MULTI-TASK LEARNING IN DEEP NEURAL NETWORKS FOR IMPROVED PHONEME RECOGNITION

Michael L. Seltzer and Jasha Droppo

Microsoft Research
Redmond, WA 98052 USA
{mseltzer,jdroppo}@microsoft.com

ABSTRACT

In this paper we demonstrate how to improve the performance of deep neural network (DNN) acoustic models using multi-task learning. In multi-task learning, the network is trained to perform both the primary classification task and one or more secondary tasks using a shared representation. The additional model parameters associated with the secondary tasks represent a very small increase in the number of trained parameters, and can be discarded at runtime. In this paper, we explore three natural choices for the secondary task: the phone label, the phone context, and the state context. We demonstrate that, even on a strong baseline, multi-task learning can provide a significant decrease in error rate. Using phone context, the phonetic error rate (PER) on TIMIT is reduced from 21.63% to 20.25% on the core test set, and surpassing the best performance in the literature for a DNN that uses a standard feed-forward network architecture.

Index Terms— Acoustic model, speech recognition, multi-task learning, deep neural network, TIMIT

1. INTRODUCTION

Recently, significant improvements in speech recognition accuracy has been obtained on a variety of tasks using hidden markov model (HMM) acoustic models based on a deep neural network (DNN) rather than the Gaussian mixture model (GMM) [1, 2, 3]. A DNN is essentially simply artificial neural network with many hidden layers between the inputs and outputs. In the context of DNN-HMM acoustic modeling, the network's task is to compute likelihoods that can be used for the emission probabilities of the HMM.

Whereas a conventional GMM acoustic model uses separate Gaussian mixtures for each acoustic model state, the DNN jointly estimates the posterior probabilities of each acoustic state using a single neural network. One of the benefits of using a neural network aocustic model is this shared representation for all acoustic classes.

When learning the parameters of an acoustic model, the goal is to have a high recognition accuracy on a set of test data that is not included in the training data set. As a proxy, we try in increase recognition accuracy on the training data, and hope that this also improves the same metric on the test data. When the performance on the test data is much worse than the performance on the training data, we say the model did not generalize well. To the extent that many different models may get the same performance on the training data, we should prefer the one that generalizes best.

Multi-task learning (MTL) has been proposed as a method of improving the generalization of a classifier by forcing it to learn more than one related task at a time [4]. When the classifier uses the same network to perform more than one related task, it learns the shared structure of the tasks. If the tasks are chosen appropriately, what is learned for one task can help the other tasks learn better.

Using MTL is an appealing approach to improving the generalization capability of a neural network as it does not introduce any additional decoding complexity at runtime compared to a standard single task network. The additional parameters in the network associated with the secondary tasks are used only to aid in the training of the network, and in particular, the shared parameters that are common to all tasks in the network. After training is complete, the portion of the network associated with the secondary tasks is discarded and the classification is performed identically to a conventional single task classifier.

Multi-task learning using neural networks has been previously applied to various speech and language related tasks. For example, MTL was used for noise robust speech recognition for an isolated digits task, where the network was trained to predict both the digit label and the clean speech feature vector given the observed noisy feature vector as input [5]. This work was further developed in [6] which added gender prediction as a third task and used a recurrent neural network. Multi-task learning has also been used in spoken language understanding [7, 8] and natural language processing [9].

In this work, we apply multi-task learning to the task of continuous phoneme recognition. Phoneme recognition using a DNN-HMM is typically performed using a network trained to predict context-independent phonetic states. We propose to augment this primary task with one of three candidate secondary tasks. In the first system, prediction of the phoneme identity is used as the secondary task. This enables acoustic states to learn which acoustic states may be similar via their shared common source phone. The second system uses the previous and subsequent acoustic states as secondary tasks, which informs the network about the temporal evolution of the labels. Finally, the third proposed system uses the prediction of left and right phonetic context as the secondary tasks. This enables the network to learn about context dependency during training, while still using simple context-independent acoustic model in decoding.

