Assignment-2 Classification of handwritten roman numbers

Artificial Neural Network

Artificial Intelligence

Namal University Mianwali

Submitted by: Danish khan, Muhammad Kashif

Roll No: BSCS-2019-50, BSCS-2019-61

Date: 12-12-2021

Submitted to: Dr. Junaid Akhtar

Table of Contents:

1. Acknowledgment:

We would like to express our sincere gratitude to our Course Instructor Dr. Junaid Akhatr, Teacher assistant Ali Raza Khan for supporting us throughout our Artificial Intelligence project which is the **Classification of handwritten roman numbers**. Further, we thank again to our project instructor Dr. Junaid Akhtar, for his motivation, helpful information, and practical advice from natural reality theorization that have helped us a lot at all times in our research and writing of this report classification project in Neural network.

2. Abstract:

In this project, our basic goal is to recognize multiple classes' handwritten roman numbers using an Artificial Neural Network(ANN). For that, we use the machine learning library Sklearn. There are many techniques but we are restricted to learning MLP. This project consists of designing an application for matching input and recognizing what number it is written using with an optimization approach based on darwin evolutionary theory. So as in the previous project “evolutionary algorithm” the main part of this project implementation is based on science natural theory which I follow in order rather than direct programming. Such as to understand basic human mind work, and how the human mind classifies two classes, after that, we implement the same single perceptron as in the human mind. So first we design the most simple neural network and it just separates two different classes. SLP can only be limited to linearly separable cases. While we have multiple classes of data so in this project we use a multi-layer perceptron (MLP). This means we have to use multiple perceptrons because the data is in multiclass and not possible to use a single perceptron.

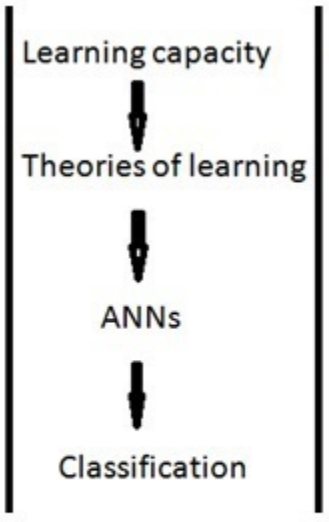


Image Understanding

3. Problem Statement:

The main problem is not just this project problem, but the real-world problems that are solved with this ANN. And the problem is how to train a machine to predict the things to which class something belongs just like human, when a child is born he can't classify or know something, but with the passage of his age and time from parents, surrounding, and the environment he tries to differentiate multiple things. So the child trains his mind for differentiation. Now anything which he remembers comes in front of him he can easily differentiate. This means with this neural network there are multiple real-world problems that have been solved i.e business gain loss prediction stock prediction, handwriting recognizing, etc.

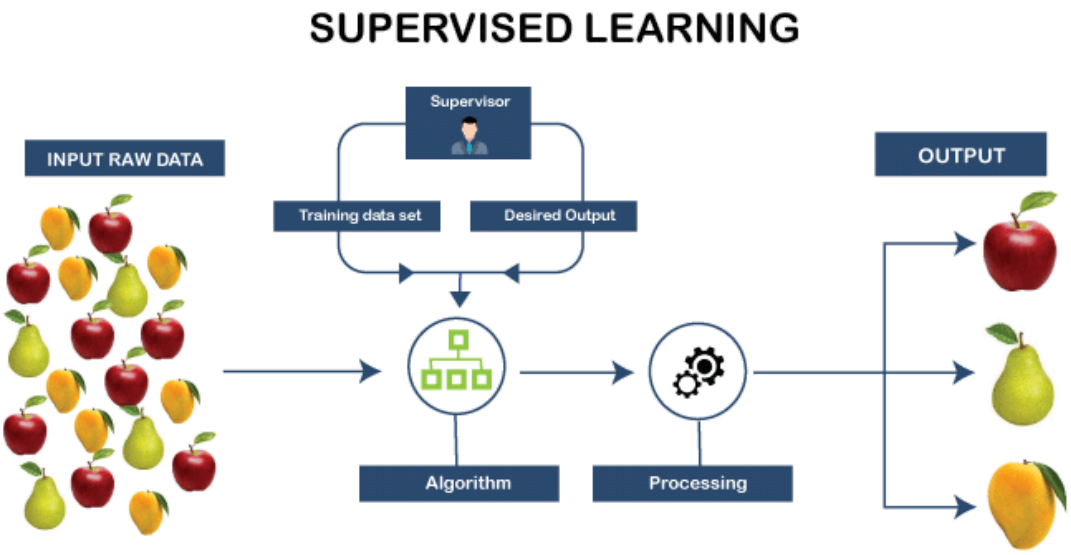
4. Purpose Statement:

The purpose of this Artificial neural network project is that we have to design a generalized application based on the Sklearn MLP algorithm, and this application will predict different handwritten roman numbers, i.e either it is from class i, ii,...x. So it's not possible to implement using a single layer perceptron for that we use MLP. All methods used in this algorithm are inspired by the natural theory of the human mind i.e single human mind perceptron, then for many classes classification just like human mind hidden layers are used in Multilayer neural network.

5. Introduction

In this project we have to train a machine and then we pass untrained data then the machine will predict whether data belongs to classes 1,2,3,4 up to 10. For that, we use a supervised classification approach.

A biological neural network is generally made up of a collection of chemically or functionally related neurons. A single neuron may be linked to a large number of other neurons, and in this way, a network in the human brain is made up. And the total work of this single is to generate a signal, While In Artificial intelligence, a neural network is a set of algorithms that attempts to recognize hidden patterns in a batch of data using a method that mimics how the human brain naturally operates. Neural networks, in this context, refer to a collection of neurons that are artificial. The background of this classification algorithm is based on the human brain working theory.

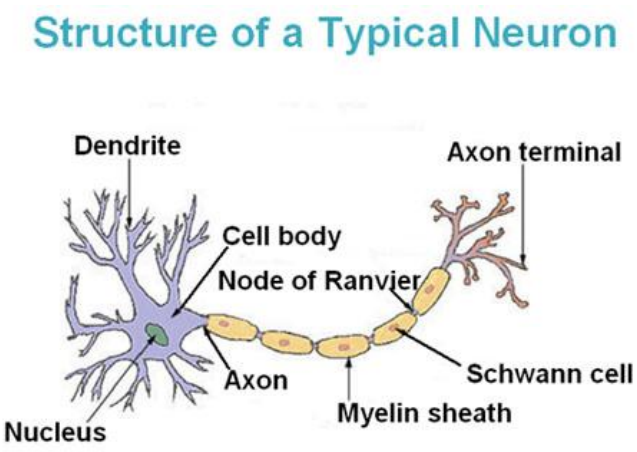


6. Machine Learning.

As the name suggests, Machine learning comes under the umbrella of Artificial intelligence that is basically based on prediction. In Machine learning the machine predicts almost accurate based results based on the data it experienced, not explicitly programmed for different inputs. Such as for a stock exchange one machine have to train on large data then the machine will predict the business for next month based on the data it experienced, classification, regression, etc.

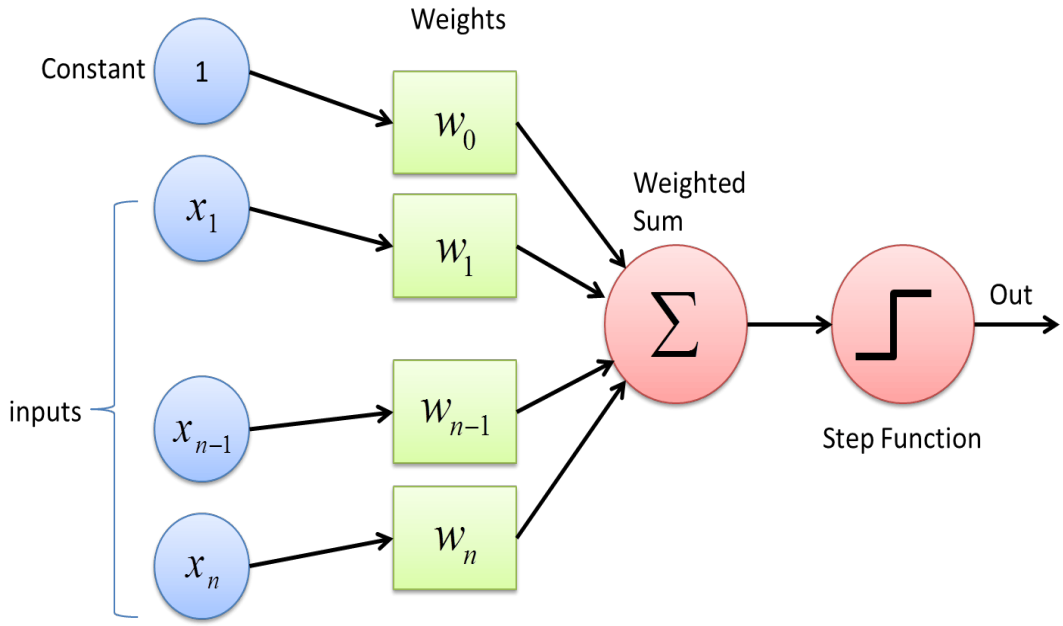
7. Biological Neural Network:

A Biological neuron in the human mind works as a messenger of transmission of data. Neurons do nothing except generate electrical signals and transmit them to neurons and other nervous systems. The basic features of the neuron are the soma (cell body), the axon (a long slender projection that conducts electrical impulses away from the cell body), dendrites (tree-like structures that receive messages from other neurons), and synapses (specialized junctions between neurons). And when billions of human brain neurons connect to each other and transmit information in the form of signals known as biological neural networks.



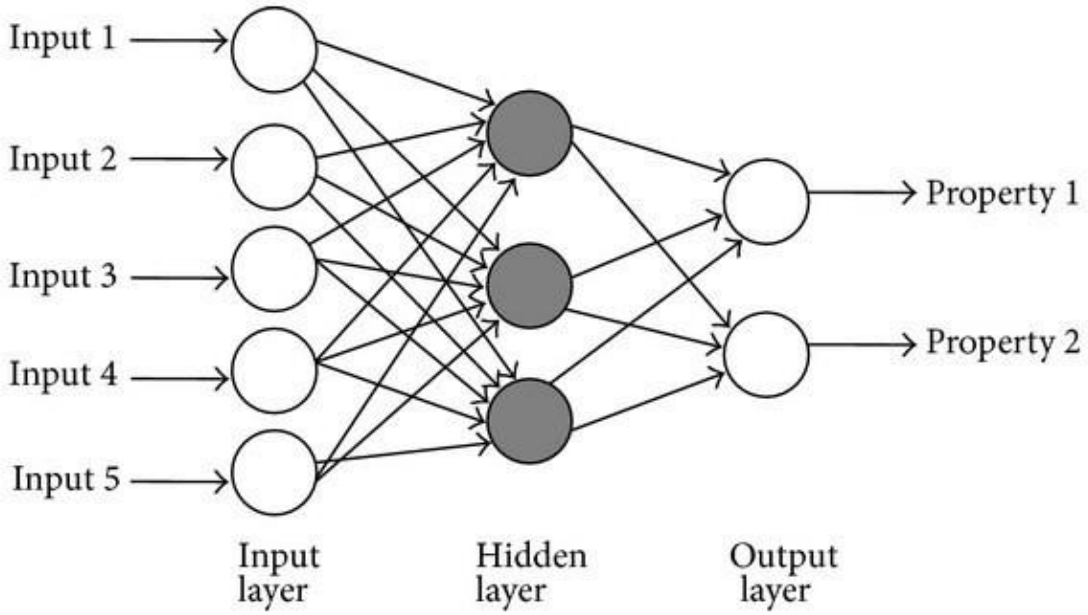
8. Artificial neuron or perceptron(SLP):

In simple words, an artificial neuron can be thought of as a function that takes some inputs and produces an output. It is the basic building block of ANN. It has some weights associated with each input and each input gets multiplied by its weights. Every neuron has a bias as well which translates the position of cuts made by the neuron. Each neuron's output i.e. summation of each input multiplied by its weight added with bias is passed through an activation function. And if the activation function is sigmoid then its value ranges from 0 to 1. Artificial neurons just like the human brain single perceptrons just generate an electrical signal.



