DL Innovative Project Report

DELHI TECHNOLOGICAL UNIVERSITY

EE328 - Deep Learning and Artificial Neural Networks

Multiclass Classification (0-9 digits)

INNOVATIVE PROJECT REPORT
SUBMITTED TO : Prof. Sudarshan Kumar Babu Valluru

SUBMITTED BY:
ISHAN KHANNA
2K19/EE/120

ISHAN KHANNA
2K19/EE/120

<u>DL Innovative Project Report</u>

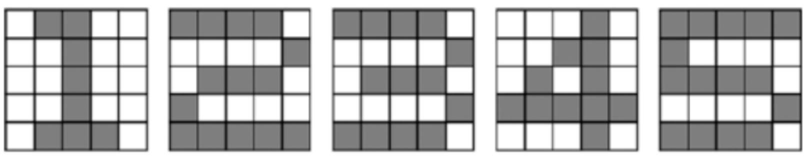
AIM

To perform Analysis of Multiclass Classification problem for image recognition model digits 0-9 for various activation functions using MATLAB.

Introduction

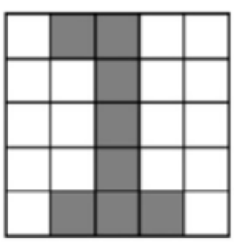
In this project, we've implemented a multiclass classifier network that recognizes digits from the input images.

The input image is considered as a black & white image with a resolution of 5X5 pixels.



For this project, we have created an array of 10 images, each representing a 5X5, 2D, Black & White image of digit from 0 to 9.

The input data X of the code is a two-dimensional matrix, which encodes the white pixel into a zero and the black pixel into a unity. For example, the image of the number 1 is encoded in the matrix shown



$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

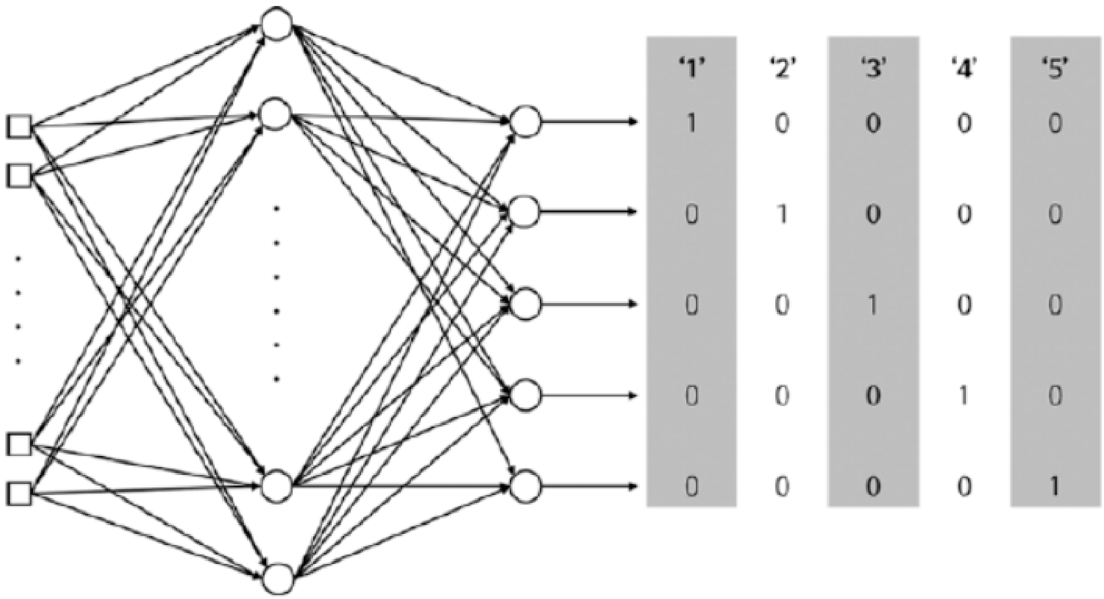
The variable D contains the correct output.

```
D = [
    1 0 0 0 0 0 0 0 0 0;
    0 1 0 0 0 0 0 0 0 0;
    0 0 1 0 0 0 0 0 0 0;
    0 0 0 1 0 0 0 0 0 0;
    0 0 0 0 1 0 0 0 0 0;
    0 0 0 0 0 1 0 0 0 0;
    0 0 0 0 0 0 1 0 0 0;
    0 0 0 0 0 0 0 1 0 0;
    0 0 0 0 0 0 0 0 1 0;
    0 0 0 0 0 0 0 0 0 1;
];
```

