**Implementation of Artificial Neural Networks & K-Nearest Neighbors using k-fold cross validation**

**Dataset description and goal:**

The goal of the project is to:

1. Implement Artificial Neural Networks with hidden layers
2. Evaluate performance of K nearest Neighbors
3. Experiment with different hyperparameters, tune and report their optimal values using learning curves.
4. Use K-fold cross validation with above algorithms to obtain best accuracy scores.

Two different datasets are chosen to evaluate and compare model performances.

**Dataset 1:**

Contains information on SGEMM GPU Kernel Performance that captures the running time of a matrix-matrix product operation. The dataset information can be viewed and downloaded at UCI Machine learning repository.

<https://archive.ics.uci.edu/ml/datasets/SGEMM+GPU+kernel+performance>.

On quick examination, the following characteristics are observed from the dataset.

* There are 18 columns and 241600 rows
* There are no missing values
* The target variable for the model is average of the four GPU run times
* The independent variables are not highly correlated. Hence there no variables are dropped from the model.
* There are few outliers present, and they need to be removed.

**Dataset 2:**

Contains information on 23 different species of gilled mushrooms that are classified as edible or poisonous based on given features. This is a very interesting dataset that allows to experiment with multiple algorithms like PCA, Logistic Regression, Classification Trees, Ensembles and Neural Networks that readily translate to real-world observations. Also, the dataset has good volume, quality and does not have any aggregations.

In modern world, Plant Biologists classify the different plant species by observing distinctive features and appearances. Drug Specialists, Nutritionists and Agencies like FDA further enhance the models by adding decision variables like adult & child health, interactions of food constituents with other chemicals in body, allergic reactions, animal testing, clinical trials, etc. to predict the positive and negative effects of the plant food item before approving and releasing to the consumer market. A similar simulation of the classification algorithm on multiple predictor variables is carried out in this dataset.

The dataset information can be viewed and downloaded at Kaggle or UCI Machine learning repository.

<https://www.kaggle.com/uciml/mushroom-classification>

<https://archive.ics.uci.edu/ml/datasets/mushroom>

On quick examination, the following characteristics are observed from the dataset.

* There are 23 columns and 8124 rows
* The target variable for the model is ‘class’
* There are 2480 missing values for the ‘stalk-root’ column.
* There are more than 2 categories for most of the predictor variables.

**Assumptions and initial dataset preparation:**

The following assumptions and initial operations are made on the datasets for the implementation of algorithms.

* Cross validation is used with all the algorithms implemented and is written from scratch.
* For building Artificial Neural Network, Keras with Tensorflow backend is used.
* For Neural networks, the number of neurons in the subsequent hidden layers is assumed from the formula:

***((2/3)\*no. of input layer neurons) + no. of output layer neurons***

* Randomized Search is used to search the best values of parameters of artificial neural network model.

**Dataset 1:**

* Columns 1-10 and 15-18 are continuous and columns 11-14 are categorical variables.
* Scaling has been performed for columns 1-10 using Standard Scaler.
* For binary classification, column 15 is converted to either 1 or 0 using its mean value as cut off.
* The number of outliers (above or below 3 standard deviations from mean) is 5251 (accounts for 2.17 % of data) and they have been removed from the given dataset.

**Dataset 2:**

* For binary classification, column ‘class’ is converted to 1 if species is poisonous, and 0 if edible.
* Missing values for the ‘stalk-root’ column is imputed with the frequent value ‘stalk-root’ = ‘b’
* There are more than 2 categories for most predictors, and ordering them (1, 2, 3...) doesn't make any sense because they are not ordinal. Hence pivot is done on categories. This increases the number of columns.

