CSELT Hybrid HMM/Neural Networks Technology for Continuos Speech Recognition

Roberto GEMELLO, Dario ALBESANO, Franco MANA

CSELT - Centro Studi e Laboratori Telecomunicazioni
via G. Reiss Romoli, 274 - 10148 Torino - Italy
Tel.: +39-011-2286224 Fax: +39-11-2286207 http://www.cselt.it
mailto: roberto.gemello@cselt.it, dario.albesano@cselt.it, franco.mana@cselt.it

Abstract

Neural networks have found their place among speech recognition technologies mainly with hybrid models that integrates the time warping ability of Hidden Markov Models (HMM) with the pattern recognition capability of neural networks (NN). Hybrid HMM-NN models have been investigated by several research teams, and constitute now a mature technology highly competitive with HMMs.

The authors' contribution to this research field is the introduction of a hybrid HMM-NN model whose original points are a training procedure which employs an integrated gradual movement of bootstrap speech segmentations to shorten training time, a time/feature architecture of the first hidden layer devoted to exploit a better feature selection using some a priori knowledge of speech, the use of a particular kind of acoustical modeling more suitable for a discriminative training, and a method to speed-up the execution of the neural component that makes possible develop real time applications using low cost hardware. Recently, the synergy of several input speech parameterizations has been experimented, leading to a further improvement in the recognition accuracy.

1. Introduction

Hybrid HMM-NN models integrate the ability of dealing with temporal patterns, typical of HMM, with the pattern classification power of NN. They inherit from HMM the modeling of words with left-to-right automata and the Viterbi decoding, delegating to a NN the computation of emission probabilities.

CSELT has been working on NN for speech recognition since 1988, and has developed his hybrid HMM-NN models in the first nineties, evolving them year by year. Starting from a first model dealing only with isolated words [1] [2], the extension to open vocabulary [4] by employing a proprietary set of subwords [5] has been developed. Then the study of a patented acceleration technique for the run-time module [3] has highly reduced the computational effort allowing large networks to run in real time on standard PCs, and opening the way to practical application of the technology. Recently the study and experimentation of Multi-Source approach, that foresees the parallel use of several speech parameterizations, has led to an important increase in performances [6][7][8]. Presently CSELT hybrid HMM-NN recognition technology is employed in some important applications for Telecom Italia and in a successful application for the Italian Railways.

This paper will be devoted to summarize CSELT hybrid HMM-NN recognition technology, starting from the common background described for example in [9], and pointing out the original points of our approach. The main differences between CSELT model (also called in the following NNA – *Neural Network Automata*) and other hybrid HMM-NN models are:

- a time/feature locally connected architecture on the first hidden layer (described in section 2);
- the use of Stationary-Transitional units (described in section 3);
- the different training procedure (described in section 4);
- an efficient way to extended the model to deal with Multi-Source input (described in section 5);
- the use (at run time) of the acceleration method (described in section 6).

0-7695-0619-4/00 $10.00 © 2000 IEEE

2. Neural Network Automata

The recognition model we use is a hybrid HMM-NN model devoted to recognize sequential patterns, named *Neural Network Automata (NNA)*. Each class is described in terms of a left-to-right automaton (with self loops) as in HMM. The emission probabilities of the automata states are estimated by a Multi-layer Perceptron (MLP) neural network, instead than by mixtures of gaussians, while the transition probabilities are not considered. The MLP may be recurrent or feedforward: this architectural choice has to be decided experimentally case by case depending on the kind of acoustic units that are modeled. In the case of whole word models recurrent networks have proved to be superior while in the case of phonemes or STU acoustical models feedforward MLP seems preferable.