Because we are use the resulting DNN in an HMM recognition system, we refer to our proposed model as the MTL-DNN-HMM. We evaluate the performance of these three candidate tasks for multi-task learning through a series of phoneme recognition experiments using TIMIT. We show that our MTL-DNN-HMM acoustic model outperforms the previously best published feedfoward DNN-HMM system.

The rest of the paper is as follows. In Section 2, we review deep neural networks and how they can be used in an HMM-based speech recognizer. Multi-task learning is introduced in Section 3, and the three proposed secondary tasks are described in detail. The perfor-

mance of multi-task learning for phoneme recognition is evaluated in Section 4. Finally, we summarize our findings in Section 5.

2. DEEP NEURAL NETWORKS

A deep neural network (DNN) is simply a multi-layer perceptron (MLP) with many hidden layers between its inputs and outputs. In this section, we review the main ideas behind the MLP and show how it can be used as an acoustic model for speech recognition.

2.1. Multi-Layer Perceptrons

In this work a MLP is used to classify an acoustic observation $\mathbf{x}$ into an acoustic model state s. The MLP can be interpreted as a stack of log-linear models. Each hidden layer models the posterior probabilities of a set of binary hidden variables $\mathbf{h}$ given the input visible variables $\mathbf{v}$, while the output layer models the class posterior probabilities. Thus, in each of the hidden layers, the posterior distribution can be expressed as

$$p(\mathbf{h}_l|\mathbf{v}_l) = \prod_{j=1}^{N_l} p(h_{l,j}|\mathbf{v}_l), \quad 0 \leq l < L \tag{1}$$

where

$$p(h_{l,j}|\mathbf{v}_l) = \frac{1}{1+e^{(-z_{l,j}(\mathbf{v}_l))}}, \qquad z_{l,j} = \mathbf{w}_{l,j}^T \mathbf{v}_l + b_{l,j} \tag{2}$$

Each observation is propagated forward through the network, starting with the lowest layer ($\mathbf{v}_0 = \mathbf{x}$) . The output variables of each layer become the input variables of the next, i.e. $\mathbf{v}^{l+1} = \mathbf{h}^l$. In the final layer, the class posterior probabilities are computed using a soft-max layer, defined as

$$p(s|\mathbf{x}) = p(s|\mathbf{v}_L) = \frac{e^{(z_{L,s}(\mathbf{v}_L))}}{\sum_{s'} e^{(z_{L,s'}(\mathbf{v}_L))}} \tag{3}$$

Note that the equality between $p(s|\mathbf{v}_L)$ and $p(s|\mathbf{x})$ is valid by making a mean-field approximation [10].

In this work, networks are trained by maximizing the log posterior probability over the training examples, which is equivalent to minimizing the cross-entropy.

$$\mathcal{L} = \sum_t \log p(s_t|\mathbf{x}_t) \tag{4}$$

The objective function is maximized using error back propagation which performs an gradient-based update

$$(\mathbf{w}_{l,j}, b_{l,j}) \leftarrow (\mathbf{w}_{l,j}, b_{l,j}) + \eta \frac{\partial \mathcal{L}}{\partial(\mathbf{w}_{l,j}, b_{l,j})}, \quad \forall\, j, l \tag{5}$$

where η is the learning rate.

2.2. Pre-training DNNs

The gradient-based optimization in (5) can result in a poor local optimum, especially as the number of layers increases. To remedy this, pre-training methods have been proposed to initialize the parameters prior to back propagation. In some sense, this is similar to the manner in which maximum likelihood acoustic models are used as the initialization for discriminative training in traditional GMM-HMM acoustic models. The most well-known method of pre-training grows the network layer by layer in an unsupervised manner. This is done by treating each pair of layers in the network as a restricted Boltzmann machine (RBM) that can be trained using an objective criterion called contrastive divergence. Practical details about the pre-training algorithm can be found in [11].

