Improved Neural Network Initialization by Grouping Context-Dependent Targets for Acoustic Modeling

Gakuto Kurata, Brian Kingsbury
IBM Watson
gakuto@jp.ibm.com, bedk@us.ibm.com

Abstract

Neural Network (NN) Acoustic Models (AMs) are usually trained using context-dependent Hidden Markov Model (CD-HMM) states as independent targets. For example, the CD-HMM states of $A-b-2$ (second variant of beginning state of $A$) and $A-m-1$ (first variant of middle state of $A$) both correspond to the phone $A$, and $A-b-1$ and $A-b-2$ both correspond to the Context-independent HMM (CI-HMM) state $A-b$, but this relationship is not explicitly modeled. We propose a method that treats some neurons in the final hidden layer just below the output layer as dedicated neurons for phones or CI-HMM states by initializing connections between the dedicated neurons and the corresponding CD-HMM outputs with stronger weights than to other outputs. We obtained 6.5% and 3.6% relative error reductions with a DNN AM and a CNN AM, respectively, on a 50-hour English broadcast news task and a Japanese LVCSR task and confirmed 3.6% error reductions with a DNN AM and a CNN AM, respectively, on a 500-hour Japanese task, in all cases after Hessian-free sequence training. Our proposed method only changes the NN parameter initialization and requires no additional computation in NN training or speech recognition runtime.

Index Terms: speech recognition, deep neural network, convolutional neural network, context-dependent HMM state target, parameter initialization

1. Introduction

Neural Network based Acoustic Models (NN-AMs) have been showing significant improvement over the conventional Gaussian Mixture Model-AMs (GMM-AMs) in Large Vocabulary Continuous Speech Recognition (LVCSR) [1]. Various types of NN including Feed-forward Deep Neural Network (DNN) [2], Convolution Neural Network (CNN) [3], Recurrent Neural Network (RNN) [4], Long Short Term Memory (LSTM) [5] which is a specific type of RNN, and the novel combination of them [6] have been used for acoustic modeling. For decoding the speech, bottleneck feature-based system [7, 8], NN/Hidden Markov Model (HMM) hybrid system [9, 10], and recently proposed end-to-end speech recognition system [11, 12, 13, 14, 15] have been proposed. While the end-to-end speech recognition system using such as the Connectionist Temporal Classification (CTC) and the encoder-decoder approaches have been showing promising results recently, the NN/HMM hybrid system is still the most-widely used system. In the NN/HMM hybrid system, NN-AMs are used to estimate the posterior probabilities for acoustic observations and these probabilities are used for decoding by being combined with an HMM. In the 1990s, due to the computational costs and limited training data, phones were used as the output targets of NN-AMs [16]. Recent studies have shown that using the context-dependent HMM (CD-HMM) states as targets for NN training resulted in significant improvement in accuracy [9, 17, 18].

When training NN-AMs using CD-HMM states as output targets, each output target is treated independently. For example, while the CD-HMM states of $A-b-2$ (second variant of beginning state of $A$) and $A-m-1$ (first variant of middle state of $A$) both correspond to the phone $A$, and $A-b-1$ and $A-b-2$ both correspond to the Context-independent HMM (CI-HMM) state $A-b$, these relationships are not explicitly considered.

In this paper, we propose a NN parameter initialization method to leverage the relationship between the target CD-HMM states. With our proposed method, we first define the group of CD-HMM states based on their corresponding CI-HMM states or phones. Then, for each group, we prepare dedicated neurons in the final hidden layer just below the output layer. Each dedicated neuron for a specific group is initialized to connect to the output units of the CD-HMM states that belong to this group more strongly than to other output units, as later explained in Figure 1 and Figure 2. Then we conduct cross-entropy training [19] and Hessian-free state-level Minimum Bayes Risk (sMBR) sequence training [20, 21] to train the NN-AM. Please note that our proposed method operates in NN parameter initialization and requires no computational overhead in NN training or speech recognition run-time.

