Programming Project: Restricted Boltzmann Machine

Christoph Kirchner
Mario Rose
Nga Pham

21/2/2018

1 Introduction

For this assignment we implemented a generic RBM over binary random variables, trained and tested MNIST example with different settings in order to achieve better result.
Please note, that in the following results, if not stated otherwise, we used following default settings:

- Maximum epochs: 1.000
- Sample data size of 10.000
- Learning rate: 0.1
- Number of random instances: 1.000
- Number of repetition in Gibbs sampling techniques: 5
- Number of samples in Markov chains: 10
- Number of visible units: 784 (one for every pixel in the training images)
- Number of hidden units: 1000

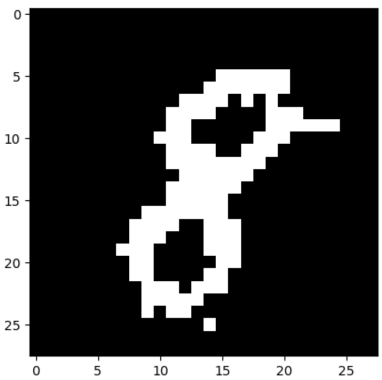
2 Implement a generic RBM

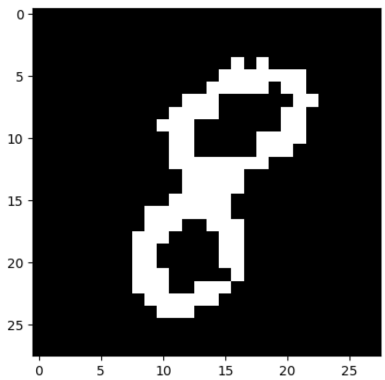
To implement our generic RBM over binary random variables, we followed the theory of the lecture slides:

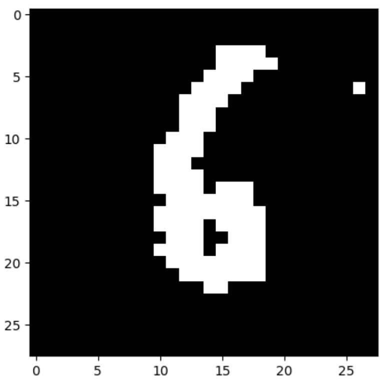
1. Initial class RBM with weights for visible, hidden and bias unit
2. Create training algorithm. We used Persistent Contrastive Divergence (PCD)
3. Handle data: We used the original test set for both training and test. The code snippet was from the assignment description.

3 Train the machine on MNIST data and draw random images

Best results with GibbsSampling, 20000 epochs, 1000 hidden units and Chain length of 20:
(The Random Number Generator took 2 times 8)



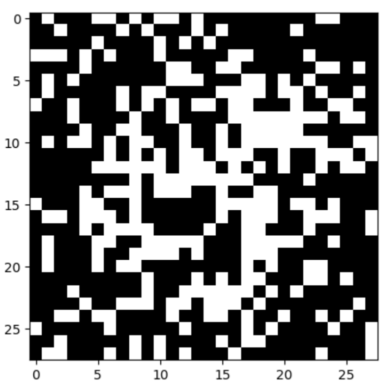




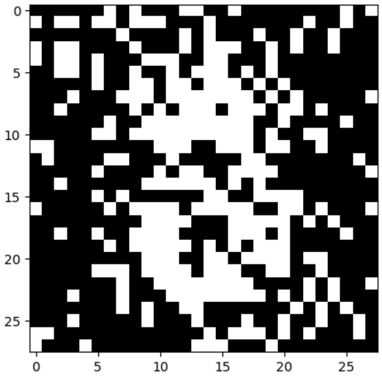
4 Explore the number of hidden units

Gibbs Sampling, 5000 epochs, Chain length 20

500 hidden units:



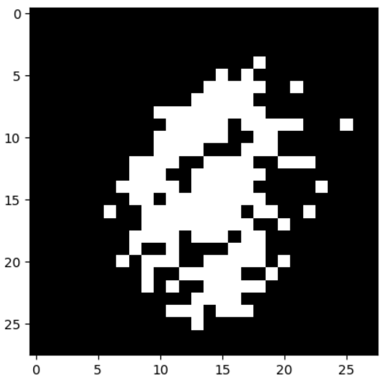
784 hidden units:

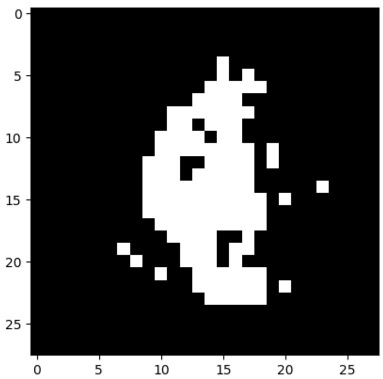


5 Explore different number of Markov chains

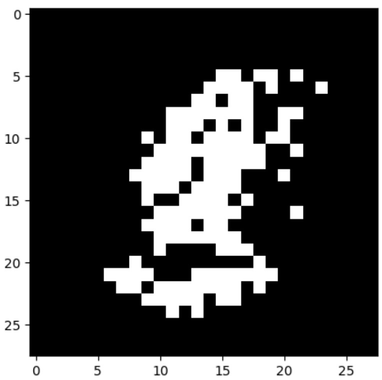
GibbsSampling, 20000 epochs, 1000 hidden units

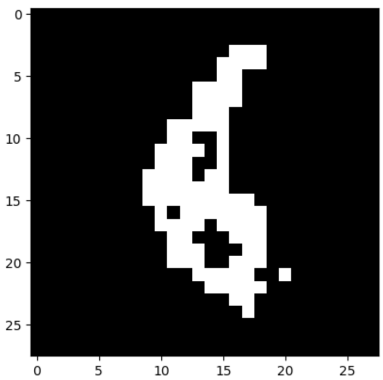
Chain Length of 5:



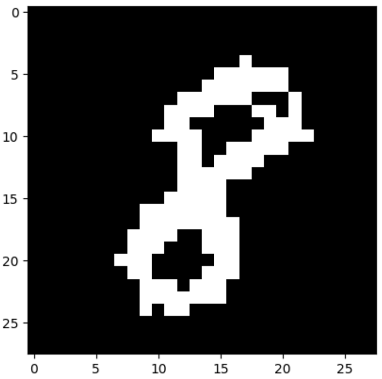


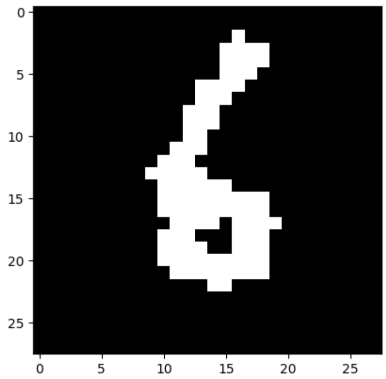
Chain Length of 10:



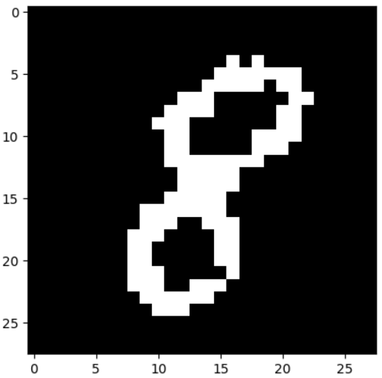


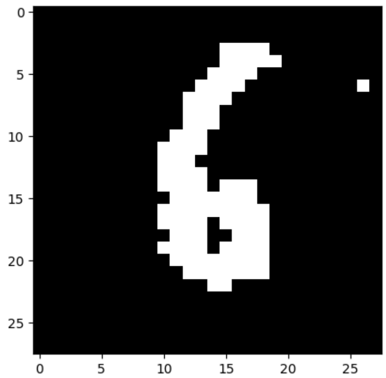
Chain Length of 15:





Chain Length of 20:

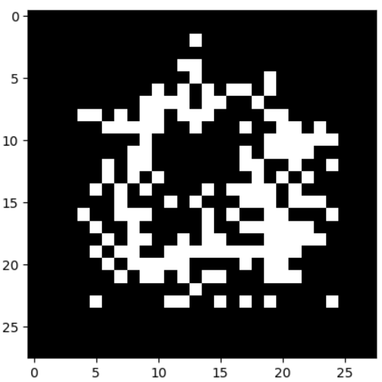




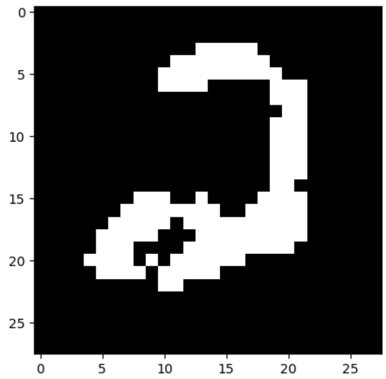
These results show that the higher the number of Markov Chains, the better the result.
But the results with a length of 15 look almost as good as the results with 20, which leads to the conclusion that there will probably be a point where increasing the number of Markov Chains does not increase the quality of the generated image any more.

6 Gibbs Sampling

Without Gibbs Sampling, 30000 epochs:



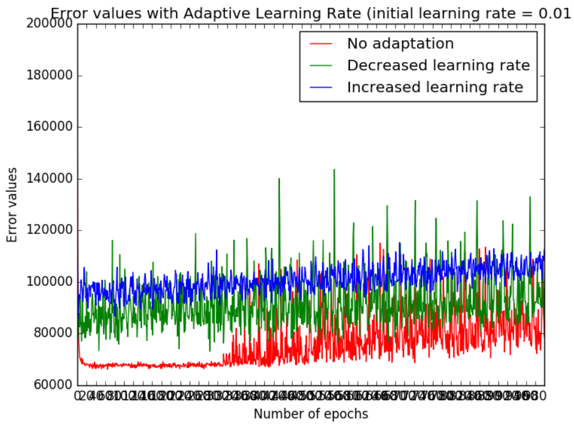
With Gibbs Sampling, 20000 epochs:



7 Explore Adaptive Learning Rate

We observed if adjustment in learning rate impacts the performance of the trained RBM. First, we compared different fixed learning rate: 0.01, 0.1 and 0.5. The result is shown in figure below.
As can be seen from the figure, the learning rate with value 0.01 gave the best result. Second, we tested the case in which the learning rate was increased and decreased gradually by multiplying the original with (1 + ϵ) and (1 - ϵ), respectively. ϵ is a constant, in this case we used learning rate = 0.01 and ϵ = 0.001.

```
if (learning_rate_decay):
    learning_rate = (1 - epsilon) * learning_rate
```

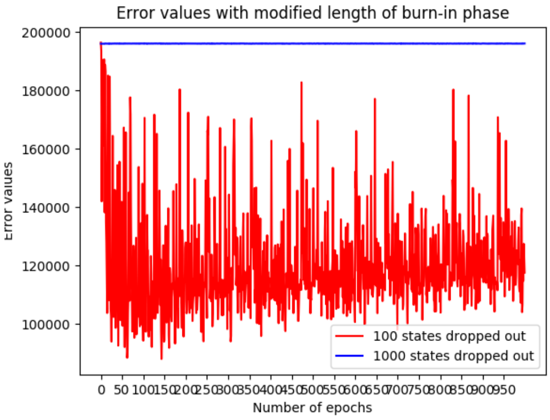


As stated in the above figure, best result can be achieved with original learning rate, then the decreased learning rate. However, they lead to more fluctuate error values curve than the case of increased learning rate.
Reference: G. Hinton, A Practical Guide to Training Restricted Boltzmann Machines 2 . Neural Networks: Tricks of the Trade, 2nd edition, pp. 599619, 2012

8 Explore the length of the burn-in phase in Gibbs sampling

We explored the impact of burn-in phase by drop out the first few states of the Markov chain. The cases when first 100 and 1000 states are thrown away are tested.

```
if (burn_in_drop):
    for j in range(1, num_drop):
        hidden_states[:j] = 0  # Start dropping out first 100 states in Gibbs sampling
```



The plot below compared two result:
It is obviously that with 100 states dropped out, the result was much better, however, produced significantly fluctuate curve. Error values in case of 1000 states is stable, around 200.000 for each epochs.