0.1. Unsupervised learning

So far it was assumed that the couplings in analysed structures (along with bias terms) were known a priori. However, in general when we analyse some phenomena we don't know this values and we are interested in learning an unknown distribution Q based on some observed data $\mathcal{D}$. The theoretical results suggests that the RBM structure is a natural candidate for approximating underlying distribution from which the data were generated. Thus, the unsupervised learning in this case consists of learning the parameters θ of the approximate distribution P. Therefore, our general goal is to maximize the probability of $\mathcal{D}$ under the MRF distribution i.e. we are looking for the vector of parameters θ that maximize the likelihood given the training data:

$$\max_{\theta} \ln \mathcal{L}(\theta|\mathcal{D}) = \max_{\theta} \ln \prod_{i=1}^{N} p(\mathbf{v}_i|\theta) = \max_{\theta} \sum_{i=1}^{N} \ln p(\mathbf{v}_i|\theta) \tag{1}$$

where N is the size of $\mathcal{D}$.

The experiments on toy models suggest that the initial unsatisfactory results with naive mean field approaches [**tieleman2008training**] might be greatly improved if we include additional terms responsible for connections between the spins.

0.2. Training of Boltzmann Machines

With large graphical models, it is not possible to find an analytical solution to the maximum likelihood estimation of parameters and we need to resort to some approximation methods. That is also the case of the RBM and learning the parameters of this structure relies on the gradient ascent of the log-likelihood. At time t during training, the update of the vector containing all parameters of the RBM θ has the form:

$$\theta^t = \theta^{t-1} + \eta \frac{\partial}{\partial \theta^{t-1}} \ln \mathcal{L}(\theta|\mathcal{D}). \tag{2}$$

This relies on the fact that the gradient w.r.t. parameters θ informs us how fast function increases in the current point θ^{t-1}. By taking appropriately small learning rate, these iterative updates converge to stationary points. With large data set it is common to use a stochastic gradient ascent method [**robbins1951stochastic**] where we sample a minibatch of datapoints and take a noisy gradient estimate which results in the update rule:

$$\theta^{t+1} = \theta^t + \eta \frac{1}{M} \frac{\partial}{\partial \theta^t} \sum_{m=1}^{M} \ln \mathcal{L}(\theta|\mathbf{x}^{(m)}), \tag{3}$$

where M is the size of the minibatch. It can be shown that updates via 3 guarantee to converge to a local optimum under weak conditions [**bottou1998online**].

For a given data point $\mathbf{v}$ the log-likelihood can be seen as the difference between two energies:

$$\mathcal{L} = \ln P(\mathbf{v}) = -\ln(\sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}) - \ln Z = F^c(\mathbf{v}) + F \tag{4}$$

where F is the *free energy* of the RBM and F^c denotes the clamped free energy as we operate on the fixed visible units $\mathbf{v}$. The gradient of the log-likelihood w.r.t θ given a training example $\mathbf{v}$ takes the form:

$$\begin{aligned}
\frac{\partial \log \mathcal{L}(\theta|\mathbf{v})}{\partial \theta} &= \frac{\partial F^c}{\partial \theta} - \frac{\partial F}{\partial \theta} \\
&= -\frac{\sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})} \frac{\partial E(\mathbf{v},\mathbf{h})}{\partial \theta}}{\sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}} + \frac{\sum_{\mathbf{v},\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})} \frac{\partial E(\mathbf{v},\mathbf{h})}{\partial \theta}}{\sum_{\mathbf{v},\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}} \\
&= -\sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{v}) \frac{\partial E(\mathbf{v},\mathbf{h})}{\partial \theta} + \sum_{\mathbf{v},\mathbf{h}} p(\mathbf{v},\mathbf{h}) \frac{\partial E(\mathbf{v},\mathbf{h})}{\partial \theta} \\
&= -\mathbb{E}_{p(\mathbf{h}|\mathbf{v})} \left(\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial \theta} \right) + \mathbb{E}_{p(\mathbf{v},\mathbf{h})} \left(\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial \theta} \right)
\end{aligned} \tag{5}$$

