INCORPORATION OF HMM OUTPUT CONSTRAINTS IN HYBRID NN/HMM SYSTEMS DURING TRAINING

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

This paper describes a method to incorporate the HMM output constraints in frame based hybrid NN/HMM systems during training. While usually the NN parameters are adjusted to maximize the cross-entropy between the frame target probabilities and the network predictions assuming statistically independent outputs in time, in the approach described here the full likelihood of the utterance(s) using also the HMM output constraints, like for conventional HMM systems, is maximized. This is achieved by maximizing the state occupation probabilities after a forward/backward pass using the scaled likelihoods coming from the network. Making a simplifying approximation for the derivative for the back-propagation through the forward/backward pass, tests show that the proposed method gives consistently higher string (phoneme) recognition rates than the conventional approach that aims at maximizing cross-entropy at the frame level.

1. INTRODUCTION

Hybrid NN/HMM systems [1, 2, 3] have become more and more recognized as serious speech recognition systems next to their conventional continuous density (CD) HMM counterparts. They have a number of advantages including discriminative training at the frame level, easy incorporation of wide input windows to relax the frame independence assumption used in conventional HMM systems, they give rise to fast search algorithms because the use of the posterior probabilities coming from the network allows early pruning, and the system architecture is in general a lot simpler and contains usually around a magnitude less parameters than comparable conventional CD-HMM systems.

During recognition, a frame-based hybrid NN/HMM system is usually used like shown in Fig.1. A rough recipe for its usage is: The

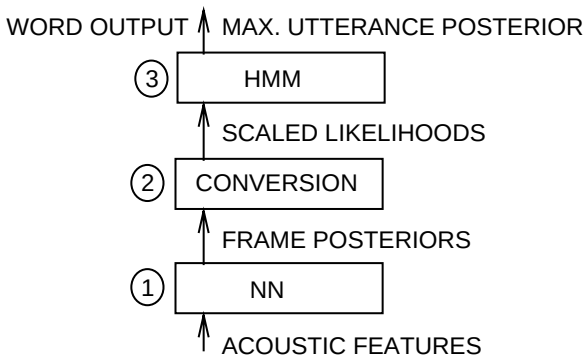

Figure 1. Usage of a hybrid NN/HMM system during recognition

acoustic features of an utterance are fed into the first module (a NN), that estimates frame posterior probabilities P(phone class|input frame) assuming neighboring data pairs (in time) to be independent. The posteriors are fed through a second module to give *scaled likelihoods* $C \cdot P$(input frame|phone class), which is usually achieved by dividing through the frame prior probability estimated from the training data. The formulae below give a justification for this procedure. The third module finally uses HMMs with a given structure and transition probabilities estimated from the training data and the observation likelihoods from the second module to run a search to give the output frame sequence with the highest *utterance* likelihood P(all input|phoneme sequence), which may be converted to a word output sequence. Note that

the last step is the same as in conventional HMM systems - it involves maximization of the utterance likelihood for the *phoneme* sequence, which is not necessarily the same as minimizing the word error rate being the quantity of interest. In practice although, these two quantities seem to be strongly correlated, which justifies this approach.

Writing down Fig.1 in mathematical terms: Assuming an observed acoustic feature vector sequence $X = \mathbf{x}_1^T = \{\mathbf{x}_1, \mathbf{x}2_2, \ldots, \mathbf{x}_T\}$ of length T and a possible frame class sequence $C = \mathbf{c}_1^T = \{c_1, c_2, \ldots, c_T\}$, then the likelihood plus transition and language model probabilities (for modeling with mono-phones, 1-state HMMs per phone and a bigram language model):

$$
\begin{aligned}
L &= P(\mathbf{x}_1^T|\mathbf{c}_1^T) \cdot P(\mathbf{c}_1^T) \\
&= \Big[\prod_{t=1}^{T} P(\mathbf{x}_t|\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{t-1}, \mathbf{c}_1^T)\Big] \\
&\quad \cdot \Big[\prod_{t=1}^{T} P(c_t|c_1, c_2, \ldots, c_{t-1})\Big] \\
&\approx \Big[\prod_{t=1}^{T} P(\mathbf{x}_t|c_t)\Big] \cdot \Big[\prod_{t=1}^{T} P(c_t|c_{t-1})\Big] \\
&= \Big[\prod_{t=1}^{T} \frac{P(c_t|\mathbf{x}_t) \cdot P(\mathbf{x}_t)}{P(c_t)}\Big] \cdot \Big[\prod_{t=1}^{T} P(c_t|c_{t-1})\Big] \\
&\propto \Big[\prod_{t=1}^{T} \frac{P(c_t|\mathbf{x}_t)}{P(c_t)}\Big] \cdot \Big[\prod_{t=1}^{T} P(c_t|c_{t-1})\Big]
\end{aligned}
$$

using $p(y|x) = p(x|y)p(y)/p(x)$ and $p(x,y) = p(y|x)p(x)$.

