

COMPARISON OF LEARNING ALGORITHMS FOR FEEDFORWARD NEURAL NETS

J. A. Nossek P. Nachbar A. J. Schuler

Institute for Network Theory and Circuit Design
Technical University of Munich
Germany, e-mail: nossek@nws.e-technik.tu-muenchen.de
Neural Network Systems, Inc.
86316 Friedberg, Germany

ABSTRACT

Several well known learning algorithms for feedforward two-layer neural nets and an improved version of Madaline I have been investigated and compared with respect to learning effort and classification capacity. These results, based on random training patterns, and their significance for generalization have been verified with real life data for ICR/OCR.

1. INTRODUCTION

The ability of neural network to generalize, i.e. to extract a rule from a given set of training data, is probably the most interesting feature of them in terms of real applications. If the paradigm of supervised learning is adopted, the generalization ability is determined by the architecture and the learning algorithm used to adapt the weights. Certainly the overall performance of the system is also influenced by other parameters such as the pre- and postprocessing [1, 2] and the significance of the training data [3], still a superior learning algorithm should improve the overall performance.

In our investigation we focused on two-layer feedforward neural networks and compared several well known learning algorithms for this architecture based on their classification capacity on random data [4, 2]. Our investigation shows that the Madaline I algorithm of Widrow and Hoff [5, 6] gives the best performance in this setting. By suitably interpreting the principle of "minimal weight disturbance" we were able to generalize the Madaline I algorithm to the extent that in each iteration several training pattern are considered. This leads to an improvement of approximately 20% as compared to the original version, which is also confirmed by a comparison based on the classification of handwritten digits.

2. COMPARISON

When comparing learning algorithms basically three different approaches are utilized: artificial benchmark problems such as the XOR or encoder-decoder problem [7, 8], selected applications or the ability to learn random data [4]. Surely a statement based on a real application is significant, but it is virtually impossible to compare results of different authors. We preferred random data over benchmark-problems, mainly because the rule to be extracted is very "artificial" as compared to real applications, since very simple solutions do exist. In this sense the random data can be interpreted

as a worst case scenario. Moreover it seems harder to optimize an algorithm for the task of learning random patterns, than for some simple benchmark problem. The disadvantage is that the generalization ability of the network cannot be measured directly.

2.1. The classification capacity

One theoretical framework for classification and generalization of parameterized classifiers is the Vapnik-Chervonkis theory (VC-theory) [9, 8]. The central result of the VC-theory is intuitively clear: the generalization ability is proportional to the number of correctly learned training patterns and inversely proportional to the "complexity" of the architecture. More precisely

$$\text{Prob} \left(\max_{\tilde{B}: \{0,1\}^n \rightarrow \{0,1\}} |g(\tilde{B}) - g_p(\tilde{B})| > \epsilon \right) \leq 4m(2p)e^{-\frac{\epsilon^2 p}{8}},$$

where g is the fractional error of the Boolean function \tilde{B} with respect to the perfect rule B , and g_p the estimate thereof on a arbitrary test pattern sample of size p . The complexity of the architecture is grasped by the growth function $m: \mathbb{N} \rightarrow \mathbb{R}$. In the case of neural networks it is useful to define the classification capacity [4].

Definition 1 (Classification capacity) *Let N be the total number of adjustable weights in a neural network F . Further let M be the training pattern set of size p consisting of independent identically distributed Gaussian random variables, assigned randomly and with equal probability to the classes $+1$ and -1 . The classification capacity is defined as*

$$\alpha = \frac{pcr}{N},$$

where pcr is the size of the learning set M , which can be learned without error, with a probability of 0.5.

It was shown for feedforward multilayer perceptrons, that the generalization ability is bounded by a constant times the classification capacity. Further there is strong experimental evidence [10, 2] that the generalization ability is directly proportional to the classification capacity.

2.2. Experimental results

In our investigation we considered several versions of the standard Backpropagation algorithm [11, 7, 12], the Tiling algorithm [13, 8] as a representative for the family of constructive learning algorithms and various versions of the

Madaline I algorithm [10]. In order to fix the terminology let us firstly define a Madaline algorithm for a “committee machine” (Fig. 1). Since the majority logic is a self dual boolean function, we assume without loss of generality that there are no thresholds in the input layer and that the desired output τ^μ for each training ξ^μ is equal to +1, for $\mu = 1, \dots, p$.

