Introduction to Deep Learning 2016-2017

Instructor: Iasonas Kokkinos, i.kokkinos@cs.ucl.ac.uk

Assignment 2: Boltzmann Machine (4 Points/20) Deliverable: 5/1/17

1 Introduction

In this exercise we will consider the training of a Boltzmann machine for the 'shifter' example discussed in class - and presented in p. 524-525 of D. MacKay's book. We will compare (i) Boltzmann machines (BMs), (ii) Restricted Boltzmann Machines (RBMs) and (iii) the Ising model. As in Assignment 1, we use a small state vector, so that brute force enumeration of all states is possible - and we will then compare brute force and Monte Carlo-based techniques.

Necessary background for solving the assignment In the file 'theory.pdf' in Dropbox you can find a formal presentation of notions that we covered in class.

Please read the document carefully before trying to solve the assignments. The code assumes that you have understood this text, so it is not extensively documented.

2 Analytic exercise (0.5 Points)

As detailed in the supplement to the theory, given a set of M training examples $\mathbf{X} = \{\mathbf{x}^1, \ldots, \mathbf{x}^M\}$, Boltzmann machine training aims at estimating $\mathbf{W}$ (or $\mathbf{W}_{\mathbf{x},\mathbf{x}}, \mathbf{W}_{\mathbf{x},\mathbf{h}}$ for Ising/RBM training respectively) so as to maximize the log-likelihood $\mathbf{X}$:

$$\begin{aligned} S(\mathbf{X}, \mathbf{W}) &= \sum_{m=1}^{M} \log P(\mathbf{x}^m; \mathbf{W}) && (1) \\ &= \sum_{m=1}^{M} \log \sum_{\mathbf{h}} \frac{1}{Z} \exp(-E(\mathbf{x}^m, \mathbf{h}; \mathbf{W})) && (2) \end{aligned}$$

The partial derivative of this quantity with respect to parameter $\mathbf{W}_{k,m}$ is given by:

$$\frac{\partial S(\mathbf{X}, \mathbf{W})}{\partial w_{k,l}} = \sum_{m=1}^{M} \left[\langle \mathbf{y}_k \mathbf{y}_l \rangle_{P(\mathbf{h}, \mathbf{x}^m; \mathbf{W})} - \langle \mathbf{y}_k \mathbf{y}_l \rangle_{P(\mathbf{h}, \mathbf{x}; \mathbf{W})} \right]. \quad (3)$$

This expression shows up Lecture 3 - understanding it is most important for the rest of your assignment.

Show in full detail the steps involved in obtaining this expression.

- Hint: think about how we obtained the update rule for the exponential family, shown in Lecture 2.

- Hint: You may find a more extensive presentation regarding this update rule in Eq. 43.17 on p. 525 of D. MacKay's book `http://www.inference.phy.cam.ac.uk/itprnn/book.pdf`. T
- Note: Here we are summing over m, while in the code of your assignments we are averaging. This allows us to use the same update step for gradient ascent, irrespective of the size M of the training set.

3 Exact summations (1.5 Points)

Use brute-force summation to compute the exact value of the gradient of the log-likelihood for the Ising Model, the Boltzmann Machine and the Restricted Boltzmann Machine. BM/Ising model).

Implement gradient ascent, based on the code template provided for you, and compare the log-likelihood of the data under the three different models for $M = 8$).

If your code is working correctly, you should be getting the plots shown below.

What do you observe? How can you explain your findings? (no more than five lines of text are needed in your reply).

