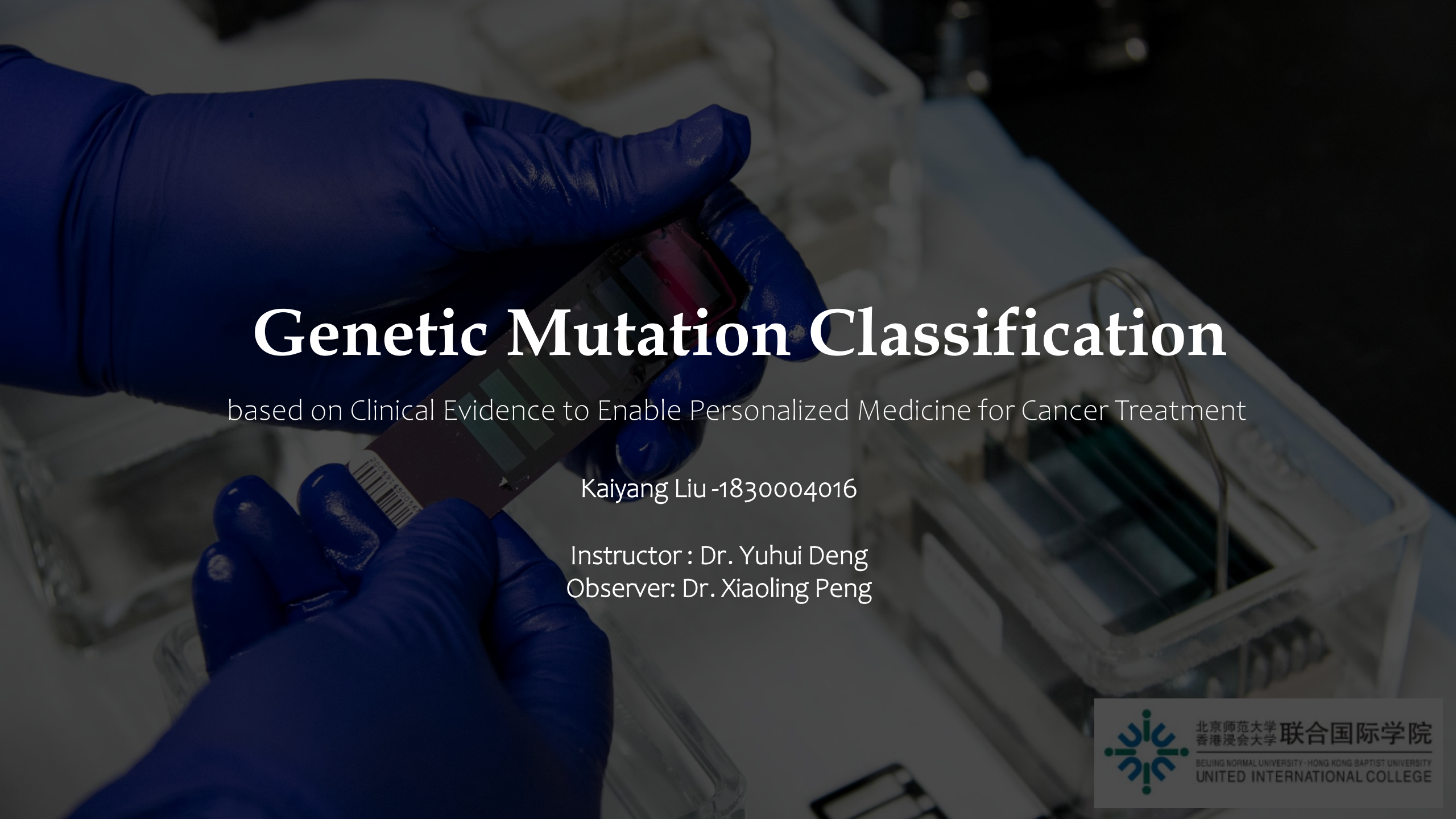


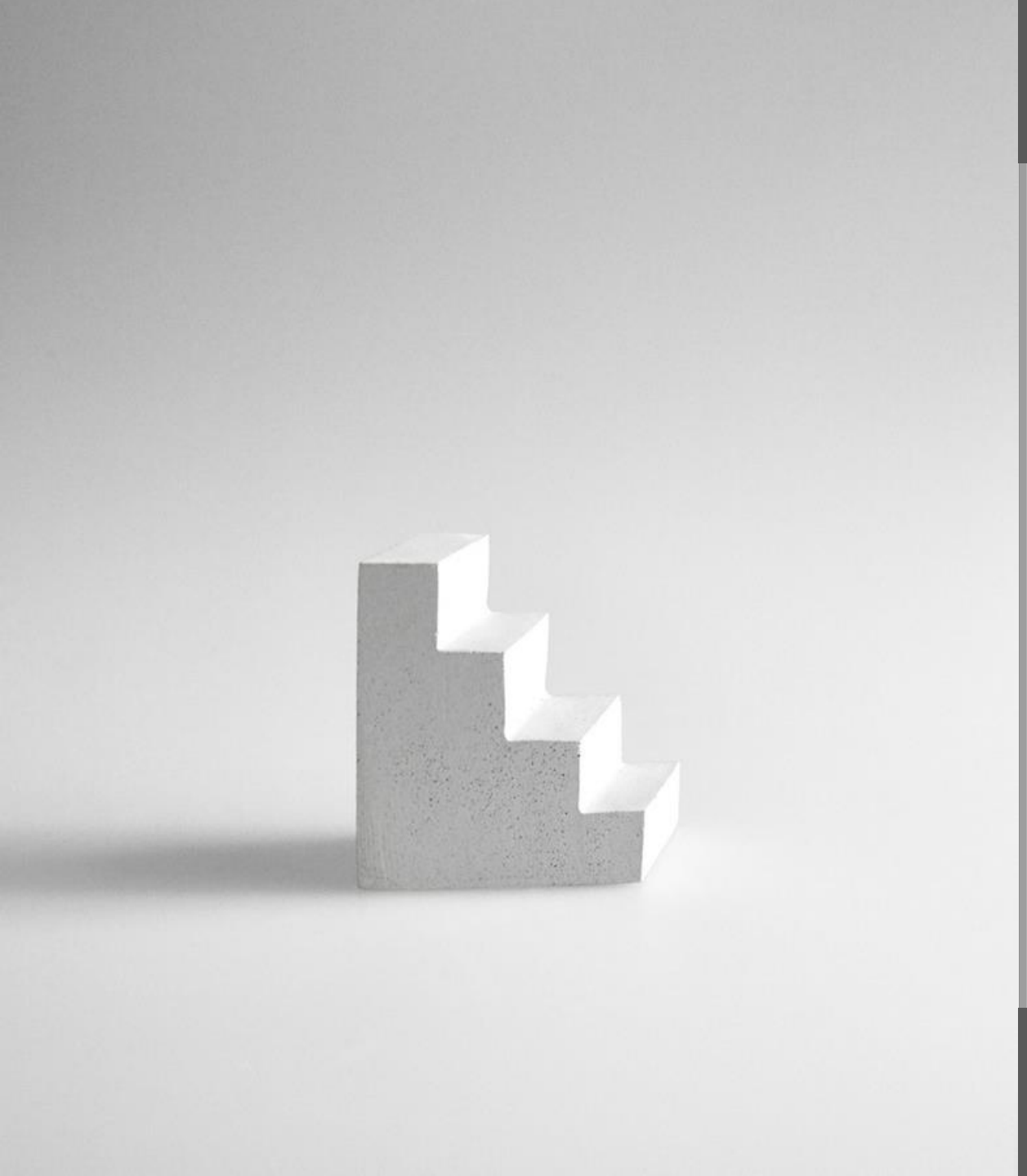
Genetic Mutation Classification

based on Clinical Evidence to Enable Personalized Medicine for Cancer Treatment

Kaiyang Liu -1830004016

Instructor : Dr. Yuhui Deng
Observer: Dr. Xiaoling Peng

北京师范大学
香港浸会大学 联合国际学院
BEIJING NORMAL UNIVERSITY - HONG KONG BAPTIST UNIVERSITY
UNITED INTERNATIONAL COLLEGE



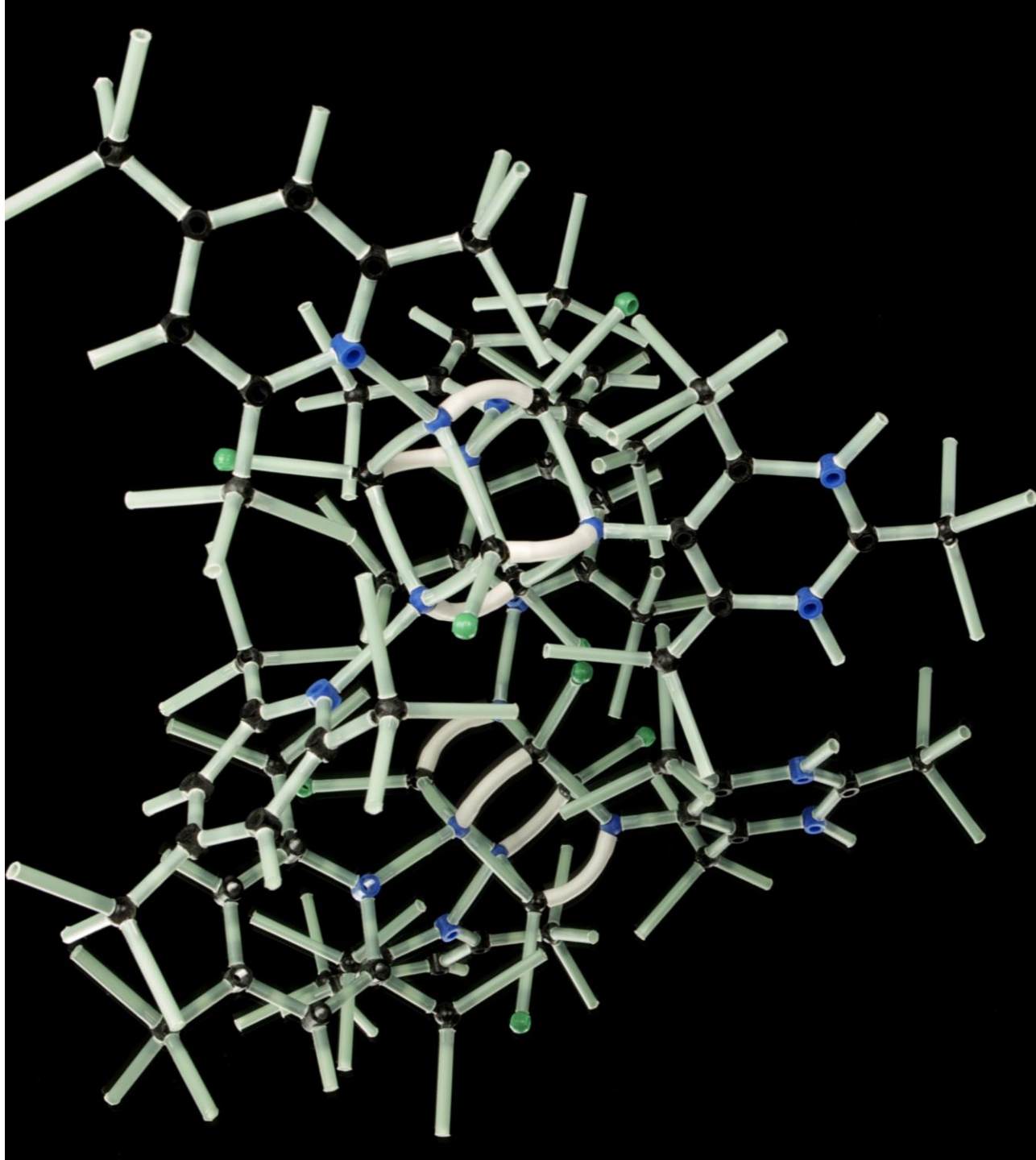
Content

Part 1 - Introduction

Background Information

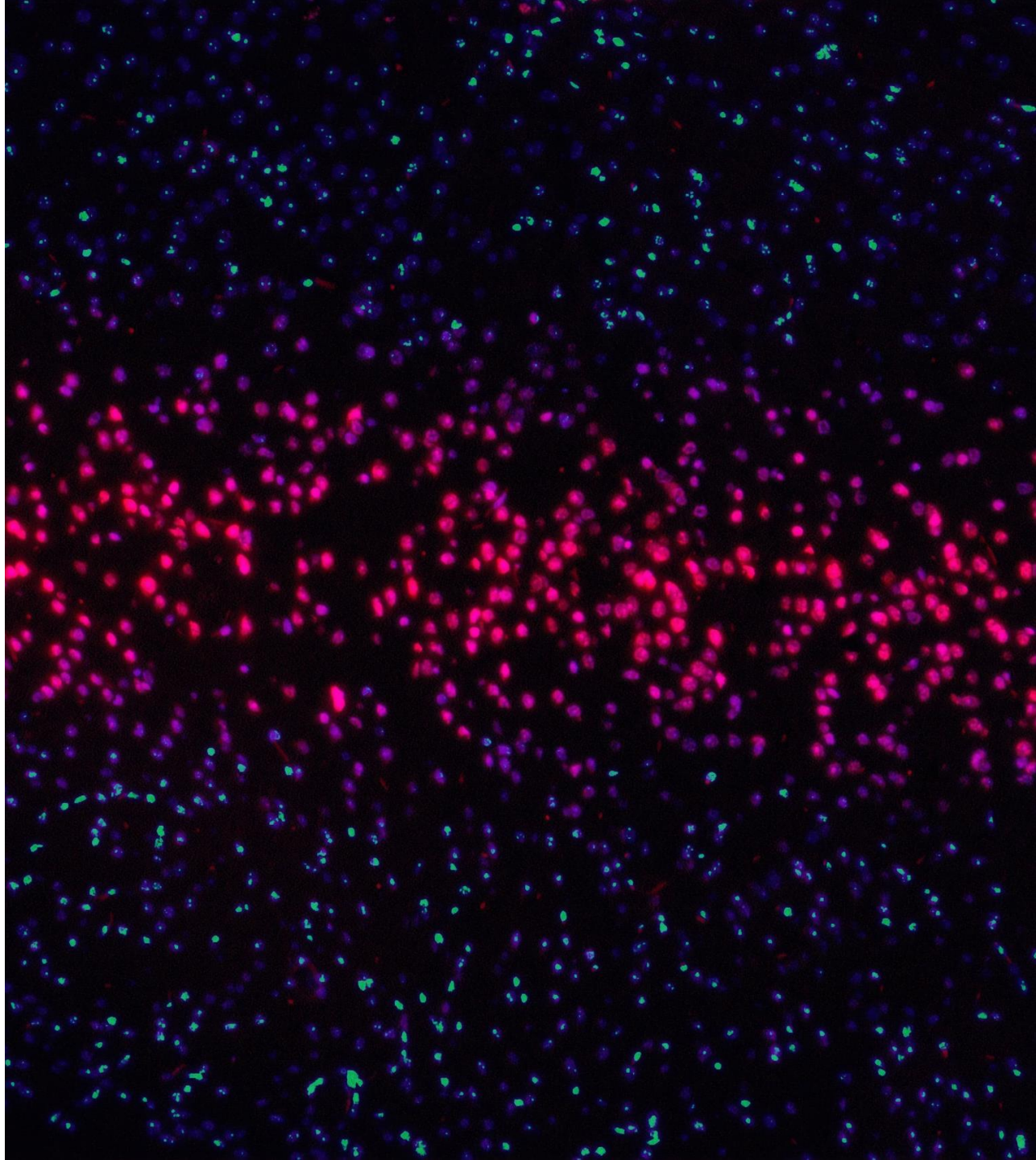
According to International Agency for Research on Cancer, nearly ten million death in 2020 make cancer continually become the disease that dominates death around the world. Cancer is caused by genetic mutation and normal cells thus transform into tumor cells. Since tumors always contain many types of genetic. Therefore, more attention should be paid to heterogeneity to enable