Sushanth Raman

sr3355

Understanding the theoretical underpinnings behind Deep Networks

Introduction:

This paper documents the results of a project trying to understand the mathematical properties behind deep neural networks through the Ising Model. The focus of the project was to gain a base level understanding on what makes DNNs (deep neural networks) so effective at uncovering the underlying features in structured data. A potential approach for this is to view DNNs as an iterative coarse graining process. In other words with each layer, DNNs learn how to successfully ignore irrelevant features and place emphasis on relevant ones.

This iterative process shares many similarities with an important tool in theoretical physics known as the renormalization group, which help deal with problems that involve multiple scales of length. In our project, we specifically look at the two-dimensional Ising Model just above the curie temperature. At the curie temperature, the correlation length of the model reaches infinite. In other words, at this temperature any two spins within the model will be correlated regardless of their distance. This makes it notably difficult to be able to predict the macroscopic properties of the model given a configuration of spins. The renormalization group process allows us to create block spins that represent a group of spins within the lattice. We then determine the new coupling strength for these block spins and then decrease all dimensions of the lattice by a factor depending on the size of each block. If we successfully apply this process multiple times, we are able to condense the Ising model from a 40x40 representation to a 5x5 representation.

For this project, we used DBNs (deep belief networks), a specific type of Deep Neural Network, to successfully replicate this coarse graining process. Our network consisted of three stacked Restricted Boltzmann Machines, resulting in four layers of size 1600, 400, 100 and 25 neurons respectively. We trained the DBN on 20,000 samples of a 40x40 representation of the Ising model near Curie temperature and then successfully compressed them to a 5x5 form. Afterwards, we back propagated the 5x5 compressions of the lattice to measure the accuracy and

performance our DBN. Finally, we examined how the weights of the DBN converge over time to obtain further insights on how neural networks function.

Methodology:

1) Generating Samples of the Ising Model:

The first key step in the process was being able to construct 20,000 samples of the Ising Model to train the Deep Belief Networks. In order to do this, we followed a Monte-Carlo simulation based approach. The below table enumerates the conditions in which we carried out the simulation:

Ising Model Simulation Configuration	
Lattice Size:	40x40
Increments	100
Temperature	J = 0.408 (slightly above the critical temperature)
Number of samples	20,000

Table 1: Simulation Configuration

2) Understanding Neural Networks:

One of the biggest challenges initially was to get a baseline familiarity with deep learning and understanding how to work with Deep Neural Networks. In order to do this, a good amount of time was spent getting familiar with how to work with frameworks like TensorFlow and Keras.

Initially, we built a neural network with an architecture as follows: 12,8,1 with the first two layers using a rectifier activation function and the last one using a sigmoid activation

function. We then trained the neural network on the pima-indians diabetes Dataset for 150 epochs and measured its accuracy.

3) Training and Building the Deep Belief Network:

The next major step revolved around training and constructing the actual Deep Belief Network to compress the Ising model from a 40x40 representation into a 5x5 form. Initially, we spent a majority of time cleaning and reformatting the dataset generated from the simulations to ensure it could be used for training the DBN.

Steps involved in Reformatting and Cleaning Data:

1) Converting lattice from a 40x40 matrix into a vector of size 1600 and writing it to a text file.
2) Removing out all non numeric characters from the text file.
3) All down spins in our simulations were represented by a -1. I had to, however, convert the down spins to a 0 as the DBN would otherwise not accept the input.
4) Formatting the data from the text file into 1600x2000 matrix to ensure it is of an appropriate format for the neural network.

