BT6270: Computational Neuroscience

Assignment-3: Hopfield Network

Siddharth Betala
BE19B032

References

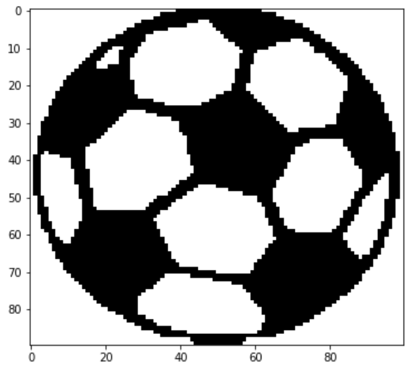
1. Professor VS Chakravarthy's Classnotes and Slides
2. BT6270 Computational Neuroscience Aug-Nov 2021 Playlist by CNS Lab IITM
3. MIT 9.40 Introduction to Neural Computation, Spring 2018 by Michale Fee

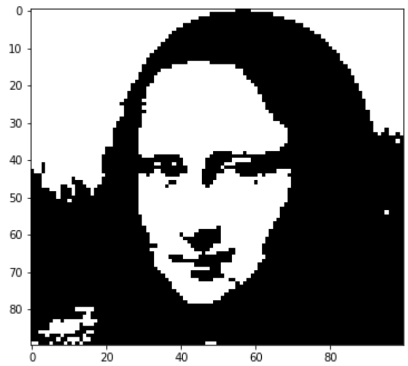
Question-1

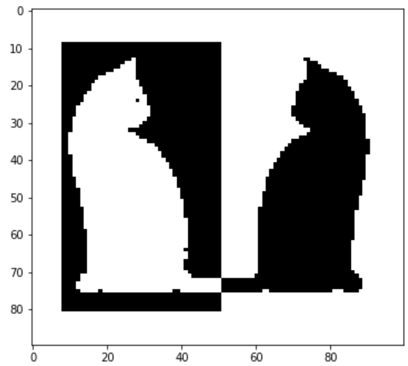
Loading the data

Using the **loadtxt** function, the images were read and converted into numpy arrays. The images were loaded using the matplotlib.pyplot library's **imshow()** function. The data was cleaned up with the **np.sign(**) function, which is the numpy signum function. This was done to ensure that all elements in the matrices represented black pixels as +1 and white pixels as -1. Furthermore, the pixels with a value of 0 were assigned a value of 1. The final assignment ensured that monotonic error was reduced.

The visualized images are as follows:







Hopfield Network

I built both continuous and discrete Hopfield networks. Continuous Hopfield networks were used instead of discrete Hopfield networks because they converged rapidly.

The discrete Hopfield network matrix equations are as follows:

$$W = \frac{1}{N} S S^T$$

$$V(t+1) = \sigma(WV(t))$$

where S is the matrix containing the pattern of the ball. N is the number of neurons which is taken to be 9000.

The matrix equations used for continuous hopfield networks are as follows:

$$W = \frac{1}{N} S S^T$$

The next equation used to calculate the dynamics of the Hopfield network:

$$\frac{dU}{dt} = -U + W tanh(\lambda U)$$

We set

$$V = tanh(\lambda U)$$

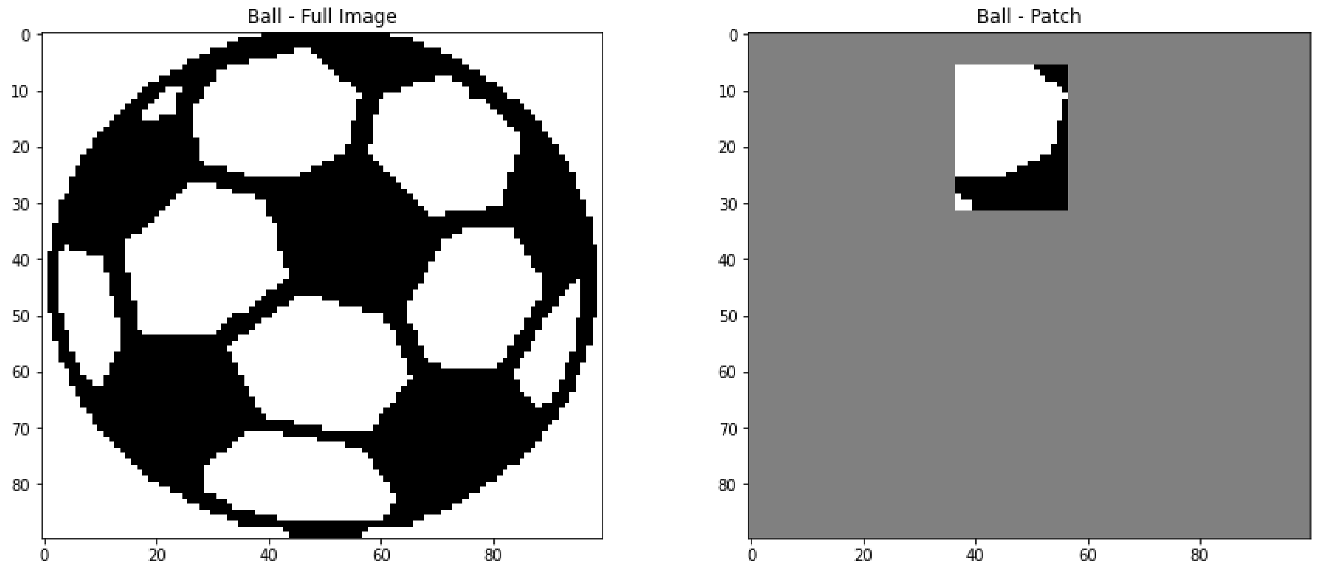
The parameters: λ and dt used are 20 and 0:01 respectively.

For best understanding of the models, please refer to the notebook [here](),

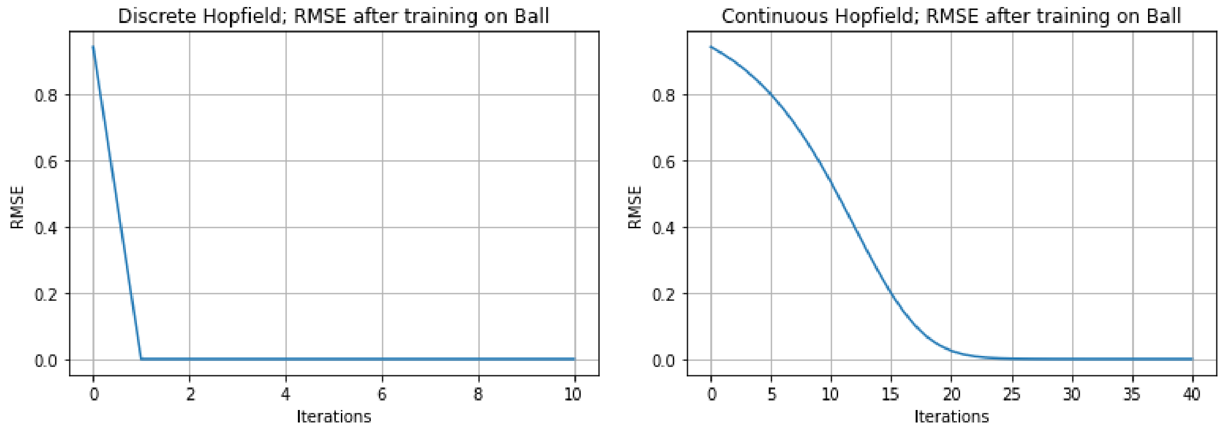
Question-2

At random locations throughout the image, patches are created with a minimum width of 20 pixels, a maximum width of 45 pixels, a minimum height of 15 pixels, and a maximum height of 45 pixels.

