**Part 1: Forward Pass**

In the first part of the project, it is asked to implement the forward pass for Inner Product Layer, Pooling Layer, ReLU, and Convolution Layer for which code is done in their respective scripts found in the “matlab” folder of the project. After implementing the forward pass function in the mentioned layer scripts, the “test\_components.m” script is run to get the test results of the implemented layers.

Below are the results of the **test\_components.m:**

Graphical user interface, application

Description automatically generated with medium confidence

**A picture containing text, scoreboard, first-aid kit

Description automatically generated**

**Square

Description automatically generated with medium confidence**

**Graphical user interface, application, Teams

Description automatically generated**

**Part 2 Back Propagation**

After implementing the forward propagation in Part 1, back propagation is implemented using chain rule. Backward pass is implemented for the ReLU and Inner Product Layer in their respective scripts.

**Part 3 Training**

**Q 3.1 Training**

In this question, the script “train\_lanet.m” is run and test accuracy is noted. The training ran for 3000 iterations and the test accuracy came out to be ( **96.6%).**

Below is the result of the last iteration of the above-mentioned script:

|  |
| --- |
| cost = 0.111115 training\_percent = 0.960000  cost = 0.113216 training\_percent = 0.940000  cost = 0.134874 training\_percent = 0.960000  cost = 0.067548 training\_percent = 0.990000  cost = 0.095426 training\_percent = 0.980000  test accuracy: 0.966000 |

**Q 3.2 Test the network**

In this question, script “test\_network.m” has been provided which runs the test data through networks and obtains the predictions probabilities.

**Note:** For getting the confusion matrix, two MathWorks functions “confusionmat” and “confusionchart” are used.

The confusion matrix is made out of the prediction probabilities. Below is the generated confusion matrix.

**Confusion Matrix**Chart, scatter chart

Description automatically generated

**Confused Pairs**

It is obvious that the confusion matrix will be generated differently with running the script “test\_network.m”. But according to the stated confusion matrix the top 2 confused pairs are

1. **Position (4,9) –** Since the 4 and 9 integers are hard to distinguish as both the numbers are having a step and a closed figure structure from the top with only difference of curve type. “4” has right angle triangle closed structure and “9” has an oval structure which makes it hard to distinguish.
2. **Position (8,3)** – Both the integers have the same formation structure. “3” is the subfigure structure of digit “8”. “3” can be get on splitting “8” vertically. So again, the numbers are hard to be distinguished.

**AND**

**Position (5,8)-** The reasoning will be same of integers “5” and “8”. Number 8 can be gotten by closing number 5 from both the ends to form a curve structure. So, they are also hard to distinguish.

**Q 3.3 Real-World Testing**

In this question, model is tested with real-world data. So, a script named “real\_world\_testing.m” is made which is taking inputs of the “.jpg” files (downloaded and saved in a folder named “real\_testing” in the images folder.

**Note:** relative path is provided to read the .jpg files and test the model.

Below are the results: The model predicted 2 out of 5 test images.

**Test-image1**

**Shape

Description automatically generated with low confidenceOriginal Digit= 2 and Prediction for this test-image is: 5.**

**Test-image2**

**Icon

Description automatically generatedOriginal Digit= 3 and Prediction for this test-image is: 3.**

**Test-image3**

Icon

Description automatically generated **Original Digit= 7 and Prediction for this test-image is: 3**

**Test-image4**

**A picture containing background pattern

Description automatically generated Original Digit= 5 and Prediction for this test-image is: 5.**

**Test-image5**

**Icon

Description automatically generated Original Digit= 9 and Prediction for this test-image is: 8.**

**Part 4 Visualization**

**Q 4.1 ReLU and CONV layer Visuals**

In this question, “vis\_data.m” script needs to be implemented which visualize the output og the second and third layers (i.e., CONV and ReLu layer).

**Note:** Subplots are usedand organized in 4 x 5 format. The output visualizations are saved in “vis\_layers” folder made in the “images” folder. The relative path is provided to save the images. It is requested to make a new folder with the mentioned name inside images folder to see the resulted visuals.

Below are the CONV and ReLU layer resulted visuals:

**CONV Layer**

A black and white photo of a person's face

Description automatically generated with low confidence

**ReLU Layer**

A black and white photo of a person's face

Description automatically generated with low confidence

**Q 4.2 Compare the feature maps to the original image and explain the differences**

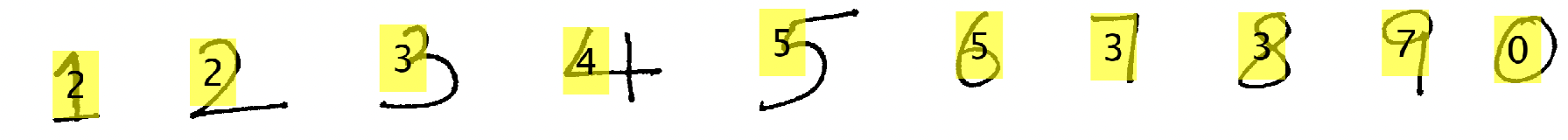
From the above part visuals, it is seen that both layers have same effect on the input image. The reason behind it would be that the convolutional layer has negative activations so it can’t be represented visually. It will be represented once the activations are changed to 0 which is similar to ReLu activations. Thus, both layers have the same visuals.

**Part 5 Image Classification**

In this question, it is asked to perform the task of Optical Character Recognition using fully trained model.

Below are the visuals of the image provided.

**For image1**



Recognizing 5/10 digits

**For image2**

****

Recognizing 3/10 digits

**For image3**

**A picture containing shape

Description automatically generated**

Recognizing 5/5 digits

**For image4**

****

Recognizing 37/50 digits

In total the model is recognizing 50/75 digits making prediction percentage to 67%.