Neuro Nîmes 90

TDNN-Extracted Features

Xavier Driancourt *, Léon Bottou

L.R.I. - Bat 490
Université de Paris Sud
91405 Orsay - FRANCE

tel: 69 41 63 90
fax: 64 46 19 92

ABSTRACT

Time Delay Neural Network (TDNN) is a technique, derived from MLP, which performs a time invariant processing in its lowest layers. This time invariant processing may be extracted from the network, in order to code the speech for an other classifier such as Dynamic Time Warping (DTW). The resulting hybrid system shows improved performances, with respect to both techniques used in isolation.

This paper describes this technique, gives results on a multi-speaker, isolated word recognition task, and discusses its advantages.

RESUME

Le réseau neuronnal à délai (TDNN), est une technique dérivée du perceptron multicouche (MLP), qui effectue un traitement invariant dans le temps dans ses couches inférieures. Ce traitement invariant peut être extrait du réseau, afin de coder la parole pour un autre classifieur comme l'alignement temporel par programmation dynamique (DTW). Le système hybride résultant se montre plus performant que les deux techniques utilisées isolément.

Cet article décrit cette technique, donne des résultats pour un problème de reconnaissance multi-locuteurs de mots isolés, et commente ses avantages.

KEYWORDS

Feature Extraction, Time-Delay Neural Network, Dynamic Time Warping, Speech Recognition.

ACKNOWLEDGEMENTS

This work was mainly developped in Pygmalion, ESPRIT project number 2059. Grants from DRET to support both authors are gratefully acknowledged(XD: 89/1591, LB: 87/808/19).

TDNN-Extracted Features

1 Principle

Neural networks are now well known as powerful learning tools in a variety of tasks related to automatic speech recognition problems, where they have achieved encouraging results (Bridle, 1984) (Prager, 1986) (Kohonen, 1988) (Watrous,1987) (Bourlard,1988).

However, speech recognition is a difficult task. Problems such as speaker independence, continuous speech, noisy environment, vocabulary size, signal variability, coarticulation effect, etc... remain partly unsolved. All the existing methods successfully overcome some of the difficulties but fail solving the others. Our aim here was to compare neural networks, and more precisely Multi-Layer Perceptrons (MLPs), to classical methods on a widely studied and well mastered task for today's speech recognition systems.

The first MLPs developed for speech recognition tasks were fully connected, with usually one hidden layer. Such networks have performed correctly on various small sized problems: (Elman, 1987) (Lippmann, 1987) (Lubensky, 1988). However, on real sized data, those architectures are very inefficient. The number of training examples is too small to succeed in specifying the whole set of parameters in the network. As a consequence, generalization is relatively poor, unless a large enough set of examples has been available for training. But in that case, learning would take very long.

One way to reduce the complexity of the network is to use local connections, i.e. local fields. Hidden units are thus assigned local interest tasks, i.e. feature extraction. Since we cannot ensure very precise time alignment of the signal, and since also the speech signal can be significantly stretched in time without altering its meaning, we would like to design time invariant feature extractors. This led to the idea of building networks with position-independent local fields or so-called time-delay neural networks or TDNN (Waibel, 1987) (Lang, 1988), described in figure 1.

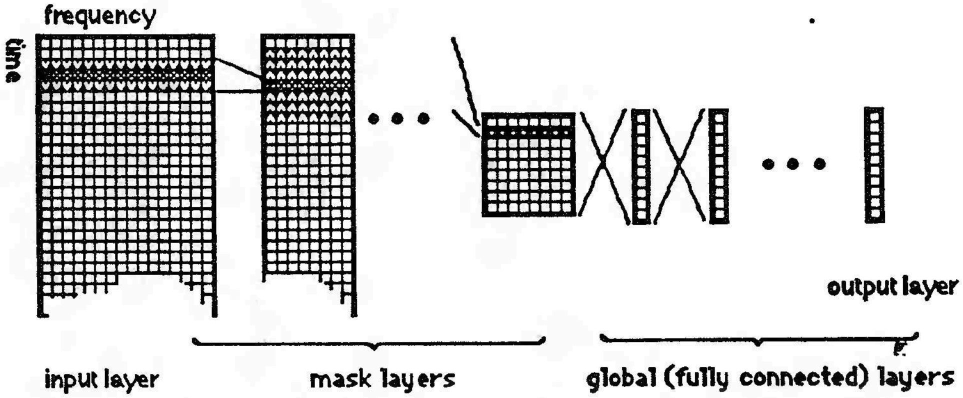

Fig ure1: Time Delay neural Network (TDNN)

However part of the problem remains. The feature extractors have a reduced set of parameters, but after them comes the classifier of the TDNN, which is made of fully connected layers. This classifier has no specific ability to deal with the temporal dimension, and contains a considerable set of parameters (independent weights), which requires a huge database for proper training.

