Unsupervised Task Discovery in Multi-Task Acoustic Modeling

Josh Meyer

University of Arizona

joshua.richard.meyer@gmail.com

Abstract

This study investigates acoustic model training via Multi-Task Learning, where the auxiliary tasks are discovered in an unsupervised setting. Past work has shown that linguist-crafted auxiliary tasks can help train more robust acoustic models in low-resource settings. However, to create these tasks the researcher must have access to expert linguistic knowledge. The following study demonstrates that we can improve accuracy in a low-resource setting by training the acoustic model with an auxiliary task discovered via k-means clustering. Specifically, we train a Multi-Task DNN acoustic model, such that the model has two separate output layers which represent (1) traditional phonemes defined by a phonetic decision tree or (2) clusters of audio discovered by standard k-means clustering. Given only 1.59 hours of audio, we observed a 1.66% improvement in Word Error Rate, and in an extremely limited data setting, we observed a .78% improvement in WER. While these preliminary results show relatively small increases, this line of research promises easily scalable and unsupervised improvement in WER, and as such we believe warrants further exploration.

Index Terms: unsupervised learning, multi-task learning, acoustic modeling

1. Introduction

In the Multi-Task Learning (MTL) framework, data from a related task updates hidden layers in parallel with the target task [1]. A task here is defined as a mapping of data to labels, and as such we can create a new task by creating new labels for existing data. In general it is difficult to create relevant labels for a new classification problem. The current study investigates auxiliary tasks which are not hand-crafted by an expert or human, but automatically discovered from training data via unsupervised clustering (i.e. k-means clustering). The cluster identities are then assigned as target labels for DNN training via backpropagation.

2. Background

Past work in Mutli-Task acoustic modeling falls into two categories: monolingual vs. multilingual. Multilingual MTL acoustic modeling involves training a single DNN with multiple output layers, where each output layer represents triphones from a different language. [2, 3, 4, 5] Monolingual MTL acoustic modeling involves designing multiple tasks for a single language, where each task is a linguistically relevant classification: predicting triphones vs. predicting monophones vs. predicting graphemes. Multilingual MTL aims for language transfer, whereas monolingual MTL aims for generalization from the training data to unseen data.

With regards to monolingual MTL, research has aimed to find tasks (from the same language) which are phonetically relevant to the main task [6]. The aim being to improve generalization to new data. Both [7] and later [8] looked at a very similar approach, defining additional auxiliary tasks in MTL via broad, abstract phonetic categories for English. With regards to low-resource languages, [9] and later [10] created extra tasks using graphemes or a universal phoneset as extra targets.

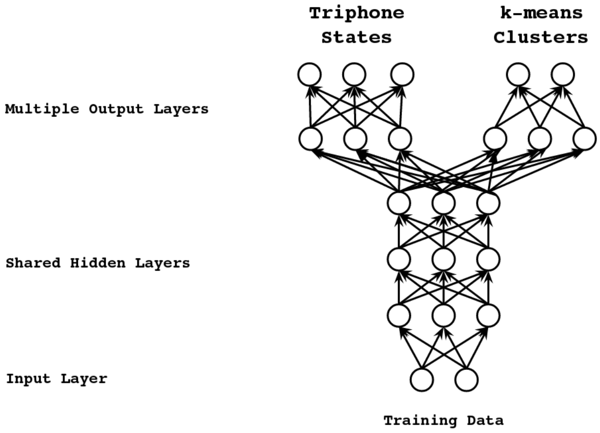

Figure 1: *Multi-Task Acoustic Model Architecture. Audio features are extracted via standard Kaldi scripts and then imported into TensorFlow. Standard k-means clustering is performed in TensorFlow, and the cluster identities are then exported back into Kaldi as targets for an auxiliary task. The final Multi-Task acoustic model is trained within Kaldi.*

The current work continues the monolingual line of research on Multi-Task Learning for acoustic modeling. We investigate automatic, unsupervised auxiliary task discovery in a low-resource setting.

3. Experiments

3.1. Data

The speech corpus used in the following experiments comes from an audiobook of a female speaker of Kyrgyz. A total of 1.59 hours of transcribed speech were used in training, and a held out 30 minutes were reserved for testing.

3.2. Model Building

Our experimental acoustic models are deep neural networks which have two separate output layers instead of just one (c.f. Figure (1)). The two output layers encode target labels for either (1) the main task (i.e. standard Kaldi triphone IDs) or (2) an auxiliary task (i.e. target labels which have been discovered via k-means clustering).

The intuition behind using k-means clustering is the following: unsupervised class discovery will surely lead to different classes compared to standard triphone GMM-HMM state alignment via the Baum-Welch algorithm. These k-means class assignments will be hopefully similar enough to the standard triphone labels that the clusters encode linguistically relevant categories. If the clusters are in fact linguistically relevant, then according to past work in MTL we should observe improved performance on the main task.

Instead of using raw k-means cluster labels for the auxiliary task, we first map all training examples which share a HMM triphone label to the same cluster. This ensures a certain degree of similarity between the main and auxiliary task. This effectively results in a merging of triphone labels, hopefully leading to a higher degree of linguistic abstraction.

3.2.1. Auxiliary Task Discovery

The new labels for the auxiliary task were discovered as such:

- Feature Extraction
  - 13 PLP features extracted via 25ms Hamming windows at a 10ms shift
  - Resulting vectors spliced to have context of 16 frames to the left and 12 frames to the right
  - CMVN normalization applied to each training example
- k-means Clustering
  - A set number of clusters is discovered via TensorFlow's standard k-means clustering.[1]
- Mapping triphone states onto k-means clusters
  - Given all training examples aligned to a given triphone state, the most commonly assigned k-means centroid is chosen as new target label for those examples. As such, training examples aligned to the same triphone will share the same k-means cluster.

