0.1. Updates

In the chapter 3 different schedules of updates were analysed on the toy model where the parameters of the model were known a priori and no substantial discrepancies were observed in terms of the quality of approximation. However, in the case of the `MNIST` data set estimated magnetizations allow us to perform learning of unknown parameters. Thus, in this case we combine uncertainty related to both magnetizations and parameters – this may lead to substantial differences in performance. Figure

Figure 1: updatesTAP2

Figure 2: updatesMF

Figure 3: updatesTAP3

0.2. Generated samples from the models

0.3. AIS vs TAP

Make several runs of AIS to get reasonable approximation! For all models we also used 1,000 spaced uniformly from 0 to 0.5, 4,000 spaced uniformly from 0.5 to 0.9, and 5,000 spaced uniformly from 0.9 to 1.0, with a total of 10,000 inter mediate distributions.

0.3.1. Annealed Importance Sampling (AIS)

The most widely used technique is based on a very simple identity. Assume we have two distributions $p_A = \frac{1}{Z_A}p^*_A(\mathbf{x})$, $p_B = \frac{1}{Z_B}p^*_B(\mathbf{x})$ where $p^*(\dot{})$ denotes unnormalized distribution and Z_A, Z_B are partition functions. Assuming that a proposal p_A distribution p_A supports tractable sampling and tractable evaluation of both the unnormalized distribution $p^*_A(\mathbf{x})$ and the partition function Z_A we can use the following relation:

$$\begin{aligned} Z_B &= \int p^*_B(\mathbf{x})\mathrm{d}\mathbf{x} \\ &= \int \frac{p_A(\mathbf{x})}{p_A(\mathbf{x})} p^*_B(\mathbf{x})\mathrm{d}\mathbf{x} \\ &= Z_A \int \frac{p^*_B(\mathbf{x})}{p^*_A(\mathbf{x})} p_A(\mathbf{x})\mathrm{d}\mathbf{x} \end{aligned} \tag{1}$$

Sampling from the tractable distribution, we can derive Monte Carlo estimator of the ratio between partition functions:

$$\frac{Z_B}{Z_A} \approx \frac{1}{N}\sum_{i=1}^{N} \frac{p^*_B(\mathbf{x}^{(i)})}{p^*_A(\mathbf{x}^{(i)})} = \hat{r}_{SIS} \tag{2}$$

where $\mathbf{x}^{(i)}$ comes from p_A. Assuming that distribution p_A is close to p_B, the estimator from 2 called simple importance sampling proves to work well [**minka2005divergence**]. However, in high-dimensional spaces where p_B is usually multimodal as it is considered in this thesis, the variance of the estimator from 2 might be very high.

The idea presented above might be improved by following the classic approach from probabilistic optimization i.e. simulated annealing. The idea is to introduce intermediate distributions that will allow to bridge the gap between two considered distributions p_A and p_B [**jarzynski1997nonequilibrium**], [**neal2001annealed**].

Consider a sequence of distributions $p_0, p_1, ..., p_M$ where $p_0 = p_A$ and $p_M = p_B$. If the intermediate distributions p_m and p_{m+1} are close enough, a simple estimator from 2 can be used to estimate each ratio $\frac{Z_{m+1}}{Z_m}$. Using the the following identity:

$$\frac{Z_M}{Z_0} = \frac{Z_1}{Z_0}\frac{Z_2}{Z_1}...\frac{Z_M}{Z_{M-1}} \tag{3}$$

those intermediate ratios are then combined to obtain the estimate of $\frac{Z_B}{Z_A}$. There is no need to compute the normalizing constants of any intermediate distributions. The intermediate distributions are chosen to suit a given problem domain. However in most cases, we are able to draw exact samples only from the first tractable distribution p_A. In order to sample from intermediate distribution we have be able to draw a sample $\mathbf{x}^{'}$ given $\mathbf{x}$ using Markov chain transition operator $T_m(\mathbf{x}^{'}|\mathbf{x})$ that leaves $p_m(\mathbf{x})$ invariant, i.e.:

$$\int T_m(\mathbf{x}^{'}|\mathbf{x})p_m(\mathbf{x})\mathrm{d}\mathbf{x} = p_m(\mathbf{x}^{'}) \tag{4}$$

These transition operators represent the probability density of transitioning from state $\mathbf{x}$ to $\mathbf{x}^{'}$ [**salakhutdinov2008learning**]. Having obtained the sequence of samples from the intermediate distributions we can obtain the improved estimator of the ratio between partition functions following the procedure:

Algorithm 1 Annealed Importance Sampling.

```
Set $p_A$ and $p_B$ with appropriate parameters
for $i \in \{1, ..., N\}$ do
    sample $\mathbf{x}_1$ from $p_0 = p_A$
    sample $\mathbf{x}_2$ via $T_1(\mathbf{x}_2|\mathbf{x}_1)$
    ...
    sample $\mathbf{x}_M$ via $T_M(\mathbf{x}_M|\mathbf{x}_{M-1})$
    $r^{(i)}_{AIS} = \frac{p^*_1(\mathbf{x}_1)}{p^*_0(\mathbf{x}_1)}\frac{p^*_2(\mathbf{x}_2)}{p^*_1(\mathbf{x}_2)}...\frac{p^*_M(\mathbf{x}_M)}{p^*_{M-1}(\mathbf{x}_M)}$
end for
$\hat{r}_{AIS} = \frac{1}{N}\sum_{i=1}^{N}\hat{r}^{(i)}_{AIS}$
```

It was proven that the variance of $\hat{r}_{AIS}$ will be proportional to $1/MN$ assuming we used sufficiently large numbers of intermediate distributions M [**neal2001annealed**]. Moreover, the estimate of Z_M/Z_0 will be unbiased if each ratio is estimated using $N = 1$ and a sample $\mathbf{x}^m$ is obtained using Markov chain starting at previous sample. This follows from the observation that the AIS procedure is an simple importance sampling defined on an extended state space $\mathcal{X} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_M)$.

The procedure described above can be adapted to the RBM case – assume that we have estimated parameters θ_B of the model that we want to evaluate. Following [**salakhutdinov2008quantitative**] as a tractable starting distribution p_A we can use "clamped" restricted Boltzmann machine where there is no hidden layer. The sequence of intermediate distribution is then defined as:

$$p_m(\mathbf{v}) = \frac{1}{Z_m}p^*_m(\mathbf{v}) = \frac{1}{Z_m}\sum_{\mathbf{h}}\exp(-E_m(\mathbf{v}, \mathbf{h})) \tag{5}$$

where $m = 0, ..., M$, $\mathbf{h} = \mathbf{h_B}$, and the energy function has the form:

$$E_m(\mathbf{v}, \mathbf{h}) = (1 - \beta_m)E(\mathbf{v}; \theta_A) + \beta_m E(\mathbf{v}, \mathbf{h}; \theta_B) \tag{6}$$

where $\beta_m \in [0, 1]$ with $\beta_m = 0$ yielding p_A and $\beta_m = 1$ giving p_B. Annealing slowly the "temperature" from infinity to zero we gradually moves from the state space of proposal distribution to the space defined by the untractable distribution. Following the approach from **??** we can obtain transition operators for hidden and visible variables:

$$\begin{aligned} p(h^A|\mathbf{v}) &= \sigma((1-\beta)NONE \\ p(h^B) &= \end{aligned} \tag{7}$$

Algorithm 2 Appendix – Implementation for BM and RBM.

```
$\theta, \phi \leftarrow$ .
while not converged in $\theta, \phi$ do
    Pick subset of size $\mathbf{x}_{1:M}$ from the full dataset uniformly at random.
    Compute $g \leftarrow \nabla_{\theta,\phi}\tilde{\mathcal{L}}^B\theta, \phi; \mathbf{x}_{1:M}, \epsilon_{1:M}$
end while
```

Figure 4: AIS for TAP2 and TAP3

0.3.2. Comparison

Two models were estimated with 5 of magnetizations to compare the quality of the approximation of the variational free energy. Figure 0.3.2 presents the estimates of **??** for the `TAP2` and `TAP3` models along with AIS estimates for different numbers of runs.

TAP2, TAP3 3 or 10 steps of magnetization!

