ACOUSTIC MODEL BUILDING BASED ON NON-UNIFORM SEGMENTS AND BIDIRECTIONAL RECURRENT NEURAL NETWORKS

Mike Schuster

ATR, Interpreting Telecommunications Research Lab.
2-2 Hikari-dai, Seika-cho, Soraku-gun, Kyoto 619-02, JAPAN
gustl@itl.atr.co.jp http://www.itl.atr.co.jp/

ABSTRACT

In this paper a new framework for acoustic model building is presented. It is based on non-uniform segment models, which are learned and scored with a time bidirectional recurrent neural network. While usually neural networks in speech recognition systems are used to estimate posterior "frame to phoneme" probabilities, they are used here to estimate directly "segment to phoneme" probabilities, which results in an improved duration model. The special MAP approach allows not only incorporation of long term dependencies on the acoustic side, but also on the phone (output) side, which results automatically in parameter efficient context dependent models. While the use of neural networks as frame or phoneme classifiers always results in discriminative training for the acoustic information, the MAP approach presented here also incorporates discriminative training for the internally learned phoneme language model. Classification tests for the TIMIT phoneme database gave promising results of 77.75 (82.38)% for the full test data set with all 61 (39) symbols.

1. INTRODUCTION

Speech recognition has been partially successful since it has been treated as a general dynamic pattern recognition problem which can approximately be solved with statistical methods. The general recognition problem can for example be described as follows. Given acoustic data as an utterance U, the theoretically lowest (utterance) error rate is achieved by evaluating the posterior probability $\Pr(S|U,M)$ for all possible class (phoneme, word) sequences $S = c_1, c_2, ..., c_N$ given our model $M(\vec{\delta})$ with parameters $\vec{\delta}$, and picking the output S with the highest posterior probability. The model parameters $\vec{\delta}$ were estimated during a training process to maximize some objective function for the training data (for example maximum likelihood).

To handle the acoustic data in practical models, it has to be in the form of a temporal vector sequence, where the vectors have a fixed dimensionality. There are at least two ways of transforming the acoustic data U into a vector sequence $X = \vec{x_1}, \vec{x_2}, ..., \vec{x_N}$. One way is to divide the acoustic data in short time intervals and transform the intervals into fixed dimensional vectors. Another way is to assume some segmentation of the acoustic data and transform the segments into fixed dimensional vectors. This approach has for example been used in [9] and is also used here.

2. OVERVIEW

This paper consists of two major parts. The first part shows how to generally estimate the posterior probability of complete utterances as an alternative approach to the regular split-up into acoustic model and language model likelihood. The second part shows how a bidirectional recurrent neural network can efficiently be used to model the occuring probability terms in practice.

3. DIRECT MAP ESTIMATION OF COMPLETE UTTERANCES

Instead of splitting up into acoustic and language model likelihood ($\Pr(S|X) \sim p(X|S) \cdot \Pr(S)$) there is the possibility of estimating the posterior probability $\Pr(S|X)$ directly as follows. With $p(x,y) = p(x)p(y|x)$ it can be decomposed into:

$$\begin{aligned}\Pr(S|X) &= \Pr(c_1, c_2, ..., c_N | \vec{x_1}, \vec{x_2}, ..., \vec{x_N}) \\ &= \underbrace{\prod_{i=1}^{N} \Pr(c_i | c_{i+1}, c_{i+2}, ..., c_N, \vec{x_1}, \vec{x_2}, ..., \vec{x_N})}_{\text{MAP-backward probability}} \\ &= \underbrace{\prod_{i=1}^{N} \Pr(c_i | c_1, c_2, ..., c_{i-1}, \vec{x_1}, \vec{x_2}, ..., \vec{x_N})}_{\text{MAP-forward probability}}\end{aligned}$$

One probability term within the product is the conditional probability of a output class given all the input to the right and left hand side plus the outputs on one side. Note that these decompositions are only a simple application of probability rules - to apply them no assumptions concerning the shape of the distributions were made.

The two ways of decomposition of $\Pr(S|X)$ (many more complicated decompositions are possible) are here referred to as *MAP-forward* and *MAP-backward* probability. The goal is to train some classifier to estimate conditional probabilities of the kind $\Pr(c_i | c_{i+1}, c_{i+2}, ..., c_N, \vec{x_1}, \vec{x_2}, ..., \vec{x_N})$.

