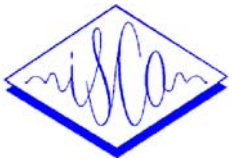

EUROSPEECH '97
5th European Conference on Speech Communication and Technology
Rhodes, Greece, September 22-25, 1997

A MODULAR INITIALIZATION SCHEME FOR BETTER SPEECH RECOGNITION PERFORMANCE USING HYBRID SYSTEMS OF MLPs/HMMs

Roxana Teodorescu, Dirk Van Compernolle and Ioannis Dologlou*

K. U. Leuven - E.S.A.T., Kardinaal Mercierlaan 94, B-3001 Heverlee, Belgium
E-mail: Roxana.Teodorescu@esat.kuleuven.ac.be

ABSTRACT

This paper proposes a novel modular initialization scheme of Multilayer Perceptrons (MLPs) trained for phoneme classification.

Small MLPs are trained to discriminate between a phoneme and all the others. In the next step they are merged using our novel initialization scheme in broad classes and trained further. In the last step we merge the broad phonetic MLPs using the same scheme to generate the final phonetic MLP.

Experiments done on a Dutch language isolated word database showed that the scheme gives faster and better estimates of Bayesian a posteriori probabilities compared to random initialization. Moreover, given its modularity, the method offers the possibility to deal with high dimensional problems.

1. INTRODUCTION

Combining MLPs and Hidden Markov Models (HMMs) in speech recognition systems was proved to be successful [1]. This is due to the powerful dynamic time warping capabilities of HMM and the discriminative pattern recognition power of the MLPs.

In the current systems the MLPs are used either as an approximation of the probability of being in a certain HMM state or as labelers for a discrete HMM [2]. The speech recognition rate of those hybrid models is very dependent on the MLPs training accuracy.

In many current systems a very large two-layered simple MLP consisting of a single large hidden layer (500-4000 hidden units) that receives input from several hundred acoustic variables is used.

Even though the MLPs are a very powerful pattern recognition tool, the training of such networks is a time-consuming procedure that may converge towards a poor local optimum [3]. The search for the global minimum becomes harder and more time consuming as the dimensionality of the network increases.

Earlier work of Waibel shows that networks with a large number of output classes can be trained in a modular, incremental way from networks with fewer number of classes [4].

Our belief is that we can put more structure in the monolithic MLP net, making it modular. Therefore we propose a modular initialization scheme for the MLPs weights.

There exist several approaches to tackle the weights initialization problem of pattern classifying MLPs. Some of these methods are based on selecting small random initial values for the weights according to statistical properties of the inputs trying to match the sigmoid activation function with the empirically detected transition region [5, 6]. Some other methods locate the initial weights in the neighborhood of a good optimum in the weight space by means of standard pattern recognition techniques [7, 8].

Our method determines the vicinity of a good local optimum based on merging previously trained small MLPs. The method reduces the design time and provides better estimates of the Bayesian a posteriori probabilities compared to random initialization. Thus, the advantage of splitting high dimensional problems using the proposed modular approach becomes apparent.

The rest of this paper is organized as follows: Section 2 describes the initialization method. Section 3 presents the experiments and the obtained results of the new initialization method. For that purpose a Dutch language isolated word database has been used. Concluding remarks are given in Section 4.

2. THE INITIALIZATION METHOD

The initialization scheme is illustrated in Figure 1. It consists of the following three steps:

1. In the first step we train 41 small, two-layered, two outputs MLPs to classify between a phoneme and

* Currently with Lernout & Houspie Speech Products.

all the others. The 40 MLPs correspond to the Dutch phonemes and the extra one is trained to discriminate silence from speech.

2. In the second step we merge the small neural networks in broad phonetic classes MLPs and train them further in order to enhance their discriminative power, especially in the case of phonetically close phonemes. The method used to merge the neural networks is described below.

3. In the third step the final phonetic MLP is designed from the broad phonetic MLPs through a similar merging scheme. Furthermore, some additional training iterations are carried out to enhance the design.