Patches are generated as inputs to the network by employing the full image. The patch generated is as follows:

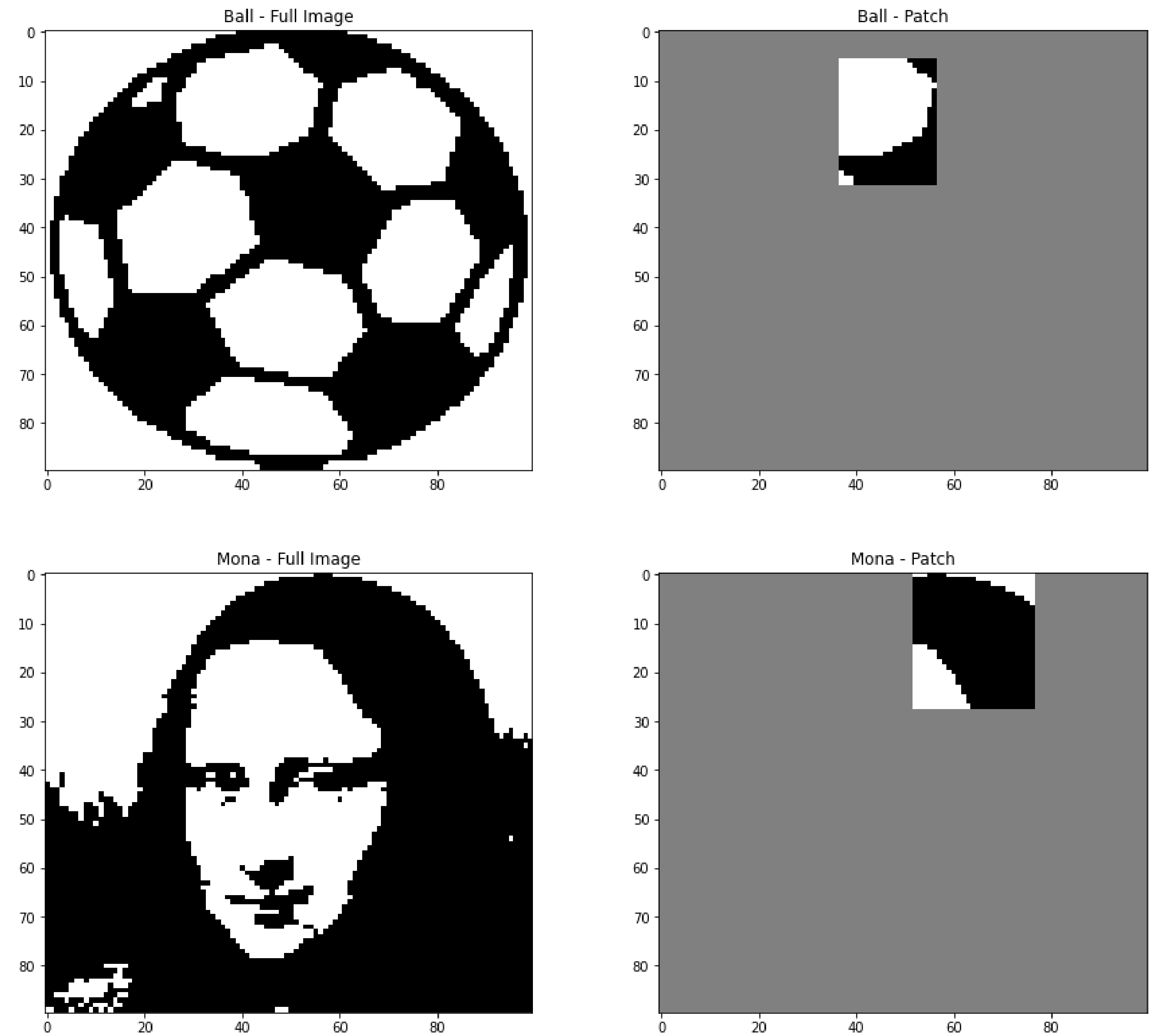


The variation of RMSE over iterations for both the discrete and continuous Hopfield networks is as follows:

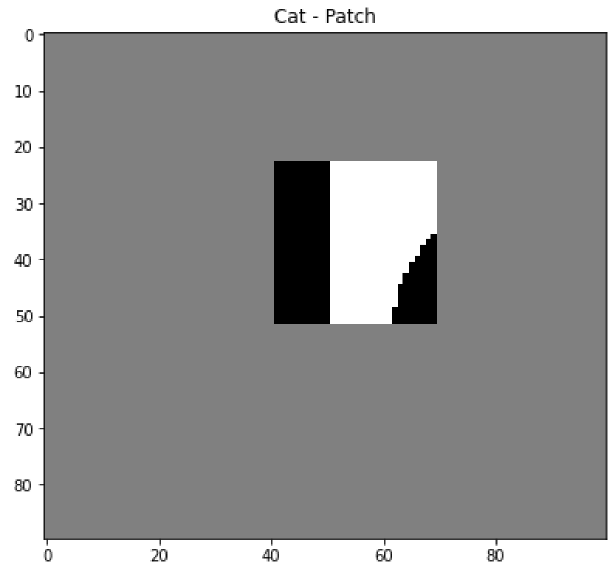


Question-3

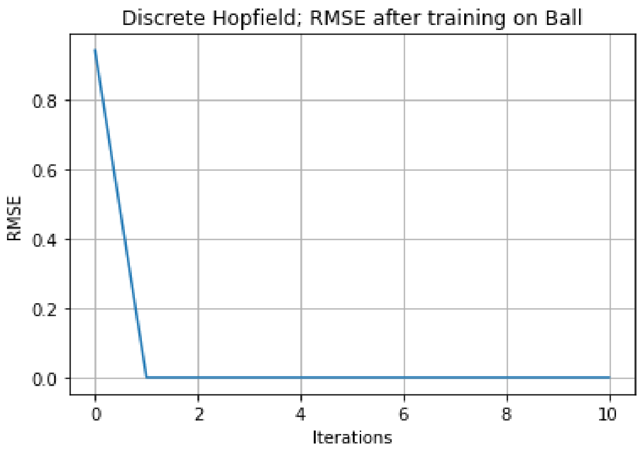
Plot of Input Patches

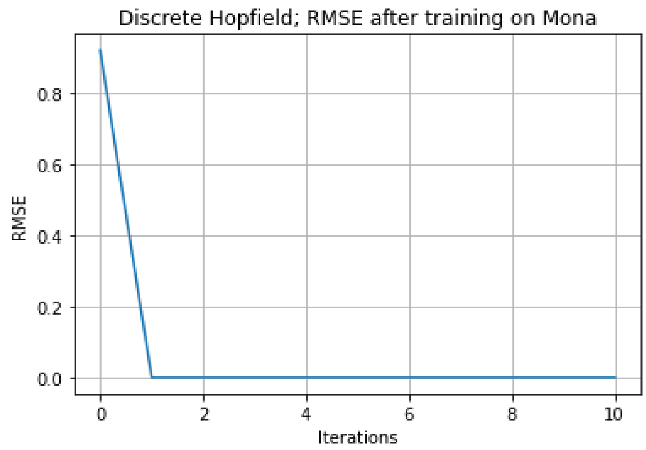


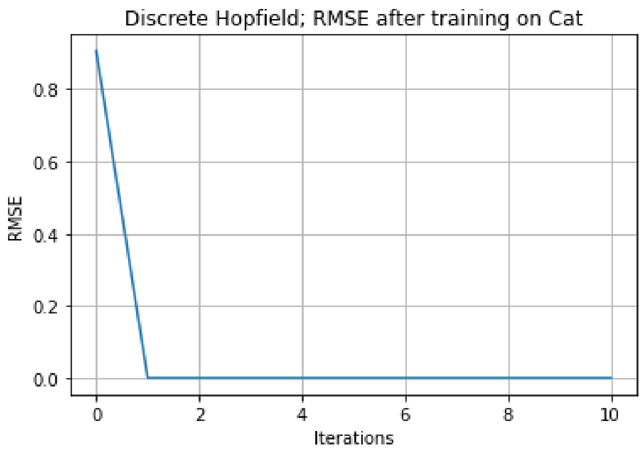

Cat - Full Image
0
10
20
30
40
50
60
70
80
0
20
40
60
80


Cat - Patch
0
10
20
30
40
50
60
70
80
0
20
40
60
80

Discrete Hopfield Network: Variation of RMSE over iterations

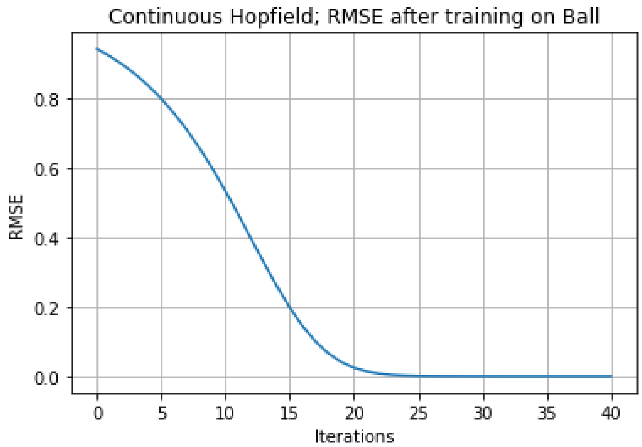


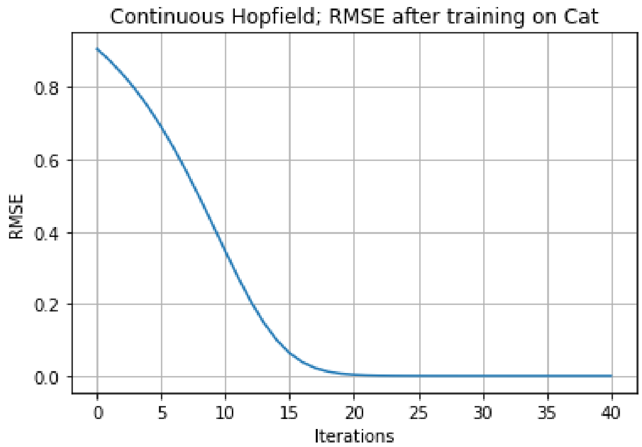




Rapid convergence of error to 0 is observed. Hence, we use continuous Hopfield networks for further analysis.

Continuous Hopfield Network: Variation of RMSE over iterations

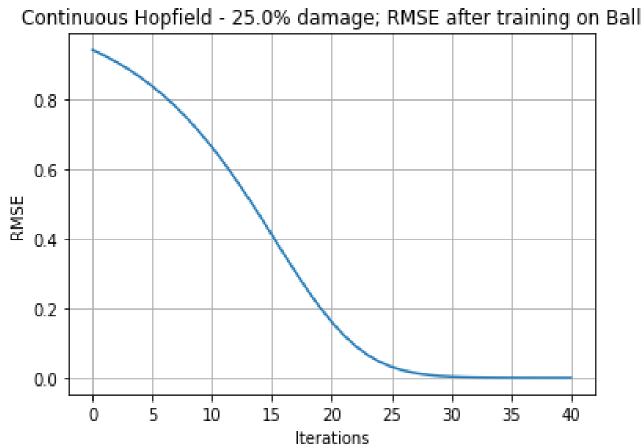


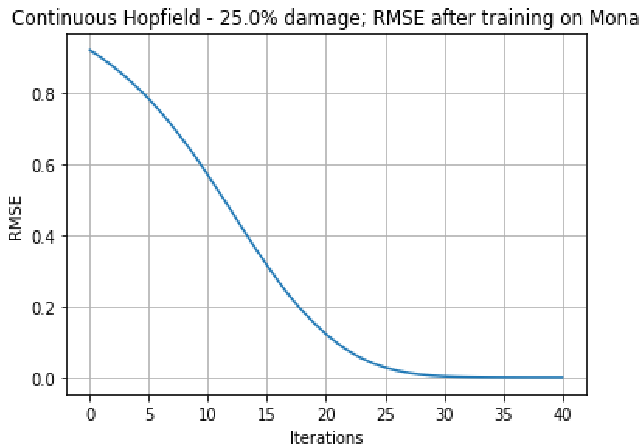


Only RMSE plots of Continuous Hopfield Networks were assessed for image recovery after weight damage due to their smooth nature.

We'll further look at the pace of convergence and reconstruction (error) with a certain degree of damage to weights going ahead. Damaging weights means setting the weights of certain neurons in the network to 0.