personalized cancer treatment. That is also to say, identification and classification of the particular type of gene mutation from the clinical documents that cause cancer are essential.

However, this process suffers from multiple drawbacks. The identification of genetic mutation requires professional knowledge in gene and medicine science. Also, the interpretation of those clinical texts is based on the subjective perspective from person to person even though they are professional because there are huge difference between patients. It is complicated and extremely time consuming.



Motivations and Project Objective

The ultimate purpose of this project is to develop some classification models to give the classified result of the gene mutation classes. Based on the clinical evidence, different genetic mutation types are asked to be classified by referring to the information given in the gene variation document. In the future, with the development of the application of machine learning and deep learning methods in clinical text, personalized medicine will be facilitated and benefit cancer treatment.



Motivations and Project Objective

In the past…

Medical record → (doctor) → Cancer treatment

How to make some changes to build a medical database? → Doctor annotated

By now …

Clinical documents with class label that are annotated by doctors → Classification Model

Clinical documents without labels → (Model) → Classes

OUR FOCUS!

In the future …

Medical record → (Model) → Personalized cancer treatment

What we are going to do?

ID	Text
1	Abstract Background Non-small cell lung cancer (NSCLC) is a heterogeneous group of disorders with a number of genetic and proteomic alterations. c-CBL is an E3 ubiquitin ligase and adaptor molecule important in normal homeostasis and cancer. We determined···

