Neural Networks used for Speech Recognition

Wouter Gevaert, Georgi Tsenov, Valeri Mladenov, *Senior Member, IEEE*

***Abstract*— In this paper is presented an investigation of the speech recognition classification performance. This investigation on the speech recognition classification performance is performed using two standard neural networks structures as the classifier. The utilized standard neural network types include Feed-forward Neural Network (NN) with back propagation algorithm and a Radial Basis Functions Neural Networks.**

***Index Terms*—speech recognition, neural networks, Feed-forward Neural Networks, Radial Basis Functions Neural Networks**

I. INTRODUCTION

SPEECH is probably the most efficient way to communicate with each other. This also means that speech could be a useful interface to interact with machines. For a long time research on how to improve this type of communication has been done. Some successful examples based on it during the past years, since we have knowledge about electromagnetism; includes the invention of the megaphone, telephone and etc.

Even in 18th century people were experimenting on speech synthesis. For example, in the late 18th century, Von Kempelen developed a machine capable of 'speaking' words and phrases. Nowadays, thanks to the evolution of computational power it has become possible not only to develop, test and implement speech recognition systems, but also to have systems capable to real-time conversion of text into speech. Unfortunately, despite the good progress made on that field, the speech recognition process is still facing a lot of problems, with most of them contributed to the fact that speech is a very subjective phenomenon.

In general some of the most common ones are:

- Speaker variation: in this case exactly the same word is pronounced differently by different people because of age, sex, anatomic variations, speed of speech, emotional condition of the speaker and dialect variations.
- Background noise: a noise environment can add noise to the signal. Even the speaker himself can add noise by the way he speaks.
- Suprasegmental aspects: Influence of intonation and putting stress on syllables. These aspects influence the pronunciation of a word.
- Continuous character of the speech: when we speak, seldom there is a break between words. Speech is mostly one uninterrupted stream of sounds. This makes it very hard to detect individual words.
- Other external factors are: position of the microphone in respect to the speaker, direction of the microphone and many others.

Neural networks are composed of simple computational elements operating in parallel [1]. The network function is determined largely by the connections between elements. We can train a neural network so that a particular input leads to a specific target output.

In this paper we discuss the usability of two different types of neural networks like Feedforward-back propagation neural netwoprk and Radial Basis Functions neural network for speech recognition using MATLAB. With all of them we try to classify the input samples to known output words.

In the next chapter of this paper, a general introduction to speech recognition will be given. Some basic ideas, problems and challenges of the speech recognition process is discussed. In the third chapter we focus on the signal pre-processing necessary for extracting the relevant information from the speech signal. The implementation of the neural network classifiers is a subject of the fourth chapter. The conclusion remarks are given in the last chapter of the paper.

II. GENERAL STRUCTURE AND PROBLEMS OF A SPEECH RECOGNITION PROCESS

The speech recognition process can generally be divided in many different components illustrated in Fig. 1

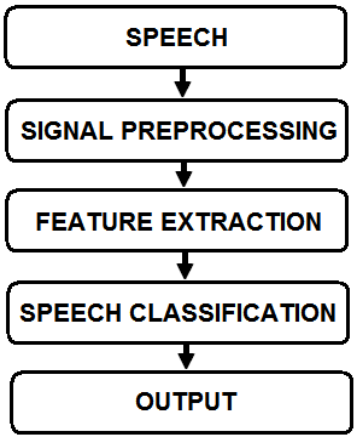

Fig. 1. Speech recognition process

Wouter Gevaert is with the Department of Electronics-ICT, University College West Flanders, 5 Graaf Karel De Goedelaan, Kortrijk-8500, Belgium and with the Video Coding & Architectures Research group, University of Technology of Eindhoven, The Netherlands
E-mail: wouter.gevaert@howest.be

Valeri M. Mladenov and Georgi T. Tsenov are with the Department Theory of Electrical Engineering, Technical University of Sofia, 8 Kl. Ohridski Str., Sofia-1000, Bulgaria
E-mail: {valerim, gogotzenov}@tu-sofia.bg

DOI: 10.2298/JAC1001001G

The first block, which consists of the acoustic environment plus the transduction equipment (microphone, preamplifier and AD-converter) can have a strong effect on the generated speech representations. For instance we can have additional impact generated from additive noise or room reverberation.