After cleaning and formatting the dataset, we constructed a DBN with the following architecture:

Deep Belief Network architecture	
Input Layer	1600 neurons
Hidden Layer #1	400 neurons
Hidden Layer #2	100 neurons
Output Layer	25 neurons
epochs	200
batch size	100
shuffle (randomizing order of data	True

before each iteration)	

Table 3: Architecture of DBN

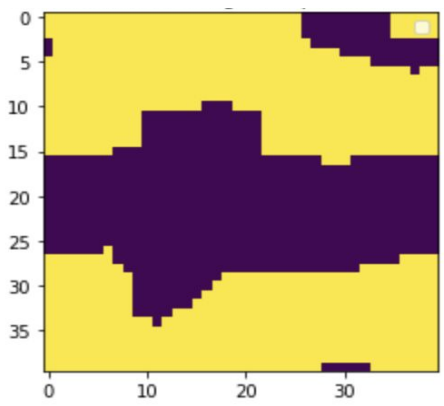
Then, the DBN was trained on two datasets with 200 and 20,000 samples of the Ising model respectively. After this, we evaluated the performance of the DBN for both samples on various metrics such as magnetization accuracy to see how it performed in both situations. We also measured how the weights of the DBN converged over time.

Results and Discussion:

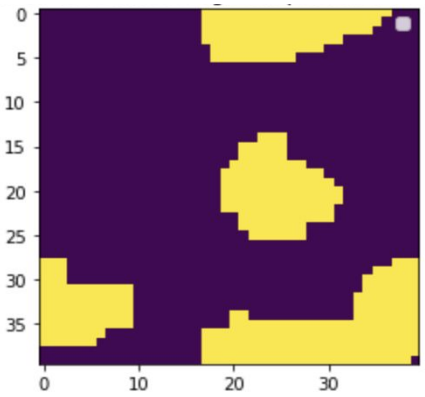
1) Ising Model

Below are snapshots of the simulation, starting with the initial configuration. Initially, there are three rectangular islands with a majority of the spins being downwards. As we progress, however, with each iteration, there is a shift towards a more scattered island formulation. It is interesting to notice that across all samples all the positive and negative spins tend to aggregate in certain areas, leading to the formation of clear islands. Additionally, there appears to be no observable correlation between the magnetization and the placement of spins across all the snapshots.

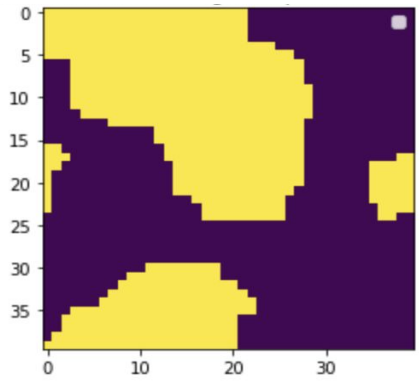
Screenshots of Simulation:



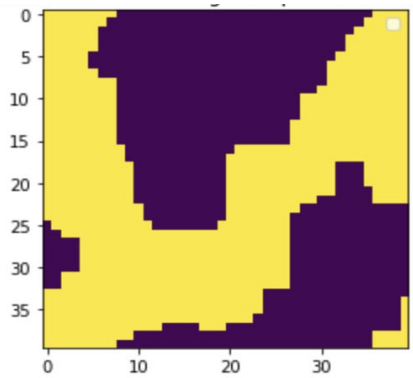
Initial Configuration



After 100 iterations



After 2000 iterations



After 20,000 iterations

2) Understanding Neural Networks:

Performance of Neural Network for diabetes dataset	
Accuracy	77.08%

Table 2: Measuring performance of NN on pima dataset

Accuracy: A measure of how accurately the neural network is able predict patient's diabetes status based on a series of 12 attributes such as age, BMI, number of times pregnancy etc. It is evaluated by comparing the output of the neural network to the actual label.

Besides familiarizing ourselves with the tools needed to be able to train the DBN for the project, we also spent some time understanding the theory behind how neural networks functions. Below are the key steps that elaborate the underlying principles behind how neural networks learn from a dataset:

1) First we feed in an input, specify the architecture (layers), and the learning rate
2) The neural network then initializes all weights to a small random number
3) Then until we reach converge, that is until there is a trivial change in weights, the neural network repeatedly executes the below steps:

1) Feed Forward: The Neural Network propagates each example through the network and determines the output for every neuron in each layer.
2) Propagate Backward: The Neural Network then propagates the error backward and calculates the error rate for each output neuron and hidden neuron using the below formulas:

   Output neuron: $\delta_k = o_k(1 - o_k)(y_k - o_k)$, where y_k is the actual output and o_k is the output calculated by the neuron k.