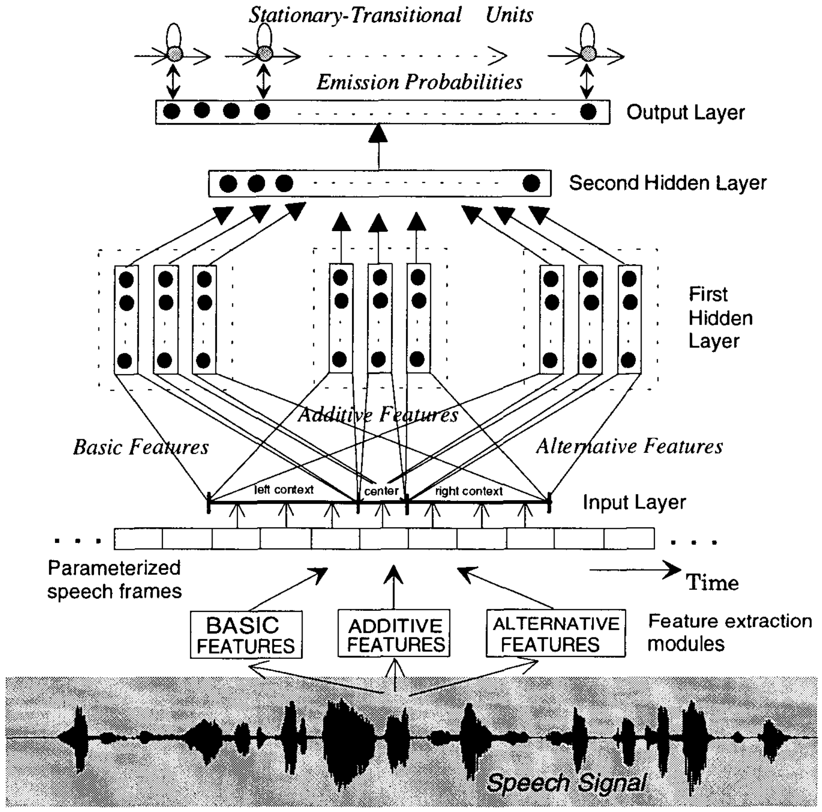

Figure 1. **General Architecture of Neural Network Automata for speech recognition**

The NNA has an input window that comprises some contiguous frames of the sequence, one or more hidden layers and an output layer where the activation of each unit estimates the probability P(Q|X) of the corresponding automaton state Q given the input window X.

An NNA has many degrees of freedom: the architecture of the MLP, the input window width, the number of automaton states for the different words or phonemes employed. A general NNA structure for the Basic and Multi-Source streaming, where additional and alternative features can be added to the basic one, is depicted in figure 1. Focusing on the basic architecture, the input window is 3-7 frames wide, and each frame contains 39 parameters (log Energy, 12 MFCCs, and their first and second derivatives). The first hidden basic sub-layer is divided into three feature detector blocks, one for the central frame, and two for the left and right context. Each block is in its turn divided into six sub-blocks devoted to keep into account the six types of different input parameters. It was empirically found that this a priori structure is generally better than a fully connected layer. The second hidden layer is fully connected with the output layer that estimates the emission probabilities of the states of the word or phoneme automata, and is virtually divided in several parts, each one corresponding to an automaton.

3. Stationary-Transitional Acoustical Modeling

Neural networks are discriminative classifiers and are well suited for output classes which are a partition of the output space, that is, that are not overlapping and cover the whole output space. For this reason HMM-NN hybrids usually model context independent phonemes (which are a partition of the sounds) and not context dependent biphones and triphones (which are overlapping). Thus the first kind of acoustical modeling was realized with the 27 Italian phonemes.

In the meantime, Fissore *et alii* [4] introduced a new kind of units, called *Stationary-Transitional Units (STU)*, which have very interesting features, and are very suitable to be modeled with neural networks. These units are made up by stationary parts of the context independent phonemes plus all the admissible transitions between them (in the language: in this case in Italian) for a total of 375 units. With these units a sequence of three phonemes *xpy* is modeled by the sequence of five units ...<*x*><*x-p*><*p*><*p-y*><*y*>... where <*x*>,<*p*>,<*y*> are the stationary parts of phonemes *x, p, y,* and <*x-p*>, <*p-y*> are the corresponding transitions between *x* and *p,* and *p* and *y.* The background noise "@" is treated like a standard phoneme, so it is present its stationary part @ and its transitions to and from all the phonemes (e.g. @-a,..,@-z,a-@,..,z-@).