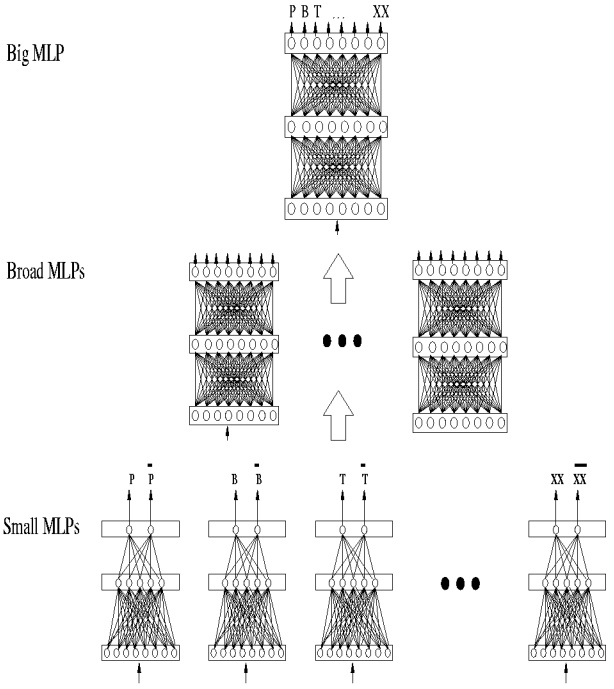

Figure 1: The layout of the initialization method

To merge the already trained small phonetic MLPs in a bigger MLP we use the hidden nodes (weights) of the small ones as illustrated in Figure 2.

More precisely we build the hidden layer of the big MLP by putting together one by one all the hidden nodes corresponding to the small phonetic MLPs. So, the total number of the hidden nodes in the hidden layer of the big net is the sum of the hidden nodes of the small nets.

The inputs of the big MLP are identical to the inputs of the small MLPs.

The outputs of the big MLP correspond to the outputs of the small MLPs, the ones which classify the phonemes. For example in the case of $(P, \bar{P})$, the chosen output is P.

The weights are initialized as follows:

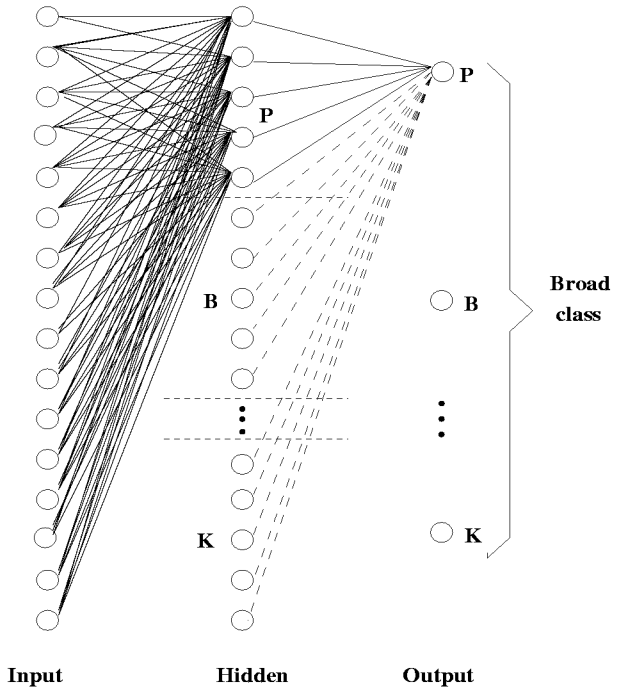

Figure 2: The merging procedure. *In the figure we illustrate the merging procedure for the case of the small MLP classifying between* $(P, \bar{P})$. *Solid line represents the weights that are copied from the small net classifying P. The dashed line represents the weights initialized with small random numbers.*

1. We copy all the input-to-hidden weights of the small phonetic MLPs to the input-to-hidden weights corresponding to the same hidden node of the big MLP.

2. We copy the hidden-to-output weights of the small phonetic MLPs, for the output which classify a phoneme, to the hidden-to-output weights corresponding to the same output of the big MLP.