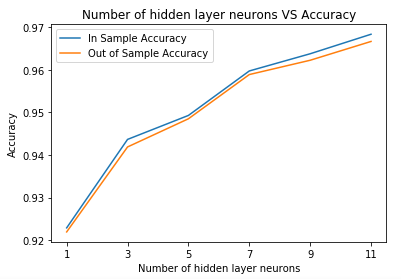
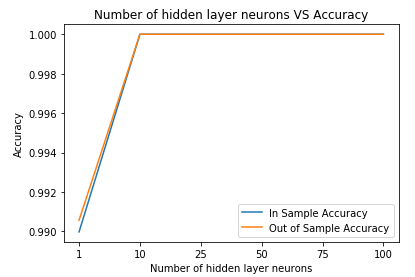
**Tasks 1:**

The dataset is scaled and partitioned into train set with 70% data, and test set with 30% data. Artificial Neural Network model is built using Keras and Tensorflow backend. Cross validation is used to split the train and test sets further into k partitions and the algorithm is applied on each partition. The final accuracy of model is the mean of all the k different accuracy scores. An initial value of k=4 is assumed to reduce time complexity.

**Experiment with different number of neurons in hidden layer (N):**

Each time the number of neurons in the hidden layer is varied, and the value corresponding to lowest error, or highest accuracy is considered as optimal number of neurons.

***Sgemm Product* *Mushroom***

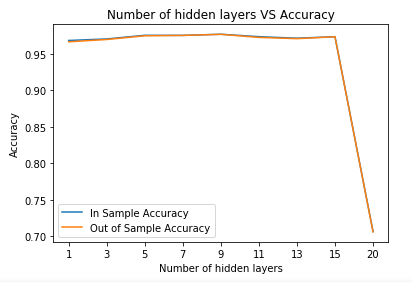
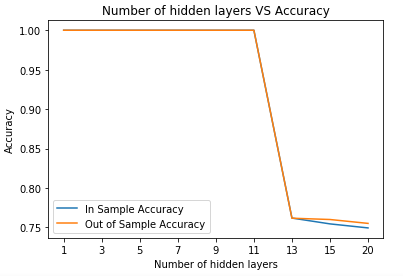
 

It is observed that N=7 is best for sgemm product dataset, after which the gap between train and test error starts to increase. N=10 is better for Mushroom dataset, after which the model performance remains constant. N is chosen such that the variance in train and test set are minimal (so that the model generalizes well) and accuracy is high.

**Experiment with different number of hidden layers:**

Each time the number of hidden layers is varied, and the value corresponding to lowest error, or highest accuracy in train and test sets is considered as optimal number of hidden layers.

***Sgemm Product* *Mushroom***

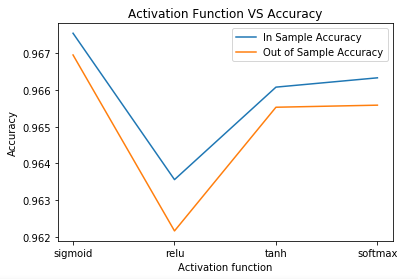
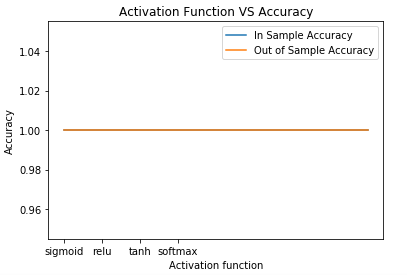
It is observed that increasing the number of hidden layers doesn’t necessarily increase the performance of the neural network. For sgemm product dataset, the maximum accuracy of train and test set is obtained when number of hidden layers = 9, after which it starts decreasing. When number of hidden layers is increased to 20, there is a steep drop in the performance of the model.

For mushroom dataset, the optimal number of hidden layers = 1. The accuracy of train and test sets remain constant up to hidden layer 11, after which it starts decreasing. When number of hidden layers is increased to 13, there is a steep drop in the performance of the model.

**Experiment with different activation functions:**

The model is tested with different activation functions like 'sigmoid','relu','tanh' and 'softmax' in the hidden and output layers and the performance of the train and test sets is evaluated.

