ISyE 6740 – Fall 2020
The Stability prediction of COVID-19 mRNA Vaccine
Project report

Team :

Xu_Lifei, lxu803@gatech.edu;

Ramachandran Seetharaman_Devanathan, dseetharaman3@gatech.edu;

Fang_Xiaoguang, xfang70@gatech.edu ;

Introduction

The objective of the project is to improve the Stability of the COVID-19 mRNA Vaccine by predicting the reactivity and degradation of RNA sequences under various environmental parameters like temperature and PH levels. The project is chosen based on the recently concluded Kaggle competition, https://www.kaggle.com/c/Stanford-covid-vaccine/overview.

Although the competition is completed and multiple Kaggle users have submitted their solutions, we would focus on applying our learnings from this course on the problem and try to achieve comparable accuracy with the top solutions submitted in the competition.

Problem Statement

The COVID-19 disease has caused great threats to humans and harms society. Up to now, there are more than 40 million infection cases and 1 million death cases. The main symptom it caused is acquired pneumonia, it also caused severe problems on vessels, heart, kidney, etc. There is also no effective medicine for this disease. Vaccines are believed to be the most promising solution to stop this pandemic. Among all vaccines in a clinical trial, one is called the RNA vaccine.

The ribonucleic acid used for translating the antigen protein is packed in the nanoparticle's capsules. When the capsules are injected into the human body, the lipid layer of capsules will merge with the cell membrane of human cells. The RNA of S protein will be released in the cytoplasm and translated into the antigen protein. mRNA vaccine has a lot of advantages. First, because the S protein of the virus contains the glycoprotein, the glycoprotein can't be produced in the prokaryotes like bacteria. So, the vaccine of COVID-19 is not suitable to be produced by the traditional protein recombinant method. Second, the translated product of the RNA vaccine is the same as the COVID-19 virus. The translated antigen protein includes every protein structure of COVID-19 S protein. It can stimulate the immune reaction to produce the antibody against the S protein. Third, the main component of the mRNA vaccine is the ribonucleic acid, it can be synthesized by chemical reaction or transcribed in vitro. So, it can be produced quickly on large scale. Fourth, there is no viral vector in the RNA vaccine. This avoids the off-target problem that the antibodies produced in the body attack the vector virus, not the antigen. So, the RNA vaccine is safe for a human being. However, the RNA vaccine is ribonucleic acid, it is not stable. The long sequence of RNA vaccine is easily affected by the environments like temperature, PH value, etc. The RNA is easy to degrade quickly. The RNA vaccine will lose effect when they are at room temperature and should be stored in a cold environment (around 4 cent degree) when they are distributed widely. This is a tough hurdle to supply it in cold temperatures. However, the sequence of RNA can be modified to improve its stability without affecting the effect. This makes it available to deliver the RNA vaccine at room temperature.

The chemical activity of single ribonucleic acid is known. So, the degradation rates of each base of the RNA vaccine can be predicted. We will use a different model to predict the degradation rate of the RNA vaccine. This will be used for screening the most stable mRNA vaccine. The model will be trained on the subset including more than 3000 RNA units. The molecule will include all RNA sequences and structures. The model will also predict the RNA degradation rates in every position. The model will be tested by the data that has been confirmed by experimental data.

In summary, we hope our model will help with the development of the COVID-19 RNA vaccine.

Data Overview

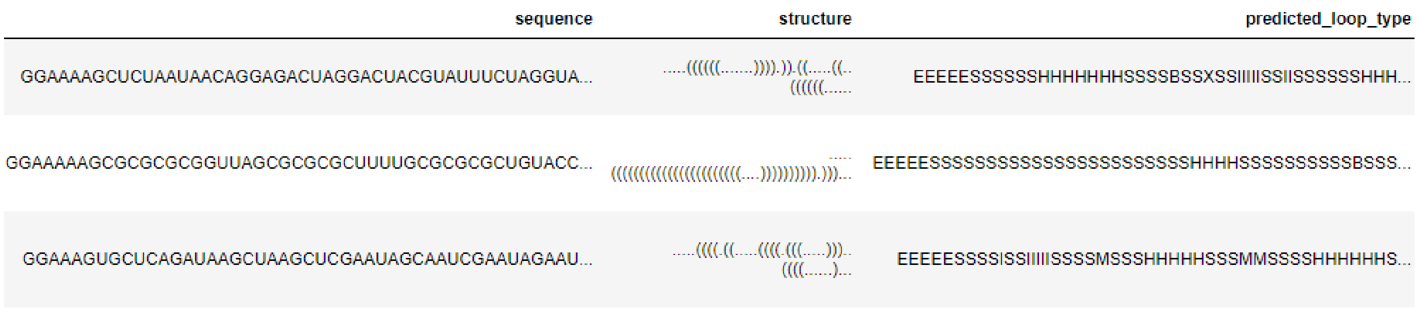
The data for the project is provided in the Kaggle competition, https://www.kaggle.com/c/stanfordcovid-vaccine/data. The data includes the structural information of different RNA sequences as well as reactivity value under certain circumstances. The details of the various attributes in the data are given below.

