary minima. An example is shown in Table I.

IV. Constrained optimization method in perceptron learning

The most popular algorithms for training the single layered perceptron, namely Rosenblatt's perceptron rule [5] and the Widrow-Hoff algorithm [6], are very effective in solving

many linear discriminant analysis problems. However, for difficult problems with inhomogeneous input spaces, prohibitively long training times are reported [7]. The main difficulty stems from the fact that in difficult problems patterns that were correctly classified in previous epochs may become misclassified again later during learning. We have recently developed a novel algorithm, based on constrained optimization techniques, that overcomes difficulties with inhomogeneous input spaces.

Our algorithm takes advantage of the knowledge that patterns in the training set are represented in weight space by hyperplanes, whose position is known (it is determined by the pattern vector components). Using this knowledge, we attempt to minimize the perceptron cost function taking care not to affect the classification of already correctly classified patterns. By explicitly insisting that the weight vector does not cross hyperplanes corresponding to already correctly classified patterns, we add linear constraints to the formalism. Interestingly, the problem of achieving locally the greatest possible decrease of the cost function subject to the linear constraints, turns out to be a generalization, in a number of dimensions equal to the dimensionality of the perceptron input space, of a familiar problem from physics: the problem of finding the path followed by a particle falling under the influence of gravity and constrained by one or more planes. Mathematically, the generalized problem can be stated as a quadratic programming task, to which a fast and effective solution is proposed.

The resulting algorithm can find the solution to large scale linearly separable problems much faster that the perceptron and Widrow-Hoff algorithms. An example is shown in Table I.

Table I

Problem	Constrained Learning			Conventional		
	Method	Epochs	CPU time (s)	Method	Epochs	CPU time (s)
XOR	section II	48	0.0031	BP	182	0.119
8-3-8 Encoder	section II	38	0.1143	BP	145	0.429
XOR	section III	45	0.0027	BP	182	0.119
OCR	section IV	17	15.64	Perceptron	162	105.01

V. Problem specific example: Polynomial factorization

Polynomial factorization is an important problem with applications in various areas of mathematics, mathematical physics and signal processing. It is a difficult problem for polynomials of more than one variable, where the fundamental theorem of Algebra is not applicable.

Consider, for example, a polynomial of two variables z_1 and z_2:

$$f(z_1, z_2) = \sum_{i=0}^{N} \sum_{j=0}^{N} a_{ij} z_1^i z_2^j, \quad \text{with} \quad a_{00} = 1 \tag{2}$$

for which we seek an exact or approximate factorization of the form:

$$f(z_1, z_2) \approx \prod_{i=1,2} h_i(z_1, z_2), \quad h_1 = \sum_{l,m} w_{lm} z_1^l z_2^m, \quad h_2 = \sum_{l,m} u_{lm} z_1^l z_2^m \tag{3}$$

We can try to find the coefficients w_{lm} and u_{lm} by considering P training patterns selected from the region $|z_1| < 1,\ |z_2| < 1$. The primary purpose of the learning rule is thus to minimize with respect to the w_{ij} a cost function of the form

$$E = \sum_{p} || \prod_{i=1,2} h_i(z_{1p}, z_{2p}) - f(z_{1p}, z_{2p}) || \tag{4}$$

This cost function corresponds to a sigma-pi neural network. Unconstrained minimization of the cost function has been tried, but often leads to unsatisfactory results, because it can be easily trapped in flat minima. However, there is extra knowledge available for this problem, in the form of constraints between the coefficients of the desired factor polynomials and the coefficients of the original polynomial:

$$a_{pq} - \sum_{l,m} w_{lm} u_{p-l,q-m} = 0 \tag{5}$$

Such constraints can be incorporated into the formalism and constrained gradient descent [8]

[3] can be used to solve the resulting constrained optimization problem. It turns out that the modified constrained learning algorithm can determine the factor polynomials in factorable cases and gives good approximate solutions in cases where the original polynomial is non-factorable [9].

