Mathware & Soft Computing 4 (1997) 195-202

Neural Networks Learning as a Multiobjective Optimal Control Problem

Maciej Krawczak
Systems Research Institute
Polish Academy of Sciences
ul. Newelska 6, 01-447 Warsaw, Poland,
e-mail krawczak@ibspan.waw.pl

Abstract

The supervised learning process of multilayer feedforward neural networks can be considered as a class of multi-objective, multi-stage optimal control problem. An iterative parametric minimax method is proposed in which the original optimization problem is embedded into a weighted minimax formulation. The resulting auxiliary parametric optimization problems at the lower level have simple structures that are readily tackled by efficient solution methods, such as the dynamic programming or the error backpropagation algorithm. The analytical expression of the partial derivatives of systems performance indices with respect to the weighting vector in the parametric minimax formulation is derived.

Key words. Artificial neural networks, supervised learning, multi-objective optimization, minimax solution.

1 Introduction

The supervised learning process of feedforward neural networks can be considered as a dynamic process, [7]. The results obtained there in a very simple and elegant way show the application of the principle of dynamic programming to the artificial neural networks supervised learning. A feedforward neural network consists of a number of simple elements (called artificial neurons) arranged in layers. Each neuron is a single input - single output static element with nonlinear activation function. The neurons are interconnected in a way shown in Fig. 1.

The supervised neural network learning is related to feeding the network with input patterns x_{pi}, where $p = 1, ..., P$ denotes the number of the pattern and $i = 1, ..., N$ denotes the number of the input, as the system input

$$X_p(0) = [x_{p1}(0), x_{p2}(0), ..., x_{pN_1}(0)]^T \quad p = 1, 2, ..., P \tag{1}$$

Fig. 1. Multi-layer feedforward neural network.

The neurons of the first layer generate outputs which are muliply by proper weights, $W_{ij}(k)$, and they are inputs for the second layer, and so on. In this way the signal flows from the first layer to the last one.

Each layer in the neural network can be considered as a stage and the neural network can be considered as a multi-stage dynamic system described by the following equation

$$X_p(k+1) = F(X_p(k), W(k), k), \quad X_p(0) \text{ given } k = 0, 1, ..., N-1 \tag{2}$$

where $\mathbf{dim}X_p(k) = N_k$, $\mathbf{dim}W(k) = M_k$, and $F(\cdot, k)$ is an aggregated activation function of k-layer.

For any input pattern $x_{\mathrm{pi}}(0)$, $p = 1, ..., P$, there exists corresponding output pattern T_p, it means there are pattern pairs $\{X_p, T_p\}$.

The cost function, it means the learning error to be minimised w.r.t. controls (weighted) is defined as

$$E = \frac{1}{2}\sum_{p=1}^{p}(T_p - X_p(N))^2 \tag{3}$$

where $X_p(N)$ is the system output (of the last layer) corresponding to the input pattern $X_p(0)$.

For any layer (stage) $k = N-1, n-2, ..., 1$ the weights (controls) must be chosen as

$$W(k) = \arg\min_{W(k)} E(\{W(k)\}, X(k)) \tag{4}$$

where $\{\cdot\}$ is a sequence of the argument from k to $N-1$.

In a pictorial way the supervised learning process of feedforward neural networks is shown in Fig. 2.

Fig. 2. Neural network learning process.

Up to now, the best, it means the most efficient and the most popular way to solve the optimization problem (2) - (3) is to use the well known the error backpropagation algorithm developed by Rumelhart, Hinton and Williams in 1986, [11]. The algorithm is based on parameter sensitivity analysis and generally speaking it is a gradient algorithm while the objective function is a multimodal function. Advantages as well as disadvantages of the backpropagation algorithm (gradient descent) are pretty well known.

Another approach [7] based on the dynamic programming applied to the multistage system, and application of the gradient method to the return function gives results similar to those of the backpropagation. These results can elucidate the nature of the backpropagation algorithm.

