

**Practical Machine Learning**

**ENGR 491/891**

**Programming Assignment 4**

**Spring 2022**

**Multi-Layer Perceptron**

ENGR 891: 100 points

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Last Name 1: Liew

First Name 1: Chia Sin

NUID 1:

Last Name 2:

First Name 2:

NUID 2:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Obtained Score:**

a) Part A: Using the table (given in Part A) report training accuracy, test accuracy, and epochs at which the training stopped.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Layer Configuration** | | | | **Result** |
|  | kernel\_initializer  (all layers) | activation  (hidden layers) | Dropout  (hidden layers) |
| Experiment 1 | zeros | sigmoid | No | Train accuracy = 0.469  Test accuracy = 0.463  Epochs = 50 |
| Experiment 2 | ones | sigmoid | No | Train accuracy = 0.963  Test accuracy = 0.960  Epochs = 50 |
| Experiment 3 | random\_normal | sigmoid | No | Train accuracy = 0.985  Test accuracy = 0.975  Epochs = 50 |
| Experiment 4 | random\_normal | tanh | No | Train accuracy = 0.999  Test accuracy = 0.980  Epochs = 50 |
| Experiment 5 | random\_normal | relu | No | Train accuracy = 0.999  Test accuracy = 0.981  Epochs = 20 |
| Experiment 6 | random\_normal | relu | Yes  Hidden layer 1: rate=0.1  Hidden layer 2: rate=0.1 | Train accuracy = 0.999  Test accuracy = 0.982  Epochs = 23 |
| Experiment 7 | random\_normal | relu | Yes  Hidden layer 1: rate=0.5  Hidden layer 2: rate=0.1 | Train accuracy = 0.999  Test accuracy = 0.982  Epochs = 50 |
| Experiment 8 | random\_normal | relu | Yes  Hidden layer 1: rate=0.5  Hidden layer 2: rate=0.1  SGD learning rate=0.5 | Train accuracy = 0.998  Test accuracy = 0.982  Epochs = 43 |
| Experiment 9 | random\_normal | relu | No | Train accuracy = 0.112  Test accuracy = 0.114  Epochs = 12 |
| Experiment 10 | random\_normal | relu | No | Train accuracy = 0.112  Test accuracy = 0.114  Epochs = 12 |

b) Part A: learning curves for experiments 1 to 10.

c) Part A: Answers to Q-1 to Q-6.

* Q-1) Among experiments 1 to 5, which experiment performed the best (based on

test accuracy & epochs)? Explain why.

Model in Experiment 5 performs the best with kernel initializer of “random\_normal” and activation for hidden layers of “relu” with the lowest epochs of 20 and test accuracy of 0.981. Kernel initialization with normally distributed tensors performs better than tensors with zeros and ones. The “relu” activation is more efficient because it does not activate all the neurons at the same time.

* Q-2) Among experiments 1 to 5, which experiment performed the worst (based on

test accuracy)? Explain why.

Model in Experiment 5 performs the best with kernel initializer of “zeros” and activation for hidden layers of “sigmoid” with test accuracy of 0.463. Kernel initialization with all zeros performs worse than ones and normally distributed tensors.

* Q-3) Compare experiment 6 with experiment 7 and determine which model experiences less overfitting. Explain why. To answer this question, use the train accuracy, test accuracy, and learning curves of these two experiments. In your answer show these measures (statistics and learning curves).

The train and test accuracy for both experiments are similar: Train accuracy = 0.999 and Test accuracy = 0.982. Therefore, we cannot tell if the models are overfitted or not from the numbers alone. However, if we look at the learning curves, model in Experiment 7 is less overfitted compared to Experiment 6.

Experiment 6

Chart, line chart

Description automatically generated

Experiment 7

Chart, line chart

Description automatically generated

Q-4) Compare experiment 7 with experiment 8. Explain the change in the number of epochs in experiment 8. If it has increased, explain why. If it has decreased, explain why.

The number of epochs has decreased from 50 (Experiment 7) to 43 (Experiment 8). This is because with a higher learning rate (0.5 in Experiment 8 vs 0.1 in Experiment 7), convergence of the loss function is reached faster.

* Q-5) Compare experiment 8 with experiment 9. Show the accuracy learning curves of these two models. Explain the difference. Which model performed poorly? Explain why.

Model in Experiment 9 performs poorly. This is probably due to the reduction of the number of neurons in the first hidden layer from 300 to 100. This detrimental effect cannot be offset by adding more layers (from 2 to 10) to the model.

Experiment 8

Chart, line chart

Description automatically generated

Experiment 9

Chart, line chart

Description automatically generated

* Q-6) Compare experiment 9 with experiment 10. Which model performed poorly? Explain why.

Both models in Experiments 9 and 10 perform poorly, with a test accuracy of 0.114. As mentioned in question 5, the detrimental effect of reducing the number of neurons in the first hidden layer cannot be offset by adding more layers (10 in Experiment 9 and 20 in Experiment 10). Doubling the number of layers from 10 to 20 does not seem to improve the model at all.

Experiment 10

Chart, line chart

Description automatically generated

d) Part B: a 2-column table that shows the model attributes in the 1st column and

their values in the 2nd column, train accuracy, test accuracy, test confusion matrix,

the test classification report, learning curves (accuracy and loss).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Layer Configuration** | | | | **Result** |
|  | size of mini batch | activation | Dropout  (hidden layers) |
| Experiment 1 | 64 | relu | No  Learning rate=0.1 | Train accuracy = 0.473  Test accuracy = 0.399  Epochs = 23 |
| Experiment 2 | 64 | relu | Yes  Hidden layer 1: rate=0.1  Hidden layer 2: rate=0.1  Learning rate=0.1 | Train accuracy = 0.468  Test accuracy = 0.419  Epochs = 22 |
| Experiment 3 | 64 | sigmoid | No  Learning rate=0.1 | Train accuracy = 0.532  Test accuracy = 0.4817  Epochs = 37 |
| Experiment 4 | 64 | sigmoid | Yes  Hidden layer 1: rate=0.1  Hidden layer 2: rate=0.1  Learning rate=0.1 | Train accuracy = 0.508  Test accuracy = 0.464  Epochs = 50 |
| Experiment 5 | 1000 | relu | No  Learning rate=0.1 | Train accuracy = 0.526  Test accuracy = 0.481  Epochs = 50 |
| Experiment 6 | 1000 | relu | No  Learning rate=0.2 | Train accuracy = 0.568  Test accuracy = 0.507  Epochs = 50 |
| Experiment 7 | 500 | relu | No  Learning rate=0.1 | Train accuracy = 0.583  Test accuracy = 0.500  Epochs = 50 |
| Experiment 8 | 1000  epochs=100 | relu | Yes  Hidden layer 1: rate=0.1  Hidden layer 2: rate=0.1  Learning rate=0.2 | Train accuracy = 0.636  Test accuracy = 0.531  Epochs = 90 |

e) Part B: Answer to Q-7.

* Q-7) Explain the quality of your test performance.

The best model that I designed does not perform very well. Its test accuracy is only 0.531. After trying hyperparameters of varying values, here is a summary of my observations:

* The number of neurons in each hidden layer and the number of hidden layers do not seem to improve the model very much.
* Sigmoid seems to outperform relu in certain cases.
* Dropout improves model performance for models with relu activation but not sigmoid activation.
* Size of mini batch and the number of epochs improve the model a little.
* Increasing learning rate by 0.1 also improves the model but further increment easily introduces overfitting.

As an educated guess, I think that there might be better neural network methods better suited for the classification of CIFAR dataset instead of the MLP.