[M2, MVA]

Deep Learning

Maha ELBAYAD

maha.elbayad@student.ecp.fr

Homework 2

December 10, 2015

1 Analytic exercise

1. Given a set of training samples $X = \{x^1, ...x^M\}$. The loglikelihood of X given W is:

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

where

$$E(x^m,h;W) = E(y;W) = -\frac{1}{2} y^T W y \text{ (with } y = [x^m \;\; h]^T) \text{ and } Z = \sum_y \exp(-E(y;W))$$

We start with the derivative of $\log Z$ with respect to the weight w_{ij}:

$$\begin{aligned} \frac{\partial \log Z}{\partial w_{ij}} &= \frac{1}{Z} \sum_y \frac{\partial}{\partial w_{ij}} \exp(\frac{1}{2} y^T W y) \\ &= \frac{1}{Z} \sum_y y_i y_j \exp(\frac{1}{2} y^T W y) \qquad \text{(taking into consideration that } W \in \mathbf{S}^{N+K}) \\ &= \sum_y y_i y_j P(y;W) = \langle y_i y_j \rangle_{P(y;W)} \end{aligned}$$

Similarly:

$$\begin{aligned}
\frac{\partial}{\partial w_{ij}} \sum_m \log \sum_h \exp(-E(x^m, h; W)) &= \sum_m \frac{\sum_h y_i y_j \exp(-E(x^m, h; W))}{\sum_h \exp(-E(x^m, h; W))} \\
&= \sum_m \sum_h y_i y_j P(h|x^m; W) \\
&= \sum_m \langle y_i y_j \rangle_{P(h, x^m; W)}
\end{aligned}$$

Therefore:

$$\begin{aligned}
\frac{\partial}{\partial w_{ij}} S(X, W) &= \sum_m \langle y_i y_j \rangle_{P(h, x^m; W)} - M \langle y_i y_j \rangle_{P(y; W)} \\
&= \sum_m \left[\langle y_i y_j \rangle_{P(h, x^m; W)} - \langle y_i y_j \rangle_{P(y; W)} \right]
\end{aligned}$$

2 Exact summations

We compute the exact value of the gradient of the log-likelihood with brute force for the Ising Model, the BM and the RBM to implement gradient ascent for each model on the log-likelihood of the data. In our model we use 8 hidden units (BM,RBM) with a learning rate of 0.1

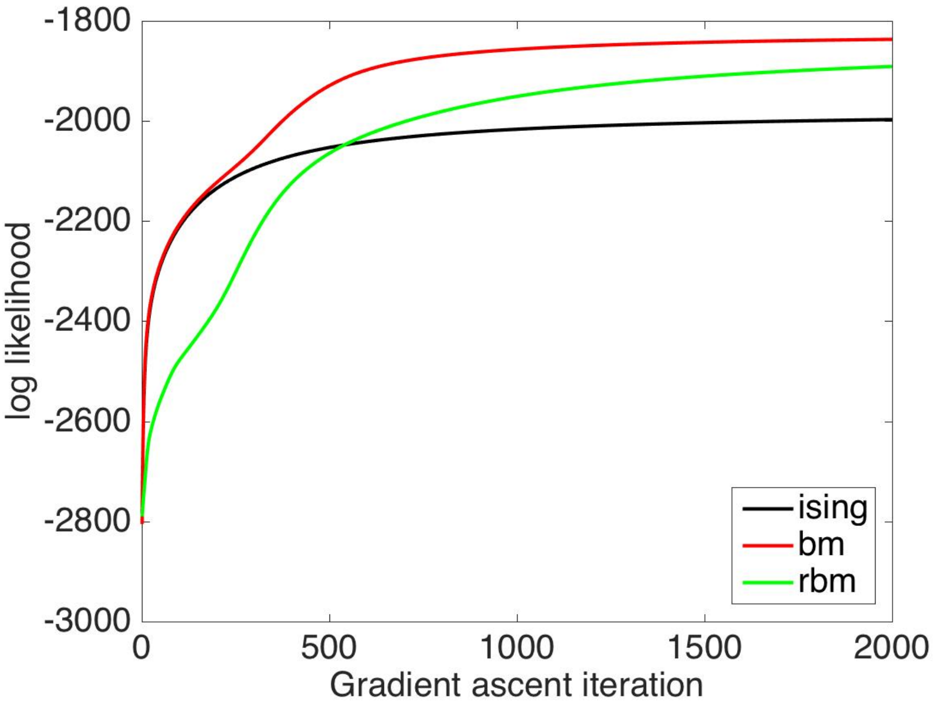

Figure 1: Brute force: Gradient ascent log-likelihhod maximization
Ising - bm - rbm