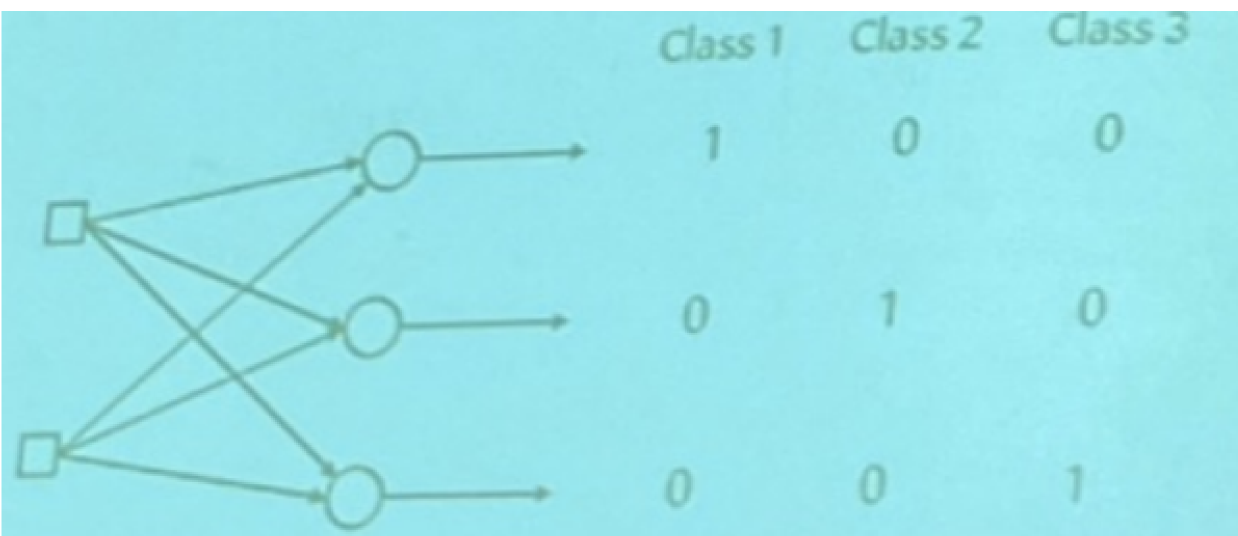
ISHAN KHANNA
2K19/EE/120

DL Innovative Project Report

Neural Network Architecture



The neural network model contains a single hidden layer, as shown in above figure. As each image is set on a matrix, we set 25 input nodes. In addition, as we have 10 digits to classify(0, 1, ..., 9), the network contains 10 output nodes. The softmax function is used as the activation function of the output node. The hidden layer has 50 nodes and the sigmoid function is used as the activation function.