The second block is intended to deal with these problems, as well as deriving acoustic representations that are both good at separating classes of speech sounds and effective at suppressing irrelevant sources of variation.

The third block must be capable of extracting speech specific features of the pre-processed signal. This can be done with a variety of techniques like cepstrum analysis and the spectrogram.

The fourth block tries to classify the extracted features and relates the input sound to the best fitting sound in a known 'vocabulary set' and represents this as an output.

The commonly used techniques for speech classification include:

A. Dynamic Time Warping (DTW)

This technique compares words with reference words. Every reference word has a set of spectra; but there is no distinction between separate sounds in the word. Because a word can be pronounced at different speeds, a time normalization will be necessary. Dynamic Time Warping is a programming technique where the time dimension of the unknown word is changed (stretched and shrinked) until there is a similarity with a reference word.

B. Hidden Markov Moddelling (HMM)

Untill now, this is the most successful and most used pattern recognition method for speech recognition. It's a mathematical model derived from a Markov Model. Speech recognition uses a slightly adapted Markov Model. Speech is split into the smallest audible entities (not only vowels and consonants but also conjugated sound like ou, ea, eu,...). All these entities are represented as states in the Markov Model. As a word enters the Hidden Markov Model it is compared to the best suited model (entity). According to transition probabilities there exist a transition from one state to another. For example: the probability of a word starting with xq is almost zero. A state can also have a transistion to it's own if the sound repeats itself. Markov Models seems to perform quite well in noisy environments because every sound entity is treated separately. If a sound entity is lost in the noise, the model might be able to guess that entity based on the probability of going from one sound entity to another.

C. Neural Networks (NN)

Neural networks have many similarities with Markov models. Both are statistical models which are represented as graphs. Where Markov models use probabilities for state transitions, neural networks use connection strengths and functions. A key difference is that neural networks are fundamentally parallel while Markov chains are serial. Frequencies in speech, occur in parallel, while syllable series and words are essentially serial. This means that both techniques are very powerful in a different context.

As in the neural network, the challenge is to set the appropriate weights of the connection, the Markov model challenge is finding the appropriate transition and observation probabilities. In many speech recognition systems, both techniques are implemented together and work in a symbiotic relationship. Neural networks perform very well at learning phoneme probability from highly parallel audio input, while Markov models can use the phoneme observation probabilities that neural networks provide to produce the likeliest phoneme sequence or word. This is at the core of a hybrid approach to natural language understanding.

In this paper speech features (spectrogram and cepstrum) will be sequentially presented at neural network inputs and will be classified at the output of the network. This process is visualised in Fig. 2. classification process in the NN

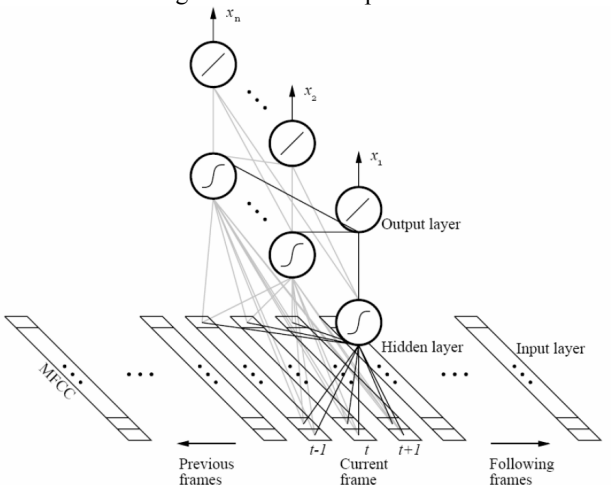

Fig. 2. Classification process in the NN

In this paper we focus on two typical NNs: Multilayer Feedforward (backpropagation) networks and Radial basis networks. These network topologies as their performance in speech recognition are discussed into detail in the next chapters.