ID	Gene	Variation	Class
1	CBL	W802*	2

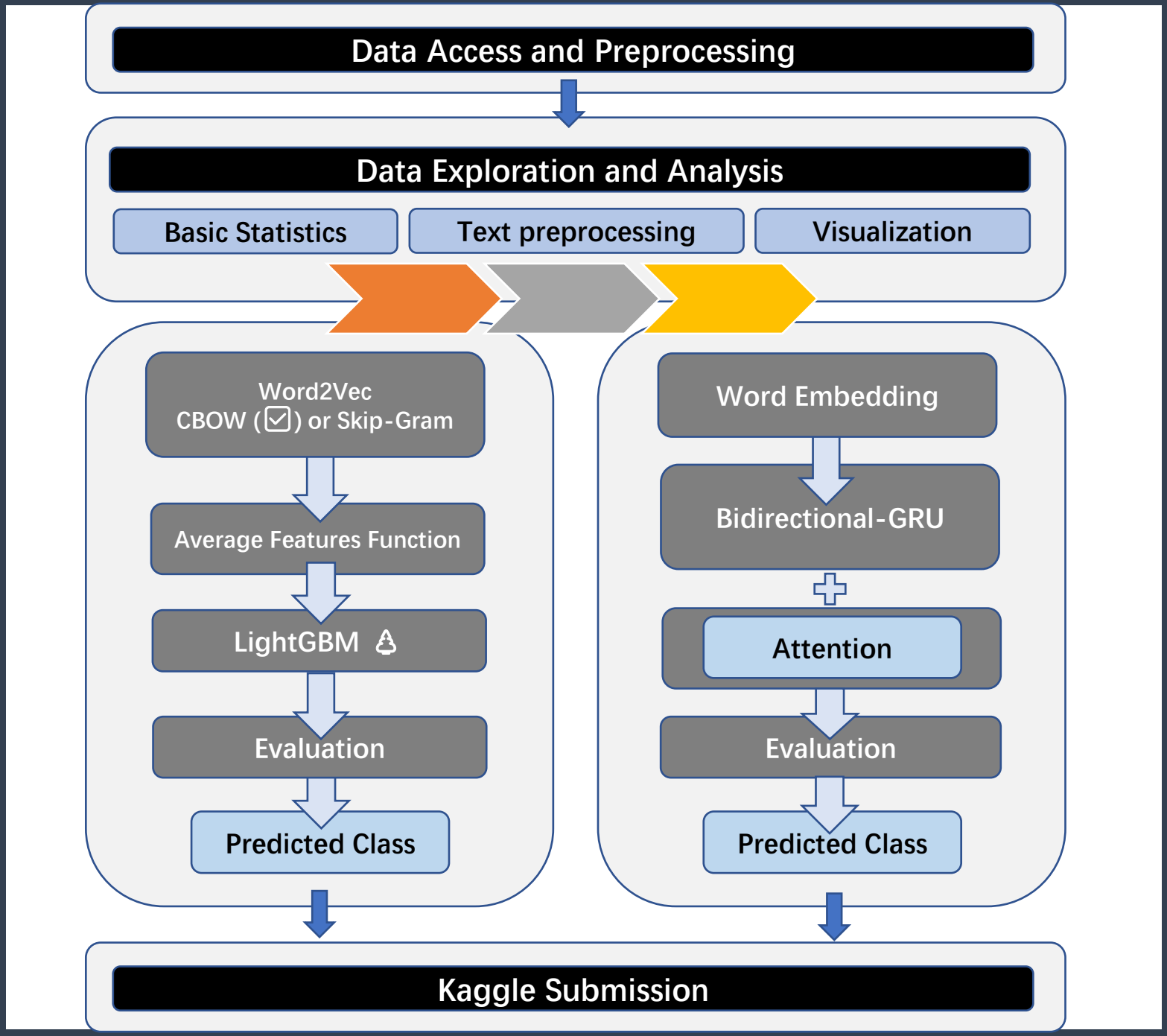
Genetic mutation type are divided into 9 Classes,
and it was Annotated By the doctor. What if we don't have a doctor?

Now, the text information become our "doctor", because we are going to learn from it and become more and more professional to automatically give the class result using the classification model!

Related Work

Machine Learning in Classification
Mark Singh et al; Naïve Bayesian; Detect abnormal radiology
Adam Wright et al.; SVM; Identifying HER progress documents
Ti Kam Ho; Random decision forests
Jerome H. Friedman; Gradient Boosting Decision Tree
Chen; Xgboost;
Gupta et al.; random forest, Xgboost; Gene classification
Xuan Qin et al. LightGBM
Deep Learning in Classification
Meenu Gupta et al.; word2Vec+RNN; Gene classification
Hochreiter; LSTM
J Chung; GRU
Tang et al; Bi-LSTM-CRF
Mnih; Attention + RNN
Yujia Bao et al.; CNN; cancer susceptibility genes
Yanshan Wang et al; CNN; clinical text classification

Overview Diagram



Part 2 – Exploration of data and Analysis

2.1 Dataset Description

Dataset	Description	Purpose
Training_variants.csv	3321x4	Training and Testing
Training_text.csv	3321x2	
Stage2_test_variants.csv	986x3	For submission
Stage2_test_test.csv	986x2	
test_variants.csv	5668x3	Validation
test_text.csv	5668x2	

ID	Gene	Variation	Class

ID	Gene	Variation

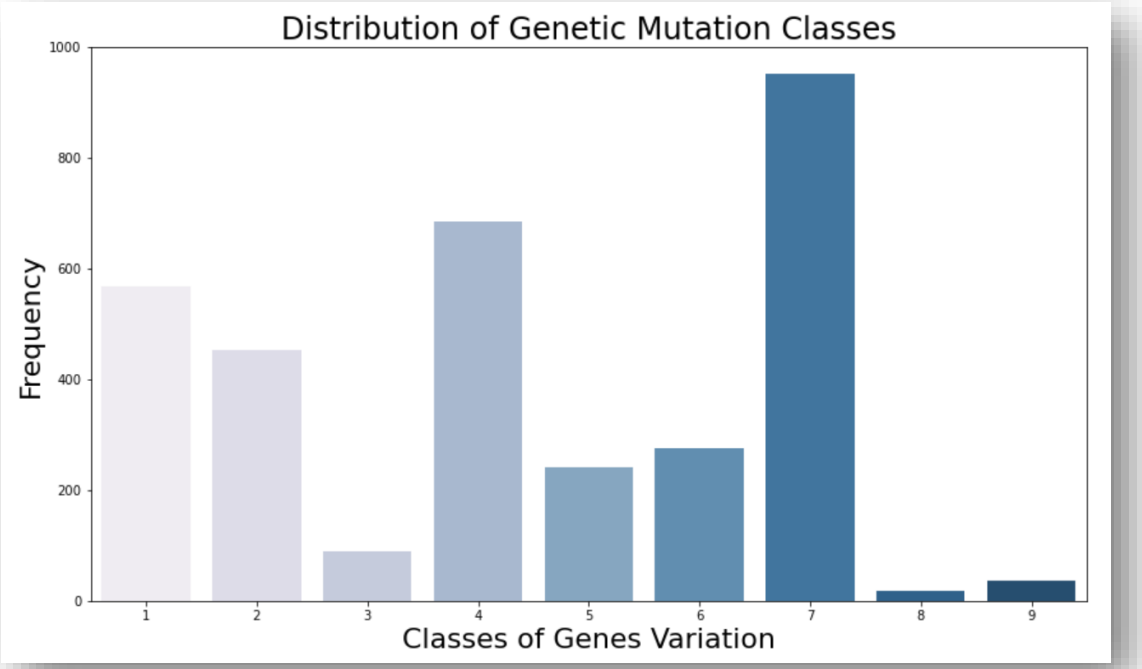
ID	Text

2.1 Dataset Description (2)

ID	Gene	Variation	Class
0	FAM58A	Truncating Mutations	1
1	CBL	W802*	2
2	CBL	Q249E	2
3	CBL	N454D	3
4	CBL	L399V	4