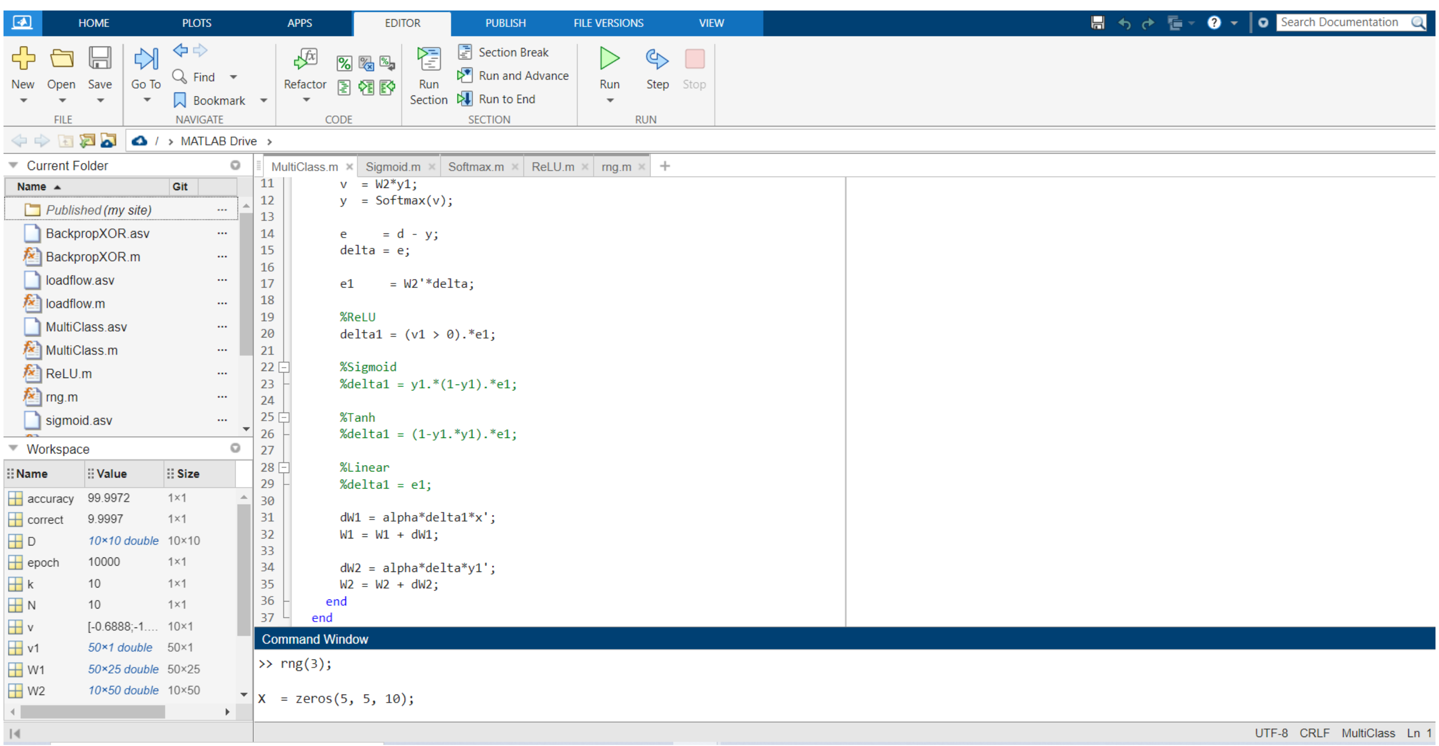
ISHAN KHANNA
2K19/EE/120

DL Innovative Project Report

OBSERVATIONS

MultiClass.m

```
MultiClass.m × | Sigmoid.m × | Softmax.m × | ReLU.m × | rng.m × | +

function [W1, W2] = MultiClass(W1, W2, X, D)
  alpha = 0.1;

  N = 10;
  for k = 1:N
    x = reshape(X(:, :, k), 25, 1);
    d = D(k, :)';

    v1 = W1*x;
    y1 = ReLU(v1);
    v  = W2*y1;
    y  = Softmax(v);

    e     = d - y;
    delta = e;

    e1     = W2'*delta;

    %ReLU
    delta1 = (v1 > 0).*e1;

    %Sigmoid
    %delta1 = y1.*(1-y1).*e1;

    %Tanh
    %delta1 = (1-y1.*y1).*e1;

```



Softmax.m

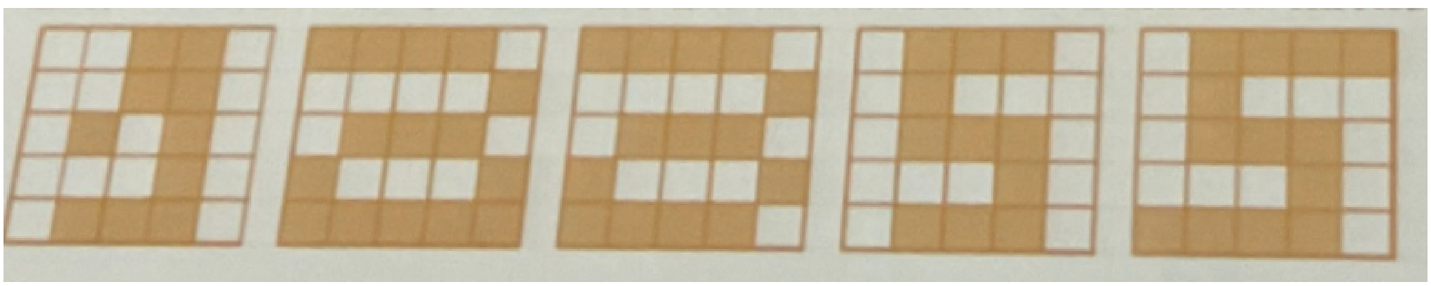
ISHAN KHANNA
2K19/EE/120

DL Innovative Project Report

```
function y = Softmax(x)
  ex = exp(x);
  y  = ex / sum(ex);
end
```

