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. 13-18

On Unlearnable Problems –Or– A Model for Premature Saturation in Backpropagation Learning

Christian Goerick and Werner von Seelen
Institut für Neuroinformatik, Ruhr-Universität Bochum
44780 Bochum, Germany
email: goerick@neuroinformatik.ruhr-uni-bochum.de

Abstract. In this paper we investigate the learning of an unlearnable problem and how this relates to the premature saturation of hidden neurons in error backpropagation learning. General aspects of our model are discussed. A sketch of the derivation of equations for the development of the significant weights in time is given.

1. Introduction

The phenomenon of *premature saturation* of hidden neurons in feedforward neural networks trained by error backpropagation learning has repeatedly been reported by different researchers [2]. Different approaches have been proposed to circumvent this severe problem that can prevent proper learning. In [4] it is stated that the saturation is due to improperly chosen initial weights, where improper is to be regarded with respect to network parameters. We show that the relationship between these network parameters and the data to be learned is the major effect leading to the undesirable growth of some weights. Therefore we will suggest and discuss a model for an extremly difficult learning task, relate it to backpropagation learning and then sketch the derivations of equations for the weight development during saturation.

2. The Model

During many experiments reported elsewhere [1], we could observe that the probability for premature saturation depends on the relationship between the network parameters and the data, with which the network is to be trained. Especially when a problem is difficult to learn for a network (which does not imply that the chosen configuration is not well suited to accomplish the task), saturation can be observed. An extreme task that can never be learned by any deterministic network are *statistically independent* input and target data. In a stochastic framework this is stated as

$$f_{XY}(x,y) = f_X(x) f_Y(y), \tag{1}$$

where f_{XY} denotes the joint density function of the input random variables X and the target random variable Y. We consider this case as the limit of increasing complexity and as a suitable model for the saturation phase. Our further reasoning will be based on this assumption. In section 3. it will be shown, that this model implies the convergence towards a constant plateau of the surface of the expected squared error between the target and the output value of the network. This behaviour exposes certain similarities to the one reported for feedforward neural networks during the saturation phase. They are known to be on a saddle point of the error surface during saturation [2]. Furthermore, it can be observed that during the learning of complex tasks (i.e. parity bit problems involving many bits), the correlation coefficient between the target and the current output of the network is approximately zero (in our cases 0.00012). This is a strong hint for the independence of the output and target values.

For real data, an additional tendency towards the stated behaviour can be induced by randomly chosen initial weights. The point of proper initialization will not be addressed in this paper. At this point we would like to emphasize that this assumption is only valid at the onset of learning where the saturation usually occurs. For continued learning it must be immediately dropped.

3. General aspects

In our stochastic framework the problem of finding a function g that maps some input data x onto some target data y is commonly stated as the minimization of the expected squared difference between the desired output y and the corresponding value of $g(x)$, i.e.

$$\min \quad \left(E\left\{(g(x)-y)^2\right\}\right). \tag{2}$$

Using our independence assumption (1) this term can be rewritten as

$$\begin{aligned} E\left\{(g(x)-y)^2\right\} &= E\left\{g(x)^2 - 2g(x)y + y^2\right\} \\ &= E\left\{g(x)^2 - 2g(x)y\right\} + \sigma_Y^2 + E\left\{y\right\}^2 \\ &= E\left\{\left(g(x) - E\left\{y\right\}\right)^2\right\} + \sigma_Y^2, \end{aligned} \tag{3}$$

where σ_Y^2 denotes the variance of y defined by $\sigma_Y^2 = E\left\{\left(y - E\left\{y\right\}\right)^2\right\}$. This implies that a minimum is achieved for $g(x) \equiv E\left\{y\right\}$, i.e. that the expected value of the output data will be learned and that the remaining expected error is determined by the variance of the output variable, provided that g can assume a constant value for all inputs. This result is solely based on the independence assumption and the chosen error criterion. It is valid for all possible deterministic maps g and minimization methods. A similar reasoning for vector valued functions can be given.

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. 13-18

4. Backpropagation

The following work is a case study of a single hidden layer feedforward neural network trained by the error backpropagation algorithm, where the so called on-line mode is chosen. For a comprehensive stochastic *ansatz* for the investigation of the learning behaviour of neural networks see [3]. Our architecture and nomenclature is the following. The variables s_k, s_j, s_i denote the activations of the input, hidden and output layer respectively, w_{ij} (w_{jk}) the weights between the hidden neuron j (input neuron k) and the output neuron i (hidden neuron j). The activations are computed according to the equations

$$s_j = \varphi(\sum_k w_{jk} s_k) \qquad s_i = \varphi(\sum_j w_{ij} s_j) \qquad \varphi(x) = \frac{1}{1+e^{-x}}.$$

