**GENOMICS**

**Data Processing:**

Since the input sequences are English literals(alphabets) which the network cannot actually use for activations, I have modified the input sequences into real numbers as indices to each base type. Where the mapping {'A':0, 'C':1, 'G':2, 'T':3} has been used to represent TF sequences in base type to TF sequences of real numbers.

**Model:**

I am using Keras machine learning library to construct my model. I am using convolutional neural network to predict whether a particular sequence is a binding site or not. My model has an embedding layer to convert base types into sequence of latent factors, where the latent factors is used to capture the surrounding context area of a base type which might act as a binding site for the sequence. The embedding input is passed to 3 convolutional layers with intermediate pooling layers. (1) Conv 2x2, kernels = 32, activation = ReLU, (2) Max-Pooling, stride = 2, (3) Conv 3x3, kernels = 64, (4) Max-Pooling, stride = 2,   
(5) Conv 2x2, kernels = 128.

The last output from conv layer 3 is passed to a dropout layer for regularization with a dropout factor of 0.5, which is then passed to a fully-connected layer of size 128. Fully connected dense layer is then connected to a single neuron with sigmoid activation to calculate the probability of binding site.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Layer (type) Output Shape Param#** ================================================================= **embedding\_1 (Embedding)**  (None, 14, 32) 160

**conv1d\_1 (Conv1D)** (None, 13, 32) 2080

**max\_pooling1d\_1(MaxPooling1)** (None, 6, 32) 0

**conv1d\_2 (Conv1D)** (None, 4, 64) 6208

**max\_pooling1d\_2(MaxPooling1)** (None, 2, 64) 0

**conv1d\_3 (Conv1D)** (None, 1, 128) 16512

**dropout\_1 (Dropout)** (None, 1, 128) 0

**flatten\_1 (Flatten)** (None, 128) 0

**dense\_1 (Dense)** (None, 128) 16512

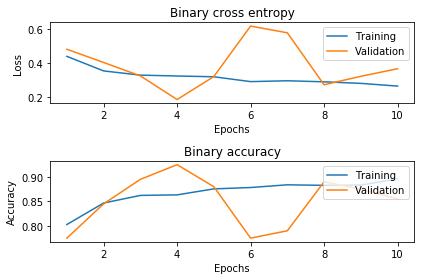
**dropout\_2 (Dropout)** (None, 128) 0

**dense\_2 (Dense)** (None, 1) 129 ==========================================================================

**Total params**: 41,601 **Trainable params:** 41,601 **Non-trainable params:** 0

**Results:**

Three convolutional layers seems to give better results. Below results have been done on a training data of 1800 sequences shuffled in each epoch and tested on 200 sequences for validation purposes. For the final model the model is trained on 2000 sequences.



P.S: submission.csv gets generated with ‘#’ in the first line please remove it while testing purposes

Kaggle team: jazzycrazzy

Github: