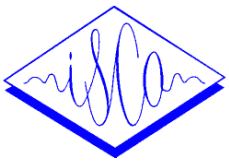

3rd European Conference on
Speech Communication and Technology
EUROSPEECH'93
Berlin, Germany, September 19-23, 1993

NEURAL TIME WARPING

Bruno Apolloni, Dario Crivelli, Marco Amato

Laboratorio Laren, Dipartimento di Scienze dell' Informazione
Via Comelico 39/42, Tel +39.2.55006284 Fax +39.2.55006276
20133 Milano,Italy

ABSTRACT

We try to capture subsymbolically through a recurrent neural network the structure of a word utterance in terms both of the dynamic properties of the process generating it [*positive knowledge*] and of a companion hidden process which declares in its final state the inconsistency of that utterance with some deceiving candidate generating processes [negative *knowledge*].
Namely, we work on a bench of neural networks. Each one was trained on a set of template utterances of a same word for generating a trajectory of the Mel-cepstral parameters which get close to that of the training word and an adviser signal which is frustrated in its growing when the net runs on some misleading words.
We use the generalization capability of that network for adapting the output trajectory to the utterances under recognition. This gives rise to a *neural time warping* which stretches or compresses the template signal in function of the actual utterance, unlike the usual time warpers which work on the current vs template utterance.
A proper two-phase training strategy is developed. Classifying the word with the label of the better warping network not rejected by the adviser signal gives rise to a rate of success about 96% on a speaker independent vocabulary of the ten digits.

KEY WORDS: Time Warping, Speech Recognition, Recurrent Neural Networks.

1. INTRODUCTION

In the frame of supervised algorithms a general paradigm for speech recognition is made from the following items: a bench R of template utterances represented as a sequence of points in a multidimensional space of parameters (in what follows 9 Mel-cepstral parameters); a current word u; a distance measure between $r \in R$ and u; a competition rule for attributing u to an item of R [1].
Warping is the way for defining the distance or, from another common viewpoint, a preprocessing of the signal before computing an euclidean distance between that and the template. Actually, both the length of the voiced sequence and the distribution of the sound along this depends on the pronunciation style of the different speakers, but it also depends on accidental conditions, such as context, tiredness, stress, etc. However the Dynamic Time Warpers (DTW)[2],the most spread warpers, focus on the temporal matching between the two signals, compressing or stretching the frame sequence of u, but doing nothing on their value.
Hidden Markov models may be viewed as a time warping of the hidden states of the Markov process which generates the trajectories (one transition of the Markov process each signal frame) of the uttered word parameters [3]. As the single state can produce signals of different intensity, the above weakness of the DTW model is overcome. Actually, the evolution of the hidden states stands for a dynamic model synthetyzing the evolution of the signal, possibly in a way which is intrinsically independent from the speaker.
Agreeing with this approach, we propose here to pass from the symbolical identification of the model underlying an utterance to the subsymbolical following of its tracks with the time in the (Mel-cepstral) parameters space, together with the exploiting of an associated adviser process which appraises the compatibilility of the model As it concerns the warping, this allows us to:
1. undertake a process more complex than a Markov one
2. visualize the differences between the observed and modeled signals.
In regard to the definition of the metric, our method allows us to
3.take into account not only the attraction (via the euclidean distance) of the model to but even the repulsion (of the adviser) from the signal.

2. THE NEURAL TIME WARPER

We used a very simple network (see fig. 1) made up of a nine (one per Mel-cepstral parameter) nodes input and a nine nodes output layers and one hidden layer. The connections lay-out is that of a MultiLayer Perceptron (MLP), but the hidden nodes are one each-other and itself connected and so constitute the *motor* of the network. A further hidden neuron **a** receives connections from all the input and hidden neurons and from itself, its state being the adviser signal.

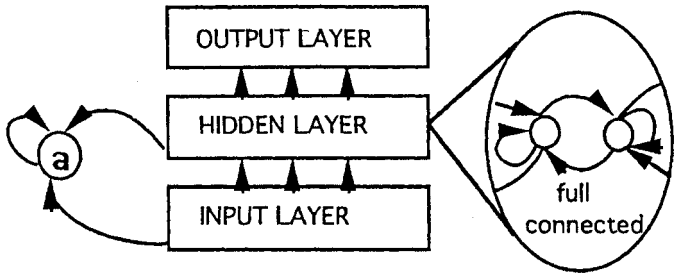

Fig. 1: Recurrent neural network lay-out

Now, for a connection weights matrix W (where w_{0j} connects the dummy neuron with constant state = 1 to neuron j), the evolution of the state vector $\underline{X}(t) \in [0,1]^n$ of the network on the basis of the input vector $I(t) \in [0,1]^n$ and of the previous state vector $\underline{X}(t-1)$ is so ruled [4]:

$$x_j(t) = \begin{cases} i_j(t) & \text{if j is an input neuron} \\ f_j(net_j(t)) & \text{otherwise} \end{cases}$$

being

$$f_j(z) = \frac{1}{1+e^{-\beta z}} \quad \text{with } \beta \text{ a real slope parameter and}$$

$$net_j(t) = \begin{cases} \sum_{k \in \Lambda_h} w_{kj} x_k(t) & \text{if j is an output neuron} \\ \sum_{k \in \Lambda_i} w_{kj} x_k(t) + \sum_{r \in \Lambda_h} w_{rj} x_r(t-1) & \text{if j is a hidden neuron} \\ \sum_{k \in \Lambda_i} w_{kj} x_k(t) + \sum_{r \in \Lambda_d} w_{rj} x_r(t-1) & \text{if j is the adviser} \end{cases}$$

with the initial condition $net_j(0) = c_{j0}$, being Λ_i and Λ_h the sets of the input and hidden neurons, respectively, $\Lambda_d = \Lambda_h \cup \mathbf{a}$ the set of the dynamics neurons and c_{j0} proper constants.

The attitude of synthetizing and properly representing the input claimed for the inner layers of an MLP here translates in the shape of the time evolution that the hidden neurons are producing on their states. Namely we decided of assign to each hidden neuron an inner episode of the evolution of our nine-dimensional trajectory consisting of a single smooth ramp between the levels 0 and 1. We assumed that a small number of these internal tracks, each one evolving in a limited segment of the signal duration and almost non-overlapping the others (see fig.2.a), are enough for coping with the parameters trajectory, considering its minor ripples as remnant follow-out of these principal modes. The tentative addition of further nodes did not increase the modalities of the trajectories but made some nodes evolve in pairs (see fig.2.b).

Actually we force these ramps in a temporal sequence in order to produce essentially *m* switches in the tracks, their jumping being delayed and modulated along the tracks of the single Mel-cepstral parameters by the hidden-output connections.

As illustrated in greater detail in previous papers [5,6] the learning procedure consists of two steps:

1. First of all, building up of the *motor* of the network, without input.

As it is well known, the feed-back connections generate an inner dynamics of the network which is trainable on complex trajectories of the state vector [7]. We trained 3 to 7 hidden neurons - 9 output neurons nets without input (i.e. with input-hidden connections interrupted) to follow in the mel-cepstral space the trajectories of prototypal representatives of the utterances of the ten digits. As mentioned before, we call *motor* the set of the hidden nodes since it actually generates the dynamics of the network.

The learning algorithm is backward [8] and some rules of thumb are used for initializing the connection weights.

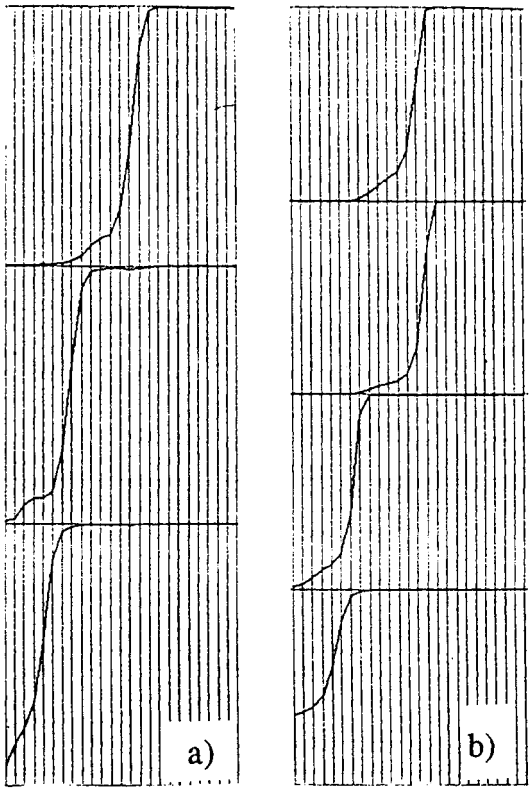

Fig.2 Ramps with time of the states of the hidden neurons trained without input
a. well done non-overlapping tracks
b. redundant trscks from an oversized hidden layer

