PROJECT 4: NEURAL NETWORKS

File used: 2.csv

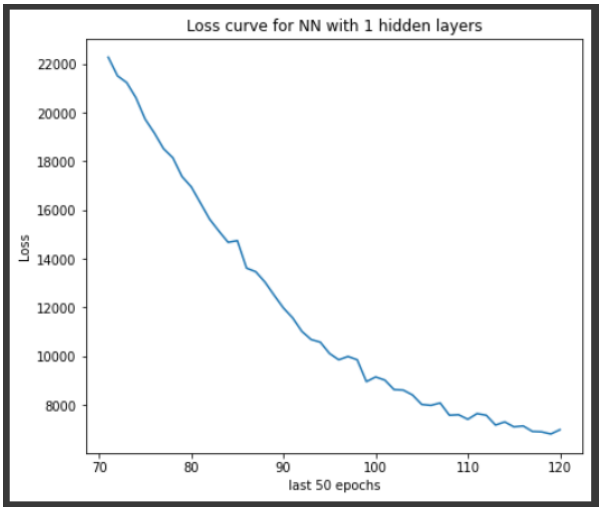
Task 1: Automatic Grid search

While performing grid search for each hidden layer few arguments for the MLPRegressor is fixed. These are: -
Max epochs = 200
Batch size = 32
I am also using early stopping to stop the weight update once the validation score is no longer improving.

The following hyper-parameters are varied using grid search.
Hidden layer size: typically, from 5-25 (10-25, 15-25 for higher layers)
Alpha (regularization coefficient): from 5e-5 to 0.5
Initial learning rate: from 1e-5 to 1

Grid Search for hidden layer 1:

```
Best parameters found:  {'alpha': 5e-05, 'hidden_layer_sizes': 22, 'learning_rate_init': 0.1}
Best score:  0.9892187162744086
```

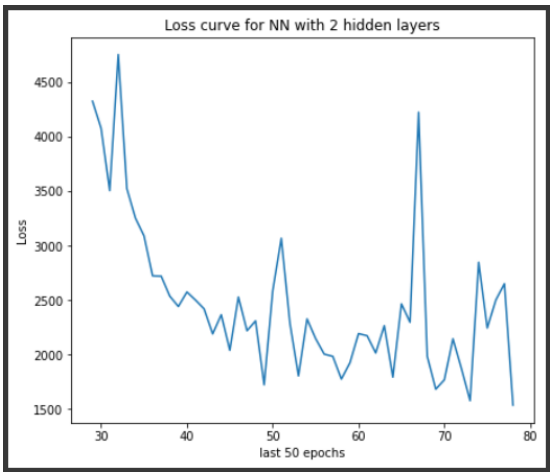



```
MSE loss for NN with 1 hidden layer: 12908.736466393017
```

From the MSE curve for the last 50 epochs we can see that the error is gradually decreasing, indicating that the model is not overfitting the training data. Also, it goes on for approx. 120 epochs. An MSE of 12908.736 is achieved for train set for NN with 1 hidden layer with 22 units. The model has a score of 0.989. R-square value is the scoring criterion for MLPRegressor.

Grid Search for hidden layer 2:

```
Best parameters found:  {'alpha': 0.0005, 'hidden_layer_sizes': (22, 19), 'learning_rate_init': 0.1}
Best score:  0.9988508161482812
```

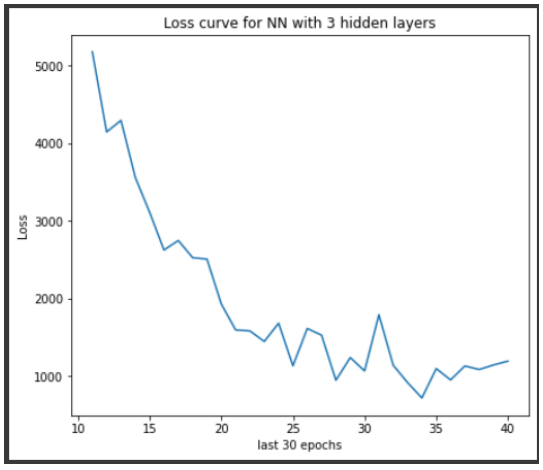



```
MSE loss for NN with 2 hidden layers: 2666.791376367218
```

For hyperparameter tuning for the second layer, we fix the first layer to 22 units. The R2 value for 2-layer NN is better than that for 1-layer NN by 0.01. Also, the MSE if much lower when compared to the previous model. The MSE curves shows that although for some epochs the error increases, the general trend of the error is decreasing.