Column	**Description**
id	A string for each sample as the identifier
seq_scored	The integer defines the length of the RNA segment used for training and test.
seq_length	The Integer values defining the length of the sequence.
$sequence$	The string describes the RNA sequence, which consists of A, G, U, and C for each sample.
$structure$	A list of (,), and. characters that describe whether a base is estimated to be paired or unpaired.

predicted_loop_type	Describes the structural context (also referred to as 'loop type')of each character in the sequence.
$reactivity$,deg_PH10, deg_MG_PH10,deg_50C,deg_MG_50C	The target columns indicating reactivity and likelihood of degradation at the base/linkage before/after incubating with magnesium in high pH (pH 10) and high temperature (50 degrees Celsius)

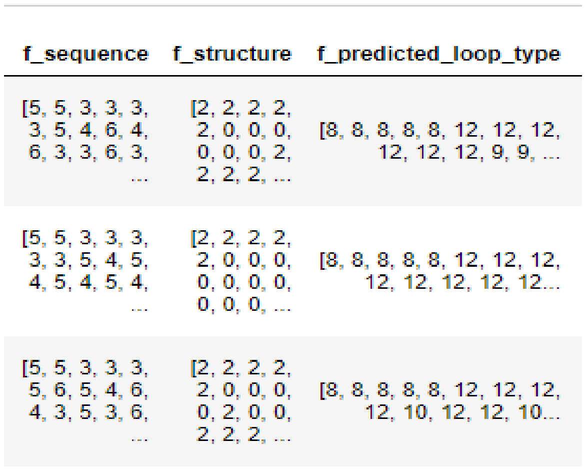
Exploratory Data Analysis

The core features of this problem are mRNA sequence, structure, and predicted loop type. All these 3 variables are string sequences. The sample data for the features are given below.



sequence	structure	predicted_loop_type
GGAAAAGCUCUAAUAACAGGAGACUAGGACUACGUAUUUCUAGGUA...	.....((((((.......)))).)).((.....((..((((((......	EEEEESSSSSSHHHHHHHSSSSBSSXSSIIIIISSIISSSSSSSHHH...
GGAAAAAGCGCGCGCGGUUAGCGCGCGCUUUUGCGCGCGCUGUACC...	.....((((((((((((((((((((((((....))))))))))).)))...	EEEEESSSSSSSSSSSSSSSSSSSSSSSSSHHHHSSSSSSSSSSSBSSS...
GGAAAGUGCUCAGAUAAGCUAAGCUCGAAUAGCAAUCGAAUAGAAU...	.....((((.((.....((((.(((.....)))..((((......)...	EEEEESSSSISSIIIIISSSSMSSSHHHHHSSSMMSSSSHHHHHHS...

The string values in the features are encoded to integers element by element. The encoder encodes each character in the above 3 columns to an integer and below is the snapshot of the features after encoding is done



f_sequence	f_structure	f_predicted_loop_type
[5, 5, 3, 3, 3, 3, 5, 4, 6, 4, 6, 3, 3, 6, 3, ...	[2, 2, 2, 2, 2, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, ...	[8, 8, 8, 8, 8, 12, 12, 12, 12, 12, 12, 9, 9, ...
[5, 5, 3, 3, 3, 3, 3, 5, 4, 5, 4, 5, 4, 5, 4, ...	[2, 2, 2, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[8, 8, 8, 8, 8, 12, 12, 12, 12, 12, 12, 12, 12...
[5, 5, 3, 3, 3, 5, 6, 5, 4, 6, 4, 3, 5, 3, 6, ...	[2, 2, 2, 2, 2, 0, 0, 0, 0, 2, 0, 0, 2, 2, 2, ...	[8, 8, 8, 8, 8, 12, 12, 12, 12, 10, 12, 12, 10...

Apart from the core features, the model also uses derived features from the BPPS data provided in the Kaggle competition. BPPS is a Base Pairing Probabilities matrices for each of the mRNA sequence. BPPS gives the probability that each pair of nucleotides in the mRNA forms a base pair. Below is the example structure of the BPPS Matrix.

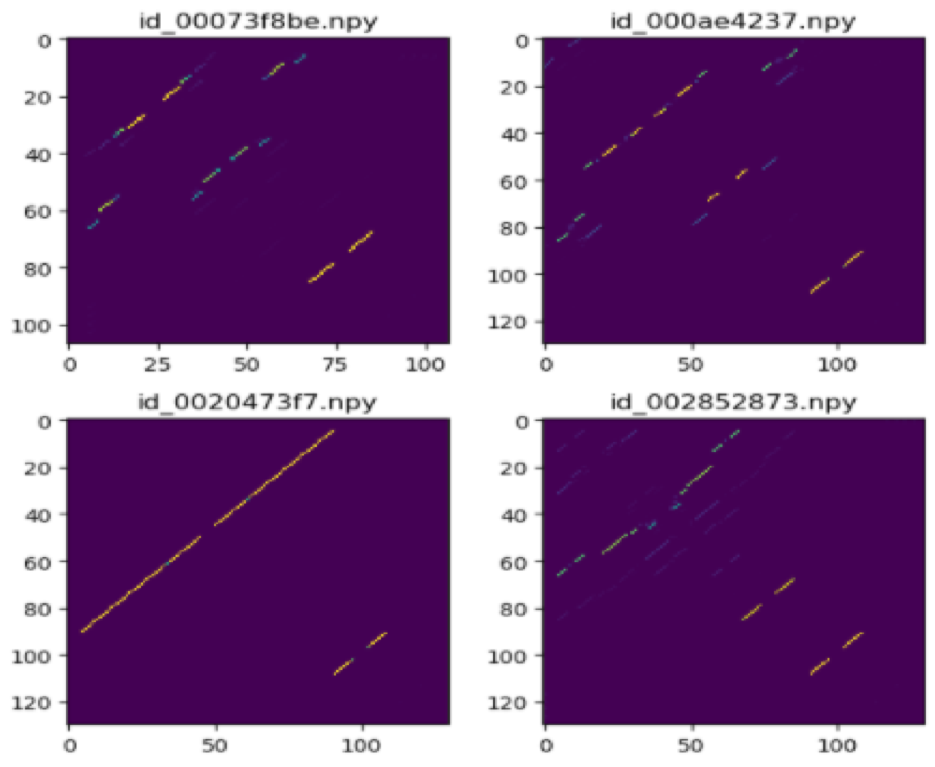
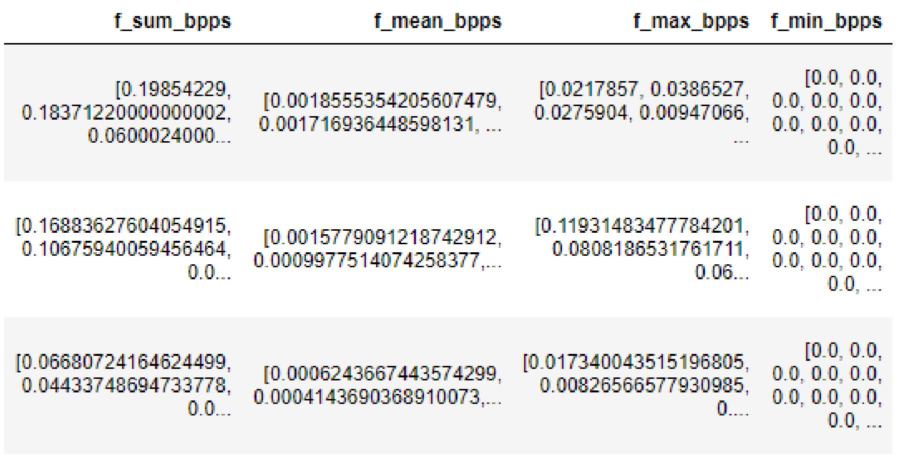


