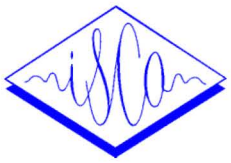

First European Conference on
Speech Communication and Technology
EUROSPEECH '89
Paris, France, September 27-29, 1989

NEW BACKPROPAGATION ALGORITHM USING QUADRATIC POTENTIAL FUNCTIONS, AND AN EXPERIMENT ON ISOLATED WORD RECOGNITION .

ENRIC MONTE, EDUARDO LLEIDA, JOSE B. MARIÑO.

DPTO. TEORIA DE LA SEÑAL Y COMUNICACIONES, UNIVERSIDAD POLITECNICA DE CATALUÑA, SPAIN.

ABSTRACT.

This paper presents a new algorithm to train multilayered perceptrons, using quadratical potential functions. This new algorithm is compared in an isolated word recognition task, with the back propagation algorithm that uses linear combinations of the inputs. Some pattern recognition techniques are also used to reduce the dimensinality of the input pattern in order to reduce the computational burden of the training and the recognition. The algorithm that uses quadratic potential functions yields better results in the recognition task.

1 INTRODUCTION

In this paper we propose a new algorithm, to teach multilayered classifiers, and we will compare the back propagation algorithm using quadratical potential functions with the traditional back propagation algorithm (1),(3). This algorithm is based on the use of quadratical potential functions (4) instead of hyper planes in the hidden units. The reason for choosing potential functions is due to the fact that the multilayered perceptron needs a great number of hidden units when the classes to be classified form closed surfaces in the space of the inputs , and the distribution of the classes is not gaussian. This paper also compares the traditional multilayered perceptron, with linear decision surfaces, with the new algorithm; the comparison is done on the experiment of the recognition of the catalan digits. Pattern recognition techniques are used to reduce the dimensionality of the test signals. This is due to the fact that the set of frames of the digits after the preprocessing was excessively long: 30 frames of 8 parcor coeffients each; so it was decided to do a feature selection through orthogonal expansion of the digits (2), in order to reduce the dimensionality of the input patterns, this orthogonal expansion reduced dimensionality in time, representing each digit with three frames only. In this way the dimensionality of the input vector was reduced from 240 real numbers to 24 real numbers.

2 THE QUADRATICAL BACKPROPAGATION ALGORITHM.

First of all, we are going to give a geometrical justification of the algorithm that is presented in this paper. This justification will show that for a fixed number of hidden units, the backpropagation algorithm that uses quadratical potential functions in the hidden units yields a better classification performance. It must be emphasized that the following is a justification, not a demonstration; nevertheless, the experimental results confirm the assumptions that are made.

In the case of multilayered networks, where the units combine linearly the inputs, the hidden units form the hyperplanes (if the input vector is of dimension two as in the following example, it will be lines) that are used to close the part of the plane that has to be classified as a separate class. As it is shown in the figure 1, the output units, classify the class if the point at the input is inside the area that is surrounded by the lines.

$$output = f(W^T \ in\_vec) \qquad (1)$$

$$ouput = f(in\_vec^T A in\_vec + b^T \ in\_vec + c) \qquad (2)$$

Where: A: is a weight matrix

b: is a weight vector

c: is a weight scalar

in_vec: is the input data vector to the network.

W: is a weight vector.

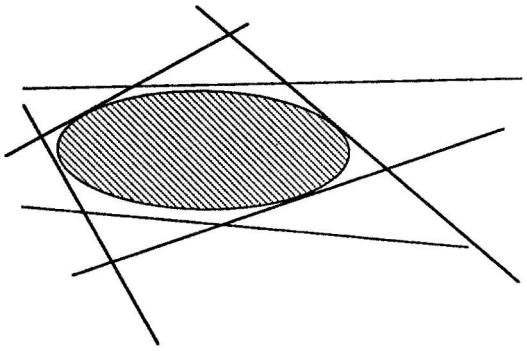

Figure 1: The elipsoid is the decision boundary of the class and the lines are the decision boundary generated by a multilayered perceptron.

This work was supported by the PRONTIC grant nº 105/88

As it can be seen in Figure1, in order to represent the decision boundary with lines, we need as many lines as points of the surface, which means that we need a network with an infinite number of hidden units. If instead of using units with a transfer function as shown in 1 we use units with transfer function that is a potential function of the input as shown in 2, each hidden unit will generate a closed desicion boundary.

In the case of units that use this kind of functions, the classification boundaries shown in the figure 1 , which correspond to an elipsoid can be done with only one unit in the hidden layer.

This kind of algorithm is best tailored to problems where the surfaces to be classified are closed, and complicated. In this case the number of units in the hidden layer should be much lower in the quadratical case than in the back propagation algorithm with units that combine linearly the inputs. Of course the dimension of the input vectors must not be too high, because the number of weights in each unit grows quadratically. Nevertheless the experiments that we have done, show that when the dimension of the input vector is high, the linear back propagation algorithm converges to a solution after a number of iterations prohibitive in CPU time. So any way the quadratical backpropagation algorithm behaves better than the linear back propagation algorithm when the decision boundaries are complicated.

