BUAN 6341.501 - Applied Machine Learning
Assignment 3 - Pavan Gorantla (PCG180000)

Objective
Experimentation of classification algorithms like neural networks and K nearest neighbors to predict the class labels output of 2 datasets that are downloaded from archives of UCI Machine Learning Repository.

Dataset 1 - Appliances Energy Prediction (Link: appliances+energy+prediction)
Dataset consists of 19735 observations on a total of 29 variables. More information about these variables can be found in the above given weblink. The response variable is created using Appliances variable where appliances greater than median are classified as Heavy Appliances and the goal is to predict this classifier.

Dataset 2 - Online Shoppers Purchasing Intention (Link: online+shoppers+purchasing+intention)
Dataset consists of 12330 observations on a total of 18 variables. More information about these variables can be found in the above given weblink. The response variable Revenue specifies if an online user on the shopping website has purchase intent or not based on the pages visited and duration spent on each page.

Test/Train and Normalization
Both datasets are split into train and test sets using 80-20 split percentage and features are normalized to a range of [0,1]. Target from Dataset 1 is classified as 1 for Heavy Appliances and 0 for the remaining. Response from Dataset 2 is classified into 1 for users with purchase intent and 0 for the remaining users. All features in Dataset 1 are numeric and hence they are just normalized for our classification algorithm. There are few categorical features in Dataset 2 which are transformed into dummies before normalizing.

Experiments and Parameter Tuning
Using 2 algorithms like Neural Nets (using keras) and KNN on both datasets with 3 fold cross-validation.

Neural Nets: Four experiments are done and parameter tuning is done for *Hidden Activation Function*, *Batch Size* and *Epochs*. Each experiment has combination of 2 and 3 layers with larger and smaller nodes. Other parameters like optimizer and output activation function are fixed to *SGD* and *Sigmoid* respectively.

Experiment 1: Network of 2 hidden layers with layer nodes as 20, 15 and tuning for the below parameters.
Experiment 2: Network of 2 hidden layers with layer nodes as 15, 10 and tuning for the below parameters.
Experiment 3: Network of 3 hidden layers with layer nodes as 20, 15, 10 and tuning for below parameters.
Experiment 4: Network of 3 hidden layers with layer nodes as 15, 10, 5 and tuning for below parameters.

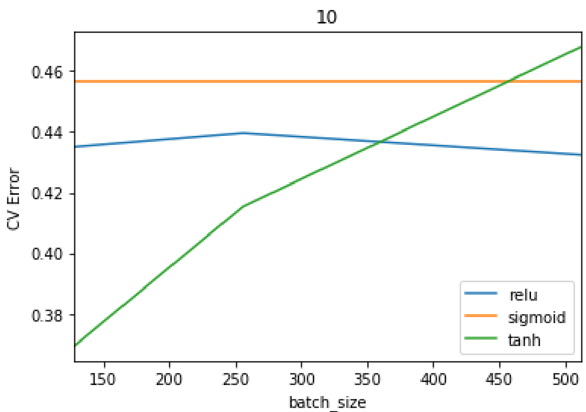
Experiment	Hidden Layers	Nodes	Hidden Activation	Batch Size	Epochs
1	2	20, 15	Sigmoid, Tanh, ReLU	128, 256, 512	10, 20
2	2	15, 10	Sigmoid, tanh, ReLU	128, 256, 512	10, 20
3	3	20, 15, 10	Sigmoid, tanh, ReLU	128, 256, 512	10, 20
4	3	15, 10, 5	Sigmoid, tanh, ReLU	128, 256, 512	10, 20

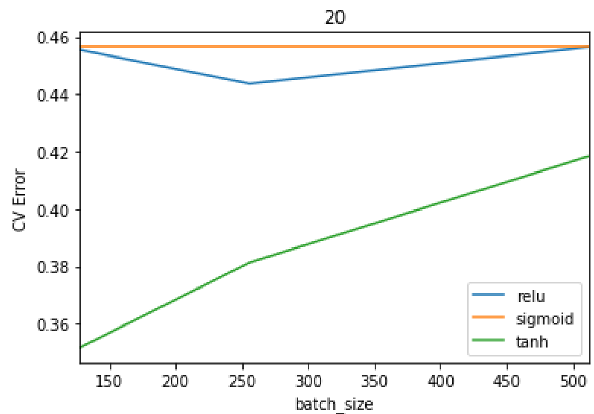
KNN: Experiment is performed with number of neighbors from 1 to 10 and tuning for few parameters like *Weights* and *Exponent (p)* for distance. *Algorithm*, *Metric* are fixed to *KD Tree* and *Minkowski* respectively. The distance metric is chosen as *Minkowski* so that we can experiment with the values of the exponent p.