Figure 1: Exhibition of BPPS data

We used the BPPS data to derive the column-wise Sum, Mean, Max, and Min from the BPPS matrices. Below is a snapshot of the derived features.



f_sum_bpps	f_mean_bpps	f_max_bpps	f_min_bpps
[0.19854229, 0.18371220000000002, 0.0600024000...	[0.0018555354205607479, 0.0017169364485981131, ...	[0.0217857, 0.0386527, 0.0275904, 0.00947066, ...	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...
[0.16883627604054915, 0.10675940059456464, 0.0...	[0.0015779091218742912, 0.0009977514074258377,...	[0.11931483477784201, 0.0808186531761711, 0.06...	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...
[0.06680724164624499, 0.04433748694733778, 0.0...	[0.0006243667443574299, 0.0004143690368910073,...	[0.017340043515196805, 0.008265665779309850, 0....	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...

Model Approach

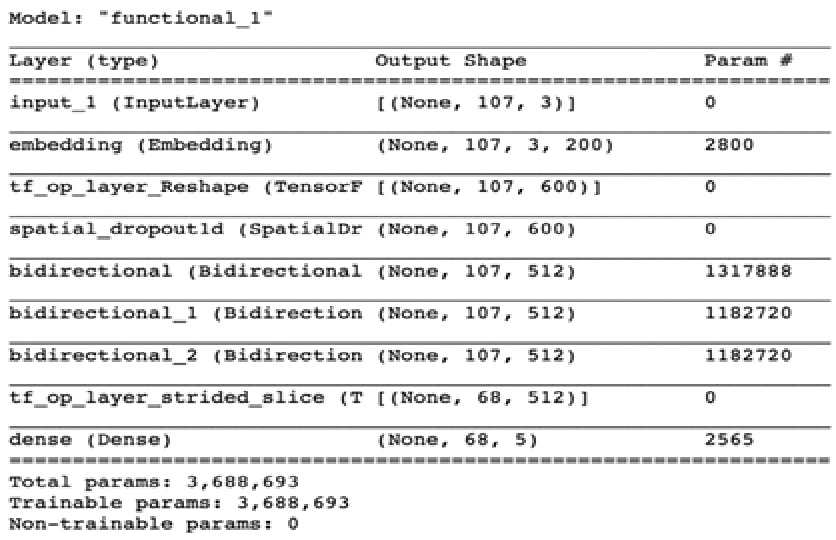
As we can see from the data, the input features ‘sequence’,’ structure’,’predicted_loop’ are all sequences of characters. The output 'deg_Mg_pH10', 'deg_pH10', 'deg_Mg_50C', 'deg_50C' are sequences as well. So we need to build a sequence to sequence model. And the RNN model is an ideal model to solve this kind of problem. Therefore, we choose to build RNN models. As we know, typical RNN models could be built by LSTM or GRU layers. In this project, Each group member has built one kind of neural network model to make predictions.

The model is designed to predict the five degeneration ratios based on criteria high temperature, degradation at high PH, etc. The 5 target variables are 'reactivity', 'deg_Mg_pH10', 'deg_Mg_50C', 'deg_pH10', 'deg_50C'.The model is fitted with the training data to optimize based on the loss function of MCRMSE(mean Column wise root mean squared error). We test different models from two perspectives: RNN layer structure and input embedding. All the models we tested are shown as follows:

Model	**RNN Structure**	**Input embedding**
Model 1	GRU-bidirectional(3 hidden layers	Embedding shape (107,3,200)
Model 2	GRU-non_bidrectional(3 hidden layers)	Embedding shape (107,3,200)
Model 3	GRU-bidirectional(3 hidden layers)	Embedding shape (107,3,100)
Model 4	GRU-bidirectional(2 hidden layers)	Embedding shape (107,3,100)
Model 5	GRU-bidirectional(4 hidden layers)	Embedding shape (107,3,100)
Model 6	GRU-bidirectional(3 hidden layers)	No embedding
Model 7	LSTM-bidirectional(3 hidden layers)	Embedding shape (107,7,200)

