Speech Recognition Based on Artificial Neural Networks

Veera Ala-Keturi

Helsinki University of Technology
Veera.Ala-Keturi@hut.fi

Abstract

In this survey I will first look at some basic theory of neural networks. I will then study hybrid (connectionist) models where hidden Markov models (HMM's) and neural neworks (NN's) are used together in speech recognition. In a hybrid HMM/NN system the neural network estimates the posterior probabilities, which can enhance the discrimination ability of the system. Finally I will look at a more novel field of research dealing with extracting features from the data by neural networks.

1. Introduction

The purpose of this survey is to obtain an understanding of the state-of-the-art in usage of neural networks for speech recognition. The goal is to find out about different neural network related methods that can be used for speech recognition and compare pros and cons of each technique if possible. Section 2 is a short introduction to neural networks; what they are and what can be used for. Section 3 deals with a hybrid approach where both artificial neural networks and a hidden Markov model are used together for speech recognition. Section 4 describes the possibility of obtaining a good set of features foe speech recognition using neural networks – this is one possibility to enhance speech recognition accuracy but is not yeat a mature technology.

2. Artificial neural networks from the viewpoint of speech recognition

Artificial neural networks (ANN's) are systems consisting of interconnected computational nodes working somewhat similarly to human neurons. Neural networks can be used e.g. to approximate functions or classify data into similar classes than can be e.g. phonemes, sub-phoneme units, syllables or words in the speech recognition domain. The ability to learn by adapting strengths of inter-neuron connections (synapses) is a fundamental property of artificial neural networks.

2.1. Different kinds of neural networks

Researchers all over the world have come up with countless different structures for a neural network. Here some of them are given a short description of.

2.1.1. Feedforward networks

A feedforward network has connections only forward in time: a neuron in layer i can only send data to neuron j if j>i. Only adjacent layers can be connected to each other as in multi-layer perceptrons, or there can be forward "shortcuts" between layers that are not next to each other.

A time delay neural network (TDNN) uses feedforward connections with weighted delays, which makes it possible to use contextual information (i.e. previous values of e.g. acoustic speech vectors) in classification of data. This adds complexity to the network and requires a more complex learning rule such as backpropagation through time, but enhances accuracy [17].

Also multi-layer perceptron networks discussed below are feed-forward networks.

2.1.2. Perceptrons and multi-layer perceptrons

A perceptron is a simple neuron model that has a set of inputs, a weight for each input and a (often nonlinear) activation function that the neuron performs to the weighted sum of inputs (plus possible bias) before sending the value to its output [10]. The perceptron model is shown in Figure 1, where x is an input vector, w is a weight vector, w_0 is the bias and the activation function is a step function.

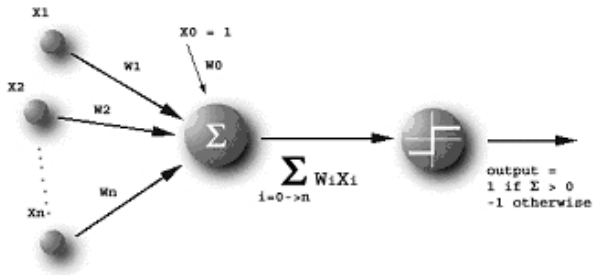

Figure 1. Model of a perceptron [3]

A multi-layer perceptron (MLP) consists of at least two layers of perceptrons: it has an input layer, one or more hidden layers and output layer. The hidden layers act as a feature extractor and use a nonlinear function such as sigmoid or a radial-basis function to generate (often complex) functions of input. The outputs of all the neurons in the hidden layer serve as input to all of the neurons on the next layer. The output layer acts as a logical net that chooses an index to send to the output on the basis of inputs it receives from the hidden layer, so that the classification error is mimimized [16].

2.1.3. Recurrent neural network

In recurrent neural networks (RNN) the output of a neuron is multiplied by a weight and fed back to the inputs of the neuron itself with a delay. This means that the state of a neuron is a function of its inputs and also the neuron's previous states. Recurrent neural networks have achieved better speech recognition rates than MLP, but the training algorithm is again more complex and dynamically sensitive, which can cause problems.

