Shallow Thoughts on Deep Learning

Jeff Bilmes

University of Washington, Seattle.
Department of Electrical Engineering
http://ssli.ee.washington.edu/~bilmes

Dec 12th, 2009

Invited talk at NIPS 2009 Workshop “Deep Learning for Speech Recognition and Related Applications”, Li Deng, Dong Yu, Geoffrey E Hinton
Outline

• What is interesting about deeply layered models
• Some questions about deeply layered models
• Thoughts about deep models and speech recognition
Deep models and the brain

- The brain is an organ we are not yet able to understand but we wish to.
- The brain is something that we can only measure very coarsely (a small number of simultaneous neural probes).
- Uncertainty problem - we disturb the system as we measure it.
- A deep network is, like the brain, something we do not precisely know why it works (high dimensional space with a very non-linear classifier).
- Unlike the brain, a deep model is measurable, no inherent uncertainty problem. We may “see” it not only statically, but also in action in response to input, and perhaps its study will reveal techniques we can subsequently use to study the brain.
Deep models and privacy

- Presume, for the moment, deep (parametric) models and (say) non-parametric SVMs are roughly equal in performance.
- Parametric models are potentially more secure, as training data is only encoded, compressed, and summarized rather than explicitly represented with a small number of support vectors.
Deep models and privacy

- Presume, for the moment, deep (parametric) models and (say) non-parametric SVMs are roughly equal in performance.
- Parametric models are potentially more secure, as training data is only encoded, compressed, and summarized rather than explicitly represented with a small number of support vectors.

Spam classification, no permanent storage of support vector email messages.
Deep models are relatively easily trainable with version of contrastive divergence followed by stochastic gradient descent.

This is mostly matrix multiply, and with a little bit of bunching (grouping training items together), can be done with matrix-matrix multiply to get small integer factor speedups, even on a single serial machine.

See ICASSP-97 paper entitled “Using PHiPAC to speed Error Back-Propagation Learning”

Computation is inherently regular, and dense.

Other optimization methods are often not as easy to do low-level optimization.
Deep semi-supervised training

Standard back-propagation training objective

\[ J(\theta) = \sum_{i=1}^{\ell} D(t_i \parallel p_{\theta}(x_i)) + \lambda \|\theta\| . \] (1)

Where \( D(\cdot, \cdot) \) is the Kullback-Leibler divergence. Semi-supervised training objective applicable to deep belief networks.

\[ J(\theta) = \sum_{i=1}^{\ell} D(t_i \parallel p_{\theta}(x_i)) + \gamma \sum_{i,j=1}^{n} \omega_{ij} D(p_{\theta}(x_i) \parallel p_{\theta}(x_j)) + \kappa \sum_{i=1}^{n} D(p_{\theta}(x_i) \parallel u) + \lambda \|\theta\| . \] (2)

- Rather than just back-propagation, also use unlabeled data.
- Still easy to train, semi-supervised training scales to very large data sets using stochastic training.
- Our work at INTERSPEECH-2009 (few months ago).
Semi-supervised deep belief networks

\[
J(\theta) = \sum_{i=1}^{\ell} D(t_i \| p_{\theta}(x_i)) + \gamma \sum_{i,j=1}^{n} \omega_{ij} D(p_{\theta}(x_i) \| p_{\theta}(x_j)) + \kappa \sum_{i=1}^{n} D(p_{\theta}(x_i) \| u) + \lambda \| \theta \| .
\]

- An alternative would regularize the weights in the network with say $L_2$ norm rather than the answers like above.
- Our objective above asks for the \textbf{solution} to be smooth with respect to a graph-represented manifold, rather than the \textbf{implementation} be smooth with respect to the manifold.
- Flexibility to change implementation even if nearby in input — particularly well suited to deep belief networks.
- Doesn’t discourage sparse and/or distributed representations.
- For more details, see Malkin, Subramanya, Bilmes “\textit{On the Semi-Supervised Learning of Multi-Layered Perceptrons}” at Interspeech 2009.
Deep Adaptation

- Sometimes test condition is slightly different than training condition, i.e., $p_{\text{train}}(x, y) \neq p_{\text{test}}(x, y)$.
- Speech recognition has had effective “adaptation” algorithms for years (mostly of the form of affine transforms on the mean parameters of Gaussians).
- Adaptive training objective for deep networks

\[
J(\theta) = \sum_{i=1}^{\ell} D(t_i \mid p_\theta(x_i)) + \lambda \|\theta - \theta_0\|.
\]  

where $\theta_0$ are the parameters of a system trained on test data, and where we have a small number $\ell$ of samples of test data from the test data distribution.

- Works very well. See Li & Bilmes “Regularized Adaptation of Discriminative Classifiers”, ICASSP-2006
Shallow vs. Deep

For fixed number of free parameters, does deep provably have inherently more capacity (VC-dimension, Rademacher complexity, etc.) than shallow network for other than monotone weighted threshold circuits?

Need general mathematically theory to include all that a shallow network can do.

For the purpose of classification and natural signals, might not need all functions, only those that map input to output.
Shallow vs. Deep

- Shallow has potential better matrix-multiply optimizability, larger matrices for fixed total number of parameters.
- Fewer larger matrices rather than more smaller matrices.
- Less serialization via dependency of latter layers on earlier layers (but pipelining possible).
Deep and Pre-training Basis

- Deep networks require some form of pre-training.
- After pre-training, a couple rounds of “gentle” back-propagation will lead it to produce good solutions.
- Features that explain correlation of input might be good at explaining the labels as well, but they might do more than required.
- A form of “overcomplete basis” for the input - pre-training has no clue about what the network is ultimately going to be asked to do — when output doesn’t need all aspects of the input, this could be wasteful.
Deep and Pre-training Basis

- For example, classification tasks with very different labels but identical input data sets and features. Two examples:
  - **A1** Speech recognition - try to remove (or at least don't care about) the identity of the speaker as much as possible to concentrate on what is being said.
  - **A2** Speaker identification - try to remove (or at least don't care about) what is being said to concentrate on who is speaking.
  - **B1** Image emotion recognition - try to remove (or at least don't care about) the identity of the face as much as possible to concentrate on what the expression is (happy, sad, angry, etc.).
  - **B2** Image face identification - try to remove (or at least don't care about) emotion to concentrate on who is speaking.

- Pre-training would do the same at each, the result couldn't be optimal for both tasks.
- Quest: bring the labels into pre-training, could only lead to more efficient result of pre-training.
Is Old Just New?

Isn’t this just neural network research and didn’t it go out of style 15 years ago? The “old is new” phenomenon?
Isn’t this just neural network research and didn’t it go out of style 15 years ago? The “old is new” phenomenon?
Caveat about TIMIT Phone Recognition

- Good TIMIT phone recognition is great to have, but there has not really been much effort on that task in the past 10 years compared to the LVCSR Switchboard and Fisher corpora.
- TIMIT phone recognition results often do not generalize to Switchboard/Fisher.
- Note: phone recognition in general is indeed useful!!
- Hybrid HMM/MLPs of the past - locally discriminative at the frame level but not “globally discriminative” at the sequence level.
- But TANDEM features have helped quite a bit. Could be that “deep TANDEM” would do much better.
- Fritsch & Finke’s ACID/HNN framework (decomposition of many classes into a hierarchy of classes and have a network for each) would also work here — “Deep ACID/HNN”
Pre-training in speech recognition

- Has happened for many years. Pretty much all discriminatively trained HMM-based ASR systems must start with ML generatively trained one using EM.
- Issue of supervised vs. unsupervised pre-training. In speech, mostly used supervised pre-training, since labels were available.
- Could be that unsupervised training works better, but recall caveat before about overcomplete basis for input. Wasteful at best.
Deep sequential

- No reason it can’t be done and should definitely be tried.
- Various models already proposed (see this workshop).
Questions?

Any questions?