Performance analysis on speech recognition using neural networks

Mihai Sima
Victor Croitoru
Dragoş Burileanu
"Politehnica" University of Bucharest
P.O. Box 49-20, Bucharest 73400, Romania
sima@ADComm.pub.ro,
croitoru@ADComm.pub.ro,
bdragos@mESsnet.pub.ro

Abstract. *The paper presents a neural network approach for speech recognition tasks in Romanian language. We describe the structure of a speaker-independent system for isolated word recognition, based on a neural network paradigm combined with a dynamic programming algorithm. The experimental results demonstrates that a hybrid model leads to higher recognition rates than the classic technologies.*

Keywords: *speech recognition, neural networks, hybrid systems, acoustic modeling.*

Introduction

Neural networks have been used for many different tasks in several domains and they have proved to be very efficient for learning complex input-output mappings. Neural algorithms offer alternatives to classical techniques and have an important potential for implementing discrimination, nonlinear feature extraction , or classification based on the distance to learned reference patterns. In the speech processing domain, neural networks have been used for speech recognition for several years; many experiments were made for isolated word recognition on small vocabularies to continuous speech recognition.

We begin this paper with a general approach concerning speech recognition technology and a brief comparison with classical systems, based on our previous experience [1], [2] and related results presented in the last years. We especially discuss some aspects concerning dynamic programming (DP) and vector quantization (VQ), presenting both their strengths and limitations. A basic idea is that even if DP has been proven successful in modeling the temporal structure of the speech signal, it is not capable of assimilating a wide variety of speaker-dependent spectrum pattern variations; on the other hand, training neural networks for recognizing speaker-independent speech is not a trivial task, because a large amount of labelled speech data must be handled and consequently is very time consumming. So, for example, combining the high time alignment capabilities of DP with the flexible learning function of the neural paradigms is expected to lead to an advanced recognition model suitable to speaker independent recognition problems. Several activities have already been reported concerning this last subject.

Then the paper describes the design philosophies of a speaker-independent system for isolated digit recognition in Romanian language. We present a comparative set of experiments based on two different approaches:

- a classical approach, where the recognition task was accomplished using a dynamic time warping (DTW) algorithm and two different parametrization methods; DTW is used due to its abilities to time-align two perceptually equivalent words spoken at different rates;
- a neural network based approach, in which the DP was combined with a learning vector quantization (LVQ) algorithm; the result is a dynamic full word recognizer model which increases the temporal invariance of isolated words and achieves shift-invariant speaker recognition task.

A great deal of effort was made for establishing the optimum choice at different levels of processing; we believe that only in this way, good performances can be obtained (i.e., real time processing or higher rate of recognition accuracy).

A Speech Recognition Primer

A general speech recognition model is shown in figure 1 [4]. The block diagram of the model include:

- a signal processing module for obtaining a reprezentation of the speech signal;
- a feature extraction module for identifying the key components of this reprezentation and eliminating redundant information;
- time alignment and pattern matching algorithms for performing word detection;
- language processing for selecting a final word string.

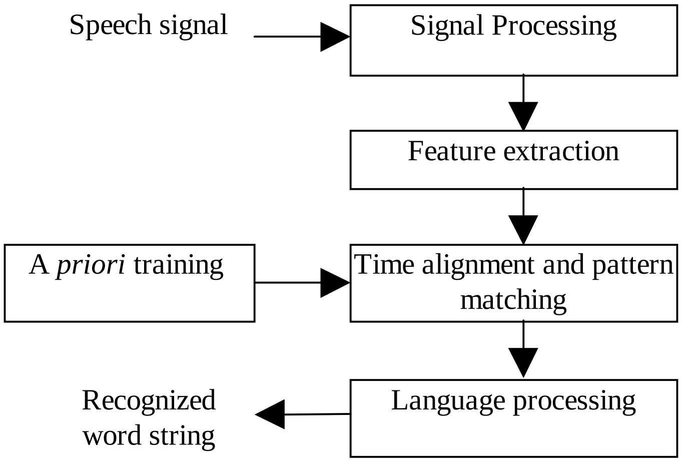

Figure 1 – A block diagram of a general speech recognition model

The *signal-processing module* processes the sampled speech signal in order to produce basic spectral and temporal measurements. The *feature-extraction module* computes a set of parameters, reprezenting parametrically collated and smoothed versions of the measurements, usually computed at fixed-time intervals.

