Electrical Engineering Department
Sharif University of Technology

Title:

Midterm

Neural Networks

Author:
Mohammad Hossein Najafi (97103938)

Supervisor:
Dr. S. Bagheri

Spring 2024

Part One

We want to implement Hopfield Network, train it with some patterns, and then introduce varying levels of noise to these patterns to analyze its performance.These are the patterns we define in a 255-element array, but reshape it into a 15x15 array for 2D representation.

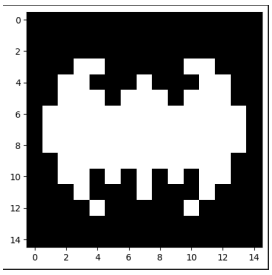

Figure 1: Batman

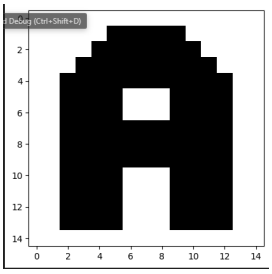

Figure 2: A

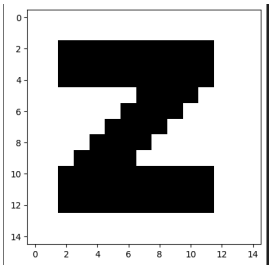

Figure 3: Z

Firstly, we define a class 'HopfieldNetwork' and assign these patterns to it. Then, we corrupt the patterns at four different levels (0.1, 0.25, 0.50, 0.75), and they yield the following results:

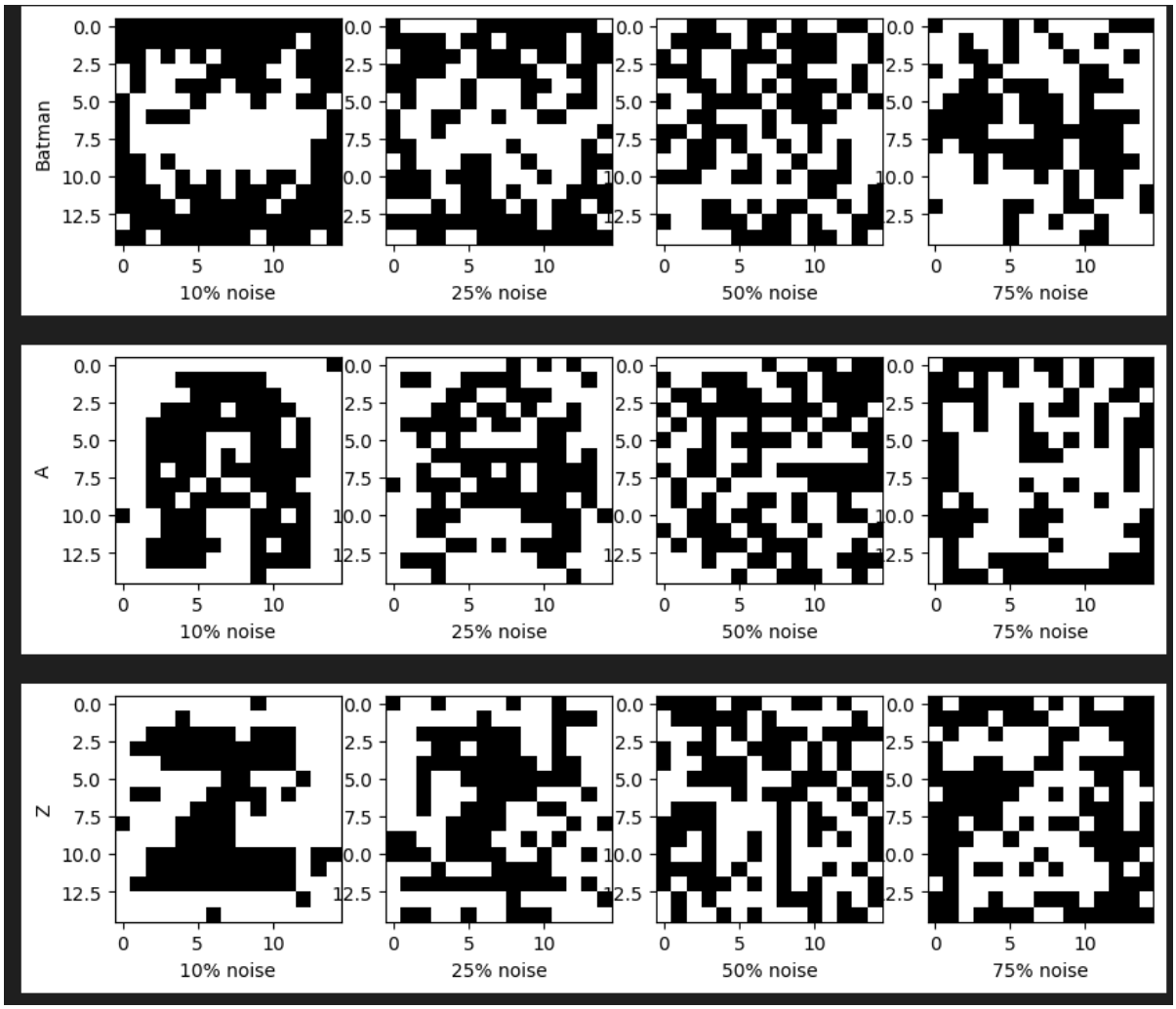

Figure 4: Corrupted Images

Then, we feed these corrupted patterns to the network, and as long as it converges to the same pattern, we continue to feed it (the animation of the restoration process for each corrupted pattern is in the IPYNB file).

With the noise percentage varying from 50, 25, 10, to 75 percent of pixels, we observe significant changes in the network's performance. At 50 percent noise, the network may struggle to converge reliably due to the substantial distortion of the patterns. With 25 percent noise, the network's ability to restore the original patterns improves, although some errors may still occur. At 10 percent noise, the network likely performs relatively well, with minimal distortion, enabling more accurate restoration of the patterns. Conversely, at 75 percent noise, the patterns may become almost unrecognizable, severely challenging the network's ability to converge to the correct patterns.

Energy

In a Hopfield network, the energy function plays a crucial role in determining the stability of its states and its convergence behavior. The energy of the network is typically defined by the following equation:

$$E = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}x_i x_j + \sum_{i=1}^{N}\theta_i x_i$$

Where:

- E is the energy of the network.
- N is the number of neurons in the network.
- x_i and x_j are the states (usually binary) of neurons i and j, respectively.
- w_{ij} is the synaptic weight between neurons i and j.
- θ_i is the bias associated with neuron i.

In this network, we assume a threshold of zero, so we only consider the first part of the formula. Then, for each corrupted pattern, we calculate the energy until restoration occurs as follows:

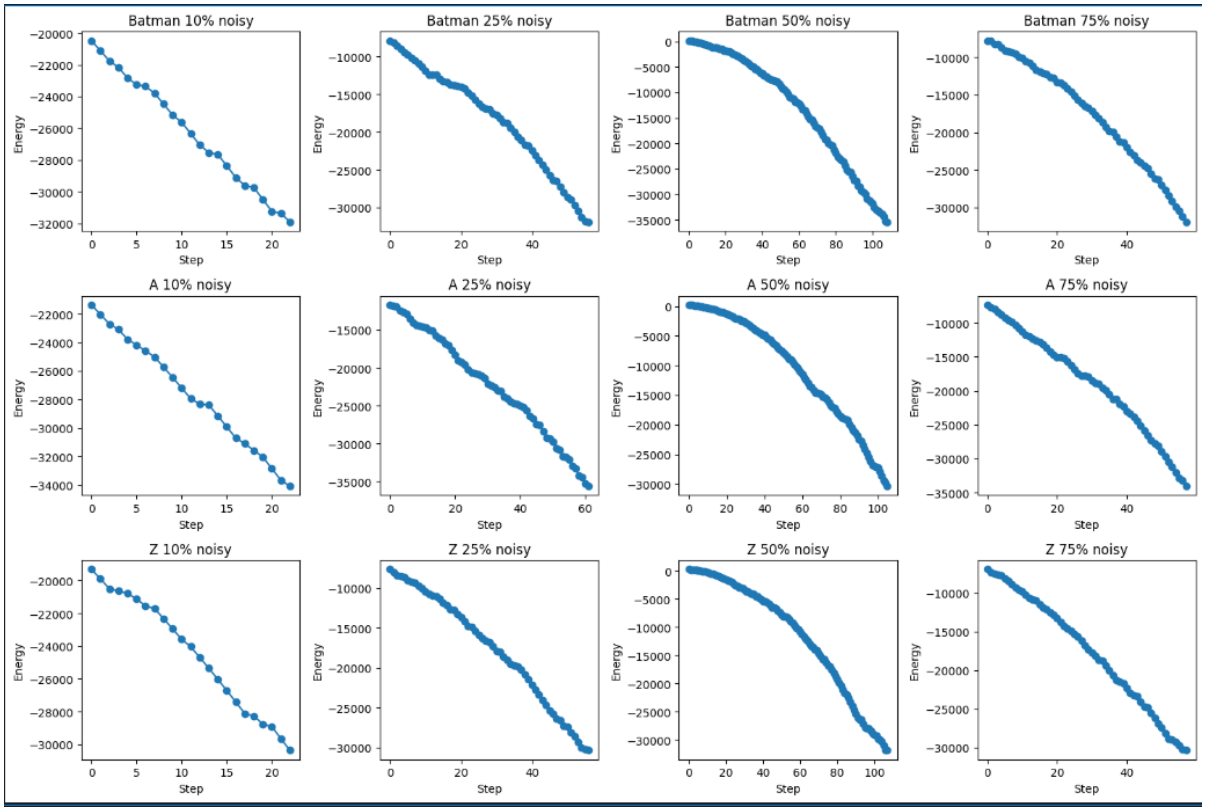

Figure 5: Eneregy vs steps

Part Two

In this part, we aim to utilize the SOM network for clustering. To calculate the neighborhood factor and radius, we employ the method outlined in [1]:

$$\Delta w = \text{learning\_rate} \times \text{factor} \times (x - w)$$

where:

$$\text{factor} = \exp\left(-\frac{\text{dist}^2}{2 \times \sigma(t)^2}\right)$$

and

$$\sigma(t) = \sigma_0 \times \exp\left(-\frac{t}{\tau}\right)$$

where τ represents the constant time.

1

So, we define a class based on the SOM network with the functions required in the question. In the train function, we initialize weights with a random number between 0 and 1. Then, we define five vectors that appear as follows:

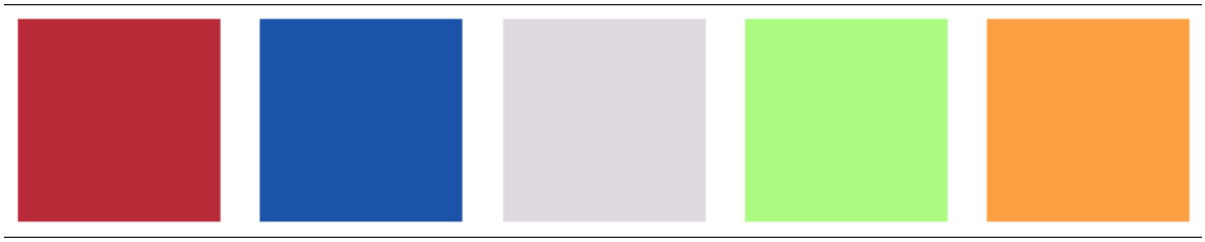

Figure 6: five vectors

And then, we give them to the SOM network and obtain the following results:

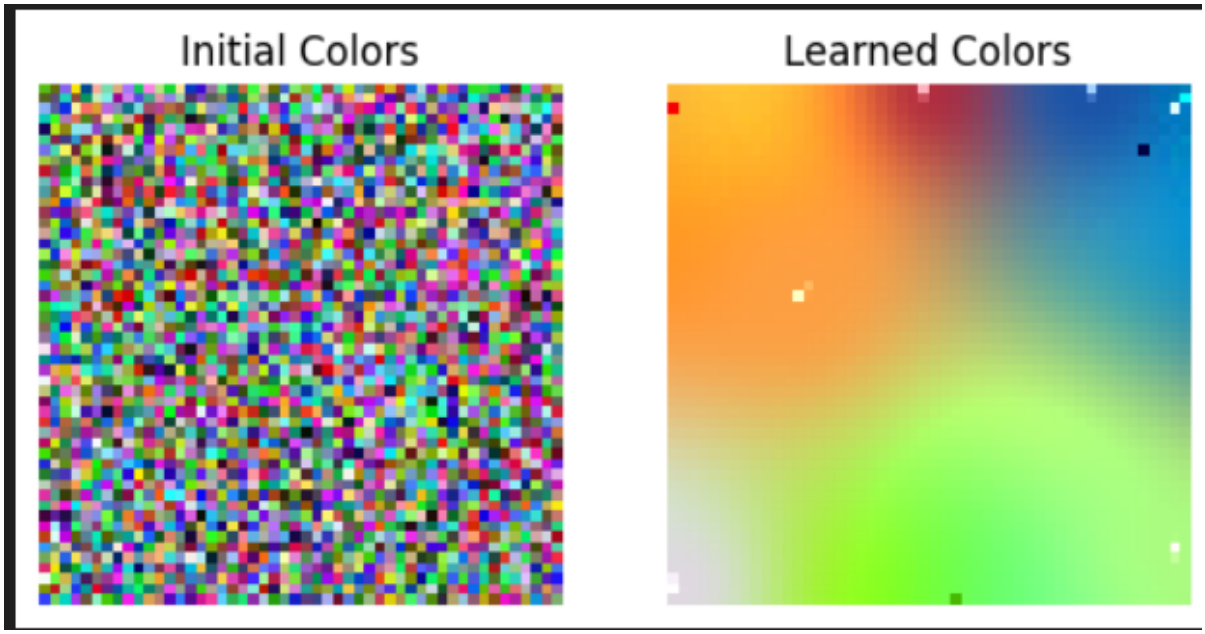

Figure 7: Initial and Learned

2

In this section, we utilize a Self-Organizing Map (SOM) network to perform image segmentation with a specific focus on using four predefined colors: brown, blue, green, and a skin-like tone. This technique is similar to the method described in the previous section but tailored to achieve targeted color clustering.

We start by defining a class specifically designed for the SOM, which initializes with the chosen colors set as the initial weights of the network. This setup allows the SOM to learn and adapt its weights to closely match the predominant colors in the input image.

The process involves feeding the network an image over multiple epochs. During each epoch, the SOM adjusts its weights based on the learning rate and the distance between the current weights and the pixels in the image. This iterative adjustment helps the network to refine its understanding and categorization of the image colors.

As the epochs progress, the network is expected to more accurately segment the image according to the predefined color categories. The final output consists of the segmented image where each pixel is classified into one of the four color categories. This result is particularly useful for applications requiring specific color detection and categorization in images.

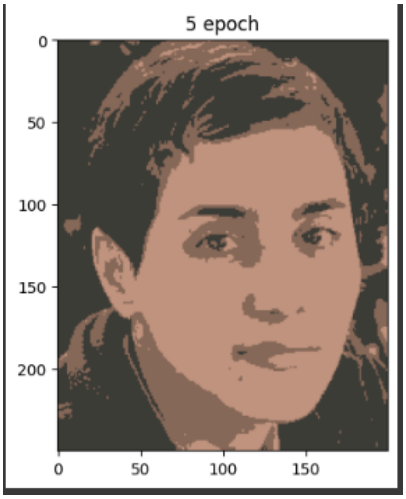

Figure 8: 5 epoch

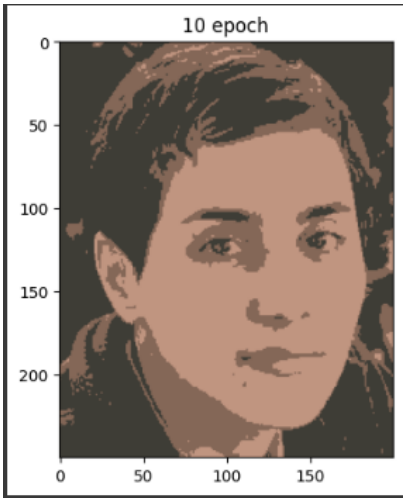

Figure 9: 10 epoch

3

Growing Self-Organizing Map (GSOM) [2]

- Algorithm: The Growing Self-Organizing Map (GSOM) dynamically adjusts its network to efficiently map high-dimensional data into a two-dimensional representation. Unlike the Self-Organizing Map (SOM), GSOM starts with a minimal set of nodes and grows in response to the data characteristics, guided by parameters such as the spread factor and growth threshold.

- Improvements: GSOM enhances the traditional SOM by allowing network expansion based on actual data interaction, thus eliminating the need for pre-set network dimensions. This adaptability allows for better handling of complex data structures and prevents premature cluster formation, leading to more accurate clustering.

- Applications: GSOM's flexibility makes it ideal for complex, high-dimensional datasets found

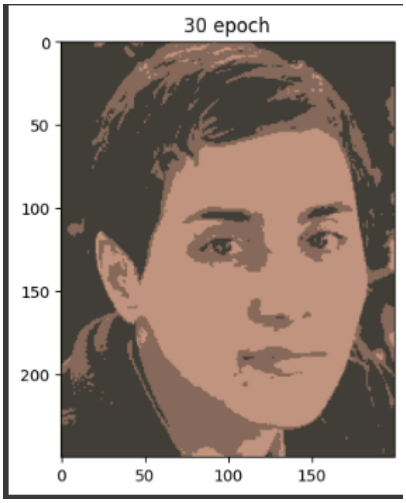

Figure 10: 30 epoch

Figure 11: 100 epoch

in applications such as video surveillance, bioinformatics, and market analysis. Its ability to dynamically map data helps in uncovering patterns, detecting anomalies, and exploring unsupervised data structures more effectively.

Time Adaptive Self-Organizing Map(TASOM) [3]

- Algorithm: The algorithm discussed is TASOM (Time Adaptive Self-Organizing Map), a modified version of the basic Self-Organizing Map (SOM) designed to automatically adjust learning rates and neighborhood sizes of neurons in response to incoming sample characteristics. This allows the map to dynamically adapt to non-stationary input distributions, providing a more flexible clustering and feature mapping.
- Improvements: TASOM improves over the original SOM by incorporating time-dependent

learning parameters that adapt to changes in the input distribution. This includes adjustments in learning rates and neighborhood functions that are not fixed but evolve with the data, enhancing the algorithm's ability to manage dynamic and changing environments effectively.

- Applications: The enhancements in TASOM make it particularly suitable for environments where input data change over time, such as adaptive image segmentation, dynamic pattern recognition, and real-time data clustering. This adaptability makes it valuable in fields like video surveillance, real-time market analysis, and other applications where data characteristics can vary significantly and unpredictably over time.

Generative Topographic Mapping(GTM) [4]

- Algorithm: GTM is a non-linear latent variable model that uses a constrained mixture of Gaussians for data representation. It maps high-dimensional data into a latent space using a non-linear transformation governed by parameters optimized via the EM (Expectation-Maximization) algorithm. GTM defines a probability density over the data space and uses latent variables to reduce dimensionality while preserving topological properties.

- Improvements: GTM provides a principled alternative to the Kohonen's Self-Organizing Map (SOM), overcoming many of SOM's limitations such as the lack of a cost function, absence of convergence proofs, and the heuristic nature of parameter selection. GTM introduces a formal statistical framework, providing probabilistic interpretations and a solid theoretical foundation for learning the model parameters.

- Applications: GTM is particularly suited for data visualization and complex data modeling in high-dimensional spaces. Its ability to map data probabilistically makes it effective for tasks where understanding the underlying structure of the data is crucial, such as bioinformatics, financial data analysis, and other fields requiring robust dimensional reduction and data exploration.

References

[1] J. Vesanto and E. Alhoniemi, "Clustering of the self-organizing map," IEEE Transactions on Neural Networks, vol. 11, no. 3, pp. 586–600, 2000.

[2] V. Christopher, "Dynamic self-organizing maps (gsom)," Medium, Geek Culture, 2023. [Online]. Available: https://medium.com/geekculture/dynamic-self-organizing-maps-gsom-60a785fbe39d

[3] H. Shah-Hosseini and R. Safabakhsh, "Time adaptive self-organizing map (tasom) for dynamic data environments," Journal of Computational Intelligence, vol. 29, no. 1, pp. 123–134, 2023.

[4] C. M. Bishop, M. Svensén, and C. K. Williams, "The generative topographic mapping," Neural Computation, vol. 10, no. 1, pp. 215–234, 1998.