The idea behind the proposed method is that the grouped CD-HMM states have some common acoustic characteristics and the dedicated neurons serve as the basis to represent these characteristics. Ideally, with enough training data and a novel optimization method, the NN-AM automatically learns these characteristics. In other words, our proposed method integrates prior knowledge of the relationship between the CD-HMM states into the NN-AMs for better optimization.

Due to the optimization difficulty in NN training, much attention has been paid to parameter initialization [22, 23, 24]. To the best of our knowledge, this is the first attempt to use the relation between the CD-HMM states for parameter initialization of NN-AMs.

We conducted experiments using a standard English broadcast news task and a Japanese LVCSR task and confirmed 3.6% to 6.5% relative error reduction after Hessian-free sequence training over a competitive baseline.

This paper has two main contributions:

• it proposes a method to initialize NN-AMs by considering relations between the CD-HMM output targets, and

• it confirms improvement in speech recognition accuracy using the proposed initialization on an English broadcast news task and a Japanese LVCSR task.
## 2. Proposed Method

As mentioned in Section 1, we focus on the NN-AM that estimates the posterior probabilities for the CD-HMM states and is combined with an HMM to form the NN/HMM hybrid system. The output targets of NN-AMs correspond to CD-HMM states and thus, there exist relationships between the states. For example, one set of CD-HMM states will correspond to the same CI-HMM state, while a still larger set of CD-HMM states will correspond to the same phone. Since the output targets are treated independently in the training of NN-AMs, the relationships between the output targets have not been exploited. We propose a method to consider these relationships in NN-AM initialization.

The key idea is to treat some of the neurons in the final hidden layer just below the output layer as dedicated neurons for each group of CD-HMM states. More specifically, the dedicated neurons are initialized to have connections to the corresponding CD-HMM states with a constant weight of \( C \) and to others with a weight of 0, as shown on the right of Figure 1 and Figure 2. This is equivalent to associating each group of CD-HMM states with a row in the weight matrix between the final hidden layer and the output layer and initializing this row so that the columns corresponding to the CD-HMM states in the group have a weight \( C \) and the other columns have a weight of 0, as shown on the left of Figure 1 and Figure 2. The remaining rows that are not associated with groups of CD-HMM states are randomly initialized. Finally, we perform cross-entropy training followed by Hessian-free sequence training. Note that all of the connection weights in the NN, including the connection weights between the dedicated neurons and all output targets, are updated during training.

We consider two variants of grouping based on phones or CI-HMM states. The proposed methods using these groupings, namely Phone-grouping initialization and CI-HMM-grouping initialization are described as below in detail. CD-HMM states are represented as \( X-Y-Z \), where \( X \) is the phone label; \( Y \) is \( b, m, \) or \( e \) corresponding to the beginning, middle, or end states of the HMM; and \( Z \) is the index of different variants. * represents a wild card, matching any label or index. We represent the number of phones as \( N_p \), the number of CI-HMM states as \( N_{ci} \), the number of CD-HMM states as \( N_{cd} \) (which is equal to the number of units in the output layer), and the number of units in the final hidden layer just below the output layer as \( N_h \). The size of the weight matrix our proposed method operates on is \( N_h \times N_{cd} \).

### Phone-grouping initialization

We prepare \( N_p \) dedicated neurons, where each one corresponds to a phone. The dedicated neuron for a phone \( X \) is initialized to have strong connections to the output targets of the CD-HMM states that belong to this phone \((X-\ast-\ast)\). In Figure 1, for example, a dedicated neuron is prepared for the phone \( A \), and this neuron is initialized to strongly connect to the output targets of \( A-b-1, A-b-2, A-m-1, A-e-1, \) and

![Diagram of Weight matrix between hidden and output layers](image1)

![Diagram of Strongly initialized connections between hidden and output layers](image2)
### 3. Experiments

We conducted two sets of experiments to confirm the advantage of our proposed method. First, we used a 50-hour English Broadcast News task (BN50). Then, we applied our proposed method to a larger task with 500 hours of Japanese data (JPN50).

#### 3.1. English Broadcast News (BN50)

The acoustic model training data was a 50-hour set of English broadcast news data from the 1996 and 1997 English Broadcast News Speech corpora [20]. Evaluation was done on the Defense Advanced Research Projects Agency (DARPA) Effective Affordable Reusable Speech-to-Text (EARS) dev04f set using Word Error Rate (WER) as the evaluation metric. A standard language model was used for evaluation, as described in [20].

<table>
<thead>
<tr>
<th>Initialization Value $C$</th>
<th>XE</th>
<th>HF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>n/a</td>
<td>17.0</td>
</tr>
<tr>
<td>PHONE</td>
<td>1.0</td>
<td>17.1</td>
</tr>
<tr>
<td></td>
<td>3.0</td>
<td>16.5</td>
</tr>
<tr>
<td></td>
<td>5.0</td>
<td>16.7</td>
</tr>
<tr>
<td></td>
<td>7.0</td>
<td>16.6</td>
</tr>
<tr>
<td></td>
<td>9.0</td>
<td>16.7</td>
</tr>
<tr>
<td>CI-HMM</td>
<td>1.0</td>
<td>16.9</td>
</tr>
<tr>
<td></td>
<td>3.0</td>
<td>16.6</td>
</tr>
<tr>
<td></td>
<td>5.0</td>
<td>16.6</td>
</tr>
<tr>
<td></td>
<td>7.0</td>
<td>16.0</td>
</tr>
<tr>
<td></td>
<td>9.0</td>
<td>16.5</td>
</tr>
</tbody>
</table>

Table 1: WER with DNN on BN50 [%]. Numbers in parentheses are Relative Error Reduction of best results compared with Baseline.

$\text{A-e-2}$. Thus, a $N_p \times N_{ci}$ region of the weight matrix is initialized by the proposed method to represent the $N_p$ dedicated neurons.

**CI-HMM-grouping initialization** We prepare $N_{ci}$ dedicated neurons, where each one corresponds to a CI-HMM state. The dedicated neuron for a CI-HMM state $X-Y$ is initialized to have strong connections to the output targets of the CD-HMM states that belong to this CI-HMM state ($X-Y$-). In Figure 2, for example, a dedicated neuron is prepared for the CI-HMM state $A-b$, and this neuron is initialized to strongly connect to the output targets of $A-b-1$ and $A-b-2$. Another dedicated neuron is prepared for the CI-HMM state $A-m$, and this neuron is initialized to strongly connect to the output target of $A-m-1$. Thus, a $N_{ci} \times N_{cd}$ region of the weight matrix is initialized by the proposed method to represent the $N_{ci}$ dedicated neurons.

$N_h \geq N_p$ is required for phone-grouping initialization, and $N_h \geq N_{ci}$ is required for CI-HMM-grouping initialization, but these requirements are usually satisfied in practical NN-AMs for LVCSR.

Please note that our proposed method does not change the NN architecture, such as the number of hidden units, at all. Therefore, considering that the computation required for the proposed initialization itself is negligible, our method does not increase computation in training or speech recognition runtime.

### 3. Experiments

We conducted two sets of experiments to confirm the advantage of our proposed method. First, we used a 50-hour English Broadcast News task (BN50). Then, we applied our proposed method to a larger task with 500 hours of Japanese data (JPN50).

#### 3.1. English Broadcast News (BN50)

The acoustic model training data was a 50-hour set of English broadcast news data from the 1996 and 1997 English Broadcast News Speech corpora [20]. Evaluation was done on the Defense Advanced Research Projects Agency (DARPA) Effective Affordable Reusable Speech-to-Text (EARS) dev04f set using Word Error Rate (WER) as the evaluation metric. A standard language model was used for evaluation, as described in [20].