During GMM alignment, monophones were allotted 1,000 Gaussian components, and trained over 25 iterations of EM. These monophones were then expanded into context-dependent triphones via a phonetic decision tree, with a maximum of 2,000 leaves & 5,000 Gaussians. The resulting tied-state triphones are then trained over 25 iterations of EM. The main GMM alignment script can be found on GitHub.[2].

Final models are trained in Kaldi as nnet3 Time-Delay Neural Networks (TDNNs) via a cross-entropy objective function [11, 12]. Given the alignments from the GMM-HMM models in addition to the discovered cluster assignments, a 5-layer, 1024-dimensional TDNN is trained over 2 epochs of backprop on a single GPU instance. The main neural net run script used in this paper can be found on GitHub.[3].

The main task (i.e. triphone state classification) and auxiliary task (i.e. cluster assignment classification) are implemented as separate output and penultimate layers. All other hidden layers of the TDNN are trained in parallel. A declining learning rate was used, with an initial $\alpha_{initial} = 0.0015$ and a final $\alpha_{final} = 0.00015$. A $ReLU$ activation function was used at every layer.

3.2.2. Baseline Model

The Single-Task baseline model has an identical architecture to the Multi-Task models without the additional task (5 hidden layers, 1024-dimensional layers, ReLU activations, same linear objective function).

3.3. Preliminary Results

During decoding, *only* the main task is used. This highlights the purpose of the auxiliary task: to force the learning of robust representations during training; the auxiliary task serves as "training wheels" which are removed once the net is ready.

Below is shown performance on the same held-out 30-minute section of Kyrgyz audiobook. Decoding is performed with a bigram backoff language model trained on a Wikipedia Kyrgyz dump, and contains, 103,998 unigrams and 56,871 bigrams. The bigram language model, lexicon, and main-task decision tree are built into a standard decoding graph (ie. a Weighted Finite State Transducer) in the traditional Kaldi pipeline. The experimental results are shown in Table (1) as percent Word Error Rate (WER).

Table 1: *Word Error Rates (WER%)*

	Amount of Training Data	
	1.59 hours	15 minutes
STL Baseline (context-dependent triphone targets)	49.56	83.51
+ 250 k-means cluster targets	48.88	83.71
+ 500 k-means cluster targets	47.90	82.83
+ 1000 k-means cluster targets	49.07	82.73

4. Discussion

When 1.59 hours of training data are used, every experimental condition shows improvement over the baseline, and 500 clusters shows the most improvement. In the extremely limited data condition (15 minutes of data), two of the three experiments showed improvement over the baseline, and we find a trend where more clusters correlates to more improvement.

This approach warrants further investigation, because we already observe improvements with very little hyper-parameter tweaking. We plan to investigate this auxiliary task discovery by adjusting (1) number of clusters, (2) relative weighting of main task to auxiliary task during backprop, (3) different feature projections before clustering, and other avenues.

5. Acknowledgements

I'd like to thank Dan Povey for answering my questions on the kaldi-help Google Group. I'd like to also thank Chorobek Saadanbekov and Murat Jumashev for making the Kyrgyz audiobook available to me through the Bizdin.kg group.

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. (DGE-1746060). Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors(s) and do not necessarily reflect the views of the National Science Foundation.

[1] TensorFlow k-means scripts: https://github.com/JRMeyer/kaldi-tf

[2] GMM alignment script: www.github.com/JRMeyer/multi-task-kaldi/blob/master/mtk/run_gmm.sh

[3] DNN training script: www.github.com/JRMeyer/multi-task-kaldi/blob/master/mtk/run_nnet3_multitask.sh

6. References

[1] R. Caruana, “Multitask learning,” *Machine Learning*, vol. 28, no. 1, pp. 41–75, Jul 1997. [Online]. Available: https://doi.org/10.1023/A:1007379606734

[2] J.-T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, “Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers,” in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7304–7308.

[3] G. Heigold, V. Vanhoucke, A. Senior, P. Nguyen, M. Ranzato, M. Devin, and J. Dean, “Multilingual acoustic models using distributed deep neural networks,” in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 8619–8623.

[4] Z. Tuske, D. Nolden, R. Schluter, and H. Ney, “Multilingual mrasta features for low-resource keyword search and speech recognition systems,” in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 7854–7858.

[5] F. Grézl and M. Karafiát, “Boosting performance on low-resource languages by standard corpora: An analysis,” in *Spoken Language Technology Workshop (SLT), 2016 IEEE*. IEEE, 2016, pp. 629–636.

[6] P. Bell and S. Renals, “Regularization of context-dependent deep neural networks with context-independent multi-task training,” in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4290–4294.

[7] M. L. Seltzer and J. Droppo, “Multi-task learning in deep neural networks for improved phoneme recognition,” in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 6965–6969.

[8] Z. Huang, J. Li, S. M. Siniscalchi, I.-F. Chen, J. Wu, and C.-H. Lee, “Rapid adaptation for deep neural networks through multi-task learning,” in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[9] D. Chen, B. Mak, C.-C. Leung, and S. Sivadas, “Joint acoustic modeling of triphones and trigraphemes by multi-task learning deep neural networks for low-resource speech recognition,” in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 5592–5596.

[10] D. Chen and B. K.-W. Mak, “Multitask learning of deep neural networks for low-resource speech recognition,” *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 23, no. 7, pp. 1172–1183, Jul. 2015. [Online]. Available: http://dx.doi.org/10.1109/TASLP.2015.2422573

[11] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, “The kaldi speech recognition toolkit,” in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[12] V. Peddinti, D. Povey, and S. Khudanpur, “A time delay neural network architecture for efficient modeling of long temporal contexts,” in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.