**Warsaw University of Technology**

Faculty of mathematics and

Information Sciences

**First semester 2021/2022**

**Neural network**

**Project 3**

**By**

Swetha Suresh Kumar (324353)

Andra Umoru (324334)

**Submitted To:**

Stanislaw Kazmierczak

# Table of Contents

[Table of Contents 2](#_Toc103083053)

[1. List of Figures 3](#_Toc103083054)

[2. Problem description 4](#_Toc103083055)

[3. Theoretical Description 4](#_Toc103083056)

[2.1 Adaptive Resonance Theory (ART) 4](#_Toc103083057)

[2.2 Advantage of Adaptive Resonance Theory (ART) 5](#_Toc103083058)

[4. Algorithm for ART 6](#_Toc103083059)

[5. Implementation 6](#_Toc103083060)

[5.1. Requirements 6](#_Toc103083061)

[5.2. Implementation of the algorithm 7](#_Toc103083062)

[6. Results 10](#_Toc103083063)

[7. Conclusion 12](#_Toc103083064)

[8. Reference 12](#_Toc103083065)

# List of Figures

[Figure 1: ART for binary Input 5](file:///C:\Users\sweth\Downloads\Project%203%20-%20Report.docx#_Toc103075420)

[Figure 2: Algorithm for ART Architecture 6](#_Toc103075421)

[Figure 3: Conversation of image to binary 7](file:///C:\Users\sweth\Downloads\Project%203%20-%20Report.docx#_Toc103075422)

[Figure 4: Primary functions 8](file:///C:\Users\sweth\Downloads\Project%203%20-%20Report.docx#_Toc103075423)

[Figure 5: primary functions 8](file:///C:\Users\sweth\Downloads\Project%203%20-%20Report.docx#_Toc103075424)

[Figure 6: Testing 9](file:///C:\Users\sweth\Downloads\Project%203%20-%20Report.docx#_Toc103075425)

[Figure 7: plot 9](file:///C:\Users\sweth\Downloads\Project%203%20-%20Report.docx#_Toc103075426)

[Figure 8: Data preparation and division 10](#_Toc103075427)

[Figure 9: Update of top-down weight and bottom-up weight 10](file:///C:\Users\sweth\Downloads\Project%203%20-%20Report.docx#_Toc103075428)

[Figure 10: Contingency matrix 10](file:///C:\Users\sweth\Downloads\Project%203%20-%20Report.docx#_Toc103075429)

[Figure 11: Testing 11](file:///C:\Users\sweth\Downloads\Project%203%20-%20Report.docx#_Toc103075430)

[Figure 12: Plot 11](file:///C:\Users\sweth\Downloads\Project%203%20-%20Report.docx#_Toc103075431)

# Problem description

Implementation of the Adaptive Resonance Theory for MNIST dataset.

* Compare different vigilance values
* Compare clusters with MNIST classes.

# Theoretical Description

## 2.1 Adaptive Resonance Theory (ART)

Adaptive Resonance Theory (ART) is a cognitive and neurological theory that explains how the brain learns to attention to, categorize, recognize, and predict things and events in a changing environment on its own. ART presently has the most comprehensive set of cognitive and neurological theories for explanation and prediction. The ability of ART to autonomously carry out quick, incremental, unsupervised and supervised learning in response to a changing reality, without destroying previously learned memories, is essential to its predictive capability.

The stability-plasticity issue of a system is addressed by the Adaptive Resonance Theory, which asks how learning may progress in response to large input patterns while maintaining stability for irrelevant patterns. Aside from that, the stability-elasticity paradox is concerned with how a system can adapt to new data while maintaining previous knowledge. A feedback mechanism is added among the ART neural network layers for such a task. The data in the form of processing element output reflects back and forth across layers in this neural network. Adaption can occur during this period if an adequate pattern is built up and the resonance is reached.

**Stability**: The stability of the ART architecture connotes those irrelevant events have no effect on system behavior.

**Plasticity:** The system adapts its behavior in response to important occurrences.

**Dilemma:** Thedilemmaasks the following:

* How to achieve stability while avoiding rigidity and chaos.
* Continuous learning ability.
* Learning knowledge is preserved.

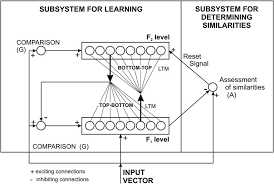


Figure 1: ART for binary Input

## ****2.2 Advantage of Adaptive Resonance Theory (ART)****

1. It is stable and unaffected by a wide range of inputs to its network.
2. It can be combined with a variety of different approaches to produce even better outcomes.
3. It can be utilized in a variety of applications, including mobile robot control, face identification, land cover classification, target recognition, medical diagnosis, signature verification, and web user clustering, among others.
4. It offers benefits over competitive learning. Competitive learning does not have the capacity to add additional clusters as needed.
5. It does not ensure cluster formation stability.