2. Once the *motor* is trained, the second stage of our learning procedure consists in re-establishing the connections between the input and hidden layer. This allows us to reach two targets: i)the network is trained to closely follow slight variations on the learned mean

trajectory which might be considered as items of the same family and then of the same semantical meaning; ii)the adviser neuron is forced to be down when the motor is incompatible with some cheating trajectories in input. For achieving this task we retrain the network, ever via the backward algorithm, starting from a random initialization of the new connections and maintaining the achieved values of the parameters on the rest of the network. The training set is constituted by positive examples,which have to be reproduced, and negative examples, which have to be removed from the network under training. The former are constituted by pairs of identical trajectories which are put synchronously both in input and in output of the network, plus the final state one on the adviser neuron. The negative examples are constituted by trajectories of insidious words for which the sole target is the final state zero on the adviser neuron. Actually we abandoned the usual idea of foreseeing the next step on the trajectories of the positive examples, since it could introduce further noise in the final classification; it is enough that the network reproduces the run time value of the input. For removing the suspicion that the network trivially computes the identity function we verified that the network still evolves according to its own dynamics when we connect the output with the input.

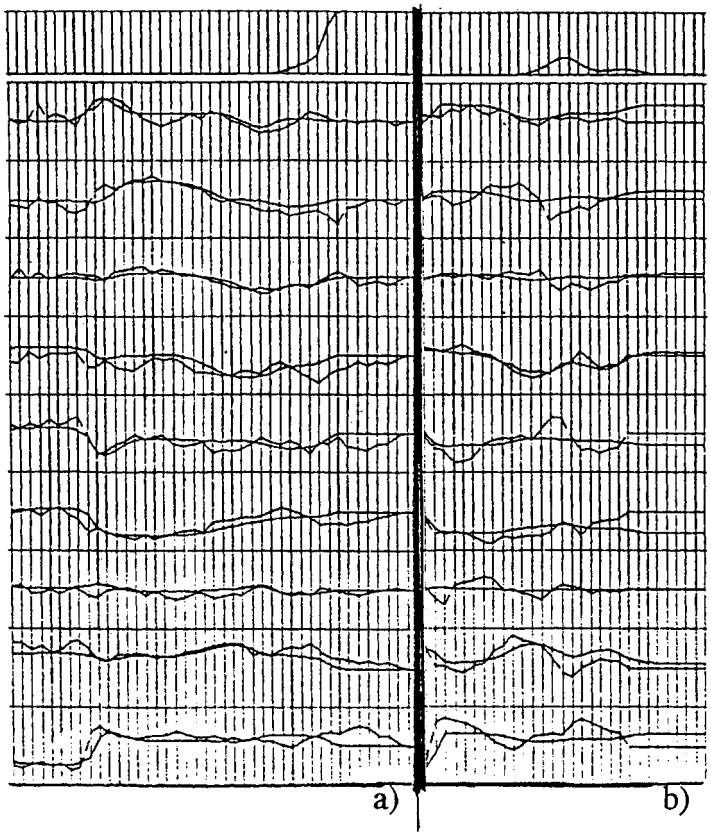

Fig. 3 Tracks of the adviser (on the top) and output neurons of a NTW trained on "sei" (six)
a. the net runs after the 9 Mel-cepstral tracks of an uttered "sei". The advbiser is up, the output neurons go close to the actual trajectories.
b. the net runs after the 9 Mel-cepstral tracks of an uttered "tre" three. The adviser reject apriori any comparison on the trajectories.
observed trajectories → smooth lines,
computed trajectories → rippled lines

At the end of the learning procedure we have one or more networks per word which are the subsymbolic templates of that word. On input a new utterance we expect that the sole networks which output trajectories close to the voiced signal in the parameters space are templates of the uttered word. This happens very often . Nevertheless in order to discriminate between the mentioned insidious words, s.a. "tre" and "sei" in Italian (see fig.3), we relay on the adviser **a** according to the following rule:
RULE: first of all consult **a**; if it judges compatible the pair network-utterance, then evaluate the mean square difference between the actual and computed trajectories. In case no adviser is compliant, classify on the basis of the sole quadratic distance.

3. NUMERICAL EXPERIMENTS AND CONCLUDING REMARKS

We used as database a home-made collection of utterances of the ten digits stored with a sampling rate of 8KHz in the research laboratory of ALCATEL-FACE, Pomezia, in quiet office rooms. The training set is made of 3 utterances times 20 speakers of the digits, segmented by hand. For the test set 37 speakers uttered 10 times the digits, and the voice signal is segmented automatically. The speakers were all males.
NTW is the last module of the architecture of fig. 4 and is simulated on a workstation.

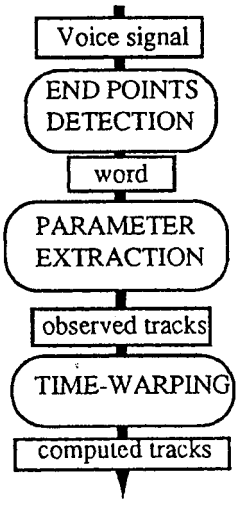

Fig. 4 Neural Time Warper architecture.