Variation of RMSE over iterations for 25% damage to weights





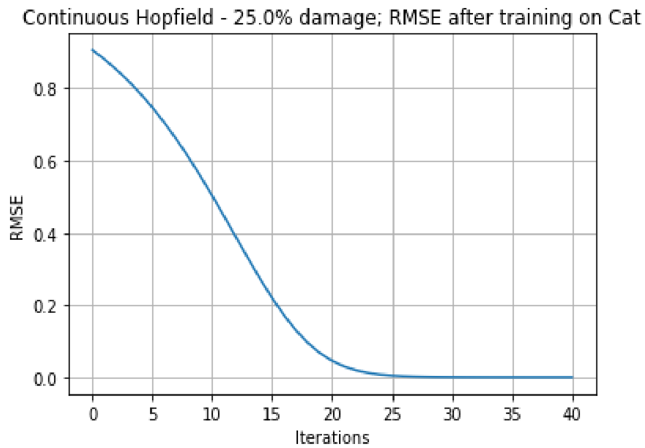
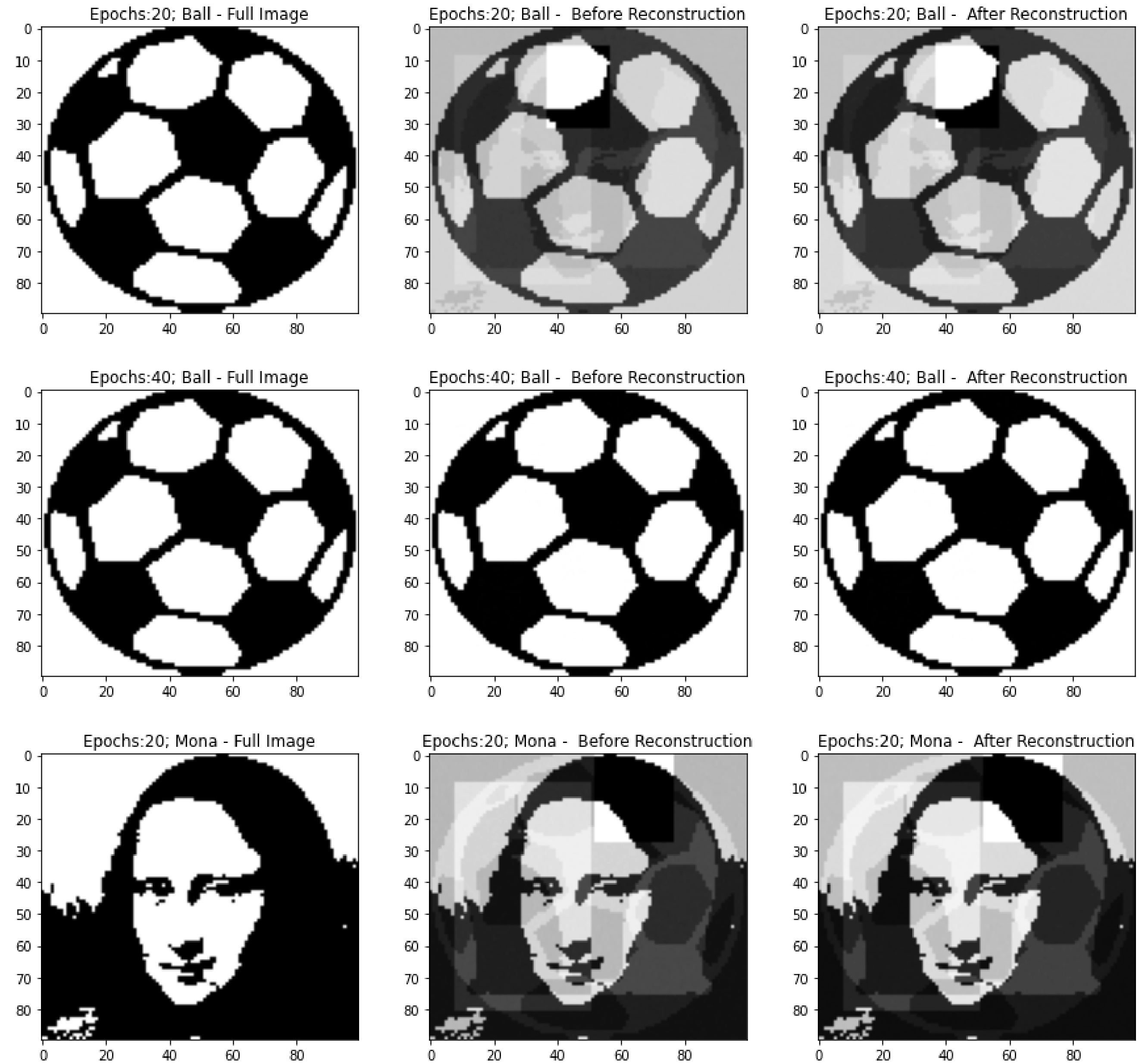
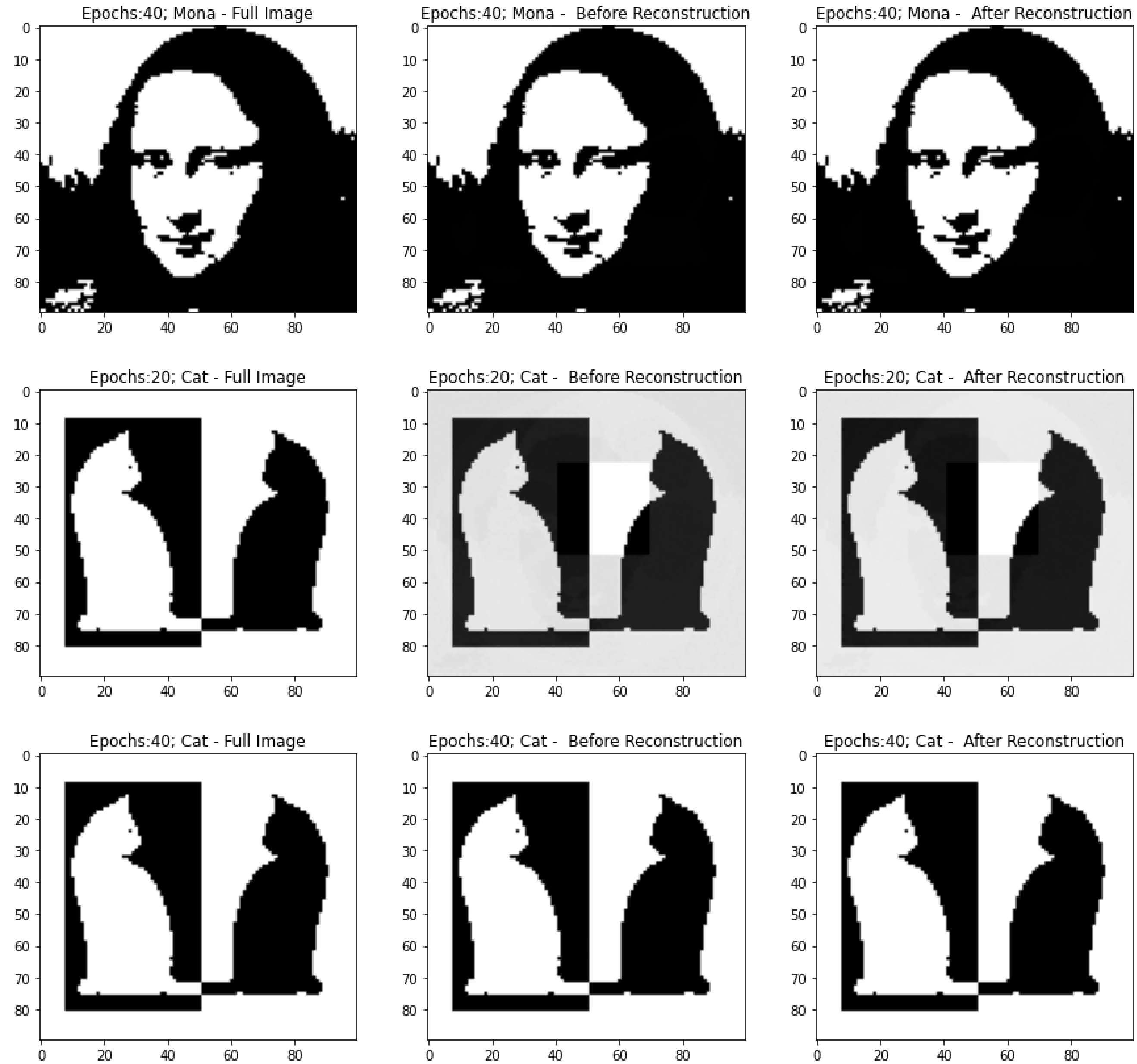


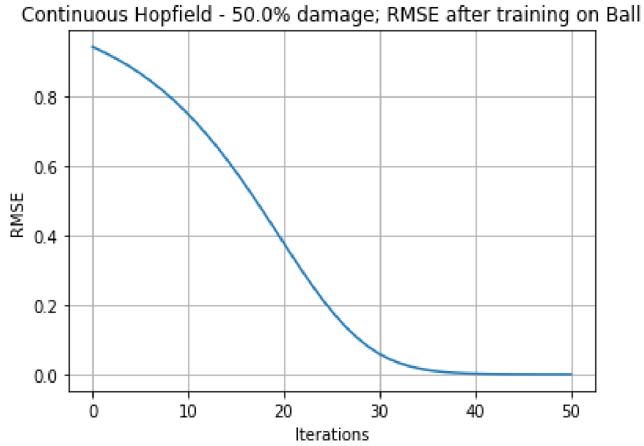
Image Reconstruction

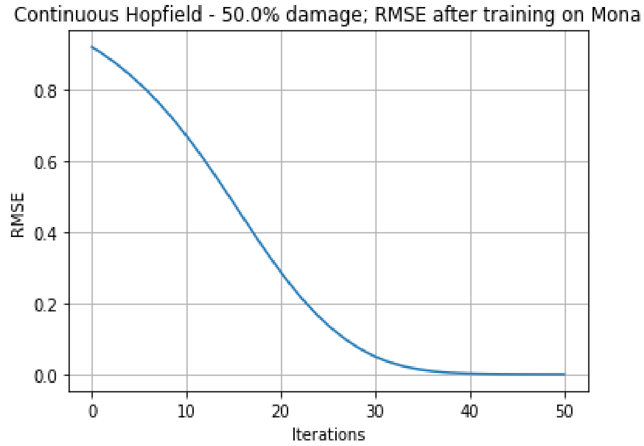




Epochs:40; Mona - Full Image
Epochs:40; Mona - Before Reconstruction
Epochs:40; Mona - After Reconstruction
Epochs:20; Cat - Full Image
Epochs:20; Cat - Before Reconstruction
Epochs:20; Cat - After Reconstruction
Epochs:40; Cat - Full Image
Epochs:40; Cat - Before Reconstruction
Epochs:40; Cat - After Reconstruction

