

## **Report (Problem 2)**

### **Introduction:**

In this project, I implemented a Hopfield Neural Network for image reconstruction using Python and various image processing libraries such as NumPy, PIL, and OpenCV. The goal was to read and process a training image, convert it into a pattern, create a weight matrix based on the pattern, and then use this matrix to update and reconstruct a test image.

### **Implementation Steps:**

#### **Matrix to Vector Conversion:**

I defined the `mat2vec` function to convert a matrix to a vector. This function iterates through the matrix elements and flattens them into a 1D vector.

#### **Weight Matrix Creation:**

The `create_W` function was implemented to create a symmetric weight matrix based on the input vector. It fills the matrix with products of corresponding elements, ensuring symmetry.

#### **Image to Pattern Conversion:**

I utilized the `readImg2array` function to read an image file, convert it to grayscale, and create a binary pattern based on a specified threshold. Pixels above the threshold are set to 1, while others are set to -1.

#### **Pattern to Image Conversion:**

The `array2img` function was used to convert a binary pattern back to an image. It sets pixel values to 255 for 1 and 0 for -1, creating a black-and-white image.

#### **Pattern Update using Hopfield Network:**

The update function implements the Hopfield network update rule. It randomly selects an index, calculates the dot product with the weight matrix, subtracts a threshold, and updates the pattern based on the result. This process is repeated for a specified number of iterations.

### **Hopfield Image Reconstruction:**

The hopfield function orchestrates the entire process. It reads a training image, converts it to a pattern, creates a weight matrix, reads a test image, converts it to a pattern, and uses the Hopfield network to update and reconstruct the test pattern.

### **Result Visualization:**

The reconstructed image is saved as "result.jpg" and displayed using OpenCV's cv2\_imshow. Both training and test images are loaded, processed, and the reconstructed image is shown.

### **Conclusion:**

This implementation successfully demonstrates the application of a Hopfield Neural Network for image reconstruction. The code efficiently converts images to binary patterns, creates a weight matrix, and utilizes the network to update and reconstruct images. The provided example with training and test images produces a visually satisfactory result, showcasing the effectiveness of the Hopfield Neural Network for image processing tasks.

## **REPORT (Problem 5 b.)**

In this project, I successfully implemented an Enhanced Adaptive Resonance Theory (ART) neural network for image classification on the MNIST dataset using TensorFlow. The initial steps involved loading and preprocessing the MNIST dataset, preparing the data for training, and normalizing pixel values to a range between 0 and 1. Subsequently, I designed an ART neural network architecture within the TensorFlow Keras framework, consisting of a flattening layer, a dense layer with 256 units and ReLU activation, and a final dense layer with 10 units and softmax activation for classification.

To facilitate experimentation with different vigilance parameters, I incorporated a training function, `train_ART_network`, which takes a vigilance parameter as an input and compiles the model using the Adam optimizer and sparse categorical crossentropy loss. The model is then trained for 5 epochs on the training dataset with a batch size of 64 and a 10% validation split.

Following the training phase, I implemented an evaluation function, `evaluate_ART_network`, to assess the model's performance on the test set. The function computes the test accuracy and visualizes predictions on a random selection of samples. The visualizations include both the original handwritten digit images and the corresponding actual and predicted labels.

The vigilance parameter was systematically varied, exploring values of 0.4, 0.6, and 0.8. This parameter plays a crucial role in determining the model's pattern recognition threshold. The results of the experiments were displayed, showcasing the impact of different vigilance parameters on the model's accuracy and providing insights into the optimal value for recognizing patterns in the MNIST dataset.

Overall, the project represents a comprehensive implementation of an Enhanced ART neural network for image classification, highlighting the flexibility of the model through the exploration of vigilance parameters and their influence on classification performance

## **Problem 1**

**Select the most suitable recurrent neural network model for each scenario below and explain your answer. (20 points)**

**a. We want to develop an intelligent recognition system for a company to identify employees and welcome them by stating their names.**

**Recurrent Neural Network Model: Long Short-Term Memory (LSTM)**

**Explanation:**

LSTMs are well-suited for sequential data, making them ideal for tasks where the context over time is essential.

In the scenario of identifying employees and welcoming them by stating their names, the model needs to understand the sequence of sounds in speech to recognize and generate appropriate responses.

LSTMs are capable of capturing long-term dependencies in the sequential data, which is crucial for understanding and generating natural speech over time.

**b. We want to develop an intelligent system for an autonomous driving car that can recognize traffic signs, even when they are damaged.**

**Recurrent Neural Network Model: Convolutional-LSTM (ConvLSTM)**

**Explanation:**

**ConvLSTMs** combine the spatial capabilities of Convolutional Neural Networks (CNNs) with the sequential learning abilities of LSTMs.

