Architecture Selection for a Multilayer Feedforward Network

Felix ALBU, Adelaida MATEESCU, Neculai DUMITRIU
"POLITEHNICA" University of Bucharest, 1-3 Bd. Iuliu Maniu
felix@pns.comm.pub.ro, mateescu@pns.comm.pub.ro, dumi@pns.comm.pub.ro

Abstract In this paper, a method for pruning hidden neurones is presented and illustrated on voiced-unvoiced-silence detection problem. It is based on the statistical study of the derivatives of the outputs of the model with regards to each hidden neurone[1]. We claim that if the model is not using a particular neurone to estimate its outputs, then the corresponding sensitivities will have a low degree of significance. The aim of this analysis is the selection of an appropriate subset of neurones for each layer, to finally obtain a more parsimonious model.

1. Introduction

This method is based on the analysis of the partial derivatives of the outputs of the model with regards to its inputs. Under certain assumptions and for a particular class of models [2-4] it is possible to prove that when the model converges towards the underlying process, then all the model derivatives also converge towards the derivatives of the underlying process. We use this property to show that the proposed sensitivities are a very useful insights in the dynamics of the underlying process, since they are a measure of the influence input variables have on the output of the process.

In an *n* hidden layer Multilayer Feedforward Network (MFN) each hidden layer has N_k neurones and it is numbered from the input (I or H_0) to the output (O or H_{n+1}). We use three databases during the estimation process (the training set, the validation set and the test set).

Each input and output variable is normalised by linear transformations in order to have zero mean and unit variance. We will use these normalised variables, because the bounds of a variable should not be taken account in its relevance rating.

In function analysis the usual way to study the influence of a variable *x* on a function *f (x,y,z)* is to compute the derivative: $\frac{\partial f(x,y,z)}{\partial x}$. It represents the sensitivity of the output *f (x,y,z)* with the respect to the variable x at the point *(x,y,z).* If this sensitivity is insignificant all over the input space, then *x* is an irrelevant input variable which does not influence the output.

We have only a finite number of samples with known outputs (the learning database or training set) so if *f* is differentiable we'll deal with a vector (for each input variable) of the form: $\left[\frac{\partial f(x_1,y_1,z_1)}{\partial x},\ldots,\frac{\partial f(x_n,y_n,z_n)}{\partial x} \right]^T$. The analysis of the distributions of these sensitivities on the training set

allows the selection of a candidate subset of irrelevant input variables. We will first select the relevant variables of the input layer, and then consider layer H_1 as the input of a (n-1) hidden layer MFN and so on.

The corresponding equations for the MFN structure in this paper are:

$$H_1^j = \tanh\left(\sum_{i=1}^{N_0} w_{ji} I^i\right)$$
$$H_k^j = \tanh\left(\sum_{i=1}^{N_{k-1}} w_{ji} H_{k-1}^i\right) \quad (1)$$
$$O^j = \tanh\left(\sum_{i=1}^{N_n} w_{ji} H_n^i\right)$$

We thus want to compute the derivatives of the outputs O^j with regards to an input variable I^i or to $H_k^i, i = 1,...N_k$. They can be computed recursively as follow [5]:

$$\frac{\partial O^i}{\partial H_k^j} = \tanh'\left(\sum_{p=1}^{N_n} w_{ip} H_n^p\right)\sum_{p=1}^{N_n} w_{ip}\,\frac{\partial H_n^p}{\partial H_k^j}$$
$$\frac{\partial H_\alpha^i}{\partial H_k^j} = \tanh'\left(\sum_{p=1}^{N_{\alpha-1}} w_{ip} H_{\alpha-1}^p\right)\sum_{p=1}^{N_{\alpha-1}} w_{ip}\,\frac{\partial H_{\alpha-1}^p}{\partial H_k^j}, \qquad while\ (\alpha-1) \neq k \quad (2)$$
$$\frac{\partial H_{k+1}^i}{\partial H_k^j} = \tanh'\left(\sum_{p=1}^{N_k} w_{ip} H_k^p\right) w_{ij}$$

All the derivatives are pre-filtered to take off all the derivatives which correspond to an error of more than twice the standard deviation of the model estimation error (computed on the learning database). We compute for each column *i* of derivatives matrix the mean E_i and the standard deviation S_i. If the derivatives corresponding to a neurone is close to zero, then its mean and variance will be small, and the neurone is deleted. To measure this we computed for each layer H_k the vector *C* containing the distance to the origin (0,0) for each neurone:

$$C_i = \left(E_i^2 + S_i^2\right) / max_{j=1,...,N_k}\left(E_j^2 + S_j^2\right) \quad (3)$$

Sometimes is difficult to decide which neurones we should keep and which we discard. We have observed that especially in the presence of high noise and small amount of data [5]. But in this case the least important neurones of each layer are identified, we discard them and we continue the process until finding the parsimonious architecture.

2. Simulations

The example to illustrate this approach is a Voiced-Unvoiced-Silence (V/U/S) detection problem with a MFN structure [6-7]. This classification provides a preliminary acoustic segmentation of speech, which is important for speech analysis. The feature vector for the classification is a combination of waveform features and parametric features.

Additional waveform features are included to enhance the separation in pattern space when spectral information alone is not sufficient for making the classification.

