EFFICIENT COMPUTATION OF MMI NEURAL NETWORKS FOR LARGE VOCABULARY SPEECH RECOGNITION SYSTEMS

Jörg Rottland, Andre Lüdecké, Gerhard Rigoll

Department of Computer Science
Faculty of Electrical Engineering
Gerhard-Mercator-University Duisburg, Germany
e-mail: {rottland,luedecke,rigoll}@fb9-ti.uni-duisburg.de

ABSTRACT

This paper describes, how to train Maximum Mutual Information Neural Networks (MMINN) in an efficient way, with a new topology. Large vocabulary speech recognition systems, based on a Hybrid MMI/connectionist HMM combination, have shown good performance on several tasks [1] and [2]. MMINNs are trained to maximize the mutual information between the index of the winning output neuron (Winner-Takes-All network) and the phonetical class of the corresponding acoustic frame. One major problem of MMI-neural networks is the high computational effort, which is needed for the training of the neural networks. The computational effort is proportional to the input and output size of the neural network and to the number of training samples. This paper shows two approaches, that demonstrate, how these long training times can be reduced with very low or even no loss in recognition accuracy. This is achieved by the use of phonetical knowledge, to build a network topology based on phonetical classes.

1. INTRODUCTION

MMINNs can be used as a high performance vector quantizer (VQ) for a discrete HMM speech recognizer. It can be shown, that this paradigm is optimal for statistical pattern classification in the maximum likelihood sense [3]. The computational effort for the training of the single layer MMINN is proportional to the input and output size of the neural network and to the number of training samples. To reduce the computing time, one could decrease the number of input or output neurons or the number of training data. But by doing this, one would also decrease the mutual information of the network, which leads to worse recognition results. A solution for this problem is to split up this single network into several smaller networks, each for a subset of the phonetical classes (e.g. vowels, etc.) The decision, which of the smaller networks is chosen, is done by another network, which now tries to maximize the mutual information between its output and the phonetical subclasses of the corresponding frame (decision network). Those small networks can be trained much faster, due to their smaller number of parameters. The goal is, that the overall mutual information, of all small networks will be approximately the same as for one big network.

2. BASELINE SYSTEM

The baseline system is a single layer neural network with an input size of 12 times the number of frames looked at. This means for a single frame network the size is 12, for a three frame network the size would be 36. The output size is chosen to 200, to have the same topology as the system in [1]. Figure 1 shows the structure for a single frame network. The network is trained to maximize the mutual information between the label stream Y produced by the network and the corresponding phonetical information W.

$$I(W,Y) = H(W) - H(W|Y) \quad (1)$$

In Eq. 1 $H(W)$ is not affected by the neural network, thus only $H(W|Y)$ has to be minimized in order to maximize $I(W,Y)$. This means, that the loss of information, which will occur because of the quantization error, will be minimized.

$$H(W|Y) = -\sum_{I}\sum_{M} P(w_i, y_m) \cdot logP(w_i|y_m) \quad (2)$$

[3] describes a method how to perform this training with a gradient descent approach using the softmax function in the output layer.

For the training procedure phonetical knowledge is necessary, thus the training data has been aligned with 47 phones. (45 phones plus silence and an optional inter-word silence).

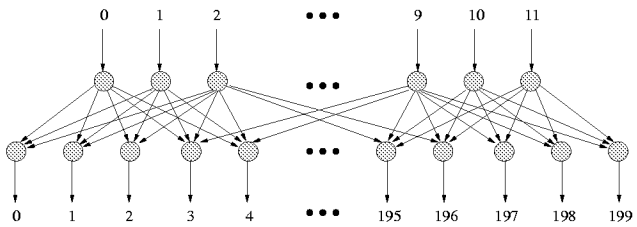

Figure 1. Baseline topology for the MMI neural network, with an input layer size of 12 and an output layer size of 200