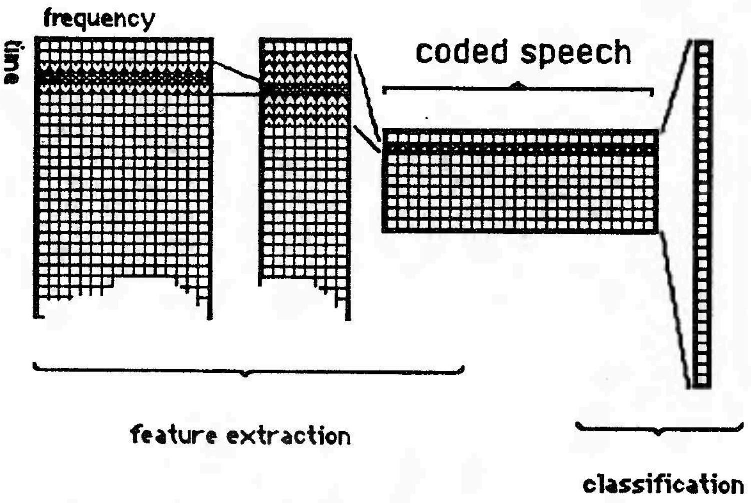

Figure 2: TDNN = Feature Extractor + Classifier

Actually, using fully connected layers as classifier is not required: various classifiers may be used on the extracted features as well, after the network is trained. General classifiers with reduced sets of parameters (such as LVQ) could be tried, but it also sounds interesting to use specific classifiers (such as DTW, HMMs or recurrent networks), to take into account the temporal dimension.

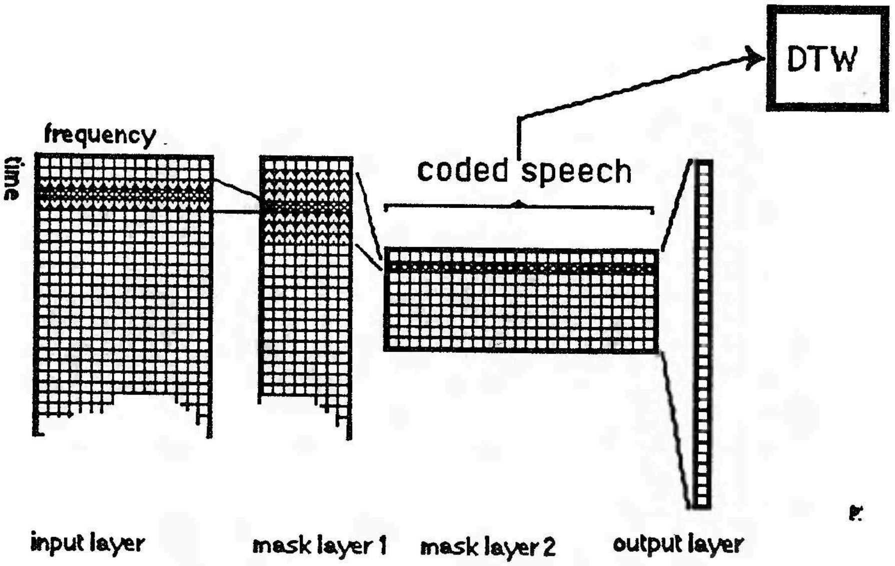

Figure 3: Use of DTW on Extracted Feature

We propose in this paper a validation of the concept of feature extraction on an Isolated Word Recognition (IWR) Task. We describe a first experiment of IWR recognition with simple TDNNs. Vector quantization on extracted features is explored in the third section. A second, larger experiment of IWR is described in the fourth part. The fifth section contains results of Dynamic Time Warping (DTW) applied to extracted features.

2 First IWR experiment with TDNN (IWR1)

2.1 Database and preprocessing

A speech data base, in French, has been elaborated at LIMSI. In the experiment reported in §2, we have only used part of the data base, namely the utterances of the 10 digits by 26 speakers, male (40%) and female. Each of the speaker pronounced each digit once.

