



ARTIFICIAL NEURAL NETWORK

Classifying Hand Written Roman Numerals through MLP



SUBMITTED TO: DR. JUNAID AKHTER

GROUP MEMBERS

SAFIA SULATANA - NIM-BSCS-2019-05

ALI MURTAZA – NIM-BSCS-2019-39

BACKGROUND

Artificial Neural Networks are computing systems inspired by neural network that exists in animal and human brain. ANN consists of collection of nodes connected with each which are called artificial neurons. They can send data to each other just as animal brain transmit signals to each other. A neuron receives signal then processes it and after that it transfers forwards it to another neuron. ANN does exactly the same, Neurons also called as **perceptron**, receive inputs, processes these inputs using some functions and then sends output to other perceptron.

PROBLEM STATEMENT

Before going into detail of ANN and our model structure. We think it is very important to first understand the problem and afterwards we would everything in light of this problem and hopefully this would help to understand things easier.

We are given dataset of hand written numerals and we have to make a model of Artificial Neural Network and then train that model using BackPropagation algorithm and the function of that model would be to correctly classify these Roman Numerals.

	I	II	III	
IV	V	VI	VII	VIII
	IX	X		

PROCESS

Under this section, we will discuss about process we followed to solve the given problem.

NATURAL REALITY:

The phenomenon of human learning starts when they are born. Infant babies interact with many different things.

THEORY OF LEARNING:

COMPUTATIONAL MODEL:

Experiment No.1:

Setting of Model and Features:

Used default setting and feature given was only black points in pictures. The main thought behind was that it should differentiate black and white points.

Result:

The result was not good. The accuracy was **0.01** and the reason was that since black points could be put anywhere in picture. So, one thing was certain, new features should be introduced.

Experiment #2:

Setting of Model and Features:

Setting was exactly the same as above but I changed features. This time give feature of count of both diagonals. The idea was that since X and IX lie on diagonal then 20 % of data should be correctly classified.

Result:

The result didn't change and it was some what similar to first experiment.

Experiment # 3:

Setting Changes:

We were using relu activation function previous experiments but now we changed it to logistic. Since logistic is sigmoid function but relu activates value between min and max so losses are much more using this activation function.

Result:

When changing the activation function then result increases. Then we understand that there is need of change of settings. But after changings in settings like using logistic function, increasing hidden layer size and random state equals to None. Even then result was not much increasing. It was 0.11 in case I put X_test for testing and 0.19 in case I check X_train as testing.

```
Iteration 357, loss = 0.32981784
Iteration 358, loss = 0.33780782
Iteration 359, loss = 0.33340252
Iteration 360, loss = 0.32921930
Iteration 361, loss = 0.32292115
Iteration 362, loss = 0.32262666
Iteration 363, loss = 0.32337083
. Stopping.
0.1968019680196802 SCORE
PS C:\Users\Manzar Computers\OneDrive\Desktop\Artificial Intelligence\ANN\data
```

Experiment 4:

Here we stuck, then we discuss our problem our TA, Mr. Ali Raza Khan. He advised us to use shuffle and hinted us about using center row as feature. Then we applied these changes.

Result:

The result was quite good since X_train when passed as testing data for model then accuracy was 78 % but even then X_test accuracy was not increasing.

```
Iteration 440, loss = 0.33520428
Iteration 441, loss = 0.34286833
Iteration 442, loss = 0.36608266
Iteration 443, loss = 0.38910708
Iteration 444, loss = 0.40715816
Iteration 445, loss = 0.48751980
Iteration 446, loss = 0.57411215
Iteration 447, loss = 0.70051297
Training loss did not improve more than tol=0.000100 for 10 consecutive epochs. Stopping
0.7895500725689405 SCORE
PS C:\Users\Manzar Computers\OneDrive\Desktop\Artificial Intelligence\ANN\data>
```

Experiment # 5:

We not only considered center row but also some rows above it and some row below it and from these rows we selected columns that are close center columns. And passed all these points as features.

Result:

Result was good than previous one. X_train accuracy was 92 % and X_test accuracy also improved and it goes to 23%. It is not much good but I has improved almost 10%.

179

PROBLEMS OUTPUT TERMINAL SQL CONSOLE DEBUG CONSOLE

```
Iteration 414, loss = 0.30172803
Iteration 415, loss = 0.29826136
Iteration 416, loss = 0.29917924
Iteration 417, loss = 0.30084226
Iteration 418, loss = 0.29940367
Iteration 419, loss = 0.29847171
Iteration 420, loss = 0.30062624
Iteration 421, loss = 0.29661445
Training loss did not improve more than tol=0.000100 for 10 consecutive epochs. Stopping.
0.9230769230769231 SCORE
PS C:\Users\Manzar Computers\OneDrive\Desktop\Artificial Intelligence\ANN\data> |
```

```
Iteration 398, loss = 0.29027036
Iteration 399, loss = 0.28837974
Iteration 400, loss = 0.28685127
C:\Users\Manzar Computers\AppData\Local\Programs\Python\Python39\lib\site-packages\sk
learn\normalization\_multilayer_perceptron.py:692: ConvergenceWarning: Stochastic Op
timizer: Maximum iterations (400) reached and the optimization hasn't converged yet.
  warnings.warn(
0.2091020910209102 SCORE
PS C:\Users\Manzar Computers\OneDrive\Desktop\Artificial Intelligence\ANN\data> |
```