2.3. Using the DNN as an acoustic model

The speech recognition system takes as input a sequence of frames representing the acoustic input signal $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$, and determines the most likely sequence of symbols $\mathcal{W} = \{\mathbf{w}_1, \ldots, \mathbf{w}_N\}$, typically words or phones, that correspond to that signal.

$$\hat{\mathcal{W}} = \text{argmax}_{\mathcal{W}} p(\mathcal{X}, \mathcal{W})$$

Inside the acoustic model, a set of hidden markov models describe the joint probability of the symbol sequence $\mathcal{W}$ and state sequences $\mathcal{S} = \{s_1, \ldots, s_T.\}$. In Viterbi decoding, the probability of any sequence $\mathcal{W}$ is found as the maximum over all acoustic state sequences $\mathcal{S}$ that could have produced it. Additionally, the probability of the acoustic sequence is independent of the word sequence given the state sequence.

$$p(\mathcal{W}, \mathcal{X}) \approx \max_{\mathcal{S}} p(\mathcal{W}, \mathcal{S}) p(\mathcal{X}|\mathcal{S})$$

The acoustic likelihood calculation we are interested in is embedded in $p(\mathcal{X}|\mathcal{S})$. Because $\mathcal{S}$ and $\mathcal{X}$ have the same length, we can write

$$p(\mathcal{X}|\mathcal{S}) = \prod_{t=1}^{T} p_{\mathbf{x}|s}(\mathbf{x}_t|s_t)$$

The value $p(\mathbf{x}_t|s_t)$ represents the likelihood of the acoustic observation at frame t for the given state of the acoustic model. Until recently, it was common to model and calculate it using a Gaussian mixture model (GMM). To perform speech recognition using a DNN acoustic model, the state emission likelihoods are instead computed from the posterior probabilities $p_{s|\mathbf{x}}(s_t|\mathbf{x}_t)$ generated by the DNN [12]. This can be done using Bayes' rule to reverse the conditioning.

$$p_{\mathbf{x}|s}(\mathbf{x}_t|s_t) = \frac{p_{s|\mathbf{x}}(s_t|\mathbf{x}_t) p_{\mathbf{x}}(\mathbf{x}_t)}{p_s(s_t)}$$

Because the probability $p_{\mathbf{x}}(\mathbf{x})$ is constant, it is irrelevant to the recognition process and can be safely ignored. The distribution over state labels $p_s(s)$ is estimated by counting frame labels in the training data.

In this work, the network is trained to predict context-indpendent states. The training data is labeled by Viterbi alignment of the observations to the acoustic states of a GMM-HMM system.

3. MULTI-TASK LEARNING

Multi-task learning [4] is a technique wherein a primary learning task is solved jointly with additional related tasks using a shared input representation. If these secondary tasks are chosen well, the shared structure serves to improve generalization of the model, and its accuracy on an unseen test set. When testing on unseen data, the secondary tasks can be safely ignored.

As discussed previously, the primary task for the DNN in this work is predicting the acoustic state given the observation. This is done by training the network using the objective function shown in (4).

When trained on the primary task only, the model is not told which states may be similar because they belong to the same phonetic label, how context may affect the acoustic realization, or how this classificaion fits as part of a HMM's trajectory through its state space. Because it predicts one acoustic state in isolation, it needs to learn these structures blindly from data.

In multi-task learning, the key aspect is choosing appropriate secondary tasks for the network to learn. When choosing secondary tasks for multi task learning, one should select a task that is related to the primary task, but gives more information about the structure of the problem.

3.1. Phone Label Task

The phone label task is designed to give the combined system hints about which acoustic states may be similar, because they share the same phone label. To create the phone label for each training example, the acoustic state symbol s_t is mapped down to its corresponding phone label w_t.