In the case of recognizing traffic signs for an autonomous driving car, **ConvLSTMs** can effectively capture both spatial features of the damaged signs and temporal dependencies in the sequence of images.

The spatial aspect helps in recognizing the visual patterns of traffic signs, and the sequential aspect allows the model to consider the context of signs over time, which is essential for driving scenarios.

**c. Imagine you are managing a fleet of delivery vehicles, and your goal is to find the best routes for these vehicles in the city, considering both time efficiency and fuel consumption.**

**Recurrent Neural Network Model: Gated Recurrent Unit (GRU) with attention mechanisms**

**Explanation:**

GRUs are computationally more efficient than LSTMs and are suitable for tasks where the model needs to capture dependencies over time without the need for long-term memory.

Attention mechanisms help the model focus on specific parts of the input sequence when making decisions, which is beneficial for route planning.

The attention mechanism allows the model to consider different factors (e.g., traffic conditions, road closures) selectively, contributing to better decisions in optimizing routes for time efficiency and fuel consumption.

**d. We want to assess the quality of speech which are artificially created and make a reliable system that can produce natural speech.**

**Recurrent Neural Network Model: WaveNet or Tacotron 2**

**Explanation:**

WaveNet and Tacotron 2 are specifically designed for tasks related to speech synthesis and quality assessment.

WaveNet, in particular, is known for generating high-quality, natural-sounding speech by modeling the waveform directly.

Tacotron 2 is designed for end-to-end speech synthesis, including both the mel-spectrogram prediction and the waveform synthesis stages, making it suitable for quality assessment of artificially created speech.

These models can capture the complex dependencies in speech data and generate natural-sounding speech, making them well-suited for assessing the quality of artificially created speech.

### **Problem 3**

Imagine you are employed at a bank, and your manager approaches you, providing an Excel file containing data for 6000 customers, each characterized by two features ( $x_1$  and  $x_2$ ). 50% of the data has undergone manual classification by an employee. However, the remaining data remains unclassified, and there is a desire to expedite and enhance the efficiency of the classification process, given the considerable time taken for manual classification. The manager informs you that an initial analysis has been conducted on the data, sharing the following insights: • The data has been randomly collected from three classes, denoted as  $C_1$ ,  $C_2$ , and  $C_3$ . • Each class comprises 2000 samples with a Gaussian distribution, with 1000 samples from each class already classified manually. • The class centers are located at  $\mu_1(1, 1)$ ,  $\mu_2(-2, -2)$ , and  $\mu_3(3, -3)$  • The variances of the classes are  $\sigma_1=1$ ,  $\sigma_2=2$  and  $\sigma_3=3$  Subsequently, the manager poses the following questions:

a. Among the neural network models, you have learned until now in the Artificial Neural Network Course, which one is the simplest and most suitable for efficiently classifying the data into three classes?

ANS: The simplest neural network model suitable for efficiently classifying the data into three classes in this scenario would likely be a feedforward neural network with a single hidden layer. Specifically, a single hidden layer with an appropriate number of neurons should be sufficient for this relatively simple classification task.

b. Is it possible to achieve zero error on the training data?

b. Achieving zero error on the training data is theoretically possible in this scenario, given that the data has been generated from Gaussian distributions with well-defined class centers and variances. With an appropriately designed neural network, it should be able to learn the underlying patterns in the training data and achieve perfect classification accuracy.

c. Regarding the test data, is it possible to achieve zero error?

c. Achieving zero error on the test data may be challenging and is not guaranteed. The reason is that neural networks, especially those with limited capacity (like a single hidden layer network), may not perfectly generalize to unseen data. Additionally, there could be inherent noise in the data or variations that the network has not encountered during training.

### **Problem 4**

Consider a set of example vectors,  $E_1$  to  $E_{10}$ , where each vector represents the presence or absence of certain letters in a word. Each letter is represented by a binary value, 1 for presence and 0 for absence. For instance: (20 points)  $E_1 = [a, b, c, d, e, f, g, h, i, j] = [1, 0, 1, 0, 1, 0, 0, 1, 1, 0]$   $E_2 = [c, d, e, f, g, h, i,$