III. IMPLEMENTATION OF SIGNAL-PREPROCESSING

In the previous section we have discussed the general structure of a speech recognition system. In this paper we put the main focus on the neural networks and not on the signal pre-processing, although signal pre-processing has a big impact on the performance of the speech classifier. It is important to feed the neural network with normalized input. Recorded samples never produce identical waveforms; the length, amplitude, background noise may vary. Therefore we need to perform signal pre-processing to extract only the speech related information. This means that using the right features is crucial for successful classification. Good features simplify the design of a classifier whereas weak features (with little discrimination power) can hardly be compensated with any classifier. We can divide this process on some distinctive steps like:

A. Representing the speech

Speech can be represented in different ways. Depending on the situation and the kind of speech information that needs to be present, one representation domain might be more appropriate than the other one.

a) Waveform

This is the most general way to represent a signal [2],[3],[4]. Variations of amplitude in time are presented. The biggest disadvantage of this method is that it cannot represent speech related information. A time-domain signal as such contains too much irrelevant data to use it directly for classification.

Fig.3 shows the time domain representation of the words 'left' and 'one'. It is immediately clear that based upon this representation, it would be difficult to extract relevant speech information and thus cannot be used as input for the neural network classifier.

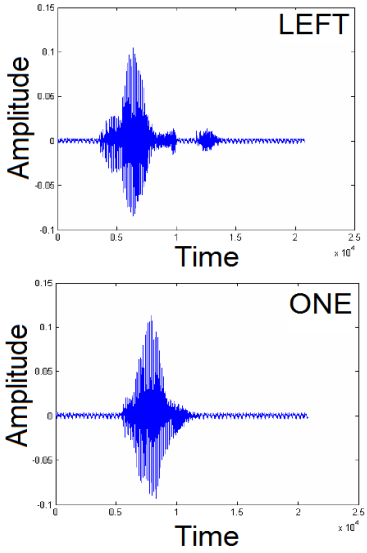

Fig. 3 Time domain representation of the words 'left' and 'one'

b) Spectrogram

There is a better representation domain, namely the spectrogram. This representation domain shows the change in amplitude spectra over time. It has three dimensions:

X-axis: Time (ms)

Y-axis: Frequency

Z -axis: Color intensity represents magnitude

The complete sample is split into different time-frames (with a 50% overlap). For every time- frame, the short-term frequency spectrum is calculated.

Although the spectrogram provides a good visual representation of speech it still varies significantly between samples. Samples never start at exactly the same moment, words may be pronounced faster or slower and they might have different intensities at different times.

Fig.5 represents two spectrograms of the word 'left' but they are calculated from two different samples. As you can see, they both show somewhat the same pattern, but the second sample is shifted in time compared to the first sample. As these patterns vary so much, makes them useless as input for the neural network unless some more signal pre-processing is performed.

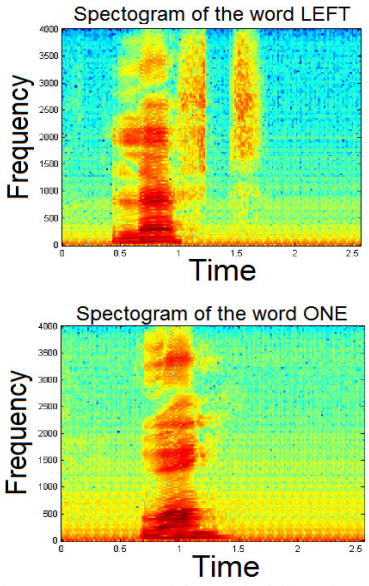

Fig. 4 Spectrogram of the words 'left' and 'one'

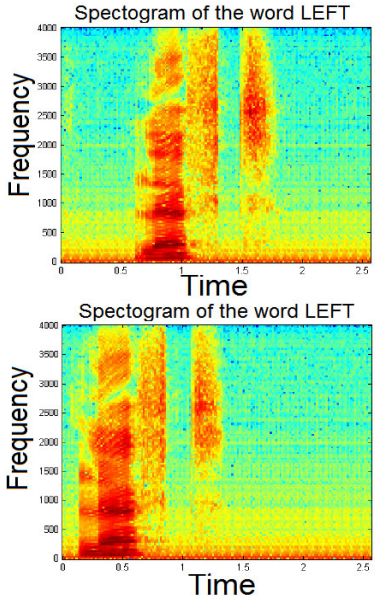

Fig. 5 Two spectrogram samples of the word 'left'

c) Cepstrum and Mel Frequency Cepstrum Coefficients

We know that human ears, for frequencies lower than 1 kHz, hear tones with a linear scale instead of logarithmic scale for the frequencies higher that 1 kHz. The mel-frequency scale is a linear frequency spacing below 1000 Hz and a logarithmic spacing above 1000 Hz. The voice signals have most of their energy in the low frequencies. It is also very natural to use a mel-spaced filter bank showing the above characteristics.