As we expected the BM with more degrees of freedom reached a higher log-likelihood than the two others. The ising model, as it doesn't capture higher-order statistics, scores the lowest log-likelihood. We can also note that the convergence rates of the BM and ising model are higher than that of the RBM.

3 Block-Gibbs sampling and Contrastive Divergence

In this section we use contrastive divergence with L = 1 and L = 10 to infer an RBM with 8 hidden units.

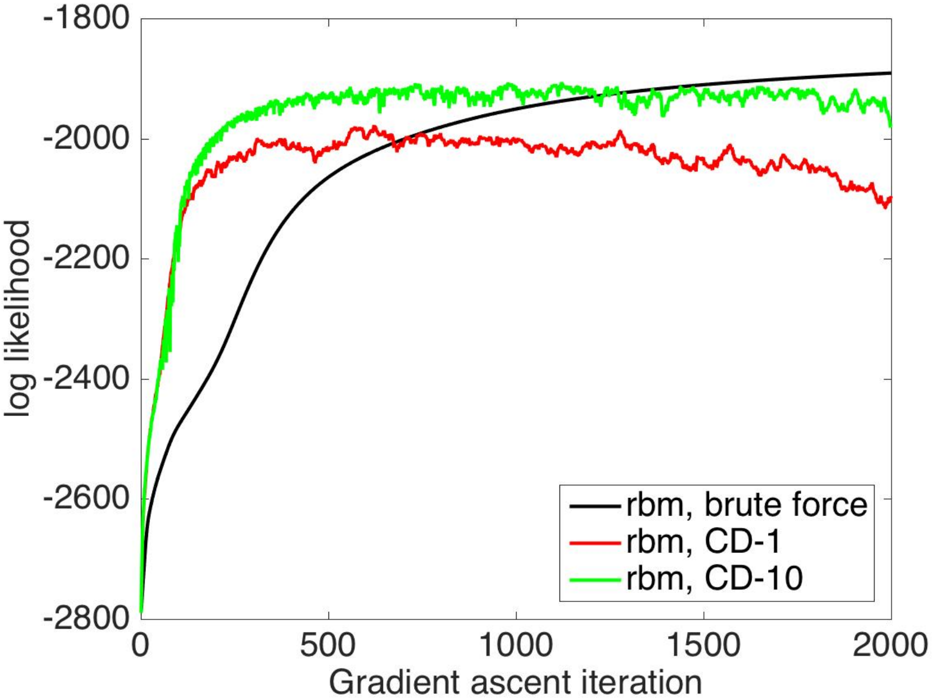

Figure 2: CD: Gradient ascent log-likelihhod maximization
rbm-cd1 - rbm-cd10

The intuition behind CD is that we would like the MC implemented by Gibbs sampling to conserve the distribution over the visible variables. For this we run the chain L=1 (resp L=10) then update the rbm weights to correct the chain. The larger is L the closest we are to the MC's equilibrium, which explains why rbm-cd-10's likelihood is closer to the brute force estimation. Moreover, we observe that the log-likelihood is not non-decreasing over iterations since the CD is just an approximate noisy gradient ascent.

4 Fun part

4.1 Q1:

We sample from the learned rbm brute force, CD1, CD10 with Gibbs-sampling using K ranging from 10 to 100. During training we track the network precision as the probability of satisfying the shifting constraint (figure 3).