Neighbors	Metric	Algorithm	Weights	Exponent (p)
1 to 10	Minkowski	KD Tree	Uniform, Distance	1,2,3

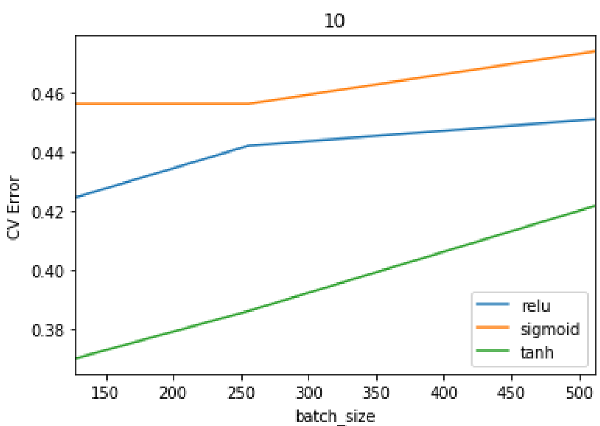
Neural Networks with 2 Hidden Layers (Dataset 1 - Heavy Appliances Prediction)

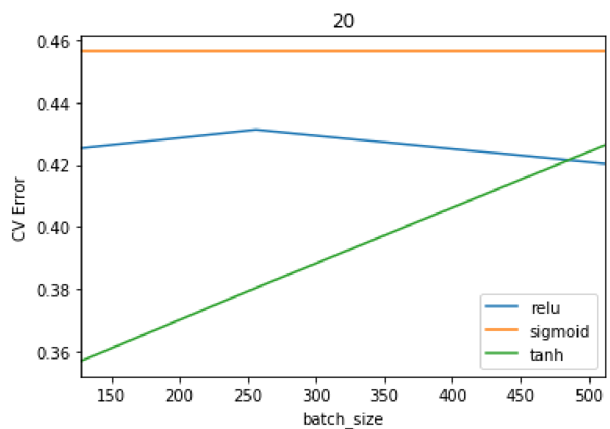
Experiment 1: Following plots compare the error results of using various parameters mentioned before. We can observe that Tanh as Hidden Activation does well for all Batch Sizes when number of Epochs is 20.



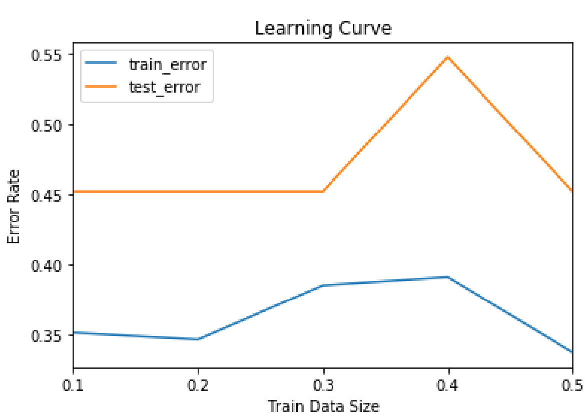


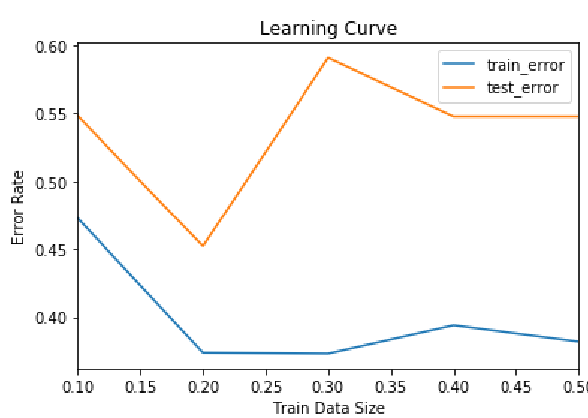
Experiment 2: Following plots compare the error results of using various parameters mentioned before. We can observe that Tanh as Hidden Activation does well for all Batch Sizes when number of Epochs is 10.





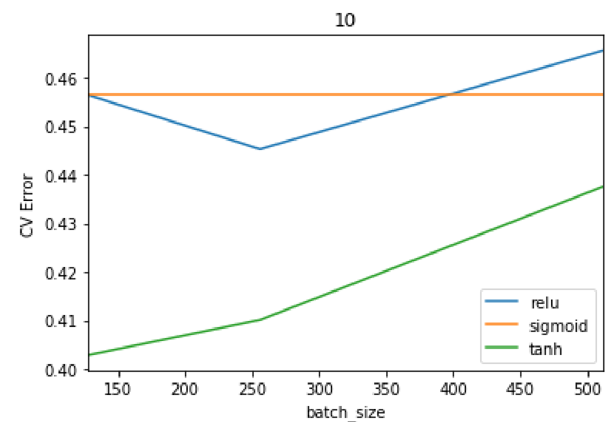
Using above found parameters as best ones, below learning curves are plotted for varying train data size. The left plot is for Experiment 1 with larger nodes and the right plot is for Experiment 2 with smaller nodes. We see that test error is comparatively better for experiment with larger number of hidden layers nodes.

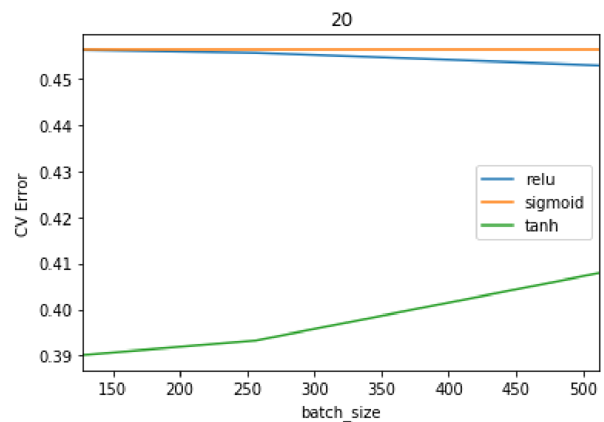




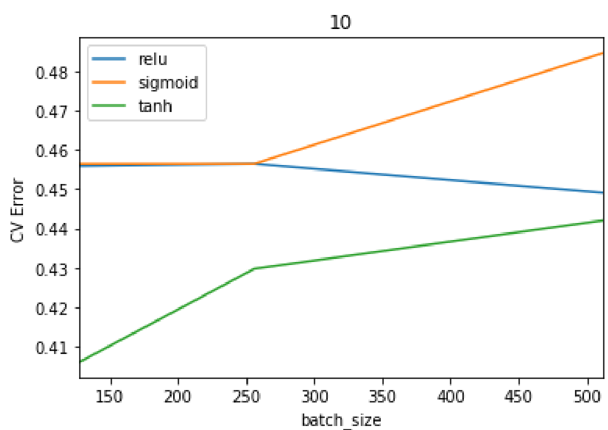
Neural Networks with 3 Hidden Layers (Dataset 1 - Heavy Appliances Prediction)

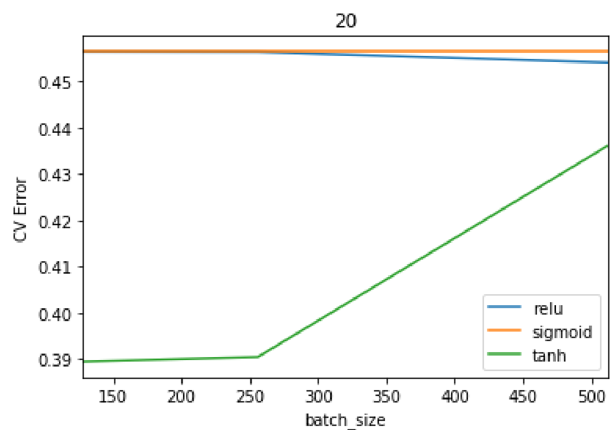
Experiment 3: Following plots compare the error results of using various parameters mentioned before. We can observe that Tanh as Hidden Activation does well for all Batch Sizes irrespective of Epochs number.



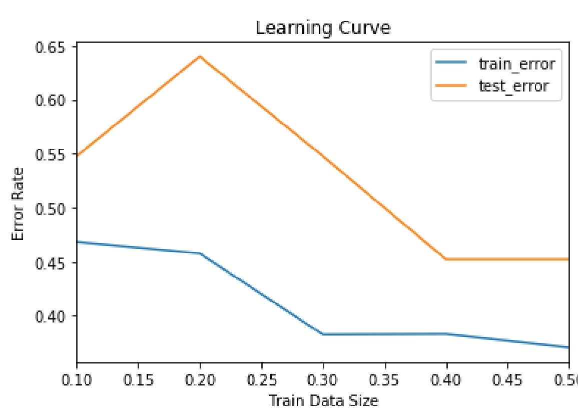


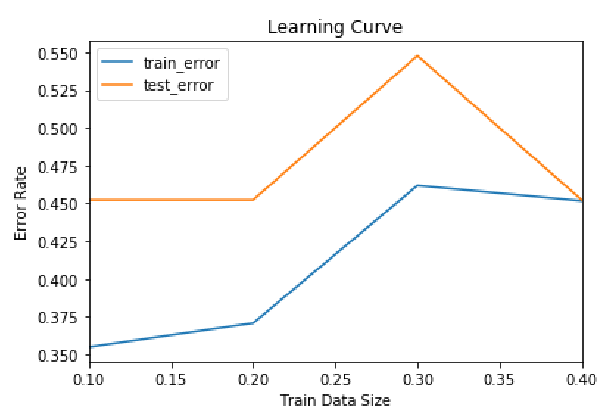
Experiment 4: Following plots compare the error results of using various parameters mentioned before. We can observe that Tanh as Hidden Activation does well for all Batch Sizes irrespective of Epochs number.