The signal has been processed in the following way, classically used at LIMSI (Gauvain, 1986). The speech signal, from the microphone in a quiet room, has been filtered at 5 KHz through a low-pass filter, then sampled at 10 KHz with a 12 bits A/D converter. High frequency amplitudes are increased at 6 dB per octave. A DFT is applied on successive 25.6 ms time frames, overlapping by 12.8 ms. Thus 128 energy spectra values are generated in the 0-5 K Hz frequency domain. A Bark scaled 16-channels filterbank is then simulated by averaging on triangular frequency windows. The energy spectra are then log-compressed.

This processing thus results in coding the speech signal into sixteen eight bits values per 12.8 ms time frame. The digits, in French, are all monosyllabic words, except "zéro". The ten digits used in this experiment lasted between 15 and 61 time frames.

We defined a 16 speakers learning set, both male and female The remaining ten speakers were used as a test set. Speakers were assigned to the two sets using alphabetical order, thus independently of any phonetical clue.

2.2 Experiment

In a preliminary experiment, we trained the network without randomly shifting the training data in the input window. We easily got a perfect recognition of the training set, and a peak 95% performance on the test set.

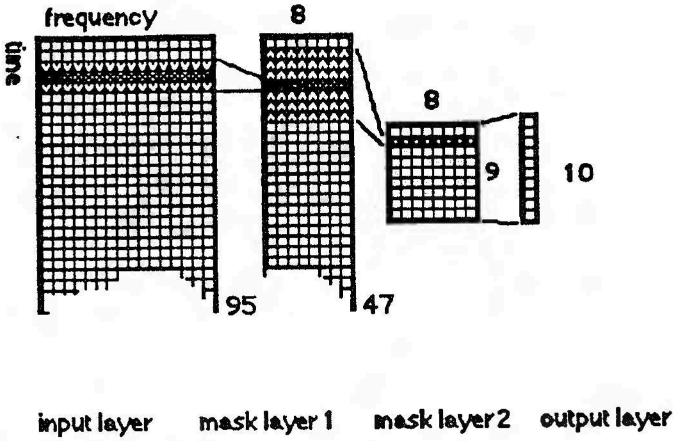

Figure 4: TDNN for Digit Recognition (IWR1)

Unfortunately, although the hidden layers of our network are designed for shift invariance, the last weight layer is not. We threfore tested the previous weight configuration on the same patterns that were used for training, this time randomly shifting them. When the test utterances were randomly shifted in the 0 to 51ms range, the performance sank to 89%. With even more shifted data, in the 0 to 102ms range, the network achieved a poor 70% performance.

In order to get a real shift invariance capacity, we then decided to train the network with the randomly shifted data described above (shifted by 0 to 128ms). Moreover, such a random shift simulates a poor word segmentation, and is in fact much closer to real speech tasks than perfectly aligned test and training data.

The best run produced a network able to correctly classify 99.21% of the training patterns and 99% of the test patterns (i.e. 1 only unrecognized utterance out of the 100 test utterances). Unfortunately, this speech database is clearly too small for statistically validating such performances. A LIMSI dynamic time warping system has been ran on the same data, and also achieved one error only.

Further experiments were achieved, aiming at improving the statistical significance of these results, and also at getting a better knowledge of the connectionist IWR behaviour. We decided to lower the performances by using less speakers for training, and also to use four training and test set within our single small digits database.

Reducing the number of training speakers to 10 strongly degrades the network performance. The network achieved 4.8% errors over 640 test patterns.

Comparisons have been made with the DTW developped at the LIMSI. This system achieved 1.9% errors only on the same data sets, thus showing a better ability to learn with less training data.

Analysis of the feature extraction performed by the network are provided in §3.

3 Vector quantization on extracted feature

We have used the IWR network described in §2, trained with 16 speakers on the french digit problem (IWR1). In order to test the feature extraction ability in our network, we have studied the activities of the last hidden layer cells: there are 6x8 such cells, whose activities can be viewed as 6-vectors in an 8-dimension space. Each of these vectors represents an encoding of the signal over about 100 milliseconds. They are analyzed through a k-means clustering technique. The result is a description of each digit as a sequence of labels. We reproduce below (figure 7) the encoding found for some digits, uttered by four different speakers, using this clustering technique, with eight reference vectors.