   Hidden neuron: $\delta_h = o_h(1 - o_h)\Sigma_{k \in Succ(h)} w_{hk} \delta_k$, where succ(h) refers to the neurons whose immediate inputs include the output of neuron h, o_h is the output of hidden neuron h, and w_{hk} is the weight between neuron h to its successor neuron k
3) Update weights, $w_{ij} = w_{ij} + \alpha\delta_j x_{ij}$, where w_{ij} is the weight from neuron i to j, α is the learning rate, δ_j is the error for neuron j and x_{ij} refers to the ith input to neuron j.

3) Performance of DBN

Backpropagation:

DNN Backpropagation Performance		
Number of Samples	200	20,000
Average Net Magnetization	0.51	51.74
Magnetization Accuracy for Back Propagation	58.3%	82.1%
Chi-Square	7.36	317.8

Table 4: Performance of Deep Belief Network

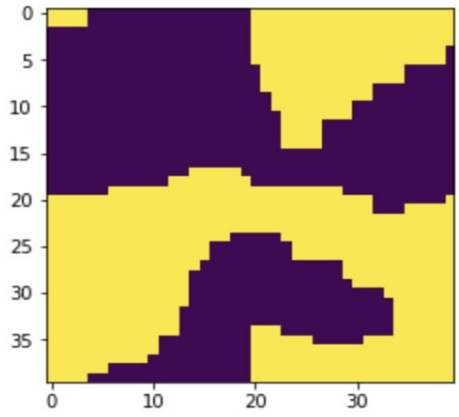
Formula for Magnetization: $\sum_{i=1}^{N} S_i$, where S_i refers to the value of each spin

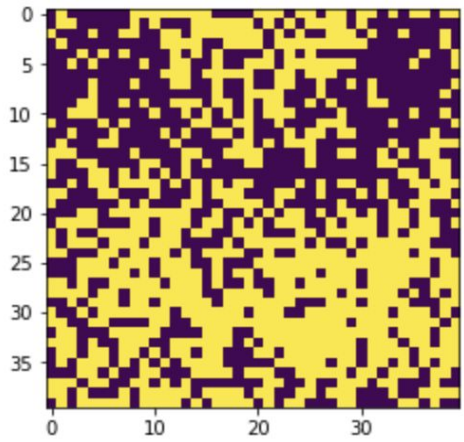
Formula for Average Magnetization: $\frac{1}{n}\sum_{i=1}^{N} S_i$, where S_i refers to the value of each spin.

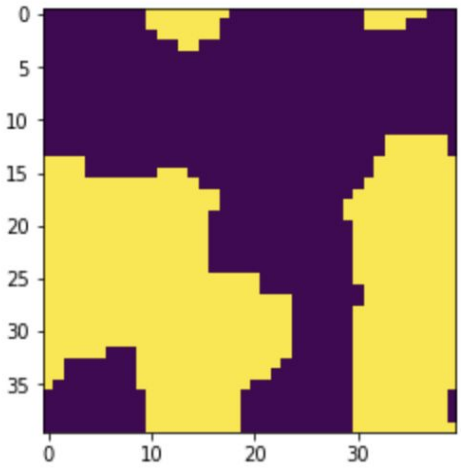
Formula for Chi-Square: $\sum_{i=1}^{N} \frac{(M_{observed}-M_{actual})^2}{M_{actual}}$, where $M_{observed}$ is the observed magnetization and M_{actual} is the actual magnetization value

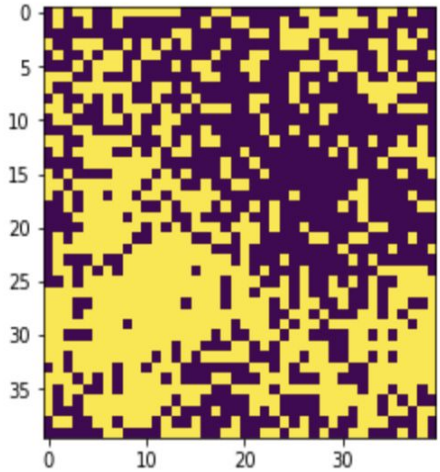
As evident by the above table, the DBN's performance drastically improved when we trained on 20,000 samples as opposed to 200. For instance, we were able to capture magnetization roughly 23.8% more effectively with 20,000 samples.