9. Multilayer perceptron(MLP):

An artificial Neural network is a network of neurons, Multilayer perceptron is the collection of SLPs, which means a collection of neurons made of a multilayer perceptron that makes artificial neural networks. Why do we need a multilayer perceptron instead of an SLP? Because in SLP we can just use one neuron and one neuron can just separate two classes which are linearly separable, but in real-world problems, we have multiclass problems just as in our project we have ten classes and in every class, there are more than 200 forms of the same entity. So for that, we can't use SLP, because these are not linearly separable we have to add multi neurons and hidden layers for classifying multiclass problems.



10. Neural Network architecture.

To understand the problem, you need to carefully study the features of ANN. There are four elements that makeup ANN Architecture:

1. Number of hidden layers in the network.
2. Number of neurons in each layer.
3. Activation function.
4. Training algorithm because it will decide the final value of the weights.

10.1: Number of hidden layers in Network:-

Hidden layer is the layer which is located between the input and output. Basically in hidden layer functions apply the weight from the inputs and then transfer them as outputs to the activation function. In simple words, the hidden layer performs the nonlinear transformation of inputs of the network to the activation function as outputs. Hidden layer depends on the function of neural networks and similarly layers depend on the associated weights.

10.2: Number of neurons in Each Layer:

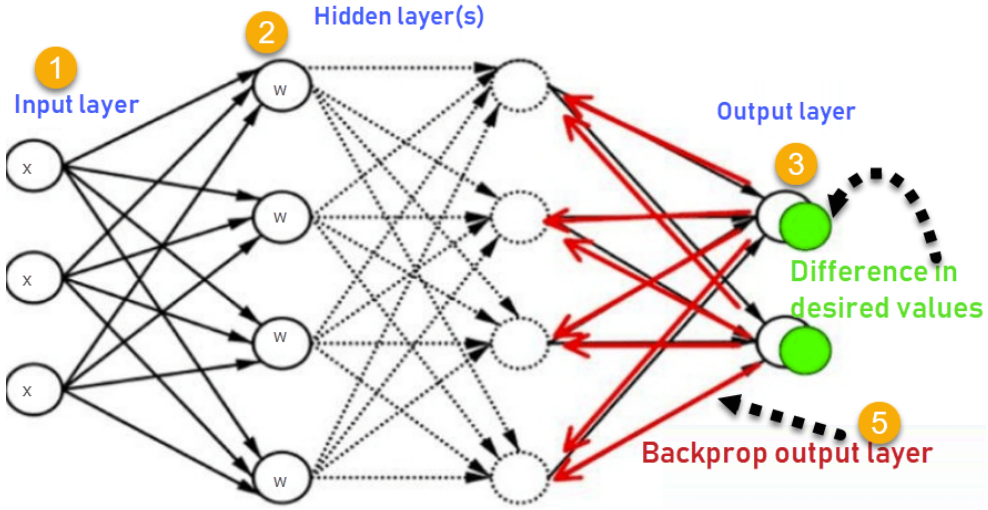
The number of neurons in the input layer is equal to the number of features in the data. Whereas the number of neurons depends on whether the model is used as a regressor or classifier. In case if the model is the regressor then the output layer has only one neuron but if the model is the classifier then it may be single or multiple neurons depending on the class label of the model as in our class we use multiple neurons for classification.

10.3: Activation Function:

The activation function in neural networks is the weighted sum of inputs that is transferred into an output from nodes in the layer of the network. Some activation functions are nonlinear and may be referred to as nonlinearity in the network design.

10.4: Back Propagation Algorithm:

Backpropagation is basically the essence of neural network training. This is the method of fine-tuning weights of neural networks which are based on the error rate obtained in the previous iteration. This tuning of the weights allows us to reduce the rate of errors and then make the model more reliable by increasing its generalization. Backpropagation is used for computing the gradient of loss function for a single weight by the chain rule. It is more efficient in computing one layer at a time, unlike a native direct computation. As it computes the gradient but it does not define how the gradient actually works or is used.



11. Tools/ IDE

- For this project the development tool which is used is Vs code and the language is python while the classifier is sklearn MLP.