3. The non-initialized hidden-to-output weights in the big MLP are initialized with small uniform distributed random numbers.

This way we start training the big MLP from a vicinity of a good local optimum. Experimental results show that our algorithm offers a faster convergence to a better optimum compared to the conventional case of random initialization. Hence, the proposed method on one hand reduces the training time and on the other hand offers a flexible modular design.

3. EXPERIMENTS AND RESULTS

3.1. The Database

Testing has been carried out using a Dutch isolated words database called *du32ow*. The database contains 77 isolated words with full phonetic coverage of the Dutch lan-

guage (40 phonemes). We divided the data from 535 speakers into two parts, 431 for training and 104 for testing. The segmentation of the database is done automatically.

3.2. The Input Features

We tested the method using a 39 parameters vector consisting of 12 Mel scale cepstra, their first and second derivatives and 3 energy related parameters, the log of energy, and its first and second derivative. The parameters are mean-normalized.

3.3. MLPs Design and Training

The MLPs used in our experiments are two-layered fully connected feed-forward neural networks. The activation functions of the neurons in the hidden and output layer are sigmoids. The MLPs are trained with the back-propagation algorithm with a momentum term of 0.9.

Experiments showed that 10 hidden nodes for the small MLPs is an optimal choice. Hence, the small phonetic MLPs have 39 inputs, 10 hidden nodes and 2 outputs. The two outputs are trained to discriminate between a phoneme and all the others (see Figure 2).

As initial weights for the small phonetic MLPs we use small random values. Since the learning is faster in the transition region of a sigmoid function than in the saturation regions, we normalize the MLPs inputs to have a variance of 1. The variance is computed for all the 39 input parameters over all the frames in the training set.

3.4. Experimental Results

Our experiments have been carried out using Dutch plosives (6 plosives) and Dutch vowels (13 vowels).

We trained the small phonetic MLPs with balanced input data with respect to the outputs. Once trained, extra training steps are applied with input data corresponding to the same broad class. Thus, a better discrimination between the classified phoneme and all the others of its broad class is achieved. Since we have multiple small and independent neural networks, the training is very fast and can be done in parallel.

In the next step we merge the phonetic MLPs as described in Section 2. Running experiments with different random starting points, we obtained results using two different versions of our initialization method, Initialization 1 and Initialization 2.

In the case of Initialization 1, after the merging step, we train the broad net by updating only the hidden-to-output weights. In the case of Initialization 2 we update all the weights (input-to-hidden and hidden-to-output). In Figure 3 and Figure 4 we present the plosives and vowels

Method	Steps	Training Error	Test Error
Random	900	49.63	50.37
Initialization 1	900	48.08	48.22
Initialization 2	900	46.99	47.05

Table 1: Plosives classification error rates for the training and testing sets.

Method	Steps	Training Error	Test Error
Random	900	72.88	76.95
Initialization 1	900	42.94	43.82
Initialization 2	900	40.93	42.95

Table 2: Vowels classification error rates for the training and testing sets.

classification error rates versus the number of training steps on the training set for all three methods.

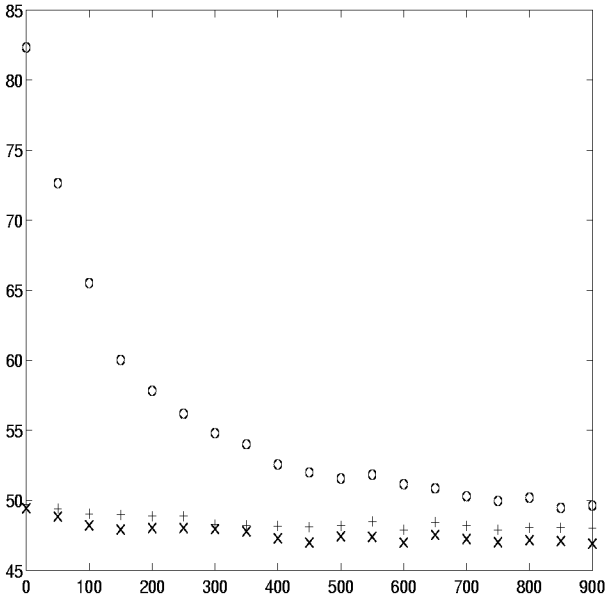

Figure 3: The classification error of plosives. o - Random, + - Initialization 1, x - Initialization 2

From the experimental results we observe that our initialization methods provide faster convergence and better performance than the random one. Initialization 1 is suboptimal because only the last layer is trained. However, the training is more than two times faster than that of Initialization 2 while the classification results differ by only 1%-2%.