2.1.4. *Self-organizing maps*

Self-organizing maps (SOM) are a technique or mapping a high-dimensional space to a smaller-dimensional map (usually a grid or line) so that input vectors close to each other correspond to neurons close to each other in the map. Each neuron in the network includes a code vector that points to a neuron in the map. The network is trained using competitive learning. Self-organizing maps have been used for various purposes, also some related to speech recognition (e.g. generating quasi-phoneme patterns for small-vocabulary recognition), but SOM's have been found to be more useful in other tasks but classification.

2.1.5. *Probabilistic vs. binary-output neural networks*

Output neurons of a neural network often have binary outputs: an output neuron tells whether the input data fills some criteria or not (that is, whether the neuron thinks that the input belongs in a certain class). When building a speech recognition system, the classes often are overlapping and it is necessary to know how likely it is that the input belongs in each of the phoneme or word classes. Thus the outputs of neurons can be set to be numbers between 0 and 1, so that the sum of all outputs equals one. This is then called a probabilistic neural network, and the outputs can be used to find the best sequence of states in order to find the pronounced utterance.

2.2. Model structure and order, initialization

When planning on building a neural network based system, one must choose the model order, which usually means the amount of hidden neurons. Too little hidden neurons can mean not learning the data well enough, but on the other hand too many hidden neurons can result in overfitting, which decreases performance with new data. The amount of parameters is most often done empirically, but also cross validation (trying with different setups and comparing results) can be used. The amount of free parameters can be reduced by sharing weights or other parameters over the network.

One also needs to choose the number of outputs, which means choice of classes (phonemes, biphones, triphones, syllables or words) in the case of supervised learning or (usually) the number of clusters wanted in the case of unsupervised learning.

The structure of the model can be a "big dum network", a network can be build of hierarchically structured subnetworks each of which is specialized for one class, or one can initially train a large network and then cut the amount of parameters by pruning (deleting or combining neurons with minimum loss of accuracy). Molecular networks consisting of highly specialized parts can enable building very complex but efficient neural networks in the future.

2.3. Learning

The goal of training a neural network is that the network classifies new data well, not to overlearn details of the training data. This is why an independent test set should be used as a criteria for continuing training of the network. Training the network can be done either in a supervised manner, in which the network is given a set of labeled data for learning, or in an unsupervised way, where the task of the network is to find clusters of data that are similar in some way (maybe not the one obvious to human experts). In the next subsections some learning algorithms are discussed; the interested reader is referred to [6] for more detailed information.

2.3.1. *Supervised learning*

In supervised learning the training data is first classified manually or by using another speech recognition system, and then the network is trained with this data to output the classifications given. The network iteratively changes its weights to minimize a given cost function E. This can be the difference between the output of a neuron and the desired output, which is the maximum likelihood (ML) criterion, the optimal solution for linearly separable functions. The squared error is used in least mean square (LMS) learning, which enables classification of data that is not linearly separable. The cost function of LMS learning is

$$E = \sum_k \sum_n c e_k^{\,2} \tag{1}$$

where e_k is the difference between the k:th output and its desired value, n is the time index of input data and c is a constant. This formula is for off-line learning, where all the data is shown to the network before counting the weight updates; in practical situations the cost function is often calculated after each sample to get an estimate local in time. This means the updates are no longer optimal, but the learning is faster. This is called online learning.

In steepest descent methods the weights of the network are changed towards the negative gradient of the error surface, in order to find a local or global minimum. One needs to choose a learning rate to determine how much the weights are updated each time, to find a balance between slow convergence and oscillation of weights. In some algorithms also penalty terms are defined to avoid unnecessary fluctuations. Error back-propagation learning (EBP) can be used for updates: first the error is calculated at the output, then it is sent back to the network and partial derivatives of the error is calculated, after which the updates in each neuron are done and the iteration starts over with new input.

2.3.2. *Unsupervised learning*

In unsupervised learning the network is given no labeled data. Instead the network is given some similarity measure such as cosine distance or n-dimensional norm, that it uses to find groups of input vectors whose distance according to the similarity measure is small. The network develops a set of classes or patterns to which it assigns input vectors. The weights of the network change iteratively until it reaches a stable state.

2.3.3. *Reinforcement learning*

Sometimes there is no pre-classified data available, but the network gets some external feedback of its decisions. This is called reinforcement learning. The training of a network in this way takes longer and usually requires more data than in supervised learning.