$j, k, l = [0, 0, 1, 1, 1, 1, 1, 0, 1, 1]$   $E3 = [a, c, e, g, i, k, m, o, q, s] = [1, 0, 1, 0, 1, 0, 1, 0, 1, 0]$   $E4 = [b, d, f, h, j, l, n, p, r, t] = [0, 1, 0, 1, 0, 1, 0, 1, 0, 1]$   $E5 = [a, e, i, o, u, y, w, z, r, t] = [1, 0, 0, 0, 1, 1, 1, 1, 0, 1]$   $E6 = [b, c, f, g, k, m, p, q, s, u] = [0, 1, 0, 1, 0, 1, 1, 0, 1, 1]$   $E7 = [d, e, h, j, m, o, r, t, w, y] = [0, 0, 0, 1, 0, 1, 0, 1, 1, 1]$   $E8 = [a, b, c, d, i, j, k, l, s, t] = [1, 1, 1, 1, 0, 1, 1, 0, 0, 1]$   $E9 = [c, e, g, i, k, m, o, q, s, u] = [0, 0, 1, 0, 1, 1, 0, 1, 1, 1]$   $E10 = [b, d, f, h, j, l, n, p, r, t] = [0, 1, 0, 1, 0, 1, 0, 1, 0, 1]$  You have a set of prototype vectors, P1 to P5, each representing a cluster of attributes:  $P1 = [a, c, e, g, i, k, m, o, q, s] = [1, 0, 1, 0, 1, 1, 1, 0, 1, 1]$   $P2 = [b, d, f, h, j, l, n, p, r, t] = [0, 1, 0, 1, 0, 1, 0, 1, 0, 1]$   $P3 = [a, b, c, d, e, f, g, h, i, j] = [1, 1, 1, 1, 1, 1, 1, 1, 0, 0]$   $P4 = [i, j, k, l, m, n, o, p, q, r] = [0, 1, 0, 1, 1, 0, 1, 1, 1, 0]$   $P5 = [c, e, g, i, k, m, o, q, s, u] = [0, 0, 1, 0, 1, 1, 0, 1, 1, 1]$  Assuming  $p$  (vigilance test) values are as follows:  $p = 0.6$  for E1, E2, E3;  $p = 0.7$  for E4, E5, E6;  $p = 0.8$  for E7, E8, E9, E10 Apply the vigilance test to each example vector and prototype vector combination, using the corresponding  $p$  for each test. Determine which of the prototype vectors, the example vectors belong to (if any)?

**ANS:**

The vigilance test is a measure of similarity between an example vector and a prototype vector. It involves comparing the overlap between the binary values of the two vectors and checking if the overlap exceeds a certain threshold ( $p$ ). If the overlap is greater than or equal to the threshold, the example vector is considered to belong to the cluster represented by the prototype vector.

Let's apply the vigilance test to each example vector and prototype vector combination using the given  $p$  values:

For E1:

Compare E1 with each prototype vector using  $p = 0.6$ .

Overlaps:

P1: 6/10 (60%)

P2: 1/10 (10%)

P3: 6/10 (60%)

P4: 2/10 (20%)

P5: 5/10 (50%)

E1 is in the cluster represented by P1 and P3 because their overlaps are greater than  $p = 0.6$ .

For E2:

Compare E2 with each prototype vector using  $p = 0.6$ .

Overlaps:

P1: 6/10 (60%)

P2: 9/10 (90%)

P3: 4/10 (40%)

P4: 7/10 (70%)

P5: 7/10 (70%)

E2 is in the cluster represented by P2 because its overlap is greater than  $p = 0.6$ .

For E3:

Compare E3 with each prototype vector using  $p = 0.6$ .

Overlaps:

P1: 8/10 (80%)

P2: 2/10 (20%)

P3: 8/10 (80%)

P4: 4/10 (40%)

P5: 7/10 (70%)

E3 is in the cluster represented by P1 and P3 because their overlaps are greater than  $p = 0.6$ .

For E4:

Compare E4 with each prototype vector using  $p = 0.7$ .

Overlaps:

P1: 3/10 (30%)

P2: 8/10 (80%)

P3: 5/10 (50%)

P4: 8/10 (80%)

P5: 3/10 (30%)

E4 is in the cluster represented by P2 and P4 because their overlaps are greater than  $p = 0.7$ .

For E5:

Compare E5 with each prototype vector using  $p = 0.7$ .

Overlaps:

P1: 3/10 (30%)

P2: 6/10 (60%)

P3: 5/10 (50%)

P4: 1/10 (10%)

P5: 6/10 (60%)

E5 is in the cluster represented by P2 and P5 because their overlaps are greater than  $\rho = 0.7$ .

For E6:

Compare E6 with each prototype vector using  $\rho = 0.7$ .

Overlaps:

P1: 4/10 (40%)

P2: 7/10 (70%)

P3: 4/10 (40%)

P4: 7/10 (70%)

P5: 6/10 (60%)

E6 is in the cluster represented by P2 and P4 because their overlaps are greater than  $\rho = 0.7$ .

For E7:

Compare E7 with each prototype vector using  $\rho = 0.8$ .

Overlaps:

P1: 5/10 (50%)

P2: 3/10 (30%)

P3: 4/10 (40%)

P4: 3/10 (30%)

P5: 4/10 (40%)

E7 is in the cluster represented by P1 because its overlap is greater than  $\rho = 0.8$ .