- **11.1 Library**

There are other libraries for this project as well such as sklearn.tree DecisionTreeClassifier in this all setting is used by default and machine get learn very fast but we use sklearn.neural_netwrok MLPcClassifier in this we have to set input neuron layer hidden neuron layer and output neuron layer, activation function etc. So in this first we install sklearn from terminal and then use it.

12. Methodology & implementation:

12.1 Feature Extractions

Feature extractions are the different test on different feature through MLP and some features which we test are below.

- **Whole Darker Part.**

In this feature we pass the whole darker part of every image to MLP, train and check the accuracy, with this feature our prediction was going on but not much accurate

- **Diagonal Part**

In this feature we pass only diagonal part of the image to MLP and train and check the accuracy with this feature we got the accuracy compared to whole dark part.

- **Crop all images in rectangular form**

In this feature we crop the all image in specific diminsion and then train on MLP and check the accuracy and got more efficient prediction comparatively previous diagonal one.

12.2. In implementation we have different libraries, functions which are explain below in detatils.

- **12.2.1 MLP Library:**

The library MLP is used for training machine and in this library we pass x-test and y test, with some other builtin parameter i.e learning rate, activation function, sgd, fit etc.

- **12.2.2 OS Library(For reading Files from directory):**

By using the OS library (os.listdir) we read all the files from the current directoryby giving the path of those files.

- **12.2.3 MatPlot(For read images):**

For reading the images we use the Matplot, and after reading all the images an array of these images is formed.

- **12.2.4 X-Train:**

The array which is formed from the reading of images using Matplot, we make a count variable in which we store all the darker parts of the image in that array. It is one of our features on which we can train the classifier MLP. Later on we also added some

more features in X-train like finding the diagonal and its black part count. At last we combined both the features and appended them to X-Train.

- **12.2.5 Y-Train:**

In Y-train we enter into that directory where all images are present for example we enter in the directory where all images of (1) are present then print all the images. Same method for printing images of all other numbers.

