

CS 203 Software Tools and Techniques for AI

Group: 12

Name: Harinarayan J (23110128) , Vyomika Vasireddy (23110363)

Section 1: MLP Model Implementation & Experiment Tracking

We used Python 3.11.11 for the first part.

We loaded the data, normalized it, and split it for the data cleaning process.

We used a MLP model with 4 neurons (input), 16 neurons (hidden), 3 neurons (output). We used the Cross Entropy Loss and Adam Optimizer.

We ran the model for 50 epochs.

These are the accuracy, precision, recall, F1 score for the first 50 epochs.

```
Accuracy: 0.7333333333333333
Precision: 0.9466666666666667
Recall: 0.7333333333333333
F1 Score: 0.7925925925925926
```

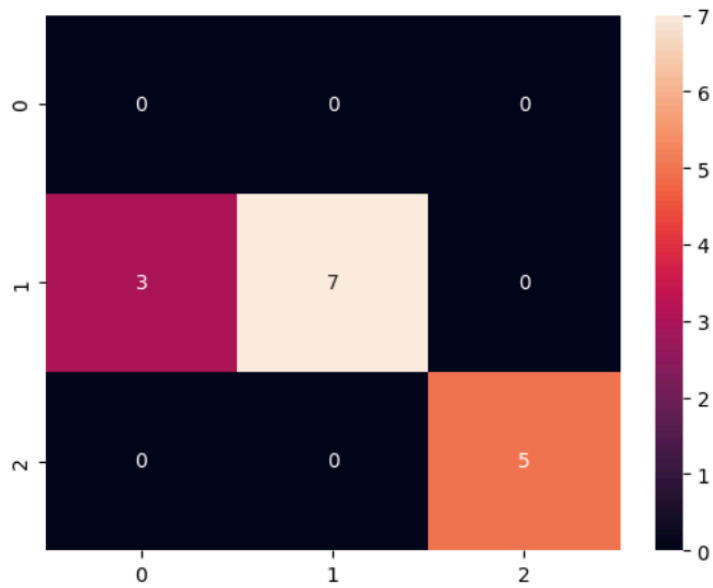


Fig. Confusion Matrix

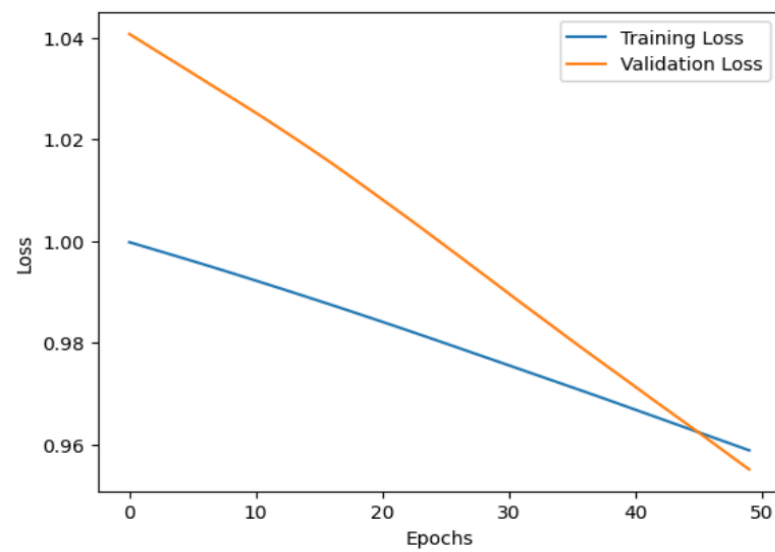


Fig. Loss vs epoch

This is the comparison between the training and validation loss over the epochs.