# Algorithm for ART

The figure below as captured from the class lectures slides shows how the ART problem is solved:

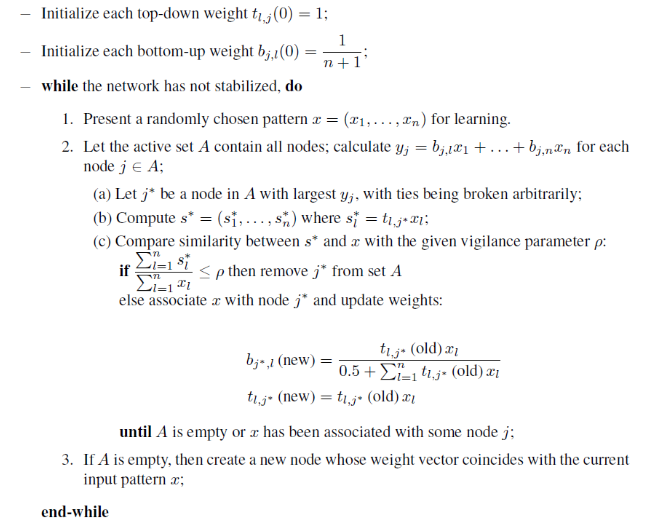


Figure 2: Algorithm for ART Architecture

# Implementation

This implementation was carried out using python programming language. The dataset used for this implementation is MNIST dataset. This dataset was converted from the image dataset to binary form. The implementation was carried using the algorithm above.

## Requirements

**2.1 Requirements**

The application is implemented using python 3. The libraries used are:

1. Numpy

2. Pandas

3. Matplotlib.pyplot

4. Random

5. Keras.datasets

6. Sklearn.metrics.cluster

## Implementation of the algorithm

* + 1. Data preparation and division into training and testing dataset

We have converted the MNIST dataset image to binary. And divided it into training and testing dataset (60.000 / 10.000 respectively)

Text

Description automatically generated with medium confidence

Figure 3: Conversation of image to binary

* + 1. Primary functions
* compare\_value: This function is used to calculate a value which is used to compare with vigilance value
* vigil\_cond: We used this function to check whether the vigilance is satisfied or not, which will return true if satisfied and false if not satisfied
* update\_B: It is used to update the bottom-up weight
* update\_T: Used to update the top-down weight

Text

Description automatically generatedText

Description automatically generated

Figure 4: Primary functions

Figure 5: primary functions

* + 1. Testing

We have used the following input vectors for testing purposes.

1. (1,1,0,0,0,0,1)

2. (0,0,1,1,1,1,0)

3. (1,0,1,1,1,1,0)

4. (0,0,0,1,1,1,0)

5. (1,1,0,1,1,1,0)

* Vigilance parameter – 0.7

Text

Description automatically generated

Figure 6: Testing

A picture containing text

Description automatically generated

Figure 7: plot

# Results

* 1. Data preparation and division into training and testing dataset

After training and testing the MNIST dataset, the following results were observed.

A picture containing diagram

Description automatically generated

Figure 8: Data preparation and division

* 1. Primary functions

Table

Description automatically generatedText

Description automatically generated with medium confidence

Figure 9: Update of top-down weight and bottom-up weight

Figure 10: Contingency matrix

* 1. Testing

Screenshot showing the expected result.

A picture containing diagram

Description automatically generated

Figure 11: Testing

Chart

Description automatically generated

Figure 12: Plot

# Conclusion

This project implements the Adaptive Resonance Theory for MNIST datasets. The MNIST datasets were fetched using the Keras Library function. After fetching the MNIST datasets, they were converted from image to binary using grey\_to\_bin function. In conclusion, the results presented above are the output of implementing the Adaptive Resonance Theory.

# Reference

* The neural network lecture slides
* <https://www.google.com/search?q=adaptive+resonance+theory&rlz=1C1GCEA_enNG992NG993&sxsrf=ALiCzsaTBT1vof6D77kAJMOko0JiSfgotQ:1652126903946&source=lnms&tbm=isch&sa=X&ved=2ahUKEwjm5u6pnNP3AhVmpIsKHVM7DUYQ_AUoAXoECAEQAw&biw=1366&bih=568&dpr=1>
* <http://www.scholarpedia.org/article/Adaptive_resonance_theory>
* <https://www.geeksforgeeks.org/adaptive-resonance-theory-art/>
* <https://www.javatpoint.com/artificial-neural-network-adaptive-resonance-theory>