***Sgemm Product* *Mushroom***

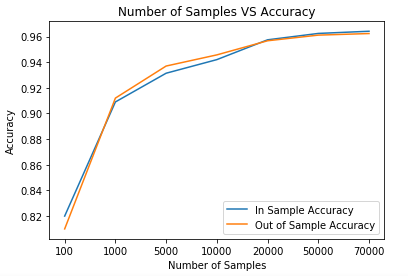
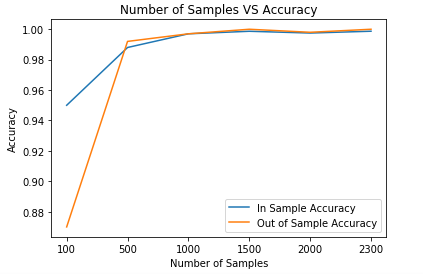
 

It is seen that for sgemm product dataset, ‘sigmoid’ activation function gives better accuracy values and ‘relu’ performs the least. For mushroom dataset, all the activation functions provide maximum accuracy values.

**Experiment with different sample sizes:**

The model is tested with different input sample sizes and the performance of the train and test sets is evaluated.

***Sgemm Product* *Mushroom***

It is seen that more the sample size, better is the performance of the neural network model. Neural Networks perform well for large datasets.

**Estimating other parameters using Randomized search:**

A randomized search is performed to get the optimal values of parameters for ANN model using KerasClassifier for both datasets.

**Sgemm product**: Best parameter values for ANN model are

‘epochs’: 100, ‘optimizer’: RMSProp, 'Batch\_size': 40, ‘activation function for hidden layers’: sigmoid

**Mushroom**: Best parameter values for ANN model are

‘epochs’: 50, ‘optimizer’: Adam, 'Batch\_size': 10, ‘activation function for hidden layers’: softmax

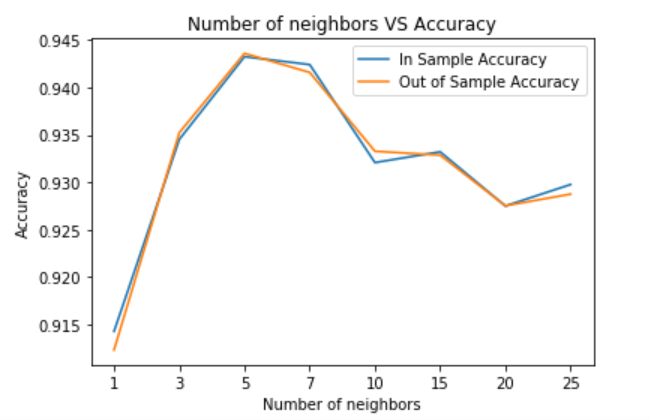
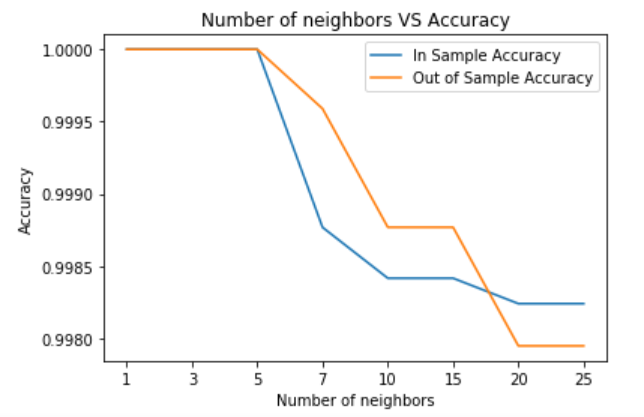
**Task 2:**

Next, K – Nearest neighbors algorithm is run on both the datasets and the performances are evaluated. Cross validation is used to split the train and test sets further into partitions and the algorithm is applied on each partition.

**Experiment with different number of neighbors (k):**

Each time the number of neighbors is varied, and the value corresponding to lowest error, or highest accuracy is considered as optimal number of neighbors.