Here, in the cross entropy-driven learning rule that uses the softmax activation function, the delta and error are identical.
Even though the back-propagation algorithm applies to the hidden layer : e1 = W2'*delta;
delta1 = y1.*(1-y1).*e1;
The function Softmax, which the function MultiClass calls in, is implemented in the Softmax.m file.

TestMultiClass10.m



In order to test the ability of the trained Neural Network to recognize the digits, after training the NN using the correct data, we will give corrupted data as input.

In TestMultiClass10.m file, we have changed the input matrix for each digit, such that a 0 and a 1 have been swapped in the matrix for each digit, thus slightly altering the shape of the figure.

ISHAN KHANNA
2K19/EE/120

DL Innovative Project Report

MultiClass.m | Sigmoid.m | Softmax.m | ReLU.m | rng.m

Command Window

```
>> rng(3);

X  = zeros(5, 5, 10);

X(:, :, 1) = [ 0 1 1 0 0;
               0 0 1 0 0;
               0 0 1 0 0;
               0 0 1 0 0;
               0 1 1 1 0
             ];

X(:, :, 2) = [ 1 1 1 1 0;
               0 0 0 0 1;
               0 1 1 1 0;
               1 0 0 0 0;
               1 1 1 1 1
             ];

X(:, :, 3) = [ 1 1 1 1 0;
               0 0 0 0 1;
               0 1 1 1 0;
               0 0 0 0 1;
               1 1 1 1 0
             ];

X(:, :, 4) = [ 0 0 0 1 0;
               0 0 1 1 0;
               0 1 0 1 0;
               1 1 1 1 1;
               0 0 0 1 0
```

MultiClass.m | Sigmoid.m | Softmax.m | ReLU.m | rng.m

Command Window

```
               0 1 0 1 0;
               1 1 1 1 1;
               0 0 0 1 0
             ];

X(:, :, 5) = [ 1 1 1 1 1;
               1 0 0 0 0;
               1 1 1 1 0;
               0 0 0 0 1;
               1 1 1 1 0
             ];

X(:, :, 6) = [ 1 1 1 1 1;
               1 0 0 0 0;
               1 1 1 1 1;
               1 0 0 0 1;
               1 1 1 1 1
             ];

X(:, :, 7) = [ 1 1 1 1 1;
               0 0 0 1 0;
               0 0 1 0 0;
               0 1 0 0 0;
               1 0 0 0 0
             ];

X(:, :, 8) = [ 0 1 1 1 0;
               1 0 0 0 1;
               0 1 1 1 0;
               1 0 0 0 1;
```

ISHAN KHANNA

2K19/EE/120

DL Innovative Project Report

MultiClass.m | Sigmoid.m | Softmax.m | ReLU.m | rng.m

Command Window

```
               1 0 0 0 1;
               0 1 1 1 0
             ];

X(:, :, 9) = [ 0 1 1 1 0;
               1 0 0 1 0;
               0 1 1 1 0;
               0 0 0 1 0;
               0 0 0 1 0
             ];

X(:, :, 10) = [ 0 0 1 0 0;    %For Zero
                0 1 0 1 0;
                1 0 0 0 1;
                0 1 0 1 0;
                0 0 1 0 0
             ];

D = [ 1 0 0 0 0 0 0 0 0 0;
      0 1 0 0 0 0 0 0 0 0;
      0 0 1 0 0 0 0 0 0 0;
      0 0 0 1 0 0 0 0 0 0;
      0 0 0 0 1 0 0 0 0 0;
      0 0 0 0 0 1 0 0 0 0;
      0 0 0 0 0 0 1 0 0 0;
      0 0 0 0 0 0 0 1 0 0;
      0 0 0 0 0 0 0 0 1 0;
      0 0 0 0 0 0 0 0 0 1;
    ];
```

