ANNALES UNIVERSITATIS MARIAE CURIE-SKŁODOWSKA LUBLIN - POLONIA

VOL. LV/LVI, 2

SECTIO AAA

2000/2001

Theoretical Physics Chair Maria Curie-Skłodowska University

MAREK JASZUK and WIESŁAW A. KAMIŃSKI*

A new neural network optimization technique with application to nuclear data basis

ABSTRACT

The dependence between the neural network structure and predictive abilities of the network is studied. To optimize the network structure we introduce a method based on the weight scaled backpropagation with a weight decay. We analyze performance of this method using the experimental nuclear data. Neural networks have been trained on a defined set consisting of nuclear mass excesses. We check the predictive power of such learned networks on another set of nuclei not involved in the training procedure. We show that starting with networks consisting of a relatively large number of nodes one can increase their predictive power with considerably reduced complexity of the network topology.

^{*} Uniwersytet Marii Curie-Skłodowskiej, Katedra Fizyki Teoretycznej, ul. Radziszewskiego 10, 20-031 Lublin, Poland.

1. INTRODUCTION

Artificial neural networks have became a very popular tool for solving a wide range of computational problems involving, e.g., classification, optimization and approximation tasks. Despite its popularity, the mostly used feed-forward network architecture with the backpropagation learning algorithm (BP) is suffering from various difficulties. One of them is frequently a low generalization ability. Usually one expects that the neural network (NN) trained on a subset of systematic data will be able to predict the remaining part of the data. However, in many applications the networks generalize quite poorly. Especially the trained network extrapolates systematic tendencies with small accuracy.

One of the methods to overcome this deficiency is to apply network optimization techniques. In general, there are no rules determining how to choose the NN optimal structure for the particular problem. The simplest method is based on construction of networks with different architectures (topologies) and evaluation of their predictive abilities in numerical experiment. But such a procedure is rather inefficient. Another possible approach is to optimize the network structure automatically. In this paper we describe such a technique based on the weight decay (WD) procedure with self-scaling backpropagation (SSBP). The method provides an easy way to reduce the NN structure and to examine a wide range of network topologies from the point of view of their generalization ability. Its details are described in Sec. 2. In the next two sections we discuss simulation procedure and its results. Conclusions are drawn in Sec. 5.

2. SELF-SCALING BACKPROPAGATION AND WEIGHT DECAY

The backpropagation is the most popular learning algorithm for the feed-forward neural networks. Architecture of such networks is defined

a priori and not optimized for a particular problem which has to be solved. In general, the trained network demonstrates poor generalization ability: the errors for the learning set are much smaller than those for the test set.

There exist many modifications of the BP formula allowing to obtain NN with optimal structure. Most of them are based on cutting connections and units considered to be unnecessary ones. After a defined training procedure the final structure of NN is expected to possess larger generalization ability than the starting network. Computer simulations support such an observation.

In our simulations the method based on the weight decay [1] has been used. In this approach an additional term is incorporated into the usually used BP criterion function: besides the normal quadratic error E it consists of the penalty term.

$$E_{d} = E + \lambda \sum_{i,j} w_{ij}^{2} = \sum_{k} (o_{k} - t_{k})^{2} + \lambda \sum_{i,j} w_{ij}^{2}, \qquad (1)$$

where λ is a decay parameter that determines the strength of the penalty term.

Minimizing criterion function (1) with respect to the connection weight w_{ij} leads to the weight change rule in each step of the learning procedure:

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} = \Delta w_{ij} - \varepsilon w_{ij}, \qquad (2)$$

where Δw_{ii} means the weight change for the BP training:

$$\Delta w_{ij} = \eta \frac{\partial E}{\partial w_{ij}}.$$
(3)

 η is the learning parameter and $\varepsilon = \eta \lambda$.

A key idea of the method is to force some particular weights to become very small. Introducing a threshold value all connections with smaller weights are removed, and the unit with input or output connections completely removed is pruned. In the result of such a training procedure the skeletal network emerges.

In this paper we modify the backpropagation algorithm adding the so-called self-scaling term. Such a version of BP (SSBP) displays some desirable features discussed below. We propose to multiply each standard weight change used in the BP approach by its absolute value. So, change of the weight is always proportional to the weight absolute value. With this modification learning formula (3) reads

$$\Delta w_{ij}^{SS} = -\eta \left| w_{i,j} \right| \frac{\partial E}{\partial w_{ij}} = \eta \left| w_{i,j} \right| \left(d_j - y_j \right) \frac{d f(I_j)}{d I_j} x_i.$$
(4)

During the network training procedure some weights are reduced considerably, while others remain relatively large. Combining the SSBP and WD approaches we postulate a new rule of the weight change in the following form:

$$\Delta w_{ij} = -\eta \Big| w_{i,j} \Big| \frac{\partial E}{\partial w_{ij}} = \eta \lambda w_{ij} = \Delta w_{ij}^{SS} - \varepsilon w_{ij} \,. \tag{5}$$

We use the algorithm based on Eq. (5) in our computer experiment described in Sec. 4.

3. DATA USED IN SIMULATIONS

Our simulations were carried out within 1,775 nuclei for which the mass excess is experimentally known [2]. The mass excess is defined as the difference between the atomic mass M(Z, N) (measured in mass units) and the mass number A.

$$\Delta M(Z,N) = M(Z,N) - A, \qquad (6)$$

where Z(N) is a number of protons (neutrons). These experimental data were divided into three different sets. First of them containing 1,413 samples for nuclei with the proton number ranging from 21 to 89 was used in the training procedure. Other two sets were utilized in tests of the predictive ability of the trained networks. They consist of 196 lighter nuclei with the proton number in the range 8–20 and 166 remaining nuclei with the proton number in the range 90–108, respectively. Nuclei with the number of protons below 8 were not used in our calculations due to non-typical large fluctuations of their mass excess.