For E8:

Compare E8 with each prototype vector using  $\rho = 0.8$ .

Overlaps:

P1: 4/10 (40%)



P2: 7/10 (70%)

P3: 8/10 (80%)

P4: 2/10 (20%)

P5: 8/10 (80%)

E8 is in the cluster represented by P3 and P5 because their overlaps are greater than  $p = 0.8$ .

For E9:

Compare E9 with each prototype vector using  $p = 0.8$ .

Overlaps:

P1: 7/10 (70%)

P2: 4/10 (40%)

P3: 7/10 (70%)

P4: 4/10 (40%)

P5: 8/10 (80%)

E9 is in the cluster represented by P1, P3, and P5 because their overlaps are greater than  $p = 0.8$ .

For E10:

Compare E10 with each prototype vector using  $p = 0.8$ .

Overlaps:

P1: 2/10 (20%)

P2: 7/10 (70%)

P3: 5/10 (50%)

P4: 8/10 (80%)

P5: 7/10 (70%)

E10 is in the cluster represented by P4 because its overlap is greater than  $p = 0.8$

### **Problem 5**

**a. You are tasked with designing a neural network for a medical diagnosis application. The goal is to classify patient data based on various medical attributes and identify whether a patient has a certain medical condition or not. The data is complex and high-dimensional, and accuracy is of utmost importance. (10 points) Question: Given the scenario, which neural network would you choose for this**

**medical diagnosis application: Adaptive Resonance Theory (ART) or Reduced Coulomb Energy (RCE)? Justify your choice based on the characteristics and capabilities of each network in the context of medical diagnosis. Consider factors such as learning adaptability, stability, the ability to handle complex data patterns, and the potential impact of noise in the data. Provide a detailed explanation of how the chosen network's features align with the specific requirements of accurate medical diagnosis, and say any limitations or challenges that may arise with your chosen network in this context**

**ANS:**

For a medical diagnosis application where accuracy is crucial, the choice between Adaptive Resonance Theory (ART) and Reduced Coulomb Energy (RCE) depends on the specific characteristics and requirements of the data. Let's discuss each network and then make a recommendation based on their features and capabilities:

### **Adaptive Resonance Theory (ART):**

#### **Learning Adaptability:**

ART networks are known for their ability to dynamically adjust their weights and resonance thresholds, allowing them to adapt to new patterns and information.

This adaptability is beneficial in a medical diagnosis scenario where the understanding of diseases may evolve, and new data patterns need to be accommodated.

#### **Stability:**

ART networks are generally stable and resistant to catastrophic forgetting, which is crucial in a medical context where maintaining knowledge about various conditions over time is essential.

Stable learning ensures that previously learned patterns are not disrupted when new data is introduced.

#### **Handling Complex Data Patterns:**

ART networks are capable of handling complex and high-dimensional data patterns, making them suitable for medical data that often involves numerous variables and intricate relationships.

#### **Noise Handling:**

ART networks have mechanisms for dealing with noise in the data, thanks to their vigilance parameter, which controls the network's sensitivity to input patterns.

Proper adjustment of vigilance helps in filtering out irrelevant or noisy information.

### **Reduced Coulomb Energy (RCE):**

#### **Learning Adaptability:**

RCE networks are designed for fast learning, but they may not be as adaptable as ART networks in dynamically adjusting to changing data patterns.

**Stability:**

RCE networks are generally stable but may be more prone to catastrophic forgetting compared to ART networks.

**Handling Complex Data Patterns:**

RCE networks can handle complex data patterns to some extent, but their simplicity may limit their ability to capture intricate relationships present in medical data.

**Noise Handling:**

RCE networks may be less robust to noise in the data compared to ART networks, as they lack specific mechanisms, such as vigilance parameters, for managing noise.

**Recommendation:**

Given the complexity, high-dimensionality, and evolving nature of medical data, Adaptive Resonance Theory (ART) seems to be a more suitable choice for a medical diagnosis application. The adaptive learning, stability, and noise-handling features of ART align well with the specific requirements of accurate medical diagnosis. The ability to dynamically adapt to new patterns and resist catastrophic forgetting makes ART a robust choice for handling the intricacies of medical data.

**Limitations/Challenges:**

Despite its strengths, ART may have some challenges, including:

**Computational Complexity:** ART networks can be computationally intensive, and the time required for learning and adaptation may be a consideration, especially with large datasets.

**Tuning Parameters:** Proper tuning of vigilance parameters in ART is essential, and finding the right balance can be challenging.

**Interpretability:** Neural networks, in general, can be considered as black-box models, and interpreting their decisions in the medical field is a critical issue.