BUAN 6341 APPLIED MACHINE LEARNING ASSIGNMENT 3

Project Report

Artificial Neural Networks (ANN), K Nearest Neighbors (K-NN)

By Animesh Kansal (axk169531)

Datasets Used:

1. Online news popularity data set	(39644, 61)
2. Telecom Churn data set	(7043, 21)

Scripting Language: Python as scripting language

Tasks:

1. Download and use any neural networks package to classify your classification problems. Experiment with number of layers and number of nodes, activation functions (sigmoid, tanh, etc.), and may be a couple of other parameters.

   I used Keras package which run on top of Tensorflow, I Experimented with Number of hidden la yers, Nodes in a layer, changing activation function, batch size for error propagation

2. Download and use any KNN package to classify your classification problems. Experiment with number of neighbors. You can use any distance metric appropriate to your problem. Just be clear to explain why you used the metric that you used.

   I used sklearn package to implement KNeighborsClassifier. I Experimented with Number of neighbors, Metric, cross validation

3. Small description of the data sets and the classification problem.

Telecom Churn is a typical Classification data set. It has 21 features and 7043 observations. It has 3 Numerical features, rest all are categorical features except customer ID, which we dropped as a first thing. Then we divided the data set as this :

from sklearn.model_selection import train_test_split and test_size=0.3

I did, all the hyper parameter tuning on the validation set and recorded the results on the test set

4. Error rates (train and test) for the two algorithms on your two data sets. Plot various types of learning curves that you can think of (e.g. but not limited to – error rates vs. train data size, error rates vs. clock time to train/test, etc.).

Telecom Churn Analysis Dataset

Algorithm	Support Vector Machine	Decision Tree	Decision Tree with Boosting	ANN	KNN
Train Set Accuracy	77.08%(Linear) 91.37%(RBF) 73.3%(Sigmoid)	99.837	99.837	80.29%	82.44%
Test Set Accuracy	75.97%(Linear) 77.35%(RBF) 73.7%(Sigmoid)	69.763	72.654	79.72%	76.07%
How to overcome overfitting/ improve model	Optimizing C and gamma values	Pruning, depth = 4	Pruning, depth = 1	Experimented with Number of hidden layers, Nodes in a layer, changing activation f unction, batch size for error propagation	Experimented with Number of neighbors, Metric, cross validation
Train set Accuracy after improvements	77.06%(RBF) 79.99%(Sigmoid)	79.622	81.119	79.25%	82.44%
Test set Accuracy after improvements	76.73%(RBF) 79.15%(Sigmoid)	79.384	79.143	77.91%	76.07%

In telecom churn dataset, ANN earlier was showing accuracy of 80.29% and KNN was showing 82.44% before applying any improvements in that model.

I tuned the parameters for ANN on a validation set, Experimented with Number of hidden layers, Nodes in a hidden layer, changing the activation function, batch size for error propagation. But no improvement in the training or test set.

Similar to ANN in KNN, when I experimented with Number of neighbors and Metric used cross validation to measure the Mean accuracy and std.

In this case, I will choose DT with Adaboost as the best model for the Telecom project, as the depth of the resulting model is 1, hence this is simplest and has Test accuracy is almost equivalent to max among all the five.

Online News Sharing Dataset

Algorithm	**Support Vector Machine**	**Decision Tree**	**Boosting**	**ANN**	**KNN**
Train Set Accuracy	50.5%(Linear) 53.5% (Sigmoid)	100.000	100.000	69.31%	73.67%
Test Set Accuracy	49.67%(Linear) 53.04% (Sigmoid)	57.685	57.945	65.45%	60.12%
How to overcome overfitting/ improve model	Optimized C & gamma values but it was very slow. used PCA, because number of features are too much, no significant change	Pruning, depth = 7	Pruning, depth = 1	Experimented with Number of hidden layers, Nodes in a layer, changing activation function, batch size for error propagation	Experimented with Number of neighbors, Metric, cross validation
Train set Accuracy after improvements	50.58%	67.286	68.177	69.23%	73.67%
Test set Accuracy after improvements	51.1%	64.175	64.810	65.65%	60.12%

In Online News Popularity dataset, ANN earlier was showing accuracy of 69.31% and KNN was showing 73.67% before applying any improvements in that model.

I tuned the parameters for ANN on a validation set, Experimented with Number of hidden layers, Nodes in a hidden layer, changing the activation function, batch size for error propagation. But no significant improvement in the training or test set.