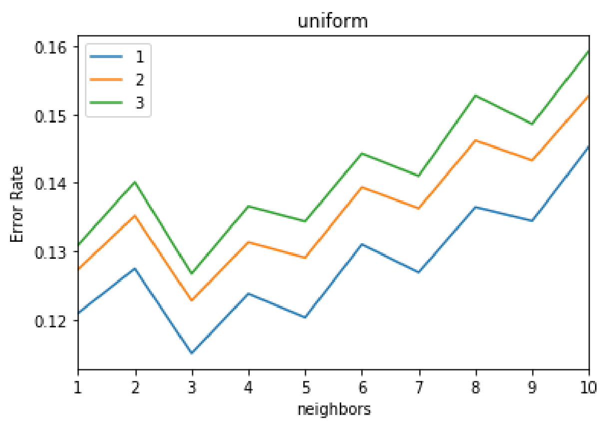
Using above found parameters as best ones, below learning curves are plotted for varying train data size. The left plot is for Experiment 1 with larger nodes and the right plot is for Experiment 2 with smaller nodes. We see that test error is comparatively better for experiment with larger number of hidden layers nodes.

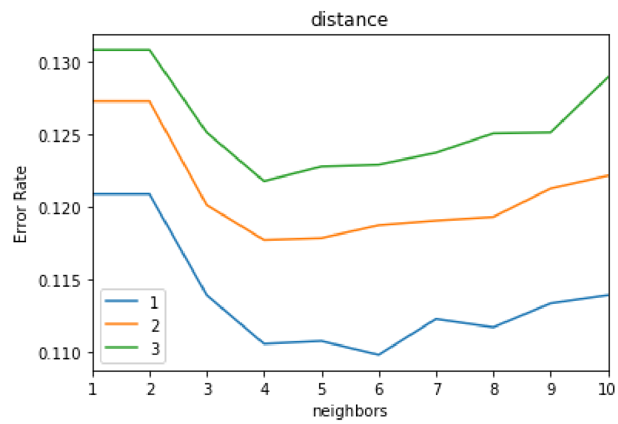




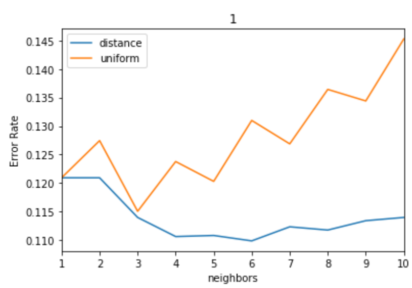
K - Nearest Neighbors with Minkowski Metric (Dataset 1 - Heavy Appliances Prediction)

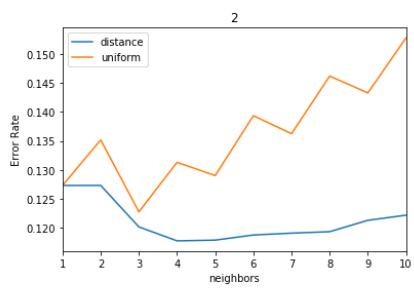
Following plots compare the error results of using various parameters mentioned before. We can observe that p = 1 as distance exponent does well for all neighbors from 1 to 10 with best error for Distance weights