0.4. Deep RBM

0.4.1. Unsupervised Pre-training of Neural Networks

add Erham here

renormalization group erham In the previous chapter it was argued that the unsupervised pre-training . science Deep learning methods

0.4.2. Deep belief nets

Following the approach from The breakthrough to effective training strategies for deep architectures came in 2006 with the CD algorithm for training Deep Belief networks (DBN) [**hinton2006reducing**]. DBNs are generative graphical models with many hidden layers of hidden causal variables which joint distribution has the following form:

$$p(\mathbf{x}, \mathbf{h}^1, \mathbf{h}^2, ..., \mathbf{h}^l) = p(\mathbf{x}|\mathbf{h}^1)P(\mathbf{h}^1|\mathbf{h}^2)...P(\mathbf{h}^{l-2}|\mathbf{h}^{l-1})P(\mathbf{h}^{l-1}|\mathbf{h}^l). \tag{8}$$

It was shown that adding an extra layers always improve a lower bound **??** on the training data if the number of feature detectors per layer is sufficiently large and the weights are initialized correctly. It was empirically proven that Figure **??** depicts the exemplary deep belief network. DBNs can be formed using a greedy layer-wise unsupervised training of stacked RBMs – algorithm 3 presents how the process folows: http://www.yann-ollivier.org/rech/publs/deeptrain.pdf - show the picture This simple and intuitive algorithm proved to be an effective way of pretraining deep structures which laid the foundations of the resurgence of deep neural networks. Originally, the building blocks are trained following constrastive divergence procedure. However, the positive results obtained using extended mean-field approximation suggests that we may follow this procedure

Algorithm 3 Learning Deep Belief Nets.

```
Train the first layer as an RBM, learning $P(\mathbf{x} = \mathbf{h^0}, \mathbf{h^1}$
for $l \in \{2, ..., L\}$ do
    Pass the mean activities $\mathbf{x}^l = P(\mathbf{h^l}|\mathbf{h^{l-1}}$ which become a representation of the input at the layer $l$.
    Train the $l$-th layer treating it as an RBM with $\mathbf{x}^l$ as an input.
end for
```

[**montufar2010refinements**]

Theorem 1 (Guido-Ay, 2010) *Let $n = \frac{2^b}{2} + b$, $b \in \mathbb{N}$, $b \geqslant 1$. A DBN containing $\frac{2^n}{2(n-b)}$ hidden layers of size n is a universal approximator of distributions on $\{0, 1\}^n$.*

The guarantee that we improve the bound is no longer valid if the size of subsequent hidden layers is not large enough however it was empiracally proven that such approach still can learn an effective generative model. After pretraining multiple layers of feature detectors, the model can be "unfolded" to form an autoencoder structures where the decoder network uses transposed weigths of the encoder network. At this stage, such network might be considered as feed forward deep neural architecture and might be used as a starting point for supervised fine-tuning with respect to any training criterion that depends on the learnt representation **??**.

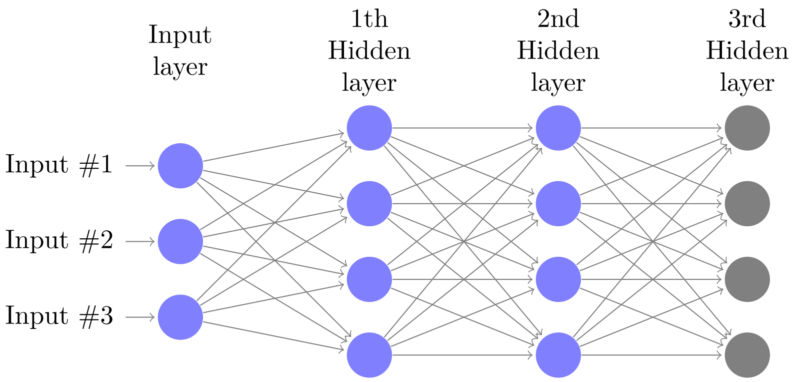

Figure 5: dsd

0.4.3. Reconstructions analysis

Figure **??** presents the reconstructions of randomly chosen samples from the validation data set produced by deep autoencoders trained with four different methods of pre-training DBNs. The autoencoder consists of three hidden layers of sizes 500, 250 and 25 accordingly. The reconstructions created by `PTAP2` and `PTAP3` are of similar quality as `PCD`.

TODO - learn model long and maybe on samples to get 2 dimensions.

Figure 6: DRBM TAP2 TAP3 PCD

The average squared errors on training and validation data sets (Figure **??**) confirms the visual assessment of reconstructions. The mean field obtains the highest score while TAP2 and TAP3's scores are slightly higher then with training DBN using PCD approach.

Figure 7: shows the mean squared error on training and validation data sets for different methods for pre-training the RBMs.

TAP2 0.0515471685931 MF 0.0610472665952 TAP3 0.0511828595047 PCD 0.048851695881

Those results confirms the observations from the previous chapter and shows that additional higher-order approximations substantially improves the quality of learned magnetizations which in turns helps learning a better model.