

## Introduction

### Boltzmann Machines

A Boltzmann Machine consists of a set of binary units,  $s$  and all these units are connected to each other with a weight,  $w_{ij}$  associated with each connection. The global energy,  $E$  of the Boltzmann Machine is given by:

$$E = - \left( \sum_{i < j} w_{ij} s_i s_j + \sum_i \theta_i s_i \right)$$

While learning from data, we try to adjust the weight parameters such that the stationary distribution of the Boltzmann machine closely represents our dataset. For determining the stationary distribution of the Boltzmann Machine we start with random states for all the binary units and iterate over them setting them to +1 with a probability of  $\frac{1}{1+e^{-2a_i}}$ , -1 otherwise. After a few iterations the machine converges to its stationary distribution.

Now for comparing this stationary distribution to our dataset we generate some samples from the Boltzmann Machine and using that we compute the free expectations using:

$$\langle s_i \rangle = \sum_s s_i p(s), \quad \langle s_i s_j \rangle = \sum_s s_i s_j p(s)$$

Similarly we compute clamped expectations from our dataset using:

$$\langle s_i \rangle_c = \frac{1}{P} \sum_{\mu} s_i^{\mu}, \quad \langle s_i s_j \rangle_c = \frac{1}{P} \sum_{\mu} s_i^{\mu} s_j^{\mu}$$

and then we update our weights based on these values:

$$\begin{aligned} w_{ij}(t+1) &= w_{ij}(t) + \eta \frac{\partial L}{\partial w_{ij}} \\ \theta_i(t+1) &= \theta_i(t) + \eta \frac{\partial L}{\partial \theta_i} \\ \frac{\partial L}{\partial \theta_i} &= \langle s_i \rangle_c - \langle s_i \rangle \\ \frac{\partial L}{\partial w_{ij}} &= \langle s_i s_j \rangle_c - \langle s_i s_j \rangle \end{aligned} \tag{1}$$

### Mean Field Theory

### Research Questions

1. Adding noise to the MNIST Dataset
2. Sampling vs Mean Field

### 3. Burn-in period of the sampling algorithm

## Results

### Accuracy of the model with varying levels of noise

Each image in the MNIST dataset is represented by a 28x28 pixel. We first binarized it and added noise so that the matrix  $C$  doesn't get singular.

For adding the noise to the image we created a noise mask for each pixel. Each pixel in the mask has a value of  $-1$  with probability  $p$  and value  $1$  with probability  $1-p$ . Then we do an element-wise multiplication of the binary image and the noise mask. Therefore when  $p = 0$ , there shouldn't be any distortion in the image whereas for  $p = 1$  every bit should be flipped but there won't be any change in the structure. For  $p = 0.5$ , the maximum noise will be obtained and the image will be completely distorted. We can see these effects in Fig. 1.

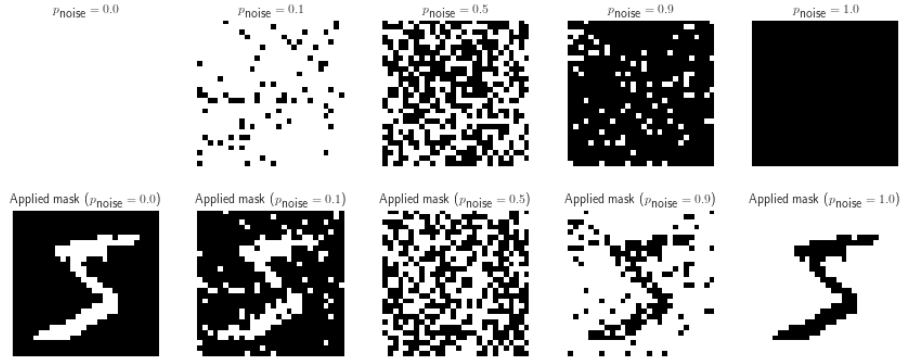


Figure 1: Different noise levels along with the final masked images

We tried to run our Boltzmann machine on varying levels of noise and computed the accuracy of our model.

There are 3 main features of the graph:

#### 1. Low accuracy at low noise levels

In Fig. 2, we can see that for very low levels of noise (around 0.00 - 0.03) we have an accuracy of around 0.1 which is equivalent to random guessing. This low accuracy can be explained by the fact that due to low noise levels some of the pixels in the all the images are the same. For example the top-left pixel or the top-right pixel will be off in almost all the images. And these constant pixels makes the matrix  $C$  approach towards singularity resulting in low accuracy.

#### 2. Highest accuracy at around noise level of 0.1

The accuracy of the model starts increasing at around  $p = 0.04$ , gets

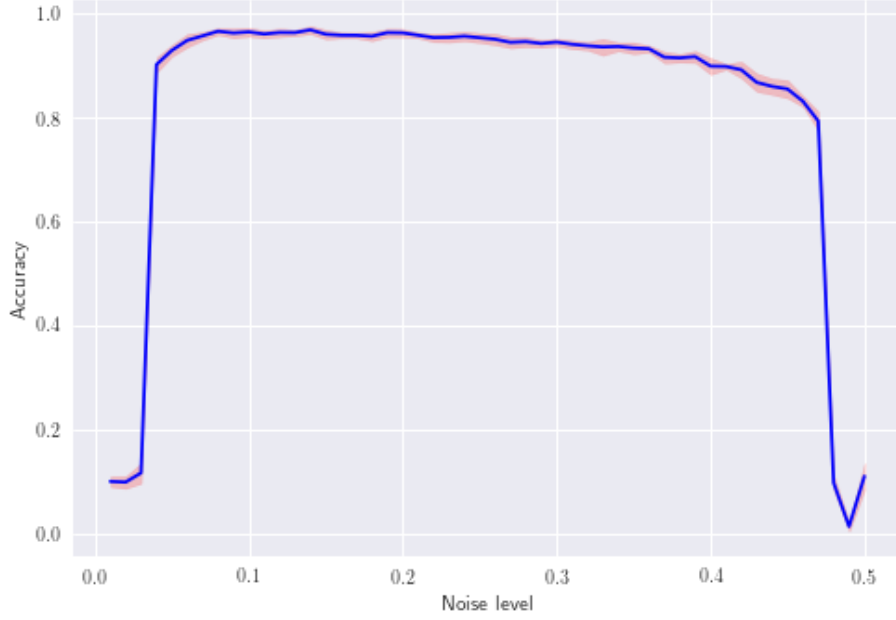


Figure 2: Accuracy of the model with varying noise level

around 0.9 accuracy for  $p \approx 0.5$  reaching the maximum at  $p \approx 0.1$  and the gradually decrease as we increase the noise level in the data.

3. **Dip at  $p = 0.47$  and accuracy of 0.1 at  $p = 0.5$**   
 We see a dip in the accuracy value at  $p = 0.47$  where it is almost 0. We couldn't find any good possible explanation for this behaviour. For  $p = 0.5$  we again get an accuracy of 0.1. This behaviour can be explained since our images are completely distorted now, the model's performance is equivalent to random guessing.

### Comparison of Sampling Methods and Mean Field Approximation

Since the total possible number of states in a Boltzmann Machine is  $2^n$ , therefore it's not feasible to use exact method for computing the free statistics. Therefore, we used sampling methods and Mean Field and compared the results.

Variation with burn in period

Conclusion

Appendix