Regularizing Stochastic Pott Neural Networks by Penalizing Mutual Information

G. Deco and T. Martinetz
Siemens AG, Corporate Research and Development, ZFE ST SN 41
Otto-Hahn-Ring 6, 81739 Munich, Germany

Abstract

In this paper we present a method for eliminating overtraining during learning on small and noisy data sets. The key idea is to reduce the complexity of the neural network by increasing the stochasticity of the information transmission from the input layer to the hidden-layer. The architecture of the network is a stochastic multilayer perceptron the hidden layer of which behaves like a Pott-Spin. The stochasticity is increased by penalizing the mutual information between the input and its internal representation in the hidden layer. Theoretical and empirical studies validate the usefulness of this novel approach to the problem of overtraining.

1.0 Introduction

Two principal complications appear when a neural network is trained for extracting the underlying structure in the data of a real world problem. Firstly, the data is noisy, and secondly, there is usually only a finite amount of training patterns. These two properties of the training data set make it difficult for a neural network to learn only the useful structure and ignore the particularities of each training data, e.g., the noise. Learning particularities like noise leads to overtraining and bad generalization capabilities. Overtraining can be avoided by employing regularization techniques. In learning with neural networks this is usually done by controlling the complexity of the network based on penalty terms (Hinton 1986; Weigend et al., 1991). These penalty terms reduce the effective number of parameters, which, however, still causes a number of problems (Deco et al, 1993; Nowlan and Hinton, 1991).

The present work introduces an alternative approach which regulates the complexity of the network not by reducing the effective number of parameters but by increasing the stochasticity of the data representation. Nowlan and Hinton use a similar but different approach. They insert noise on each weight of the network and control the amount of noise in order to regularize the model (Nowlan and Hinton, 1991). In the present work we use a stochastic network and control the stochasticity of the network by reducing the mutual information between the input and its internal representation. The motivation of this approach is the fact that most of the information describes the noise and should not be transmitted from the input to the hidden layer.

2.0 The Neural Network Architecture

The first layer is just to represent the input data ξ_i^a, with ξ_i^a denoting pattern a of dimension n. The second layer is a layer of m probabilistic Boolean hidden units S_j; i.e., each hidden unit can have the discrete output value 1 with probability P_j and the discrete output value 0 with probability $(1-P_j)$. We choose P_j such that the hidden layer represents a Pott spin, i.e.,

$$P_j = \frac{e^{(\omega_j \cdot \xi^a)}}{\sum_k e^{(\omega_k \cdot \xi^a)}} \tag{2.1}$$

(Peterson and Soederberg, 1989). The output layer is given by a set of T neurons with linear activation functions. The mean output values of the network are then given by

$$O_i^a = \sum_j W_{ij} P_j^a . \tag{2.2}$$

The mean output values O_i^a are used to establish a continuous input-output mapping $\xi^a \rightarrow O_i^a$. In order to learn this input-output mapping, we train the stochastic network such that the squared error between the desired outputs and the mean output values O_i^a is minimized.

To reduce the complexity of the network we increase the stochasticity in the internal representation of the input patterns. This is achieved by reducing the amount of information conveyed from the input layer to the hidden layer, i.e., by reducing the mutual information between the input and the internal representation. Shannon defined the mutual information as the amount of information transmitted in a stochastic channel. In our case the stochastic channel lies between the input layer and the hidden layer of our network and is defined by the Pott probability function (2.1). The mutual information M between input layer and hidden layer is given by

$$M = \sum_a p(a) \sum_j P_j^a \log\left(\frac{P_j^a}{\sum_a p(a) P_j^a}\right) . \tag{2.3}$$

We add the mutual information M as penalty term to the quadratic cost function, obtaining

$$E = \sum_a \sum_i (Y_i^a - O_i^a)^2 + \lambda M \tag{2.4}$$

with λ as a Lagrange multiplier and Y_i^a as the desired outputs. The network learns the training data, and at the same time the penalty term avoids the excessive transmission of information, i.e., information which might describe the noise. Note that by minimizing the mutual information we are increasing the stochasticity of the network. The gradient descent learning rule which corresponds to the quadratic cost function (2.4) can easily be derived. After some algebra we obtain

$$\Delta W_{ij} = \eta \sum_a (Y_i^a - O_i^a) P_j^a \tag{2.5}$$

$$\Delta \omega_{ji} = \eta \sum_a \xi_i^a \sum_{l,k} (Y_l^a - O_l^a) W_{lk} (P_j^a \delta_{kj} - P_j^a P_k^a) \tag{2.6}$$

$$-\lambda \eta \sum_a p(a) \sum_k (P_j^a (\delta_{kj} - P_k^a) \xi_i^a) \log\left(\frac{P_k^a}{\sum_a p(a) P_k^a}\right)$$

with η as the learning step size.

3.0 Simulations

In this section we present the result we obtained by applying our model to a synthetic data set. For demonstrating the performance of our approach we choose a common benchmark from the statistic community. The benchmark, which was introduced by Friedman (1991), is a function of ten variables and is given by

$$f(x_1, \ldots, x_{10}) = 0.1 e^{4x_1} + \frac{4}{(1 + e^{-20(x_2 - 0.5)})} + 3x_3 + 2x_4 + x_5 \quad . \qquad (3.1)$$

This function has a nonlinear additive dependence on the first two variables, a linear dependence on the next three, and is independent of the last five variables (pure noise). Random values within the unit hypercube were chosen for the ten variables x_i. Then the corresponding response values were calculated according to

$$Y_t = f(\underline{x}_t) + \nu_t \quad , \quad 1 \le t \le N, \qquad (3.2)$$

with ν randomly generated from a standard normal. The signal to noise ratio is 3.28 so that the true underlying function accounts for 91% of the variance of the response. Two data sets, one for training and one for testing, were generated using equation (3.1) and (3.2), with 100 and 300 data points, respectively. The network architecture consisted of 10 inputs, 15 hidden units and one output. The learning step size was η = 0.01, and the Lagrange multiplier had a value of λ = 1.

Without penalty term the neural network learns the noise and the spurious dependence on the last five variables, which leads to overtraining and a very bad generalization. Figure 1 shows the evolution of learning with and without the mutual information regularizer. Without the mutual information regularizer the typical overtraining occurs. After adding the mutual information regularizer to the cost function the overtraining disappears and the error on the test set remains asymptotically constant. This indicates that the deterioration of the generalization due to the learning of noise (real or semantic) is now avoided by limiting the amount of information transmitted from the input layer to hidden layer.

Table1 shows the average relative variance (arv) calculated on the test set after training with and without mutual information penalty term. We see that the mutual information penalty term leads to a significant reduction of the generalization error.

4.0 Discussion

In the neural network architecture two regularization effects are incorporated: First, the mutual information penalty term stops excessive decorrelation between the hidden units, which usually takes place in multilayer perceptrons trained with backpropagation learning rules. Backpropagation activates all the resources of the network for learning the training set, which leads to a strong decorrelation between the hidden units. However, by decorrelating the hidden units also the noise is learnt. The mutual information penalty term acts as an entropy that tries to uniform the activations in the hidden layer, i.e., tries to stop decorrelation. The second regularization effect is provided by the stochasticity of the network. An increase of the stochasticity of the network leads to a reduction of its complexity. The stochasticity is regulated by regulating the transmission of entropy between the input and the hidden layer through the Lagrange multiplier in the cost function.

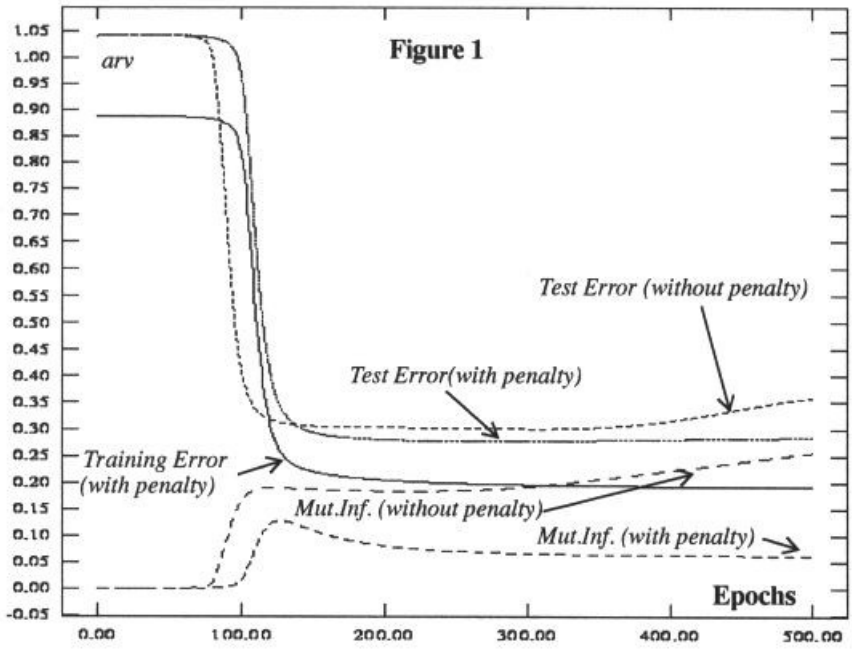

TABLE 1.

Model	arv (test set)
BP-Pott-Net	0.36
BP-Pott-Net with mut. inf.	0.28

References:

- Deco G, Finnoff W. and Zimmermann H.G., 1993, "Elimination of Overtraining by a Mutual Information Network", ICANN'93, Amsterdam, Proc. p. 744-749.
- Le Cun Y., Denker J. and Solla S., 1990, "Optimal Brain Damage", in Proceedings of the Neural Information Processing Systems, Denver, 598-605.
- Nowlan S. and Hinton G., 1991, "Adaptive Soft Weight Tying using Gaussian Mixtures", Neural Information Proccesing Systems, Vol. 4, 847-854, San Mateo, C.A. Morgan Kaufmann.
- Peterson C. and Soederberg B., 1989, "A new method for mapping optimization problems onto neural networks", Int. J. Neural Syst., **1**, 68.
- Weigend A., Rumelhart D. and Huberman B., 1991, "Generalization by weight elimination with application to forecasting", in Advances in Neural Information Proccesing, III, Ed. R. P. Lippman and J. Moody, Morgan Kaufman, 1991.
- Friedman J.H., "Multivariate adaptive regression splines", 1991, Annals of Statistics, **19**, 1-141.