Grid Search for hidden layer 3:

```
Best parameters found:  {'alpha': 0.005, 'hidden_layer_sizes': (22, 19, 21), 'learning_rate_init': 0.1}
Best score:  0.9993976980697162
```

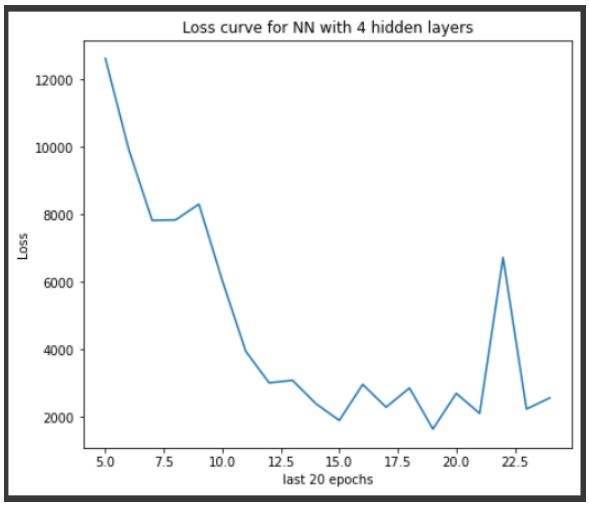



```
MSE loss for NN with 3 hidden layers: 1273.0927515551098
```

We fix the first 2 layers as 22 and 19 units. The R2 value does increase by a significant amount (only about 0.001, but there is not much scope to improve beyond 0.999). The MSE however reduces by a significant amount. From the curve we can notice two things; First, the total number of epochs before early stopping has reduced significantly (half when compared to NN with 2 layers, we compared to drop from 1 layer to 2 layer model, the drop was 1/3). Second, just before early stopping we can see that the MSE curve has started to flatten (this might that if trained for further epochs or if more layers are added the performance of the model will decrease)

Grid Search for hidden layer 4:

```
Best parameters found:  {'alpha': 0.0005, 'hidden_layer_sizes': (22, 19, 21, 20), 'learning_rate_init': 0.1}
Best score:  0.9992568210560183
```




```
MSE loss for NN with 4 hidden layers: 3557.31942629588
```

The first 3 layers are fixed to 22, 19, and 21 hidden units. The best score for this model actually decreases (although not by a very significantly). The MSE on the other hand, increases by a very significant amount. From the MSE curve, we can see that it only takes about 24 epochs for the model to reach early stopping stage. Also, the curve seems to flatten out. We can confidently say that this model with 4 hidden layers performs worse than the 3-layer model.

Since, the R2 value for all the models are pretty good (close to 1) we can compare the MSE for each model to determine the best one. Since, NN model with three hidden layer has the least MSE, and the highest score, we can concur that our best model has the following hyperparameters.

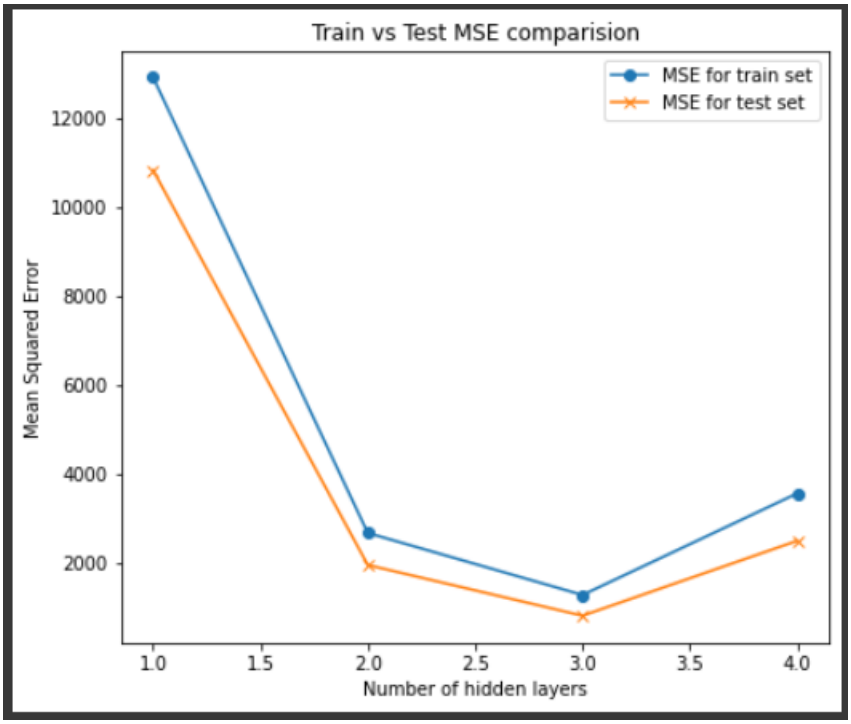
```
MLPRegressor(activation='relu', alpha=0.005, batch_size=32, beta_1=0.9,
             beta_2=0.999, early_stopping=True, epsilon=1e-08,
             hidden_layer_sizes=(22, 19, 21), learning_rate='constant',
             learning_rate_init=0.1, max_fun=15000, max_iter=200, momentum=0.9,
             n_iter_no_change=10, nesterovs_momentum=True, power_t=0.5,
             random_state=10, shuffle=True, solver='adam', tol=0.0001,
             validation_fraction=0.1, verbose=False, warm_start=False)
```

