Training Setup:

Neural Network Architecture:

I used a MLPModel class that inherits from nn.Module which utilizes nn.Linear inside its constructor to initialize layers in neural network.

As I decided to create a hyperparameter related to hidden layer width (number of layers), I implemented a conditional check where if the constructor takes this value as 2, creates 3 layers instead of 2 (extra layer comes from inbetween hidden layers)

I adjusted the forward function accordingly to the given hidden layer size.

Seperation of input data is an important aspect of model training and it was readily provided as train,validation and test data separately.

I trained the model for each 32 hyperparameter configuraions with 10 iterations for each and averaged the accuracy scores. After finding the best configuration by validation accuracy score, I performed another model training with these configurations by feeding an input combination of training and validation inputs (again with 10 iterations). Lastly I averaged the scores and found the test accuracy results for this optimal model.

Hyperparameters:

lr = learning rate, either set to 0.001 or 0.0005

hiddenLayerSize = Determines whether there will be 1 or 2 hidden layers.

hiddenLayerNodeSize = Determines # of nodes in both hidden layers.

epochs = determines how many iterations will be performed while performing weight updates (number of epochs)

activation = whether to use nn.LeakyRelU or nn.Sigmoid

Accuracy performance results of 32 hyperparameter configurations on validation dataset – Also the confidence intervals (Each configuration is clearly indicated at the top)

```
Configuration --> Rate: 0.0010 - Hidden Layer Size: 1 - Hidden Layer Node
Size: 32 - Activation Function: LeakyReLU(negative_slope=0.01) - Epoch Size -
100
Average Train Accuracy : 88.79 - Average Validation Accuracy : 83.75
Test accuracy : 84.21
Confidence interval for this configuration: (83.54888343431404,
83.95811936680641)
------------------------------------------------------------------------------
------------------------
Configuration --> Rate: 0.0005 - Hidden Layer Size: 1 - Hidden Layer Node
Size: 32 - Activation Function: LeakyReLU(negative_slope=0.01) - Epoch Size -
100
Average Train Accuracy : 84.57 - Average Validation Accuracy : 81.19
Test accuracy : 82.13
Confidence interval for this configuration: (80.44650993040752,
81.92844004958447)
------------------------------------------------------------------------------
------------------------
Configuration --> Rate: 0.0010 - Hidden Layer Size: 1 - Hidden Layer Node
Size: 64 - Activation Function: LeakyReLU(negative_slope=0.01) - Epoch Size -
100
Average Train Accuracy : 90.51 - Average Validation Accuracy : 83.39
Test accuracy : 83.50
Confidence interval for this configuration: (83.13293447680566,
83.6537802090687)
------------------------------------------------------------------------------
------------------------
Configuration --> Rate: 0.0005 - Hidden Layer Size: 1 - Hidden Layer Node
Size: 64 - Activation Function: LeakyReLU(negative_slope=0.01) - Epoch Size -
100
Average Train Accuracy : 87.35 - Average Validation Accuracy : 82.74
Test accuracy : 83.20
Confidence interval for this configuration: (82.51987763935426,
82.96631683843685)
------------------------------------------------------------------------------
------------------------
Configuration --> Rate: 0.0010 - Hidden Layer Size: 2 - Hidden Layer Node
Size: 32 - Activation Function: LeakyReLU(negative_slope=0.01) - Epoch Size -
100
Average Train Accuracy : 86.35 - Average Validation Accuracy : 80.73
Test accuracy : 81.19
Confidence interval for this configuration: (80.32305698467448,
81.14152884965924)
------------------------------------------------------------------------------
------------------------
Configuration --> Rate: 0.0005 - Hidden Layer Size: 2 - Hidden Layer Node
Size: 32 - Activation Function: LeakyReLU(negative_slope=0.01) - Epoch Size -
100
Average Train Accuracy : 80.17 - Average Validation Accuracy : 74.76
Test accuracy : 75.91
```

```
Confidence interval for this configuration: (73.6519082368943,
75.87790368787559)
-------------------------------------------------------------------------
-----------------------
Configuration --> Rate: 0.0010 - Hidden Layer Size: 2 - Hidden Layer Node
Size: 64 - Activation Function: LeakyReLU(negative_slope=0.01) - Epoch Size -
100
Average Train Accuracy : 88.27 - Average Validation Accuracy : 82.33
Test accuracy : 83.12
Confidence interval for this configuration: (82.08300966741673,
82.57285267752125)
-------------------------------------------------------------------------
-----------------------
Configuration --> Rate: 0.0005 - Hidden Layer Size: 2 - Hidden Layer Node
Size: 64 - Activation Function: LeakyReLU(negative_slope=0.01) - Epoch Size -
100
Average Train Accuracy : 83.41 - Average Validation Accuracy : 79.74
Test accuracy : 80.47
Confidence interval for this configuration: (79.48767443700366,
79.99611908040329)
-------------------------------------------------------------------------