Variation of RMSE over iterations for 50% damage to weights





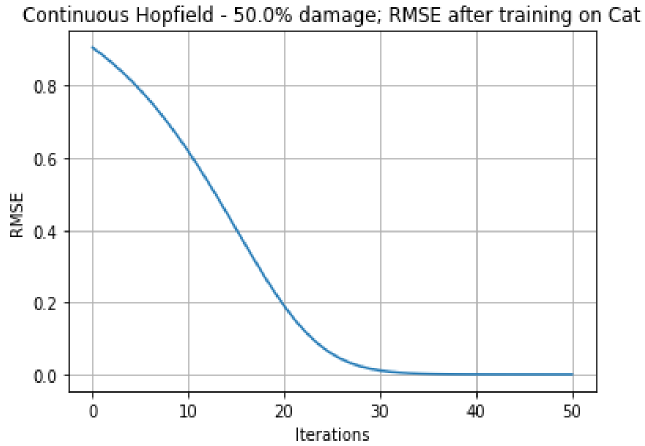
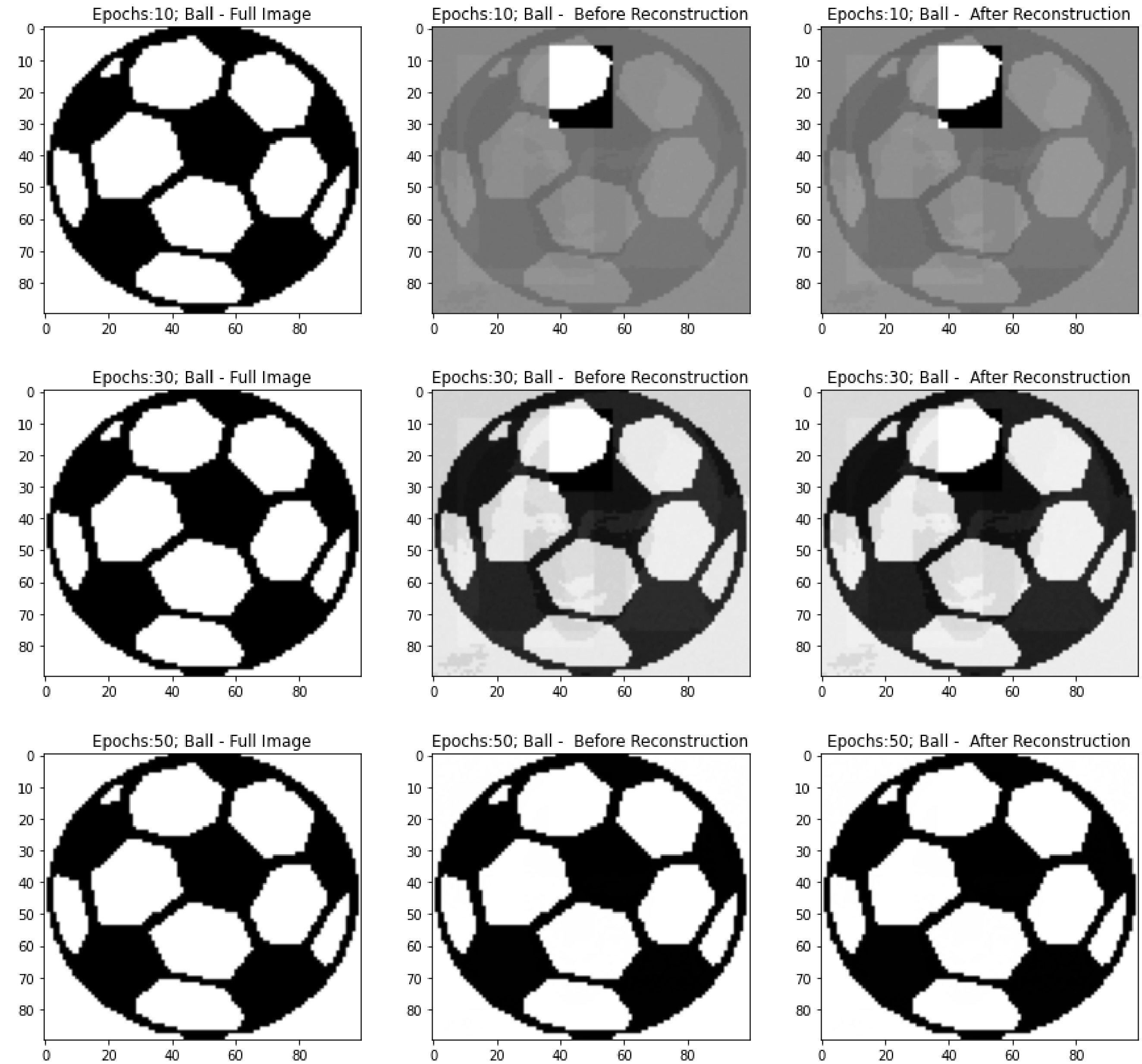
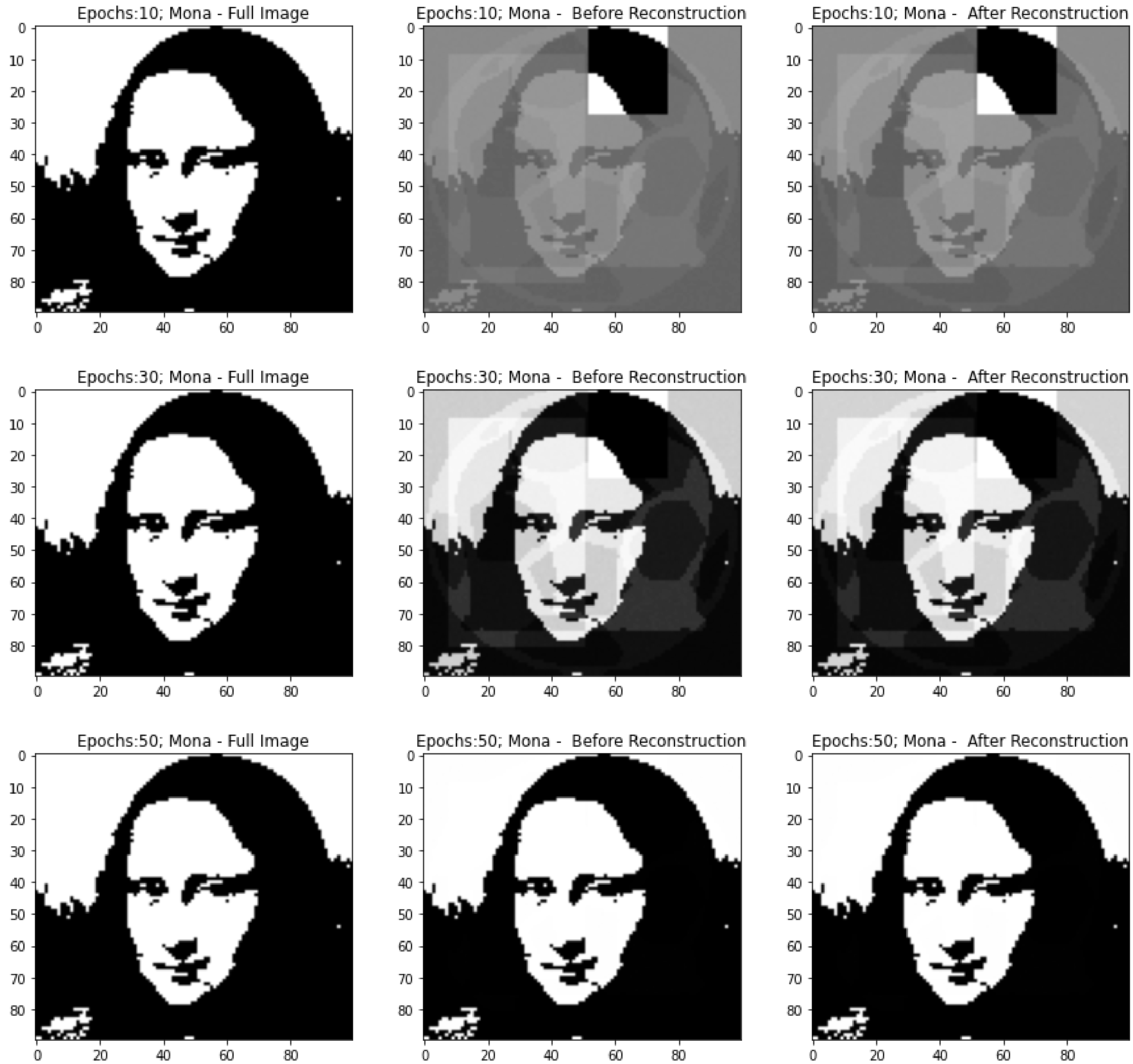
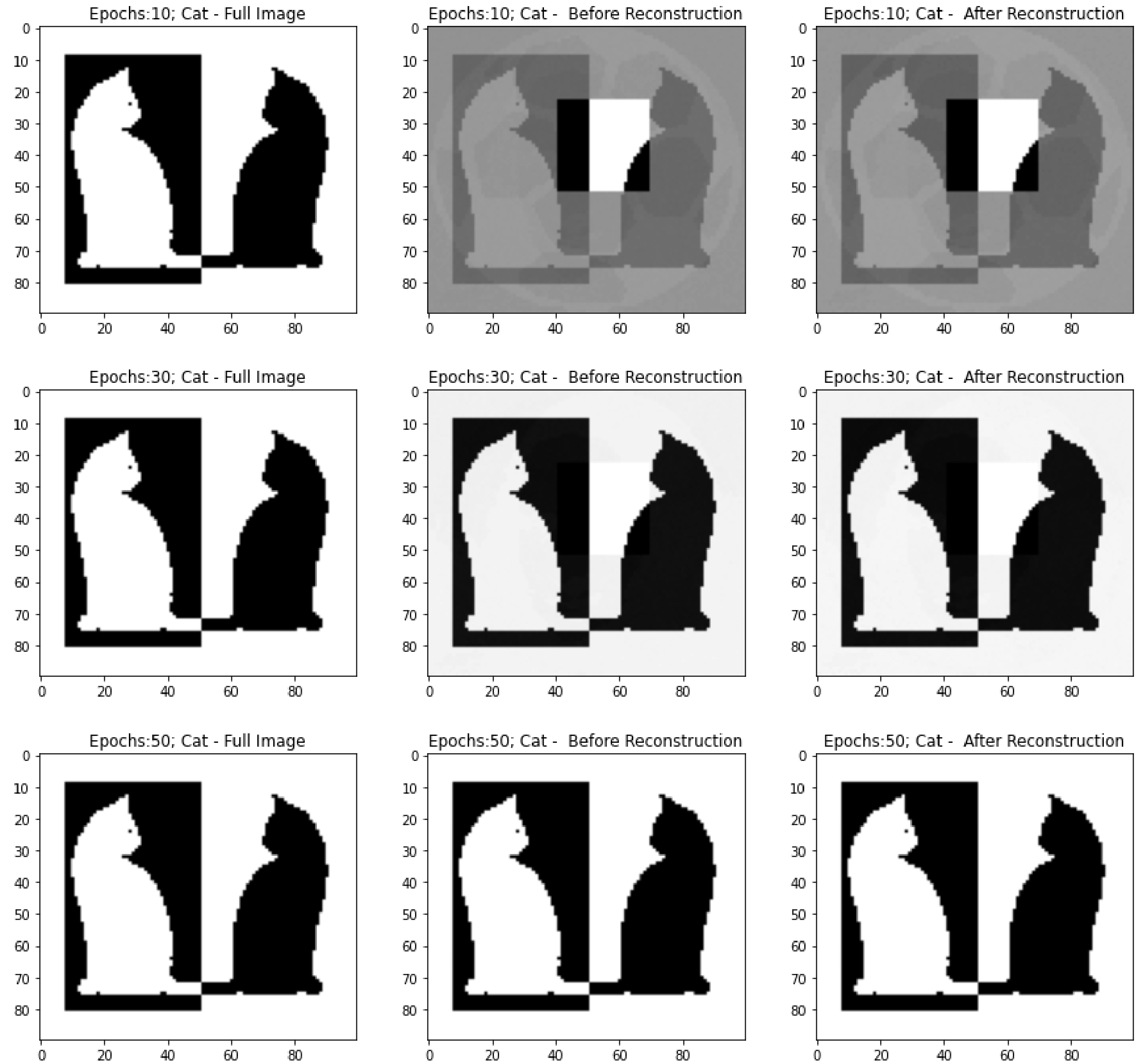