As we can see the gradient is the difference of two expectations – the expected value of the gradient of the energy function under the model distribution and under the conditional distribution of the hidden variables given the observed variables $\mathbf{v}$. Thanks to the restriction imposed on the structure of the Boltzmann machine, the clamped free energy can be computed explicitly. However, as it was mentioned previously, direct calculations of the second term leads to the complexity that is exponential in the number of variables in the model.

0.3. Monte Carlo methods

The second expectation from the gradient in 5 is intractable to compute explicitly in the case of large models and we have to resort to some kind of approximations. Monte Carlo methods rely on stochastic generations of random variables w.r.t. the desired expectation needs to be computed. Denote by:

$$\theta = \mathbb{E}_p(f(X)) = \int f(\mathbf{x})p(\mathbf{x})\mathrm{d}\mathbf{x}$$

the quantity of interest where $X \sim p(\cdot)$. The Monte Carlo estimate has the form:

$$\hat{\theta} = \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}^i)$$

where $\mathbf{x}^i$, $i \in \{1, ..., N\}$ are random samples from X and N is the number of samples. This simple procedure provides unbiased and consistent estimate of θ as $n \to \infty$.

0.3.1. Markov chain Monte Carlo

Monte Carlo method relies on the fact that we are able to generate independent random samples from the distribution of interest. In the case of the RBM, we are not able to generate random samples $\{\mathbf{v}, \mathbf{h}\}$ from the complex joint posterior to approximate the expectation of interest. However, we can use Monte Carlo Markov chain (MCMC) framework to generate approximate samples from the joint distribution $p(\mathbf{v}, \mathbf{h})$.

A discrete stochastic process $X = \{X_t, t \in \mathbb{N}\}$ which takes values in discrete set S is a Markov chain if the Markov property holds, i.e.

$$p_{ij}^t = P(X_t = j | X_{t-1} = i, ..., X_0 = i_0) = P(X_t = j | X_{t-1} = i)$$

for every $t \in \mathbb{N}$ and $i, j, i_0 \in S$. In the case of the discrete process, we usually operate on the transition matrix defined as $\mathbf{P} = (p_{ij})_{i,j \in S}$. The fundamental concept of the theory of the MCMC is stationarity or a stationary distribution π for which it holds $\pi = \mathbf{P}\pi$. MCMC methods focus on constructing an appropriate Markov chain that converges to the desired distribution.

0.3.2. Gibbs sampling

A particular class of MCMC algorithms is the Gibbs sampling algorithm which enables us to produce samples from the joint probability distribution using full conditional distributions. This method is also often called "block-at-a-time" as the transition probabilities are related with subblocks of the vector $\mathbf{x}$. Let $\mathbf{x}$ be divided into two blocks of variables $\mathbf{x}_1$ and $\mathbf{x}_2$. The Gibbs sampler subsequently generates samples from $\mathbf{x}_1^i = p(\mathbf{x}_1|\mathbf{x}_2)$ and $\mathbf{x}_2^i = p(\mathbf{x}_2|\mathbf{x}_1)$ which forms samples from the joint $(\mathbf{x}_1^i, \mathbf{x}_2^i)$ assuming we reached a convergence of the chain.

In the case of the RBM, the structure of the model suggests that we can divide the variables from the joint into two blocks – visible and hidden units. No connections between variables from the same layer enables us efficiently sample from conditionals $p(\mathbf{v}|\mathbf{h})$ and $p(\mathbf{h}|\mathbf{v})$ using **??**.

0.4. Contrastive Divergence

The main challenge related with MCMC methods is the computational burden related with ensuring that the Markov chain has been run sufficiently long to ensure convergence to a stationary distribution. However, it was proven empirically that the chain might be run only a few steps in order to train an effective model [**hinton2002training**] which is called contrastive divergence (CD) learning.

There are two steps which differ CD from the naive MCMC sampling to approximating the second expectation from the gradient 5. Firstly, instead of running the Markov chain until it obtains a stationary distribution, the chain is initialized using training data point $\mathbf{v}^0$ from the training data set. Secondly, the Gibbs chain is run only for k steps (CD-k) where k is usually smaller than 20. Figure 1 presents the procedure for the CD-1:

The approximation to the gradient by the single data point $\mathbf{v}^0$ in the case of CD-k takes the form:

$$-\sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{v}^0)\frac{\partial E(\mathbf{v}^0, \mathbf{h})}{\partial \theta} + \sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{v}^k)\frac{\partial E(\mathbf{v}^k, \mathbf{h})}{\partial \theta} \tag{6}$$

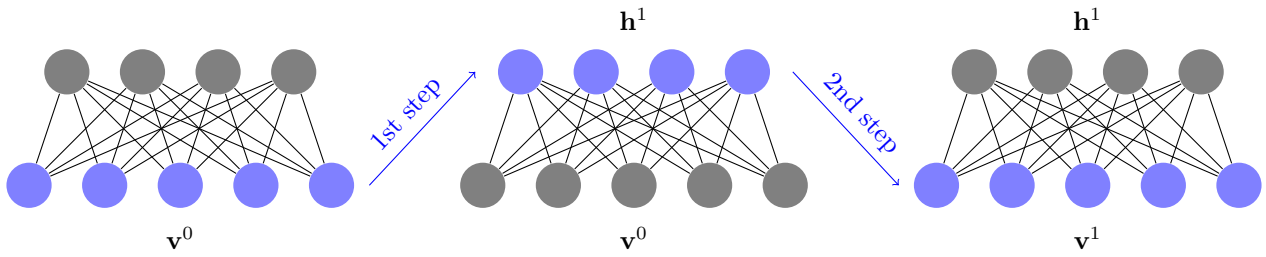

Figure 1: The first step of the Gibbs sampler for the RBM for a particular data point $\mathbf{v}^0 \in \mathcal{D}$.

It should be noted here that as we run the Gibbs chain only a few (k) steps, the samples $\{\mathbf{v}^k, \mathbf{h}^k\}$ don't come from the stationary distribution and the approximation 6 is biased as it doesn't maximize the likelihood of the data but the difference of two KL-divergences [**hinton2002training**], [**fischer2012introduction**]:

$$KL(Q|P) - KL(P_k|P)$$

where Q is the empirical distribution and P_k is the distribution after k step of the Gibbs chain and this explains the name of the algorithm.

0.4.1. Persistent contrastive divergence

It was observed that the contrastive divergence procedure still requires many steps to be run in order to learn a good generative model. The rate of learning might be significantly improved when we don't reinitialize the Markov chains with a new training batch in order to obtain a sample $\{\mathbf{v}_i^k\}_{i=1}^N$ where N is the size of the batch but rather keep "persistent" chains (PCD) [**tieleman2008training**]. Thus, the starting state for the Gibbs chain is equal to the last step from the previous update. The assumption made here is that between parameter updates, the model changes only slightly in terms of parameters' values[**neal1992connectionist**]. Thus, the initialization from the last state of the Gibbs chain taken from the previous model should be closer to the model distribution. The empirical results suggest to keep one persistent chain per one training data point in a batch.

0.5. Learning using extended mean field approximation

The stochastic procedure described in the previous section can be exchanged with the fully deterministic approach as the log-likelihood in the case of the EMF approximation has the form:

$$\mathcal{L} = \ln P(\mathbf{v}) = F^c(\mathbf{v}) - F^{EMF}. \tag{7}$$

As the first term from 4 can be computed explicitly, it is independent from the approach taken during training and we only have to derive the updates using the EMF approximation of the free energy.