-----------------------
Configuration --> Rate: 0.0010 - Hidden Layer Size: 1 - Hidden Layer Node
Size: 32 - Activation Function: Sigmoid() - Epoch Size - 100
Average Train Accuracy : 79.34 - Average Validation Accuracy : 77.75
Test accuracy : 78.46
Confidence interval for this configuration: (77.70769662915414,
77.79450425119798)
-------------------------------------------------------------------------
-----------------------
Configuration --> Rate: 0.0005 - Hidden Layer Size: 1 - Hidden Layer Node
Size: 32 - Activation Function: Sigmoid() - Epoch Size - 100
Average Train Accuracy : 67.80 - Average Validation Accuracy : 66.63
Test accuracy : 67.21
Confidence interval for this configuration: (64.97920757001566,
68.28409775211321)
-------------------------------------------------------------------------
-----------------------
Configuration --> Rate: 0.0010 - Hidden Layer Size: 1 - Hidden Layer Node
Size: 64 - Activation Function: Sigmoid() - Epoch Size - 100
Average Train Accuracy : 84.29 - Average Validation Accuracy : 80.32
Test accuracy : 81.29
Confidence interval for this configuration: (80.0741070742725,
80.57015062880872)
-------------------------------------------------------------------------
-----------------------
Configuration --> Rate: 0.0005 - Hidden Layer Size: 1 - Hidden Layer Node
Size: 64 - Activation Function: Sigmoid() - Epoch Size - 100
Average Train Accuracy : 77.74 - Average Validation Accuracy : 76.26
Test accuracy : 77.47
Confidence interval for this configuration: (76.0291823291232,
76.48182207263751)
-------------------------------------------------------------------------
-----------------------
Configuration --> Rate: 0.0010 - Hidden Layer Size: 2 - Hidden Layer Node
Size: 32 - Activation Function: Sigmoid() - Epoch Size - 100
Average Train Accuracy : 70.77 - Average Validation Accuracy : 64.67
```

```
Test accuracy : 65.14
Confidence interval for this configuration: (63.264543304973806,
66.06718938810343)
--------------------------------------------------------------------------------
------------------------
Configuration --> Rate: 0.0005 - Hidden Layer Size: 2 - Hidden Layer Node
Size: 32 - Activation Function: Sigmoid() - Epoch Size - 100
Average Train Accuracy : 63.86 - Average Validation Accuracy : 56.72
Test accuracy : 58.19
Confidence interval for this configuration: (52.37610742900666,
61.069270722253854)
--------------------------------------------------------------------------------
------------------------
Configuration --> Rate: 0.0010 - Hidden Layer Size: 2 - Hidden Layer Node
Size: 64 - Activation Function: Sigmoid() - Epoch Size - 100
Average Train Accuracy : 77.28 - Average Validation Accuracy : 74.59
Test accuracy : 75.55
Confidence interval for this configuration: (73.45513630337216,
75.72453556537535)
--------------------------------------------------------------------------------
------------------------
Configuration --> Rate: 0.0005 - Hidden Layer Size: 2 - Hidden Layer Node
Size: 64 - Activation Function: Sigmoid() - Epoch Size - 100
Average Train Accuracy : 70.73 - Average Validation Accuracy : 65.65
Test accuracy : 66.15
Confidence interval for this configuration: (65.31762964852112,
65.97488735828159)
--------------------------------------------------------------------------------
------------------------
Configuration --> Rate: 0.0010 - Hidden Layer Size: 1 - Hidden Layer Node
Size: 32 - Activation Function: LeakyReLU(negative_slope=0.01) - Epoch Size -
200
Average Train Accuracy : 91.80 - Average Validation Accuracy : 82.08
Test accuracy : 82.51
Confidence interval for this configuration: (81.46707750593026,
82.68858475897571)
--------------------------------------------------------------------------------
------------------------
Configuration --> Rate: 0.0005 - Hidden Layer Size: 1 - Hidden Layer Node
Size: 32 - Activation Function: LeakyReLU(negative_slope=0.01) - Epoch Size -
200
Average Train Accuracy : 89.12 - Average Validation Accuracy : 83.44
Test accuracy : 83.79
Confidence interval for this configuration: (82.64160539066626,
84.23514530961384)
--------------------------------------------------------------------------------
------------------------
Configuration --> Rate: 0.0010 - Hidden Layer Size: 1 - Hidden Layer Node
Size: 64 - Activation Function: LeakyReLU(negative_slope=0.01) - Epoch Size -
200
Average Train Accuracy : 93.36 - Average Validation Accuracy : 80.55
Test accuracy : 81.08
Confidence interval for this configuration: (80.30229738250276,
80.79214039260728)
--------------------------------------------------------------------------------
------------------------
Configuration --> Rate: 0.0005 - Hidden Layer Size: 1 - Hidden Layer Node
```

```
Size: 64 - Activation Function: LeakyReLU(negative_slope=0.01) - Epoch Size -
200
Average Train Accuracy : 90.80 - Average Validation Accuracy : 84.17
Test accuracy : 84.51
Confidence interval for this configuration: (84.09926293460668,
84.24807600096756)
--------------------------------------------------------------------------------
------------------------