The following approximate formula is used to compute the mels for a given frequency in Hz:

$$mel(f) = 2595.^{10} \log(1 + \frac{f}{700}) \qquad (1)$$

For each tone with an actual frequency f (in Hz), a subjective pitch is measured on a scale called the 'mel' scale. The pitch of a 1 kHz tone, 40 dB above the perceptual hearing threshold is defined as 1000 mels.

The cepstrum is the forward Fourier transform of a spectrum. It is thus the spectrum of a spectrum, and has certain properties that make it useful in many types of signal analysis. One of its more powerful attributes is the fact that any periodicities, or repeated patterns, in a spectrum will be sensed as one or two specific components in the cepstrum. If a spectrum contains several sets of sidebands or harmonic series, they can be confusing because of overlap. But in the cepstrum, they will be separated in a way similar to the way the spectrum separates repetitive time patterns in the waveform.

In Fig. 6 the cepstrum of the words 'left' and 'one' is shown. Both charts show a different shape characteristic for that specific word. We discussed that the spectrogram have time dependant problems and the cepstrum is an ideal method for coping with these problems.

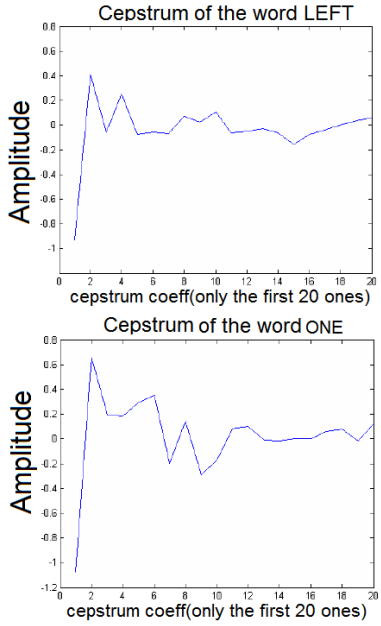

Fig. 6 Cepstrum of the words 'left' and 'one'

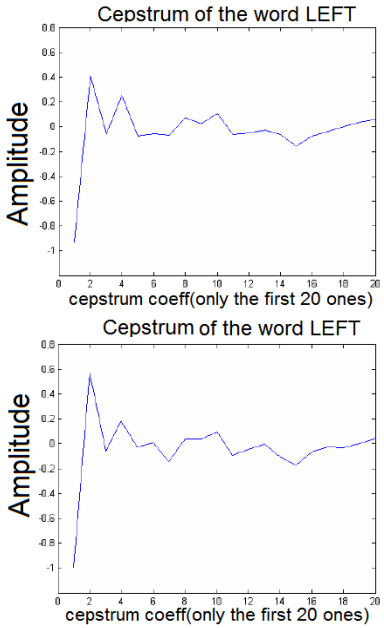

Fig. 7 Two cepstrum samples of the word 'left'

Fig.7 represents the cepstrum of two different samples of the word 'left'. It is clear that they almost have the same shape. A cepstral analysis is a popular method for feature extraction in speech recognition applications, and can be accomplished using the Mel Frequency Cepstrum Coefficient analysis (MFCC).

B. Signal Pre-processing

As the neural network will have to do the speech classification, it is very important to feed the network inputs with relevant data. It's obvious that an appropriate pre-processing is necessary in order to be sure that the input of the neural network is characteristic for every word while having a small spread amongst samples of the same word. Noise and difference in amplitude of the signal can distort the integrity of a word while timing variations can cause a large spread amongst samples of the same word [5],[6].

These problems are dealt with in the signal pre-processing part which is composed of different sub stages: Filtering, Entropy based endpoint detection and Mel Frequency Cepstrum Coefficients.