3.1. Model Merging

Let's assume there are trained models which give estimates for $\Pr(S|X)$ for both decompositions. Since neither of the estimates will be perfect, a better estimate can be achieved by merging the two estimates somehow. One way of *merging opinions of different experts* is to assume the opinions

to be independent which leads for probabilities to a geometric averaging or alternatively to an arithmetic averaging in the log-domain. This process is referred to as *logarithmic opinion pooling* [3]. It should be noted that it is in general very difficult to create experts with independent opinions which justify such a merging approach. For example, in real applications the different experts are usually trained on the same data set which represent possible samples from the underlying distribution. This makes the experts already dependent. Although, in practice the dependency is often negligible for a small number of experts.

4. BIDIRECTIONAL RECURRENT NEURAL NETWORKS

Neural networks are excellent tools for general conditional probability estimation. It has been shown a number of times (for example [1, 2]) that neural networks can estimate for input distribution X and output distribution Y at least the conditional average $E\{Y|X\}$ for regression problems and the conditional probability of class membership $\Pr\{Y|X\}$ for classification problems.

In speech recognition, for the estimation of posterior "frame to phoneme" probabilities especially recurrent neural networks (RNNs) have been very successful [4], because they allow efficient parameter sharing and make efficient use of context. Training and usage is not more complicated than for regular NNs with for example MLP or TDNN structure, since the RNN can be unfolded in time into a feed forward network. A typical structure can be seen in Fig.1.

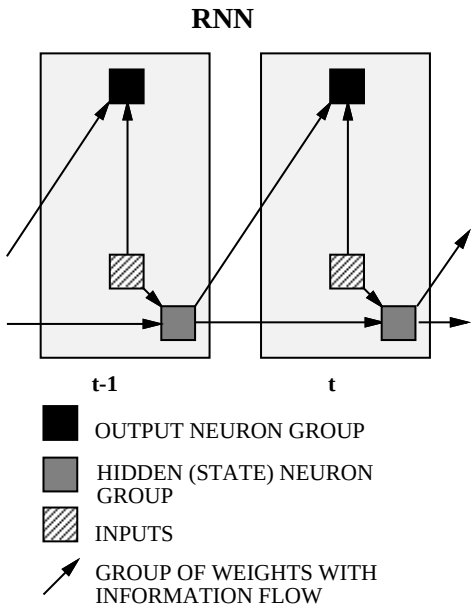

Figure 1. Regular (uni-directional) recurrent neural network structure (RNN)

Recurrent neural networks of that type have the principal disadvantage that they can only make use of context information in one time direction (of the past with respect

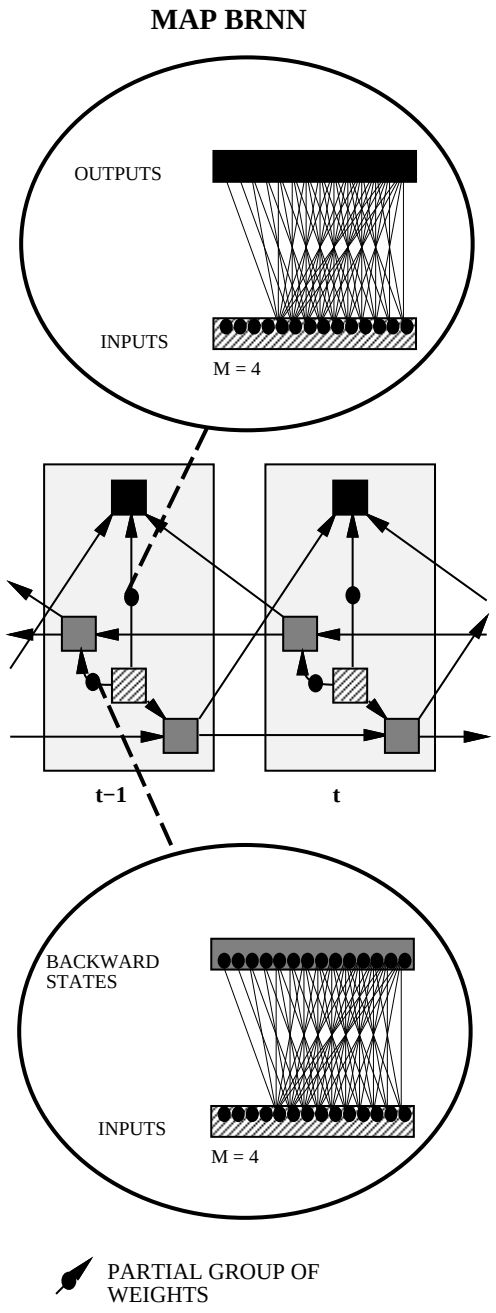

Figure 2. Bidirectional recurrent neural network structure (BRNN), with the state neurons split up into forward and backward state neurons, here with extensions for MAP forward probability estimation with the target class input coded in the first 4 channels