3. Hybrid HMM/NN models

Using NN's and Hidden Markov Models (HMM's) together for speech recognition can sometimes give better results than either of the techiques alone. In hybrid HMM/NN models the neural networks are used to estimate posterior probabilities of classes given the input data. Neural networks usually achieve roughly the same performance for this task as a Gaussian model but require smaller amount of parameters [17] and can possibly be more computationally efficient when running. A thorough descpription of hybrid speech recognition theory and concepts from the year 1994 is found in [4]. After that some enhancements to the basic techniques have been developed, but the principles remain the same.

3.1. Example approach: Speech Recognition Using Neural Networks at CSLU

A general-purpose speech recognition system used at the CSLU, Oregon Graduate Institute of Science and Technology [12] is depicted in Figure 2.

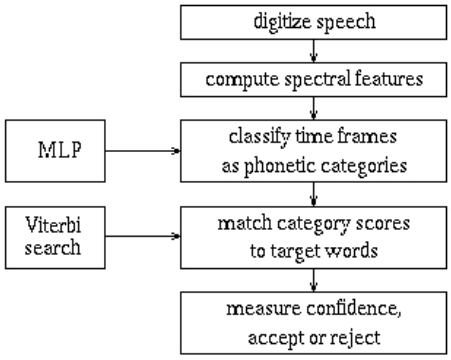

Figure 2. Overview of a frame-based speech recognition process at CSLU [12]

The speech signal is digitized at 8000 Hz and divided into frames of 10 ms. A spectrum and a mel-cepstrum are obtained for each frame, and a set of features are calculated. There are 26 features: 12 mel-frequency cepstral coefficients (MFCC coefficients), 12 MFCC delta features, one energy feature, and one delta-energy feature – a total of 26 features per frame. For each time instant, also 2 feature sets from the left context (30 and 60 ms back) and 2 from the right context (30 and 60 ms forward) are included to provide information about the acoustic context.

The features of the time instant in interest and its context window (which means 5 times 26 = 130 features) are then fed into an MLP that calculates the posterior probabilities of each output phonetic-based class given the feature vector. The MLP has 200 hidden units, which has been empirically determined. There are a maximum of three classes per phone, depending on the quality of the phone: one for steady state part of the phone that does not depend on context (e.g. <E>), one for the part that depends on the left context (e.g. $mid<E where $mid is a cluster of similar phones) and one for right-context-dependent part (e.g. E>$sil where $sil means silence). The number of phonetic classes is 544. The neural network is shown in Figure 3.

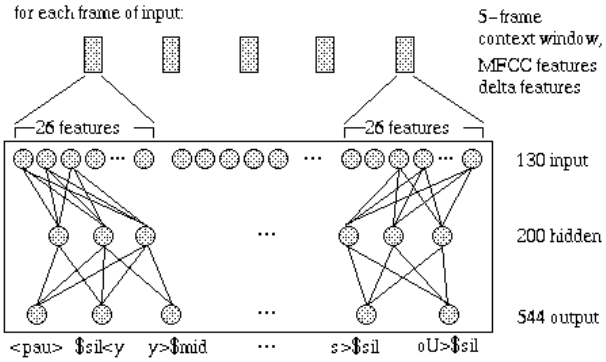

Figure3. The neural network used in the frame-based speech recognition process at CSLU [10]

The classification results of each frame are then passed to a HMM that is used to calculate the best-matching target words. The information from a grammar and a pronounciation model are combined in order to get a set of legal sequences of phonetic categories (output classes of the NN). The matching is done with an exhaustive Viterbi search that calculates probabilities of every legal sequence of classes given the phone probabilities from the NN. There is a separate model for pronounciations, which can be used to make new phoneme sequences for words in the grammar.

3.2. Assets of hybrid HMM/NN models

When a neural network is used to estimate class probabilities, the discrimination is enhanced by using a learning rule that minimizes classification error and maximizes discrimination between classes [4]. The NN requires fewer parameters than a Gaussian model, has a flexible architecture and is fast and computationally efficient after the training phase has completed. In addition it does not require the strong statistical assumptions of the input that are needed in the traditional Gaussian model, nor expert understanding of the complex linguistic, acoustic and phonetic dependencies in speech. This makes the use of neural networks for posterior probability estimation beneficial.