ID	Text
0	Cyclin-dependent kinases (CDKs) regulate a variety of fundamental cellular processes. CDK10 stands out as one of the last orphan CDKs for which no activating cyclin has been identified and no kinase activity revealed. Previous work has shown that CDK10
1	Abstract Background Non-small cell lung cancer (NSCLC) is a heterogeneous group of disorders with a number of genetic and proteomic alterations. c-CBL is an E3 ubiquitin ligase and adaptor molecule important in normal homeostasis and cancer. We determine
3	Recent evidence has demonstrated that acquired uniparental disomy (aUPD) is a novel mechanism by which pathogenetic mutations in cancer may be reduced to homozygosity. To help identify novel mutations in myeloproliferative neoplasms (MPNs), we performed a
4	Oncogenic mutations in the monomeric Casitas B-lineage lymphoma (Cbl) gene have been found in many tumors, but their significance remains largely unknown. Several human c-Cbl (CBL) structures have recently been solved depicting the protein at different st

2.2 Data Analysis (1)



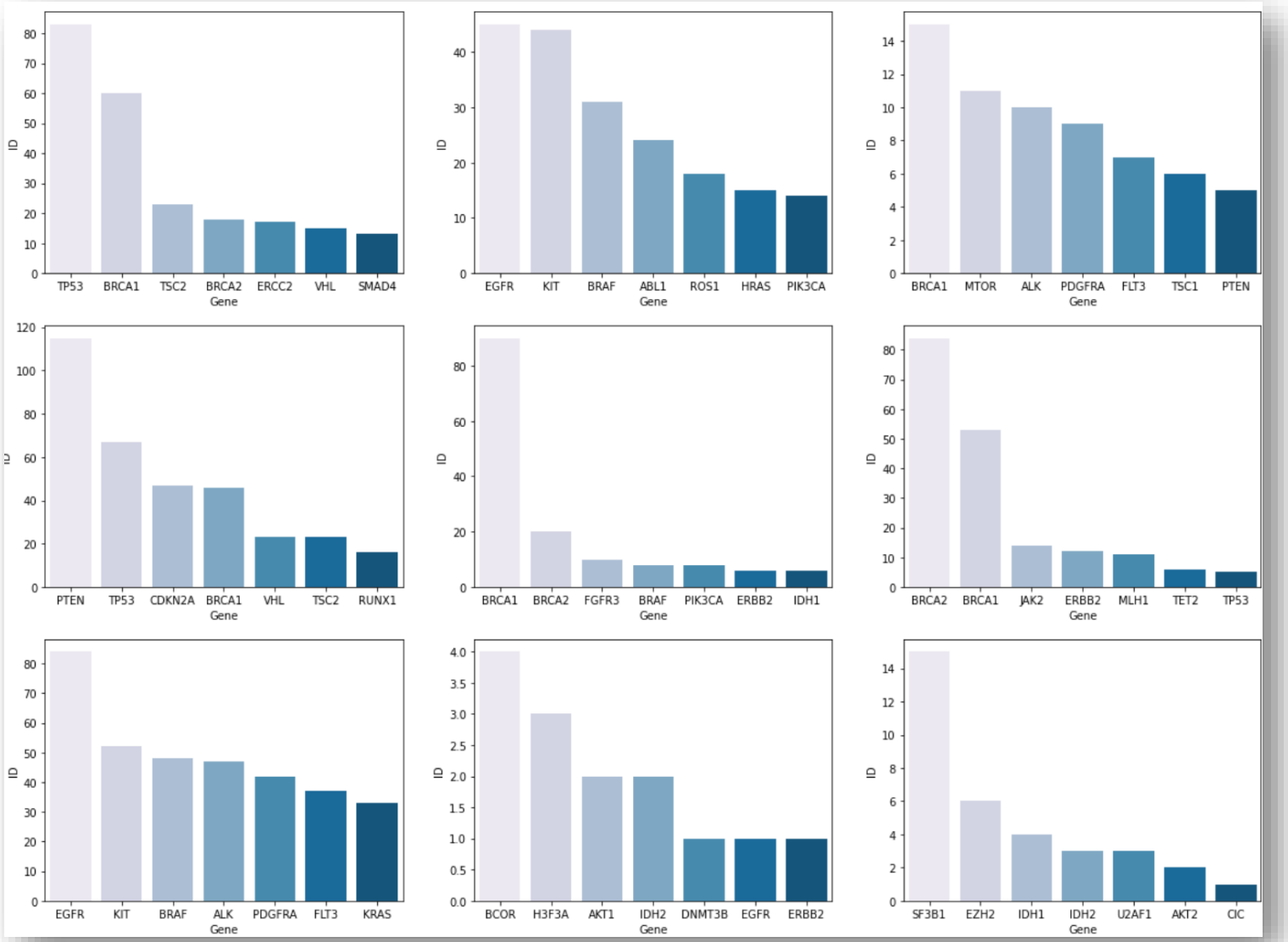
Genes with maximal occurences	
Gene	
BRCA1	264
TP53	163
EGFR	141
PTEN	126
BRCA2	125
KIT	99
BRAF	93
ALK	69
ERBB2	69
PDGFRA	60

Genes with minimal occurences	
Gene	
KLF4	1
FGF19	1
FANCC	1
FAM58A	1
PAK1	1
ERRFI1	1
PAX8	1
PIK3R3	1
PMS1	1
PPM1D	1

Class 7 have the highest frequency and class 3, 8, 9 have a relatively lower frequency.