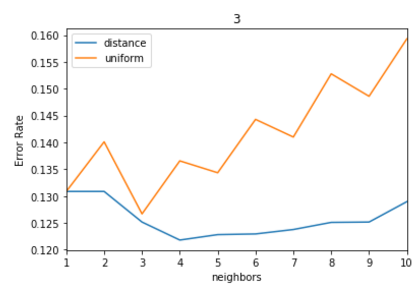




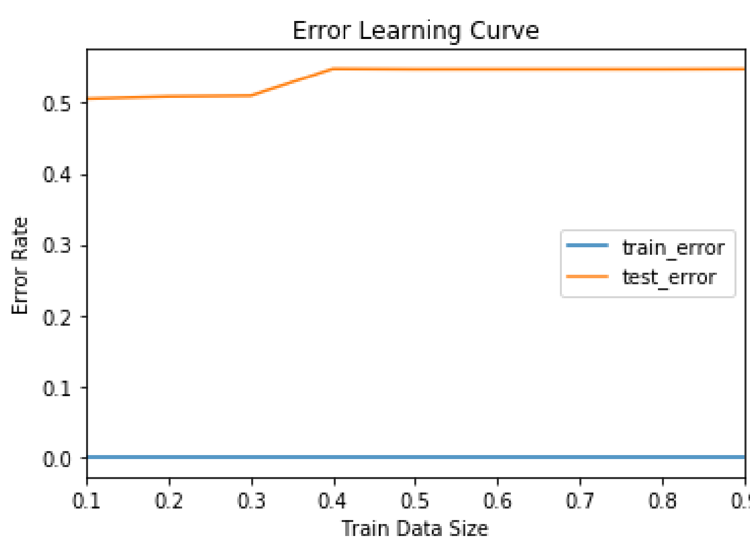
Experiment 2: Following plots compare the error results of using various parameters mentioned before. We see that Distance is best performing weights parameter for all neighbors from 1 to 10 and for p = 1,2,3





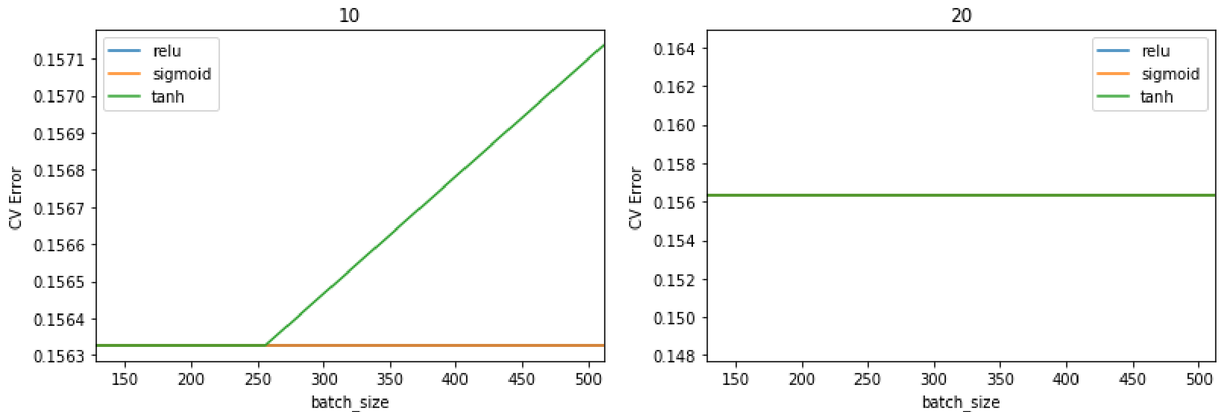


All the above plots indicate best neighbors are 6 as we see least error when weights = distance and p = 1. Using above found parameters as best ones, below learning curve is plotted for varying train data size. We see that train error is zero and test error is above 0.5, which could be because of an overfitting model.

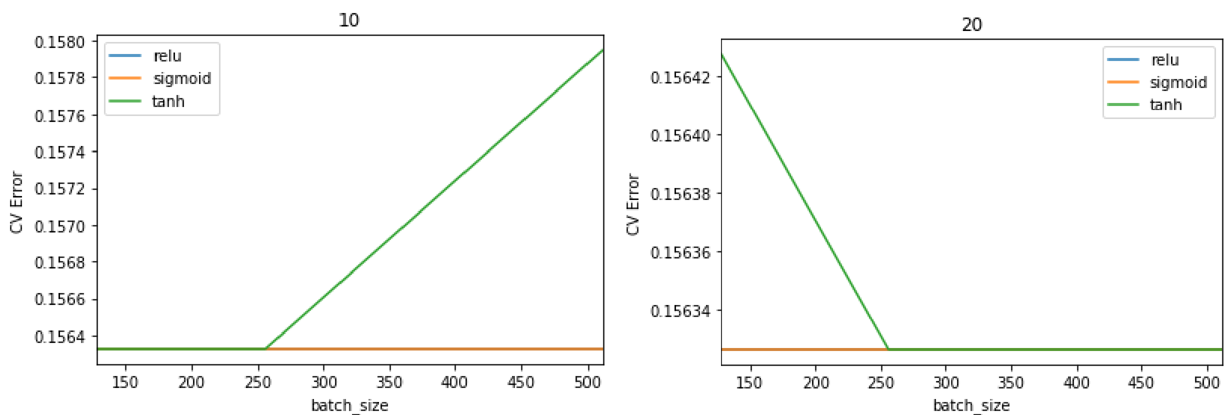


Neural Networks with 2 Hidden Layers (Dataset 2 - Online Shopping Intention Prediction)

