1999年度日本オペレーションズ・リサーチ学会
秋季研究発表会

Reducing the Hidden Units in Neural Networks by Using Least Square Method

Liu Yingmin   Wu Cang-Pu
Beijing Institute of Technology
Dept. of Automatic Control
Beijing P.R.China

In the applications of neural networks, the problem of determining the number of hidden units is very crucial.A neural network that is too small will not be able to learn the data while one that is just big enough may learn very slowly and be very sensitive to initial conditions and learning algorithms [1].The approach tackling this problem is to train a larger than necessary network,and then remove unnecessary nodes.

The discussion in this paper is contrained to the multilayer feedforward neural networks,but the idea can also be applied to other systems.

When the number of the hidden units to be deleted is known, removing these units among all hidden units is a combinatory problem.If the number is large,to solve the problem is very time consuming. So the method proposed here is a step by step method,that is,the whole procedure includes many steps and in each step only one unit is eliminated.

In each step we first remove a hidden unit according to a rule and then appropriately adjust the weights incoming into the succeeding layer from the layer including the deleted unit except those with the deleted unit so as to preserve the network's performance on the training set. Denote the set of units in the succeeding layer of a hidden unit h by P_h,and the set of units in the preceding layer of h by R_h.Suppose that there are $1, 2, \cdots, p$ training samples,the cardinality of P_h is n,and that the weight comes from unit h to unit i is w_{hi},after removing the hidden unit h,the sum of the squared errors over its succeeding layer and the training set is

$$\sum_{j=1}^{p}\sum_{i=1}^{n}\left[f\left(net_i^j\right)-f\left(net_i^{j'}\right)\right]^2 \tag{1}$$

where net_i^j and $net_i^{j'}$ are the net input of the ith unit in h's succeeding layer on sample j before and after removing h, and $f(\cdot)$ is the activation function of the neurons.Since the training samples is known,and the parameters of the trained neural networks is fixed, (1) can be computed for each hidden unit, representing the importance of the hidden unit h. So,if a hidden unit h attains the minimum of the criteria function (1),h should be removed first.After determining the unit removed according to (1),the weights entering the units in succeeding layer will be adjusted to keep the ouputs of these units remaining fixed. This amounts to requiring that the folowing relation holds

$$f\left(\sum_{k\in R_i} w_{ki}y_k^j\right)=f\left(\sum_{k\in R_i-\{h\}}\left(w_{ki}+\delta_{ki}\right)y_k^j\right) \tag{2}$$

for all $j = 1, 2, \cdots, p$,and $\forall i \in P_h$, where δ_{ki}'s are appropriate adjusting factors to be determined.There are $p \times n$ equations.In general,the number of variables is much less than the one of equations,and $f(\cdot)$ is nonliner,thus the problem formulated forms a nonlinear least square one.

Solving the nonlinear least square problem is very difficult.Consider the case where the activation function is invertable,(2) becomes

$$\sum_{k\in R_i} w_{ki} y_k^j = \sum_{k\in R_i-\{h\}} (w_{ki}+\delta_{ki})\, y_k^j. \tag{3}$$

Simple algebraic manipulations yield

$$w_{hi} y_h^{(j)} = \sum_{k\in R_i-\{h\}} \delta_{ki} y_k^{(j)}. \tag{4}$$

To sum up,in each step,we identify the unit to be removed in the network accroding to (1),and apply an algorithm to find δ_{ki}s that slolves problem (4).The process is iterated until the performance of the reduced network is satisfactory to the designer.

In [3] the rule for detecting the unit to be removed is

$$h = argmin_{h\in V_H} \sum_{i\in P_h} w_{hi}^2 ||\bar{y}_h||^2, \tag{5}$$

where V_H is the set of all hidden units,$||\bar{y}_h||^2$ is the squared Euclid norm of the vector which takes the outputs of h on the samples set as entries. In [3],the rule was derived from a special algorithm solving the least–squares problem called CGPCNE.We could show that (1) becomes (5) when the activation function is linear.

In (4) the remaining weights are adjusted to keep the inputs of the succeeding layer as close as possible to the old ones.Another selection rule can be obtained from projection operations,which makes the weight w_{hi} not appearing in (1) and (5) [2].Suppose that there are m units in a hidden layer,the outputs of each hidden unit h in the layer constitute a vector $\bar{y}_h = (y_h^{(1)}, y_h^{(2)}, \cdots, y_h^{(p)})$. We have m vectors for all m units,$\{\bar{y}_h, h = 1,2,\cdots,m\}$, denoted by Y.At the same time,for the inputs of all n units in the next layer we have n vectors, $\{\bar{x}_i, i = 1,2,\cdots,n\}$,denoted by X. Removing a hidden unit is equevalent to removing a vector from Y, and adjusting the remaining weights to determining the coefficients of the linear combination of the vectors in X.It is obvious that, after the removal of vector $\bar{y}_h$,the best representation of the vector in X in least square sense equals to its projection on the space spanned by the remaining vecters in Y.So

$$\bar{x}_i|(\bar{y}_h|Null(Y-\{\bar{y}_h\})) \tag{6}$$

can be used to evaluate the importance of $\bar{y}_h$ on $\bar{x}_i$,where $\cdot|\cdot$ denote the projection operator, null($\cdot$) the null space. The sum of the norms of all error vectors can be used to measure the importance of the vector $\bar{y}_h$ in Y. The smaller the sum is,the less important $\bar{y}_h$ is. After computing the sum for each $\bar{y}_h$,the vector to be removed can be selected,so the corresponding hidden unit.

We used the proposed method to investigate the relationship between gross domestic product(GDP) and the gross import(GI) and gross export(GE).Since the precondition to utilize the proposed method is that a well trained neural network is available,moreover,the relationship is very complex, some preprocessings had been done.When the samples were provided to train the network,the original data had been nomalized. To evaluate the behavior of the proposed method,three different measures were adopted:1)the number of hidden nodes in systems;2)the PRE on training set;3)the PRE on test set.PRE is the mean value of $|y^j - \hat{y}^j|/y^j$,where y^j and $\hat{y}^j$ denote the target value and the output of the network,respectively.All the neural networks were three-layered feedforward neural networks,with one output unit and two input units. Each sample consisted of one GDP value as target,one GI value and one GE value as inputs corresponding to the same month.The total number of samples is 72. The simulation results showed very promising.

References

[1] R. Reed,"Pruning Algorithms—A Survey",IEEE Trans. Neural Networks,No. 5,vol. 4,1993

[2] S.Y. Kung and Y.H. Hu,"A Frobenius Approximation Reduction Method (FARM) for Determining Optimal Number of Hidden Units", IJCNN 91,Seattle,July 8-12,vol 2,1991

[3] C. Castellano,A.M. Fanelli and M.Pelillo,"An Iterative Pruning Algorithm for Feedforward Neural Networks",IEEE Trans. Neural Networks,vol. 8,No. 3,1997