Discriminative Training of RBMs using Bhattacharyya Distance

Nicolas Le Roux, Hugo Larochelle and Yoshua Bengio
Dept. IRO, Université de Montréal
{*lerouxni,larocheh,bengioy*}*@iro.umontreal.ca*

Interest in deep networks has surged in the last couple of years since the discovery of efficient new techniques to train them (Hinton, Osindero, & Teh, 2006; Bengio, Lamblin, Popovici, & Larochelle, 2007; Larochelle, Erhan, Courville, Bergstra, & Bengio, 2007; Ranzato, Boureau, & LeCun, 2008). The principle underlying all these techniques is the use of an unsupervised criterion to pretrain the layers of the network. While this seems to yield improved performance on some datasets, the absence of discriminative criteria is troubling. Bengio et al. (2007) proposed a mixture of a generative and a discriminative criterion which seemed promising; we present here another mixture of such criteria in the context of classification which, besides improving overall performance, gives new insight on the existing Contrastive Divergence training criterion used by Hinton et al. (2006).

Introduction

A Restricted Boltzmann Machine (RBM) with n hidden units is a parametric model of the joint distribution between hidden variables $\mathbf{h}_i$ and observed variables $\mathbf{x}_j$, of the form

$$P(\mathbf{x}, \mathbf{h}) \propto e^{\mathbf{h}' W \mathbf{x} + b' \mathbf{x} + c' \mathbf{h}}$$

with parameters $\theta = (W, b, c)$. We consider here the case of binary units. It is straightforward to show that $P(\mathbf{x}|\mathbf{h}) = \prod_i P(\mathbf{x}_i|\mathbf{h})$ and $P(\mathbf{x}_i = 1|\mathbf{h}) = \text{sigm}(b_i + \sum_j W_{ji}\mathbf{h}_j)$, and $P(\mathbf{h}|\mathbf{x})$ has a similar form.
A Deep Belief Network (DBN) is a generative model where the top layer is an RBM and the lower layers form a sigmoid belief network. After being trained as a generative model, the parameters of the DBN can be fine-tuned on a discriminative task and eventually used as a standard feed-forward neural network.

Using Bhattacharyya distance to improve classification accuracy

The original training criterion for RBMs is to minimize the negative log-likelihood of the data:

$$\mathcal{L}(\theta) = -\frac{1}{N} \sum_{i=1}^{N} \log P_\theta(\mathbf{x}_i) \tag{1}$$

where P_θ is the marginal distribution induced over the visible units by the RBM with parameters θ. Its gradient with respect to θ is

$$\frac{\partial \mathcal{L}}{\partial \theta} = -\frac{1}{N} \sum_i \mathbf{x}_i \text{sigm}(W\mathbf{x}_i + c)^T + E_{\mathbf{x},\mathbf{h}}[\mathbf{x}\mathbf{h}^T] \tag{2}$$

Since the expectation over $\mathbf{x}$ and $\mathbf{h}$ is intractable, the contrastive divergence algorithm (Hinton et al., 2006) replaces it by a sample obtained by running an MCMC for a few steps. The idea behind the algorithm is to pull up the energy of points that are close to the training points instead of pulling up the energy of all the points in the space, which would be too expensive. The notion of closeness is defined by the RBM itself.
When computing the output for a test example, a feed-forward pass is performed through the network. The activations of the layer $k+1$ (denoted $\mathbf{h}$) given the activations of the layer k (denoted $\mathbf{x}$) are equal to

$$\mu_j = \text{sigm}\left(\sum_i W_{ji}\mathbf{x}_i + c_j\right) = P(\mathbf{h}_j = 1|\mathbf{x}) \tag{3}$$