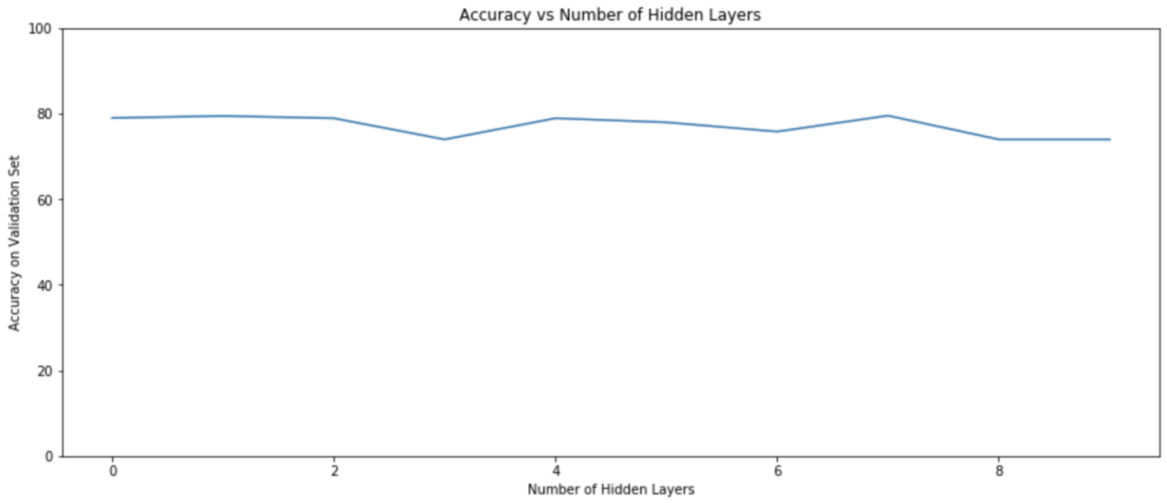
Similar to ANN in KNN, when I experimented with Number of neighbors and Metric used cross validation to measure the Mean accuracy and std.

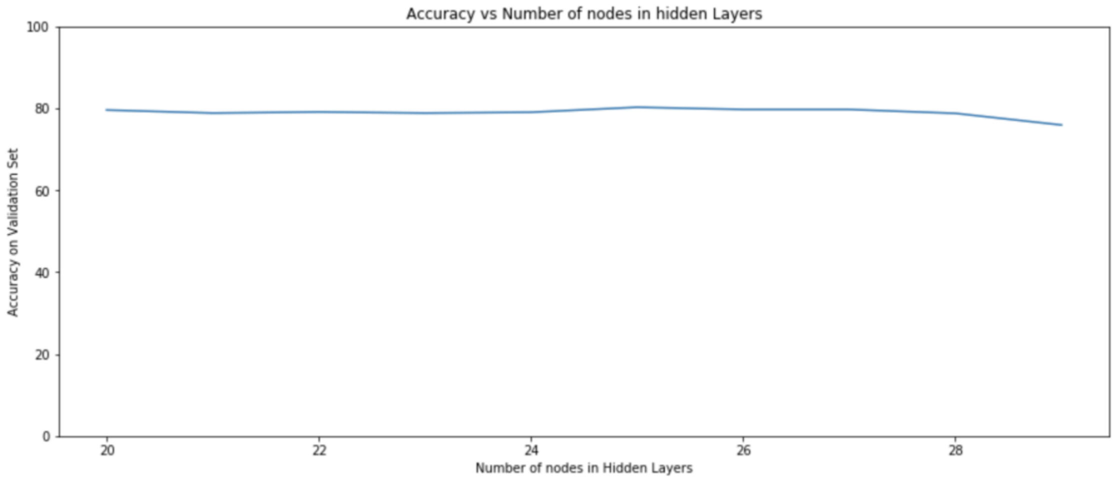
In this case, I will choose ANN as the best model for the Online News popularity project, as theTest accuracy is max among all the five.

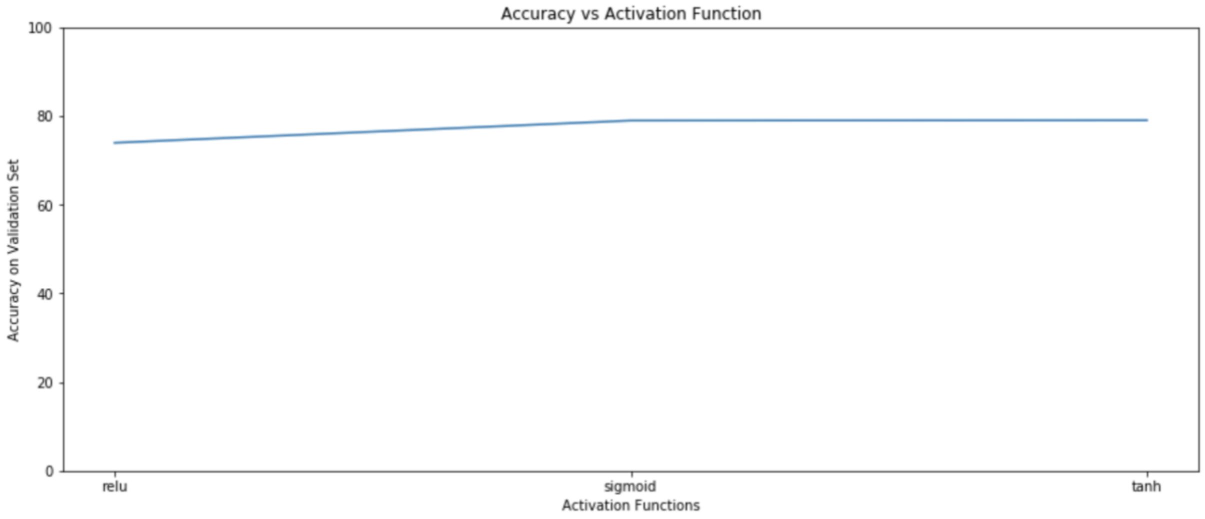
Both the datasets have shown almost similar graphs for these:

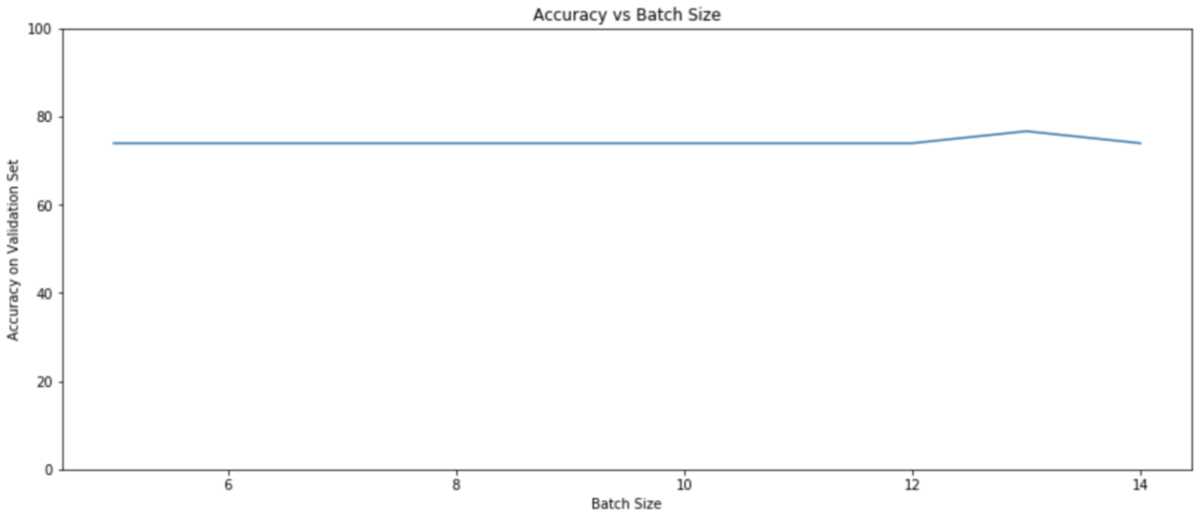
1. **ANN** -> Accuracy vs Number of Hidden Layers
2. **ANN** -> Accuracy vs Number of nodes in Hidden Layer
3. **ANN** -> Accuracy vs Activation Function
4. **ANN** -> Accuracy vs Batch Size
5. **KNN** -> Accuracy vs Number of Nearest Neighbors
6. **KNN** -> Accuracy vs Metric function