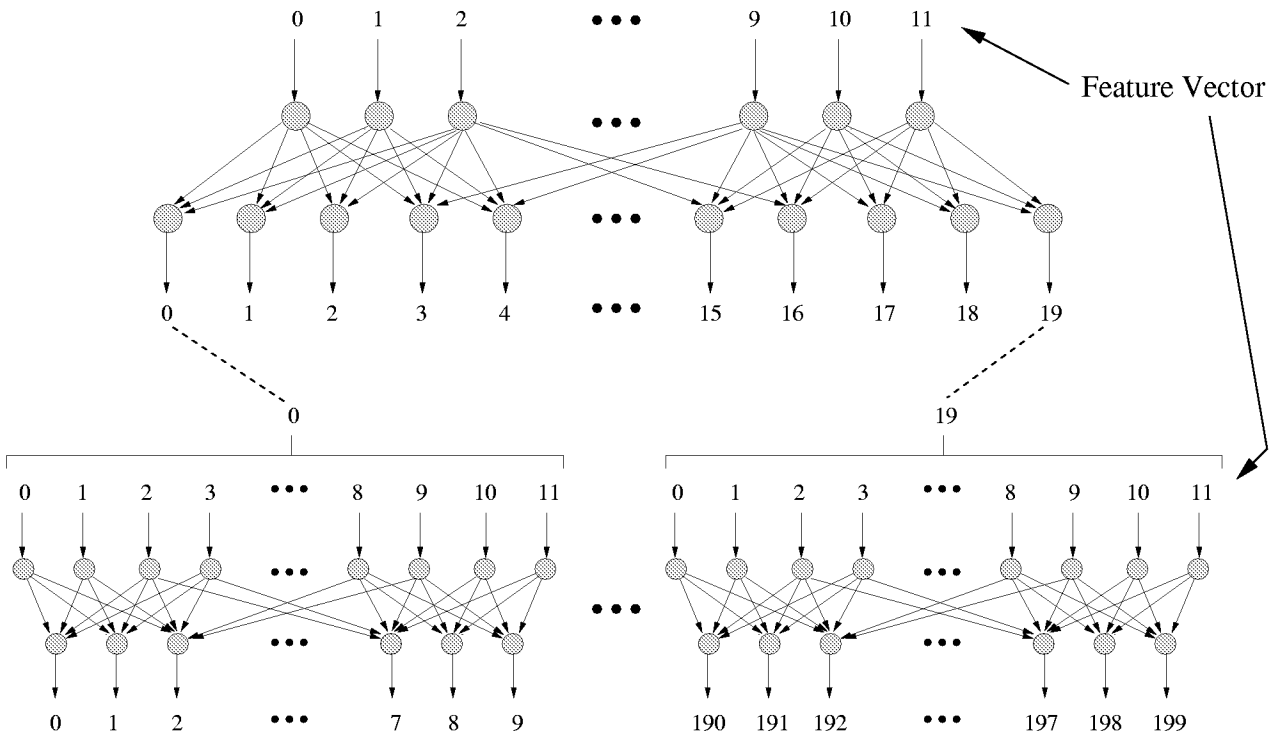

Figure 2. Structure of a flat decision network and the corresponding MMI networks

3. NEW TOPOLOGIES

The goal of a new topology is to split up the network in Fig. 1 into several smaller networks, which can be trained faster, because of their smaller output layer. An additional decision network is necessary to choose, which of the smaller networks will be used.

The idea of this paper is to use phonetical knowledge to create the decision network. This is inspired by the idea, that sounds belonging to the same phonetical class like e.g. vowels can be easily separated from sounds not belonging into this class. In the following sections two different topologies for this decision network and the small MINNs are presented:

3.1. Flat decision network

The topology of the first approach (see Fig. 2) presented here consists of a single layer (flat) decision network. The disjunct phonetical classes used in this approach are vowels, fricatives, glides, plosives, nasals and affricates. Those classes were used to train the decision network. The objective function is to maximize the mutual information between the phonetical classes and the indices of the winning neurons. So this approach is very similar to the baseline approach, with the difference, that here only classes of phones are looked at and at the baseline system makes use of the phones itself. Another difference is the size of the output layer, which is only 10% of the size of the baseline system.

After the training of the decision network, all feature vectors are quantized by the decision network. In a second step, using all feature vectors, which have been quantized to the same class, a new MINN is trained. So for each output neuron of the decision network there will be such a MINN. Only those training data are used for each new MINN, which correspond to the winning neuron of the decision network. Thus each of the new MINNs will only see a fraction of the whole training data. The input of these new networks is the same feature vector, the decision network sees. The sum of the output sizes of all of those second level networks together is the same as the size of the output layer of the baseline system. The output size of those second level networks can be computed as:

$$\frac{\text{output size of the baseline system}}{\text{output size of the decision network}} \tag{3}$$

Another possibility to determine the size of the output layer, is to consider the number of training data each network gets. So for more training data more parameters can be estimated, thus the size of the output layer can be larger. Because the sum of all output nodes has to stay the same, the networks with less training data will become smaller.