In the following, the superscripts n and $n+1$ denote the values of variables at time steps n and $n+1$. Using the well known derivation of the weight updates per time step the resulting equations are

$$w_{ij}^{n+1} = w_{ij}^n + \alpha e_i^n (\varphi_j')^n s_j^n, \tag{4}$$

$$w_{jk}^{n+1} = w_{jk}^n + \alpha (\varphi_j')^n s_k^n \sum_i e_i^n (\varphi_i')^n w_{ij}^n \tag{5}$$

where the error $e_i = y_i - s_i$ is used and the φ' are defined by

$$\varphi_j' = \varphi'(\sum_k w_{jk} s_k) \qquad \varphi_i' = \varphi'(\sum_j w_{ij} s_j) \qquad \varphi'(x) = \varphi(x)(1-\varphi(x))$$

Many experiments have been conducted using this architecture and independent, identically and equally distributed input variables

$$S_k \sim f_{S_k}(s_k) = \begin{cases} 1, & 0 \le s_k \le 1 \\ 0, & else \end{cases} \tag{6}$$

and target values $Y_i \sim f_{Y_i} = f_{S_k}$. The results of these experiments lead to reduced forms of the equations (5) and (4) as well as to separate approaches for the treatment of the dynamics of the weights of the hidden and the output layer. The following derivation is very condensed and does not show all the intermediate steps due to space limitations of this paper.

4.1. Input-Hidden weight dynamics

The reduced form of equation (5), that we now want to investigate, is given as

$$W_{jk}^{n+1} = W_{jk}^n + \alpha \varphi_j' (W_{jk}^n S_k + b) S_k A \tag{7}$$

for fixed k and j not equal to zero. The term A models the influence of the weights w_{ij}, the error e_i and the derivative of s_i. The constant b replaces the influence of the neglected weights w_{jk} and activations s_k and can be used for

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. 13-18

an analysis of the structural stability of the results. Due to the independence of the inputs S_k the examination of only one weight should be sufficent for a first order approximation. For the random variable A the existence of $f_A(a)$, the expected value $\mu_A \neq 0$ and the variance σ_A^2 is assumed. The assumptions for these properties of A are justified by the very short relaxation time scale of the weights w_{ij} as stated in section 4.2.. A more complex model is subject to current research. To analyse the equation (7) we consider it as an iteration equation of a first order markov process, i.e. $w^{n+1} = f(w^n, s, a)$ where we dropped the subscripts of the weight. The conditional density function of the transition probabilities is then given as

$$p(w^{n+1}|w^n) = \iint_{-\infty}^{\infty} \delta(w^{n+1} - f(w^n, s, a)) f_S(s)\, ds\, f_A(a)\, da, \tag{8}$$

where in our case

$$f(w^n, s, a) = w^n + \alpha\varphi'(w^n s + b)sa \tag{9}$$

holds. Then the probability density function for the state of w^{n+1} is given as

$$p(w^{n+1}) = \int_{-\infty}^{\infty} p(w^{n+1}|w^n)\, p(w^n) dw^n \tag{10}$$

$$= \iiint_{-\infty}^{\infty} \delta(w^{n+1} - f(w^n, s, a)) f_S(s)\, ds\, f_A(a)\, da\, p(w^n) dw^n \tag{11}$$

Our goal is to establish an equation for the development of the expected value of the ensemble of the w^n's in time. Therefore, we compute the expected value of the preceeding equation giving

$$E\left\{w^{n+1}\right\} = \int_{-\infty}^{\infty} w^{n+1}\, p(w^{n+1})\, dw^{n+1} \tag{12}$$

$$= \int_{-\infty}^{\infty} \left[w^n + \alpha\mu_A \left(\frac{1}{w^n}\varphi(w^n + b) - \frac{1}{w^{n2}} \ln\left(\frac{e^{w^n+b}+1}{e^b+1} \right) \right) \right] p(w^n)\, dw^n. \tag{13}$$

The result is obtained by using equation (11) and performing the w^{n+1}, a and s integration. It can be rewritten as

$$E\left\{w^{n+1}\right\} - E\left\{w^n\right\} = \alpha\mu_A \int_{-\infty}^{\infty} \left[\frac{1}{w^n}\varphi(w^n + b) - \frac{1}{w^{n2}} \ln\left(\frac{e^{w^n+b}+1}{e^b+1} \right) \right] p(w^n)\, dw^n \tag{14}$$

$$= \alpha\mu_A E\left\{h(w^n)\right\} \tag{15}$$

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. 13-18

with

$$h(y) = \frac{1}{y}\varphi(y+b) - \frac{1}{y^2}\ln\left(\frac{e^{y+b}+1}{e^b+1}\right) \tag{16}$$