There are several well-known methods to determine such speech signal parameters [6]:

- filtering the speech signal by a groupings of logarithmically spaced filters (filterbanks);
- linear predictive coding;
- cepstral coding.

Time alignment and pattern matching (TAPM) algorithms attempt to match a spoken word against a given representation of that word [10]. Time alignment refers to the alignment of acoustic or phonetic events which have been modeled, to those that are “time warped” in the utterance due to changes in speaking rate. The simplest way to recognize an *isolated* word sample is to compare it against a number of stored word templates and determine which is the “best match”. But the rate of speech may not be constant throughout the word, the optimal alignment between a template and the speech sample being a nonlinear function. Dynamic Time Warping (DTW) [10] is an efficient algorithm for finding this optimal nonlinear alignment. The algorithm makes a single pass through a matrix of frame scores while computing locally optimized segments of the global alignment path. The resulting alignment path may be visualized as a low valley of Euclidian distance scores, meandering through the hilly landscape of the matrix. An optimal alignment path is computed for each reference word template, and the one with the lowest cumulative score is considered to be the best match for the unknown speech sample.

The final stage of the recognition process consists of a language-processing module, which attempts to account for the perplexity (the branching factor in the grammar, i.e., the number of words that can follow any given word), in order to select the word using language specific “constraints” or “knowledge”.

Almost all speech recognition systems use a parametric representation of speech rather than the waveform itself as the basis for pattern recognition. These parameters usually carry the information about the short time spectrum of the signal.

Among the most popular representations, produced by various forms of signal analysis, are spectral (DFTC) coefficients, cepstral (CEPC) coefficients, and linear predictive coding (LPC) coefficients.

- Fourier analysis (DFT) yields discrete frequencies over time:

$$S(f) = \sum_{n=0}^{N_s-1} s(n)\, e^{-j\,2\pi \frac{f}{f_s} n}$$

where f_s is the sampling frequency and N_s is the length of the analysis sequence.

- Linear predictive coding yields coefficients of a linear equation that approximate the recent history of the raw speech values.
- Cepstral analysis calculates the inverse Fourier transform of the logarithm of the power spectrum of the signal.

$$c(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \log|\hat{X}(e^{j\omega})|\, e^{j\omega n}\, d\omega$$

$$c(n) = \frac{1}{N_s}\sum_{k=0}^{N_s-1} \log|S(k)|\, e^{\frac{2\pi}{N_s}kn}$$

Neural networks can be taught to map an input space to any kind of output space. A very useful type of mapping is classification, in which input vectors are mapped into one of N classes. A neural network can represent these classes by N output units, of which the one corresponding to the input vector’s class has a “1” activation while all other outputs have a “0” activation.

In order to perform *word level training*, we must define a neural network that classifies a whole word at a time (i.e., its inputs represent all the frames of speech in a whole word, and its outputs represent the *N* words in the vocabulary), so that we can compare the output activations against the desired targets of “1” for the correct word and “0” for all incorrect words.

Learning vector quantization (LVQ) [3], [8] is a pattern classification method in which each output unit represents a particular class. The weight unit for an output unit is often referred to as a reference or *codebook vector* for the class that the unit represents. During training, the output units are positioned (by adjusting their weights through supervised training) to approximate the decision surfaces of the theoretical Bayes classifier. After training, an LVQ network classifies an input vector by assigning it to the same class as the output unit that has its weight vector (reference vector) closest to the input vector.

The architecture of an LVQ network is presented in figure 2.