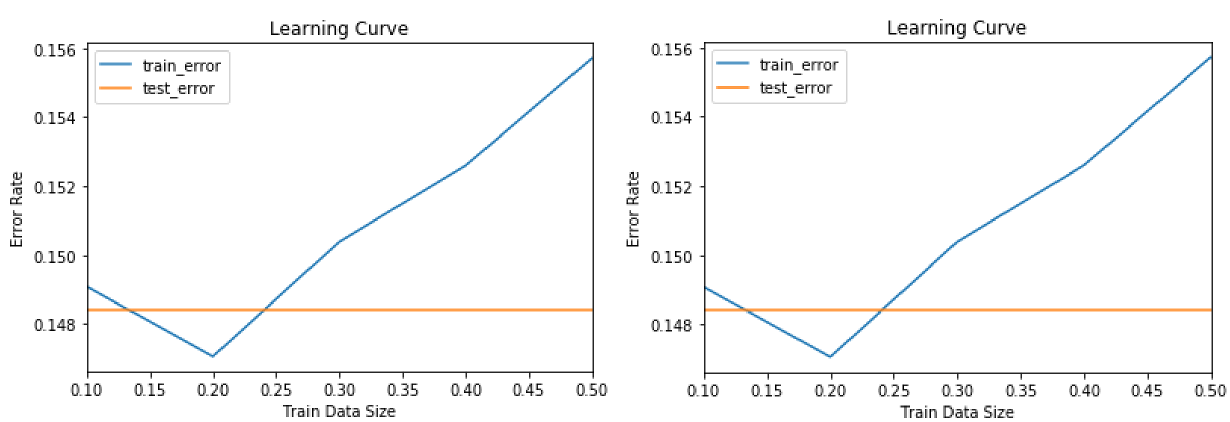
Experiment 1: Following plots compare the error results of using various parameters mentioned before. We can observe that Sigmoid as Hidden Activation does well for all Batch Sizes when number of Epoch=10.



Experiment 2: Following plots compare the error results of using various parameters mentioned before. We can see that Sigmoid as Hidden Activation does well for all Batch Sizes irrespective of Epochs number.

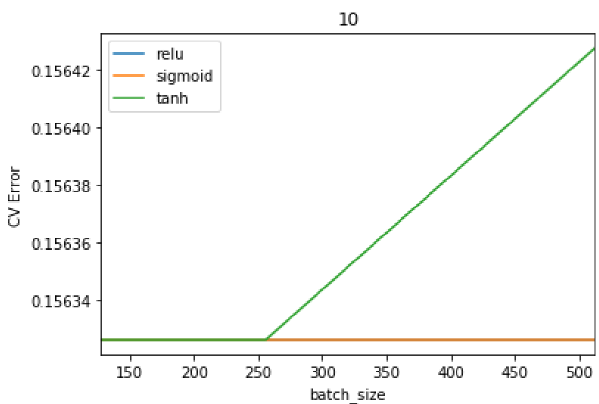


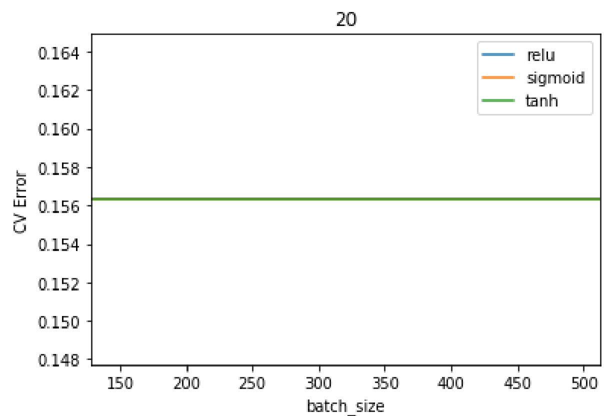
Using above found parameters as best ones, below learning curves are plotted for varying train data size. The left plot is for Experiment 1 with larger nodes and the right plot is for Experiment 2 with smaller nodes. Both plots have same test error irrespective of hidden nodes because of lower number of test instances.



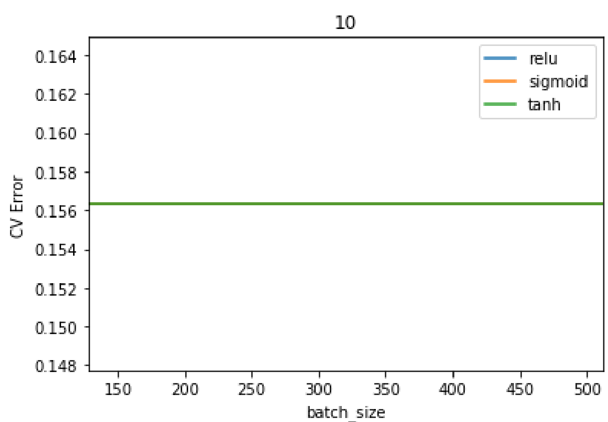
Neural Networks with 3 Hidden Layers (Dataset 2 - Online Shopping Intention Prediction)