Now, we used the Weights and Biases module to check for the hyperparameters.

```

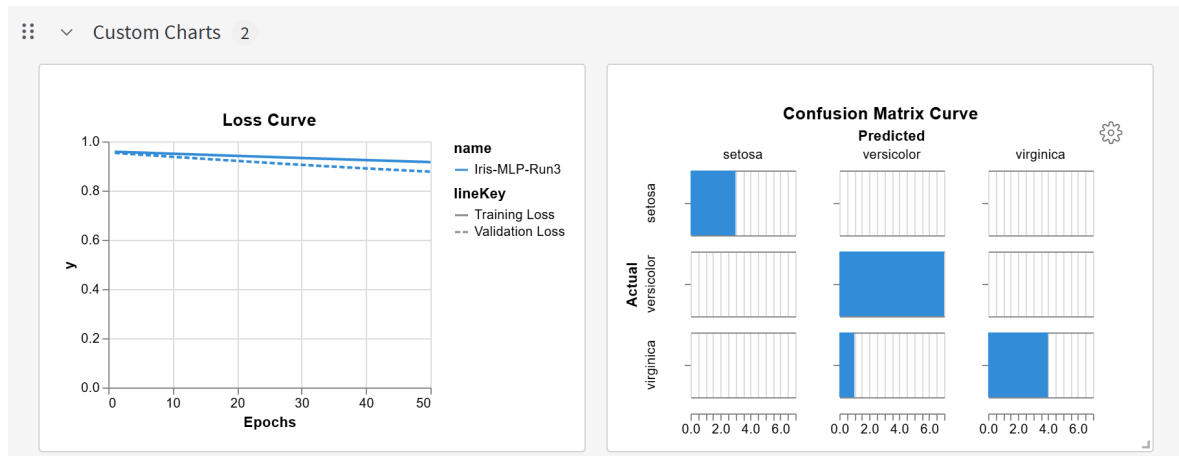
Activation:
  value:
    - ReLU
    - Softmax
Hyperparameters:
  value:
    Learning Rate: 0.001
    Loss Function: CrossEntropyLoss
    Optimizer: Adam
    batch_size: 32
    epochs: 50
architecture:
  value: MLP
dataset:
  value: Iris
layers:
  value:
    hidden: 16
    input: 4
    output: 3

```

Parameters from the wandb site.



Charts of the losses and metrics.

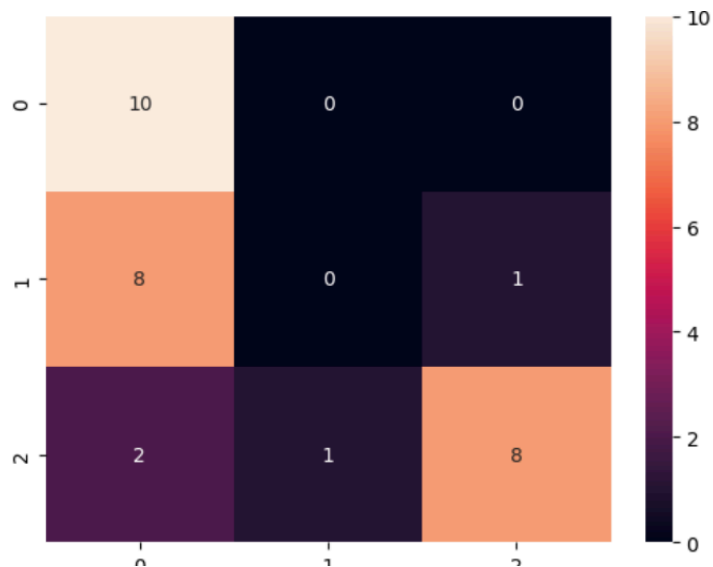


Two custom charts with the loss curves and the confusion matrices.

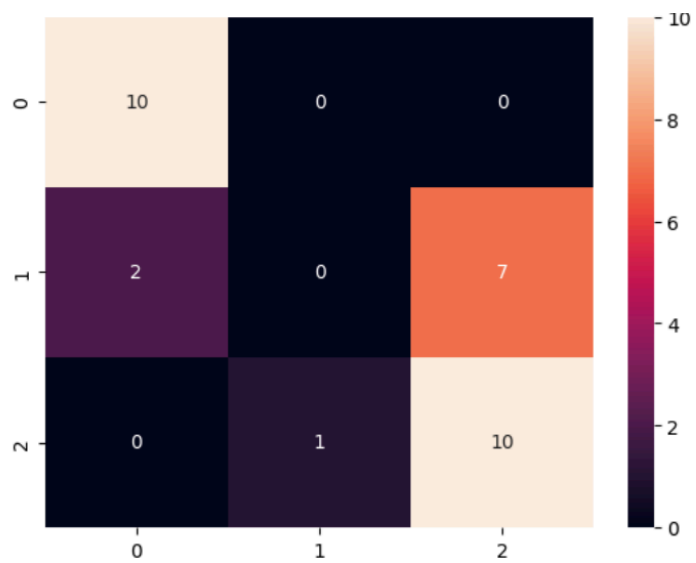
Section 2: Hyperparameters

First, we use the manual tuning, where we just iterate through all the possibilities, the following are the confusion matrices from each of the possibilities of the hyperparameters.