Let's now fix visible and hidden magnetizations $\{\mathbf{m}^v, \mathbf{m}^h\}$. The gradient of the log-likelihood w.r.t a coupling parameter W_{ij} up to the third-order term is:

$$\begin{aligned}\frac{\partial F^{EMF}}{\partial W_{ij}} =& - m_i^v m_j^h - W_{ij}^t (m_i^v - (m_i^v)^2)(m_j^h - (m_j^h)^2) \\ & - 2W_{ij}^2 (m_i^v - (m_i^v)^2)(\frac{1}{2} - m_i^v)(m_j^h - (m_j^h)^2)(\frac{1}{2} - m_j^h),\end{aligned}$$

while the updates for the bias terms are just negative of the fixed-point magnetizations:

$$\begin{aligned}\frac{\partial F^{EMF}}{\partial a_i} &= -m_i^v, \\ \frac{\partial F^{EMF}}{\partial b_j} &= -m_j^h.\end{aligned} \tag{8}$$

Thus, the training procedure using a deterministic approach goes as follows: given a data point $\mathbf{v}$ we obtain expected values of the hidden units $\mathbf{h} = \text{sigm}(W\mathbf{v}+\mathbf{b})$ which are starting points for magnetizations, i.e. $\mathbf{m}_0^v = \mathbf{v}$ and $\mathbf{m}_0^h = \mathbf{h}$. Then, we perform an iterative algorithm (which can have the form as presented in the previous chapter) until convergence to obtain magnetizations $\{\mathbf{m}^v, \mathbf{m}^h\}$ that satisfy self-consistency relations. Those magnetizations can then be used to obtain gradient w.r.t the parameters of the model and to compute the approximation of the free energy.

0.6. Approximating the log-likelihood

The problems related with intractability of the partition function makes training such structure very difficult as we cannot observe directly progress of learning. Thus, we need to resort to some approximations. One of the most popular approaches to measure progress in training RBMs is due to Besag [**besag1972nearest**] – consider the following approximation of n-dimensional distribution

$$P(\mathbf{x};\theta) = \prod_i p(x_i|x_1,...,x_{i-1};\theta) \approx \prod_i p(x_i|x_1,...,x_{i-1},x_{i+1},...,x_n;\theta) = \prod_i p(x_i|x_{-i};\theta) \coloneqq PL(\mathbf{x};\theta) \quad (9)$$

where the first equation comes from the chain rule and x_{-i} denotes the set of all variables except variable x_i. We assume here that marginals given all other are independent of each other. The likelihood has then the form:

$$\ln PL(\mathbf{x};\theta) = \sum_i \ln P(x_i|x_{i-1};\theta). \quad (10)$$

If the analysed phenomena has many dimensions this approximation is still computationally expensive. Thus, another step is to choose only one marginal as a proxy, i.e.

$$\ln PL(\mathbf{x};\theta) = n \ln P(x_i|\mathbf{x}_{-i};\theta), \quad (11)$$

where i is randomly chosen from $\{1,2,...,n\}$. It can be shown that this pseudo-likelihood is maximized by the true parameters of the model. In the case of the RBM, this estimator takes especially efficient form:

$$\ln PL(\mathbf{x};\theta) \approx n \log\left(\frac{\exp\{-F^c(\mathbf{x})\}}{\exp\{-F^c(\hat{\mathbf{x}})\} + \exp\{-F^c(\mathbf{x})\}}\right) = n \ln\left(\mathrm{sigm}(F^c(\hat{\mathbf{x}}) - F^c(\mathbf{x})\right) \quad (12)$$

where $\hat{\mathbf{x}}$ represents the vector $\mathbf{x}$ with i-th variable flipped, i.e. $1 - x_i$.