- **Model 1:**

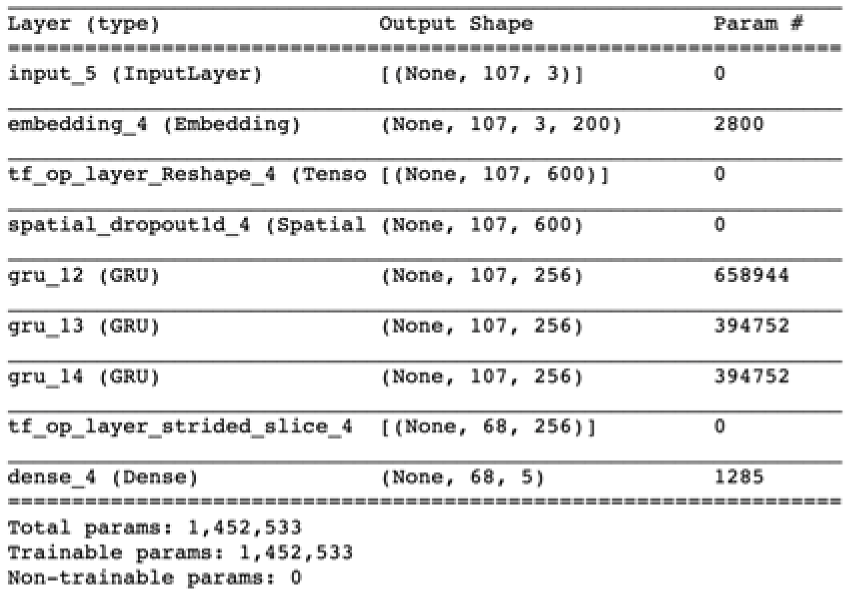
The embedding layer is added to the model. The three features ‘sequence’, ’structure’, ’predicted_loop’ are embedded in the dim before they are input into the GRU layers.



```
Model: "functional_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 107, 3)]          0
_________________________________________________________________
embedding (Embedding)        (None, 107, 3, 200)       2800
_________________________________________________________________
tf_op_layer_Reshape (TensorF [(None, 107, 600)]        0
_________________________________________________________________
spatial_dropout1d (SpatialDr (None, 107, 600)          0
_________________________________________________________________
bidirectional (Bidirectional (None, 107, 512)          1317888
_________________________________________________________________
bidirectional_1 (Bidirection (None, 107, 512)          1182720
_________________________________________________________________
bidirectional_2 (Bidirection (None, 107, 512)          1182720
_________________________________________________________________
tf_op_layer_strided_slice (T [(None, 68, 512)]         0
_________________________________________________________________
dense (Dense)                (None, 68, 5)             2565
=================================================================
Total params: 3,688,693
Trainable params: 3,688,693
Non-trainable params: 0
```

- **Model 2:**

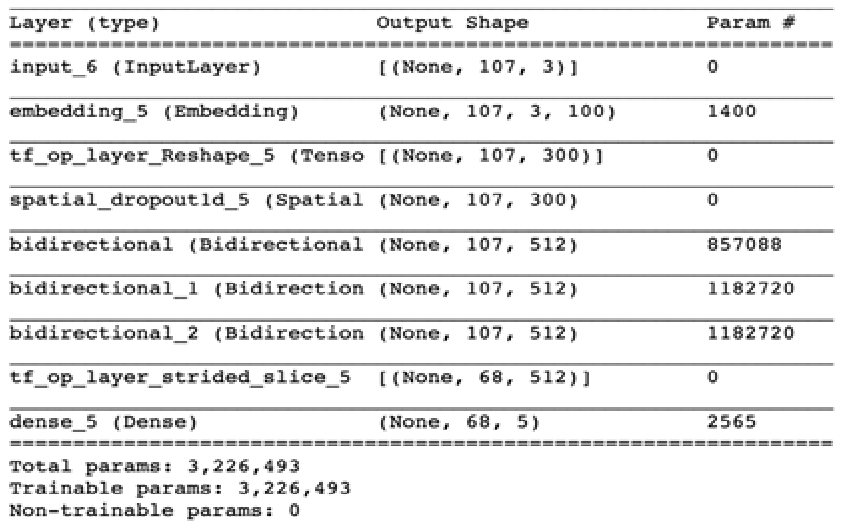
The bidirectional GRU layers in model 2 are changed to non-bidirectional layers. The three features ‘sequence’, ’structure’, ’predicted_loop’ are embedded in the dim before they are input into the GRU layers.



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_5 (InputLayer)         [(None, 107, 3)]          0
_________________________________________________________________
embedding_4 (Embedding)      (None, 107, 3, 200)       2800
_________________________________________________________________
tf_op_layer_Reshape_4 (Tenso [(None, 107, 600)]        0
_________________________________________________________________
spatial_dropout1d_4 (Spatial (None, 107, 600)          0
_________________________________________________________________
gru_12 (GRU)                 (None, 107, 256)          658944
_________________________________________________________________
gru_13 (GRU)                 (None, 107, 256)          394752
_________________________________________________________________
gru_14 (GRU)                 (None, 107, 256)          394752
_________________________________________________________________
tf_op_layer_strided_slice_4  [(None, 68, 256)]         0
_________________________________________________________________
dense_4 (Dense)              (None, 68, 5)             1285
=================================================================
Total params: 1,452,533
Trainable params: 1,452,533
Non-trainable params: 0
```

- **Model 3:**

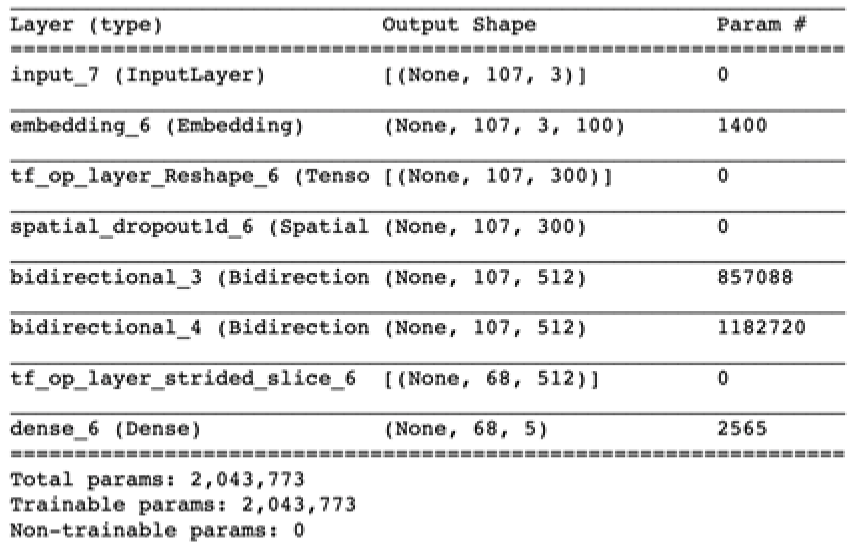
The dimension of the embedding layer in model 1 is changed to (107,3,100). The three features ‘sequence’, ’structure’, ’predicted_loop’ are embedded in the dim before they are input into the GRU layers.