Since μ will be used on the next layer to perform the classification task, we want to minimize the probability of having $\mathbf{x}$'s of different classes generating the same $\mathbf{h}$. To this end, we will add a discriminative term to this criterion to encourage $\mathbf{x}$'s belonging to different classes to generate different $\mathbf{h}$'s:

$$\mathcal{L}_B(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\log P_\theta(\mathbf{x}_i) + \frac{\lambda}{K}\sum_i \sum_{j/C(\mathbf{x}_i)\neq C(\mathbf{x}_j)} \log\left(\sum_{\mathbf{h}}\sqrt{P(\mathbf{h}|\mathbf{x}_i)P(\mathbf{h}|\mathbf{x}_j)}\right) \tag{4}$$

where $C(\mathbf{x})$ is the class of the training sample $\mathbf{x}$ and K is the number of pairs of examples belonging to different classes. The term $\sum_{\mathbf{h}}\sqrt{P(\mathbf{h}|\mathbf{x}_i)P(\mathbf{h}|\mathbf{x}_j)}$ is called the **Bhattacharyya distance** between the distributions $P(\mathbf{h}|\mathbf{x}_i)$ and $P(\mathbf{h}|\mathbf{x}_j)$. Jebara and Kondor (2003) used a similar metric to derive new kernels for support vector machines.
Denoting $F(\mathbf{x})$ the free energy of $\mathbf{x}$, i.e.

$$P(\mathbf{x}) = \frac{\exp[-F(\mathbf{x})]}{\sum_{\mathbf{x}_0}\exp[-F(\mathbf{x}_0)]}, \tag{5}$$

we have

$$\log\left(\sum_{\mathbf{h}}\sqrt{P(\mathbf{h}|\mathbf{x}_i)P(\mathbf{h}|\mathbf{x}_j)}\right) = -F\left(\frac{\mathbf{x}_i+\mathbf{x}_j}{2}\right) + \frac{F(\mathbf{x}_i)+F(\mathbf{x}_j)}{2} \tag{6}$$

This new form gives a deeper insight on what the Bhattacharyya distance term actually does. It specifies that the energies of two points belonging to different classes has to be lower than the energy of the point in between, thus creating an "energy barrier" between classes. This defines an algorithm very similar to the contrastive divergence with the difference that the point whose energy is pulled-up is chosen deterministically and data-driven instead of being stochastically chosen and model driven.

Experiments

We performed experiments on the MNIST dataset with and without background images using the following protocol:

1. we trained an RBM using the $\mathcal{L}_B$ cost for various values of λ
2. we computed the expected activations of the hidden layer for all the training samples and plugged them into an SVM
3. the hyperparameters (both of the RBM and of the SVM) have been selected using the classification error on a validation set.

Since it was too computationally expensive to do the sum over all pairs of points of different classes, we only computed it for the k nearest neighbours of every point with $k = 3, 5, 20$. In a real-world problem, finding the k nearest neighbors would be too prohibitive and we would have to rely on an approximate method such as kd-trees.
Using the discriminative criterion helped the classification accuracy, though by a small margin. We believe a careful selection of the pairs on which to maximize the Bhattacharyya distance could yield a larger improvement. We obtained poor results using $\lambda = +\infty$ (thus only considering the discriminative criterion). This suggests that the stochastic exploration of the space performed by contrastive divergence is useful. An explanation for this is that the discriminative criterion only tries to minimize the energy of training points **relative to their midpoint** without preventing other points of the space to have very low energies.

References

Bengio, Y., Lamblin, P., Popovici, D., & Larochelle, H. (2007). Greedy layer-wise training of deep networks. In Schölkopf, B., Platt, J., & Hoffman, T. (Eds.), *Advances in Neural Information Processing Systems 19*, pp. 153–160. MIT Press.

Hinton, G. E., Osindero, S., & Teh, Y. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, *18*, 1527–1554.

Jebara, T., & Kondor, R. (2003). Bhattacharyya and Expected Likelihood Kernels. In *COLT '03*.

Larochelle, H., Erhan, D., Courville, A., Bergstra, J., & Bengio, Y. (2007). An empirical evaluation of deep architectures on problems with many factors of variation. In *Twenty-fourth International Conference on Machine Learning (ICML'2007)*.

Ranzato, M., Boureau, Y.-L., & LeCun, Y. (2008). Sparse feature learning for deep belief networks. In Platt, J., Koller, D., Singer, Y., & Roweis, S. (Eds.), *Advances in Neural Information Processing Systems 20*. MIT Press, Cambridge, MA.