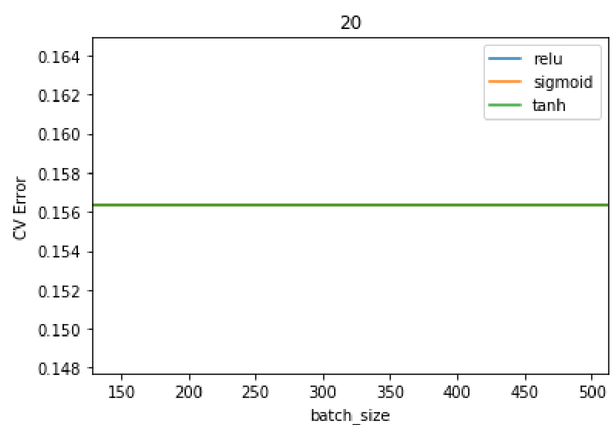
Experiment 3: Following plots compare the error results of using various parameters mentioned before. We can observe that Sigmoid as Hidden Activation does well for all Batch Sizes when Epochs number=10.



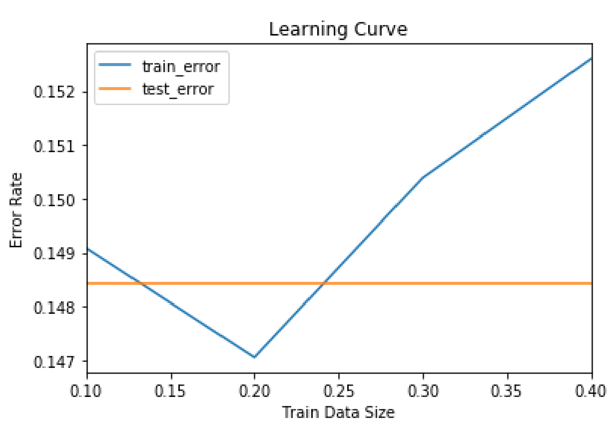


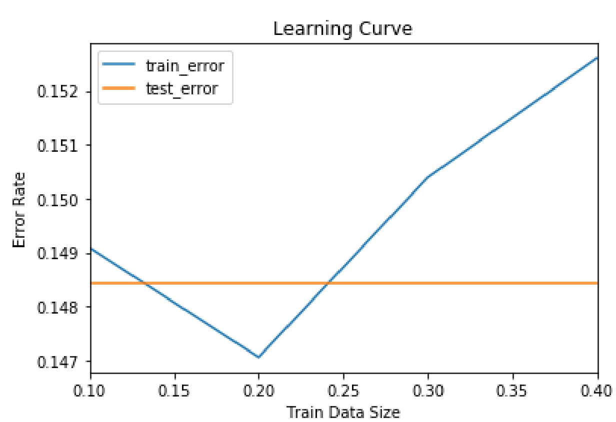
Experiment 4: Following plots compare the error results of using various parameters mentioned before. We can observe that error is not changing with change in the Hidden Activation and the number of Epochs.





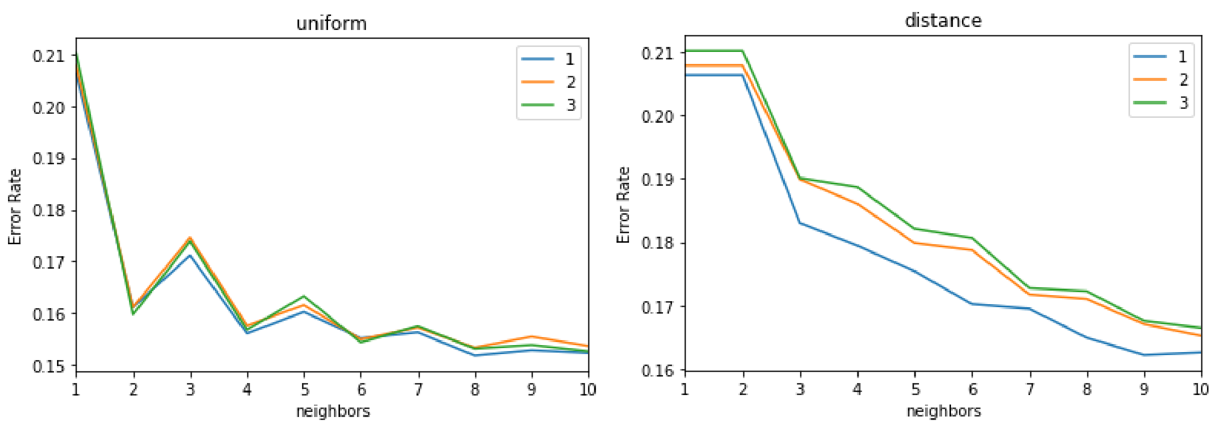
Using above found parameters as best ones, below learning curves are plotted for varying train data size. The left plot is for Experiment 3 with larger nodes and the right plot is for Experiment 4 with smaller nodes. Both plots have same test error irrespective of hidden nodes because of lower number of test instances.