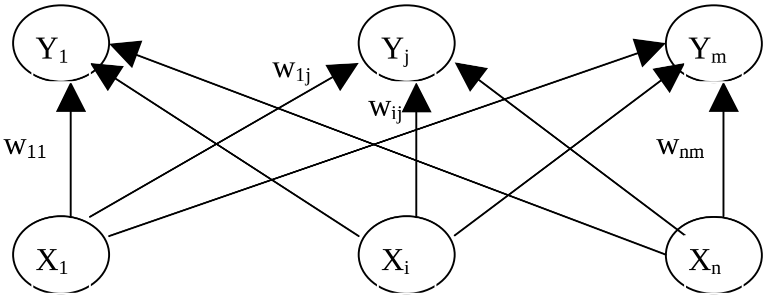

Figure 2 - Learning vector quantization neural network

Related research on speech recognition using neural networks

There are two basic approaches to speech classification using neural networks: static and dynamic. In static classification, the neural network sees all of the input speech at once, and makes a single decision. By contrast, in dynamic classification, the neural network sees only a small window of speech, and this window slides over the input speech while the network makes a series of local decisions, which have to be integrated into a global decision at a later time. Static classification works well for phoneme recognition, but it scales poorly to the level of words or sentences; dynamic classification scales better.

We will briefly present several experiments using the above mentioned approaches.

Static approaches

- Huang and Lippman (1988), demonstrated that neural networks can form complex decision surfaces from speech data. They used a multilayer perceptron (MLP) to discriminate between the vowels of English language. The inputs to the network were the first two formants of the vowels. The classification accuracy was comparable to conventional algorithms, such as k-nearest neighbor and Gaussian classification [2].
- Peeling and Moore (1987) applied MLPs to digit recognition with excellent results. They used a static buffer of 60 frames (1.2 sec) of spectral coefficients. Evaluating a variety of MLP topologies, they obtained the best performance with a single hidden layer with 50 units. The network achieved accuracy near than of an hidden Markov model (HMM) system: error rates were 0.25% versus 0.2% in speaker-dependent experiments, or 1.9%

versus 0.6% for multi-speaker experiments, using a 40-speaker database of digits. In addition, the MLP was typically five times faster than the HMM system.

- Kammerer and Kupper (1988) applied a variety of networks to the TI 20-word database [5], finding that a single-layer perceptron (SLP) outperformed both multi-layer perceptrons and a DTW template-based recognizer in many cases. They used a static input buffer of 16 frames, into which each word was linearly normalized, with 16 2-bit coefficients per frame. Error rates for the SLP versus DTW were 0.4% versus 0.7% in speaker-dependent experiments, or 2.7% versus 2.5% for speaker-independent experiments [2].

Dynamic approaches

- Waibel (1987), demonstrated excellent results for phoneme classification using a Time Delay Neural Network (TDNN) [4], [9]. This architecture has only 3 and 5 delays in the input and hidden layer, respectively, and the final output is computed by integrating over 9 frames of phoneme activations in the second hidden layer. The TDNN was trained and tested on 2000 samples of */b,d,g/* phonemes manually excised from a database of 5260 Japanese words. The TDNN achieved an error rate of 1.5%, compared to 6.5% achieved by a simple Hidden Markov Model (HMM) - based recognizer.
- McDermott and Katagiri (1989), performed an interesting comparison between Weibel's TDNN and Kohonen's Learned Vector Quantization - 2 (LVQ2) algorithm, using the same */b,d,g/* database and similar conditions. The LVQ2 system was trained to quantize a 7-frame window of 16 spectral coefficients into a codebook of 150 entries. As in the TDNN, the distance between each input window and the nearest codebook vector was integrated over 9 frames to make the network shift-invariant. The LVQ2 system achieved virtually the same error rate as the TDNN (1.7% vs. 1.5%), but LVQ2 was much faster during training, slower during testing, and more memory-intensive than the TDNN [9].