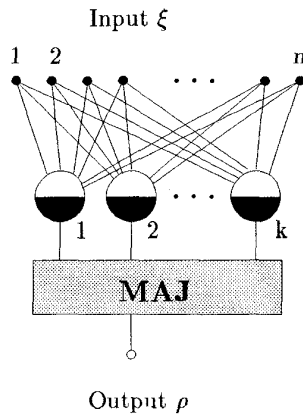


Figure 1. Committee-machine architecture (MAJ - majority logic)

Definition 2 A Madaline algorithm for the “committee machine” with $k = 2l + 1$, $l \in \mathbb{N}$ adapts the weights W_i of the hidden neuron i according to (assuming $\tau^\mu = +1$)

- 1 Choose a pattern ξ^μ .
- 2 If the network output $\rho(\xi^\mu) = +1$ go back to 1.
- 3 Else find $1 \leq m \leq \lfloor k/2 \rfloor - k_+$ neurons i_1, \dots, i_m with local fields minimally below zero (k_+ # of positive local fields $h_i^\mu = W_i \xi^\mu$).
- 4 Adapt each weight W_{i_h} according to a Hebbian rule

$$W_{i_h}^{(s+1)} = W_{i_h}^{(s)} + \Delta x \xi^\mu.$$

- 5 Go back to one.

The definition is purposely quite general and several choices have to be made in order to really specify an algorithm. One specification concerns step 3. We will call a Madaline algorithm with $m = 1$ a Least Action algorithm and with $m = \lfloor k/2 \rfloor - k_+$ a Madaline I algorithm, respectively. The experimental setting for random patterns was as follows: 10, 20 and 30 input neurons; 3, 5 and 7 hidden neurons and 1 output neuron. In all experiments p_{cr} was determined by using at least 30 different training sets or when recognizing handwritten digits 30 independent runs on the training set for each size p .

The results of our experiments are summarized in Table 1 in the case of 30 inputs. It was experimentally shown [4], that the classification capacity shows a saturation characteristic as a function of the number of inputs. We could confirm these results and with a value of 30 inputs we

were well within the saturated region. Moreover some algorithms exhibit a slight increase in the classification capacity as a function of the number of hidden neurons, though not qualitatively influencing the results. First of all we could

algorithm	Classification cap.
Tiling	≈ 2.0
Backprop. 2. Power	≈ 1.1
Quickprop.	≈ 2.0
mod. Backprop, 4. Power	2.7
Linear-Least-Action [4]	2.0
Adaptive-Least-Action [4]	2.1
Madaline I	2.8

Table 1. Classification capacity

confirm the poor performance of the standard Backpropagation algorithm and an improvement by a factor of 2 by the Quickpropagation algorithm [14]. We found that it has about the same quality as the tiling algorithm. This is further increased by the modified Backpropagation algorithm, where the weights of the second layer were fixed and implemented the “committee machine”. Hence the BP-algorithm can take no advantage of the additional degrees of freedom offered by the adaptation of the output weights, and even worse the performance deteriorates dramatically. Comparing the various Madaline algorithms, one can conclude that a random choice of patterns is by far the best and there is no real difference between the Least-Action and the Madaline I version. We would like to note that our values for the classification capacity are in accordance with other published results [4, 6]. As far as the learning rule is concerned, we only investigated the Perceptron rule ($\Delta x = 1$) and the Adaline rule, which proved to be superior. Another merit of the Madaline I algorithm is its simplicity and speed.

3. THE MADATRON ALGORITHM

In order to further improve on the results of the Madaline I algorithm, we were looking for a formulation of the Madaline algorithm applicable not only to the “committee machine” and using this as a starting point for the definition of an algorithm, which uses more than one training pattern in each iteration, hence taking into account the correlation between the patterns.