***Sgemm Product* *Mushroom***

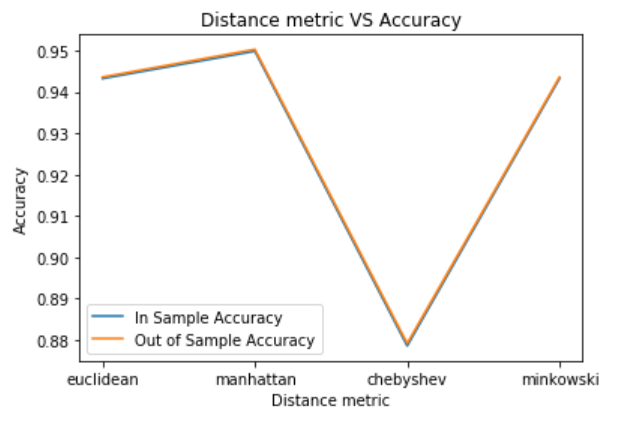
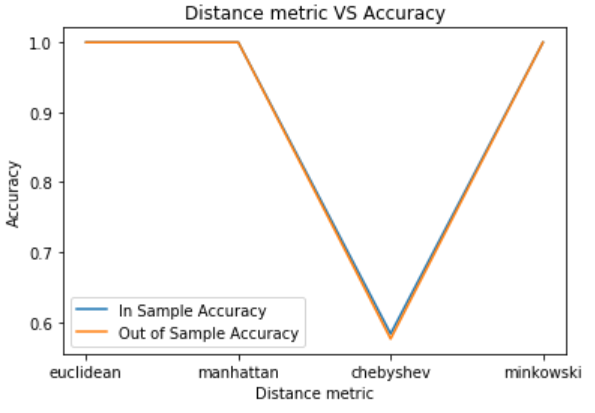
It is observed that increasing k to very large number reduces the performance of the model. For sgemm product dataset, the maximum accuracy of train and test set is obtained when number of neighbors = 5, after which it starts decreasing. When number of neighbors is increased to 10, there is a steep drop in the performance of the model.

For mushroom dataset, the optimal number of neighbors = 1 to 5. The accuracy of train and test sets remain constant up to k=5, after which it starts decreasing. When number of neighbors is increased to 10, there is a steep drop in the performance of the model.

**Experiment with different distance metrics:**

The KNN algorithm is tested with different distance metrics like Euclidean, Chebyshev, Minkowski & Manhattan, and the performance of each distance metric is evaluated for the given datasets. The value corresponding to lowest error, or highest accuracy is considered as suitable distance metric.

***Sgemm Product* *Mushroom***

For sgemm product dataset, the maximum accuracy of train and test set is obtained when distance metric = Manhattan, and Chebyshev distance gives the least performance.

For mushroom dataset, the maximum accuracy of train and test set is obtained when distance metric = Euclidean, Manhattan or Minkowski. Chebyshev distance gives the least performance.

**Inferences and Comparison of different models:**

On comparing different supervised models like Logistic Regression, SVM, Decision Trees, Ensembles, Artificial Neural Networks and K Nearest Neighbors, the following is observed.

1. Cross validation improves the model accuracy scores and reduces the bias by averaging the accuracy/error scores.
2. Artificial Neural Network models are used when dataset is large and when complex predictions are required. NN are scalable, but they consume more memory operations when number of layers and neurons are increased.
3. Increasing the number of hidden layers and number of neurons does not necessarily increase the performance of an ANN model.
4. The MLP ANN model can be used with a combination of multiple activation functions like relu, sigmoid and tanh.
5. For sgemm product dataset, ANN performs better than KNN. For mushroom dataset, both models perform equally well.
6. Among the different models, for Sgemm product dataset, maximum accuracy is given by Bagging, followed by SVM (gini), SVM (Entropy), Random Forest, XG Boost, ANN, Stochastic Gradient Boost, KNN and AdaBoost.
7. For Mushroom dataset, maximum accuracy is given by 5 algorithms – ANN, KNN, Random Forest, AdaBoost and XG Boost. It is followed by SVM (gini), SVM (Entropy), Bagging, Stochastic Gradient Boost.
8. It is best to experiment with ensembles where each subset of data is fed into an algorithm or model, and the average of the results from each model gives the best possible prediction.