This set of STU is language dependent but domain independent, and represents a partition of the sounds of a tongue, like phonemes, but with more acoustic detail, including both the stationary parts and the transitions between them. The modeling of STU with NNA can be realized by assigning one output unit of the MLP to each acoustic unit, both stationary and transitional. An alternative is to assign two states for transitions.

4. Neural Network Automata Training

NNA model differs from a pure HMM model mainly for the training. In fact a HMM estimates the probability of the input X given the model Q, P(X|Q), with a maximum likelihood method which takes into account only the correct class. On the contrary, a neural model like NNA estimates the probability P(Q|X) of the model Q given the input X with a discriminative approach, which takes into account all the classes contemporary, trying to separate in the best way the correct class from the others.

NNA training differs from the usual *"train from scratch MLP - re-segment training set - retrain from scratch MLP"* iteration generally adopted in HMM-NN hybrids because it integrates an incremental re-segmentation during a unique MLP training. That greatly reduces training time, so allowing the use of standard workstations for training. During NNA training we want simultaneously:

1) to find the best segmentation of utterances into the employed acoustic units and of acoustic units into states;
2) to train the network to discriminate that states.

Training is an iterative procedure as follows:

Initialization:

- initialize the NNA with small random weights;
- create the first segmentation by starting from a bootstrap segmentation of training utterances into the employed acoustic units, and segmenting them uniformly into the foreseen number of states;

Iterations:

For each epoch do:

- load the present segmentation;
- train the NNA for one epoch according to that segmentation;
- obtain a new segmentation by applying the dynamic programming to each utterance in the training set to re-evaluate the transition points proposed by the NNA;
- update the present segmentation by using a function of itself and of the new segmentation:

present_segm = F(present_segm, new_segm);

e.g. $F(s1, s2) = \alpha s1 + (1-\alpha)s2$, with α starting from 1.0 and decreasing during the training.

The targets are generated according to the present segmentation, putting 1.0 for the active state of the right automaton and 0.0 otherwise. All the automata are trained into a unique net, so performing a discriminative training. The MLP basic learning algorithm is the back-propagation. The error function is Cross-entropy and the output units' activation function is Softmax. The termination criterion is given by error stabilization. Intermediate weights are saved and tested on a validation set.

5. Extension of Neural Network Automata to Multi-Source Input

NNA is based on a suitable architecture to faced the problem of integrating in input two (or more) different sets of features (see Figure 1). In the case of Multi-Source input, the speech signal is analyzed by several signal-processing algorithms to extract several kinds of features: the *basic features* (the standard MFCC coefficients), the *additive features* (that is those features that are not independent but should be used in addition to the basic ones) and the *alternative features* (that is those features that are sufficiently informative and could be used instead of the basic features). Till now, we investigated successfully gravity centers and frequency derivatives among the additive features, RASTA-PLP and ear-model based among the alternative features. All these features are assembled into each input frame. Then the input frames are loaded into the network.

The input layer is made up, as before, of 7 frames (one central frame plus a 3 frame left context and a 3 frame right context), each one composed by a block of *basic features*, one or more blocks of *additive features* (if required) and one block of *alternative features* (if required). The *basic features* block contains: the Energy, 12 MFCC, their first and second degree time derivatives for a total of 13+13+13=39 values. The structure is the same for the *alternative features* block, while the *additive features* block contains the vector of computed *additive features* along with their first and second-degree time derivatives.

The problem of integrating Multi-Source information into the computation of the emission probability modeled by the NNA, is delegated to the neural network capability to deal with rough input data. The first hidden layer is divided along two dimensions, the *time dimension*, which considers a central frame, a left and a right context, and the *source dimension*, to take into account the different input sources separately. Thus, we have a sub-layer for the basic features, one for the additional features (if present) and one for the alternative features (if present). Each sub-layer is split, as before, into three temporal parts, one for the left context, one for the center frame and one for the right context. Each temporal part for basic features (and alternative features) is made up of six sub-parts devoted to the Total Energy, the Cepstrals (or Bands Energy), and their first and second derivatives, with a structure 5+30+5+30+5+30 = 105 units. These parts are connected only with the features they are devoted to. The temporal part for additive features is made up of one sub-part for each additive feature and, if present, their first and second derivatives. The number of units of each sub-part ranges from 3 to 30 depending on the feature cardinality. The second hidden layer is composed by 300 units, fully connected with the previous layer. It performs an integration of the acoustic characteristics extracted locally by the first hidden layer. The output layer contains 379 units, one for each stationary and transitional unit. While the hidden units are sigmoidal units, the output are softmax units, in order to compute a probability distribution over the subword units.