3.1. The principle of minimal weight disturbance

Already Widrow [5, 6] pointed out, that the Madaline I algorithm is based on the principal of minimal weight disturbance: *minimize the weight change necessary for fulfilling a task or for equal weight changes choose the one with the greatest yield*. This is a rather vague and flexible formulation and there are certainly different possibilities for a concrete definition of “task” and “yield”, respectively. We propose to interpret “task” as “classifying the pattern correctly with a given minimum robustness [15, 2]” and to use the euclidean norm to measure the “weight change”, since this immediately yields the Madaline I algorithm where the Hebbian rule is given by the Adaline algorithm, as will be shown. Applying the minimal weight disturbance principal to $r \geq 1$ patterns $\xi^{\mu\beta}$ ($\beta = 1, \dots, r$), yields the following

convex optimization program with linear inequality constraints for the weight update ΔW_i of the i -th hidden neuron ($i = 1, \dots, k$):

$$\min_{\Delta W_i} \|\Delta W_i\| \text{ subject to } (W_i + \Delta W_i)^t \xi^{\mu\beta} \geq \|\xi^{\mu\beta}\|. \quad (1)$$

This defines the minimal ΔW_i , such that the distance of the patterns $\xi^{\mu\beta}$ to the hyperplane $W_i + \Delta W_i$ is at least equal to one.

In order to classify a training pattern ξ^μ correctly by the "committee machine" with k hidden neurons, it is necessary that at least $\lceil k/2 \rceil$ hidden neurons have a positive local field $h_i = W_i^t \xi^\mu > 0$. Replacing the majority function of the "committee machine" machine by an arbitrary Boolean function $B : \{-1, 1\}^k \rightarrow \{-1, 1\}$, this can be rephrased as the internal representation ρ^μ of the pattern ξ^μ has to be in the set $B^{-1}(+1)$. Note that since we do not longer use the majority logic, we have to consider $+1$ and -1 outputs. Henceforth we will denote the desired output of the pattern ξ^μ by τ^μ .

As can be seen in the case of the majority logic, it is necessary to use "do-not-care" symbols to represent the set $B^{-1}(+1)$, since one has to express the fact, that for the internal representation $(+1, +1, -1)$ the $+1$ are vital for the correct classification, whereas the -1 could be replaced by a $+1$. Therefore a proper representation of this internal representation is $(+1, +1, *)$.

Definition 3 Let $M \subseteq \{-1, 1\}^k$ be a set of Boolean vectors and $\mathcal{L}(M) := \{\phi \in \{-1, *, 1\}^k : \phi \subseteq M\}$. A boolean vector $\psi \in \mathcal{L}(M)$ is prime, iff from $\psi \subseteq \phi$ follows that $\phi = \psi$ for any $\phi \in \mathcal{L}(M)$. The set of all prime vectors of M is $\mathcal{P}(M)$.

Using this definition the proper description of the internal representations, which lead to a correct classification of the pattern ξ^μ is $\mathcal{P}(B^{-1}(\tau^\mu))$ and for r patterns ξ^μ therefore $\mathcal{P}(B^{-1}(\tau_1^\mu)) \times \dots \times \mathcal{P}(B^{-1}(\tau_r^\mu)) := \mathcal{P}^{\times r}(B^{-1}(\tau^{\mu\beta}))$

One last precaution has to be taken, in order to comply with the principle of minimal weight disturbance and avoid "unnecessary" weight changes. Consider the internal representation $(+1, -1, -1)$ and the desired internal representation $(+1, +1, *)$ of the pattern ξ^μ . If the distance of the pattern to the plane defined by W_i is less than one, then (1) would still lead to a weight change, although the pattern already is "correctly" classified at the first neuron. In order to circumvent this, we define an embedding function $d : \mathbb{R} \rightarrow \mathbb{R}$:

$$d(x) := \begin{cases} 1 & \text{if } x < 0 \text{ or } x > 1 \\ x & \text{otherwise} \end{cases} \quad (2)$$

Putting the pieces together, the MadaTron algorithm is defined as follows.