With the final MLP in the third step of our initialization method (see Section 2) we expect to obtain even better results since we merge distinct phonetic classes. Thereafter

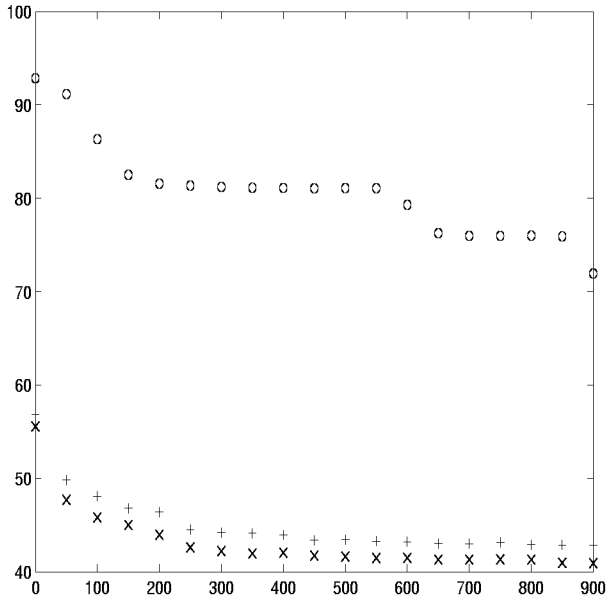

Figure 4: The classification error of vowels. o - Random, + - Initialization 1, x - Initialization 2

only a few training steps (if any) are required to reach the optimum.

In summary our method is better and far more flexible in the sense that the size of the MLPs may be adjusted according to the needs of the application.

4. CONCLUDING REMARKS

Our paper proposes a novel modular initialization scheme of MLPs trained for phoneme classification. Our method reduces the design time and offers better estimates of the Bayesian a posteriori probabilities compared to random initialization. Given its modularity, it is well suited for high dimensional problems. Further research is being carried out to evaluate the performance of the new initialization method with respect to the number of input features as well as the structure of the broad classes.

REFERENCES

1. M.D. Richard and R.P. Lippmann. Neural network classifiers estimate bayesian a posteriori probabilities. *Neural Computation*, 3:461–483, 1991.

2. Ph. Le Cerf, W. Ma, and D. Van Compernolle. Multilayer perceptrons as labelers for hidden Markov models. *IEEE Transactions on Speech and Audio Processing*, 2(1):185–193, January 1994. (Special Issue on Neural Networks for Speech Processing).

3. D.E. Rumelhart, G.E Hinton, and R.J. Williams. Learning internal representations by error propapgation. *Parallel Distributed Processing*, 1:318–362, 1986.

4. A. Waibel. Modular construction of time-delay neural networks for speech recognition. *Neural Computation*, 1:39–46, 1989.

5. H. Leung and V. Zue. Phonetic classification using multi-layer perceptrons. *IEEE International Conference on Acoustics, Speech and Signal processing*, pages 525–528, 1990.

6. J.P. Martens. A stochastically motivated random initialization of pattern classifying mlp's. *Neural Processing Letters*, 3(1):23–29, April 1996.

7. Smyth S. Designing multilayer perceptrons from nearest-neighbor systems. *IEEE Trans. Neural Networks*, 3:329–333, 1992.

8. N. Weymaere and J.P. Martens. On the initialization and optimization of two-layer perceptrons. *IEEE Transactions on Neural Networks*, 5:738–751, 1994.