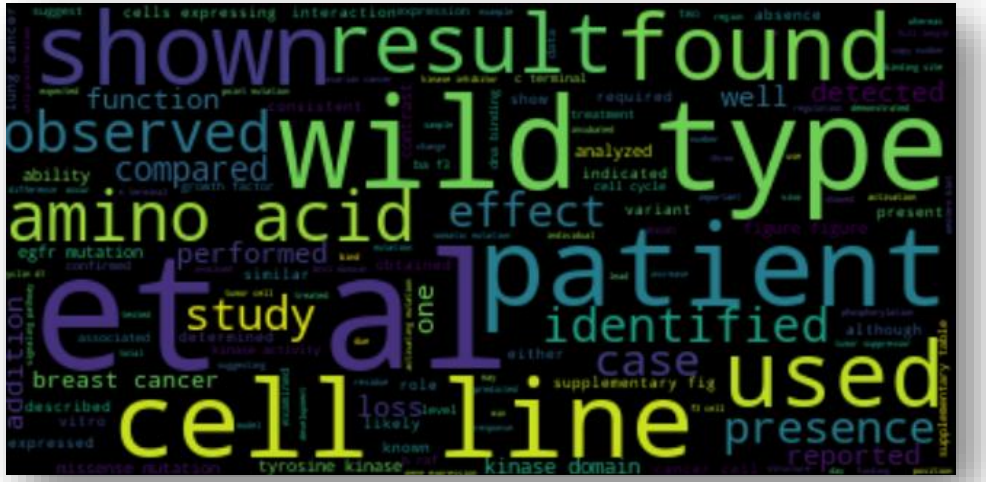
BRCA1 has the maximal occurrences time, others like KLF4, FGF19, FANCC… have the minimal occurrences.

2.2 Data Analysis (2)



The figure show the distribution of genes in 9 different classes. We can get the conclusion that some of the genes are highly dominating in their class. For example, Gene BRCA1 has the highest frequency and dominates in class 5.

2.3 Text Processing (1)



Words like "patient", "cell", "line", "amino", "acid" are frequently used in our clinical documents.

When dealing with text data, text preprocessing is always the most important part that we should consider.

General Steps

(I) Tokenization is applied to separate the sentence from words.

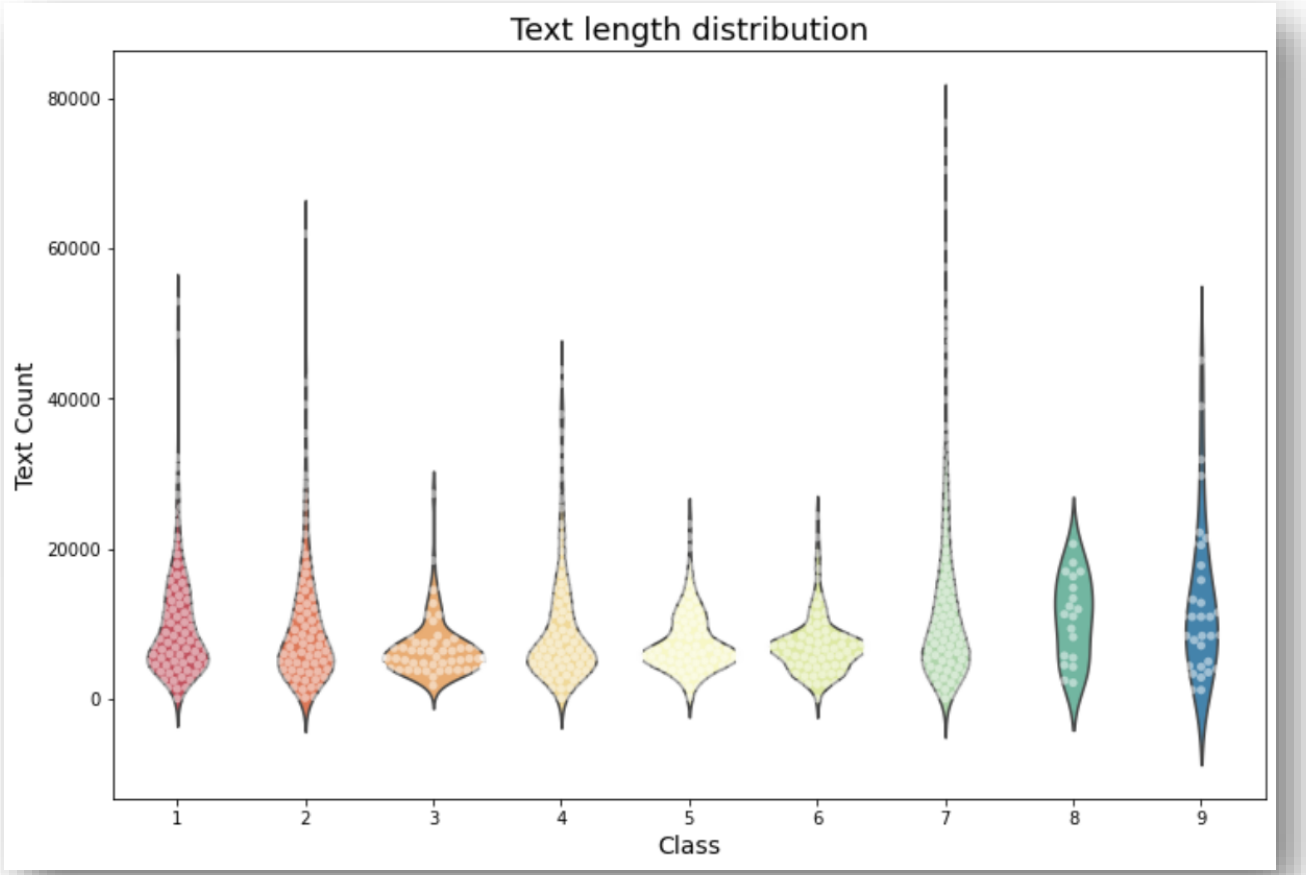
(II) The removal of punctuation and lower casting have proceeded.

(III) Some relatively meaningless stop words are removed. For example, "a", "and", "but" and so on.