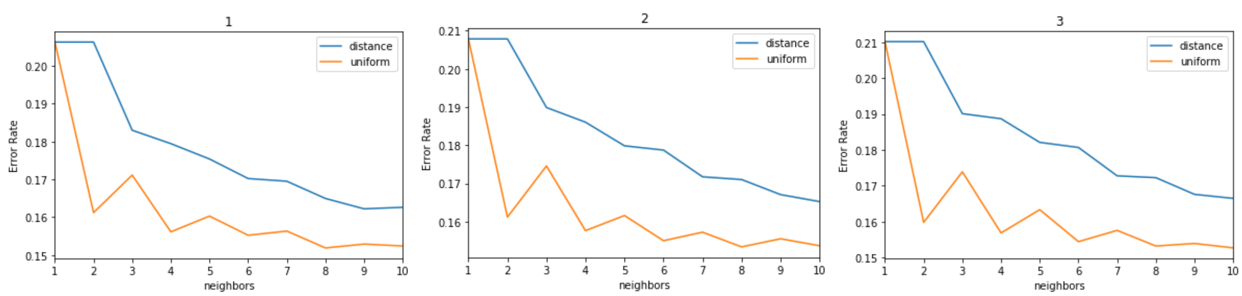


K - Nearest Neighbors with Minkowski Metric (Dataset 2 - Online Shopping Intention Prediction)

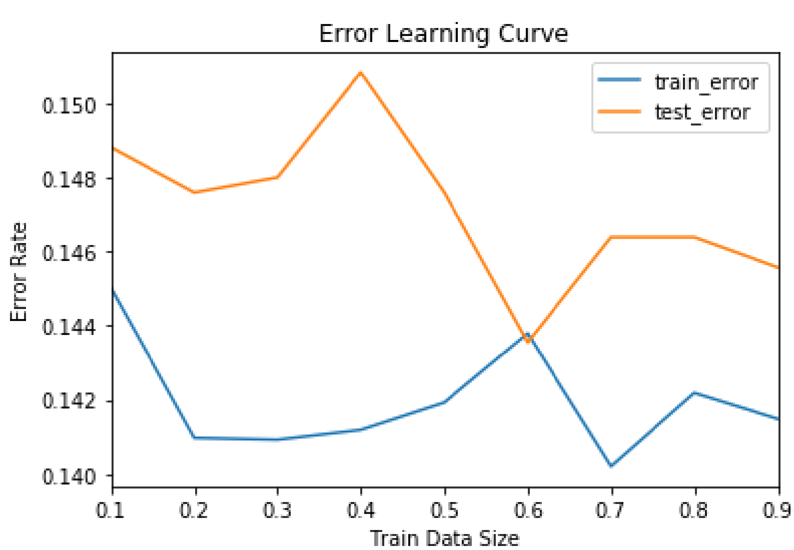
Following plots compare the error results of using various parameters mentioned before. We can observe that p = 1 as distance exponent does well for all neighbors from 1 to 10 with best error for Uniform weights



Experiment 2: Following plots compare the error results of using various parameters mentioned before. We see that Uniform is best performing weights parameter for all neighbors from 1 to 10 and for p = 1,2,3.



All the above plots indicate best neighbors are 8 as we see least error when weights = uniform and p = 1. Using above found parameters as best ones, below learning curve is plotted for varying train data size. We can observe from below that train error and test error briefly match for above 60% of train data size.



Algorithm Comparison (Dataset 1)

Comparing all algorithms Test Accuracy Scores, AdaBoost on Decision Trees achieved best accuracy.

Algorithm	Test Accuracy	Rank
SVM	0.6070	2
Decision Tree	0.6447	1
AdaBoost	0.6436	1
ANN	0.5478	3
KNN	0.4525	4

Algorithm Comparison (Dataset 2)

Comparing all algorithms Test Accuracy Scores, Support Vector Machines achieved best accuracy. However, AdaBoost, ANN and KNN al have very close Accuracy scores when compared to SVM. These scores might vary if we add add more number of test instances into the second data set.

Algorithm	Test Accuracy	Rank
SVM	0.8921	1
Decision Tree	0.7402	5
AdaBoost	0.8683	2
ANN	0.8516	3
KNN	0.8552	3