$P(c_t|\mathbf{x}_t)$ corresponds to module 1, the division through the class prior $P(c_t)$ to module 2, and $P(c_t|c_{t-1})$ to the transitions of module 3 in Fig.1, respectively.

Despite a lot of favorable features, state-of-the-art NN systems don't reach yet the word recognition performance conventional CD-HMM systems achieve. One reason for that might be, that the training procedure doesn't really pair with the recognition procedure like shown in Fig.2.

In current systems the parameters of the first module are adjusted to maximize some objective function (usually the cross-entropy between the frame target probabilities and the NN predictions) without considering the constraints implied by module 2 and 3, although during recognition the maximum posterior probability for the utterance is seeked. This is definitely a mismatch. The optimal solution would be to estimate all parameters in the three modules with respect to the utterance posterior like shown on the right hand side of Fig.2. In this paper a first attempt is made to adjust only the parameters of the NN after a (simplified) back-propagation of the errors through all three modules with respect to the state occupation probabilities after the third module.

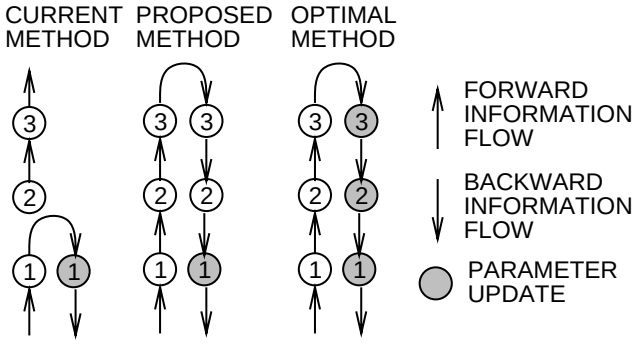

Figure 2. Current, proposed and optimal training procedure for the parameters in frame-based hybrid NN/HMM systems

2. APPROACH

Assuming adjustable parameters (weights) $\mathbf{w}$ only in the NN (although module 2 and 3 also contain parameters), the derivative of the full log likelihood for one observation sequence $O = \{\mathbf{o_1}, \mathbf{o_2}, \ldots, \mathbf{o_T}\}$ of length T after the forward/backward algorithm [4] for N states (phoneme classes with prior probability $\Pr(j)$) with respect to the n-th weight w_n is:

$$
\begin{aligned}
\frac{\partial}{\partial w_n} L_{\mathbf{w}}(O|M) &= \sum_{t=1}^{T}\sum_{j=1}^{N} \Big\{\sum_{i=1}^{N} \alpha_i(t-1)a_{i,j}\Big\}\beta_j(t) \\
&\quad \cdot \frac{1}{\Pr(j)} \cdot \frac{\partial}{\partial w_n} \Pr(j|\mathbf{o_t})
\end{aligned}
$$

$\Pr(j|\mathbf{o_t})$ stands for the prediction from the network for phoneme class j given the input $\mathbf{o_t}$. Because implementation of this correct derivative is a bit complicated the following simplifying approximation was made:

The outputs of the network trained in the conventional way to minimize cross-entropy can be viewed as a crude approximation for the state occupation probabilities. Let's assume the outputs of the neural network produce already the *correct* state occupation probabilities, which is in reality of course not the case, since output constraints are not taken into account at this stage. Then the derivatives through the last two modules simplify to the constant one. A possible training procedure is then:

1. Perform a (NN) forward pass through modules 1,2,3 to calculate the current state observation probabilities.
2. Replace the network outputs by the state occupation probabilities using the (crude) approximation that the back-propagated derivative through module 2 and 3 is one.
3. Perform a (NN) backward pass through module 1 (a regular back-propagation backward pass through a NN) and adjust the parameters in module 1.

To calculate the (NN) forward pass through module 3 in step 1 it is necessary to perform a full (HMM) forward/backward pass, which can be computationally expensive. Several fast approximations to a full (HMM) forward/backward pass are possible and were also tested here:

1. **Viterbi**:
   Instead of the sum over all states in the forward pass of the (HMM) forward/backward pass the maximum is taken and back-trace information to this state with the highest probability is stored. The backward pass of the (HMM) forward/backward pass involves then only the back-trace through the states. All output (state occupation) probabilities are therefore either zero or one.
2. **max.forward**:
   Instead of the sum in the forward pass of the (HMM) forward/backward pass the maximum is taken. No (HMM) backward pass information is used. To use this simplified forward probability only is obviously not very useful, but it can be used as one part of approximating the full forward/backward probability (see **max.lin merge** and **max.log merge** below).
3. **max.backward**:
   The same as **max.forward**, but in opposite time direction. A forward pass of the (HMM) forward/backward pass is performed starting from the last frame of the utterance. This involves a from the regular forward pass different transition matrix, since in general $P(c_t|c_{t-1}) \neq P(c_{t-1}|c_t)$.
4. **max.lin merge**:
   The probabilities of **max.forward** and **max.backward** are merged in the linear domain (linear opinion pooling, [6]).
   $$p_{new} = 1/2 \cdot (p_{forward} + p_{backward})$$
5. **max.log merge**:
   The probabilities of **max.forward** and **max.backward** are merged in the log domain:
   $$p_{new} = e^{1/2 \cdot (log(p_{forward}) + log(p_{backward}))}$$
   the correct way of merging probabilities if they are statistically independent (logarithmic or independent opinion pooling, [6]). After merging they are normalized so output probabilities over all possible states at any time point sum up to one.

3. EXPERIMENTS & RESULTS

Experiments were performed on a small part (first 100 training sentences) of the TIMIT phoneme database. As features 14 MFCCs plus log-energy and corresponding deltas were taken, making it 30-dimensional input vectors. The output vector dimension was 61 corresponding to the number of phonemes of the TIMIT database. Two NNs (3491 weights) were trained to compare the current training method (first row) and the proposed method (other rows) for Viterbi and full forward/backward decoding (Tab.1). The NNs were bidirectional recurrent NNs ([5]) with eight forward and eight backward states. As objective function the cross-entropy between the network outputs after the HMM layer (which are interpreted as state occupation probabilities) and the target probabilities was taken. Tab.1 shows the frame-to-frame and string recognition results for the training data. As can be seen, for the networks

used here the results for the training with incorporation of the HMM output constraints is better than neglecting them. The one exception is, when for the training procedure a Viterbi approximation (all output state occupation probabilities are forced to one or zero) is made.

TRAINING	FRAME (%) REC-RATE	STRING (%) REC-RATE
without HMM	62.5 (62.5)	58.7 (56.4)
full for/back	71.2 (74.5)	67.6 (68.2)
Viterbi	52.5 (53.0)	48.7 (27.3)
max. forward	59.1 (59.4)	60.7 (56.5)
max. backward	59.8 (62.1)	59.9 (57.1)
max. lin merge	69.6 (70.1)	67.0 (63.6)
max. log merge	71.1 (71.0)	67.7 (64.8)

Table 1. Comparison of different training methods using the Viterbi algorithm (or the full forward/backward algorithm in brackets) during decoding. In all cases 1-state mono-phone HMMs with transitions estimated from the training data were used.

All approximations to the full (HMM) forward/backward pass were in these experiments (since the networks are relatively small) considerably faster than the full (HMM) forward/backward pass. Tab.2 shows the normalized training time for the different approximations.

TRAINING	TIME
without HMM	1.0
full for/back	8.8
Viterbi	1.3
max. forward	1.3
max. backward	1.3
max. lin merge	1.7
max. log merge	1.7

Table 2. Normalized training time. The last four approximations to the full (HMM) forward/backward pass are not much slower than the simple Viterbi approximation.

4. DISCUSSION

This paper makes a first attempt to include all output constraints in a hybrid NN/HMM system during training, which leads for small amounts of data and small networks to considerably higher string recognition rates or in turn to models with less parameters for achieving the same string recognition rates. Future work will clarify whether this increase in training complexity is useful for large (real-world) amounts of data. This paper also evaluates in the same framework some unconventional fast approximations to a full (HMM) forward/backward pass, which ranked in performance between the worst approximation (Viterbi) and the original full forward/backward pass. These unconventional approximations were almost as fast as a simple Viterbi approximation.

Like all ML methods it is questionable whether an increase in likelihood on the training data has the same effect on unseen test data. This is a different problem and is not addressed here.

REFERENCES

[1] A.J. Robinson, "An application of recurrent neural nets to phone probability estimation", *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 298-305, 1994.

[2] T. Robinson, M. Hochberg, and S. Renals, "The use of recurrent neural networks in continuous speech recognition," in *Automatic Speech Recognition: Advanced Topics*, C.H. Lee, F.K. Soong, and K.K. Paliwal, Eds. Boston: Kluwer Academic Publishers, pp. 233-258, 1996.

[3] H. Boulard and N. Morgan, "Continuous Speech Recognition by Connectionist Statistical Methods", *IEEE Transactions on Neural Networks*, 4 (6):893-909, November 1993

[4] Lawrence Rabiner and Biing-Hwang Juang, "Fundamentals of Speech Recognition", Prentice Hall, 1993.

[5] M. Schuster, "Learning out of time series with an extended recurrent neural network", in *Proc. IEEE Neural Network Workshop for Signal Processing*, pp. 170-179, 1996.

[6] J.O. Berger, *Statistical Decision Theory and Bayesian Analysis*, Berlin: Springer-Verlag, 1985.