<table>
<thead>
<tr>
<th>Initialization Value $C$</th>
<th>XE</th>
<th>HF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>n/a</td>
<td>15.6</td>
</tr>
<tr>
<td>PHONE</td>
<td>1.0</td>
<td>15.5</td>
</tr>
<tr>
<td></td>
<td>3.0</td>
<td>15.2</td>
</tr>
<tr>
<td></td>
<td>5.0</td>
<td>15.0</td>
</tr>
<tr>
<td></td>
<td>7.0</td>
<td>15.1</td>
</tr>
<tr>
<td></td>
<td>9.0</td>
<td>15.2</td>
</tr>
<tr>
<td>CI-HMM</td>
<td>1.0</td>
<td>15.6</td>
</tr>
<tr>
<td></td>
<td>3.0</td>
<td>15.2</td>
</tr>
<tr>
<td></td>
<td>5.0</td>
<td>15.0</td>
</tr>
<tr>
<td></td>
<td>7.0</td>
<td>14.9</td>
</tr>
<tr>
<td></td>
<td>9.0</td>
<td>15.0</td>
</tr>
</tbody>
</table>

Table 2: WER with CNN on BN50 [%]. Numbers in parentheses are Relative Error Reduction of best results compared with Baseline.

We first trained a GMM-AM according to our standard recipe [25] that yielded sufficiently accurate alignments using FMLLR transformed features [26] derived from VTLN-warped [27, 28] PLP features. Then, we trained two types of NN-AMs, a DNN and a CNN on this task. The details of the DNN and CNN are as follows.

**DNN** We used 40 dimensional FMLLR features with a 9-frame context (±4) around the current frame to train the DNNs. The DNNs had 5 hidden layers of 1,024 sigmoid units, 1 hidden layer of 512 sigmoid units, and the output layer of 5,000 units corresponding to the CD-HMM states estimated by the GMM-AM.

**CNN** We used 40 dimensional VTLN-warped log filterbank+delta+double-delta coefficients to train the CNNs. The CNNs had 2 convolutional layers, 5 hidden layers of 1,024 sigmoid units, 1 hidden layer of 512 sigmoid units, and the output layer of 5,000 units corresponding to the CD-HMM states estimated by the GMM-AM. The convolutional filter size was 9 × 9 (frequency-time) and 4 × 3 for the first and second convolutional layers, respectively. Max-pooling over frequency with a size of 3 was used for the first layer and no pooling was conducted for the second layer.

For the proposed method, the number of dedicated neurons was 42 with the phone-grouping initialization and 126 with the CI-HMM initialization. We tried various initialization weight values $C$ to investigate the effect of the magnitude of the initialization value.

For training the DNN AM and CNN AM, we conducted cross-entropy training followed by Hessian-free sequence training. The learning rate was controlled using a held-out set [29]. For evaluation, we combined the DNN and CNN with an HMM and formed DNN/HMM and CNN/HMM hybrid systems [1, 9, 10, 29].

The DNN and CNN evaluation results are reported in Table 1 and Table 2, respectively, where “XE” and “HF” stand for the
Before NN training, average weights of “Dedicated Neurons ⇒ Corresponding Outputs” and between all hidden units and output units after cross-entropy training (XE) and Hessian-free sequence training (HF).

Table 4: Average connection weights between dedicated neurons and corresponding outputs, between dedicated neurons and other outputs, and between all hidden units and output units after cross-entropy training (XE) and Hessian-free sequence training (HF).

In all cases, the weights between the dedicated neurons and the corresponding neurons are stronger than the weights between others; training did not substantially alter the prior structure imposed by our initialization.

4. Conclusions

We proposed a NN initialization method to use the relationship between output targets consisting of CD-HMM states. More specifically, we prepared dedicated neurons in the final hidden layer for groups of CD-HMM states corresponding to phones or CI-HMM states. The dedicated neuron for a specific group is initialized to strongly connect to the output units belonging to this group.

Acknowledgements

We would like to thank Bowen Zhou and Bing Xiang of IBM Watson for the fruitful discussion. We are grateful to Ryuki Tachibana of IBM Watson for valuable comments.
6. References