Keeping in mind that the first order approximation of $E\{h(w^n)\}$ is given by $h(E\{w^n\})$ [5], equation (15) can be approximated by

$$E\{w^{n+1}\} - E\{w^n\} = \alpha\mu_A h(E\{w^n\}), \tag{17}$$

which is a nonlinear difference equation for the expected value of w^n. Its continuous time equivalent is given by

$$\dot{\mu}_W = \alpha\mu_A h(\mu_W), \tag{18}$$

where μ_W denotes $E\{w(t)\}$. To gain an approximate solution, the *ansatz* $\mu_W(t) = \alpha t^\beta$ is used. For small t the solution is given by

$$\mu_W(t) \propto t \tag{19}$$

and for larger t it is given by

$$\mu_W(t) \propto \sqrt[3]{t} \tag{20}$$

This means, that the expected value of the weights between the input and the hidden layer is continuously increasing with time, which directly leads to the saturation of the hidden neurons. The behaviour described by the last two equations fits the real behaviour of weights in time sufficiently well. Corresponding equations for the variance are subject to current work.

4.2. Hidden-Output weight dynamics

For the weights between the hidden and the output layer an equivalent modeling and analysis can be performed. One of the results is that the expected value of the weights converges towards a fixed point. After a high convergence rate for the first steps, the speed settles to an almost constant low rate. This is the reason for the choice of a time invariant density for A in equation (7).

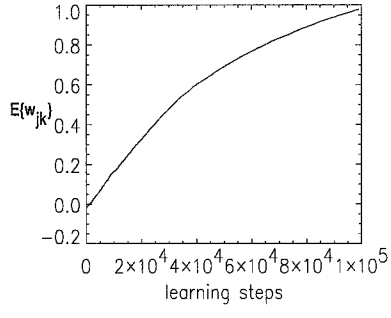

Fig. 1. Development in time, hidden weights

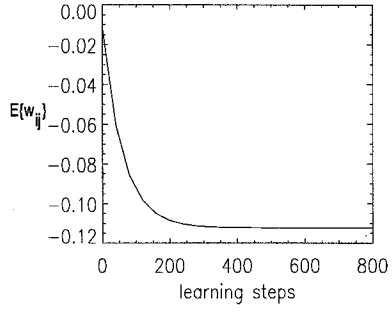

Fig. 2. Development in time, output weights

Figures (1) and (2) show the qualitative development of the expected value of a weight of the hidden layer and of the output layer in time.

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. 13-18

4.3. Comparison to the general case

From these results we can conclude that the problem of learning the expected value of the target data is solved by the network by driving the hidden neurons into saturation and mapping their constant activations statically to the desired output values. This solution is preferred to zeroing all output weights except for the ones linked to the bias, what represents a second possible solution for the problem. A possible explanation for this behavior are the different volumes of the possible solutions in weight space for the first and the second solution, i.e. there are more possibilities to realize solutions of the first kind than of the second kind. A third possibility is the zeroing of all weights linked to input units except for the bias and mapping the constant activities of the hidden units statically to the output units. This solution was never observed. An explanation for this would be the faster dynamic of the output weights.

Experiments have shown that in the case of linear activation functions no saturation occurs when independent data are presented. In this case the second solution was always realized due to the lack of the reachability of saturated hidden neurons.

5. Conclusions

In this paper we investigated the "learning" of statistically independent data. The problem was related to the phenomenon of *premature saturation* of hidden neurons in error backpropagation learning. We have considered general aspects of our model for a class of learning algorithms and sketched the derivation of equations for the development of the weights between the input and the hidden layer of a single hidden layer network in time. The results may suggest that some problems exist, that cannot be learned in practice by the chosen architecture and minimization algorithm due to the saturation of the hidden neurons induced by the complexity of the task to be learned. The proposed model seems to be a promising foundation for continued investigations of problems related to complex learning tasks.

References

[1] Christian Goerick. Eine Strukturkennzahl zur Untersuchung der Lerndynamik von vorwärtsgekoppelten neuronalen Netzen, IRINI 94-08. Technical report, Institut für Neuroinformatik, Ruhr-Universität Bochum, 1994.

[2] Simon Haykin. *Neural Networks, A Comprehensive Foundation.* MacMillan Publishing, 1994.

[3] Tom M. Heskes and Bert Kappen. Learning processes in neural networks. *Physical Review A*, Vol. 44:pp. 2718–2726, 1991.

[4] Youngjik Lee, Sang-Hoon Oh, and Myung Won Kim. An Analysis of Premature Saturation in Back Propagation Learning. *Neural Networks*, Vol. 6:pp.719–728, 1993.

[5] Athanasios Papoulis. *Probability, Random Variables and Stochastic Processes.* McGraw-Hill International Editions, 1989.