Notable hyperparameters:
Hidden layers size: (22, 19, 21)
Alpha: 0.005
Learning rate: 0.1
Batch size: 32

Comparing MSE for test data:

```
MSE for 1 hidden layer nn: 10823.803917418338
MSE for 2 hidden layer nn: 1946.5519213110208
MSE for 3 hidden layer nn: 806.8016977232735
MSE for 4 hidden layer nn: 2491.775371158666
```

We can see that model 3 performs best on the test set as well. We can say that the model is not overfitting our train set.



All the models give a lower MSE on test data than the training set. We can conclude that none of the models overfit. From the graph, we can easily see that NN with 3 hidden layers performs the best on both the train and test set.

Task 2: Neural net VS Multivariable regression

For the given dataset, the best multivariable regression model was obtained by removing X3 and X4 columns due to their low correlation with the output variable, Y. Therefore, those variables are removed to generate the multivariable regression model. Thus, our regression model has three independent variables, X1, X2 and X5 (gets renamed to X3).

Regression model summary:

Equation: 395.234 + 1.136e+4*X1 + 757.44*X2 + 783.67*X5

OLS Regression Results

Dep. Variable:	Y	R-squared:	0.964
Model:	OLS	Adj. R-squared:	0.964
Method:	Least Squares	F-statistic:	1.779e+04
Date:	Tue, 03 Nov 2020	Prob (F-statistic):	0.00
Time:	16:12:17	Log-Likelihood:	-14352.
No. Observations:	2000	AIC:	2.871e+04
Df Residuals:	1996	BIC:	2.873e+04
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	395.2340	41.702	9.478	0.000	313.450	477.018
x1	1.136e+04	49.395	229.982	0.000	1.13e+04	1.15e+04
x2	757.4449	50.450	15.014	0.000	658.504	856.386
x3	783.6737	47.406	16.531	0.000	690.704	876.643

Omnibus:	1131.966	Durbin-Watson:	1.893
Prob(Omnibus):	0.000	Jarque-Bera (JB):	10600.072
Skew:	2.533	Prob(JB):	0.00
Kurtosis:	13.076	Cond. No.	11.9

From the p values of the coefficients and constant we can conclude that null hypothesis is rejected for all these values ($p < 0.05$ for CI=95%). Therefore, they all have to be non-zero. The R2 value is close to 1 indicating a good regression model.

```
MSE loss for Regression model: 96291.56952153226
MSE loss for the best NN mode(with 3 hidden layers: 806.8016977232735
```

```
SSE loss for Regression model: 26783828.731765114
SSE loss for the best NN mode(with 3 hidden layers: 242040.50931698194
```

Comparing the MSE (or SSE) for multivariable regression model and 3-layer Neural network model for the test set we can clearly see that the NN model performs more than x100 folds better than the regression model. Regression model, in fact performs worse than every NN model. This is expected because the regression model tries to fit a linear model through the dataset and predict the output variable. As the number of independent variable increase in the dataset, it becomes harder and harder to fit a linear model which would capture the increasing complexity of dataset. Neural network models on the other hand are much more well equipped to deal with multi-variable datasets by adjusting its weights across multiple epochs and adjust to the data. Therefore, although the regression model seems to be a good fit, the results are incomparable when compared to neural network model.