00	UN	1	1	4	8	8	8	10	DEUX	7	2	2	8	8	8
100	UN	3	1	4	8	8	8	110	DEUX	7	2	2	8	8	8
200	UN	1	1	4	8	8	8	210	DEUX	7	2	2	8	8	8
300	UN	1	1	4	8	8	8	310	DEUX	7	2	2	8	8	8
400	UN	3	1	4	8	8	8	410	DEUX	7	5	5	2	8	8
500	UN	1	1	4	8	8	8	510	DEUX	7	2	2	8	8	8
600	UN	1	1	4	8	8	8	610	DEUX	7	7	2	2	8	8
700	UN	3	1	1	4	8	8	710	DEUX	7	2	2	2	8	8
800	UN	1	1	4	8	8	8	810	DEUX	7	5	5	8	8	8
900	UN	1	1	4	8	8	8	910	DEUX	7	2	2	8	8	8

20	TROIS	7	3	4	4	8	8	30	QUATRE	2	1	4	4	8	8
120	TROIS	8	3	3	4	8	8	130	QUATRE	7	4	4	4	8	8
220	TROIS	3	3	4	8	8	8	230	QUATRE	7	1	4	4	8	8
320	TROIS	7	3	3	4	8	8	330	QUATRE	7	1	4	4	8	8
420	TROIS	7	3	4	8	8	8	430	QUATRE	7	1	4	8	8	8
520	TROIS	7	3	3	8	8	8	530	QUATRE	2	4	4	8	8	8
620	TROIS	3	3	3	4	8	8	630	QUATRE	7	1	4	4	4	4
720	TROIS	8	3	3	4	8	8	730	QUATRE	7	4	4	4	4	8
820	TROIS	3	3	4	8	8	8	830	QUATRE	7	1	4	4	8	8
920	TROIS	3	3	4	8	8	8	930	QUATRE	7	1	4	4	8	8

40	CINQ	6	3	2	8	6	8	50	SIX	6	6	6	6	6	8
140	CINQ	6	7	1	7	6	8	150	SIX	6	5	6	8	8	8
240	CINQ	6	8	1	2	6	8	250	SIX	6	6	6	6	6	8
340	CINQ	6	7	1	1	8	8	350	SIX	6	6	6	6	6	6
440	CINQ	6	7	1	2	6	8	450	SIX	6	6	6	6	6	8
540	CINQ	6	3	2	8	8	8	550	SIX	6	5	6	8	8	8
640	CINQ	6	6	1	2	7	8	650	SIX	6	6	6	6	6	6
740	CINQ	6	3	1	8	7	8	750	SIX	6	6	6	6	3	4
840	CINQ	6	1	2	8	6	8	850	SIX	6	6	6	6	6	8
940	CINQ	6	3	1	8	6	8	950	SIX	6	6	5	6	6	8

60	SEPT	6	7	7	8	8	8	70	HUIT	5	6	4	8	8	8
160	SEPT	6	7	2	8	8	8	170	HUIT	5	6	4	8	8	8
260	SEPT	6	7	2	8	6	8	270	HUIT	5	6	4	8	8	8
360	SEPT	6	7	2	8	8	8	370	HUIT	5	5	6	8	6	8
460	SEPT	6	7	2	8	8	8	470	HUIT	7	5	6	8	8	8
560	SEPT	6	2	4	8	8	8	570	HUIT	5	5	4	8	8	8
660	SEPT	6	7	2	8	8	8	670	HUIT	5	5	5	8	6	8
760	SEPT	6	6	7	8	3	4	770	HUIT	5	5	8	3	8	8
860	SEPT	6	6	2	7	8	8	870	HUIT	5	6	4	8	6	8
960	SEPT	6	7	2	8	6	8	970	HUIT	5	5	4	8	8	8

80	NEUF	7	2	1	8	8	8	90	ZERO	5	2	3	8	8	8
180	NEUF	2	1	6	8	8	8	190	ZERO	5	2	3	3	8	8
280	NEUF	7	2	2	8	8	8	290	ZERO	6	5	3	8	8	8
380	NEUF	2	1	6	8	8	8	390	ZERO	7	2	3	3	3	8
480	NEUF	5	1	4	8	8	8	490	ZERO	7	5	2	3	8	8
580	NEUF	5	2	1	8	8	8	590	ZERO	7	5	2	3	8	8
680	NEUF	5	1	2	6	6	8	690	ZERO	7	5	2	3	3	8
780	NEUF	5	1	6	8	3	8	790	ZERO	5	2	3	3	8	8
880	NEUF	5	2	2	6	6	8	890	ZERO	7	5	2	3	3	8
980	NEUF	2	1	4	8	8	8	990	ZERO	7	2	2	3	8	8

FIGURE 5: Encoding Generated by a TDNN for the French Digits Problem (IWR1)

In figure 7, each line contains the number of the utterance, the corresponding french digit, and the signal encoding, after a K-means into 8 clusters. The eighth cluster is clearly the prototype of the code for silence. Further work is needed to get more precise codings: mask sizes probably have to be redefined.

Notice that the validity of the coding can be tested by the performances of the net: we know that these informations are sufficient for providing a good digit identification, comparable to our IWR network performance . The basic system for using a network for obtaining a codebook is represented in figure 8:

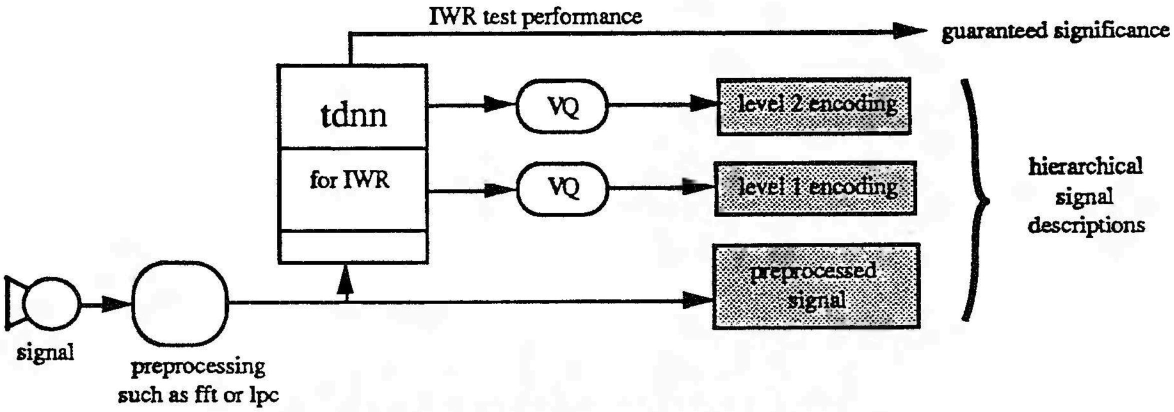

FIGURE 6: TDNN Based Encoding Device.

However, many questions remain to be solved:

- What is the loss of performance implied by the Vector Quantization step ?
- What are the effects of various preprocessing techniques on the quality of the encoding ?
- How could such a system be extended to a larger database ?

This first experiment shows that some interesting features are extracted from the network. These features give reasonable results when passed through a vector quantization algorithm, and could be used by any algorithm needing a codebook (HMMs).

The major critics which can be made about this experiment is the weak amount of data used: are the results reliable? Part 4 proposes a larger experiment with TDNNs on a larger database. Part 5 describes an experiment on the direct use of extracted features (without vector quantization) as input to a DTW..

4 Second experiment of IWR by TDNNs (IWR2)

A common database in English, German, French, Italian and Spanish was developed in Pygmalion, ESPRIT project nb 2059. Each is composed of thirty words (ten digits and twenty command words), pronounced ten times by each of ten speakers in an office environment.

A demonstrator of IWR and low-level speech processing was designed on the French database. The signal processing used was the same as described in §2.1.

Using the same network architecture as in §2.2 appears difficult, because the classification task on the top of the TDNN is performed by a simple linear classifier, whereas the database of Pygmalion has 30 classes, which should require a more complex classifier. Furthermore the number of degrees of freedom (independant weights) will increase, which should require a huge database for significant training.

Consequently, various TDNN were trained.The best network for this task was found to have the following architecture:

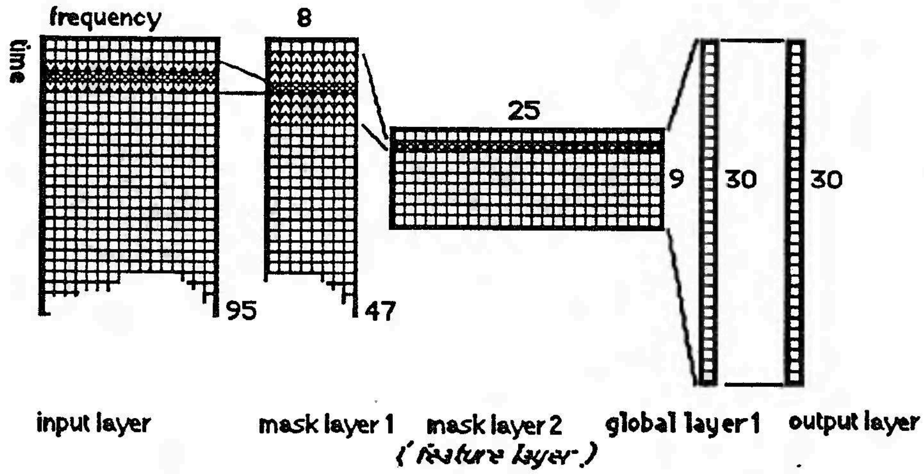

Figure 7: TDNN for Command Words Recognition (IWR2)

i.e., namely: input layer (IL), first mask layer (ML1), second mask layer (ML2), first global layer, (GL), output layer (OL).

The first mask connects one frame of ML1 to three frames of IL. The shift is 2 frames of IL for one frame of ML1. The second mask connects one frame of ML2 to seven frames of ML1. The shift is five frames of ML1 for one frame of ML2. The biases are free, i.e. they are not shared: the filters produced by the networks are not fully shift-invariant.

The network performs 99.5% on the training database, and 96.5% on the test database.

As previously said, various architecture were tested. The following misbehaviour can be reported: networks wit one fully connected layer performed 1 or 2% worse; networks with shorter masks performed 2% or 5% worse; net with more compact codings (output of each mask) performed 2% worse. Some explanations can be propose classification task between thirty classes is too complex to be easily done by a single-layer (almost linear) clas the masks perform better if they analyse a temporal window slightly larger than a phoneme.

Here are the overall characteristics of the selected network:

Layer	Cells	Connections	Weights	Biases
Input 95 * 16	1520			
		18048	384	376
Mask 1 47 * 08	376			
		12600	1400	225
Mask 2 09 * 25	225			
		6750	6750	30
Global 1	30			
		900	900	30
Output (Global 2)	30			
Total	2181	38298	9434	661

Figure 8: characteristics of the network

Mask layers collect most connections, although these connections imply a reasonable amount of weights.

Oppositly, the two global layers collect not so many connections, but these connections imply a considerable number of weights, which require a large database for significant training. Furthermore, fully-connected classifiers have no special ability to deal with the temporal dimension. An unappropriate classifier with too many degrees of freedom in respect to the size of the database.

In order to improve the system, more fitting classifier should process the speech encoding obtained from mask layer 2. Such an experiment is described in §5.

5 Use of DTW on extracted features

Feature extraction has been defined in §1 and explored through VQ in §4. It looks attractive, since it provides an intelligent filtering of the data. Figure 7 suggests that DTW should give good results when used on extracted features. Part 5 applies this idea on the Pygmalion database, already described in §4, aiming at validating the usefulness of the feature extraction (and, incidentaly, overcoming the problems raised in §2 and §4 concerning the fully connected layers performing the classification in TDNNs).

We trained several TDNNs on the IWR problem, and used their code in a simple DTW according to the principle described in §1 (Figure 3), thus obtaining as many hybrid systems as trained TDNNs.

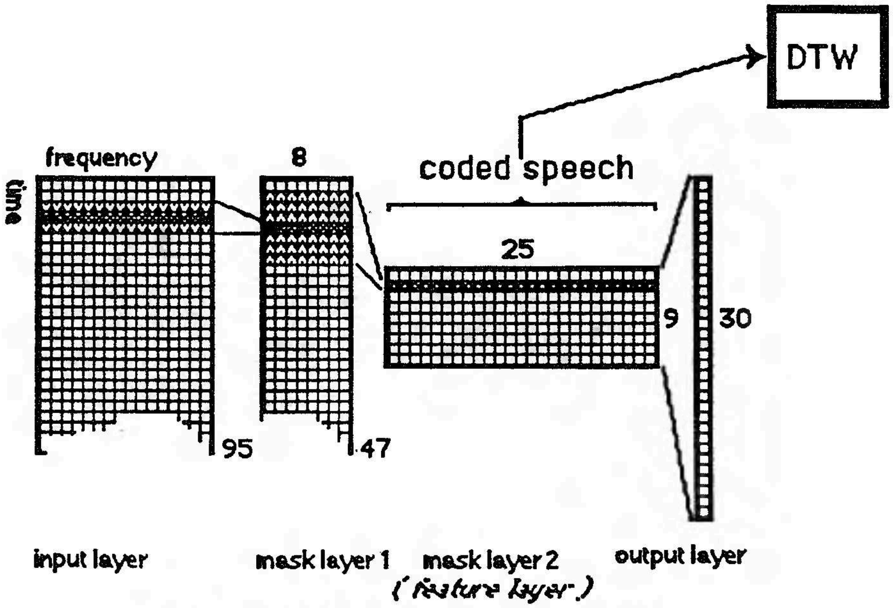

Figure 9: Best Hybrid System (IWR3)