Configuration --> Rate: 0.0010 - Hidden Layer Size: 2 - Hidden Layer Node
Size: 32 - Activation Function: LeakyReLU(negative_slope=0.01) - Epoch Size -
200
Average Train Accuracy : 90.59 - Average Validation Accuracy : 81.16
Test accuracy : 81.42
Confidence interval for this configuration: (80.74512678048586,
81.56979918990231)
--------------------------------------------------------------------------------
------------------------
Configuration --> Rate: 0.0005 - Hidden Layer Size: 2 - Hidden Layer Node
Size: 32 - Activation Function: LeakyReLU(negative_slope=0.01) - Epoch Size -
200
Average Train Accuracy : 87.86 - Average Validation Accuracy : 81.74
Test accuracy : 82.44
Confidence interval for this configuration: (81.5919822838862,
81.88340787217622)
--------------------------------------------------------------------------------
------------------------
Configuration --> Rate: 0.0010 - Hidden Layer Size: 2 - Hidden Layer Node
Size: 64 - Activation Function: LeakyReLU(negative_slope=0.01) - Epoch Size -
200
Average Train Accuracy : 91.58 - Average Validation Accuracy : 78.93
Test accuracy : 79.59
Confidence interval for this configuration: (78.37352363019839,
79.48962162790487)
--------------------------------------------------------------------------------
------------------------
Configuration --> Rate: 0.0005 - Hidden Layer Size: 2 - Hidden Layer Node
Size: 64 - Activation Function: LeakyReLU(negative_slope=0.01) - Epoch Size -
200
Average Train Accuracy : 88.78 - Average Validation Accuracy : 83.29
Test accuracy : 83.74
Confidence interval for this configuration: (83.23131188261375,
83.3553227712478)
--------------------------------------------------------------------------------
------------------------
Configuration --> Rate: 0.0010 - Hidden Layer Size: 1 - Hidden Layer Node
Size: 32 - Activation Function: Sigmoid() - Epoch Size - 200
Average Train Accuracy : 86.58 - Average Validation Accuracy : 82.56
Test accuracy : 83.12
Confidence interval for this configuration: (82.43281377701827,
82.69323664314977)
--------------------------------------------------------------------------------
------------------------
Configuration --> Rate: 0.0005 - Hidden Layer Size: 1 - Hidden Layer Node
Size: 32 - Activation Function: Sigmoid() - Epoch Size - 200
Average Train Accuracy : 80.84 - Average Validation Accuracy : 76.79
Test accuracy : 78.41
Confidence interval for this configuration: (76.72871084219759,
```

```
76.85272173083163)
--------------------------------------------------------------------------------
-----------------------
Configuration --> Rate: 0.0010 - Hidden Layer Size: 1 - Hidden Layer Node
Size: 64 - Activation Function: Sigmoid() - Epoch Size - 200
Average Train Accuracy : 89.27 - Average Validation Accuracy : 84.23
Test accuracy : 84.26
Confidence interval for this configuration: (84.1096825887569,
84.35770436602502)
--------------------------------------------------------------------------------
-----------------------
Configuration --> Rate: 0.0005 - Hidden Layer Size: 1 - Hidden Layer Node
Size: 64 - Activation Function: Sigmoid() - Epoch Size - 200
Average Train Accuracy : 85.52 - Average Validation Accuracy : 82.57
Test accuracy : 83.35
Confidence interval for this configuration: (82.57302921168468,
82.57302921168468)
--------------------------------------------------------------------------------
-----------------------
Configuration --> Rate: 0.0010 - Hidden Layer Size: 2 - Hidden Layer Node
Size: 32 - Activation Function: Sigmoid() - Epoch Size - 200
Average Train Accuracy : 83.46 - Average Validation Accuracy : 79.69
Test accuracy : 80.97
Confidence interval for this configuration: (79.45625606229557,
79.92749743910498)
--------------------------------------------------------------------------------
-----------------------
Configuration --> Rate: 0.0005 - Hidden Layer Size: 2 - Hidden Layer Node
Size: 32 - Activation Function: Sigmoid() - Epoch Size - 200
Average Train Accuracy : 69.84 - Average Validation Accuracy : 67.09
Test accuracy : 67.91
Confidence interval for this configuration: (66.3322700418103,
67.85140342757747)
--------------------------------------------------------------------------------
-----------------------
Configuration --> Rate: 0.0010 - Hidden Layer Size: 2 - Hidden Layer Node
Size: 64 - Activation Function: Sigmoid() - Epoch Size - 200
Average Train Accuracy : 86.12 - Average Validation Accuracy : 81.06
Test accuracy : 81.64
Confidence interval for this configuration: (80.68419175965414,
81.44065818032186)
--------------------------------------------------------------------------------
-----------------------
Configuration --> Rate: 0.0005 - Hidden Layer Size: 2 - Hidden Layer Node
Size: 64 - Activation Function: Sigmoid() - Epoch Size - 200
Average Train Accuracy : 78.41 - Average Validation Accuracy : 74.87
Test accuracy : 76.22
Confidence interval for this configuration: (74.42541050869356,
75.32448945129046)
--------------------------------------------------------------------------------
-----------------------
Best Hyper Parameter Config:
(Sigmoid(), 1, 64, 0.001, 200)
```

