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-dependent 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 v (visible) and h (hidden), each layer being a vector of Boolean random variables. The joint probability of an RBM is defined in terms of its energy:

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

where: \( E(v, h) = -\sum_{i, j=1}^{v} \sum_{i=1}^{H} v_i h_{i,j} - \sum_{i=1}^{v} b_i v_i - \sum_{j=1}^{H} b_{h,j} h_j \)

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

\[ p(h|v) = \prod_j p(h_j|v) \quad \text{&} \quad p(v|h) = \prod_i p(v_i|h) \]

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 by back-propagation over all the stacked RBMs. We compute

\[ \Delta w_{i,j} = \alpha \left[ \mathbb{E}_{data}[v_i h_j] - \mathbb{E}_{model}[v_i h_j] \right] \]

- fine-tuning: by back-propagation over all the stacked RBMs. We compute

\[ C = - \log p(y = \text{phone's state}|v) \]

with regard to each weight and update them by stochastic gradient descent (SGD):

\[ \Delta w_{i,j} = \lambda \frac{\partial C}{\partial w_{i,j}} \]

Results

On a restricted (hard) set of TIMIT:

![Graph showing 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.]

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):

![Graph showing 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.]

Conclusions

- Qualitative (order of errors) and quantitative (slopes) differences in hidden layer classification scores ⇒ 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


Setup

- 3 RBMs stacked with 1248 hidden units topped by a Softmax unit.
- Decaying learning rate: \( \lambda_t = 0.95^{t-0.0007} \)
- 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://htkr.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.