3.3. Downsides of hybrid HMM/NN models

Training a neural network classifier generally takes longer than training a Gaussian mixture model. EBP algorithm requires a lot of time when the network is big, and it is practical to replace it with some simpler algorithm. A large amount of units (classes) means huge amounts of memory needed and operations to calculate, which constrains the choice of basic units. For example, words or sentences cannot be chosen as classes to estimate in large-vocabulary speech recognition because there are too many of them. The whole network also has to be retrained when new data such as a new speaker is added, which can cause unpractical delays and makes it difficult to adapt the system to a new speaker rapidly [4].

Global NN estimators such as MLP are not able to generalize beyond the limits of test data, which means that unstability of the network is possible. This does not happen with local estimators such as Gaussian densities.

The neural training of a neural network often results in stabilization at a local minimum where the error is larger than

the minimum possible error. This results in suboptimal performance, but usually the solution is good enough.

3.4. Some approaches used with hybrid models

Context modeling can be done in different ways in a neural network. It is possible to introduce delays in the network so that the previous values of input affect the current state of neurons. This approach has the effect of making the network and the calculations rather complex. One can input such data to the network that already has context information: the input feature vector can be delayed several times before passing it to the network as several copies, or the context information can be included in the data itself, as has been done at the CSLU [12] (see Section 3.1). One can also use a set of neural networks, one for each context, and use their outputs to compute HMM context-dependent observation probabilities using a Bayesian factorization [1].

Speaker adaptation has been approached using linear input transformations, either global transformations or mixtures of transformation networks [1].

Different feature sets have been used in different systems. Some examples of these are mel-cepstrum frequency coefficient (MFCC), delta MFCC, energy term, delta energy term [12], modulation-filtered spectrogram (MSG) and normalized spectra derived from a modified perceptual linear prediction (PLP) [1] etc.

The search for the best sequence of words has been done in several ways less computationally heavy than the Viterbi algorithm: beam search, dynamic memory allocation and multi-pass approaches are some examples [17].

The training time has been cut down be e.g. using a hierarchical mixture of experts (HME) network and applying the principle of modularity [8].

3.5. Computational issues

Ideally the goal of speech recognition for continuous speech is to estimate the most probable sequence of words given the acoustic input. When the size of the vocabulary is large, the amount of possible word sequences whose probability should be evaluated becomes huge, and thus the amount of memory and processor time needed will also be huge. Special parallel processing device can be used to cut down the time, but this solution does not really remove the problem. In practice, sub-optimal but efficient algorithms are used in stead of the theoretically best solutions; for instance the Viterbi algorithm in the HMM part is often replaced with a beam search that limits the amount of active (possible) routes in the word sequence lattice.

4. Feature extraction using neural networks

The results obtained in speech recognition for natural (conversational and large vocabulary) speech have not been as satisfying as one would imagine after decades of research. One reason to this might be that the features that have been extracted from speech data and then used for classification are not the optimal ones. Neural networks can be used to learn a “good” set of features from speech data in an unsupervised manner [7]. When the size of the feature set is smaller than the dimension of the input data, the theory of vector quantization and coding algorithms can be applied. Sometimes the best results can be achieved by combining expert knowledge and data-driven or purely statistical approach.

4.1. Feature extraction with MLP’s

If a multi-layer perceptron is fed the acoustical information as an input and it is trained to give the same data out in its outputs, the hidden layer(s) of the network will learn a representation of the data. If the amount if hidden neurons is smaller than the size of the input data vector and the accuracy of the input-output-function is good, the network has managed to extract the essential information from the data and can reproduce the data accurately enough from this information. The internal state of the hidden layer can thus be seen to contain the features needed to classify sounds into classes.

An example of an MLP performing mapping from an binary input data vector with a dimension of 8 to the same data vector is shown in Figure 4. The amount of hidden neurons is 3, which means that the hidden neurons have a compressed representation of the signal.

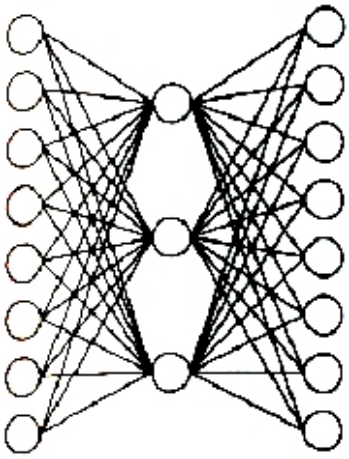

Figure 4. A multi-layer perceptron as a feature extractor: the amount of inputs and outputs is equal and the hidden layer has less neurons.[2]

In Table 1 the state of the neurons after training is shown for some input sequences (in this example case only one of the inputs can be 1 at a time; in speech recognition system this would not be true).