For the approach in Fig. 2 the total number of parameters is exactly the same as in the baseline approach, but the computational effort is much smaller, because each of the networks is trained with a fraction of the whole training data. For the case that all networks get the same amount of training data, the computational effort for this approach compared to the baseline system can be computed as:

$$\frac{\text{OL size} \cdot \#\text{of networks} \cdot \%\text{ of training data}}{\text{size of the baseline output layer}} \tag{4}$$

Which equals to the percentage of training data each net-

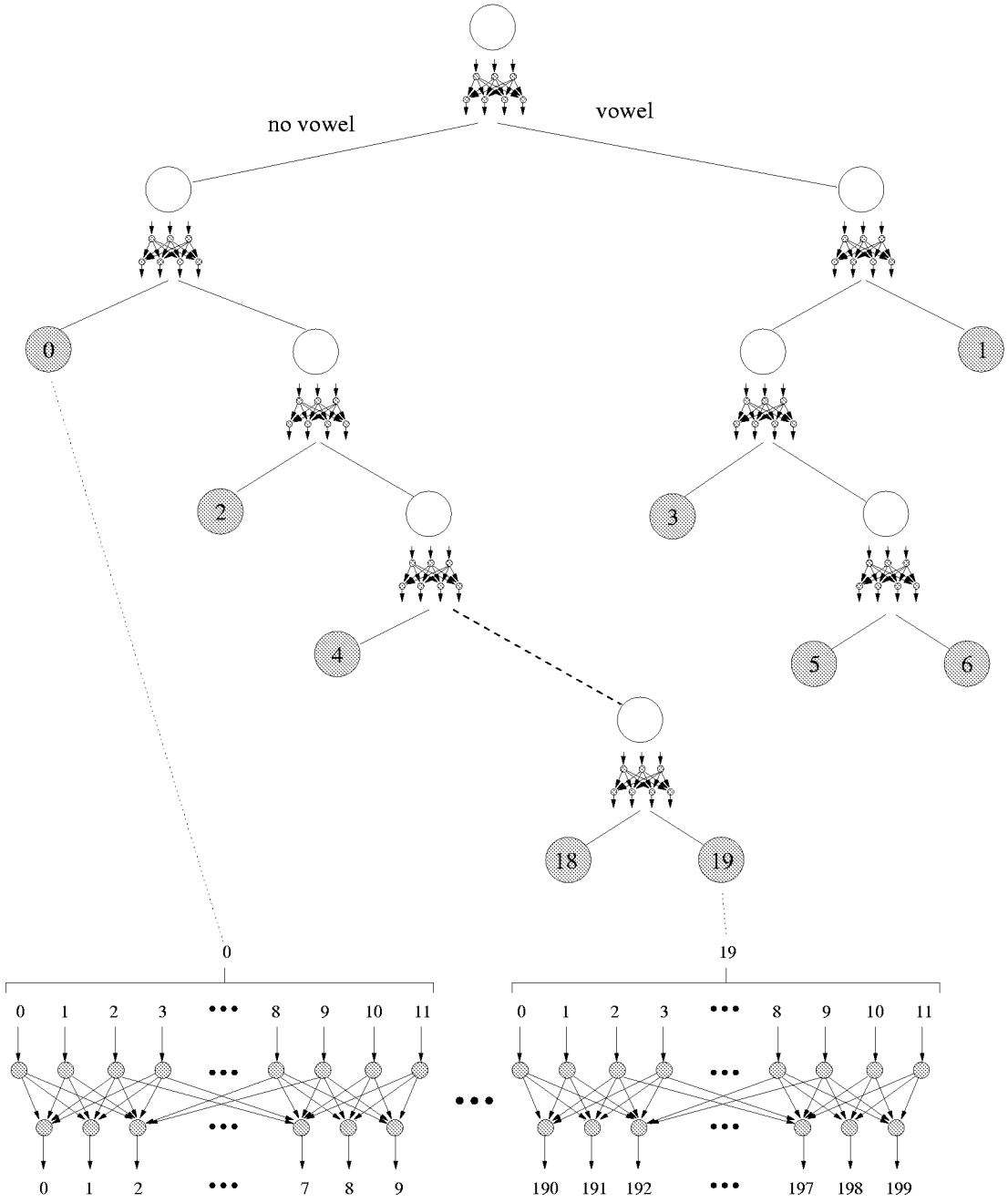

Figure 3. Hierarchical decision tree network

work sees, because the size of the output layer times the number of networks equals to the size of the baseline output layer. So for the case in Fig. 2 the computational effort is 5% of the baseline system plus the training time for the decision network, which is 10% of the baseline training time. In total this gives 15% of the original training time.