A secondary output layer is created. Like the primary output layer, it is fed by the final layer of hidden units from the DNN. It's contribution to the objective function is

$$\mathcal{F}_{\text{phone}} = \sum_t \ln p_w(w_t|\mathbf{x}_t).$$

3.2. State Context Task

The state context task uses the next and previous frame's acoustic state labels as the secondary learning tasks. This gives the model information about the time-evolution of the acoustic state that is missing from the primary task.

The secondary objective function measures the ability of the model to predict the current acoustic model state s_t as well as the previous and next acoustic model states, s_{t-1} and s_{t+1}, from the current acoustic observation $\mathbf{x}_t$. This is possible because in our system $\mathbf{x}_t$ is derived from multiple frames of acoustic data centered on frame t.

$$\mathcal{F}_{\text{scontext}} = \sum_t \ln p_{s_L}(s_{t-1}|\mathbf{x}_t) p_{s_R}(s_{t+1}|\mathbf{x}_t)$$

For the initial frame of every utterance, the previous acoustic model state is set to the initial silence state. The last frame of every utterance uses the final silence state as its next acoustic model state.

3.3. Phone Context Task

The phone context task is motivated by the past success of triphone based acoustic modeling. Although this context information is often useful in acoustic modeling, it is absent from the primary task.

The phone context task consists uses the left and right context phone labels as the secondary learning tasks. In this system, two output layers are added to the baseline system. All three output layers are connected to the same final layer of hidden units from the DNN.

The secondary objective funcion measures how well the the left and right phone labels l and r are predicted by the combined model.

$$\mathcal{F}_{\text{pcontext}} = \sum_t \ln p_l(l_t|\mathbf{x}_t) p_r(r_t|\mathbf{x}_t)$$

For every frame of training data, the left context phone and right context phone are found in the same way as a standard triphone-based acoustic model. The left context of the initial frame, and the right context of the final frame, are forced to be the silence phone.

4. EXPERIMENTS

Experiments were conducted on the TIMIT corpus [13]. This corpus contains continuous speech from 630 native English speakers, organized into eight dialects, with eight usable recorded utterances for each speaker. The core test set consists of twenty-four speakers, two males and one female from each of the eight dialect regions. The training set consists of the 462 speakers who do not speak any of the utterances contained in the core test set.

Recognition was performed using a set of 61 phoneme labels, with three possible states each, for a total of 183 possible acoustic states. The likelihood produced by the primary task was used as the emission probability of a HMM phonetic recognizer, and the likelihoods produced by the secondary tasks were discarded. The HMMs each had three states, enforced a strict left to right state ordering, and were combined with a standard bigram phonetic language model. After decoding, the 61 phone labels were collapsed into a set of 39 phone classes for scoring, following the example of [14].

4.1. Baseline DNN-HMM

A DNN for the baseline DNN-HMM system was trained with four hidden layers with 2048 hidden units in each layer. The input layer consisted of a context window of eleven frames of acoustic data formed from the target frame at time t and five previous and subsequent frames. Each frame was represented by 40 log mel filterbank coefficients plus their first and second order derivatives. Each feature vector was augmented with the energy of each frame and its derivatives. This resulted in each frame being represented by a 123-dimensional vector and an input layer that consisted of 1353 components. The output layer of the baseline system is a softmax layer, with one output for each of the 183 acoustic states. The supervised data for each frame is a one-hot encoding. The labels for the training data were determined by a Viterbi alignment of the training data to a baseline GMM-HMM recognizer.

The DNN was initialized using layer-by-layer unsupervised pre-training, and then discriminatively trained using twenty-five epochs of back propagation. A learning rate of 0.08 for the first 9 epochs and 0.002 for the remaining 16 epochs, with a momentum of 0.9. Back propagation was done using stochastic gradient descent in minibatches of 512 training examples. This DNN system produced a phone error rate (PER) of 19.43% on the dev set and 21.63% on the core test set. We believe this to be a strong baseline system with equivalent performance to the DNN system reported in [1], on the same task and using the same features and network architecture.