Screenshots of actual vs back-propagated representation for 200 samples:

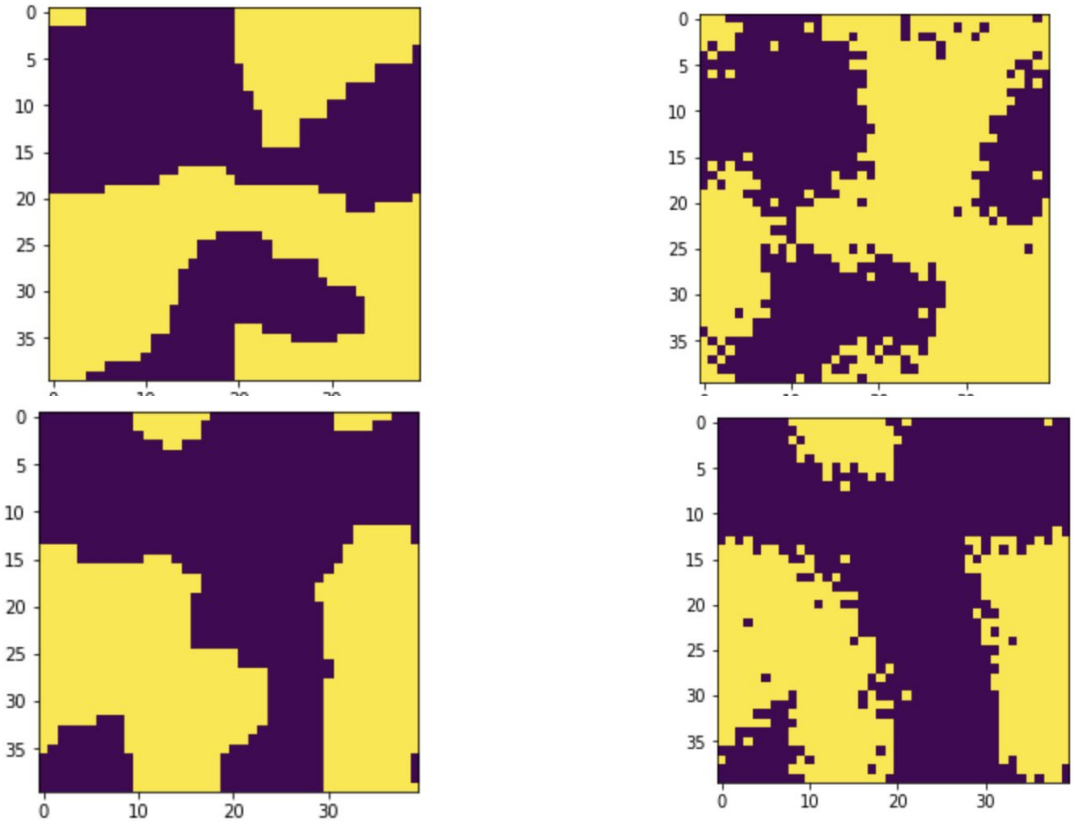








Screenshots of actual vs back propagated representation for 20,000 samples:



200 sample:

The above screenshots showcase the back propagated results of recreating the input lattice using the deep belief network. One thing that is evident with the 200 sample example is that the reconstructions were fairly grainy and the DBN was not able to recreate the island formations accurately. Additionally with 200 samples, it is hard to deduce any meaningful information about the lattice from the reconstruction as the magnetization itself was relatively inaccurate. It clearly appears as though the placement of up spins and down spins did not correspond well with with the original model.

20,000 samples:

On the other hand with 20,000 samples, the back propagated reconstructions were much more accurate when compared to the original lattice. The DBN in this case was able to capture the island formulations and magnetization fairly accurately. It was particularly surprising to observe how accurate the island mapping was between the original and the reconstructions. The results were even more accurate than the reconstructions found in the "An exact mapping between the Variational Renormalization group and Deep Learning Paper".