Each training pattern contained four inputs and one output. As an input we have chosen four parameters characterizing the particular nucleus: the number of protons Z, the number of neutrons N and their parities set 0(1) for even (odd) values, respectively.

4. SIMULATIONS

In the first stage of the learning procedure each network has been trained with the standard BP method until the minimal possible error was obtained for the training data set. Then we used the SSBP algorithm to look for possible different network architecture with approximately the same training error. In Table 1 we compare results of simulations obtained for different topologies of the one- two- and three-hidden layer perceptions. It is worth to note the reduction of the training error of 10%–20% comparing the SSBP and BP approaches as well as some cut of the connection number.

Initial network topology	Weight number after training	Training method	RMSE training set	RMSE test set 1	RMSE test set 2
4-10-10-1	150	BP	0.43	25.2	5.5
4-10-10-1	144	SSBP	0.38	22.6	4.7
4-20-1	100	BP	1.1	47.5	14.1
4-20-1	90	SSBP	0.91	34.8	15.7
4-20-20-1	500	BP	0.35	23.4	11.4
4-20-20-1	495	SSBP	0.27	21.2	16.9
4-10-10-10-1	250	BP	0.33	16.8	16.2
4-10-10-10-1	246	SSBP	0.24	16.5	13.3

Table 1. Results of simulations for the standard backpropagation method (BP) and for the self-scaling backpropagation (SSBP). Different topologies with 4 inputs and 1 output and 1-, 2- and 3-hidden layers are presented

The generalization ability for both algorithms is rather poor. One observes some cut of weights caused by pruning with the threshold value $T = 1 \cdot 10^{-6}$ for the SSBP approach, but the reduction is not noticeable. From the inspection of weight values it follows that in the standard BP weights do not reach values comparable with the threshold.

In the second phase of our simulations the weight decay algorithm was used for the networks trained with SSBP. At the beginning the decay factor λ was chosen sufficiently small for not influencing the network structure considerably. Then the factor was gradually increased. During the computer experiment the learning rate η was held to be 0.2. And each weight becoming smaller than the threshold was removed from the network. In consequence, unit which lost all of its input or output connections was cut off.

We also test another possible procedure for optimization of the NN architecture using the untrained network as a starting network. Our simulations show that such a method is less effective and consuming much more CPU time in the training phase of the computer experiment. The training procedure is also more complicated, because the decay factor makes difficult to find the proper global minimum of the error function. As mentioned by other authors (see, e.g., [3]) this version of the algorithm gives also worse predictive abilities of networks.



Fig. 1. The decay factor λ against the number of weights for different network topologies

Figure 1 shows the degree of networks complexity versus the decay factor λ . One can observe that the weight number of particular network becomes relatively close to each other when the λ parameter grows. It is worth noting that some parts of the curves presented in Figure 1 are horizontal or vertical. For the *plateau* behaviour an increase of the decay factor does not influence the network structure. But exceeding some values of the λ parameter a kind of avalanche in the weight reduction appears. Similar dependence is observed if we look for units of NN.

The dependence between λ and predictive ability of particular networks is shown in Figures 2 and 3. We observe that the best predictivity is achieved by NN for the decay factor about 10^{-5} and does not depend on the particular architecture.



Fig. 2. The dependence between the decay factor λ and generating errors for the training and testing sets in a case of the network with two hidden layers containing 20 units each

Although dependence between predictive abilities and network optimization seems to be not simple, in all investigated cases we were able to obtain more powerful networks than those initially chosen. During the WD process the error for the training set grows constantly and approaches that for the testing sets. The difference between both errors is noticeably small for the λ value mentioned above. Simultaneously, the weight number for different networks varies in the range 17–20, while the unit number ranges between 11 and 17. The reduction is relatively large comparing with the initial networks. So, one can conclude that for the particular problem discussed in this paper large network introduces some amount of unnecessary connections deteriorating proper prediction make by NN.



Fig. 3. Dependence between the decay factor λ and errors for the training and test sets. The network with one hidden layer containing initially 20 neurons is presented

5. CONCLUSIONS

In conclusion, simulations discussed in the paper show that the network architecture chosen *ad hoc* is, in general, not optimal for the specific problem. The SSBP method proposed in this paper offers an easy and effective algorithm for studies of a wide range of network architectures. Our results confirm earlier observations that neural networks with optimized topology and relatively small number of weights are more effective during the generalization procedure than more complicated (larger number of layers and units) ones.

There exists a number of similar methods leading to optimized network topologies. One of them is structural learning with forgetting (SLF) [4]. Authors studied this method in their previous paper [5]. Comparing SLF with the algorithm presented here leads to similar conclusions about the SLF approach. But the predictive power of network trained with SLF seems to be smaller than for NN obtained in the studies presented in this article.

REFERENCES

[1] Werbos P.: *Proceedings of the IEEE International Conference on Neural Networks*, IEEE Press, 1988 343–353.

[2] Audi G., Bersillon O., Bachot J. and Wapstra A.H.: "Nuclear Physics" 1997 A 624, 1–124.

[3] Finnoff W., Hergert F. and Zimmermann H.G.: "Neural Networks" 1993 6, 771–783.

[4] Ishikawa M.: "Neural Networks" 1996 9, 509–521.

[5] Jaszuk M. and Kaminski W.A.: Predictive power of neural networks with structural learning, [in:] Proc. V Conference on Computers in Science, Wroclaw Scientific Society, Wrocław 1998, 445–450 (in Polish).