A Multiple-Speaker Acoustical Data-Base in Romanian

As we already mentioned, the way to recognize an isolated word sample is to compare it against a number of stored word templates and determine what is the "best match". Therefore, we conceived a database including speech data uttered by 14 speakers (7 males and 7 females). Each of the speakers uttered every of the 10 digits with 3 speaking rates and 3 prosodic coutours as shown in Table 1. As a consequence, our database includes:

14 speakers × 10 digits × 3 speaking rates × 3 prosodic contours = ***1260 utterances***

Prosodic contour	Speaking rate	Gender
Up	Slow	Male
Horizontal	Medium	Female
Down	High	

Table 1 - Constructing the Data-Base

The Multiple-Speaker Acoustical Data-Base (MSADB) was recorded with a close-speaking microphone in a room with normal ambiental noise. Recordings were digitized at a sampling rate of 8 kHz. A 32 msec Hamming Window was used to weight samples toward the center of the window. This characteristic, coupled with the overlapping analysis discussed next, performs an important function in obtaining smoothly varying parametric estimates.

Window duration controls the amount of averaging, or smoothing. The frame duration and and window duration together control the rate at which parameters values track the dynamics of the signal [7].

We performed an overlapping analysis: with each frame, only a fraction of the signal data changes. The amount of overlap controls how quickly paramaters can change from frame to frame.

$$\%\,\text{Overlap} = \frac{T_w - T_f}{T_w} \times 100\% = \frac{32-16}{32} \times 100\% = 50\%$$

We choosed $T_w = 32\,\text{msec}$ and $T_f = 16\,\text{msec}$ as powers of 2 to reduce the computation. Moreover, this combination correspond to as high as 50% overlap, that allows for reducing the amount of noise.

The total length of every utterance was 1 sec, long enough for the longest spoken word. Briefer words were padded with zeros and positioned randomly in the 1 second interval.

We subsequently performed LP and cepstral analyses to produce 10 LP-coefficients and 10 cepstral coefficients, respectively, every 16 msec. Therefore, the data-base includes 60 groups of 10 LP-coefficients and 60 groups of cepstral coefficients.

Finally, we have to note that the window length of 32 msec gives a frequency resolution of 1/32 = 31.25 Hz, and having a sampling frequency of 8 kHz, only 4 kHz are reliable.

Performance analysis regarding an independent-speaker speech recognition task, using a DTW-ANN hybrid system

We studied the following representations (with a 16 msec frame rate and a 32 msec Hamming Window in each case):

- LPC-10: 10 linear prediction coefficients per frame;
- CEPC-10: 10 Fourier Transform derived cepstral coefficients,

in conjunction with the Data-Base, which we called “Extended Data-Base”, and with a truncated version of it, which we called “Reduced Data-Base”.

We performed three sets of experiments under rather different experimental conditions. In the first set, we used a classical DTW algorithm along with 10 linear prediction coefficients. In the second set, we used a classical DTW algorithm along with 10 Fourier Transform derived cepstral coefficients, and in the third one we used a LVQ+DTW algorithm along with with 10 Fourier Transform derived cepstral coefficients. The scores are shown in tables 2-6.

Digit	0	1	2	3	4	5	6	7	8	9	General score
Score (%)	90	73	76	91	90	96	71	67	90	65	81.0

Table 2 – LPC-10 / Extended Data-Base

Digit	0	1	2	3	4	5	6	7	8	9	General score
Score (%)	90	71	72	91	89	95	67	67	87	65	79.5

Table 3 – LPC-10 / Reduced Data-Base

Digit	0	1	2	3	4	5	6	7	8	9	General score
Score (%)	95	89	83	87	94	99	85	74	89	90	88.3

Table 4 – CEPC-10 / Extended Data-Base

Digit	0	1	2	3	4	5	6	7	8	9	General score
Score (%)	94	88	81	84	91	98	84	71	87	90	87.1

Table 5 – CEPC-10 / Reduced Data-Base

Digit	0	1	2	3	4	5	6	7	8	9	General score
Score (%)	97	94	94	91	97	99	89	90	92	96	93.9

Table 6 – LVQ+DTW / Extended Data-Base

Conclusions and further work

The key features of our approach may be summarized as follows:

1) We show that a carefully conceived database may dramatically improve the overall performance of the system (including a reduced size database). Based on a previous work describing the feature modification in Romanian spoken words (typically a change in speaking rate, which leads to a time axis modification, and two main variations in prosodic contour of the ten digit pronunciation), we choose speech data uttered by 14 speakers (7 males and 7 females), each speaker uttered each word 9 times.

