Deep Learning for Automatic Speech Recognition

Ryan Hartsfield Garrett Lewellen

May 4, 2015

Introduction

In this paper we examine two deep learning method for automatic speech recognition: deep neural networks and convolutional neural networks based on the works of [DYDA12] and [AMJ+14] respectively.

TODO: Ryan TODO

Traditional Approach

TODO: Ryan TODO

Deep Neural Networks Approach

In this section we discuss the work of [DYDA12] consisting of a context dependent hidden markov model and deep neural network hybrid architecture (CD-HMM-DNN) for the acoustic model. We will begin with an overview of the general architecture, then explain the the general procedure for training, and conclude with discussion of the algorithms used for pre-training. Discussion of experimental results for this approach are deferred to the results section so that they can be compared to the convolutional neural network approach.

Architecture

To motivate their architecture, [DYDA12] rely on the standard noisy channel model for speech recognition presented in [JM08] where we wish maximize the likelihood of a decoded word sequence given our input audio observations:

$$\hat{w} = \underset{w \in \mathscr{L}}{\operatorname{argmax}} \, \mathbb{P}(w|x) = \underset{w \in \mathscr{L}}{\operatorname{argmax}} \, \mathbb{P}(x|w) \, \mathbb{P}(w) \tag{1}$$

Where $\mathbb{P}(w)$ and $\mathbb{P}(x|w)$ represent the language and acoustic models respectively. [JM08] state that the language model can be computed via an N-gram approach, but [DYDA12] do not state their method, instead the authors put their efforts into explaining their acoustic model:

$$\mathbb{P}(x|w) = \sum_{q} \mathbb{P}(x, q|w)\,\mathbb{P}(q|w) \approxeq \max \pi(q_0) \prod_{t=1}^{T} a_{q_{t-1}q_t} \prod_{t=0}^{T} \mathbb{P}(x_t|q_t) \tag{2}$$

Here the acoustic model is viewed as a sequence of transitions between states of tied-state triphones which [DYDA12] refer to as senones TODO: EXPLAIN WHAT SENONES ARE AND WHERE THEY COME FROM AND WHY THEY ARE USEFUL which gives us the context dependent aspect of the architecture. The model assumes that there is a probability $\pi(q_0)$ for the starting state, probabilities $a_{q_{t-1}q_t}$ of transitioning to the state observed at step $t-1$ to step t, and finally, the probability of the acoustics given the current state q_t. [DYDA12] expand this last term further into:

$$\mathbb{P}(x_t|q_t) = \frac{\mathbb{P}(q_t|x_t)\,\mathbb{P}(x_t)}{\mathbb{P}(q_t)}$$

Where $\mathbb{P}(x_t|q_t)$ models the tied triphone senone posterior given mel-frequency cepstral coefficients (MFCCs) based on 11 sampled frames of audio TODO: EXPLAIN MFCCs AND WHY THEY ARE USEFUL, $\mathbb{P}(q_t)$ is the prior probability of the senone, and $\mathbb{P}(x_t)$ can be ignored since it does not vary based on the decoded word sequence we are trying to find.

Based on this formalism, [DYDA12] chose to use pre-trained deep neural networks to estimate $\mathbb{P}(q_t|x_t)$ using MFCCs as DNN inputs and taking the senone posterior probabilities as DNN outputs. The transitioning between events is bested modeled by hidden markov models whose notation, π, a, and q appears in Eqn. (2). Now that we have an overview of the general CD-DNN-HMM architecture, we can look at how [DYDA12] train their model.

TODO:

- Include figure of architecture?
- §10.3 of [JM08] discusses Context-dependent Acoustic models: triphones
- [LMM+14] talks about senone (tied-state triphones)

Pre-Training

TODO:

- pre-training / contrastive divergence for RBM
- Hinton's method for DBN

Training

Training of the CD-DNN-HMM model consists of roughly a dozen involved steps. We won't elaborate here on the full details of each step, but will instead provide a high-level sketch of the procedure to convey its general mechanics.

The first high-level step of the procedure is to initialize the CD-DNN-HMM model. This is done by first training a decision tree to find the best tying of triphone states which are then used to train a CD-GMM-HMM system. Next, the unique tied state triphones are each assigned a unique senone identifier. This mapping will then be used to label each of the tied state triphones. (These identifiers will be used later to refine the DNN.) Finally, the trained CD-GMM-HMM is converted into a CD-DNN-HMM by retaining the triphone and senone structure and HMM parameters. This resulting DNN goes through the pre-training procedure discussed in depth earlier.

The next high-level step iteratively refines the CD-DNN-HMM. To do this, first the originally trained CD-GMM-HMM model is used to generate a raw alignment of states TODO: figure out what it's aligning to which is then mapped to its corresponding senone identifier. This resulting alignment is then used to refine the DBN by backpropagation. Next, the prior senone probability is estimated based on the number of frames TODO: elaborate more on these frames somewhere paired with the senone and the total number of frames. These estimates are then used to refine the HMM transition probabilities to maximize the features. Finally, if this newly estimated parameters do not improve accuracy against a development set, then the training procedure terminates; otherwise, the procedure repeats this high-level step.

TODO: Discuss computational time to train

Convolutional Neural Networks Approach

TODO: Ryan TODO

Experimental Results

System Configurations

[DYDA12] report that their system relies on nationwide language model consisting of 1.5 million trigrams. For their acoustic model, then use a five hidden layer DNN with each layer containing 2,000 hidden units.

Datasets

TODO: Garrett

Results

		Bing Mobile		TIMIT	
	Architecture	Dev.	Test	Dev.	Test
Sentence Accuracy	CD-GMM-HMMM	70.3%	68.4%		
	CD-DNN-HMM	71.8%	69.6%		
Phone Error	CNN-HMM				20.07%

Table 1: Accuracy and error rates reported by [DYDA12] and [AMJ+14].

Direct comparison of the two systems is complicated by the fact that both papers report different metrics against different datasets. [DYDA12] reports a sentence level accuracy rate, while [AMJ+14] reports the phone error rate.

Conclusions

TODO: Garrett TODO

References

[AMJ+14] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu. Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on Audio, Speech & Language Processing*, 22(10):1533–1545, 2014.

[DYDA12] George E. Dahl, Dong Yu, Li Deng, and Alex Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech & Language Processing*, 20(1):30–42, 2012.

[Hin02] Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.

[HOT06] Geoffrey E. Hinton, Simon Osindero, and Yee Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.

[JM08] Daniel Jurafsky and James H. Martin. *Speech and Language Processing, 2nd Edition*. Prentice Hall, 2008.

[LMM+14] Haizhou Li, Helen M. Meng, Bin Ma, Engsiong Chng, and Lei Xie, editors. *INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association, Singapore, September 14-18, 2014*. ISCA, 2014.