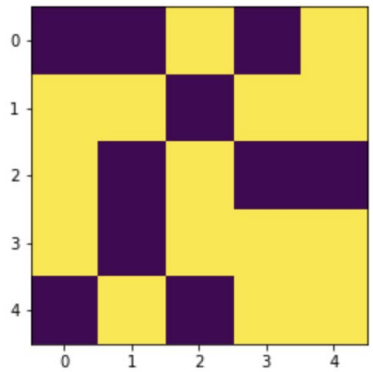
Compression:

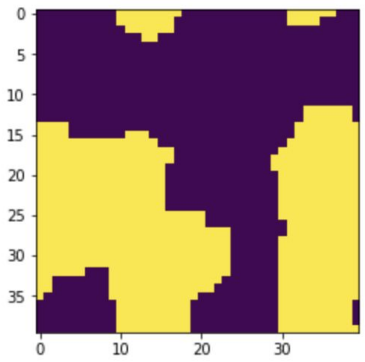
Here, we explore how effective the DBN was in capturing the relevant features of the lattice by compressing it into a 5x5 format. The below results are from training the DBN through 20,000 samples.

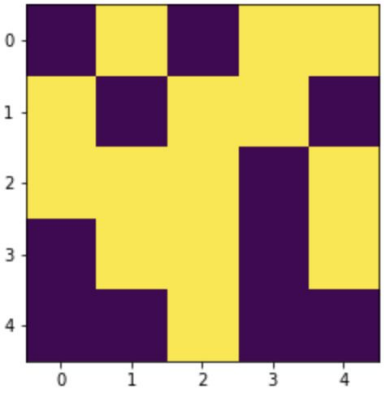
DBN Compression Performance	
Magnetization Accuracy	68.8%
Average Net Magnetization	0.2595

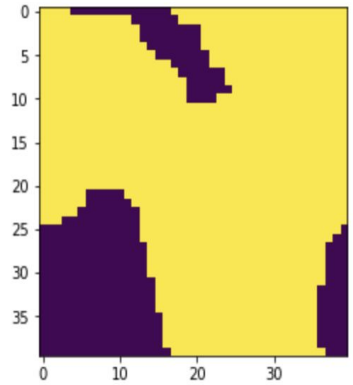
The magnetization accuracy with 20,000 samples in the compressed form was slightly lower than what we had seen in the back-propagation comparisons. Part of this has to do with the fact that in a 5x5 form even a small error can impact the overall magnetization as the lattice only consists of 25 spins. Overall, it is still impressive that the Deep Belief Network was able to capture the magnetization of a 40x40 representation in a 5x5 form with 68.8% accuracy.

Screenshots comparing compressed 5x5 samples to original representations:







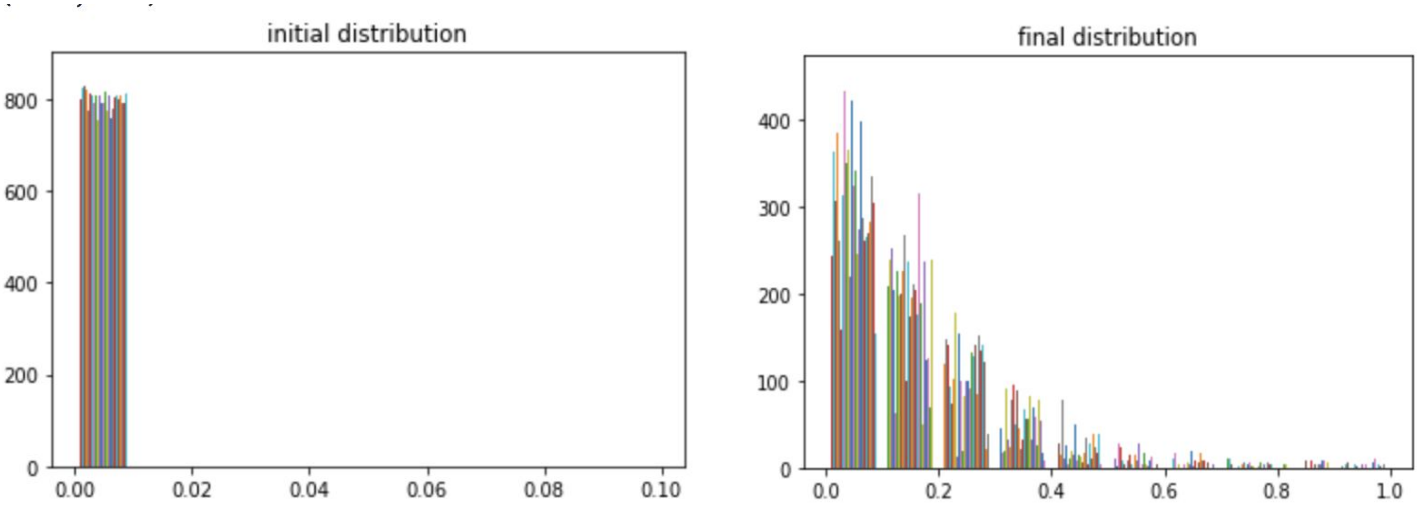


The Deep Belief Network also was not able to capture the island formations as effectively during the compression phase especially in comparison to what we had seen with the back propagated samples. This is obviously expected as with the 5x5 form as even a single spin being misaligned could result in a different formation. It is, however, interesting to notice in certain areas the concentration of up spins and down spins in certain areas between the 40x40 lattice and the compressed form are very similar. For instance, in the bottom two images the purple regions are situated on the bottom left and right corners for both the compressed lattice and the actual.

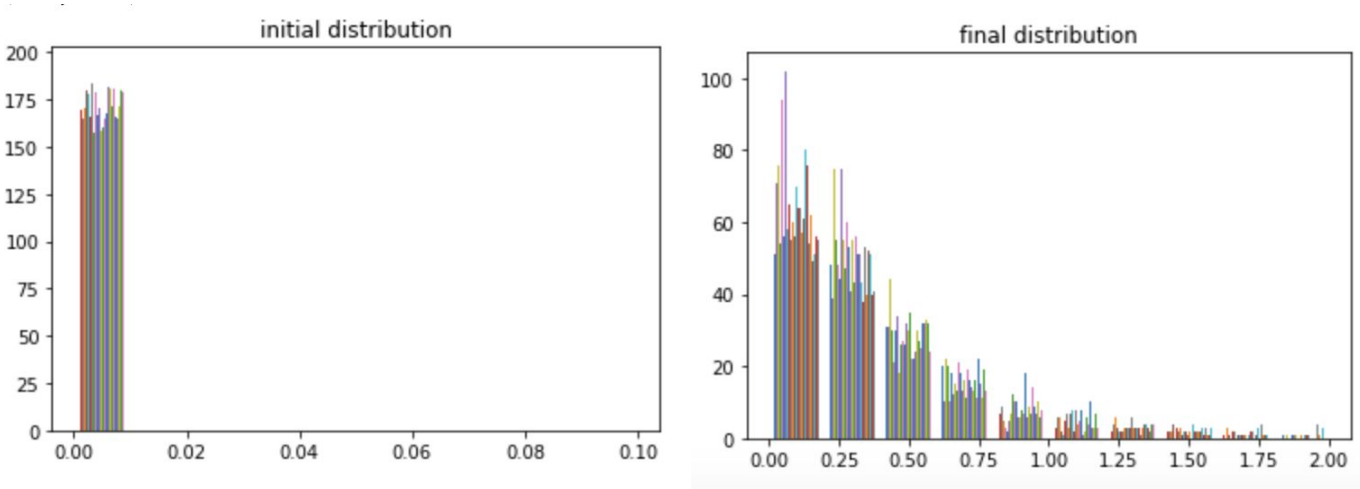
Weight Distribution:

Below are a sample of histograms that showcase the weight distributions of the initial distribution and final distribution for each layer of the deep belief network. The weights were initially recorded before we trained the DBN on any samples and after we trained the DBN on all 20,000 samples.