MultiClass.m | Sigmoid.m | Softmax.m | ReLU.m | rng.m

Command Window

```
W1 = 2*rand(50, 25) - 1;
W2 = 2*rand( 10, 50) - 1;

for epoch = 1:10000              % train
  [W1 W2] = MultiClass(W1, W2, X, D);
end

X(:, :, 1) = [ 0 0 1 1 0;
               0 0 1 0 0;
               0 0 1 0 0;
               0 0 1 0 0;
               0 1 1 1 0
             ];

X(:, :, 2) = [ 0 1 1 1 0;
               0 0 0 0 1;
               0 1 1 1 1;
               1 0 0 0 0;
               1 1 1 1 1
             ];

X(:, :, 3) = [ 1 1 1 1 0;
               0 0 0 1 0;
               0 1 1 1 0;
               0 0 0 0 1;
               1 1 1 1 0
             ];

X(:, :, 4) = [ 0 0 0 1 0;
               0 0 1 1 0;
```

ISHAN KHANNA
2K19/EE/120

DL Innovative Project Report

MultiClass.m | Sigmoid.m | Softmax.m | R

Command Window

```
X(:, :, 4) = [ 0 0 0 1 0;
               0 0 1 1 0;
               0 1 0 1 0;
               0 1 1 1 1;
               0 0 0 1 1
             ];

X(:, :, 5) = [ 1 1 1 1 1;
               1 0 0 0 0;
               1 1 1 1 0;
               0 0 0 1 0;
               1 1 1 1 0
             ];

X(:, :, 6) = [ 1 1 1 1 1;
               1 0 0 0 0;
               1 1 1 1 1;
               0 1 0 0 1;
               1 1 1 1 1
             ];

X(:, :, 7) = [ 0 1 1 1 1;
               0 0 0 1 0;
               0 0 1 0 0;
               0 1 1 0 0;
               1 0 0 0 0
             ];

X(:, :, 8) = [ 0 1 1 1 0;
```

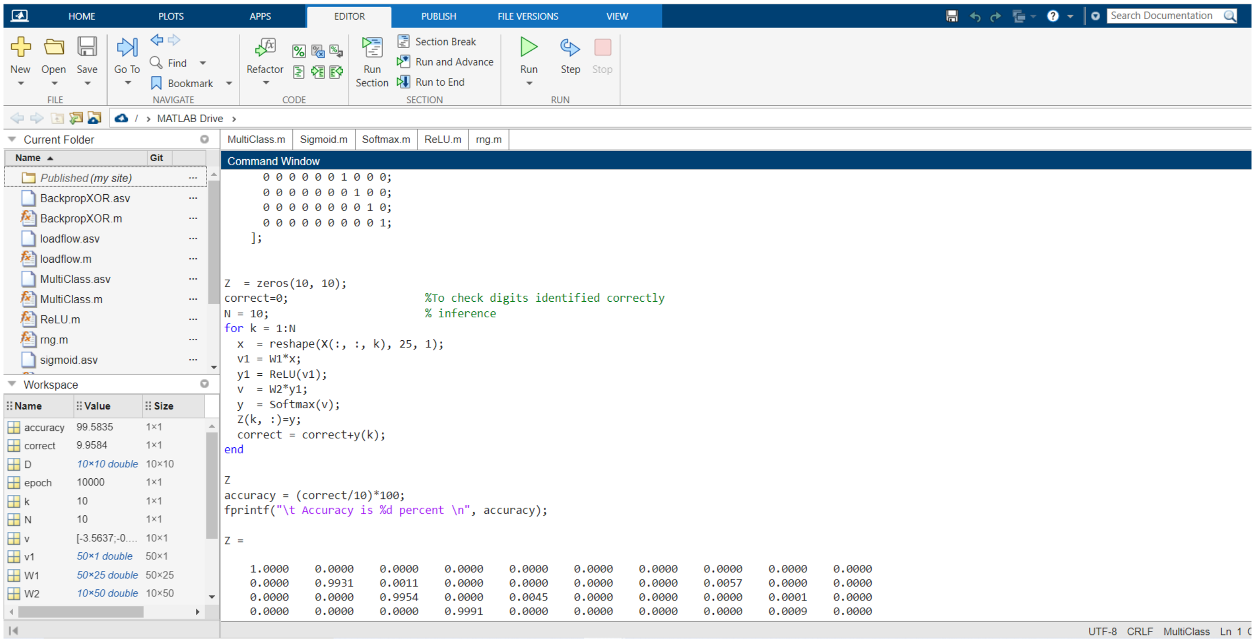
MultiClass.m | Sigmoid.m | Softmax.m | ReLU.m | rn