to the time axis). This restriction can partially be loosened by delaying the output by a predefined number of frames to partially include future context. Although this method is successfully used in practice [4], there is still the limitation of only using context up to a preset future frame. To make use of all available context it is possible to train two different networks, one in each time direction, and merge the results [5]. If the two networks are assumed to be independent experts, then a *logarithmic opinion pooling* with the possible disadvantages like discussed above can be applied.

A more elegant approach to using all available context is a bidirectional recurrent neural network (BRNN) [8], which can be trained in both time directions simultaneously and hence avoids the difficult procedure of mixing dependent experts. The general structure can be seen in Fig.2. For model problems the BRNN structure has lead to better results than the mixture of two regular uni-directional RNNs trained separately for each time direction [8].

A slightly modified BRNN structure can efficiently be used to estimate conditional probabilities of the kind $\Pr(c_i|c_1, c_2, ..., c_{i-1}, \vec{x_1}, \vec{x_2}, ..., \vec{x_N})$. A visualization of the estimation problem can be seen in Fig.3. There are continuous $(\vec{x_1}, \vec{x_2}, ..., \vec{x_N})$ and discrete inputs $(c_1, c_2, ..., c_{i-1})$.

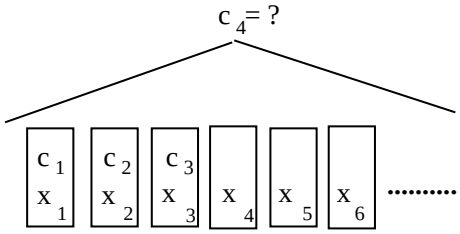

Figure 3. Visualization of the estimation problem for the MAP-forward probability

Let's assume that the input for a specific time i is coded as one long vector containing the target class c_i and the continuous inputs $\vec{x_i}$, with for example the discrete input c_i coded in the first K dimensions of the whole input vector. If the first K weight connections from the inputs to the backward states and the inputs to the outputs are cut, then only discrete input information from $t < i$ is used to make predictions. This is exactly what is required to estimate the terms within the MAP-forward probability expression. Fig.2 illustrates this change of the original BRNN architecture. Cutting the input connections to the forward states instead of the backward states gives the architecture for estimating the MAP-backward probability.

5. EXPERIMENTS & RESULTS

To test the direct MAP approach with bidirectional RNNs as probability estimators the TIMIT phoneme database was chosen. Because a complete decoder is not yet available, classification tests were performed.

5.1. Feature Extraction

Feature extraction is used on three levels. First, frame features are extracted to represent the raw waveform in a compressed form. Then, with knowledge of the boundary locations, segment features are extracted to map the information from an arbitrary length segment to a fixed dimensional vector. A third transformation was applied to the segment feature vectors to make them suitable as inputs to a neural net.

5.1.1. Frame Feature Extraction

As frame features 12 regular MFCCs (24th order) plus the log-energy were extracted at every 10ms with a 25.6ms Hamming window and a preemphasis of 0.97.

5.1.2. Segment Feature Extraction

From the frame features the segment features were extracted as the simple integrals of five equally spaced regions within the segment for each of the 13 frame features. The function values between the data points were linearly interpolated. The duration of the segment was an additional segment feature which resulted in a 66-dimensional segment feature vector.

5.1.3. Neural Network Preprocessing

Although feed-forward neural networks can principally handle any form of input distributions it was found in the experiments here that best results are achieved with Gaussian input distributions which matches the experiences from [4]. To generate an "almost-Gaussian distribution" the inputs were first normalized to mean zero and variance one on a sentence basis, and then every channel was vector quantized with 256 codebook vectors (1 byte) so the entropy of the channel was a maximum for the whole training data. The maximum entropy codebooks can easily be found for the one-dimensional arrays because an array can be split up in regions which have (almost) the same number of samples each. For presentation to the network the byte-coded value was remapped with $value = \mathrm{erf}^{-1}(2 \cdot (\mathrm{byte} + 1/2)/256 - 1)$, ($\mathrm{erf}^{-1}$ is the inverse error function, erf() is part of math.h) which produces on average a distribution that is similar to a Gaussian distribution.