```
Layer (type)                 Output Shape              Param #
=================================================================
input_6 (InputLayer)         [(None, 107, 3)]          0
_________________________________________________________________
embedding_5 (Embedding)      (None, 107, 3, 100)       1400
_________________________________________________________________
tf_op_layer_Reshape_5 (Tenso [(None, 107, 300)]        0
_________________________________________________________________
spatial_dropout1d_5 (Spatial (None, 107, 300)          0
_________________________________________________________________
bidirectional (Bidirectional (None, 107, 512)          857088
_________________________________________________________________
bidirectional_1 (Bidirection (None, 107, 512)          1182720
_________________________________________________________________
bidirectional_2 (Bidirection (None, 107, 512)          1182720
_________________________________________________________________
tf_op_layer_strided_slice_5  [(None, 68, 512)]         0
_________________________________________________________________
dense_5 (Dense)              (None, 68, 5)             2565
=================================================================
Total params: 3,226,493
Trainable params: 3,226,493
Non-trainable params: 0
```

- **Model 4:**

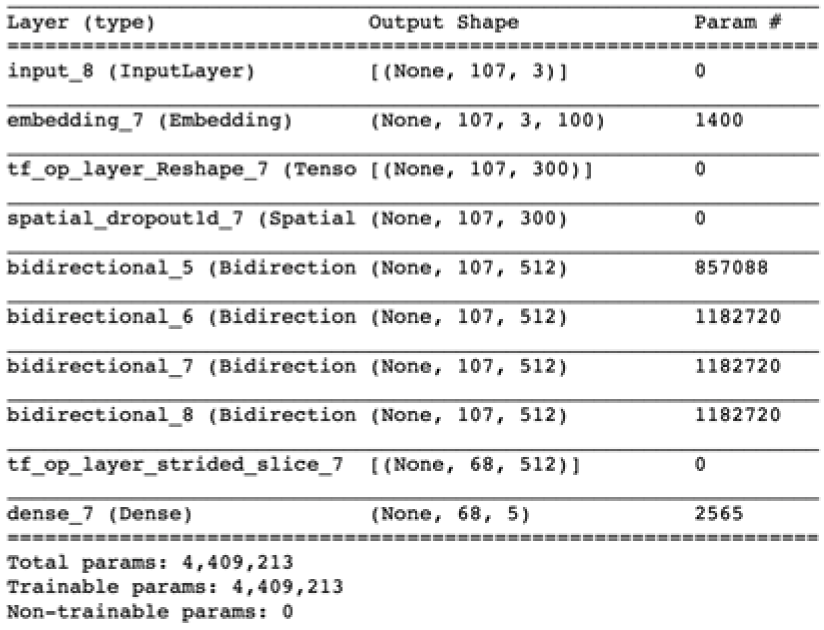
We remove one hidden GRU layer in model 3. The three features ‘sequence’, ’structure’, ’predicted_loop’ are embedded in the dim before they are input into the GRU layers.



```
Layer (type)                 Output Shape              Param #
=================================================================
input_7 (InputLayer)         [(None, 107, 3)]          0
_________________________________________________________________
embedding_6 (Embedding)      (None, 107, 3, 100)       1400
_________________________________________________________________
tf_op_layer_Reshape_6 (Tenso [(None, 107, 300)]        0
_________________________________________________________________
spatial_dropout1d_6 (Spatial (None, 107, 300)          0
_________________________________________________________________
bidirectional_3 (Bidirection (None, 107, 512)          857088
_________________________________________________________________
bidirectional_4 (Bidirection (None, 107, 512)          1182720
_________________________________________________________________
tf_op_layer_strided_slice_6  [(None, 68, 512)]         0
_________________________________________________________________
dense_6 (Dense)              (None, 68, 5)             2565
=================================================================
Total params: 2,043,773
Trainable params: 2,043,773
Non-trainable params: 0
```

- **Model 5:**

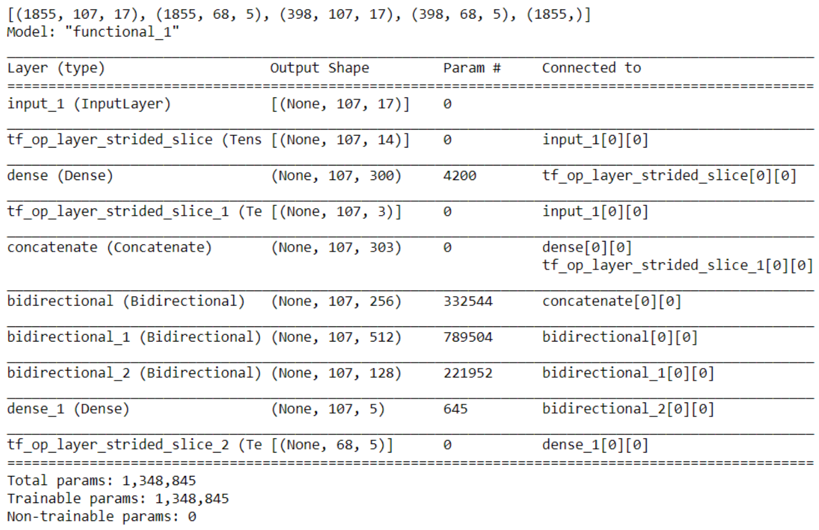
We add another hidden GRU layer in model 3. The three features ‘sequence’, ’structure’, ’predicted_loop’ are embedded in the dim before they are input into the GRU layers. embedding(dim100)+4GRU(bi-directional)