0.7. Real scale model – MNIST data set

The data set that will be used for the comparison and the evaluation of EMF and CD training algorithms is the MNIST set [**lecun1998**] which is a well-known benchmark image classification dataset that consists of 60000 training and 10000 testing images of digit numbers. They are represented on 28-by-28 grey-scale grid of pixels. Thus, the first visible layers in all analysed models consists of 784 visible units. Following [**gabrie2015training**], [**salakhutdinov2008learning**] all images were rescaled to $\{0,1\}$ and binarized by setting all non-zero pixels to 1 in all experiments. The data set was divided into 600 mini-batches which results in 100 training points per batch.

0.8. Comparison of both approaches

In order to test the efficiency of the EMF learning algorithm, I used three expansions of **??** – up to the first-order (MF), second-order (TAP2) and third order (TAP3) term. Moreover, I varied the number of iterations of self-consistency relations (3 and 10) using asynchronous updates of the form **??** to mimic the idea from the contrastive divergence approach. As a benchmark, two models were trained following the stochastic training (CD1, CD10).

Furthermore, all models described above were trained using persistent approach (PMF, PTAP2, PTAP3, PCD). In the case of the EMF approximation, the magnetizations of a batch from the previous update are the starting points in the next update [**gabrie2015training**]. Similarly to PCD, this idea is based on the fact that between updates the model changes only slightly and it should improve the convergence to the new fixed point magnetizations.

All models were trained 10 times using the same set-up of free parameters with 500 units. The purpose of this experiment is to compare different RBM trainings thus following [**gabrie2015training**] I didn't use the adaptive learning rate which was set to 0.005, learning was performed using mini-batch updates with 100 training points per batch. The couplings matrix was randomly initialised using normal distribution with zero mean and variance set to 0.01. This allows to compare the procedures in the their "raw" forms.

However, the EMF approximation was performed around the infinite temperature were the spins are independent. Thus, in general couplings should have small values – this can be enforced using regularization which at the same times allows for a better generalization. From probabilistic perspective this can be seen as adding a

weighted prior over the parameters (maximum a posteriori training). The criterion that will be maximized has now the form:

$$E(\theta, \mathcal{D}) = \ln \mathcal{L}(\theta|\mathcal{D}) - \lambda R(\theta) \tag{13}$$

where $R(\cdot)$ is the regularizer and $\lambda \in \mathbb{R}_+$ is a hyper-parameter which controls the effective power of the regularization. In all experiments Laplacian prior $R(\theta) = \|\theta\|_1$ ($L1$ regularization) was used with λ set to 0.01.

Figure 2 presents the pseudo log-likelihood 12 (left) and EMF log-likelihood 7 for the non-persistent training procedure. Firstly, by the visual inspection both approximation yield very similar results for each analysed model. However, the EMF estimates are much less noisy at a lower computational cost.