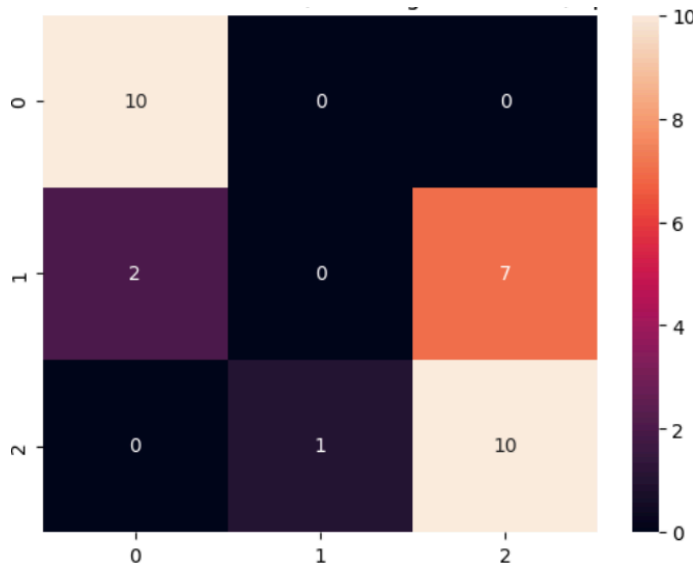
Confusion matrix



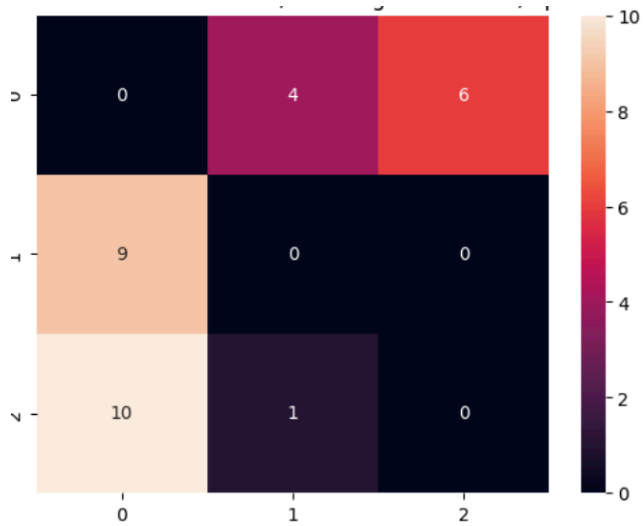
Confusion Matrix Batch Size 2, Learning Rate: 0.001, Epochs: 1



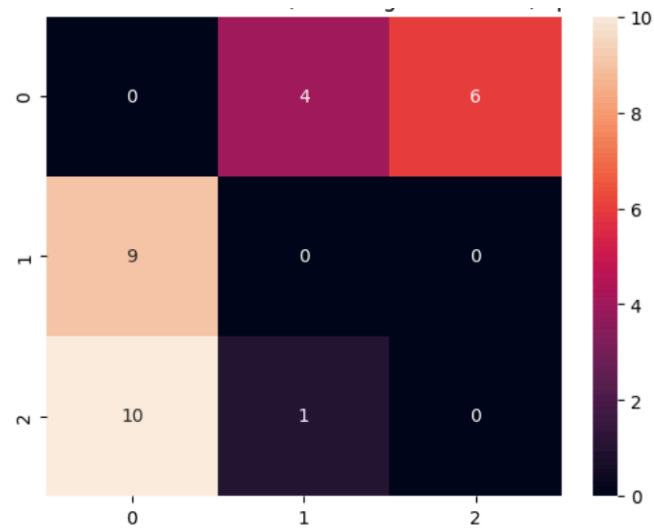
Confusion Matrix Batch Size 2, Learning Rate: 0.001, Epochs: 3



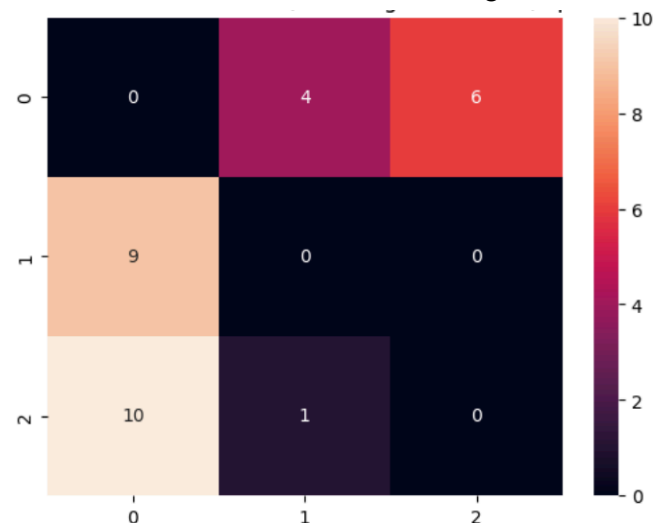
Confusion Matrix Batch Size 2, Learning Rate: 0.001, Epochs: 5



