Classification for D2D Resource Allocation
Report 1

Next Generation Wireless Labs IISc
8 - 9 July 2021

K Venkat Ramnan, Intern
Under guidance of Prof. N B Mehta sir and Bala sir

Dataset Explained

Features :
1. Rates (V_1 ..... V_{30}) : 30 values
2. Weights (w_1 ..... w_{30}) : 30 Values
3. C : 1

Target :
X_1 .... X_{30} : 30 Values

Number of columns :
- Features : 61
- Target : 30

Number of instances (rows) = 9210

Thus,
If X matrix denotes the features, the shape of matrix is (9210 x 61)
If y matrix denotes the target, the shape is (9210 x 30)

Spitting the Dataset

Training Data : 80% of total dataset
Validation/Testing Data: 20% of total dataset

Note: Here we are trying to predict X_n one by one other. We are not predicting all 30 values at once.

Machine Learning Techniques Used

Binary Classification is the path since we are classifying 0 or 1.

1. Logistic Regression

   This is a linear classification model. The norm used for penalization is L1.

2. K Nearest Classification

   This is a non linear model.

   Parameters : (algorithm='auto', leaf_size=30, metric='minkowski', metric_params=None, n_jobs=None, n_neighbors=30, p=2, weights='uniform')

3. Random Forest Classifiers

   This is a non linear model.

   Parameters : (bootstrap=True, class_weight=None, criterion='gini', max_depth=30, max_features='auto', max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=1, min_samples_split=2,

min_weight_fraction_leaf=0.0, n_estimators=10,
n_jobs=None, oob_score=False, random_state=42, verbose=0,
warm_start=False)

4. Gaussian Naive Bayes Classifier
   This is a non linear model.
   Parameters: (priors=None, var_smoothing=1e-09)

5. XGBoost Classifier
   Parameters: (n_estimators=100, learning_rate=1.0, max_depth=1, random_state=0).

Results of Machine Learning Techniques tried

Technique	Training Score (Accuracy)	Testing Score(Accuracy)
Logistic Regression	0.6080347448425625	0.6003800732998507
K Nearest Classification	0.6269852042893986	0.5570032573289903
Random Forest Classifier	0.9868331749694584	0.5597176981541803
Gaussian Naive Bayes	0.6055382109406814	0.5982627578718784
XGBoost	0.5867195152213424	0.5863192182410424

The results of validation accuracy are between **55 to 60 %**
Note : Training accuracy of Random forest so high due to Overfitting.

Deep Neural Network Classifier

Model 1

```
Layer (type)            Output Shape          Param #
=================================================================
input_7 (InputLayer)       [(None, 61)]          0       Input
__________________________________________________________
dense_30 (Dense)          (None, 128)           7936     Activation: Relu
__________________________________________________________
dense_31 (Dense)          (None, 64)            8256     Activation: Relu
__________________________________________________________
dense_32 (Dense)          (None, 32)            2080     Activation: Relu
__________________________________________________________
dense_33 (Dense)          (None, 16)            528      Activation: Relu
__________________________________________________________
dense_34 (Dense)          (None, 1)             17     Output : Sigmoid Layer
=================================================================
```

Optimizer Used : SGD

Learning Rate : 0.0001

Loss : Binary Crossentropy

Result Metric : Accuracy

Epochs Trained : 1000

Batch Size : 32

Model 2

```
Layer (type)             Output Shape          Param #
=================================================================
input_7 (InputLayer)       [(None, 61)]          0       Input

dense_5 (Dense)           (None, 128)           7936      Activation: Relu

dropout_4 (Dropout)        (None, 128)           0

dense_6 (Dense)           (None, 64)           8256      Activation: Relu

dropout_5 (Dropout)        (None, 64)           0

dense_7 (Dense)           (None, 32)           2080      Activation: Relu

dropout_6 (Dropout)        (None, 32)           0

dense_8 (Dense)           (None, 16)           528      Activation: Relu

dropout_7 (Dropout)        (None, 16)           0

dense_9 (Dense)           (None, 1)           17      Output : Sigmoid Layer
```

