CSC2535 Two Suggestions for Projects

Geoffrey Hinton

February 24, 2011

The projects are due on april 20. If you need an extension, please talk to me before the deadline.

1 Project 1: Comparing Restricted Boltzmann Machines with Mixtures of Multivariate Bernoullis

The aim of this project is to compare the log probabilities assigned to held-out binary data by several different models.

The data consists of binary occurence vectors for 100 words in 16242 postings from the 20 Newsgroups dataset. It can be loaded as a 100x16242 matlab matrix from:

`http://www.cs.toronto.edu/~hinton/20news_w100.mat`

You should use the last 5000 word-count vectors as the test data. You should also use a validation set to select the best version of each of the four models for testing.

The first model is a mixture of multivariate Bernoullis. Each component of the mixture specifies a probability for each binary word count and the word counts are treated as independent given the mixture component. You should use EM to fit this model to the data. Evaluating the log probability of the 5000 test cases should be straight-forward, but you might need to think about whether you want components of the mixture to assign probabilities of zero to some of the binary word counts.

The other models should all be Restricted Boltzmann Machines, but they should be learned using diffferent variations of the CD learning procedure.

Practical advice on how to train an RBM and on different version of CD can be found at:

`http://www.cs.toronto.edu/~hinton/absps/guideTR.pdf`

The tilde in the URL in this .pdf file does not work so you will have to replace it.

To compute the log probability that a learned RBM assigns to your test data, you will need to compute the negative free energy (the goodness) that the model assigns to each test vector and subtract the log of the partition function for that RBM. A way of estimating the partition function will be described in lecture 8. You can also read about it in “On the quantitative analysis of deep belief networks”, Salakhutdinov & Murray, ICML (2008). Matlab code for computing the log of the partition function is provided at:

```
http://www.cs.toronto.edu/˜hinton/code/partition
```

The tilde in the URL in this .pdf file does not work so you will have to replace it.

The files are:

```
free_energy.m -- computes free energy for use in RBM_partition.m
logdiff.m and logsum.m  -- used for computing log(Z+-3std)
RBM_partition.m -- is the main engine for calculating Z
demo.m -- loads some data and an RBM model trained with CD25,
runs RBM_partition.m and returns log(Z),log(Z+3std),log(Z-3std).
Occasionally you may get logZZ_down=0.000000.
If this happens just try again.
```

Computing the partition function accurately can take a long time, so you may only be able to do it a few times while learning an RBM. You can get a noisy estimate more quickly by using fewer intermediate distributions, but 5000 is already a very small number.

As an independent check you should find at least one way to compare the quality of the various models that does not require you to compute the partition function of the RBMs.

2 Project 2: Comparing different energy functions for modeling image patches using contrastive backpropagation

First, replicate the model in the notes for lecture 3a that uses two layers of logistic hidden units to learn to model a two-dimensional density composed of four squares that contain the data. The hybrid Monte Carlo method that you will need to use is explained in "Probabilistic inference using Markov chain Monte Carlo methods", Neal (1993). For this example, it is probably sufficient to use CD_1 without any repetitions of the choice of random momentum in the trajectory that is used to get the "negative" data.

Once your code works on this toy problem, try using contrastive backpropagation with multi-layer feedforward neural nets of various designs with various energy functions to learn a model of the 8x8 image patches that can be found at:

```
http://www.cs.toronto.edu/˜hinton/data/patches.mat
```

The tilde in the URL in this .pdf file does not work so you will have to replace it.

You might need to use a relatively small training set with a relatively small feedforward net in order to get your experiments finished in time.

Your report should discuss the effects of using different feed-forward architectures, different energy functions and different versions of the training procedure. You obviously do not have time to systematically explore all of these variations so it would be sufficient to have one "standard" model and to report the effects of one sensible variation of the network architecture, one sensible variation of the training procedure and one sensible variation of the energy function.