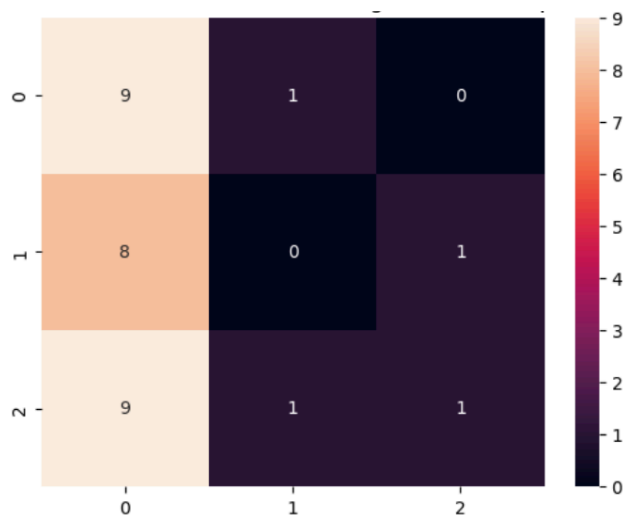
Confusion Matrix Batch Size 2, Learning Rate: 0.00001, Epochs: 1



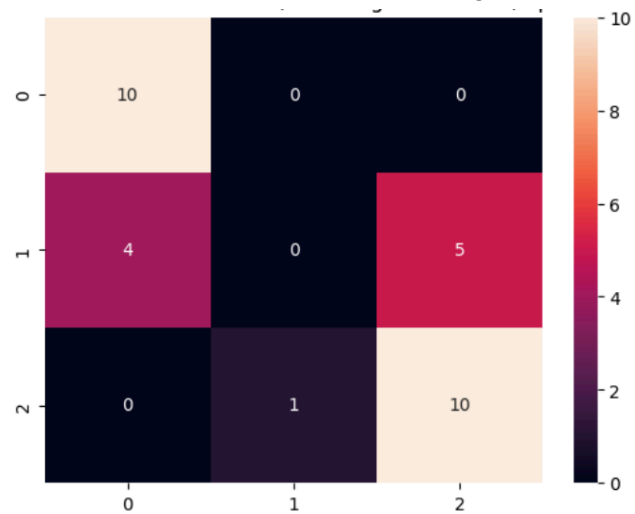
Confusion Matrix Batch Size 2, Learning Rate: 0.00001, Epochs: 3



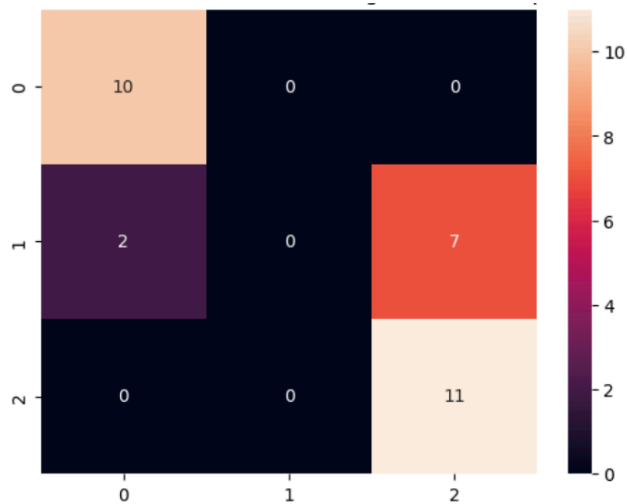
Confusion Matrix Batch Size 2, Learning Rate: 0.00001, Epochs: 5



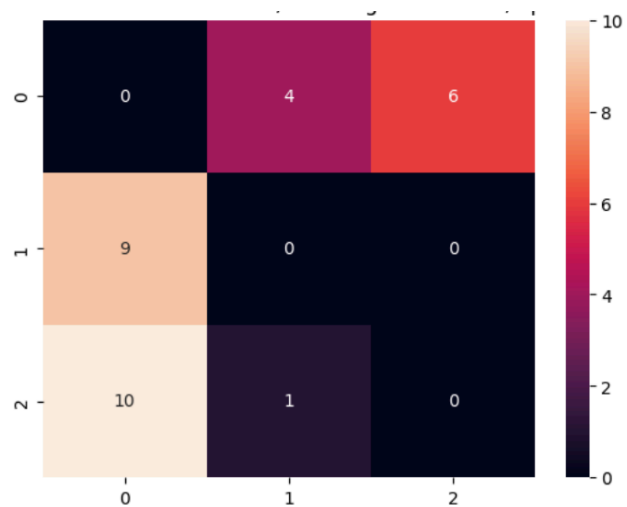
Confusion Matrix Batch Size 4, Learning Rate: 0.001, Epochs: 1