Command Window

```
               1 0 0 0 1;
               0 1 1 1 0;
               1 0 0 0 1;
               0 1 1 1 0
             ];

X(:, :, 9) = [ 0 1 1 1 0;
               1 0 0 1 0;
               1 1 1 1 0;
               0 0 0 1 0;
               0 0 0 0 0
             ];

X(:, :, 10) = [ 0 0 1 0 0;     %For Zero
                0 1 0 1 0;
                1 0 0 1 0;
                0 1 0 1 0;
                0 0 1 0 0
             ];

D = [ 1 0 0 0 0 0 0 0 0 0;
      0 1 0 0 0 0 0 0 0 0;
      0 0 1 0 0 0 0 0 0 0;
      0 0 0 1 0 0 0 0 0 0;
      0 0 0 0 1 0 0 0 0 0;
      0 0 0 0 0 1 0 0 0 0;
      0 0 0 0 0 0 1 0 0 0;
      0 0 0 0 0 0 0 1 0 0;
      0 0 0 0 0 0 0 0 1 0;
      0 0 0 0 0 0 0 0 0 1;
```

ISHAN KHANNA
2K19/EE/120

DL Innovative Project Report



1) ReLU

MultiClass.m | Sigmoid.m | Softmax.m | ReLU.m | rng.m

```
function y = ReLU(x)
    y = max(0, x);
end
```

```
Z =

    1.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.9931    0.0011    0.0000    0.0000    0.0000    0.0000    0.0057    0.0000    0.0000
    0.0000    0.0000    0.9954    0.0000    0.0045    0.0000    0.0000    0.0000    0.0001    0.0000
    0.0000    0.0000    0.0000    0.9991    0.0000    0.0000    0.0000    0.0000    0.0009    0.0000
    0.0001    0.0000    0.0000    0.0009    0.9774    0.0003    0.0000    0.0000    0.0213    0.0000
    0.0001    0.0000    0.0000    0.0018    0.0031    0.9949    0.0000    0.0000    0.0001    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.9999    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    1.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0001    0.9999    0.0000
    0.0000    0.0001    0.0000    0.0000    0.0000    0.0000    0.0002    0.0001    0.0009    0.9986

	 Accuracy is 9.958355e+01 percent
```

ISHAN KHANNA
2K19/EE/120

DL Innovative Project Report

2) Sigmoid

MultiClass.m | Sigmoid.m | Softmax.m

```
function y = Sigmoid(x)
  y = 1 ./ (1 + exp(-x));
end
```

```
Z =

    0.9989    0.0003    0.0001    0.0001    0.0000    0.0000    0.0001    0.0002    0.0002    0.0002
    0.0014    0.9833    0.0013    0.0000    0.0000    0.0020    0.0001    0.0117    0.0000    0.0001
    0.0000    0.0001    0.9964    0.0005    0.0015    0.0000    0.0011    0.0001    0.0003    0.0000
    0.0000    0.0000    0.0000    0.9995    0.0000    0.0000    0.0000    0.0000    0.0004    0.0001
    0.0002    0.0002    0.0000    0.0001    0.9886    0.0005    0.0000    0.0000    0.0103    0.0001
    0.0001    0.0001    0.0000    0.0004    0.0059    0.9918    0.0015    0.0001    0.0000    0.0000
    0.0005    0.0000    0.0000    0.0000    0.0000    0.0000    0.9994    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0001    0.0000    0.0000    0.0000    0.0000    0.9998    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0001    0.0010    0.0000    0.0000    0.0003    0.9980    0.0005
    0.0000    0.0000    0.0000    0.0002    0.0000    0.0000    0.0000    0.0000    0.0002    0.9995

        Accuracy is 9.955231e+01 percent
```