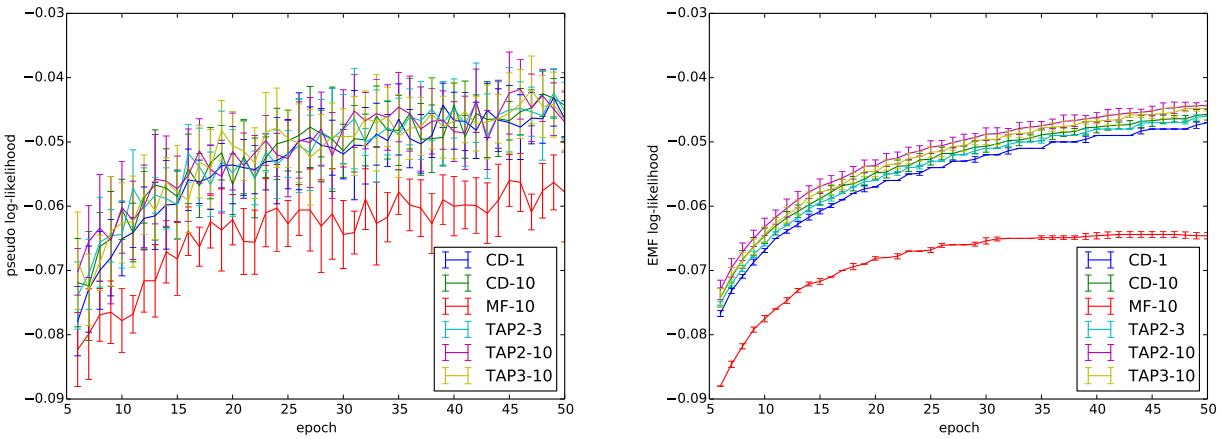

Figure 2: Per-sample pseudo log-likelihood (left) and EMF log-likelihood (right) on the validation set of the MNIST data set divided by number of all units in the model (1284) across first training 50 epochs for RBMs models trained stochastically and deterministically. Error bars shows the standard deviations of 10 trained models using a particular version of training.

Secondly, results for the MF-10[1] confirms the findings from the literature – the naive mean field approach is not able to learn an effective model. Moreover, the results for the CD, TAP2 and TAP3 are very similar. There are not significant differences between models with 3 or 10 iterations of self-consistency relations which shows that the deterministic approach is not computationally expensive.

As it was expected, the best results in terms of the EMF log-likelihood are achieved by EMF methods. However, the results for the CD models suggest that the EMF log-likelihood may be used as a reliable indicator of progress during training as those models weren't constructed to optimize over this objective [**gabrie2015training**].

Figure 3 presents the results for persistent versions of models analysed above. There are not significant differences comparing to However, as it was expected the samples from the models trained using persistent chains are of much higher quality.

Finally, in persistent and non-persistent versions of models the addition of the third order term from the EMF expansion **??** doesn't provide improvement over the TAP model. This might be partially explained by the fact that estimated weights are in general smaller than 1 (in absolute value) which are then used at the order of 3 in self-consistency equations and hence don't affect significantly estimations.

[1]The results for MF-3 weren't included as it was very similar to the MF-10

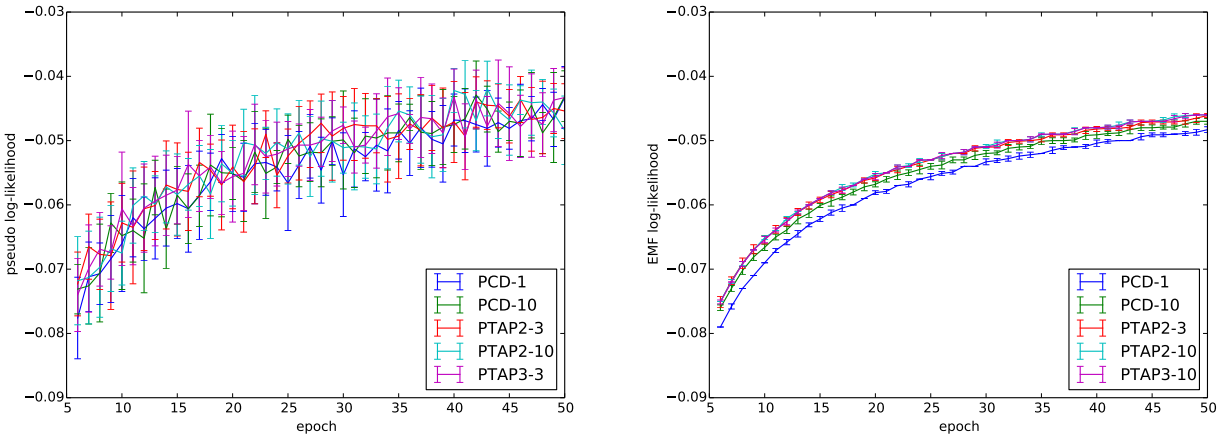

Figure 3: Per-sample pseudo log-likelihood (left) and EMF log-likelihood (right) for the same models trained using persistent Gibbs chains. Results of the naive method weren't included.