Filtering stage - samples are recorded with a standard microphone. So they contain besides speech signals a lot of distortion and noise due to the quality of the microphone or just because of picked up background noise. In this first step we perform some digital filtering to eliminate low and high frequency noise. As speech is situated in the frequency domain between 300 Hz and 3750 Hz, a bandpass filtering is performed on the input signal. This is done by passing the input signal successively trough a FIR low pass filter and then through a FIR high pass filter. An FIR filter has the advantage above an IIR filter that it has a linear phase response. The frequency response of the low pass and high pass filter is shown in Fig.8.

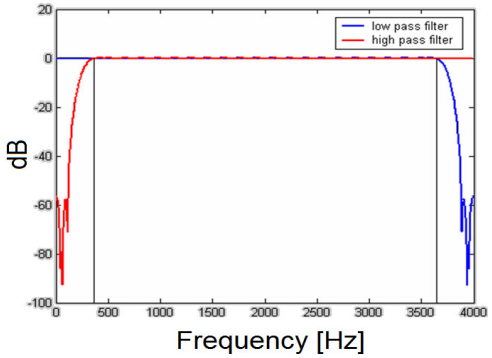

Fig. 8. Frequency response of the low and high pass filters

Entropy based Endpoint detection stage - one of the most difficult parts to deal with in speech recognition is to determine the start point (and maybe also the endpoint) of a word. Fig. 5 shows twice the spectrogram of the word 'left', but both samples are slightly shifted in time and are not appropriate as the neural network inputs. The challenge is thus to deal with that time shift.

Entropy based detection is a good method for determining the start point of relevant content in a signal [7]. In addition it performs well for signals containing a lot of background noise.

First the entropy of a speech signal is be computed. Then

a decision criterion is set to find the beginning of the signal's relevant content. This point is the new starting point of the signal. The entropy *H* can be defined as:

$$Hk = pk \log(pk) , \quad (2)$$

with *pk* as the probability density

The start point is set where the entropy curve crosses the line

$$lambda = \frac{H_{max} - H_{min}}{2} . \quad (3)$$

If we compute the entropy for the word 'left' we obtain the data shown on Fig.9. The horizontal line represents the decision criteria lambda, while the vertical line determines the start time.

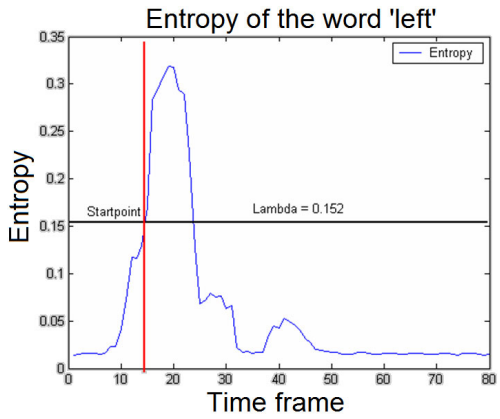

Fig. 9. Entropy of the word 'left'

Once we know the start point of the relevant information in the signal we can adapt our spectrogram and shift the start point to the beginning of the spectrogram. This is illustrated in Fig.10, where the entropy detection is performed on the word 'left'. The left picture shows the original speech signal where the right one is showing the time shifted spectrogram after entropy based begin point detection, padded with zero's.

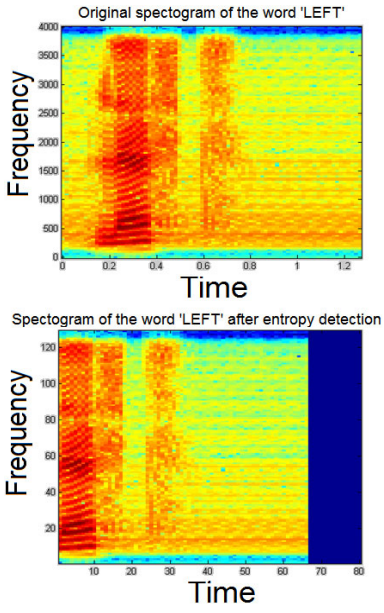

Fig. 10. Original and start point detected spectrogram of the word 'left'

These spectrograms contain 80 time frames which contain 129 frequencies each. This means a total of 80x129=10320 points which is too large to use them all as input for the neural network. Therefore a selection resulting in a smaller set of points is necessary.