From another point of view it is easy to notice that the learning error E is an overall objective function and can be treated as a simple composite function of multiple systems performance indices E_p, corresponding to number of so-called training pairs (patterns) $\{X_p(0), T_p\}$, with $p = 1, 2, ..., P$,

$$\min E = \Phi(E_1, E_2, ..., E_p) = \sum_{p=1}^{p} (T_p - X_p(N))^2 \tag{5}$$

subject to

$$X_p(k+1) = F(X_p(k), W(k), k), \quad X_p(0) \text{ given } k = 0, 1, ..., N-1 \tag{6}$$

where E being a nondecreasing function, and $\dim X(k) = N_k$, $\dim W(k) = M_k$. Each function E_p is assumed to posses a finite minimum which is positive.

In this paper it is proposed to embed a difficult optimal control problem (5) - (6) (difficult due to multimodality of the objective function as well as a large number of neurons) into a family of parametrized optimal control problems that are much easier to be solved by existing efficient solution algorithms. The solution scheme should be devised such that the solution of the parametric problem converges to the optimum of the original problem through successively adjusting the parameter vector in the iteration process.

2 Embedding into multiobjective optimization problems

The idea of using multiobjective optimization to solve some class of nonlinear programming problems was first proposed in [4]. In [8] we can find extension of this idea to dynamic systems.

Let us consider the following optimization problem of a discrete-time dynamic system,

$$\min E = \Phi(E_1, E_2, ..., E_P) = \sum_{p=1}^{P} (T_p - X_p(N))^2 \tag{7}$$

subject to

$$X_p(k+1) = F(X_p(k), W(k), k), \quad X_p(0) \text{ given } k = 0, 1, ..., N-1 \tag{8}$$

In our case, the overall performance index E is a strictly increasing function of E_p $(p = 1, .., P)$, i.e.

$$\frac{\partial E}{\partial E_p} = 1.$$

.

A corresponding multiobjective multi-stage optimization problem can be formulated as follows:

$$\min \begin{bmatrix} E_1 \\ ... \\ E_P \end{bmatrix} = \min \begin{bmatrix} (T_1 - X_1(N))^2 \\ ... \\ (T_P - X_P(N))^2 \end{bmatrix} \tag{9}$$

subject to

$$X_p(k+1) = F(X_p(k), W(k), k), \quad X_p(0) \text{ given } k = 0, 1, ..., N-1 \tag{10}$$

In general the solution to the multiobjective problem (9)-(10) is not unique. A solution W of this problem is said to be noninferior if there does not exist another feasible W such that

$$E_p(W) \leq E_p(\hat{W}) \tag{11}$$

for all $p = 1, 2, ..., P$, with strict inequality for at least one p.

For the optimization problem (9)-(10) the following theorem can be proved.

Theorem 1. *The optimal solution of problem (7)-(8) is attained by a noninferior solution of the multiobjective optimization problem given in problem (9)-(10).*

The most common approaches to generate the set of noninferior solutions are the -constraint method and the weighting method [9].

The noninferior solutions of problem (9)-(10) can be generated by solving the following ϵ-constraint method form:

$$\min E_1(W) \tag{12}$$

subject to

$$E_p(W) \leq \epsilon_p, \quad p = 2, 3, ..., P \tag{13}$$

and

$$X_p(k+1) = F(X_p(k), W(k), k), \quad X_p(0) \text{ given } k = 0, 1, ..., N-1 \tag{14}$$

Theorem 2. *Assume that the set of noninferior solutions of problem (9)-(10) can be parametrized by μ_{1p}, $p = 2, 3, ..., P$, that are the optimal Kuhn-Tucker multipliers associated with the -constraint in Equ. (13). Thus the optimal solution of the dynamic optimization problem (7)-(8) is then reached by the noninferior solution that satisfies the following equations [9]:*

$$\frac{\partial E}{\partial E_p} - \mu_{1p} \frac{\partial E}{\partial E_1} = 0, \quad p = 2, 3, ..., P \tag{15}$$

For the case considered in this paper, the objective function E is of an additive form, then the solution of Equs. (7)-(8) is attained by the noninferior solution with all μ_{1p} equal to one.