Definition 4 Let $\Phi \in \mathcal{P}^{\times r}(B^{-1}(\tau^{\mu\beta}))$ be a r tuple of prime vectors and $\Phi_i \in \{-1, *, 1\}^k$ its i -th component. The MadaTron algorithm is defined by:

1 Choose $r > 1$ patterns $\xi^{\mu\beta}$ (randomly).

2 Let $\Delta W_i(\Phi_i, \xi^1, \dots, \xi^r)$ be the solution of

$$\min \|\Delta W_i\| \text{ subject to } \phi^{\mu\beta} (W_i + \Delta W_i)^t \xi^{\mu\beta} \geq \begin{cases} \|\xi^{\mu\beta}\| d(h^{\mu\beta} / \|\xi^{\mu\beta}\|) & \text{if } \phi^{\mu\beta} \neq * \\ \text{arbitrary} & \text{else} \end{cases}$$

3 Let $\Delta W(\phi^*, \xi^{\mu_1}, \dots, \xi^{\mu_r}) := (\Delta W_1(\phi_1^*, \xi^{\mu_1}, \dots, \xi^{\mu_r}), \dots, \Delta W_k(\phi_k^*, \xi^{\mu_1}, \dots, \xi^{\mu_r}))$ be the solution of

$$\min_{\Phi \in \mathcal{P}^{\times r}(B^{-1}(\tau^{\mu\beta}))} \sum_{i=1}^k \|\Delta W_i(\phi_i, \xi^{\mu_1}, \dots, \xi^{\mu_r})\|^q$$

5 Increment W by $\Delta W(\phi^*, \xi^{\mu_1}, \dots, \xi^{\mu_r})$

6 Go back to 1.

Hence for each combination of desired internal representations, the minimal necessary weight changes at each neuron are computed. Each weight change is measured by the euclidean norm and summed up using the q -th power. This is minimized over all combinations Φ . By choosing $r = 1$ one recovers the Madaline I algorithm.

3.2. Experimental results

Computer simulations within the same setting as initially described showed an 20% improved performance. Fig.2 shows the classification capacity as a function of r , and shows that a value of $\alpha \approx 3.3$ is reached as compared to the best value of 2.8 of Table 1.

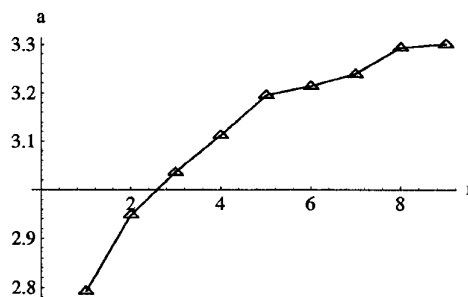


Figure 2. Class. capacity as a function of r ($a = \alpha$)

One might suspect that the minimization over the internal representations is very costly, which is true but by using a depth-first-search together with a branch-and-bound method, the computer time can be cut efficiently. As can be seen in Fig.3, using $r = 5$ instead of one pattern does not increase the computing time at all. This result applies to $k = 3$ hidden neurons and since the number of possible combinations grows exponentially with k , it further was necessary to bound the branch-width to a value of $10 - 20$, since in rare situation the branch-bound method is not efficient enough. This happens especially in the beginning when all weights are initialized with zero or are close to zero.

4. THE CLASSIFICATION OF HANDWRITTEN DIGITS

For our experiments we used with 18468 handwritten digits from the CEDAR CD-ROM. The preprocessing of the bitmaps was either a grey scale approximation with fixed resolution or a Hough transformation with respect to lines and circles. We classified the digits according to the least

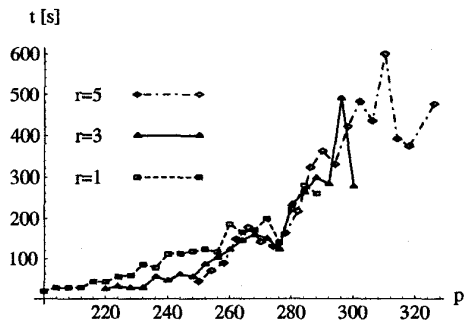


Figure 3. CPU-time in seconds on a Sparc-10 for successful runs as a function of the number of patterns p

significant bit and the results for the classification capacity confirm the improvement of 20% as can be seen in the following table.

r	Gray scale		Hough transf.	
	p_{cr}	α	p_{cr}	α
1	910	8.42	760	7.24
3	1020	9.44	870	8.29
8	1070	9.91	925	8.81
10			950	9.05

Class. capa. of the MadaTron alg. ($n = 36, k = 3$)

Within this setting the grey scale approximation was superior to the Hough transformation. By increasing the number of inputs the Hough transformation proved to be superior. The results on the generalization ability of selected algorithms confirm that the classification capacity is roughly proportional to the generalization ability, as the following table shows.

algorithm	Error rate
standard Backprop	9.0%
Quickprop	8.5%
Madaline I	7.0%
Madatron	5.2%

5. CONCLUSION

We compared different learning algorithms for feedforward neural networks with respect to their classification capacity. To our surprise the simple Madaline I algorithm showed the best performance. By suitably interpreting the principle of minimal weight disturbance we were able to further improve these results, without sacrificing computer time. These results were verified on the real life application of handwritten digit recognition. It would be very interesting to evaluate more algorithms with respect to their classification capacity, since this clearly would shed more light on the merits and performance of algorithms.