The best hybrid system (Figure 9) obtained performed a 98.5% recognition rate on the test database, to compare with 96.5% obtained with the best isolated TDNN. It is an important improvement, since the failure rate was divided by two or three.

It validates the usefulness of feature extraction. Actually many other algorithms could be used as classifiers, including more sophisticated DTWs.

It must be noticed that feature extraction is not performed the same way by different TDNNs. The best TDNN for IWR does not provide the best code for DTW, although it can be thought that the extracted features are at least as much complete in the best TDNN for IWR as in the best TDNN for DTW. The same architecture for feature extractors may lead to codings with different properties, or looking not the same way, due to the differences between the classifiers used on the top of the network during the training phase.

The advantages of this technique are numerous:

- improved recognition rate with respect to both tecniques
- possibilities of continuous-speech recognition (which can not be done by TDNNs)
- computation time reduction for DTW due to the compaction of data

Several potential advantages should be explored:

- if the vocabulary of the TDNN is phoneticaly well-balanced, one could expect to extend the vocabulary of the hybrid system just as it can be done with a DTW
- gain in speaker-independency could be obtained, compared to isolated DTW

6 Conclusion

We described experiments of Isolated Word Recognition with TDNNs. We proposed two experiments validating the concept of feature extraction: a vector quantization and a dynamic time warping applied on the last mask layer of the TDNNs. Vector quantization enables the coding of words as sequences of reference vectors. These sequences look consistent, and suggest that dynamic time warping should be efficient when used on the extracted features. Dynamic TimeWarping gave a significant improvement of the performances of the IWR system with regard to single TDNNs. Once the concept of feature extraction validated, any combination of TDNN and classifier may be considered, including discrete HMMs and recurrent networks.

7 References

Bottou : Speaker-Independant Isolated Digit Recognition: Multilayer Perceptron vs. Dynamic Time Warping
L.Bottou, F. Fogelman Soulie, P. Blanchet, J.S. Lienard
to appear in: Neural Networks 1990

Bourlard : How Connectionnist Models Could Improve Markov Models For Speech Recognition
Herve Bourlard
to appear in: Proceedings of the Internationnal Symposium on Neural Networks For Sensory and Motor Systems, march 1990, DusselDorf, F.R.G., Ed R. Eckmiller

Elman, Zipser : Learning the hidden structure of speech
J.L Elman, D. Zipser
Tech. Report, Univ. of California, San Diego, (feb. 1987)

Gauvain : A syllable-based isolated word recognition experiment
J.L. Gauvain
In Proc. IEEE ICASSP-86, (1986)

Lang : The development of TDNN architecture for speech recognition
K. Lang, G.E. Hinton
Tech. Report CMU-CS-88-152, (1988).

Le Cun : A theoretical framework for back-propagation
Y. Le Cun
In "Connectionist Models: a summer school", D.Touretzky (ed), Morgan-Kaufmann (1988)

Lippmann : Neural-net classifiers useful for speech recognition
R.P. Lippman Ben Gold
IEEE First Int. Conf. on Neural Networks, San Diego, IEEE catalog n° 87TH0191-7, IV-417-426, (1987)

Lubsensky : Learning spectral-temporal dependencies using connectionist networks
D. Lubsensky
Proceedings ICASSP 88, S-Vol.1, 418-421, (1988)

Rumelhart : Learning internal representations by error propagation
D.E. Rumelhart, G.E. Hinton, R.J. Williams
In "Parallel distributed processing", D.E. Rumelhart, J.L.McClelland eds, MIT Press, vol 1, 318-362, (1986)

Waibel : Phonem recognition using time delayed networks
Waibel A., Hanazawa T., Hinton G., Shikano K., Lang K
IEEE trans. on Accoustics, Speech and Signal Processing (march 1989)