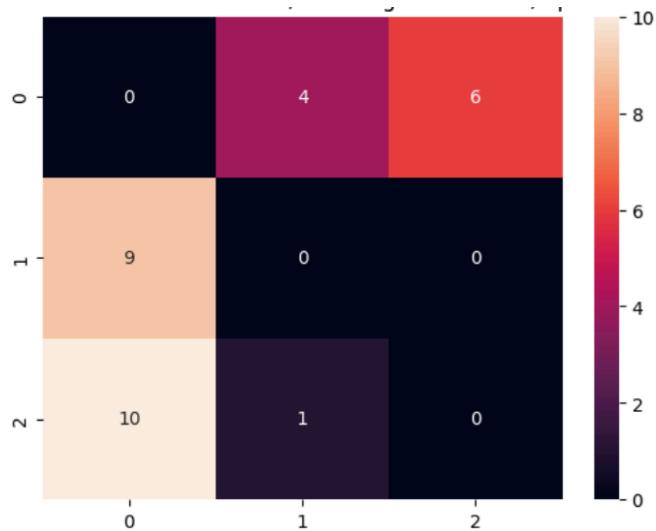
Confusion Matrix Batch Size 4, Learning Rate: 0.001, Epochs: 3



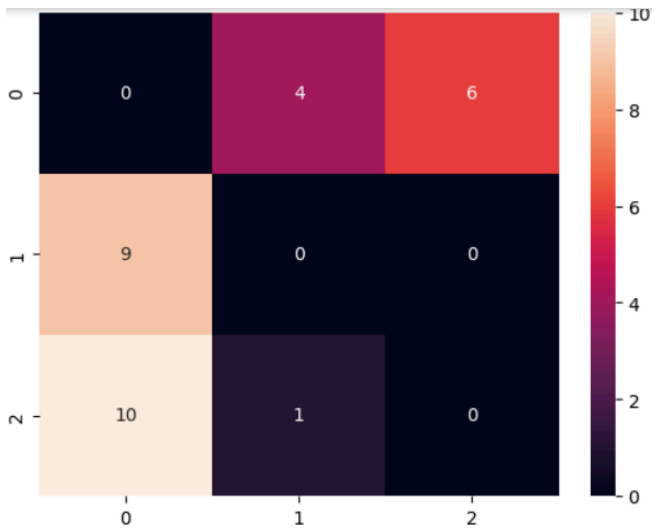
Confusion Matrix Batch Size 4, Learning Rate: 0.001, Epochs: 5



Confusion Matrix Batch Size 4, Learning Rate: 0.00001, Epochs: 1



Confusion Matrix Batch Size 4, Learning Rate: 0.00001, Epochs: 3



Confusion Matrix Batch Size 4, Learning Rate: 0.00001, Epochs: 5

The following is the metrics dataframe which shows the accuracy and F1 score of all the possibilities.

	batch	lr	epochs	accuracy	f1
0	2	0.00100	1	0.600000	0.515556
1	2	0.00100	3	0.666667	0.564935
2	2	0.00100	5	0.666667	0.564935
3	2	0.00001	1	0.000000	0.000000
4	2	0.00001	3	0.000000	0.000000
5	2	0.00001	5	0.000000	0.000000
6	4	0.00100	1	0.333333	0.223077
7	4	0.00100	3	0.666667	0.559829
8	4	0.00100	5	0.700000	0.581191
9	4	0.00001	1	0.000000	0.000000
10	4	0.00001	3	0.000000	0.000000
11	4	0.00001	5	0.000000	0.000000

Metrics(accuracy/learning rate/epoch/f1 score)

For automated search, we were able to do Random Search and Bayes Search on the current version of Python 3.11.11 and for the rest, we revert the python version to 3.9 as the AutoGluon library was updated from version 0.1.0 to 1.2.0 where we do not have the gridsearch and hyperband search functionality. So we use the old and deprecated version of autogluon 0.1.0. The documentation for that is in the second file.