Table II shows the coefficients of a factorable polynomial and the corresponding exact factor polynomials, as well as the solutions obtained by BP and the constrained learning algorithm. Obviously, only the constrained learning algorithm results compare favorably with the exact result.

Table II

	Exact	Constrained learning	BP
Product Polynomial coefficient matrix $\boldsymbol{a}$	$\begin{bmatrix} 1 & 1.5 & 0.5 \\ 4 & 5 & 2.25 \\ 3 & 6.5 & 1 \end{bmatrix}$	$\begin{bmatrix} 1.0000 & 1.5000 & 0.4997 \\ 4.0000 & 5.0038 & 2.2528 \\ 3.0020 & 6.5022 & 1.0092 \end{bmatrix}$	$\begin{bmatrix} 1.0000 & 1.2007 & 0.3604 \\ 0.8566 & 2.2905 & 1.0663 \\ 0.1834 & 0.7607 & 0.7888 \end{bmatrix}$
1st Factor Polynomial coefficient matrix $\boldsymbol{w}$	$\begin{bmatrix} 1 & 0.5 \\ 1 & 2 \end{bmatrix}$	$\begin{bmatrix} 1.0000 & 0.4995 \\ 1.0010 & 1.9997 \end{bmatrix}$	$\begin{bmatrix} 1.0000 & 0.5996 \\ 0.4282 & 0.8874 \end{bmatrix}$
2nd Factor Polynomial coefficient matrix $\boldsymbol{u}$	$\begin{bmatrix} 1 & 1 \\ 3 & 0.5 \end{bmatrix}$	$\begin{bmatrix} 1.0000 & 1.0005 \\ 2.9990 & 0.5047 \end{bmatrix}$	$\begin{bmatrix} 1.0000 & 0.6010 \\ 0.4284 & 0.8889 \end{bmatrix}$

VI. Conclusion

There are many types of *a priori* knowledge that can be incorporated into neural networks learning, in the form of additional constraints that must be satisfied by the learning rule. These

constraints are usually pointed out either by the selection of the learning rule itself, or from the specific problem at hand which the neural network tries to learn. The major benefit of this learning approach is to help relax the "black box" nature of artificial neural networks and combine the merits of both connectionist and knowledge based approaches for designing and implementing efficient information systems.

References

[1] D. E. Rumelhart, J. E. Hinton and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, vol. 1, *Foundations*, D. E. Rumelhart and J. L. McLelland, Eds. Cambridge, MA: MIT Press, pp. 318-362, 1986.

[2] S. J. Perantonis and D. A. Karras, "An efficient learning algorithm with momentum acceleration", *Neural Networks*, Vol. 8, pp. 237-249, 1995.

[3] D. A. Karras and S. J. Perantonis, "An efficient training algorithm for feedforward networks", *IEEE Tr. Neural Networks*, Vol. 6, pp. 1420-1434, 1995.

[4] N. Ampazis, S. J. Perantonis and J. Taylor, "A dynamical system model for feedforward networks in the vicinity of temporary minima", *Neural Networks*, under review, 1997.

[5] F. Rosenblatt, *Principles of Neurodynamics.* New York: Spartan, 1962.

[6] B. Widrow and M. E. Hoff,"Adaptive switching circuits," in *1960 IRE WESCON Convention Record,* New York, 1960, pp. IV-96–IV-104. Reprinted in J. A. Anderson and E. Rosenfeld Eds. *Neurocomputing: Foundations of Research.* Cambridge: MIT Press, 1988.

[7] D. J. Volper and S. E. Hampson, "Quadratic function nodes: Use, structure and training," *Neural Networks,* vol. 3, pp. 93-107, 1990.

[8] A. E. Bryson and W. F. Denham, "A steepest ascent method for solving optimum programming problems", *Journal App. Mech.*, Vol. 29, pp. 247-257, 1962.

[9] S. J. Perantonis, N. Ampazis, S. Varoufakis and G. Antoniou, "Constrained learning in neural networks: Application to stable factorization of 2-D polynomials", *Neural Proc. Lett.*, to appear, 1997.