By suitably improving our system, we were able to obtain an error rate of 2.9% on the data of the NIST competition [3], which is one of the best results reported so far. This shows, that although the "committee machine" is a very

simple architecture using hard thresholds, it is still very competitive and a serious alternative to networks based on sigmoid non-linearities. This also implies that the principle of minimal weight disturbance is an alternative to the error function approach and gradient descent techniques many learning algorithms rely on.

REFERENCES

- [1] T. Matsui, T. Tsutsumida, and S. N. Srihari. Combination of Stroke/Background and Contour-direction Features in Handprinted Alphanumeric Recognition. In *4-th Int. Work. on Frontiers in Handwritten Reco.*, pages 87-96, 1994.
- [2] Peter Nachbar. *Robuster Entwurf neuronaler Netze*. PhD thesis, Technical University of Munich, 1993.
- [3] R. Allen and J. Geist et al., editors. *The first Census OCR Systems Conference*. National Institute of Standards and Technology, 1992.
- [4] A. Engel, H.M. Köhler, F. Tschepke, and A. Zippelius. Storage capacity and learning algorithms for two-layer neural networks. *Phys.Rev.A*, 45(10):7590-7610, 1992.
- [5] B. Widrow. ADALINE and MADALINE - 1963. In *Proc. of the First Int. Conf. on Neural Networks*, volume 1, pages 143-158, San Diego, 1987.
- [6] B. Widrow and M.A. Lehr. 30 years of Adaptive Neural Networks: Perceptron, Madaline and Backpropagation. *Proc. of the IEEE*, 78(9):1415-1440, 1990.
- [7] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. *Parallel Distributed Processing*, volume 1. MIT Press, Cambridge, MA, 1986.
- [8] J. Hertz, A. Krogh, and R. G. Palmer. *Introduction to the Theory of Neural Computation*. Addison-Wesely, 1991.
- [9] V.N. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer, Berlin, 1982.
- [10] M.E. Hoff. *Learning Phenomena in Networks of Adaptive Switching Circuits*. PhD thesis, Stanford, Stanford Electronic Labs., July 1962.
- [11] P. Werbos. *Beyond Regression: New Tools for Prediction and Analysis in Behavioral Sciences*. PhD thesis, Harvard University, 1974.
- [12] S.E. Fahlmann. Fast-learning variations on backpropagation: An empirical study. In *Proce. of the 1988 Connectionist Models Summer School*, pages 38-51. Morgan Kaufmann, 1988.
- [13] Marc Mezard and Jean-Pierre Nadal. Learning in feedforward layered networks: the tiling algorithm. *J.Phys.A*, 22:2191-2203, 1991.
- [14] D. A. Karras and S. J. Perantonis. An Efficient Constrained Training Algorithm for Feedforward Networks. *IEEE Trans. Neural Networks*, 6(6):1420-1434, 1995.
- [15] P. Nachbar, J.A.Nossek, and J. Strobl. The generalized adatron algorithm. In *Proc. of the ISCAS*, pages 2152-2156, Chicago, 1993. IEEE.

COMPARISON OF LEARNING ALGORITHMS FOR
FEEDFORWARD NEURAL NETS

J. A. Nossek, P. Nachbar and A. J. Schuler

Institute for Network Theory and Circuit Design
Technical University of Munich
Germany, e-mail: nossek@nws.e-technik.tu-muenchen.de
Neural Network Systems, Inc.
86316 Friedberg, Germany

Several well known learning algorithms for feedforward two-layer neural nets and an improved version of Madaline I have been investigated and compared with respect to learning effort and classification capacity. These results, based on random training patterns, and their significance for generalization have been verified with real life data for ICR/OCR.