One such a solution is usage of Mel Frequency Cepstrum Coefficients (MFCC) [8]. As the entropy based endpoint detection alone is an efficient method in extracting the necessary input data for the NN, the need for some better pre-processing algorithm arises. Therefore using the Mel Frequency Cepstrum coefficients is be a better strategy. We used the function melcepst, which is not a standard build in MATLAB function that returns the MFCC's of a speech sample. The amount of coefficients returned can be given as a parameter to that function. Taking many coefficients yields to a better approximation of the signal (more details), but becomes more sensitive to small variations of the different input samples. Using a fewer coefficients results into a rougher approximation of the speech signal. An amount of 10 to 20 coefficients is optimal as input for the NN.

IV. NEURAL NETWORK IMPLEMENTATIONS

Many authors used neural networks for speech recognition in the past [9], [10], [11], [12]. For our implementation the MATLAB Neural Network toolbox has been used to create, train and simulate the networks [13]. For every word we used 200 recorded samples. From these 200 samples, 100 samples were used for training, while the other 100 were be used to test the network (as these not included in the training set). The trained network can also be tested with real time input from a microphone.

A. Multilayer Feedforward Network

The first type of neural nets used for speech classification is a Multilayer Feedforward Network using Back Propagation alghorithm for training.

This type of NN is the most popular NN and is used worldwide in many different types of applications. Our network consists of an input layer, one hidden layer and an output layer.

We already discussed into detail the importance of the consistency of the neural network inputs. Feeding the NN with al data points from the spectrogram would be too much, as the spectrogram consists of 80x129=10320 data points the NN would require the same amount of inputs. Therefore we use a set of Mel Frequency Cepstrum coefficients as input for the neural network. As we only need ten up to twenty of them to represent a word, the neural network will only have 10 to 20 inputs. The input values are in a range of -5 up to 1.5. For every input neuron this parameter is set. In our design we put all these input ranges in a 'InputLayer' variable matrix.

Therefore we select a much smaller set of data points from each spectrogram. For every selected time frame we pick some frequencies. Taking 8 time frames with 10 frequency points each results in a NN input of 80 values, which is still a large input set, but with lesser dimension than the full spectrogram.

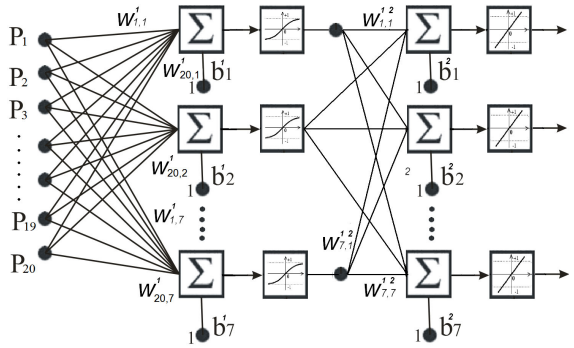

Fig. 11. Example of the feedforward backpropagation network

The hidden layer consists of non-linear sigmoidal activation function neurons using the 'tansig' MATLAB NN Toolbox function. The amount of neurons depends on some factors like the amount of input data and output layer neuron number, the needed generalization capacity of the network and the size of the training set.

First the Oja rule of thumb is applied to make a first guess on how many hidden layer neurons are required.

$$H = \frac{T}{5(N+M)} \quad , \tag{4}$$

where H is number of hidden layer neurons, N is the size of the input layer, M is the size of the output layer and T is the training set size.

If for example we want to recognize five words (with a training set of 100 samples per word). The NN has 15 inputs (MFCCs) and 5 outputs

Applying the Oja rule of thumb results in 5 neurons in the hidden layer. Tests show that this amount was ideal for recognizing five words. Recognizing more words required more hidden layer units, as the NN generalized the input data too much and was underfitted to recognize the showed input data.

The output layer consists of linear activation function elements. We used such a coding, that the amount of output neurons is equal to the amount of words we want to recognize. A value of 1 in the output matrix means that the NN classified the input to a specific word corresponding to that 1 in the output matrix.

The design of this particular network is show on Fig.11. In this example, the input layer has 20 inputs (MFCCs) and the minimum and maximum values are contained in the InputLayer matrix. The hidden layer contains 7 'tansig' neurons. The output also has 7 linear neurons. This network is designed to recognize seven different words.