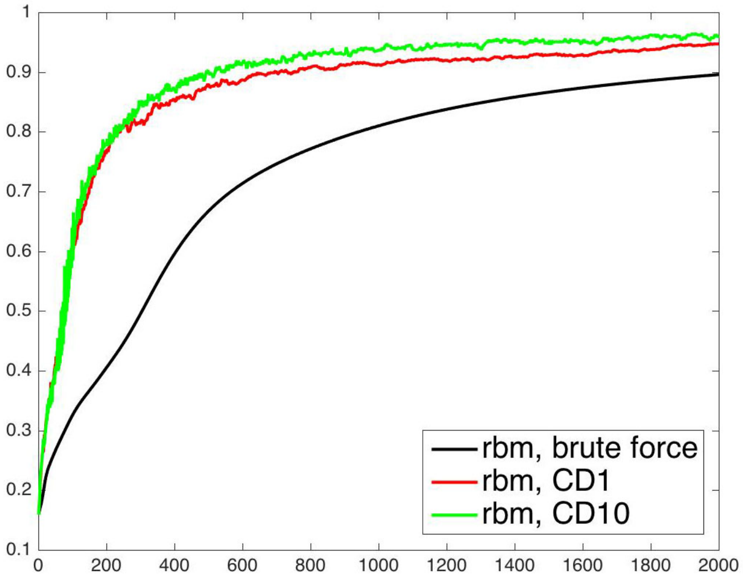

Figure 3: Network precision

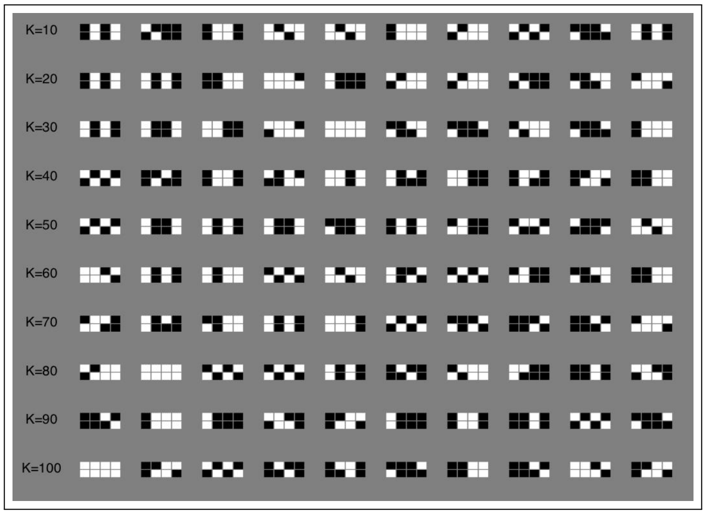

Figure 4: Samples - brute force

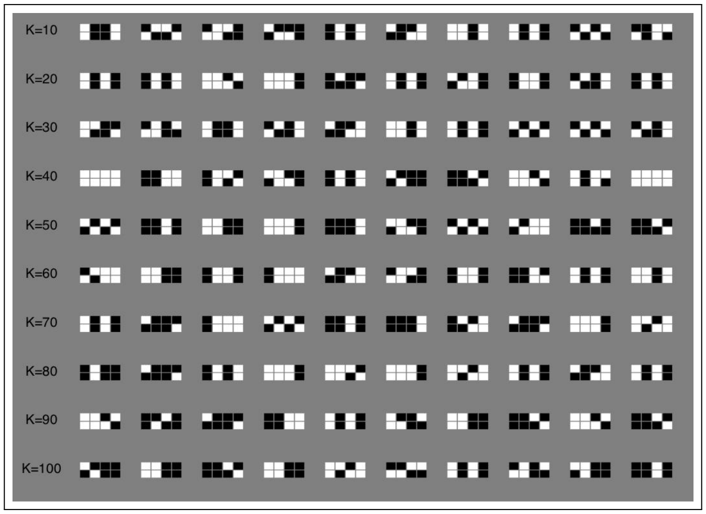

Figure 5: Samples - CD1

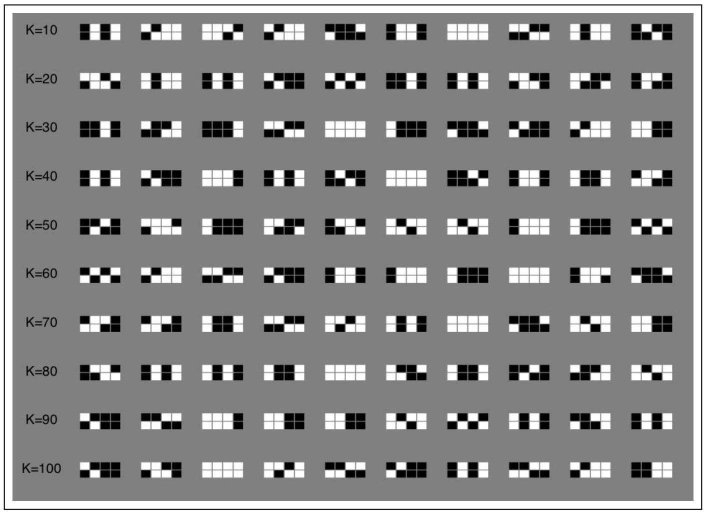

Figure 6: Samples - CD10

K	**10**	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
rbm, brute force max=0.8958	0.80	0.83	0.84	0.86	0.87	0.88	0.87	0.87	0.89	0.91
rbm, CD1 max=0.9474	0.72	0.77	0.82	0.84	0.84	0.87	0.91	0.89	0.93	0.92
rbm, CD10 max=0.9649	0.87	0.90	0.88	0.89	0.92	0.90	0.90	0.91	0.92	0.91

Table 1: Shifted samples (out of 500)

We observe that the generated samples meet the precision of the network and in general the longer we iterate in Gibbs sampling the more precise we get.

4.2 Q2:

We now repeat the same procedure with `n_input=20` and `n_hidden=20`, for this we choose `n_samples=3000`, `step_gd=.01`. We note that the network is less precise as the shifting constraint is at most 34% of the time satisfied. With this we should mention that the training need more tuning to get the CD to converge as with this configuration we only reach:

$$\min_{iter} \|E_{dream}^{(iter)} - E_{awake}^{(iter)}\|_2^2 = \begin{cases} 1.6189,\ CD1 \\ 6.9341,\ CD10 \end{cases}$$

While this convergence criterion is $\ll 1$ in the previous parts.

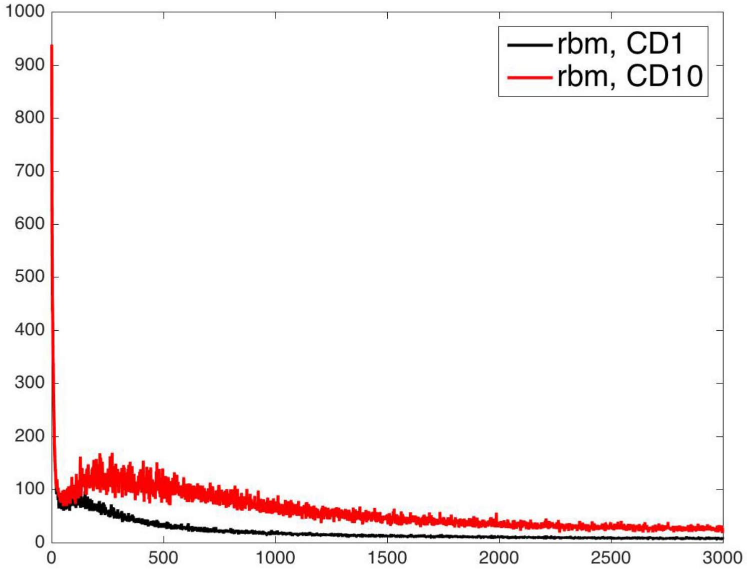

Figure 7: diffs(dream,awake), rbm-CD1 - rbm-CD10

K	**10**	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
CD1	0.17	0.20	0.20	0.26	0.25	0.24	0.28	0.31	0.30	0.33
CD10	0.19	0.25	0.24	0.27	0.28	0.30	0.31	0.31	0.34	0.28

Table 2: Shifted samples (out of 500)

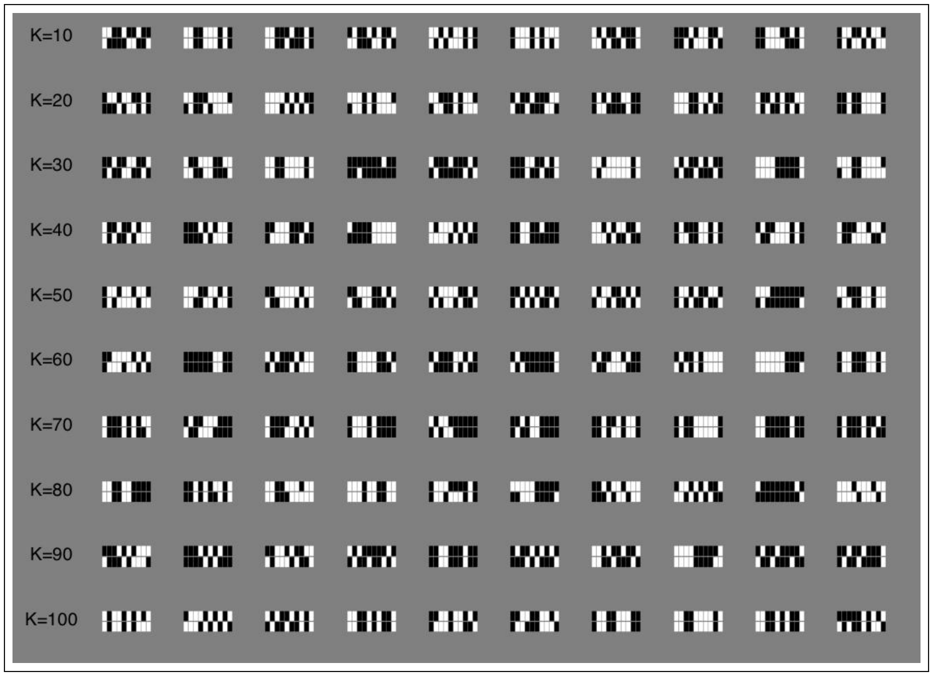

Figure 8: Samples - CD1

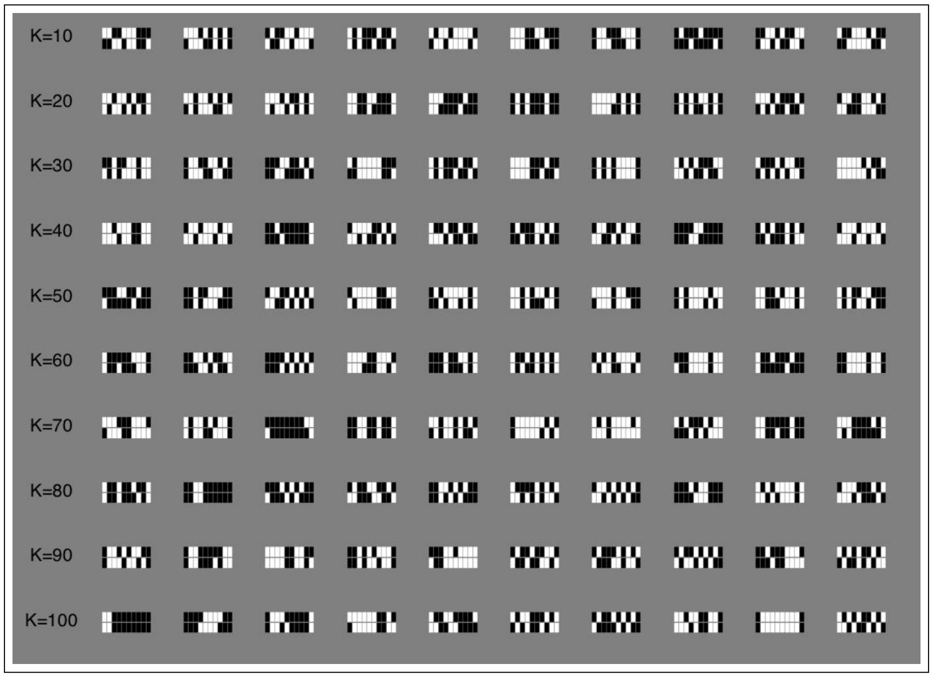

Figure 9: Samples - CD10