2110581 BIOINFORMATIC I

Progress Report III

non-coding RNA Classification

Pakkapon Wattanawaha 6130391021

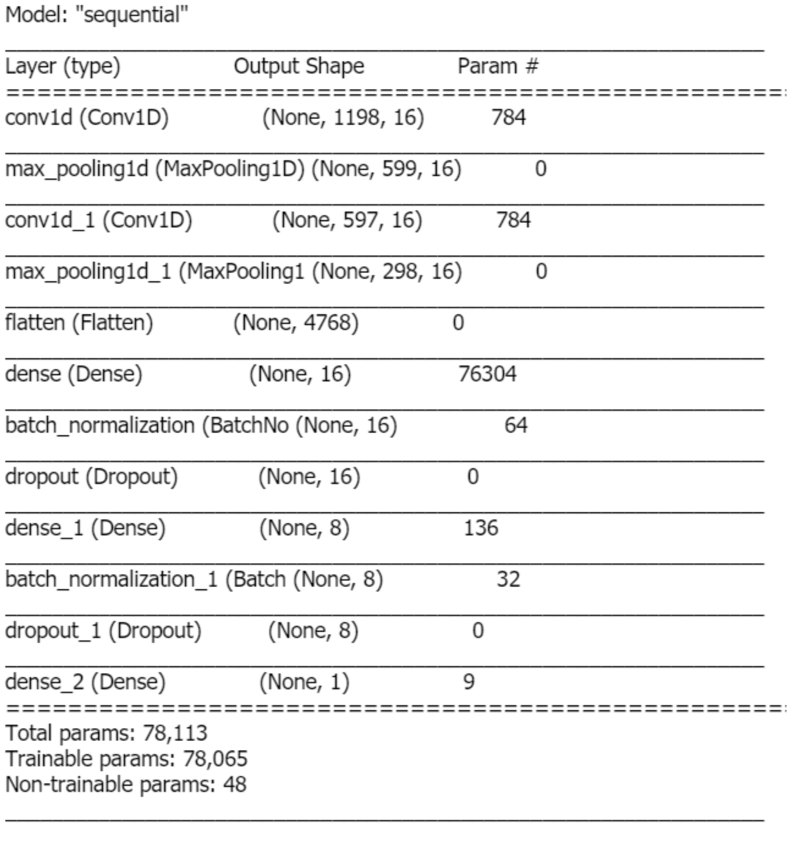
Sathianpong Trangcasanchai 6131050221

Supawit Sutthiboriban 6130535121

Report

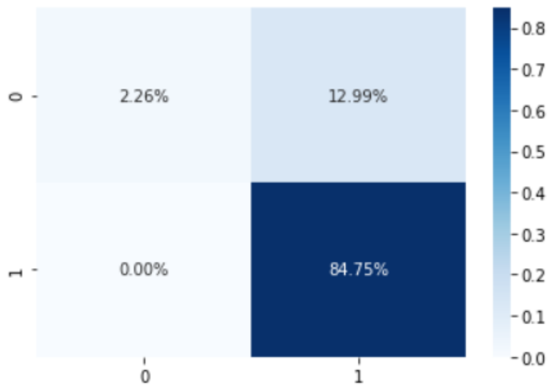
Our code written in Python consists of 3 parts. First part is to convert data from input FASTA file format to ncRNA matrix. Second part is to generate ncRNApair data and ncRNApair label from ncRNA matrix from part one. The last part is to use the ncRNA data along with its label to train the neural network to recognize the pattern of data.

In the third part of deep learning, we deploy TensorFlow library to build our convolutional neural network. Our network applies a one-dimensional CNN to accurate clustering of ncRNA sequences, we developed a new CNN-based method for classification of pairwise alignments of ncRNA sequences. The architecture of model is described as the following.

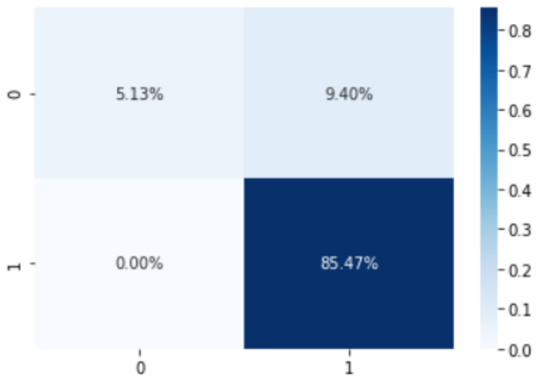


```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d (Conv1D)              (None, 1198, 16)          784
_________________________________________________________________
max_pooling1d (MaxPooling1D) (None, 599, 16)           0
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 597, 16)           784
_________________________________________________________________
max_pooling1d_1 (MaxPooling1 (None, 298, 16)           0
_________________________________________________________________
flatten (Flatten)            (None, 4768)              0
_________________________________________________________________
dense (Dense)                (None, 16)                76304
_________________________________________________________________
batch_normalization (BatchNo (None, 16)                64
_________________________________________________________________
dropout (Dropout)            (None, 16)                0
_________________________________________________________________
dense_1 (Dense)              (None, 8)                 136
_________________________________________________________________
batch_normalization_1 (Batch (None, 8)                 32
_________________________________________________________________
dropout_1 (Dropout)          (None, 8)                 0
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 9
=================================================================
Total params: 78,113
Trainable params: 78,065
Non-trainable params: 48
_________________________________________________________________
```

Then, we compile our model using ‘Adam’ optimizer and ‘Accuracy’ for a metrics. We the train the model with 10 epochs and get the final accuracy on training around 90%. We also deploy confusion_matrix and accuracy_score module from sklearn.metrics to evaluate the model’s performance.



Confusion matrix for predicting training set



Confusion matrix for predicting test set

Source code available on Github :
https://github.com/pakkaponwattanawaha/Bioinfomatics_TermProject