Optimizer Used : SGD

Learning Rate : 0.0001

Loss : Binary Crossentropy

Result Metric : Accuracy

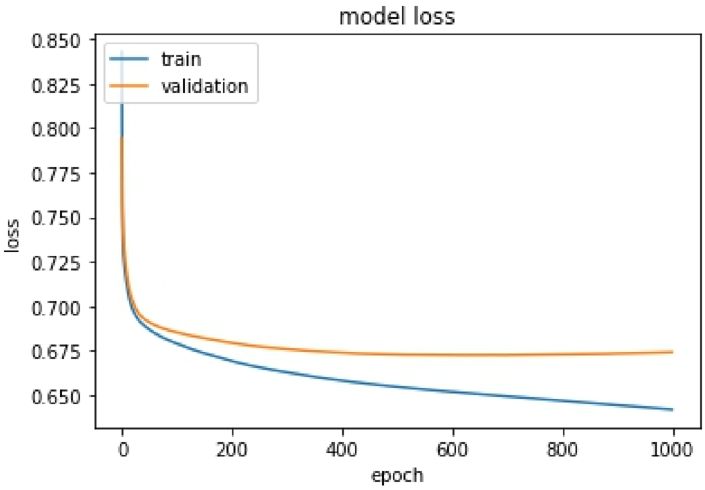
Epochs Trained : 1000

Batch Size : 32

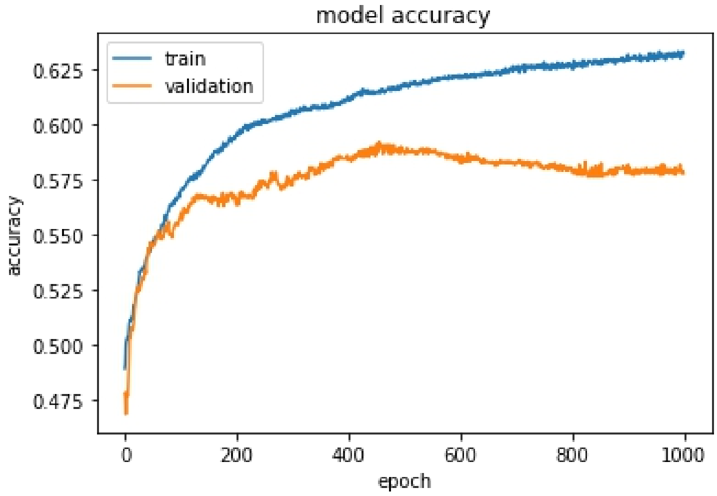
Results of DNN models

Model 1

Loss Graph

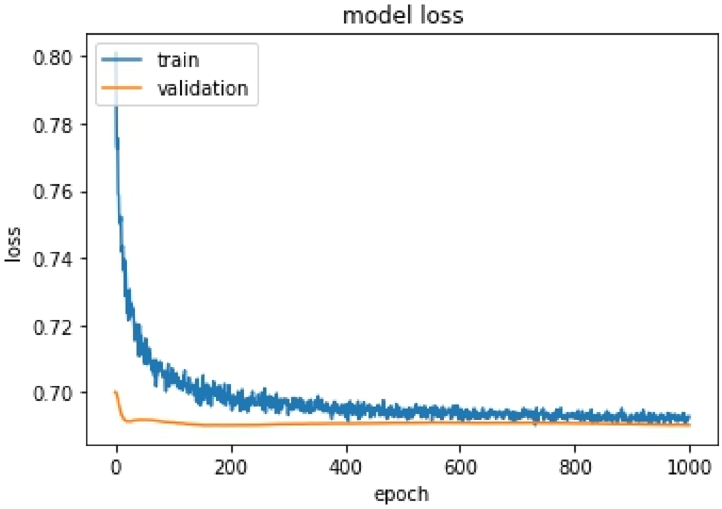


Accuracy Graph

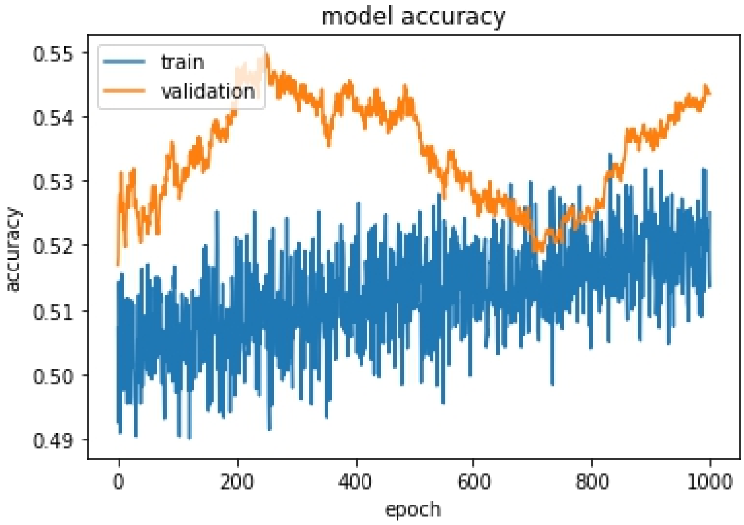


Model 2

Loss Graph



Accuracy Graph



Conclusion

The DNN models also gave a validation accuracy of **55 to 60** percent. This is also similar to the previous Machine Learning models.

The split up of 0s and 1s are also perfect, i.e 50% 1s and 50% 0s. So there is nothing wrong with the split up of data and uniformity is maintained. But size of dataset is less. Since the number of features are high but instances are less.

Note : Here none of the data can be removed. Since essentially in the non machine learning mathematical model (Knapsack Problem), all the data is of utmost importance to solve this challenge.

The next steps :
1. Increase in Dataset
2. Building more robust model
3. Since the inclusion of disjunctivity is available as an adjacency matrix, techniques like Graph Neural Network can be explored.