Final test accuracy results: (88.16%)

Model is trained on combination of (torch.cat) train and validation data with best configuration of hyperparameters.

```
Best Configuration --> Rate: 0.0010 - Hidden Layer Size: 1 - Hidden Layer
Node Size: 64 - Activation Function: Sigmoid() - Epoch Size - 200
Average Train Accuracy on combined Input : 89.42
Test accuracy : 88.16
Confidence interval for this configuration: (89.31698857884248,
89.52301142115749)
--------------------------------------------------------------------------
------------------------
```

A1: In order to prevent overfitting, I paid extreme care on re-training data in every 10 iteration. At first, I forgot to do this and faced with training results close to %100, this gave me a good intuition about the cause of the problem.

I also choose reasonable options for my hyperparameters. Choosing huge numbers of epochs, hidden layer size, nodes in hidden layers can easily cause overfitting and takes us away from generalization. Hence, I kept them acceptable like layer size (1 or 2), # of nodes (32 or 64) epoch (100 or 200)

A2: Once can easily detect the model is overfitted by looking at training accuracy. A training accuracy close to 100% can indicate an overfitting however not necessarily. Further, we need to inspect test accuracy and validation accuracy. For example, a training accuracy close to 100% and a very low test accuracy (for example 30%) means that our model is not still general.

A3: Using a bigger number of epochs can be a good idea if we don't want to deal with data too much. However, this is not something that a good data scientist / ml engineer should do. Instead we need to focus on feature engineering and more importantly utilize a technique called **"early stopping".** It is basically stopping the iteration if validation accuracy does not improve after some point.

A4: In my model, I used two different learning rates (0.001 and 0.0005). Setting the other hyperparameters constant and only changing learning rate didn't specifically give me a general result of one of them is better since in some cases one performed better than other and other cases visa versa.

A5: In my model, , I used two activation functions (Sigmoid and LeakyRelU). Setting the other hyperparameters constant and only changing activation function didn't specifically give me a

general result that one of them is better, However, I can say that LeakyRelU provided much stable results (closer validation scores in different hyperparameter configurations) than Sigmoid.

A6: Using a smaller learning rate will make it more likely to find (don't miss by jumping over) the minimum loss score. However, it will cause too many iterations which means computationally ineffective.

A7: As oppose to answer above, a higher learning rate will find the local minimum much faster; However, it may miss the real minimum score by jumping over it.

A8: It is definitely a good idea to use stochastic gradient descent learning with a very large dataset since it can reduce the calculations that otherwise would be too much with ordinary gradient descent. However, since we are sampling the data and using this sample as a way to update weights, it is more prone to catch noisy data (especially by chance the big portion of noise is inside the sample we choose). This would create a very bad generalization compared to ordinary gradient descent.

A9: Considering the sigmoid function, values after a point (around -5 and 5) are scaled really close to 0 and 1 respectively. This means that if the difference in input is actually meaningfull after these values, we lose that aspect (precision of input is lost hence generalization will produce bad results). Hence, we need to scale them with 255 in order to make sure they are scaled around 0 with still preserving their relative weight effects over each other.

Another reason is that since most of the data is out of the range of (-5,5), they will be either 0 or most likely 1s after sigmoid function. This means that we will have huge gradient values later on when finding the good approximation for the weights.