DERIVATION OF THE TRAINING ALGORITHM FOR NETWORKS WITH UNITS THAT USE QUADRATICAL POTENTIAL FUNCTIONS.

Let J be the cost function of a multilayered classifier defined as follows:

$$J = \sum_{np} \sum_{i} (r^i_{np} - y^i_{np})^2 \qquad (3)$$

where np: is the number of presentations.

r^i_{np}: is the reference signal for the output i.

y^i_{np}: is the output i of the network.

We will use a gradient search technique to find the minimum of the cost function J. The recursion used for the actualization of the weights is the following:

$$w^i_j(np+1) = w^i_j(np) + \varepsilon \bullet \partial J/\partial w^i_j + \alpha \bullet (w^i_j(np) - w^i_j(np-1)) \qquad (4)$$

where: $w^i_j(np)$: is the weight between the node j and the node i at the np iteration.

ε: is the adaptation step.

α: is the momentum.

It must be noted that when we talk of the unit j; it can be a unit at the output layer or at a hidden layer; for each case, the way of computing the value of the gradient $\partial J/\partial w^i_j$ will be indicated.

From now on: Y_i will mean the input vector to the node i ; and x_i the output of this node; y_i will be $y_i = f(x_i)$, where f(.) is the sigmoid function.

The relation between x_i and Y_i is

$$x_i = Y_i^T A Y + B^T Y_i + c \qquad (5)$$

In order to find a minimum of the cost function we will have to calculate three different kinds of

A, the one with the vector B and the one with the scalar c) which are:

a) $\partial J/\partial a^i_{jl}$: gradient of the term j l of the matrix A of the unit i.

b) $\partial J/\partial b^i_j$: gradient of the term j of the vector B of the unit i.

c) $\partial J/\partial c^i$: gradient of the scalar c of the unit i.

The general de derivation of the expression of the gradient of J with a weight is the following:

$$\partial J/\partial w = \partial J/\partial x \bullet \partial x/\partial w \qquad (6)$$

$$\partial J/\partial x = \partial J/\partial y \bullet \partial y/\partial x \qquad (7)$$

$\partial J/\partial y^i$ has a different expression which depends on wether the unit is an output unit or a hidden unit.

If the unit is an output unit we have:

$$\partial J/\partial y^i = -2(r^i - y^i) \qquad (8)$$

If the unit is a hidden unit then we have:

$$\partial J/\partial y^i = \sum_{k=1}^{n\_f} \partial J/\partial x^k \bullet \partial x^k/\partial y^i \qquad (9)$$

where n_f: is the number of units in the next layer.

If we derive the equation 5 we get:

$$\partial J/\partial y_i = \sum_{k=1}^{n\_f} \partial J/\partial x^k \bullet (2a^k_{ii} y_i + \sum_{l \neq i}^{n\_f} a^k_{il} y_l + b^k_i) \qquad (10)$$

The term corresponding to $\partial x/\partial y$ is quite straitforward, because y is related with x by a sigmoid;

$$\partial x/\partial y = y(1-Y) \qquad (11)$$

a) For the terms related with a:

The elements of the diagonal of the matrix A.

$$\partial x^l / \partial a^l_{jj} = y^{j^2} \qquad (12)$$

For the elements that are outside of the diagonal (the matrix A is a symmetric matrix)

$$\partial x^l / \partial a^l_{ji} = y^j y^i \qquad (13)$$

b) c) For the terms related with b and c:

$$\partial J / \partial b^l_j = \partial J / \partial x^l \bullet \partial x^l / \partial b^l_j \qquad (14)$$

$$\partial J / \partial c^l = \partial J / \partial x^l \bullet \partial x^l / \partial c^l \qquad (15)$$

At this moment we have all the terms that are needed, to teach the network. The complete algorithm is shown in Table I.

TABLE I

LOOP: For all the training patterns

-Present at the input a training pattern an examine the output.

-Calculate: $\partial J / \partial y^l$ for all the output units, with the equation (8).

For each output unit:

-Calculate $\partial J / \partial x^l$

- For the weights: A, B, c calculate the derivatives respect to each weight: i.e.

derivative respect matrix A: $\partial J / \partial a^l_{ji}$

derivative respect vector B: $\partial J / \partial b^l_j$,

derivative respect scalar c: $\partial J / \partial c^l$

For the previous layers:

At the layer i calculate $\partial J / \partial y^i$ with the equation (10), where the term $\partial J / \partial x^k$ is a term of the unit k of the next layer.

The derivatives of the weights are calculated in the same way once the term $\partial J / \partial y^i$ is known, and the weights are updated again using the equation: (4).