The speech was filtered at 3.5 kHz and digitised using a 12 - bit A/D at a sampling rate of 8 kHz. The digitised signals were further high-pass filtered at 300 Hz by a fourth-order Butterworth digital filter to eliminate low-frequency hum or noise. The digitised speech was preemphasized using a simple first-order digital filter with a preemphasis factor (0.95). Each frame of speech (200 samples) was weighted by a Hamming window.

Our feature vector was a combination of an inverse square-root function applied to the rms energy to limits its numerical range, the zero-crossing rate and 12 inverse sine parameters. The inverse sine parameters were derived the reflection coefficients $k_l(m)$. We can obtain the reflection coefficients from the LP parameters [8]. The inverse sine parameters are given by:

$$\sigma_l(m) = \frac{2}{\pi} \cdot \sin^{-1} k_l(m) \qquad 1 \le m \le 12 \tag{4}$$

The speech database used in our experiment included Romanian digit numbers and other words spoken by three talkers (two males and one female). The speech recordings were pre-processed and were interactively labelled in three sound categories using waveform and spectrographic displays and audio output as feedback. We have formed three equal databases (learning, validation, test database). The output was coded with 0 for voiced frames, 1 for unvoiced frames and -1 for silence frames. The initial 14-40-1 MFN structure had 641 weights, which was an over-parameterised model. Because the plot of influences of the inputs over the output (Fig. 1a) show that the forth, the eleventh and twelfth inputs have an insignificantly importance for the network output we discarded them. Amazingly, the zero-crossing rate which we had considered to be very important in fact wasn't more important that some of the sine parameter elements. The non-linear function of energy was selected with the highest influence.

We present the results by drawing the Average Relative Variance ($ARV = \frac{1}{N\sigma^2}\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2$) against the percentage of weights pruned by the deletion of the irrelevant neurones at the current step since beginning. Finally we obtained an 11-10-1 structure that corresponds to a 79.5% percentage of deletion or 510 weights cut (Fig. 1b).

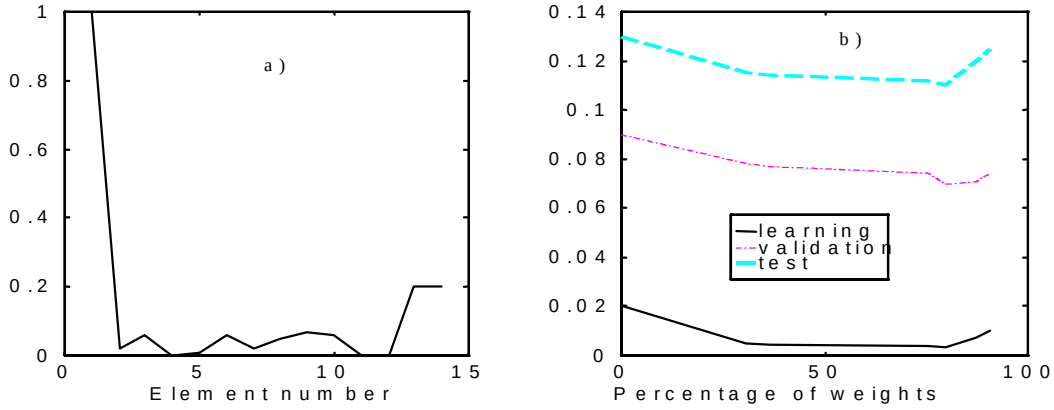

Fig. 1 a) Relative influence of the inputs over the output b) Percentage of weights pruned against the Average Relative Variance computed on the three databases.

3. Conclusion

We have presented a method to prune hidden neurones in a MFN, and have tested it on a toy problem and to voiced-unvoiced-silence detection problem. We have shown that it was possible to preserve the performances on the validation and test set while decreasing the complexity of the model by deleting the non-relevant neurones. Our approach improves the generalisation capabilities by the selection of a more parsimonious model.

REFERENCES

[1] T. Czernichow, Application des réseaux neuronaux à la prediction de séries temporelles, PhD Thesis(in French), Université Paris VI, 1996.

[2] P. Cardaliaguet, G. Euvrard, " Approximation of a Function and its Derivatives with a Neural Network," *Neural Networks*, vol. 5, pp. 207-220, 1992

[3] K. Hornik, M. Stinchcombe, H. White, "Universal Approximation of an unknown Mapping and its Derivatives using Multilayer Feedforward Networks,", *Neural Networks*, vol. 3, pp. 551-560, 1990

[4] A.R. Gallant, H. White, " On learning the derivatives of an unknown Mapping with Multilayer Feedforwards Networks, " *Neural Networks*, vol. 5, pp. 129-138, 1992

[5] T. Czernichow, "Architecture Selection through Statistical Sensitivity Analysis," , ICANN'96, Bochum, 1996

[6] Y. Qi , B. Hunt , " Voiced- Unvoiced- Silence Classifications of Speech Using Hybrid Features and a Network Classifier ", *IEEE Transactions on Speech and Audio Processing*, vol. 1, No. 2 , April 1993

[7] F. Albu, A. Mateescu, "Application of Multilayer Feedforward Network to the Voiced-Unvoiced-Silence Detection Problem", *Proceedings of International Symposium on Communications'96*, Bucharest, Romania, pp. 532-537, November 1996

[8] J. R. Deller, Jr., J. G. Proakis, J. H. L. Hansen, Discrete-Time Processing of Speech Signals Macmillan, New York, 1993