2) For the classical DTW-based system, we utilized two different sets of features for inputs: 10 PARCOR coefficients obtained from a standard linear-prediction (LP) analysis (with a 8 kHz sampling rate and a uniform quantization of 16 bit per sample), and 10 Fourier Transform-derived cepstral coefficients. The results obtained with the cepstral coeficients are significantly better than those provided by using the LP coefficients.

3) The third set of experiments was made with a neural-based system, using the cepstral coefficients. The experiments clearly demonstrates that the combination of DTW for aligning the exemplars in time, and the neural network learning process for distinguishing the exemplars in frequency and creating discriminatory template patterns for each of the words in vocabulary, provide the basis for an effective speech recognition system.

A separate network is created for each of the ten digits. Initially, the sequence of spectral representation are simply random numbers. Each network is then tested with an exemplar of all other words in vocabulary. If the right network corresponding to the exemplar gives the lowest DTW score, no reference pattern is modified. If an incorrect network gives the lowest DTW score, then the reference patterns are changed in two networks. Each of the corresponding spectral parameters is shifted further from the exemplar in the network that wrongly received the best score and closer to the exemplar in the network that did not achieve the best score. The test results show a rate of recognition of 94%.

We recently started a set of experiments of word recognition in noisy environment, basically words uttered over the telephone network. It is known that the two main sources of variation in the telephone channel are additive noise and a changing channel frequency response between different telephone lines. We used few more words in our testing database, which were spoken over a telephone line. Our results seem to be close to those above mentioned, with two improvements for the described system:

- the estimation of cepstral coefficients using some derivative of the coefficients;
- we performed a cepstral mean normalization which helps in reducing the variability of the data and allows a simple channel and speaker normalization.

References

[1] Burileanu, D., S. Marinescu, *A Neural System Architecture for a Test-to-Speech System in Romanian Language*, Proceedings of the 7th International Conference on Signal Processing, Application & Technology, Boston, Oct. 7-10, 1996, pp. 1363-1367.

[2] Burileanu, C., et al, *Prosodic Analysis of the Romanian Language Vocabulary*, Proceedings of Communications ‘96, Bucharest, Nov. 27-29, 1996, pp. 482-487.

[3] Fausett, L., *Fundamentals of Neural Networks – Architectures, Algorithms, and Applications*, Prentice-Hall, Englewood Cliffs, New Jersey, 1994.

[4] Morgan, D., C. Scofield, *Neural Networks and Speech Processing*, Kluwer Academic Publishers, 1994.

[5] Pawate B., P. Robinson, *TMS320C2x and TMS320C5x Implementation of an HMM-Based, Speaker-Independent Speech Recognition System*, Application Report, Texas Instruments, Dallas, Texas, 1996.

[6] Picone, J., *Signal Modeling Techniques in Speech Recognition*, Proceedings of the IEEE, Vol. 81, No. 9, Sept. 1993, pp. 1215-1247.

[7] Rabiner, L., R. Schafer, *Digital Processing of Speech Signals*, Prentice-Hall, Englewood Cliffs, New Jersey, 1978.

[8] Rabiner, L., B-H. Juang, *Fundamentals of Speech Recognition*, Prentice-Hall, Englewood Cliffs, New Jersey, 1993.

[9] Tebelskis, J., *Speech Recognition using Neural Networks*, PhD Thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania, 1995.

[10] Tubach, J.P., Éditeur principal, *La parole et son traitement automatique,* Masson, Paris, 1989.