If problem (9)-(10) is convex, then the noninferior solutions of problem (7)-(8) can be obtained by solving the following weighting form:

$$\min \sum_{p=1}^{p} v_p E_p(W) \tag{16}$$

subject to

$$X_p(k+1) = F(X_p(k), W(k), k), \quad X_p(0) \text{ given } k = 0, 1, ..., N-1 \tag{17}$$

where

$$v_p \geq 0, \quad p = 1, 2, ..., P; \quad \sum_{p=1}^{P} v_p = 1.$$

.

Theorem 3. *If the set of noninferior solutions of problem (9)-(10) can be parametrized by the overall weighting vector , the optimal solution of the nonseparable dynamic optimization problem given in (7)-(8) is reached under certain conditions by the noninferior solution that satisfies the following equations:*

$$\frac{\partial E/\partial E_1}{v_1} = \frac{\partial E/\partial E_2}{v_2} = ... = \frac{\partial E/\partial E_p}{v_p} \tag{18}$$

For the case considered in this paper due to the additive form of the performance index the optimal solution of problem (7)-(8) is attained by the noninferior solution with all v_p equal to $1/P$.

The motivation of this paper is to consider the optimization problem (7)-(8) as a embedding problem, it means to embed this optimization problem into a family of parametrized optimization problems that are much easier to be solved by existing algorithms.

Let us consider the following weighted minimax formulation for problem (7)-(8):

$$\min \max\{v_1 E_2, v_2 E_2, ..., v_p E_p\} \tag{19}$$

subject to

$$X_p(k+1) = F(X_p(k), W(k), k), \quad X_p(0) \text{ given } k = 0, 1, ..., N-1 \tag{20}$$

where weighting coefficient v_1 is always set to one and weighting coefficients v_p, $p = 2, ..., P$ are nonnegative. In (19) the maximization is performed among P weighted systems (for any training pair) performance indices while minimization is carried out over the feasible region of W.

3 Iterative minimax solution

Several theorem strictly related to the conversion of the optimization problem (7)-(8) into another optimization problem (19)-(20) can be proven.

If we define by W^* the set of solution of the problem (7)-(8) and by W^* the union of sets of solutions of the weighted minimax problem (19)-(20) then it is possible to prove the following

Theorem 4. *The intersection of W^* and $W_v^*$is nonempty, i.e.*

$$W^* \cap W_v^* = \emptyset.$$

.

The conclusions of this theorem is very clear, namely if the optimization problem (7)-(8) has a unique solution then the optimal solution can be generated by the weighted minimax solution of (19)-(20). In the case of existing many solutions of (7)-(8) then at least a nonempty subset of W_v^*can be generated by the weighted minimax problem (19)-(20).

It is possible to show that the minimax optimization problem (19)-(20) is equivalent to the following form:

$$\min \varphi(y) \tag{21}$$

subject to

$$\begin{aligned} & v_p E_p(W) \leq y, \quad p = 1, 2, ..., P \\ & X_p(k+1) = F(X_p(k), W(k), k), \quad X_p(0) \text{ given } K = 0, 1, ..., N-1 \end{aligned} \tag{22}$$

where y is an auxiliary variable, while $\varphi(y)$ is any increasing differentiable function. Minimizing a strictly increasing function of y will minimize the maximum value among $v_1 E_1, v_2 E_2, ..., v_p E_p$.

For positive Kuhn-Tucker multipliers occurring in the Lagrangian of problem (21)-(22) this problem can be rewritten as

$$\min \varphi(y) \tag{23}$$

subject to

$$-\frac{y}{E_p(W)} \leq -v_p, \quad p = 1, 2, ..., P$$
$$X_p(k+1) = F(X_p(k), W(k), k), \quad X_p(0) \text{ given } k = 0, 1, ..., N-1 \tag{24}$$

Under some conditions which are satisfied for the considered problem (19)-(20) the original problem can be solved in a two-level solution structure. The overall performance index is a function of v,

$$E^* = E^*(v). \tag{25}$$

After derivation of E with respect to v the search of the optimal point $E(v)$ can be done by the standard gradient method. It is a simple unconstrained problem except that all v_p must be nonnegative.