Once the network is created, it can be trained for a specific problem by presenting training inputs and their corresponding targets (supervised training). A set of 100 samples of each word can is used as training data. The network is trained in batch mode which means that the weights and biases of the network are updated only after the entire training set has been applied to the network. The gradients calculated at each training example are added together to determine the change in the weights and biases. In most cases, 100 up to 200 epochs are enough to train the network sufficiently. In the training phase the network error reaches almost zero as can be seen on Fig. 12.

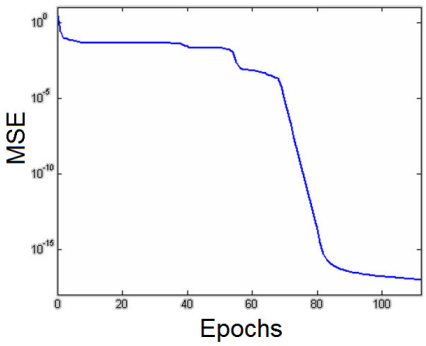

Fig. 12. Training the feedforward backpropagation network

The trained network was simulated with inputs that were not in the training set. We observed that the trained network performs very well. It is possible to recognize more than ten words.

When the number of words that have to be recognized increases the number of hidden layer neurons also has to be increased. The amount of neurons needed is almost equal to the amount of words to recognize. Increasing the number of hidden layer units causes the training time to grow sensitively. The performance of the network is mainly dependent on the quality of the signal preprocessing. The NN doesn't manage to work properly on input data coming from the spectrogram, but performs very well with MFCCs as input having more than 90% successful classification rate.

B. Radial Basis Function Network

Another approach to classify the speech samples is to make use of Radial Basis Function Network. This network also consists of three layers: an input layer, a hidden layer and an output layer. The main difference of this type of network is that the hidden layer has (Gaussian) mapping functions. Mostly they are used for function approximation, but they can also solve classification problems. Radial means that they are symmetric around their centre, basis functions means that a linear combination of their functions can generate (approximate) an arbitrary function.

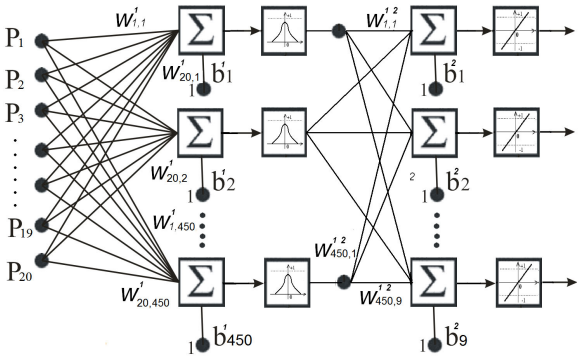

Fig. 13. RBF for recognizing 9 words

The input layer is similar to the input layer of the Multilayer Feedforward Network used. The RBF network consists of one hidden layer of neurons with basis functions. At the input of each neuron, the distance between the neuron's centre and the input vector is calculated. The

output of the neuron is then formed by applying the basis function to this distance. The RBF network output is formed by a weighted sum of the neuron outputs and the unity bias. The output layer is similar to the output layer of those in Multilayer Feed forward Networks.

Radial basis networks were designed with the MATLAB function 'newrbe'. This function can create a network with zero error on training vectors.

On Fig. 13 a RBF capable of recognizing 9 words (with 9 outputs) with an input of MFCCs is given. For a good approximation of the Mel Frequency Cepstrum Coefficients, 450 hidden layer neurons are needed, which is a lot more than the 9 sigmoid hidden layer neurons needed in the Multilayer Feedforward Network.

When simulating the trained network here also the network is capable of recognizing words that are not in the training set. The performance depends very much on the chosen spread. A too large spread causes a lower performance which means that the network tends to make more classification errors. This type of NN is practical for large training sets and it performs very well for a small spread. The amount of hidden layer neurons needed increases very fast the more words need to be recognized.

V. CONCLUSION

This paper is showing that neural networks can be very powerful speech signal classifiers. A small set of words could be recognized with some very simplified models. The pre-processing quality is giving the biggest impact on the neural networks performance. In some cases where the spectrogram combined with entropy based endpoint detection is used we observed poor classification performance results, making this combination as a poor strategy for the pre-processing stage. On the other hand we observed that Mel Frequency Ceptstrum Coefficients are a very reliable tool for the pre-processing stage, with the good results they provide. Both the Multilayer Feedforward Network with backpropagation algorithm and the Radial Basis Functions Neural Network are achieving satisfying results when Mel Frequency Ceptstrum Coefficients are used.