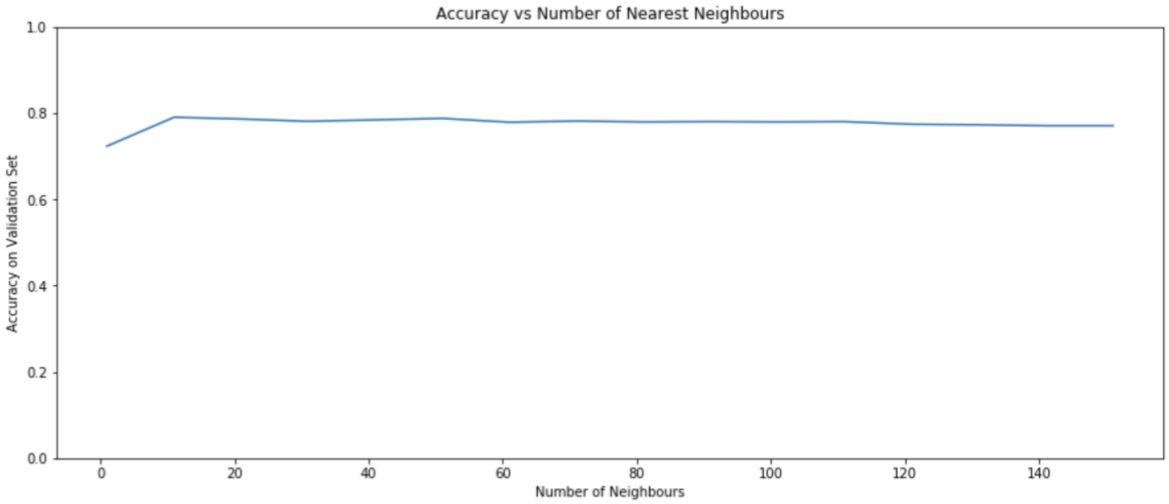
As shown below :

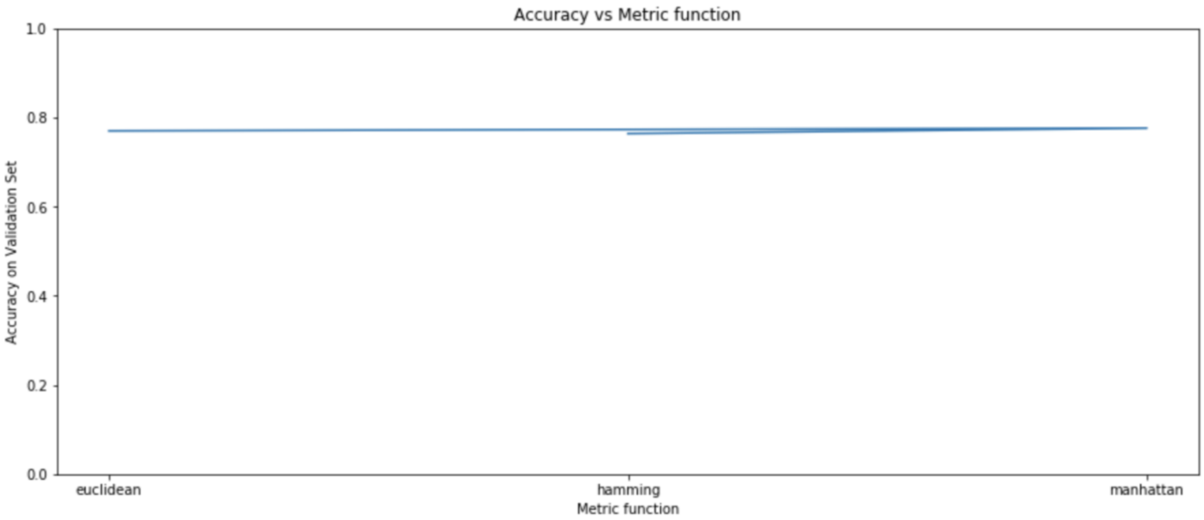







Accuracy vs Batch Size


Accuracy vs Number of Nearest Neighbours


Accuracy vs Metric function

What additional things can you do?

Additional things we can do is:

1) Grid Search
2) Dimensionality reduction

If yes then why didn't you implement it?

Yes, Grid search would help definitely, I tried implementing but it takes a long time to run and is not good to run on small machines, we need to run on powerful machines.

Also, Dimensionality Reduction would definitely help. I will implement in the next assignment.