4.2. Multi-task learning DNN-HMMs

A series of experiments was performed to examine the performance of the proposed multi-task learning approaches. All experiments used the same network architecture as the baseline system with the exception of the output layer. In these experiments, the soft-max layer for the primary task of context-independent phonetic state classification was augmented with a secondary soft-max layer for the auxiliary tasks. In all experiments, the pre-trained network from the baseline system was used as the initialization for back propagation. We again performed twenty-five iterations of back propagation as in the baseline single task network. The same learning schedule was used but the learning rates were normalized across the different MTL experiments to account for the fact that the output layer changes in size depending on the number of secondary tasks.

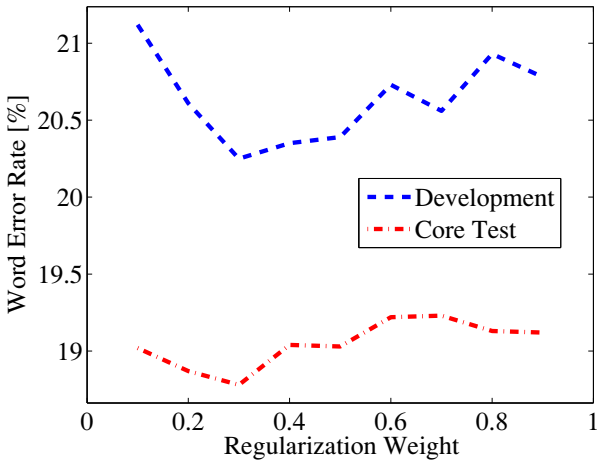

Fig. 1. Phone error rate on the dev and test sets as a function of the task weight α for multi-task learning using left and right context prediction as the secondary task.

Of course, adding an additional task increases the number of parameters in the output layer. However, because we are using a multi-task learning framework, these additional parameters are not used for classification by the primary task but rather serve to aid the training of the parameters of the network shared by both tasks. Once the network is trained, the output predictions for the secondary task are discarded and the posterior probabilities for the primary task are used in decoding exactly as in the baseline system.

The first multi-task learning system augments the primary task of context-independent phonetic state classification with a secondary task to predict the context-independent phone itself. This results in an additional soft-max layer that classifies the input into one of 61 phones. This increases the size of the output layer by 25%

The second multi-task learning system adds the prediction of the left and right phonetic context to the primary task. This results in a DNN that makes three predictions at its output, context-independent phone state, left phonetic context and right phonetic context, with class dimensionality of 183, 61, and 61, respectively.

The third multi-task learning system predicts left and right phonetic state labels in addition to the primary task. In this case, the dimensionality of each of the three tasks is 183.

4.3. Secondary Task Weight

In these experiments, the DNN was trained to maximize the following multi-task objective function

$$\mathcal{L}_{\text{MTL}} = \mathcal{L} + \alpha \mathcal{L}_{\text{aux}} \tag{6}$$

where $\mathcal{L}_{\text{aux}}$ was one of the three secondary tasks described in Section 3. This function replaces $\mathcal{L}$ in (4)

The optimal value of α in each case was determined by sweeping the value from 0 to 1 in increments of 0.1 and evaluating performance on the development set. The value which resulted in the highest score on the development set was chosen for decoding.

Figure 1 shows the phone error rate (PER) for the development and core test sets as a function of the task weight α for multi-task learning with phonetic context. The left most point ($\alpha = 0$) corresponds to conventional single-task learning while the rightmost point ($\alpha = 1$) gives equal importance to all tasks during training. In this case, both the development and core test sets have an optimum at 0.3.

DNN	Secondary Task	Task Weight	Dev PER (%)	Core Test PER (%)
4x2048	–	–	19.43	21.63
4x2048	Phone Label	0.7	19.04	21.53
4x2048	Frame Context	0.6	19.26	20.98
4x2048	Phone Context	0.3	18.78	**20.25**
8x2048 [1]	–	–	–	20.70

Table 1. Phone error rate on TIMIT for single and multi-task DNN-HMM systems.