Table 1: The internal state of the hidden layer of the MLP in Figure 4 performing identity mapping. [2]

Input	Neuron 1	Neuron 2	Neuron 3	Output
10000000	.89	.04	.08	10000000
01000000	.15	.99	.99	01000000
00100000	.01	.97	.27	00100000
00010000	.99	.97	.71	00010000
00001000	.03	.05	.02	00001000
00000100	.01	.11	.88	00000100
00000010	.80	.01	.98	00000010
00000001	.60	.94	.01	00000001

4.2. Independent component analysis using neural networks

Independent component analysis (ICA) is a statistical and computational technique for revealing hidden factors that underlie sets of random variables, measurements, or signals [13]. The data vector components are assumed to be linear combinations of these basic factors, which are assumed to be non-gaussian and as independent of each other as possible. ICA factors can be extracted from data by different means, one of which is neural networks.

The data is usuaully first whitened: component means and variations are set equal by multiplying with a whitening matrix V. Then a linear transformation is performed to the data to get independent components (IC) that can be seen as a kind of optimal features. The structure of such a neural network is shown in figure 5.

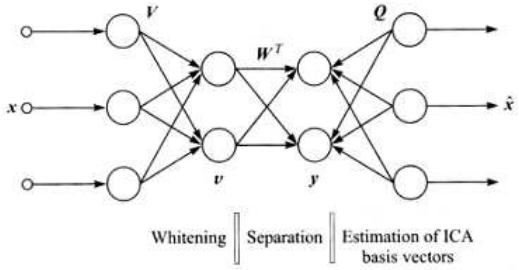

Figure 6. Independent component analysis using neural networks [2]

ICA is not a method especially developed for speech data, but it is used in several fields such as digital images, document databases, economic indicators and psychometric measurements, and it can be also applied to speech recognition. ICA for phoneme recognition (whithout neural networks) has been studied among others in [15]. An example of basic functions derived by ICA are shown in figure 6.

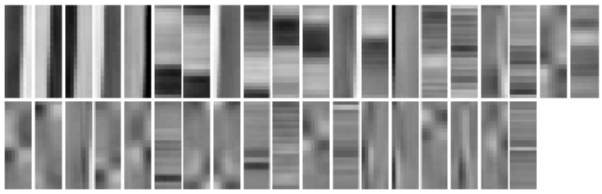

Figure 6. Basis functions from ICA in spectro-temporal domain [15].

According to the paper by Kwon et al, the biggest problems of the technique are phase sensitivity and time variance, and solutions such a mel-filters used to solve these problems lead to coefficients rather similar to standard mel cepstral coefficients. Also the phoneme recognition results were comparable to ones obtained by using MFCC coefficients.

4.3. Feature extraction using neural predictive coding

Chetouani et al [7] (related article published in ITRW on Non-Linear Speech Processing NOLISP 03 but not publicly available) have suggested a novel technique for extracting features from speech using a nonlinear model called neural predictive coding (NPC). The speech signal is fed into a neural net to get a set of nonlinear predictive codes (weights). The first layer of the predictor is common to all phonemes, and the second layer is phoneme-class specific. Each second layer neuron produces its own set of codes that are used to calculate prediction error and update the codes using it. The classification network and the feature extraction component are trained together. The technique is speaker-independent and discriminative since the minimum classification error (MCE) criterion is used. The approach seems to give better results for both phoneme classification and speaker identification than traditional feature sets such as LPC; MFCC or PLP. The pros and cons are difficult to evaluate without seeing the official article, but based on the short slides the technique seems promising.

5. Discussion

Neural networks are an interesting area of research in the sense that they operate more similarly to human brain than a conventional computer logic. They have been studied for more than a decade, and even though the results have not been as magnificent as some have expected, the consept of artificial neural network can be useful among others in the speech recognition field.

The combination of an artificial neural network and a Hidden Markov Model in speech recognition has been found to work about as well as the more traditional Gaussian Mixture Model approach. There are pros such as fast classification after training and possibility of solving more complex problems with specialized networks, but also problems like long training time and possibility of instability outside the range of training data. After the publication of "Connectionist Speech Recognition: A Hybrid Approach" [4] more refined strategies and algorithms have been developed, but the accuracy of speech recognition has not dramatically improved in real conversational situations.