It can be shown from that for the optimal weights W^* the optimal weighting vector v^* can be expressed as follows

$$v^* = \left[1, \frac{E_1(W^*)}{E_2(W^*)}, \frac{E_1(W^*)}{E_3(W^*)}, ..., \frac{E_1(W^*)}{E_P(W^*)}\right]. \tag{26}$$

For a given weighting vector

$$v = [v_1, v_2, ..., v_P]^T$$

the weighted minimax problem (19)-(20) is solved at the lower level using appropriate solution schemes e.g. the backpropagation algorithm or a method based on the dynamic programming [7]. At the upper level, the optimal stopping condition,

$$\frac{\partial E^*}{\partial v_p} = 0 \text{ for } p = 2, 3, ..., P \tag{27}$$

is checked upon receiving the solutions from the lower level. If this conditions are not satisfied then a gradient algorithm can be used to update the value of the weighting vector

$$v_p^{j+1} = \max\left\{0, v_p^j - \alpha \frac{\partial E^*}{\partial v_p}\right\}, \quad p = 2, 3, ..., P, \tag{28}$$

where p is the iteration number and α is a step size parameter which can be adjusted during the iteration to guarantee a decrement of overall objective function E. The problem (19)-(20) at the lower level is then solved again for this new value v. The iteration process continues until all

$$\frac{\partial E^*}{\partial v_p}\text{'s vanish.}$$

4 Conclusions

A method of conversion of supervised leering of feedforward artificial neural networks into an iterative minimax problem is proposed in this paper. The proposed method is actually implemented on a computer IBM PC-486. It seems that the results should be applied for updating values of weights for neural networks.

References

[1] Anderson, J.A.; Rosenfeld, E., *Neurocomputing: Foundation of Research.* Cambridge: MIT Press, 1988.

[2] Bellman, R., *Dynamic programming.* Univ. Press, Princeton, New Jersey, 6 eds. 1972.

[3] Dreyfus, S.E., Artificial Neural Networks, Back Propagation, and the Kelley-Bryson Gradient Procedure. *Journal of Guidance*, vol. **13**, no. 5, 1960, pp 926-928.

[4] Geoffrion, A.M. Solving Bicriterion Mathematical Problems, *Operation Research*, vol. **15**, 1967, pp 39-54.

[5] Jacobson, D.H.; Mayne, D.Q., *Differential Dynamic Programming.* Am. Elsevier Pub. Comp., N-Y, 1970.

[6] Kelley, H.J., Gradient Theory of Optimal Flight Paths. *ARS Journal*, vol. **30**, no. 10, 1960, pp. 947-954.

[7] Krawczak, M.; Mizukami, K. The control theory approach to perceptron learning process. *44 Conference of IEE of Japan*, Okayama, 1994.

[8] Li, D.; Haimes, Y.Y., Extension of dynamic programming to nonseparable problems. *Computer and Mathematics with Applications*, vol. **21**, 1991, pp 51-56.

[9] Li, D.; Haimes, Y.Y., Multilevel Methodology for a class of nonseparable optimization problems, *International Journal of Systems Science*, vol. **21**, 1990, pp 2352-2360.

[10] Luenberger, D.G., *Linear and Nonlinear Programming.* II Edition, Addison-Wesley, Reading, Massachusetts, 1984.

[11] Rumelhart, D.; Hinton, G.E.; Williams, R.J., *Parallel Distributed Processing,* vol. 1, edited by D. Rumelhart, J. McClelland, and the PDP Research Group, MIT Press, Cambridge, MA, Chap. 8, 1986. (Also in [1]).

[12] Werbos, P.J., Maximising Long-Term Gas Industry Profits in Two Minutes in Lotus Using Neural Network Methods. *IEEE Trans. on Systems, Man, and Cybernetics,* Vol. **19**, No. 2, 1989, pp. 315-333.