6. Speeding Up Neural Network Automata Execution

Moving from laboratory prototypes towards real voice services, the time execution of NNA turns out to be an important problem that must be faced. The method we propose assumes that no dedicated hardware is available and it is based on a more efficient use of the computations, needed to produce unit activations, performed in previous executions. Looking to the NNA forward computation, let us consider $net_i(t)$ the input at time t for a generic unit i and consider $\Delta o_j(t+1)$ the variation at time t+1 of the activation of unit j ($\Delta o_j(t+1) = o_j(t+1)-o_j(t)$) connected by an incoming weight to the i-th unit.

It is possible to write $net_i(t+1)$ as follows:

$$net_i(t+1) = net_i(t) + \Sigma_j w_{ij}\Delta o_j(t+1).$$

This formula points out how it is possible to realize the forward computation of NNA by means of the propagation through the net of the difference of the unit activation $\Delta o_j(t+1)$ instead of the activation value

$o_j(t+1)$. The forward computation of NNA follows the algorithm:

```
ForEach input unit $u_j$
    Compute the input difference $\Delta o_j(t+1) = o_j(t+1)-o_j(t)$
    If $(\Delta o_j(t+1) \neq 0)$
        ForEach weight $w_{ij}$ outcoming from unit $u_j$ to unit $u_i$
            Update $net_i$ value: $net_i(t+1) = net_i(t) + w_{ij}\Delta o_j(t+1)$
        End
End
ForEach hidden unit $u_j$
    Compute the unit activation $o_j(t+1) = SIGM(net_j(t+1))$
    Compute the difference $\Delta o_j(t+1) = o_j(t+1)-o_j(t)$
    If $(\Delta o_j(t+1) \neq 0)$
        ForEach weight $w_{ij}$ outcoming from unit $u_j$ to unit $u_i$
            Update $net_i$ value: $net_i(t+1) = net_i(t) + w_{ij}\Delta o_j(t+1)$
        End
End
ForEach output unit $u_j$
    Compute the unit activation $o_j(t+1) = SIGM(net_j(t+1))$
End
```

The gain in time achieved by the forward propagation of difference of the unit activation can be considerable and it is proportional to the number of outcoming weights of the unit where no difference takes place. In other words, the execution time is comparable with that of a smaller network without the weights coming out from the unit where the activation difference is zero.

In order to achieve the maximum speed up, we must perform a linear quantization of the codomain of the activation function for every hidden unit. In this way the unit activation assume a finite number of values and the condition $\Delta o_j(t+1)=0$ may frequently be satisfied. The accuracy of the network computation and the execution time depend on the number of steps involved in the quantization: accuracy of net outputs increases with the number of quantization steps, whereas maximum of speed up is achieved using the minimum number of steps. For this reason, a tuning of the quantization step is necessary in order to find the best compromise between fast execution and performance requirements.

From the complexity point of view, the proposed method is similar to the standard forward computation based on the propagation of the activation values. In the case of propagation of differences, we have the additional computation of the activation difference itself. However, this kind of computation depends on the number of units and can be neglected in the context of big networks. Finally, it is worthwhile noting that the proposed technique does not require additional memory.

Analizing the results we achieved in different experiments, the computational time is reduced of about 1/3 of the original one with no significant changes in recognition quality.

7. Applications and Performances

In the last years, industrial projects have exploited CSELT recognition technology in order to build intelligent call centers. Mainly, projects have dealt with directory assistance and voice access information (such as railway and fly timetables). Nevertheless, it is not easy to report performances from field applications. Thus, to give an idea of NNA performances, we report some tests performed on pre-recorded test sets. The experiments involve both isolated word and continuous speech tasks, both in Italian and in other three languages (English, Spanish and German).

