Performing Classification of Hand-written digits using Restricted Boltzmann Machine, Self-Organizing Maps and PCA

# Abstract

Ability of digital systems to interpret handwritten sources like paper documents, photographs, touch-screens devices are critical for many business solutions. For this project, we use MNIST dataset to evaluate Restricted Boltzmann Machine (RBM), Self-Organizing Maps and PCA algorithms based on their accuracy with using different parameters.

# Introduction

Ability to recognize the hand-written text in context of digital applications is very popular topics due to many applications in the digital world such as recognizing and accepting handwritten paper documents, photos and touch screen devices writing for business purposes.

Processing handwritten data has many challenges such as the elements can be different size, orientation, can vary from person to person and similarity between the elements.

For this project, MNIST (Modified National Institute of Standards and Technology) handwritten digits were used to evaluate selected algorithms. There are total of 70000 28\*28 pixels grayscale images that split into 60,000 training set and 10,000 test set. Each image has label of 0 to 9.

RBM, SOM and PCA were selected to use the dataset and compare accuracy of the models against unseen data.

# Self-Organizing Maps (SOM)

## Algorithm

A self-organizing map(SOM) is a type of artificial neural network that is trained using unsupervised learning. SOM algorithm extracts important features and segregates these features, which produce a low-dimensional representation of the data. Self-organizing maps are differ from other neural network because the apply competitive learning as opposed to error-correction learning such as backpropagation with gradient decent.

## Methodology

For this experiment, we were interested in accuracy of the models based on multiple parameters but also the clustering of the digits on the network

Accuracy of the algorithm was evaluated based on multiple parameters:

* Dimension of SOM: 10x10 and 20x20
* Number of epochs: 5,10,15,20
* Size of the data(training/test): 10000/4000, 1000/400

## Results

Some of the results were very surprising. For example, there is a deep in accuracy of the algorithm when we run 10 or 15 epochs, but when we run 20 epochs the accuracy of predictions is very high.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Epoch | Accuracy | | | |
| (10x10) (10000/4000) | (20x20) (10000/4000) | (10x10) (1000/400) | (20x20) (1000/400) |
| 5 | 0.21375 | 0.10550 | 0.3350 | 0.3650 |
| 10 | 0.19475 | 0.04425 | 0.1750 | 0.0925 |
| 15 | 0.41275 | 0.06 | 0.2225 | 0.0375 |
| 20 | 0.83500 | 0.87600 | 0.8025 | 0.84 |

We observe that there is clustering behavior of the algorithm, but the location of the clusters can change based on number of epochs. Below is example of this behavior. The color meaning: **green**-100% confidence, blue- between 50-100% confidence, red- below 50% confidence. We can see that 1 is mapped to upper left corner after 5 epochs and 7 is mapped to upper right corner. By 10 epochs they switch places and by 20 epoch they end up on bottom of the matrix.