2.3 Text Processing (2)

Class	count	mean	std	min	25%	50%	75%	max
1	568.0	9441.841549	6511.773899	1.0	4969.75	7302.0	12866.25	52918.0
2	452.0	9304.159292	7621.158837	116.0	4185.00	6808.0	12219.50	61945.0
3	89.0	6749.213483	3712.931889	1737.0	4283.00	5571.0	7409.00	27290.0
4	686.0	8975.769679	7270.444322	53.0	4560.00	6351.0	11536.25	43812.0
5	242.0	7504.384298	3895.755024	183.0	5245.00	6426.0	9513.00	24130.0
6	275.0	7177.952727	3833.400979	1.0	4498.50	6587.0	7847.00	24519.0
7	953.0	11433.295908	10104.998688	1.0	4871.00	8254.0	14592.00	76733.0
8	19.0	10809.368421	5645.232888	2111.0	5586.00	11248.0	15529.00	20615.0
9	37.0	12795.675676	10208.050296	1147.0	4937.00	10910.0	15791.00	45078.0

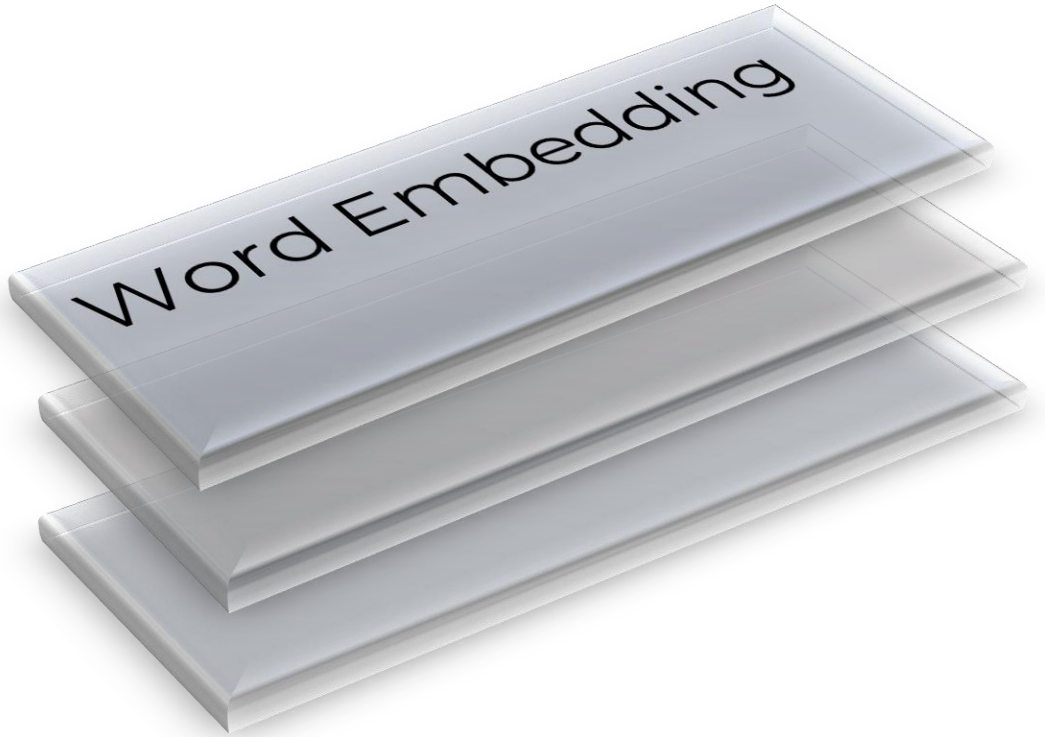
The statistics description for Word count



Texts in class 7 have a huge amount of words compared to the classes like 3,4,5,6, and 8. The average word count for class 7 is about 11000 words, and the highest one reaches 76733-word counts. Besides, most of the text in class 3 is around 7000 words.

Part 3 – Methodology

3.1 Model 1 LightGBM



For Model 1

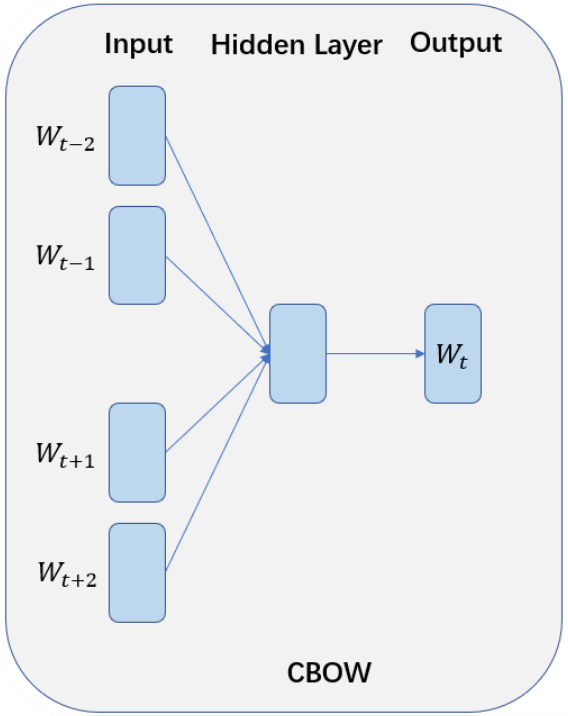
Word2Vec + Average Feature Model

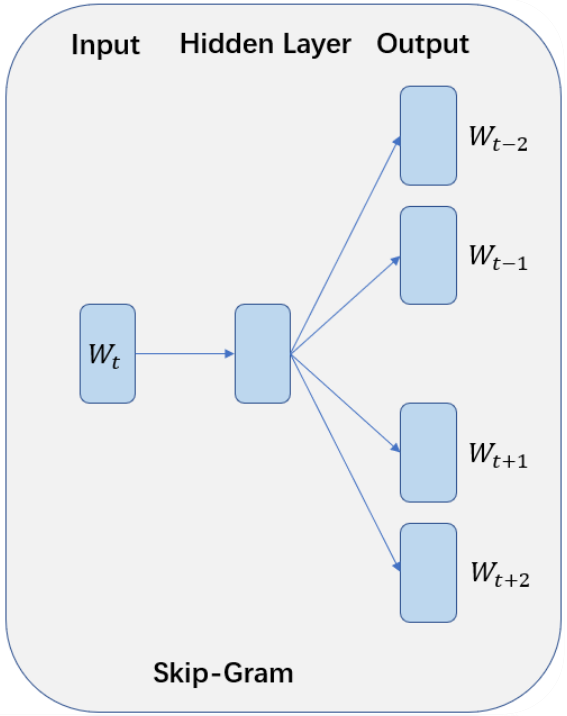
Word Embedding - Word2Vec + Average Feature Model