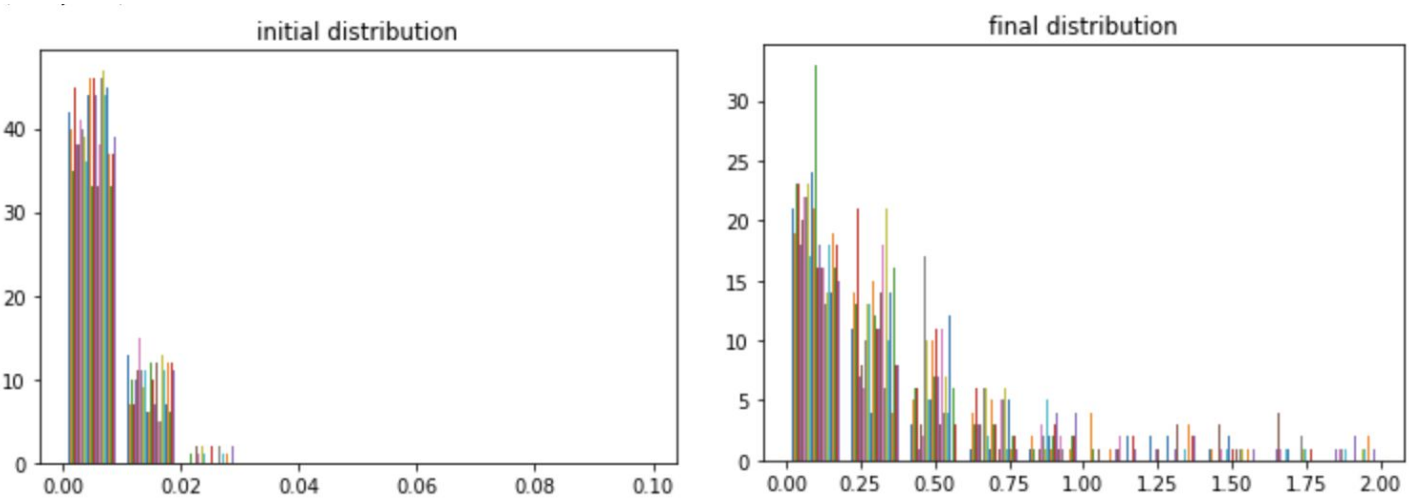
First Layer (1600,400):



Second Layer (400,100):



Third Layer (100,25):



Initially, the weights are sampled uniformly between 0 and 0.01, except in the case of the third layer where a few weights lie outside that range. What then becomes apparent across all three layers is that the weights of the DBN appears to converge towards a power law distribution over time. It is intriguing to observe how the DBN begins to function more like a scale-free network over time. We start to see a pattern, where weights associated with a few input neurons grow in importance for each output neuron while others tend to decline in importance.

Conclusion :

Deep learning has been a crucial advancement in the field of artificial intelligence and there have been a lot of questions around how it has been so effective. The results from this project so far showcase some initial results in terms of understanding the theoretical underpinnings behind why neural networks are so effective. Firstly, we are able to see that there is indeed a close one to one mapping between the DBNs and the variational renormalization group through the Ising model example. We were able to compress the lattice from a 40x40 form to a 5x5 form and yet retain information about key macroscopic properties like magnetization and island formations upon backpropagation. The 5x5 compressed lattice on its own, however, is

not as useful in terms of being able to accurately capture macroscopic properties of the Ising model

Additionally, it also was interesting to notice that overtime the Deep Belief Network appears to function more like a scale free network. That is, the neurons in the output layers tend to have high weight values associated with a few input neurons and have low weight values associated with the rest of the input neurons. These weight properties could help uncover some useful insights into how the DBN learns from each example.

Overall, this project is just the foundation of a longer study for understanding the theoretical properties of deep neural networks. There still remains a lot more to be done in terms of experimenting with a wider range of datasets to uncover further insights about the theoretical underpinnings behind DNNs.

References:

1) Mehta, P., Schwab, D.J.: An exact mapping between the variational renormalization group and deep learning. arXiv:1410.3831 (2014)