In Depth Deep Beliefs Networks for Phone Recognition

Gabriel Synnaeve, Emmanuel Dupoux

LSCP at ENS, 29 rue d'Ulm, Paris 75005, gabriel.synnaeve@gmail.com

Abstract

We studied the effect of pre-training and fine-tuning on a well-known deep architecture for phone recognition [Mohamed, Dahl, Hinton, 2009]. Particularly, we looked at the phones classification errors at all layer depths of an architecture similar to a well-performing deep belief network. The insight gained this way calls for layer-depending learning rates and layer dependent early-stopping.

Model

Deep belief networks (DBNs) are one of the best performing acoustic modeling in hybrid hidden Markov models-DBN (HMM-DBN). An RBM is an undirected graphical model restricted to a bipartite graph on two layers $\mathbf{v}$ (visible) and $\mathbf{h}$ (hidden), each layer being a vector of Boolean random variables. The joint probability of an RBM is defined in terms of its energy:

$$p(\mathbf{v},\mathbf{h}) = \frac{\exp(-E(\mathbf{v},\mathbf{h}))}{\sum_{\mathbf{u}}\sum_{\mathbf{g}}\exp(-E(\mathbf{u},\mathbf{g}))}$$

$$\text{where : } E(\mathbf{v},\mathbf{h}) = -\sum_{i=1}^{V}\sum_{j=1}^{H} v_i h_j w_{i,j} - \sum_{i=1}^{V} b_i^v v_i - \sum_{j=1}^{H} b_j^h h_j$$

Thanks to the bipartite restriction, the conditional distributions factorize into products of Bernouilli distribution:

$$p(\mathbf{h}|\mathbf{v}) = \prod_j p(h_j|\mathbf{v}) \quad \& \quad p(\mathbf{v}|\mathbf{h}) = \prod_i p(v_i|\mathbf{h})$$

For the first layer, where we use MFCC (0 mean, 1 std. dev.) as features, we use a Gaussian-bernouilli RBM (GRBM).

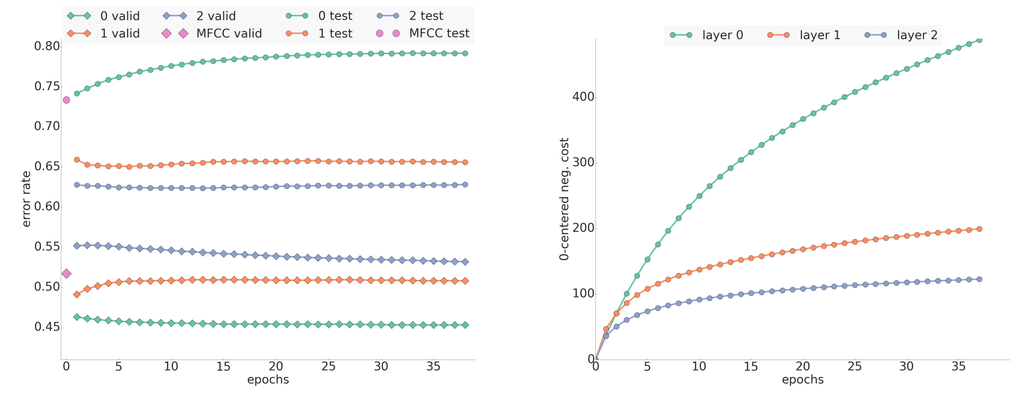
- pre-training: a single step of contrastive divergence, this approximates $\frac{\partial \log p(\mathbf{v})}{\partial w_{i,j}} = \mathbb{E}_{\text{data}}[v_i h_j] - \mathbb{E}_{\text{model}}[v_i h_j]$ by sampling $\mathbf{h}$ from $\mathbf{v}$, then $\mathbf{v}'$ from $\mathbf{h}$ and finally $\mathbf{h}'$ from $\mathbf{v}$. We update the weights by stochastic gradient descent (SGD):
$$\Delta w_{i,j} = \alpha(\mathbb{E}_{\text{data}}[v_i h_j] - \mathbb{E}_{\tilde{\text{model}}}[v'_i h'_j])$$
- fine-tuning: by back-propagation over all the stacked RBMs. We compute $C = -\log p(\mathbf{y} = \textit{phone's state}|\mathbf{v})$ with regard to each weight and update them by SGD as follows: $\Delta w_{i,j} = \lambda_t \frac{\partial C}{\partial w_{i,j}}$

Setup

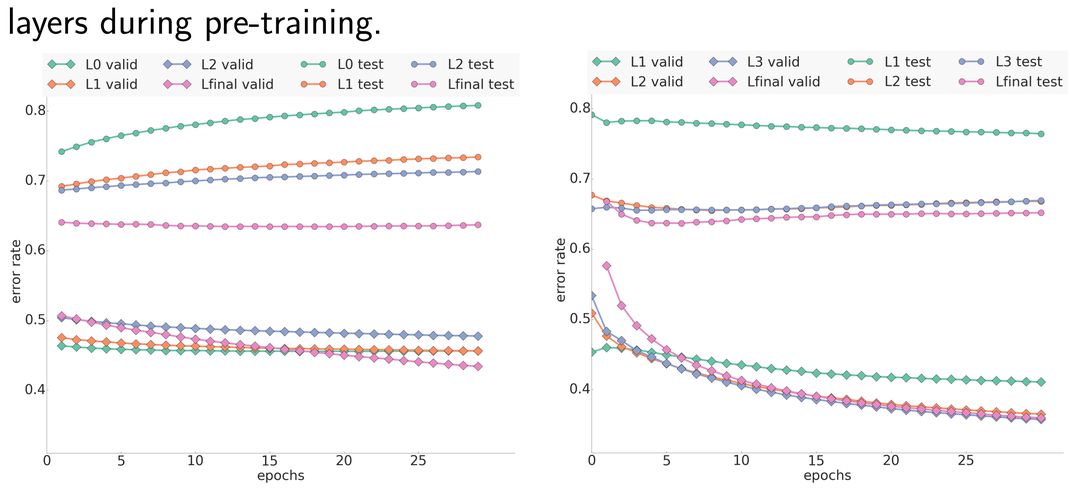
- Same architecture as in [Mohamed, Dahl, Hinton, 2009] [Mohamed, Hinton, Penn, 2012] on a restricted subset of TIMIT with a hard test set.
- 3 RBMs stacked with 1248 hidden units topped by a Softmax unit.
- Decaying learning rate: $\lambda_t = \frac{\lambda_0}{1+0.005t}$.
- 186 target class labels (60 phones + starting and ending silences, each with 3 states).
- First trained mono-phones HMMs with Gaussian mixture acoustic models with 17 components on TIMIT (without SA entries) using HTK (`http://htk.eng.cam.ac.uk/`) on 13 Mels Frequency Cepstral Coefficients (MFCC) with deltas and acceleration (39 coefficients in all).
- To check this model, we also trained it on the full TIMIT dataset (without SA shared sentences in the training set) and performed HMM decoding: the 62 phones were mapped to 40 classes for scoring and yielded a PER of 24.2%. Our HMM decoder and DBN code is available on Github at `https://github.com/SnippyHolloW/timit_tools`
- Experimental results are phone's states classification errors with and without pre-training, at each layer.

Results

On a restricted (hard) set of TIMIT:

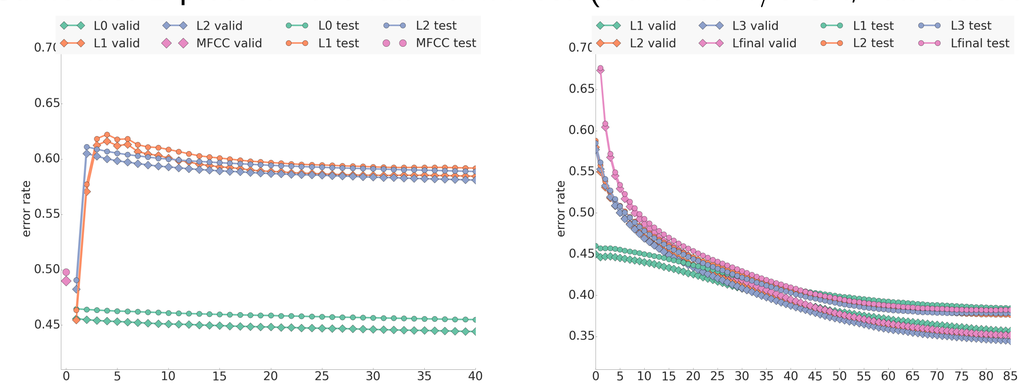


Left: evolution of the phones' states classification errors at all layers during the pre-training, along with the errors of the logistic regression trained on 13-frames of MFCC. Right: negative costs of the hidden layers during pre-training.



Evolution of the phones' states classification error at all layers during fine-tuning without (left) and with pre-training (right). On the right plot, epoch 0 represents the classification errors at the end of the pre-training.

Same last 2 plots on the full TIMIT set (train set w/o SA, full test set):



Conclusions

- Qualitative (order of errors) and quantitative (slopes) differences in hidden layer classification scores $\Rightarrow$ we think that this is a sufficient proxy to explain part of the behavior of the DBN.
- Both unsupervised pre-training [Erhan et al. 2010] and adding more depth [Mohamed et al. 2012] [Hinton et al. 2012] (more hidden layers) seem to help regularize the model.
- The discrepancy between the slopes of the classification errors at each layer indicates that learning should be adapted depending on the layer.
- Early stopping could also be layer dependent, which could prevent overfitting at the level of specific hidden layers.

References

- Mohamed, A., Dahl, G., Hinton, G. 2009. Deep Belief Networks for Phone Recognition, NIPS Workshop on Deep Learning for Speech Recognition and Related Applications.
- Mohamed, A., Hinton, G., Penn, G. 2012. Understanding how deep belief networks perform acoustic modelling, IEEE ICASSP.
- Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T., and others. 2012. Deep neural networks for acoustic modeling in speech recognition, IEEE Signal Processing Magazine.
- Erhan, D., Bengio, Y., Courville, A., Manzagol, P-A., Vincent, P., Bengio, S. 2010. Why Does Unsupervised Pre-training Help Deep Learning? JMLR.