4.4. Results

Table 1 compares the performance of the baseline DNN-HMM system and the three proposed multi-task learning systems. For the MTL systems, the selected task weight α is also shown. As the table indicates, using phone label classification improves the dev set performance but only marginally improves performance on the test set. An bigger improvement is obtained using the labels of the adjacent frames as context, while the best performance is achieved using multi-task learning with the prediction of left and right phonetic context.

The table also shows the performance of a DNN-HMM that has eight hidden layers but is otherwise identical to the baseline network used in this work [1]. This system achieved the best performance in the literature on this task for a DNN that uses a standard feed-forward network architecture. By using multi-task learning, a comparable system with only four hidden layers can achieve superior performance.

5. CONCLUSION

In this paper we propose a way to improve the performance of DNN acoustic models using multi-task learning. In multi-task learning, the network is trained to perform both the primary classification task and one or more additional related problems using a shared representation. If suitable secondary tasks are chosen, the network can leverage the common structure in the different tasks to learn a model with better generalization capability. Multi-task learning is attractive because the additional model parameters associated with the secondary task are used only in training and can be discarded at runtime. This means that improved performance can be obtained without any additional decoding complexity. For the primary task of context-independent phonetic state classification, three secondary tasks were proposed. The best performance was obtained using a system that used the prediction of the left and right phonetic context as the secondary tasks. The resulting MTL-DNN-HMM system significantly improved upon an equivalent single task network and outperformed a DNN-HMM with twice as many hidden layers. In the future, we will investigate the use of multi-task learning for context-dependent acoustic models, where the primary task is senone classification.

6. REFERENCES

[1] A. Mohamed, G.E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 14 –22, jan. 2012.

[2] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *Proc. Interspeech*, 2011.

[3] B. Kingsbury, T. N. Sainath, and H. Soltau, "Scalable minimum bayes risk training of deep neural network acoustic models using distributed hessian-free optimization," in *Proc. Interspeech*, 2012.

[4] Rich Caruana, "Multitask learning: A knowledge-based source of inductive bias," *Machine Learning*, vol. 28, pp. 41–75, 1997.

[5] S. Parveen and P. D. Green, "Multitask learning in connectionist ASR using recurrent neural networks," in *Proc. Eurospeech*, 2003.

[6] Y. Lu, F. Lu, S. Sehgal, S. Gupta, J. Du, C. H. Tham, P. Green, and V. Wan, "Multitask learning in connectionist speech recognition," in *Proc. Australian International Conference on Speech Science and Technology*, dec 2004.

[7] G. Tur, "Multitask learning for spoken language understanding," in *Proc. ICASSP*, may 2006.

[8] X. Li, Y.-Y. Wang, and G. Tur, "Multi-task learning for spoken language understanding with shared slots," in *Proc. Interspeech*, 2011.

[9] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *International Conference on Machine Learning, ICML*, 2008.

[10] L. Saul, T. Jaakkola, and M. I. Jordan, "Mean field theory for sigmoid belief networks," *Journal of Artificial Intelligence Research*, vol. 4, pp. 61–76, 1996.

[11] G. Hinton, "A practical guide to training restricted boltzmann machines," Tech. Rep. UTML TR 2010-003, University of Toronto, 2010.

[12] S. Renals, N. Morgan, H. Bourlard, M. Cohen, and H. Franco, "Connectionist probability estimators in hmm speech recognition," *Speech and Audio Processing, IEEE Transactions on*, vol. 2, no. 1, pp. 161–174, Jan.

[13] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, "The DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus CDROM," 1986.

[14] K.-F. Lee and H.-W. Hon, "Speaker-independent phone recognition using hidden markov models," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 37, no. 11, pp. 1641–1648, 1989.