3) tanh

MultiClass.m | Tanh.m | Sigmoid.m

```
function y = Tanh(x)
y = 2 ./ (1 + exp(-2*x))-1;
end
```

```
Z =

    0.9985    0.0002    0.0000    0.0006    0.0001    0.0000    0.0000    0.0002    0.0004    0.0001
    0.0007    0.9910    0.0043    0.0000    0.0000    0.0002    0.0000    0.0037    0.0001    0.0000
    0.0000    0.0000    0.9956    0.0002    0.0003    0.0000    0.0016    0.0007    0.0014    0.0000
    0.0000    0.0000    0.0000    0.9994    0.0000    0.0000    0.0000    0.0000    0.0005    0.0000
    0.0002    0.0003    0.0002    0.0009    0.9917    0.0007    0.0000    0.0000    0.0059    0.0002
    0.0008    0.0000    0.0000    0.0046    0.0060    0.9870    0.0012    0.0001    0.0000    0.0003
    0.0002    0.0000    0.0000    0.0000    0.0000    0.0000    0.9993    0.0005    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    1.0000    0.0000    0.0000
    0.0001    0.0001    0.0002    0.0002    0.0067    0.0000    0.0000    0.0006    0.9900    0.0020
    0.0000    0.0000    0.0000    0.0002    0.0000    0.0000    0.0001    0.0000    0.0000    0.9997

        Accuracy is 9.952102e+01 percent
```

ISHAN KHANNA
2K19/EE/120

4) Linear

```
MultiClass.m ×   Linear.m ×   Tanh.m
function y = Linear(x)
y = x;
end
```

```
Z =

    1.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    1.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    1.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    1.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.9987    0.0000    0.0000    0.0000    0.0013    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    1.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    1.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    1.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    1.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    1.0000

        Accuracy is 9.998687e+01 percent
```

RESULTS AND CONCLUSION

The Accuracy for various activation functions in this case was found to be:

S.No.	FUNCTION	ACCURACY
1.	ReLU	99.58355%
2.	Sigmoid	99.55231%
3.	Tanh	99.52102%
4.	Linear	99.98687%

ISHAN KHANNA
2K19/EE/120

DL Innovative Project Report

In this case, the Linear Activation Function is found to be the most accurate. However, it may not always be the most accurate activation function in all cases since accuracy will vary from case to case depending upon the training data and other factors.

Since, for all Activation functions, the accuracy is above 99.5%. Hence, the experiment has been conducted successfully.

BIBLIOGRAPHY

References

[1] Cengil, E., Cinar, A., & Ozbay, E. (2017). Image classification with caffe deep learning framework. 2017 International Conference on Computer Science and Engineering (UBMK). http://doi.org/10.1109/ubmk.2017.8093433.

[2] CS231n: Convolutional Neural Networks for Visual Recognition. Retrieved May 21, 2019, from http://cs231n.stanford.edu/.

[3] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Rabinovich, A. (2015). Going deeper with convolutions. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). http://doi.org/10.1109/cvpr.2015.7298594.

[4] JT Turner, David Aha, Leslie Smith, & Kalyan Moy Gupta. Convolutional Architecture Exploration for Action Recognition and Image Classification (publication). Convolutional Architecture Exploration for Action Recognition and Image Classification. Knexus Research Corporation. Retrieved from https://apps.dtic.mil/dtic/tr/fulltext/u2/a624847.pdf.

[5] Gupta, D., & Dishashree. (2019, June 25). Master Transfer learning by using Pre-trained Models in Deep Learning. Retrieved July 23, 2019, from https://www.analyticsvidhya.com/blog/2017/06/transfer-learning-the-art-of-fine-tuning-a-pre-trained-model.

[6] Zhou, X., Yu, K., Zhang, T., & Huang, T. S. (2010). Image Classification Using Super-Vector Coding of Local Image Descriptors. Computer Vision – ECCV 2010 Lecture Notes in Computer Science, 141–154. http://doi.org/10.1007/978-3-642-15555-0_11.

[7] Akata, Z., Perronnin, F., Harchaoui, Z., & Schmid, C. (2014). Good Practice in Large-Scale Learning for Image Classification. IEEE Transactions on Pattern Analysis and Machine Intelligence, 36(3), 507–520. http://doi.org/10.1109/tpami.2013.146.

ISHAN KHANNA
2K19/EE/120