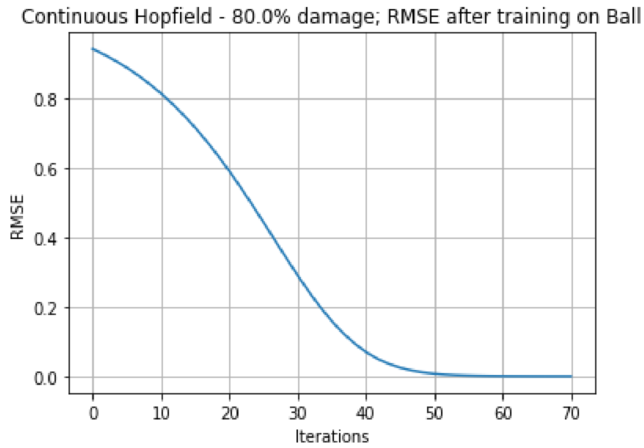
Image Reconstruction

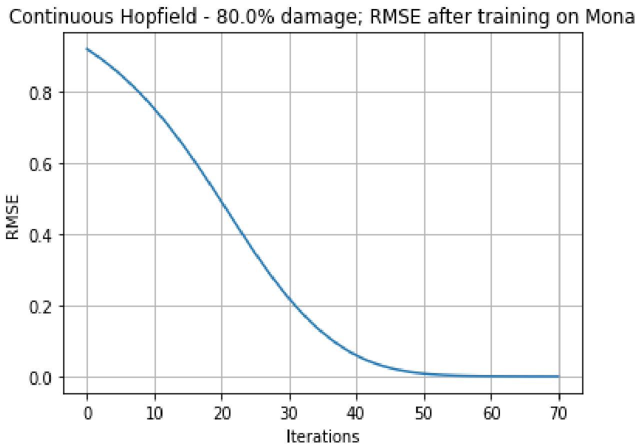



Epochs:10; Mona - Full Image
Epochs:10; Mona - Before Reconstruction
Epochs:10; Mona - After Reconstruction
Epochs:30; Mona - Full Image
Epochs:30; Mona - Before Reconstruction
Epochs:30; Mona - After Reconstruction
Epochs:50; Mona - Full Image
Epochs:50; Mona - Before Reconstruction
Epochs:50; Mona - After Reconstruction