```
Layer (type)                 Output Shape              Param #
=================================================================
input_8 (InputLayer)         [(None, 107, 3)]          0
_________________________________________________________________
embedding_7 (Embedding)      (None, 107, 3, 100)       1400
_________________________________________________________________
tf_op_layer_Reshape_7 (Tenso [(None, 107, 300)]        0
_________________________________________________________________
spatial_dropout1d_7 (Spatial (None, 107, 300)          0
_________________________________________________________________
bidirectional_5 (Bidirection (None, 107, 512)          857088
_________________________________________________________________
bidirectional_6 (Bidirection (None, 107, 512)          1182720
_________________________________________________________________
bidirectional_7 (Bidirection (None, 107, 512)          1182720
_________________________________________________________________
bidirectional_8 (Bidirection (None, 107, 512)          1182720
_________________________________________________________________
tf_op_layer_strided_slice_7  [(None, 68, 512)]         0
_________________________________________________________________
dense_7 (Dense)              (None, 68, 5)             2565
=================================================================
Total params: 4,409,213
Trainable params: 4,409,213
Non-trainable params: 0
```

- **Model 6:**

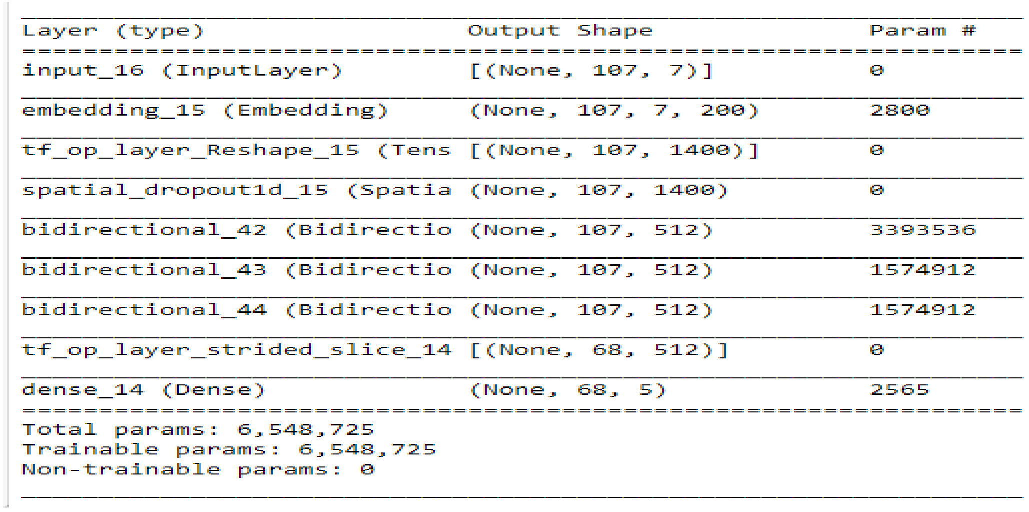
We remove the embedding layer in model 3 and change the dimensions of the three GRU layers. We also stop using the dropout layer in the model. The three features ‘sequence’, ’structure’, ’predicted_loop’ are the input of the model.



