**Report**

Title : Machine Learning model to classify Musk molecules.

Model Description :

The dataset contains 6598 rows and 170 columns where each row is an instance of a molecule and its features . Each molecules has 169 attributes where each attribute is described as –

Attribute 1 – ID

Attribute 2 – molecule name

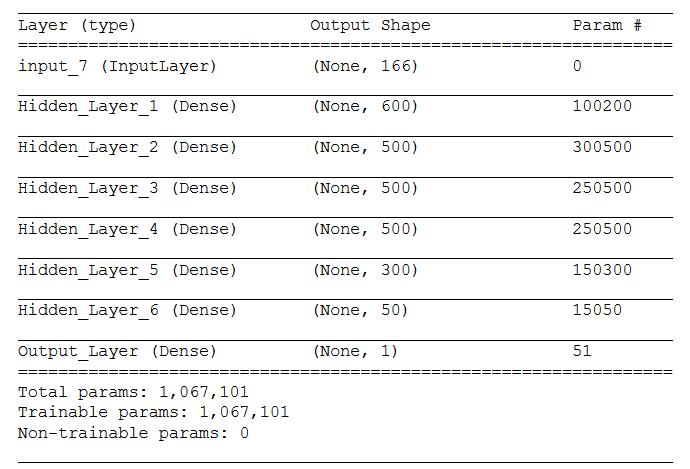
Attribute 3 – conformation name

Attribute 4 to 169 – f1 to f166 (features of each molecules)

Attribute 170 – Class (1 for Musk and 0 for non Musk)

So before selecting the model and architecture we need to preprocess the data and convert it to X features and Y labels which can be provided to the model as input. So I take the 4 to 169 columns as X features and 170th column as Y label. The features here are in integer type so I converted the features into float type. The features are merely in the range of 360 to -360, so normalization is not required.

After preprocessing the data and extracting features and labels from it I distributed the data into training and cross validation data (randomly 0.2 is selected for cross validation).

Now finally the model selection can be done. So I tested different architectures and algorithms such as ANN,CNN and RNN but the maximum and efficiency is gained while using the simple artificial neural network architecture. This architecture consists of 8 layers including the input and output layers and 6 hidden layers. The layer details are attached below –

The reason behind choosing this architecture because this gives a good accuracy over the data and is very computationally simple and efficient as compared to CNN and RNN. While training the model I have used the following settings –

Learning rate = 0.0000001

epochs = 10

batch size = 1000

optimizer = Adam

loss function = binary cross entropy

## Final Performance Measures :

After using the artificial neural network model using the above mention settings, I have obtained the following performance measurements –

Training accuracy – 0.9634

Training loss – 0.1121

validation accuracy – 0.1100

Validation loss - 0.9689

The precision and recall is ignored because those are global metrics and Keras performed the training in batch so these might be misleading sometimes.

## Graphs :