Epochs:10; Cat - Full Image
Epochs:10; Cat - Before Reconstruction
Epochs:10; Cat - After Reconstruction
Epochs:30; Cat - Full Image
Epochs:30; Cat - Before Reconstruction
Epochs:30; Cat - After Reconstruction
Epochs:50; Cat - Full Image
Epochs:50; Cat - Before Reconstruction
Epochs:50; Cat - After Reconstruction

Variation of RMSE over iterations for 80% damage to weights





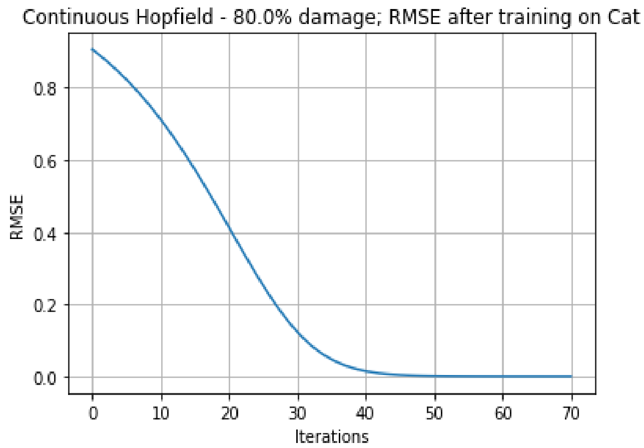
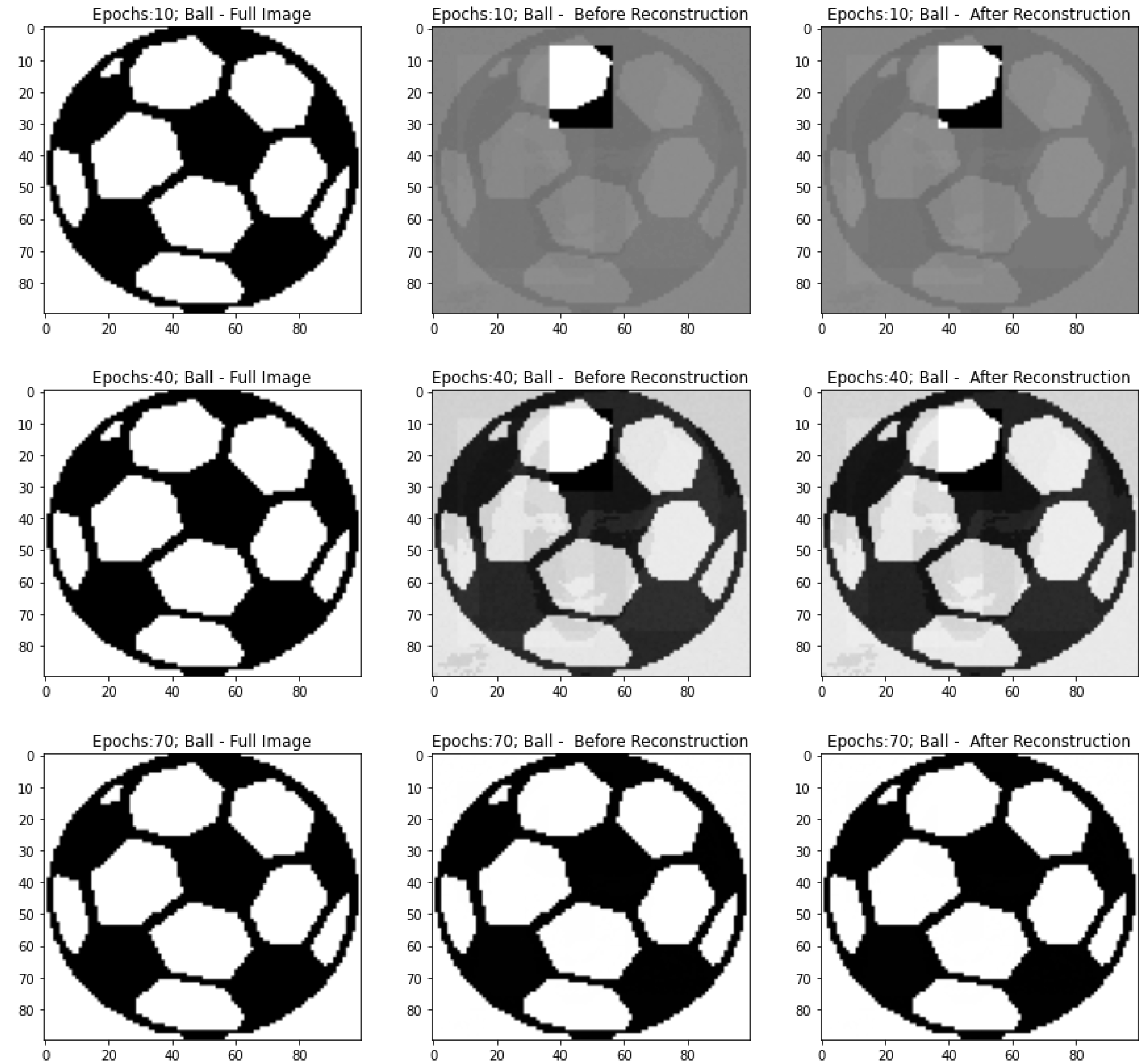
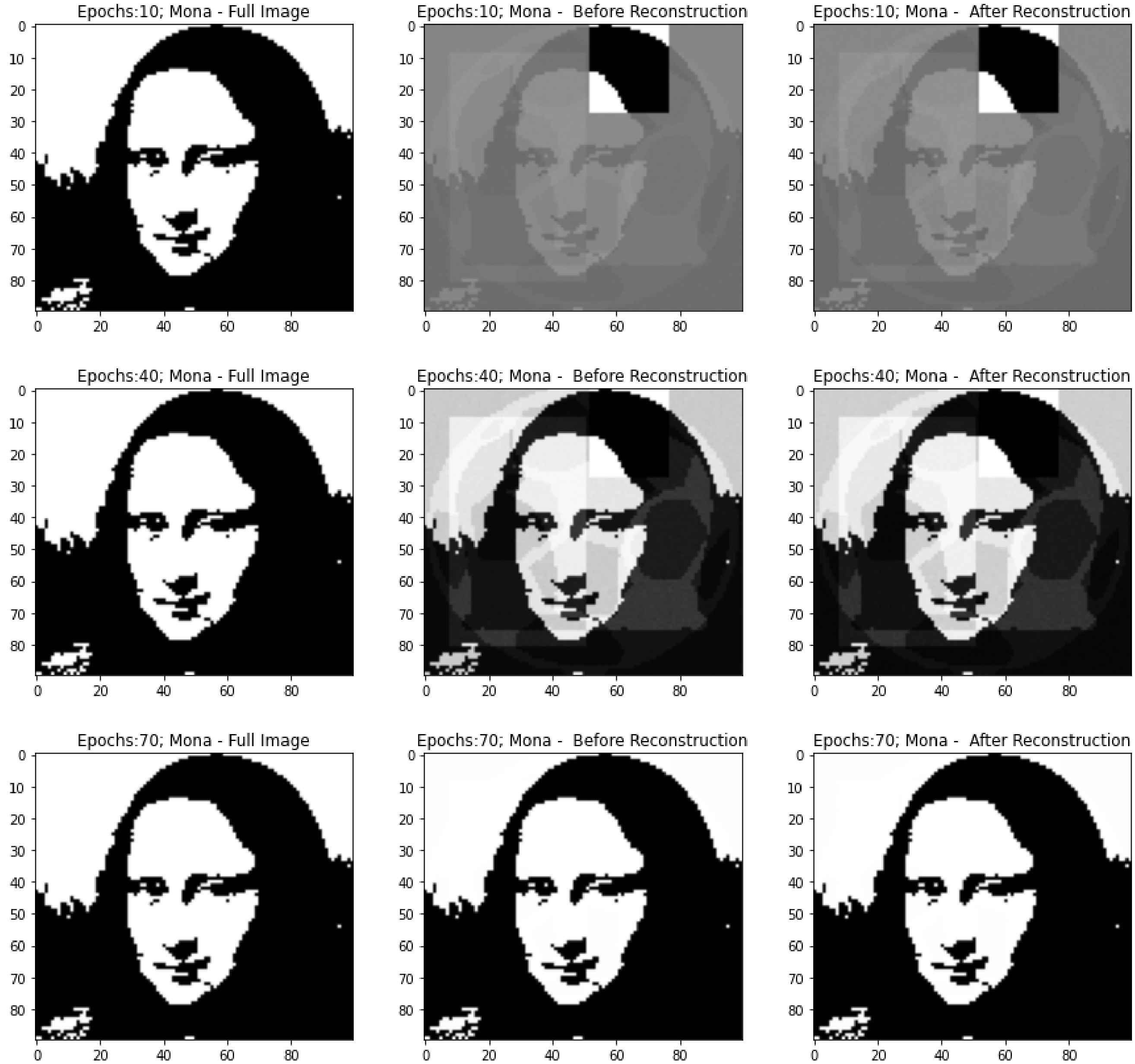
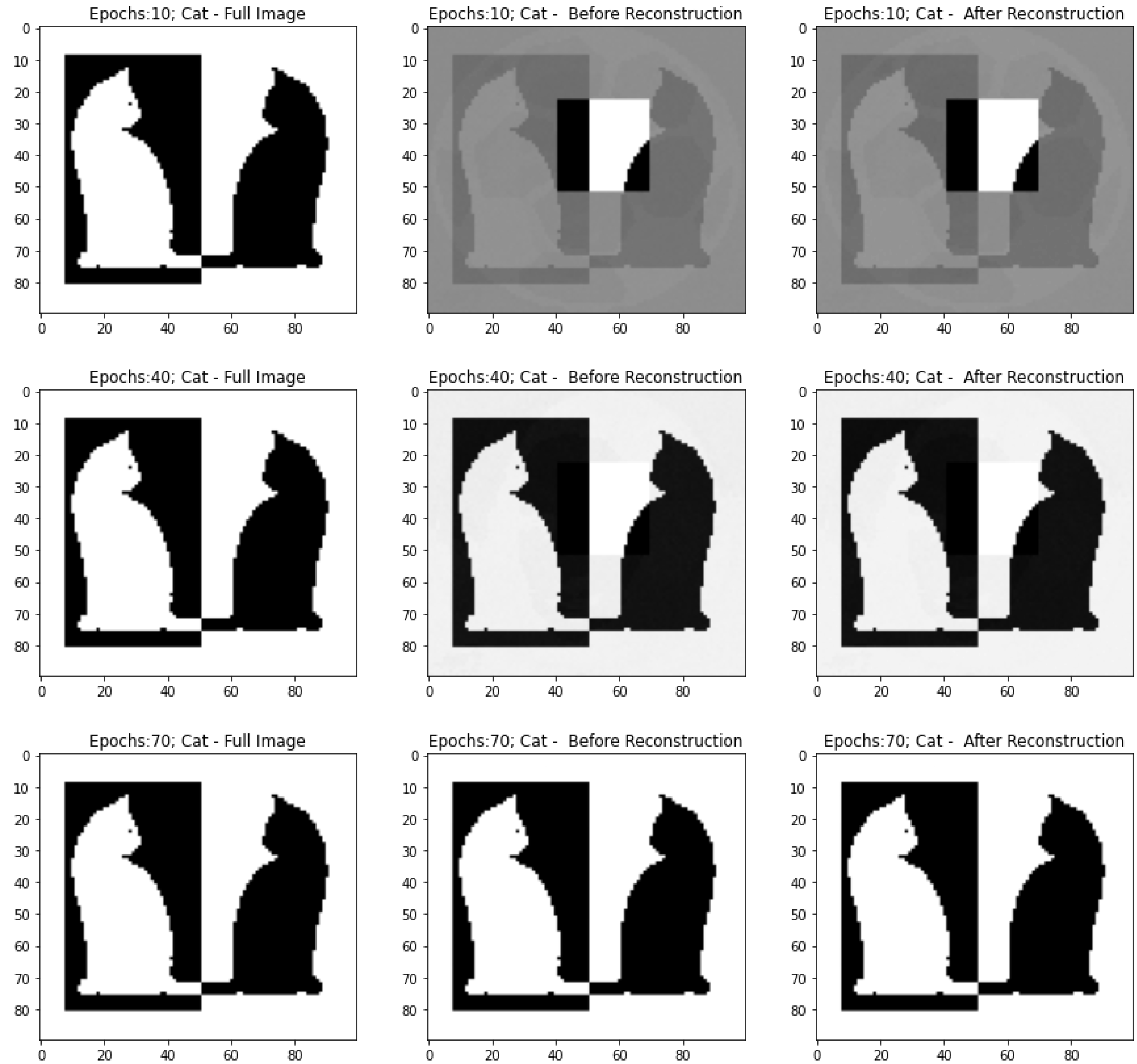