```
[(1855, 107, 17), (1855, 68, 5), (398, 107, 17), (398, 68, 5), (1855,)]
Model: "functional_1"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            [(None, 107, 17)]    0
__________________________________________________________________________________________________
tf_op_layer_strided_slice (Tens [(None, 107, 14)]    0           input_1[0][0]
__________________________________________________________________________________________________
dense (Dense)                   (None, 107, 300)     4200        tf_op_layer_strided_slice[0][0]
__________________________________________________________________________________________________
tf_op_layer_strided_slice_1 (Te [(None, 107, 3)]     0           input_1[0][0]
__________________________________________________________________________________________________
concatenate (Concatenate)       (None, 107, 303)     0           dense[0][0]
                                                                 tf_op_layer_strided_slice_1[0][0]
__________________________________________________________________________________________________
bidirectional (Bidirectional)   (None, 107, 256)     332544      concatenate[0][0]
__________________________________________________________________________________________________
bidirectional_1 (Bidirectional) (None, 107, 512)     789504      bidirectional[0][0]
__________________________________________________________________________________________________
bidirectional_2 (Bidirectional) (None, 107, 128)     221952      bidirectional_1[0][0]
__________________________________________________________________________________________________
dense_1 (Dense)                 (None, 107, 5)       645         bidirectional_2[0][0]
__________________________________________________________________________________________________
tf_op_layer_strided_slice_2 (Te [(None, 68, 5)]      0           dense_1[0][0]
==================================================================================================
Total params: 1,348,845
Trainable params: 1,348,845
Non-trainable params: 0
```

- **Model 7:**

The model architecture has 3 hidden LSTM layers. Although the original mRNA sequence length was 107, the sequence is truncated to align with the Kaggle competition. Below is the architecture of the model. Apart from the three core features ‘sequence’, ’structure’, ’predicted_loop’, we also derived 4 modules from BPPS. We used the BPPS data to derive the column-wise Sum, Mean, Max, and Min from the BPPS matrices.



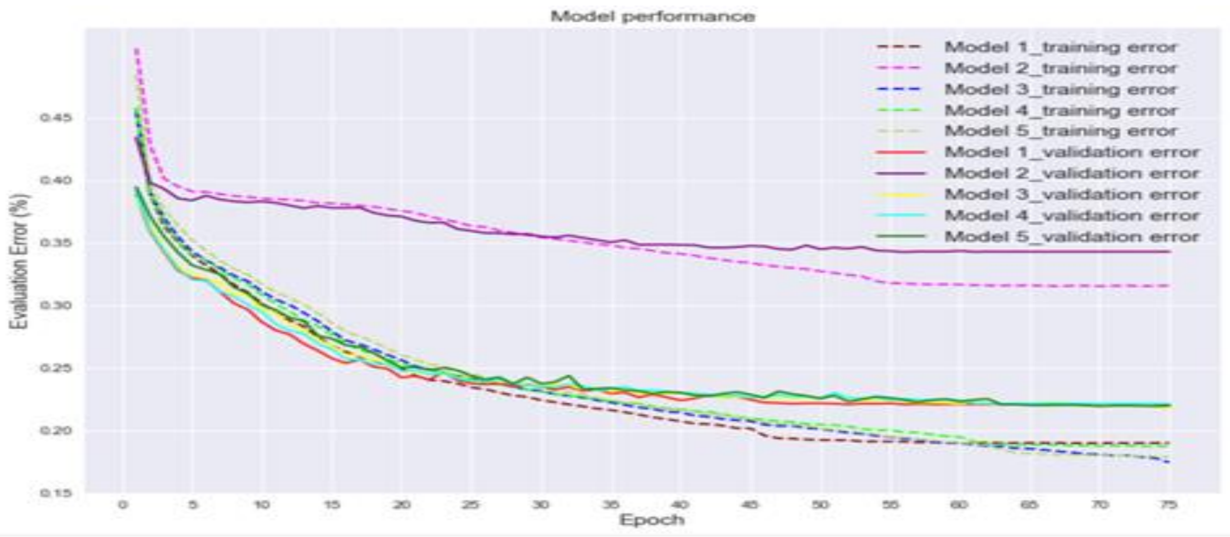
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_16 (InputLayer)        [(None, 107, 7)]          0
_________________________________________________________________
embedding_15 (Embedding)     (None, 107, 7, 200)       2800
_________________________________________________________________
tf_op_layer_Reshape_15 (Tens [(None, 107, 1400)]       0
_________________________________________________________________
spatial_dropout1d_15 (Spatia (None, 107, 1400)         0
_________________________________________________________________
bidirectional_42 (Bidirectio (None, 107, 512)          3393536
_________________________________________________________________
bidirectional_43 (Bidirectio (None, 107, 512)          1574912
_________________________________________________________________
bidirectional_44 (Bidirectio (None, 107, 512)          1574912
_________________________________________________________________
tf_op_layer_strided_slice_14 [(None, 68, 512)]         0
_________________________________________________________________
dense_14 (Dense)             (None, 68, 5)             2565
=================================================================
Total params: 6,548,725
Trainable params: 6,548,725
Non-trainable params: 0
_________________________________________________________________
```