Feature sets used so far to describe speech sounds in a compact way seem not to be optimal to produce the best possible recognition results. New feature sets can be produced with different algorithms using neural networks, the most simple of which is the 1-to-1 mapping described in section 4.1. Not very many articles on this research area have been published yet, but the results so far seem promising: already now better recognition results have been reported for feature sets derived by neural predictive coding [7] than traditional feature sets. It is not easy to state which approach would be the best, but I believe that feature extraction using NN's will be something to look forward to.

6. References

[1] Abrash, V., "Mixture Input Transformations For Adaptation of Hybrid Connectionist Speech recognizers", *Proceedings of the 5th European Conference of Speech Communication and Technology, Rhodes, Greece*, 1997

[2] Altinok, A., "ICA", http://www.cs.ucsb.edu/~cs281b/winter2002/Misc/ica.ppt *Material for CS/ECE 281B Advanced Topics in Computer Vision, Computer Science, University of California*, 2002

[3] Blankenstein, B., “Artificial Neural Networks (ANN)”, CS 527A Lecture Notes, http://www.cs.wustl.edu/~sg/CS527_SP01/diagram7_2.gif, *Department of Computer Science & Engineering, Washington University in St. Louis*, 2001

[4] Bourlard, H. A. and Morgan, N., ``Connectionist Speech Recognition: A Hybrid Approach", *Kluwer Academic Publishers, Massachusettes*, 1994.

[5] Bourlard. H., Konig, Y. And Morgan, N., “REMAP: Recursive Estimation And Maximization Of A Posteriori Propabilities In Connectionist Speech Recognition”, *Internal report of ICSI, TR-94-064*, 1995

[6] Bourlard, H., Dupont, S., Hermansky, H. and Morgan, N., “Towards Subband-based Speech Recognition”, *Proc. VIII European Signal Processing Conference EUSIPCO'96, Trieste, Italy*, 1996

[7] Chetouani, M., Faúndez-Zanuy, M., Gas, B. and Zarader, J. L., “Non-Linear Speech Feature Extraction for Phoneme Classification And Speaker Recognition”, http://www.iiass.it/school2004/School_Materials_1/oral_contributions/Chetouani_short_slides.pdf, *Nonlinear Speech Processing School, Vietri sul Mare (Salerno), ITALY*, 2004

[8] Frisch, J., “Modular Neural Networks For Speech Recognition”, *Master’s Thesis, Department Of Computer Science, Carnegie-Mellon University, Pittsburgh Pa*, 1996

[9] Franco, H., Weiniruub, M. and Cohen, M., “Context Modeling in a Hybrid HMM-Neural Net Speech recognition System”, *Proceedings of the International Conference on Neural Networks, Houston, TX*, 1997

[10] Haykin, S., “Neural Networks A Comprehensive Approach”, Prentice Hall, 1999

[11] Honkela, T., “Self-Organizing Maps In Natural Language Processing”, *Thesis for a degree of PhD., Neural Networks Research Centre, Helsinki University of Technology*, 1997

[12] Hosom, J. P., Cole, R., Fanty, M., “Speech Recognition Using Neural Networks at the Center for Spoken Language Understanding”, http://speech.bme.ogi.edu/tutordemos/nnet_recog/recog.html, *OGI School of Science and Engineering, Oregon Graduate Institute of Science and Technology*, 1999

[13] Hyvärinen, A., “ Survey on Independent Component Analysis”, National Computer Symposium 1999, http://www.cis.hut.fi/aapo/papers/NCS99web/, 1999

[14] Jurafsky, D. and Martin, J. H., “Speech And Language Processing: An Introduction To Natural Language Processing, Computational Linguistics, And Speech Recognition”, *Prentice Hall*, 2000

[15] Kwon, O-W., Lee, T-W., “ ICA-Based Feature Extraction for Phoneme Recognition“, *Interspeech 04*, 2004

[16] Robinson, A. J., Cook, G. D., Ellis, D. P. W., Fosler-Lussier, E., Renals, S. J., Williams, D. A. G., “Connectionist Speech Recognition Of Broadcast News”, Speech Communication, 37:27-45, 2000

[17] Salmela, P., “Neural Networks in Speech Recognition”, *Tampere University of Technology, Publications 295*, 2000

[18] Tebelskis, J., “Speech Recognition Using Neural Networks”, *PhD Dissertation, Carnegie Mellon University*, 1995