Image Reconstruction





Epochs:10; Mona - Full Image
Epochs:10; Mona - Before Reconstruction
Epochs:10; Mona - After Reconstruction
Epochs:40; Mona - Full Image
Epochs:40; Mona - Before Reconstruction
Epochs:40; Mona - After Reconstruction
Epochs:70; Mona - Full Image
Epochs:70; Mona - Before Reconstruction
Epochs:70; Mona - After Reconstruction


Epochs:10; Cat - Full Image
Epochs:10; Cat - Before Reconstruction
Epochs:10; Cat - After Reconstruction
Epochs:40; Cat - Full Image
Epochs:40; Cat - Before Reconstruction
Epochs:40; Cat - After Reconstruction
Epochs:70; Cat - Full Image
Epochs:70; Cat - Before Reconstruction
Epochs:70; Cat - After Reconstruction

Conclusion

- We can observe from the RMSE charts above that the number of epochs required to obtain the initial figure increases as the damage fraction increases. This is expected since the complexity of the task increases when there are a greater number of missing elements (participating neurons).
- The more patterns that are stored in memory, the more iterations are needed to retrieve the original image. This is to be expected given that the task's complexity rises as the number of patterns in the memory increases.
- In intermediate epochs while reconstruction, an overlap between different images (mona and ball, for example) can be seen in the reconstructed image. This indicates that the network retains some information while running, hinting towards its recurrent nature, which shows its potential for modeling memory.