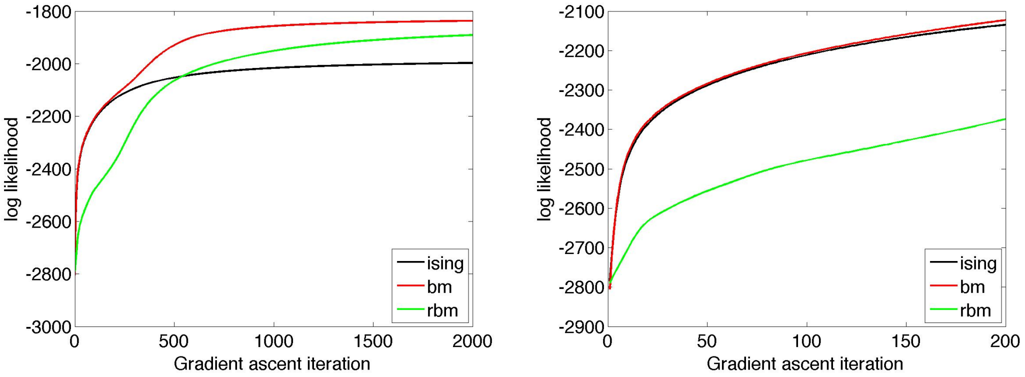

Figure 1: Expected plots of log-likelihood as a function of iteration using brute-force estimation of the gradient. The left plot shows the desired result, the right plot shows the first 200 iterations (in order to help you debug your code).

To help you debug your code, you are also provided with intermediate results for the first 200 iterations of gradient ascent - so you will not need to wait until you see if your algorithm deviates from delivering the desired result. You may also want to set $M = 2$, or $M = 4$ during debugging, in order to accelerate your computations.

4 Block-Gibbs sampling and Contrastive Divergence (2 Points)

- Implement Block-Gibbs sampling as a routine that takes as input $\mathbf{x}^0, \mathbf{W}, L$ and gives as output $\mathbf{x}^L, P(\mathbf{h}|\mathbf{x}^L)$ (see `blocks_gibbs.m` for a template).
- Use contrastive divergence with $L = 1$ and $L = 10$ to train a restricted Boltzmann machine with $M = 8$ hidden variables. Plot the log-likelihood of your two models, and compare it to the log-likelihood under the respective RBM obtained with brute-force enumeration. What do you observe?

Feel free to check existing efficient implementations of Contrastive Divergence for RBMs, e.g. from `http://www.cs.toronto.edu/~rsalakhu/code_AIS/rbm.m`. Signs, scaling factors, etc. are different in there (since a different notation is assumed), so you will need to eventually write the code on your own.

Include the plots generated by your code in your report. You should be getting something like the following plot. What do you observe? How can you explain your findings? (no more than five lines of text are needed).

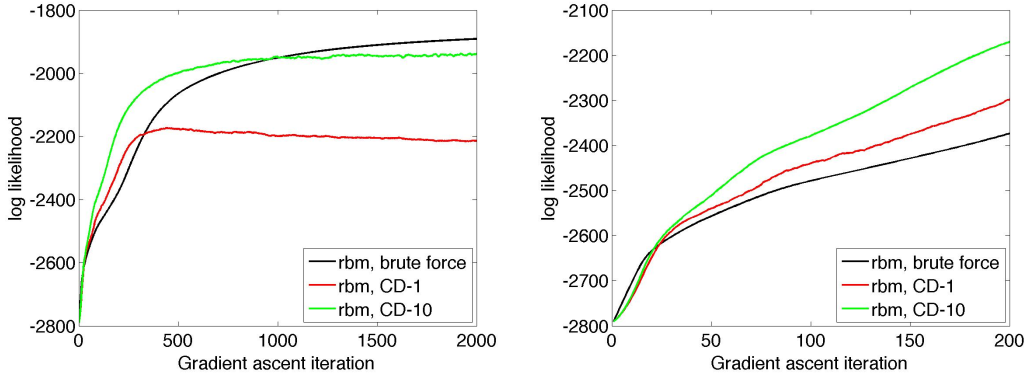

Figure 2: Expected plots of log-likelihood as a function of iteration using Contrastive Divergence. The left plot shows the desired result, the right plot shows the first 200 iterations.

4.1 Fun Part (0 Points)

- Perform Block-Gibbs sampling on your RBM and see what kind of bars are generated for $K = 10, 20, 30, \ldots$. Do they preserve the 'shifting' constraint in the training data?
- Repeat Contrastive Divergence training with larger models (set e.g. `sz_half = 10, n_hidden = 20`); for this you will need to set `diagnostic = 0;` or else you will run out of memory. Check if the synthesized samples respect the constraint.