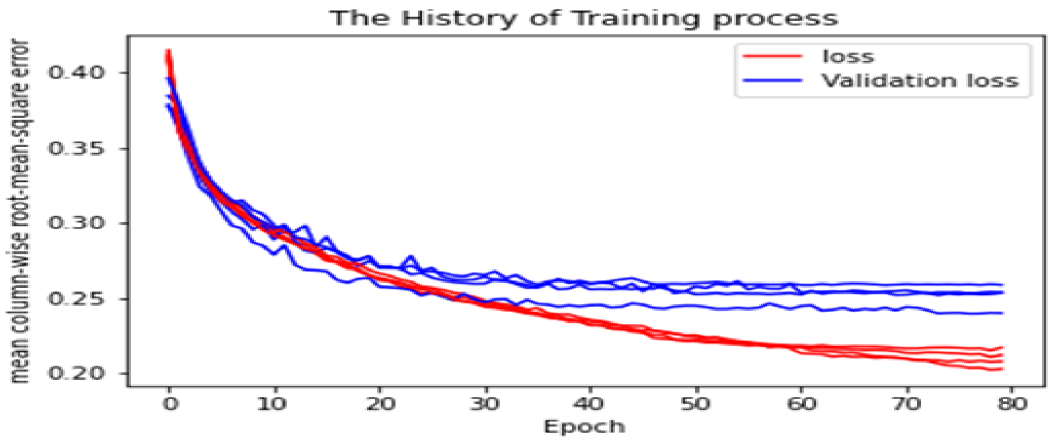
Model Evaluation

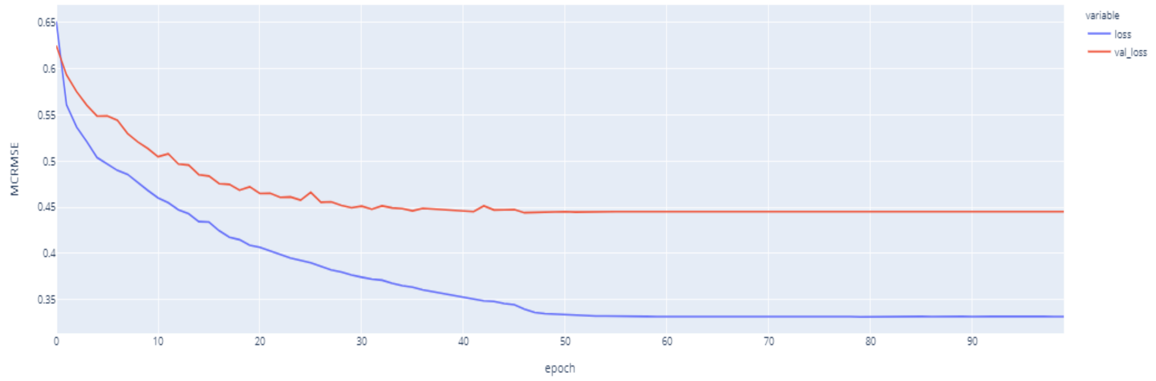
Model 1: The model with embedding vectors with length 200 and 3 bi-directional GRU layers has the optimal performance on the validation dataset.
Model 2: The bidirectional structure improves the performance a lot. The model with the 3 non-bi-directional GRU layers has the poorest performance.
Model 3: Reducing the size of the embedding vector does not hurt the performance of the neural network too much and it would cut the training time because we reduce the sum of the coefficients by 400,000.
Model 4: Increasing the hidden layer from 3 to 2 does not hurt the performance of the model. And it will save time on model training.
Model 5: Increasing the hidden layer from 3 to 4 does not improve the performance, but it requires a much longer time to train the model.
Model 6: Increasing the epoch would reduce the error on the training data, but its effect to reduce the validation error vanishes when the epoch goes over 80.
Model 7: Model replacing GRU layer with LSTM layer performs poorly. The additional features added from the BPPS matrix does not seem to improve the performance.

The below graph shows the convergence of all 7 models. The First Figure shows the convergence of the first 5 models, Model 1 to Model 5. We can see that as the epoch increases, the training error keeps decreasing, but the improvement of the model’s performance on the validation set is limited after 30 epochs.

The second graph shows the convergence of Model 6 for 4 different iterations.
The third graph shows the convergence of Model 7 with LSTM hidden layers.







Conclusion

Based on our implementation of different neural network architectures, we find that preprocessing data has a big impact on a model's performance. I use a simple method to convert characters to their location index in the string '(). ACGUBEHIMSX'. For example, '(' would be converted to 0 while 'X' would be converted to 13. It is a straightforward way to convert strings to data values that could be used as the input of a TensorFlow model in Keras. The drawback of this method is that 13 is much bigger than 0, but 'X' is not bigger than '('. A better way to do the transformation would be to use one-hot-encoding, but it would dramatically increase the size of the input.

Also beyond a certain point, increasing the architecture size does not increase the performance very much. Also, for this use case GRU seems to have better performance than the LSTM hidden layers. We also observed that the derived features from the BPPS matrix did not help much in improving accuracy. This might be because we used only the basic aggregations from the Matrix.

Some of the challenges we faced in the execution of the models are data volume and training time. Some of the models we used had a training time of around 12 hours and it was difficult to experiment with different parameters and features. Also, the data volume is quite huge to load it into the pandas data frame and apply the transformations. Although it was just around 2000 records in training data, each feature in itself an array, and each record had its own BPPS square matrix which made the problem complex.

Appendix: mRNA Advantages and Disadvantages

Finally, in this section, we discuss the general advantage and disadvantages of the mRNA vaccine in general. The below section describes have discussed the advantages of the mRNA vaccine in the introduction part. We also want to discuss the disadvantage of mRNA vaccine and the model training.

First, the goal of the mRNA sequence is to translate the RNA into the amino acid of the COVID19 spike protein. The RNA sequence can be replated by other sequences but the amino acid of S protein can't be replaced. For example, there is only one code (AUG) for the amino acid-Met. There are only two codes (UUU and UUC) for the amino acid-Phe. If C is found to be not stable following U, replacing C with U is the only option. So even if the model predicts the degradation rates of mRNA vaccine successfully, there is a limited solution to improve them. Second, there is no other mRNA vaccine reported before. So the data used for training is limited. If the development of mRNA vaccine lasts for decades, the mRNA sequence data is also accumulated. The data used for validation is also abundant. This will help the accuracy of our model. In our data set, only 6034 mRNA sequences are available. And half of them are training data and testing data. For each mRNA sequence, only part of the mRNA sequence is available. For example, in the training mRNA sequence 107, only 68 RNA base information is available. In the 130-validation mRNA sequence, only 91 mRNA base information is available. However, all mRNA sequence is important for training and the degradation rate analysis. We think this affects the accuracy of our model.

Third, the length used for training and validation is around 107 or 130. This includes the 3-D structure formed by the 100 RNA base. The loop types of the 3-D structure are important for model accuracy. However, the full length of the COVID-19 mRNA vaccine is 4000 bases. Any break of a base in the 4000 RNA will make the failure of the mRNA vaccine. The 3-D structure of the 4000 RNA is more complicated than the 100 RNA. This will contain more second structure of RNA. And the inside RNA base is more stable than the outside RNA base even if the linear base is not stable. And our model can't predict very well under these circumstances. Fourth, the Immunologic adjuvant is also important for the degradation of mRNA vaccine. The mRNA vaccine of Pfizer should be stored and distributed under the -80 cent degree. The vaccine of Moderna can be delivered under -20 cent degree. There is no significant difference between their mRNA sequence. The adjuvant will affect the degradation rate of the mRNA vaccine. However, we don't have any information in terms of adjuvant. If we have the material information of the adjuvant, the model prediction will be more accurate.

The study of mRNA vaccines is important for the development of the vaccine. The vaccine is more stable, the distribution of vaccines is faster and more convenient. Moreover, the dose used for injection may be less, this will reduce the side effect of people injected with the vaccine. Although the clinical trial result of the mRNA vaccine is encouraging and the FDA will approve the mRNA vaccine to use widely, our model is still useful. Because the COVID-19 virus always evolves, the mutation of the virus may make the mRNA vaccine useless. Then, the researcher needs to design the new mRNA sequence to adjust to the mutation of the virus. Our model will help with the development of a new mRNA vaccine.

References

• https://www.kaggle.com/c/stanford-covid-vaccine/overview
• https://www.kaggle.com/c/stanford-covid-vaccine/data