REFERENCES

[1] S. Haykin, "Neural Networks: a comprehensive foundation", 2nd Edition, Prentice Hall, 1999

[2] Rabiner L., Bing_Hwang J.,"Fundamentals of Speech Recognition", Prentice_Hall, 1993

[3] Jurafsky, Daniel and Martin, James H, "Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition" (1st ed.). Prentice Hall, 1996

[4] DSG Pollock, "A handbook of Time-series analysis, signal processing and dynamics", Academic press London, 1999

[5] J.H. McClellan, R.W. Schafer, M.A. Yoder, "Signal Processing First", Prentice Hall, 2003, pp. 415-426

[6] Daoudi, K. (2002) Automatic Speech Recognition:The New Millennium, Proceedings of the 15th International Conference on Industrial and Engineering, Applications of Artificial Intelligence and Expert Systems: Developments in Applied Artificial Intelligence, 253-263.

[7] K. Waheed, K. Weaver and F.M. Salam, "A robust algorithm for detecting speech segments using an entropic contrast", Circuits and Systems MWSCAS,vol 3. p.p 328-331, Michigan State University, 2002

[8] Fu-Hua Liu; Richard M. Stern; Xuedong Huang; Alejandro Acero, "Efficient cepstral normalization for robust speech recognition, human language technology", Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993

[9] Akram M. Othman, and May H. Riadh, "Speech Recognition Using Scaly Neural Networks", World Academy of Science, Engineering and Technology, vol. 38, 2008

[10] Mohamad Adnan Al-Alaoui, Lina Al-Kanj, Jimmy Azar, and Elias Yaacoub, "Speech Recognition using Artificial NeuralNetworks and Hidden Markov Models", IEEE MULTIDISCIPLINARY ENGINEERING EDUCATION MAGAZINE, VOL. 3, 2008

[11] Dou-Suk Kim and Soo-Young Lee, "Intelligent judge neural network for speech recognition", Neural Processing Letters , Vol 1

[12] Chee Peng Lim, Siew Chan Woo, Aun Sim Loh, Rohaizan Osman, "Speech Recognition Using Artificial Neural Networks," wise, vol. 1, pp.0419, First International Conference on Web Information Systems Engineering (WISE'00)-Volume 1, 2000

[13] Using MATLAB Version 6 The MathWorks Inc, Natick, MA, 2002.

Wouter Gevaert graduated in Industrial Engineering ICT from the University College West Flanders of Kortrijk, Belgium in 2001 where he received his masters degree. In 2002 he graduated in Industrial Management from the K.U. Leuven, Belgium. Currently he is teaching at the University College West Flanders, Belgium and graduating in signal processing systems from the University of Technology in Eindhoven, The Netherlands. His research interests are in the field of audio and video signal processing, neural networks and face recognition.

Georgi Tsenov graduated in Industrial Automation from the Technical University of Sofia, Bulgaria in 2002. He got M.Sc. in Industrial Automation from the same institution in 2004. Currently he is an Assistant Professor at Department of Theory of Electrical Engineering, at the Technical University of Sofia. His research interests are in the field of sigma-delta modulation, circuits and systems, nonlinear systems, signal and image processing and neural networks.

Valeri Mladenov (M'96–SM'99) graduated in Electrical Engineering from the Technical University of Sofia, Bulgaria in1985. He received his Ph.D. from the same institution in 1993. Currently he is a Head of the Department of Theory of Electrical Engineering, at the Technical University of Sofia. Dr. Mladenov's research interests are in the field of nonlinear circuits and systems, neural networks, artificial intelligence, applied mathematics and signal processing. He has more than 140 scientific papers in professional journals and conferences. He is a co-author of ten books and manuals for students. As a member of several editorial boards Dr. Mladenov serves as a reviewer for a number of professional journals and conferences. He is a member of the IEEE Circuit and Systems Technical Committee on Cellular Neural Networks& Array Computing and Chair of the Bulgarian IEEE Circuit and Systems (CAS) chapter.