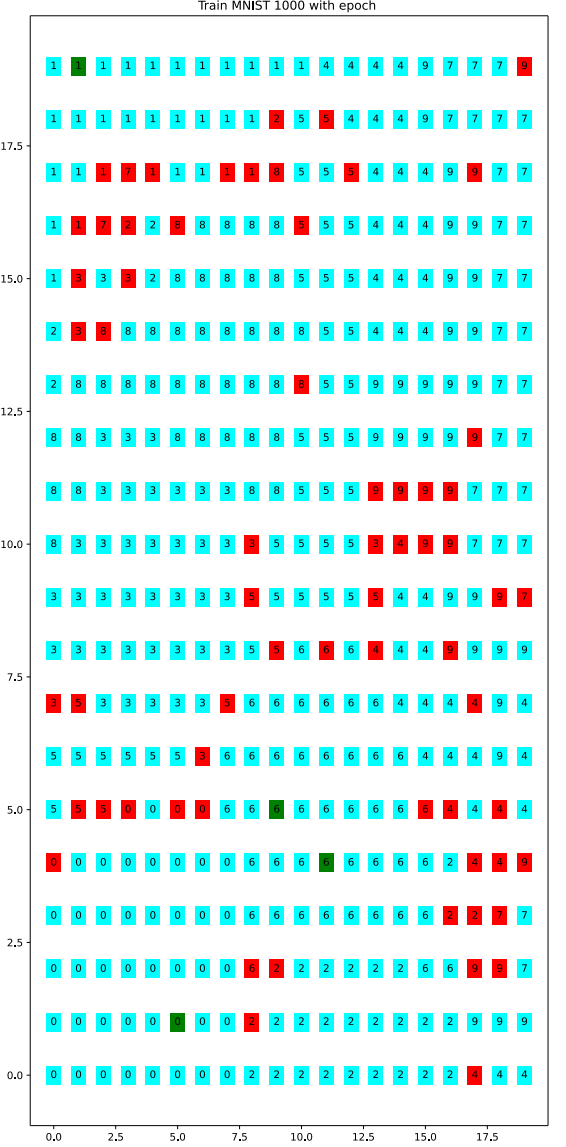


Figure 1 20x20 with training size:10000 and epoch:5

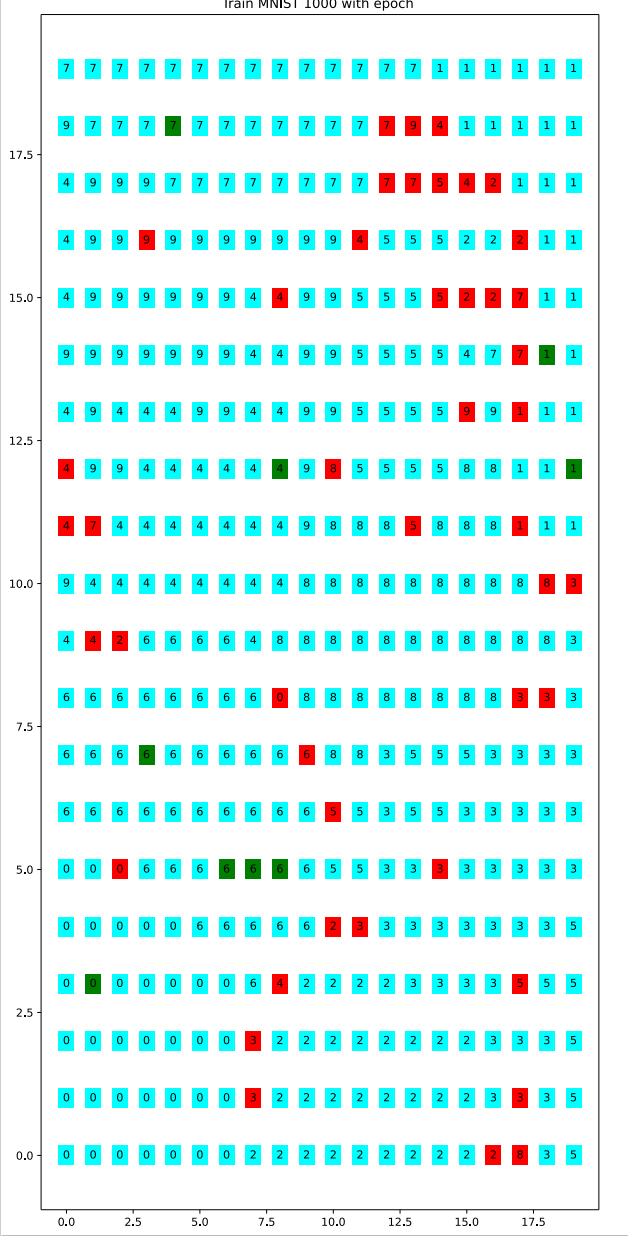


Figure 2 20x20 with training size:10000 and epoch:10