The data bases we used to train the recognizers were collected inside European Project SPEECHDAT, for the foreign languages, or inside CSELT project. All the data bases were collected on the telephone network (300Hz-3400Hz sampled at 8KHz).and are made up by male and female speakers evenly balanced.

Language	Acoustical Modeling	Input Source	Vocabulary Size (words)	Number of Utterances	Isolated Word WA
Italian	391 STU	MFCC + RASTAPLP	475	14473	97.40%
English	400 STU	MFCC + RASTAPLP	31	1796	98.83%
German	41 Phoneme.	MFCC + RASTAPLP	59	6500	98.22%
Spanish	463 STU	MFCC	70	1332	99.02%

Table I – **Summary of the best performances of NNA as Word Accuracy in a Isolated Recognition task**

Table I reports the results we obtained on a isolated word recognition in term of word accuracy. The comparison of the results of the NNA technology changing the language is not fear because the cardinality of the vocabulary varies depending on the language.

Table II reports the results we obtained on a continuous speech recognition task in term of word accuracy. The results were obtained without the use of a language model and for this reason the results are not so high.

Language	Acoustical Modeling	Input Source	Vocabulary Size (words)	Number of Utterances	Continuous Word WA
Italian	391 STU	MFCC + RASTAPLP	9400	4296	65.50%
English	400 STU	MFCC + RASTAPLP	847	6237	44.60%

Table II – **Summary of the best performances of NNA as Word Accuracy in a Continuous Recognition task**

8. Conclusions

In this paper we have described NNA, the hybrid HMM-NN recognition system developed at CSELT. The main focus of the paper is to give a complete and detailed overview of the whole system. The topic discussed are about the acoustical modeling, architectural choices, particular algorithm used during training and testing. A very brief evaluation of the performances of the system has been given through the results obtained in different recognition tasks and in different languages.

References

[1] D. Albesano, R. Gemello and F. Mana, "Word Recognition with Recurrent Network Automata", in Proc. IJCNN 92, Baltimore, June 1992, pp. 308-313.

[2] R. Gemello, D. Albesano, F. Mana, R. Cancelliere "Recurrent Network Automata for Speech Recognition: A Summary of Recent Work", in Proc. of IEEE Neural Networks for Signal Processing Workshop, Ermioni, Greece, September 1994.

[3] D. Albesano, F. Mana, R. Gemello, "Speeding Up Neural Networks Execution: An Application to Speech Recognition", in Proc. of IEEE Workshop on Neural Networks for Signal Processing VI (NNSP-96), Kyoto, Japan 1996.

[4] R. Gemello, D. Albesano, F. Mana, "Continuous Speech Recognition with Neural Networks and Stationary-Transitional Acoustic Units", in Proc. of IEEE Conference on Neural Networks (ICNN-97), Houston, USA 1997.

[5] L. Fissore, F. Ravera, P. Laface, "Acoustic-Phonetic Modeling for Flexible Vocabulary Speech Recognition", in Proc. of EUROSPEECH '95, Madrid, September 1995.

[6] R. Gemello, D. Albesano, F. Mana, "Multi-source neural networks for speech recognition", in Proc. of International Joint Conference on Neural Networks (IJCNN'99), Washington, July 1999.

[7] D. Albesano, R. De Mori, R. Gemello, F. Mana, "A Study on the Effect of Adding New Dimensions to Trajectories in the Acoustic Space", in *Proc. of Eurospeech'99,* Budapest, Hungary, 1999, pp.1503-1506.

[8] R. Gemello, D. Albesano, F. Mana, "Synergy of Spectral and Ear Model Features for Neural Speech Recognition", in *Proc. of International Conference on Artificial Neural Networks - ICANN '99,* Edimburgh, Scotland, September 1999.

[9] H. Bourlard, N. Morgan, *Connectionist Speech Recognition: A Hybrid Approach,* Kluwer Academic Publishers, 1993.