5.2. Experiments & Results

The experiments were performed on the full TIMIT data set. Training was done on the 462 training speakers (142910 phonemes), testing on the remaining 168 test speakers (51681 phonemes). To include the output class information the original 66-dimensional feature vectors were extended to 72 dimensions. In the first six dimensions the corresponding output class was coded in a binary format (binary [0,1] $\rightarrow$ network input [-1,1]). Two MAP-BRNN structures were trained as classifiers (softmax output function, cross-entropy objective function) with 500 steps of resilient propagation [7], extended to a RPROP through time variant, one for each time direction. All neurons but the output neurons had the *tanh* activation function. The forward MAP-BRNN had 64 forward and 32 backward states. Additionally 64 hidden neurons were implemented before the output layer. The backward MAP-BRNN was symmetrical to the forward MAP-BRNN (32 forward, 64 backward states), leading altogether to $2 \cdot 26333$ weights. The networks were trained to give estimations for the probability terms within the product of the MAP-forward and MAP-backward probability. The probabilities coming from the

networks were merged as a linear and logarithmic opinion pool. Tab.1 and Tab.2 show the phoneme classification results for the full training and test set. Although the database is labeled to 61 symbols, a lot of researchers have chosen to map them to a subset of 39 symbols. Here results are given for both versions, with the results for 39 symbols being simply a mapping from the results obtained for 61 symbols. Details of this standard mapping can be found in [6]. The slightly better results for the logarithmic opinion pool show that it is at least reasonable to assume the experts as independent, although they were trained on the same data set.

SET-UP	Rec-Rate TRAIN 61	Rec-Rate TEST 61
for MAP-BRNN	79.11 %	72.70 %
back MAP-BRNN	79.38 %	72.74 %
both merged, lin	83.57 %	77.53 %
both merged, log	83.89 %	77.75 %

Table 1. Classification results for full TIMIT training and test data with 61 symbols for different neural network SET-UPs

SET-UP	Rec-Rate TRAIN 39	Rec-Rate TEST 39
for MAP-BRNN	84.42 %	79.08 %
back MAP-BRNN	83.27 %	77.44 %
both merged, lin	87.17 %	82.11 %
both merged, log	87.45 %	82.38 %

Table 2. Classification results for full TIMIT training and test data with 39 symbols for the different neural network SET-UPs (mapped from the results obtained for 61 symbols)

6. DISCUSSION

The framework presented here shows how it is possible to estimate directly the posterior probability of continuous phoneme utterances without splitting into acoustic and language model likelihood. The resulting conditional probabilities can in practice efficiently be estimated with a bidirectional recurrent neural network without making any explicit assumptions about how much context is important, neither in the input nor in the output space. Because a NN is used for the conditional probability estimation, there are very little assumptions about the shape of the distributions. The segment based approach insures the proper handling of duration - it is just another feature for the classifier. The NN-MAP decoding approach results in discriminative training, also for the internal phoneme language model, and automatically in context dependent models. The complexity of the system is only controlled by the complexity of the neural net.

Several improvements are possible. The segment feature extraction is simple compared to the sophisticated techniques used in [9]. Compared to frame based approaches which try to preserve all input information until decoding, it is very likely that during this simple segment feature extraction important information is lost.

REFERENCES

[1] Richard and Lippman, "Neural Network Classifiers estimate Bayesian a posteriori probabilities", Neural Computation 3, P.461-483, 1991

[2] Chris Bishop, "Neural Networks for Pattern Recognition", Oxford University Press 1995

[3] James O. Berger, "Statistical decision theory and Bayesian analysis", Second Edition, Springer Verlag

[4] Anthony J.Robinson, "An application of recurrent neural nets to phone probability estimation", IEEE Transactions on Neural Networks, Vol.5, No.2, March 1994

[5] Tony Robinson, Mike Hochberg, Steve Renals, "Improved phone modeling with recurrent neural networks", ICASSP 1994, Page I-37

[6] Tony Robinson, "Several improvements to a recurrent error propagation network phone recognition system", Cambridge University Engineering Department Technical Report CUED/F-INFENG/TR82, September 1991

[7] M.Riedmiller, H.Braun, "A direct adaptive method for faster back-propagation learning: The RPROP algorithm", ICNN 1993

[8] Mike Schuster, "Learning out of time series with an extended recurrent neural network", Neural Network Workshop for Signal Processing 96, Kyoto, Page 170-179

[9] Glass, Chang, McCandless, "A probabilistic framework for feature-based speech recognition", ICSLP96, to appear