---

3 RECOGNITION EXPERIMENTS

Test data base

A data base consisting of ten repetitions uttered by seven male and three female speakers (1000 words) of the Catalan digits (table II) recorded in a quite room were used.

1 /u/	2 /dos/	3 /tres/	4 /kwatre/
5 /sink/	6 /sis/	7 /set/	8/vuit/ /wuit/
9 /nou/	0 /zeru/ /seru/		

table II. Pronunciation of the Catalan digits.

Preprocessing

The speech signal was sampled at 8 KHz, pre-emphasized and the beginning and end of every utterance were detected automatically by means of an algorithm based on the signal energy; 8 Log-Area coefficients were computed each 15 ms. using frames of 30 ms. of the speech signal. A typical Hamming smoothing window was applied to the data.. After the LPC analysis, the parametrized frames are normalized to a fixed number N of frames, being N equal for all the digits.

Orthogonal Expansion

The input signal, which is formed by a secuence of N frames, each one with P (8) parcor coefficients, is too long to be used efficiently in a multilayer classifier. So it was decided to use pattern recognition techniques to reduce the dimensionality of the input pattern. This reduction in the dimensionality was obtained by the use of an orthogonal expansion of the set of input frames. The following lines explain the method that was used.

Given a NxP matrix Y of Log-Area parameters representing N frames, we can find a finite family of orthogonal functions, which represent the time evolution of Log-Area parameters, by means of the Principal Component Analysis of an average of the matrix YY^T of each vocabulary word (2). Thus, given an input template of NxP, the new template of MxP dimension is computed using M orthogonal functions, where M is less than N. In our system, N is 30 frames and the optimum value of M, which gives the least recognition error in a classical pattern recognition approach (1) is 3.

EXPERIMENTS AND TOPOLOGY

The recognition was done with two networks; one of them was trained using the quadratical back propagation algorithm introduced in this paper, and the other one was trained using the back propagation algorithm, which is described in (1).

As it is well known the perfomance of back propagation type algorithms depend on the values that are given to the adaptation step and the momentum. On our experiment we trained the linear back propagation algorithm with several values of these two parameters and found that the best results were obtained when we took the adaptation step as 0.3 and the momentum as 0.8.

We followed the same procedure for the quadratical back propagation algorithm and found that the best parameters were 0.1 for the adaptation step and 0.3 for the momentum.

The topology of the network was with 16 units in the hidden layer for the back propagation algorithm presented in (1) and of 8 units in the hidden layer for the quadratical backpropagation. The reason for it, is that the new algorithm works better than the old even with a lower number of units in the hidden layer.

In this paper we are going to compare the recognition results of the two algorithms when trained with a limited number of iterations.

The training of each network was done with the nine speakers of the data base, and the recognition scores were obtained with the speaker that was left out of the training. In Table III the perfomance of the two algorithms are compared for a training of 50 presentations; where each presentation is defined as the training of the network with a sample of each digit. Each realization of each digit was taken at random. The perfomance is measured as the number of errors that the network makes, when trained with a number of presentations.

CONCLUSION

In this paper we present a new algorithm to train mutilayered networks. This algorithm uses quadratical functions instead of a linear combination of the inputs. In this way, the number of hidden units that are needed to classify classes were the decision boundaries are closed and complicated is reduced and the training time is also reduced. The algorithm that we present in this paper, yields a better perfomance: i.e. the convergence rate is faster and the recognition accurancy is beter with a lower number of hidden units; than the linear back propagation algorithm, when used in the digit recognition problem. In this paper, we also present the use of an orthogonal expansion of the parametrized voice signal, to reduce the dimensionality of the input to the network, and thus the computational burden of the recognition process. The combination of the two techniques: the quadratical units and the orthogonal expansion, gives better recognition results.

REFERENCES.

(1) D. Rumelhart, G.Hinton and the PCP Research Group, Parallel Distributed Processing: Explorations in the structure of cognition. Cambridge, MA: Bradford Books. 1986.

(2) E. LLeida, C.Nadeu, and J. Mariño, " Feature Selection Through Orthogonal Expansion In Isolated Word Recognition". Melecon-89 Lisbon 1989.

(3) D.C. Plaut and G.E. Hinton. " Learning sets of filters using back propagation". Computer speech and language. Number 1. March 1987.

(4) R.C. Gonzales, J.T. Tou. " Pattern Recognition Principles". Addison-Wesley 1974.

TABLE III

N:number of presentations.

S: speakers

Recognition rate (%) with the linear backpropagation algorithm.

N\S	0	1	2	3	4	5	6	7	8	9
50	0	11	18	2	25	43	4	19	13	21
100	0	6	30	2	20	34	12	0	10	14

Recognition rate (%) with the back propagation algorithm using quadratical potential functions.

N\S	0	1	2	3	4	5	6	7	8	9
50	3	8	15	5	7	15	4	8	11	16
100	0	3	5	1	5	10	4	2	1	3