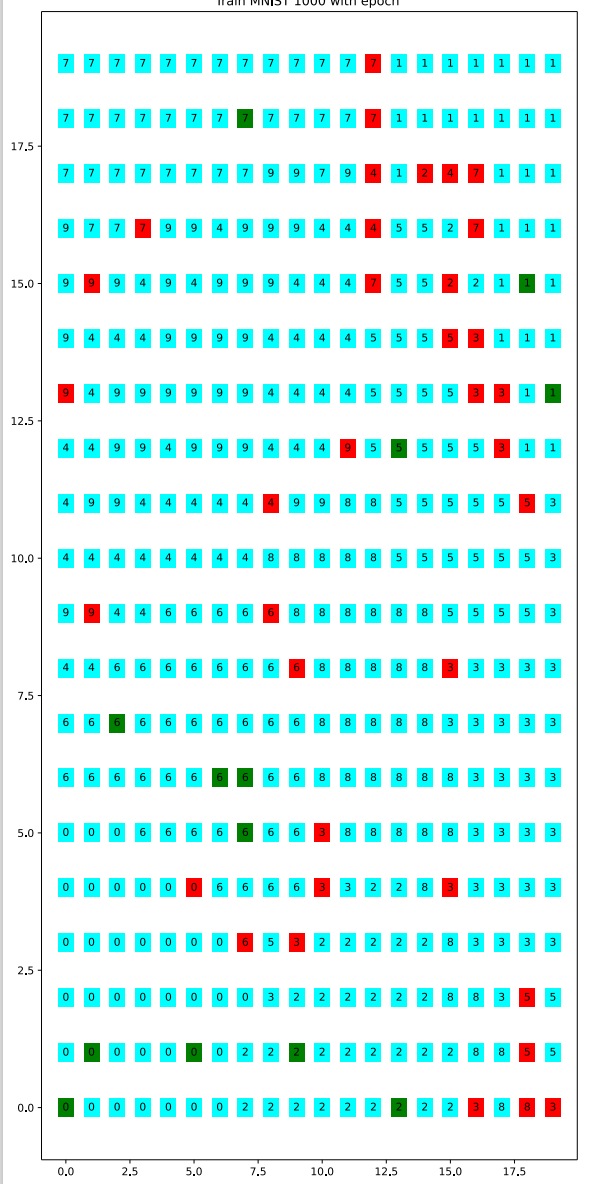


Figure 3 20x20 with training size:10000 and epoch:15

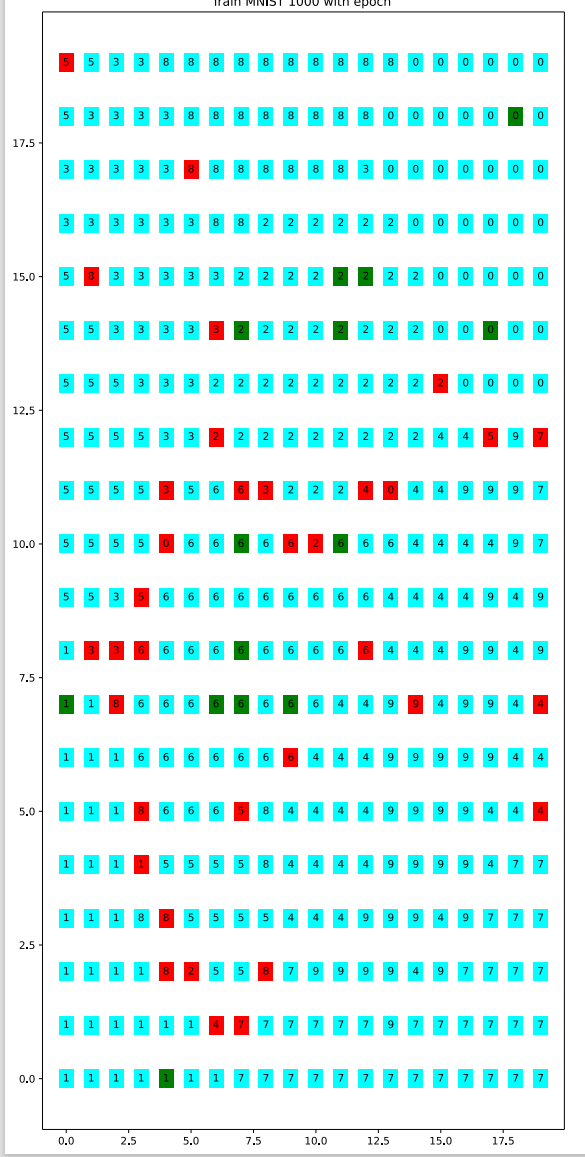


Figure 4 20x20 with training size:10000 and epoch:20