3.2. Hierarchical decision network

The second approach is not to choose a fixed number of classes, but to use a phonetical tree, similar to the approaches in [4, 5], with the difference that here the goal is to maximize the mutual information between some phonetical class and the label index of the winning neuron. The basic structure of this approach is like the previous one. There is a decision network, which decides which of the second level networks are chosen. The main difference between this approach and the previous one is the topology of the decision network. Here the decision network has a hierarchical (tree) structure. The growing of the structure itself is self organizing. This is done by using a set of non disjunct phonetical classes. For each node all class-splitting are tested, and the node which gives the best mutual information is split up into two nodes, one containing the members of the found phonetical class and one with

the rest. This procedure is repeated, until the desired number of leaves is reached. With this tree structure one can always find the phonetical class, that will improve the mutual information best. The number of end nodes (leaves) is predefined (as in the previous approach) and for each leave a second network is trained, as in the first approach. An example for this topology is given in Fig. 3.

In each node of the tree there is again a neural network. The size of each network node is dependent on the amount of training data for this node. For this work the first node (root node) has an output layer size of 16.

The training of each network node consists of three steps. In the first step the mutual information between the actual phonetical class and the index of the output neurons is maximized. In the next step one has to determine, which of the output neurons belong to the class and which do not. The decision is made by determining which class gets the majority on each neuron. The third step is to repeat the first two steps with every possible phonetical class, to find the one, which is best in the actual node. This is performed for all nodes, and the node which gives the best value for the mutual information is expanded.

The structure of the second layer network is the same as for the flat topology. Thus the computational effort for the second layer is the same as above. The computational effort for the hierarchical decision tree depends on the number of phonetical classes used. In this approach 100 classes were used. Those were e.g. vowel, front-vowel, fortis, etc. (a phone itself was a class as well). With this number of phonetical classes, the computing time for the decision network is up to 70-80% of the baseline system. This is because each class has to be tested in every node, whether the split of the node gives an improvement in the mutual information or not.

4. RESULTS

The results in this section were evaluated on the Resource Management (RM) database. All results given here are only for single stream monophone HMMs. This means only the cepstrum feature vectors were used, first and second order derivatives have not been used, as well as the power features were not used. Thus the baseline results are worse compared to the results presented in [1].

size of the output layer	10	20	50
Mutual information	3.0622 bit	3.0003 bit	2.9845 bit

Table 1. Mutual Information for different network sizes of a single layer decision network

Table 1 shows the value for the mutual information (on a subset of the training data) according to the size of the output layer of the decision network. It can be seen, that the value for the mutual information drops, the larger the output size is. This is because then there are many very small networks in the next layer networks, which are to small to improve the mutual information.

System	mutual inform.	recognition rate
Baseline MMINN	2.079 bit	75.62 %
Flat network	2.078 bit	74.99 %
Tree network	2.095 bit	75.75 %

Table 2. Performance of the different topologies

Table 2 shows the recognition rates for the baseline system and the two new topologies. The recognition rate reported is the average recognition rate (accuracy) for the 4 RM test sets (feb89-sep92). The first topology achieves nearly the same value for the mutual information, but is worse in recognition rate than the baseline system. With the second, hierarchical approach, the value of the mutual information is even higher than in the baseline case and the recognition result is nearly the same.

5. CONCLUSIONS

The new topologies for MMINNs are both faster to train than the baseline system in [1]. The first approach is very fast, and uses only about 15% of the training time. On the other side, this approach leads to recognition rates, which are about 0.5% (absolute) worse, than the recognition rates of the baseline system. The second topology achieves the same recognition rate as the baseline system (even 0.1% better), and uses only 90% of the computational effort.

6. REFERENCES

1. G. Rigoll, Ch. Neukirchen, and J. Rottland. A new hybrid system based on MMI-neural networks for the RM speech recognition task. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 865–868, Atlanta, GA, May 1996.
2. J. Rottland, Ch. Neukirchen, D. Willett, and G. Rigoll. Large vocabulary speech recognition with context dependent MMI-connectionist/HMM systems using the WSJ database. In *Proc. 5th European Conference on Speech Communication and Technology*, pages 79–82, Rhodes, Greece, September 1997.
3. G. Rigoll and Ch. Neukirchen. A new approach to hybrid HMM/ANN speech recognition using mutual information neural networks. In Michael C. Mozer, Michael I. Jordan, and Thomas Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, pages 772–778. The MIT Press, December 1996.
4. A. Glaeser. Modular neural networks with task-specific input parameters for speaker independent speech recognition. In *Proc. 4th European Conference on Speech Communication and Technology*, pages 1655–1658, Madrid, Spain, September 1995.
5. J. Fritsch, M. Finke, and A. Waibel. Context–dependent hybrid HME / speech recognition using polyphone clustering decision trees. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1759–1762, Munich, Germany, April 1997.