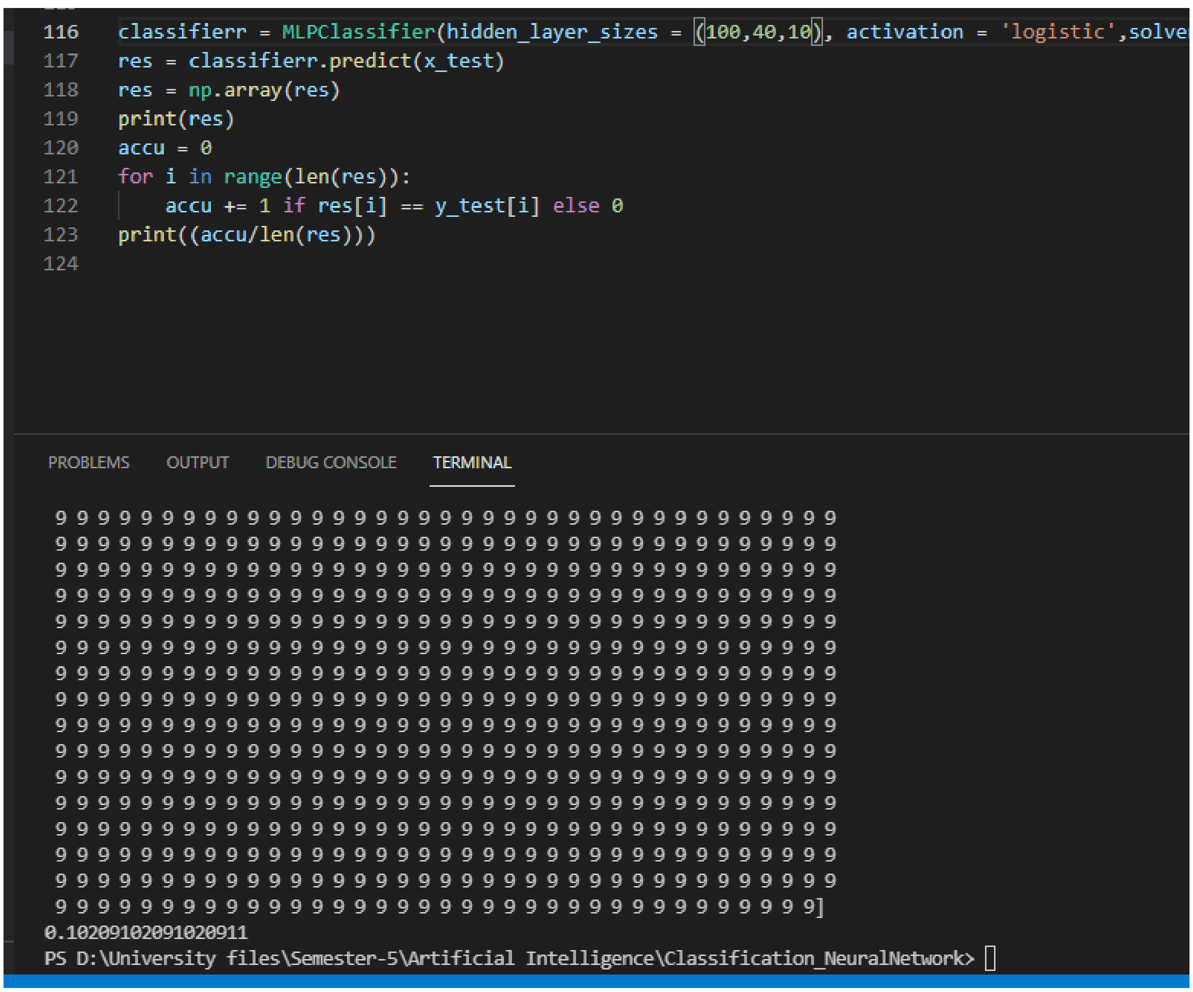
- **12.2.6 X-Test and Y-Test:**

As we perform X and Y train same is the case in X and Y test we enters in to the directory of validation and we get X and Y test there and then we pass that to the MLP function for testing to train the data and then we pass the validation of X and Y test to the fit function of MLP.

13. Testing & Experiments

- **Experiment_01.**
- **When use pass x_test, y-test to MLP with two feature diagonal and count shaded part:**

    - **Observation:**
        - By testing observation when pass x_test, y-test first time to MLP is tested by below experiment.
        -
    - **Result:** Below figures confirm this result.



- ❖ **Experiment_02.**
- ❖ **Extract darker part of image :**

  - ➢ **Observation:**
    - ★ By comparing the x_train, y_train with x_train, y_train i.e MLPClassifier(hidden_layer_sizes = (25,25,10), activation = 'logistic', learning_rate_init = 1,max_iter=1000).fit(x_test, y_test)
    - ★ The Following experiment will test this observation.

  - ❖ **Result:** When we pass all images darker shade without cropping dimension as images originally are in the multi-dimension machine didn't predict 1, below it the result [4 4 4 ... 4 4 4] accuracy 0.13594581519109822

```python
classifierr = MLPClassifier(hidden_layer_sizes = (30,18,10), activation = 'logistic',solver='sgd',
res = classifierr.predict(x_test)
res = np.array(res)
print(res)
accu = 0
for i in range(len(res)):
    accu += 1 if res[i] == y_test[i] else 0
print((accu/len(res)))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4]
0.13594581519109822
PS D:\University files\Semester-5\Artificial Intelligence\Classification_NeuralNetwork>
```

- **Experiment_03.**
- **Diagonal shaded part for all images**
  - **Observation:**
    - ★ By taking image diagonal shaded part, then train to MLP. no of neurons in hidden layer 25,25 and output neuron 10. The Following experiment will test this observation.