The previous modules are realized on a dedicated ALCATEL hardware called DIVA board.
The corresponding software procedure is made of three steps:

-End-points detection of the isolated words: through a simple thresholding algorithm.
-Parameter extraction: 9, 16-bit long, Mel-cepstral coefficients each temporal frame of 20 ms.
- NTW: The neural algorithm starts after parameter computation. Each signal flows in parallel through a set of 28 neural networks.The winner, according to RULE, gives its label (semantical meaning) to the word. The resulting confusion table, see table 1, shows a good global success rate in general, 95.87% on average, even on those set words, namely "tre","sei", "sette" and "quattro" and "otto",which in Italian are very difficult to automatically distinguish. Actually a great improvement, in the order of 20% was attained on those words by introducing the negative knowledge through the adviser.

Another benefit coming from the dynamic working of NTW is a sufficient insensitivity to segmentation errors, in the sense that it is the raising of the first ramp and the saturation of the last one between the hidden nodes which states the starting and ending point respectively of the word. So an internal clock is virtually introduced which is insensitive enough to word truncation or silence additions induced by the segmentation.

Some further improvement on NTW might come from:

a. Choice of a meaningful time scale for the comparison of the trajectories. Our guess is that the unity of this scale has to be related to the rising points of the ramps of the hidden nodes, where shorter or longer intervals between these points should come mainly from the inflection of the speaker.
b. Choice of a more appropriate auditory model. It is a general wisdom that people hearing a word uses only a part of the sound for understanding the whole word. This allows for large distortions which do not affect the understandability of a word, but, at the same time, requires removal of the distorted parts before automatically processing the sound trajectory. An interesting literature is available on this topic, dealing with systems s.a. MUSICAM [9,10].
c. Addition of noise on the tracks of the training set to make more pliant the learned model. This is a usual trick in the training practice, but has to be properly tuned.
d. Preprocessing of the tracks and/or use of other parameters. This is an extremely general point which is open to any solution.

References

1. T.W. Parsons, *Voice and speech processing*, McGraw-Hill (1986)
2. H. Sakoe & C. Chiba, *Dynamics programming algorithm optimization for spoken word recognition*, IEEE ASSP, 26,43-49 (1978).
3. L.R. Rabiner & B.H. Juang, *An introduction to Hidden Markov Models*. IEEE ASSP Magazine 3,1,4-16 [89].
4. R.J. Williams & D. Zipser, *A learning algorithm for continually running fully recurrent neural networks*, Neural Computation, 1, 270-280 (1989)
5. Apolloni B., D.Crivelli,F. Pazienti, A. Riccio *Automatic speech recognition by symbolic/subsymbolic approach: considerations about some experimental results..* Neural Networks World,1/93, 3-24, (1992)
6. Apolloni B., D.Crivelli,M. Amato, *Neural Time Warping - preliminary results.* To appear on Proc. of 2nd Italian wks on Neural Networks and Speech Processing, Firenze, December 1992, (1992).
7. B.A. Pearlmutter, *Learning state space trajectories in recurrent neural networks*, Neural Computation 1,263-269 (1989)
8. P. Werbos,*Backpropagation through time: what it does and how to do it.* IEEE Proc,(1990)
9. G. Stoll, *Source Coding for DAB and the Evaluation of its Performance: A major Application of the new ISO Audio Coding Standard.* To be present at EBU First International Symposium on Digital Audio Broadcasting, Montreux, June 1992, (1992)
10. G. Theile, G. Stoll and M. Link, *Low bit-rate coding of high-quality audio signals. An introduction to the MASCAM system.* EBU Review, Technical No. 230, August 1988, (1988)

Aknowledgments: This work has been developed under grant CNR 89.00010.69

	0	1	2	3	4	5	6	7	8	9
0	98.38	0.81	0	0.54	0	0.27	0	0	0	0
1	0.27	98.65	0	0	0.54	0	0	0	0.27	0.27
2	0.54	0.81	96.49	0	0	0.54	0	0	0	1.62
3	1.08	0	1.35	90.54	0	0	2.7	2.97	0	1.35
4	0.27	0.54	0	0	94.86	0	0	0	4.32	0
5	0	0	0	0	0	100	0	0	0	0
6	0.27	0	0	2.7	0	0	93.78	3.24	0	0
7	0.54	0	0.54	1.89	0	0	1.35	95.14	0.54	0
8	0.27	1.08	0	0	4.32	0	0	0	94.32	0
9	0.54	0	0.54	1.08	0	1.62	0	0	0	96.22

Table 1 Performance of a bench of NTW's as speaker independent recognizer of the ten digits.