  - **Result:** When We use the diagonal feature instead of passing the whole image darker part there is a good change in accuracy from 0.13 to 0.17% accuracy and getting start result 1 with 1.

```
34  for i in image_List:
35      # n_zeros = np.count_nonzero(np.round(i)==0)
36      diag=np.diagonal(i)
37      diag_zeros = np.count_nonzero(np.round(diag)==0)
38      dark_part_count.append(diag_zeros)
39
40      # dark_part_count.append(n_zeros)
41  x_train = dark_part_count
42  # print(x_train,'xt')
```

```
[1 1 4 4 1 4 4 4 4 4 1 4 4 4 4 1 1 4 4 1 4 1 4 4 4 4 1 1 1 4 1 4 9 1 4 9 1
 1 9 9 4 1 4 9 1 1 1 1 1 1 9 1 1 1 4 1 1 4 9 1 9 4 9 4 1 4 1 1 9 4 1 1 1 4
 9 1 1 9 1 1 1 1 1 1 1 1 1 1 9 1 1 1 4 1 4 1 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4
 1 1 1 1 1 4 1 4 9 4 4 4 4 1 1 1 9 1 4 4 1 4 1 4 1 1 4 9 1 4 1 4 1 9 4 1 1
 4 1 1 1 1 4 4 1 1 9 4 4 4 1 4 4 1 4 9 9 9 1 4 1 4 1 1 1 4 1 1 4 1 4 1 4 4
 9 4 9 1 1 1 1 9 1 4 4 4 4 1 1 1 9 9 4 1 1 9 4 4 1 4 4 4 4 4 4 1 1 4 4 1 4 4
 1 1 9 1 4 1 4 4 4 1 4 4 1 1 4 1 1 4 1 1 4 4 9 4 1 1 9 1 1 1 4 4 1 1 1 1 9
 9 1 1 1 4 9 4 4 4 4 1 4 1 1 9 4 4 1 9 9 4 4 4 4 1 1 4 4 9 1 1 9 4 4 9 4 4 4
 4 4 1 9 4 9 4 4 4 1 1 1 4 4 1 4 4 1 4 1 4 4 4 4 1 4 9 9 9 4 1 1 1 1 4 4 9 9
 4 4 4 4 4 4 4 4 4 1 1 4 4 1 1 4 1 9 1 4 4 4 4 4 9 4 9 9 1 4 4 4 1 4 9 4 1
 1 9 1 4 1 1 4 4 4 4 9 4 1 1 4 4 4 4 4 1 1 4 4 9 4 4 9 4 9 1 1 4 1 1 4 4 4
 4 9 1 4 4 9 4 9 4 4 1 4 4 1 9 9 9 9 1 9 1 9 4 9 4 1 9 9 4 4 9 1 1 4 4 1 1
 4 1 9 4 9 1 4 4 9 4 4 4 4 9 1 9 1 9 1 1 4 4 9 1 9 9 9 4 1 9 4 9 4 1 4 1 9
 1 1 1 9 9 1 1 9 1 4 1 9 1 4 4 1 1 9 1 4 1 9 4 9 4 1 9 9 4 9 4 1 9 4 1 1 9
```

```
0.1741654571843251
PS D:\University files\Semester-5\Artificial Intelligence\Classification_NeuralNetwork>
```

❖ **Experiment_04.**

❖ **Crop all images in equal rectangle form:**

➢ **Observation:**

★ By cropping image-specific parts like small rectangle part darker shade count, then pass x_test, y_test to MLP. no of neurons in hidden layer 25,25 and output neuron 10. The Following experiment will test this observation.

❖ **Result:** When We use the crop feature instead of passing the whole image darker part there is a good change in accuracy from 0.13 to 0.20% accuracy and getting start result 1 with 1. With crop feature our result got increasing.

```
    img = np.array(image.imread(Train_Diractory+"\\"+folders+"\\"+i))
    #crop image small rectangle and pass it to MLPS
    crop = img[100:200, 150:300]
    image_List.append(crop)
    # image_List.append(img)

darker part of all images which is 0 or x_train, i.e in first 1 image contaion 1181 darker par
in image_List:

TPUT   DEBUG CONSOLE   TERMINAL

09822
ty files\Semester-5\Artificial Intelligence\Classification_NeuralNetwork> & "C:/Users/Danish khan/AppDat
 Intelligence/Classification_NeuralNetwork/movieclass.py"
```

```
classifierr = MLPClassifier(hidden_layer_sizes = (25,25,10), activation = 'logistic',solver='sgd', learning_rate_init = 0.1,max_iter=1000).fit(x_train, y
res = classifierr.predict(x_test)
res = np.array(res)
print(res)
```

```
35    #for whole image shaded

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL

1 4 4 4 4 1 4 1 4 1 4 1 4 4 4 1 1 1 4 4 4 4 1 1 4 1 1 1 4 4 4 4 1 1 4 4 4
4 4 4 4 1 4 4 1 4 1 4 4 4 4 4 4 1 1 4 4 4 4 4 4 1 4 4 1 1 4 4 4 1 1 1 4 1
4 4 4 4 4 1 4 4 1 4 4 4 4 4 4 4 1 4 4 4 4 1 4 1 4 4 4 4 4 4 4 4 1 4 4 4 4 4 4
4 1 1 4 1 1 4 4 4 4 4 4 4 1 1 1 1 1 4 4 4 4 1 1 4 4 4 4 4 4 4 1 4 1 4 4 4 4 1
4 4 4 4 4 4 4 4 4 4 4 4 4 1 1 1 4 4 4 4 4 4 1 4 4 4 4 1 1 1 4 4 4 4 4 4 4 4 1 4 1
4 1 1 1 4 4 4 4 1 1 4 1 4 4 4 4 1 4 4 1 4 4 4 4 4 4 4 1 4 4 1 1 1 4 4 4 4
4 4 1 4 4 4 1 4 4 1 1 4 4 4 4 4 4 4 4 1 4 4 4 1 4 1 1 1 4 4 4 1 1 1 1 1 4 4
1 1 4 4 4 4 1 1 4 4 4 4 4 4 4 1 1 4 4 4 1 4 1 1 4 4 1 1 4 1 1 4 4 4 4 4 1 4
4 1 1 4 4 1 4 4 1 4 1 4 1 4 4 1 4 4 4 4 4 4 4 4 4 4 4 1 4 4 1 1 4 4 1 4 4
4 1 1 1 4 4 4 1 1 1 1 1 4 1 1 4 4 4 1 4 1 1 1 4 1 4 1 1 4 1 1 4 1 4 1 4 1 4 1
1 1 1 4 1 1 1 1 4 1 1 1 1 1 4 4 4 4 1 1 1 4 4 4 4 4 1 4 1 4 1 4 4 4 4 4 4 1 4
4 4 4 4 1 1 4 1 4 4 1 4 1 1 1 1 4 4 1 1 4 1 4 1 1 1 4 1 1 1 4 1 1 1 1 1 1 1
1 1 1 4 1 4 4 1 1 1 1 4 4 4 1 1 4 1 4 1 1 4 1 1 1 4 4 4 1 1 1 1 1 4 1 4 1
1 4 1 4 1 1 1 1 1 4 4 1 1 1 1 1 4 4 1 1 1 1 1 1 1 1 4 4 4 4 4 4 4 1]
0.20416061925495887
```

- ❖ **Experiment_05.**
- ❖ **When use SklearnTree, DecisionTreeClassifier with two feature diagonal and count shaded part:**

  - ➢ **Observation:**
    - ★ By using DecisionTreeClassifier do an experiment and the observation is tested by below experiment.
    - ★
  - ❖ **Result:** When We use the sklearnTree classifier for just testing purpose we got almost accurate result and below figures confirm this result.

```
#for diagonal shaded part
   diag=np.diagonal(i)
   diag_zeros = np.count_nonzero(np.round(diag)==0)
   dark_part_count.append([diag_zeros,n_zeros])


x_train = dark_part_count
# print(x_train,'xt')
```

```
    for j in range(values):
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9
  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9
  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9
  2  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9
  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9 10 10 10 10 10 10 10 10
 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
 10 10 10]
0.9951620706337687
PS D:\University files\Semester-5\Artificial Intelligence\Classification_NeuralNetwork>
```

14. Refrences.

https://stackoverflow.com/questions/1987694/how-to-print-the-full-numpy-array-without-truncation

https://www.geeksforgeeks.org/crop-image-with-opencv-python/

https://scikit-learn.org/stable/auto_examples/classification/plot_digits_classification.html

https://www.analyticsvidhya.com/blog/2019/08/3-techniques-extract-features-from-image-data-machine-learning-python/

https://deepai.org/machine-learning-glossary-and-terms/hidden-layer-machine-learning

https://en.wikipedia.org/wiki/Biological_neuron_model