One hot encoding

I love Statistics->

$$\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}$$





I ~~love~~ Statistics -> love

~~I~~ love ~~Statistics~~ -> I, Statistics

Input of word2vec model

# of training sample

Text

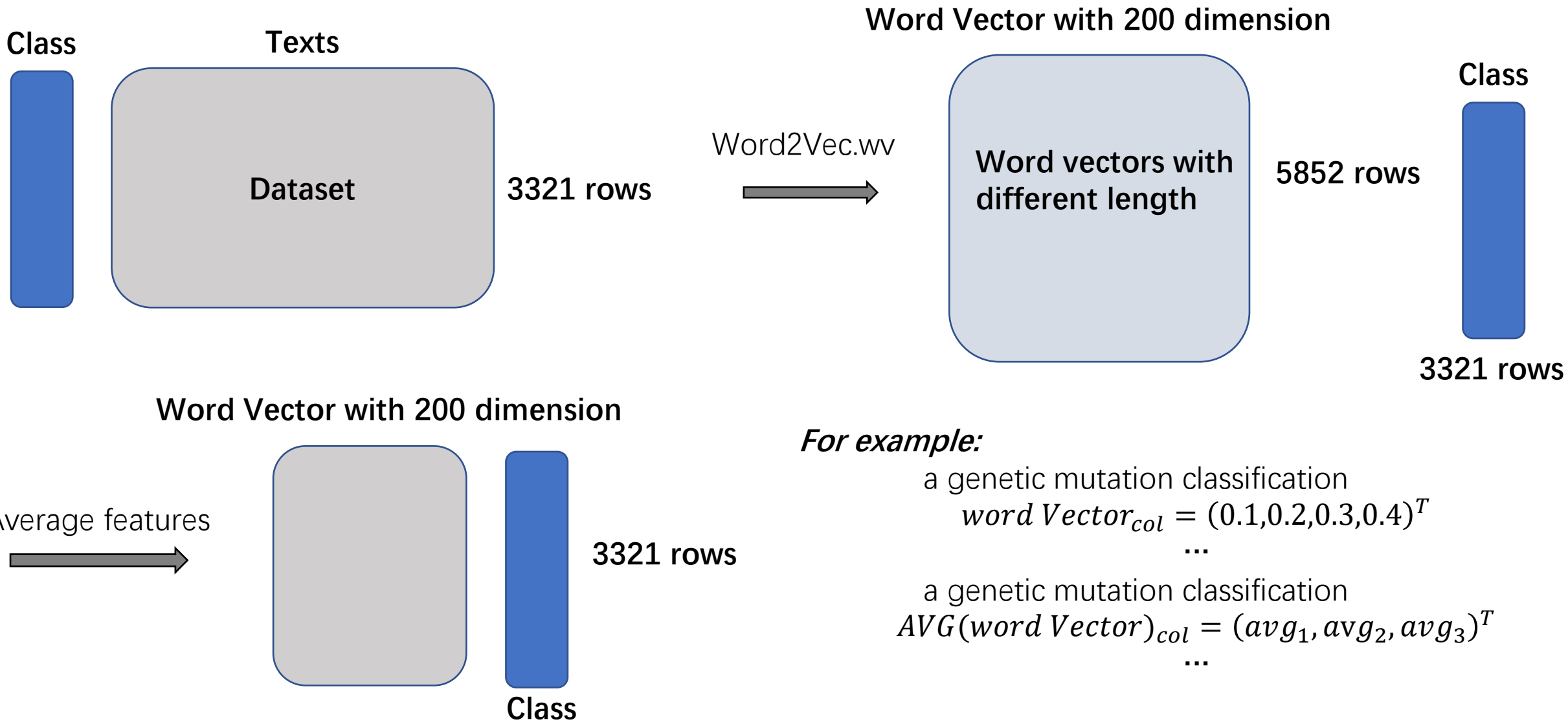
# of training sample

Word Vector

The weight matrix of the network

Text representation for Model 1

Word Embedding - Word2Vec + Average Feature Model



For example:

a genetic mutation classification

$word\ Vector_{col} = (0.1, 0.2, 0.3, 0.4)^T$

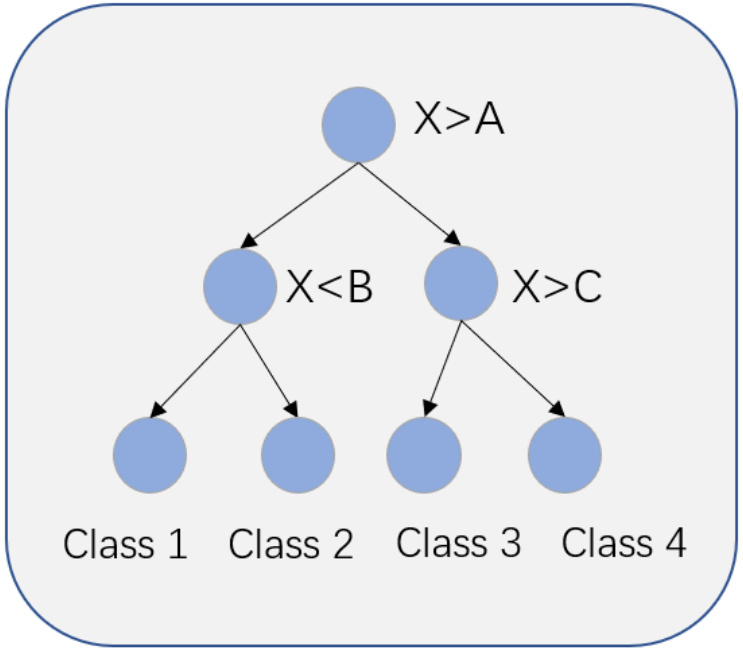
...

a genetic mutation classification

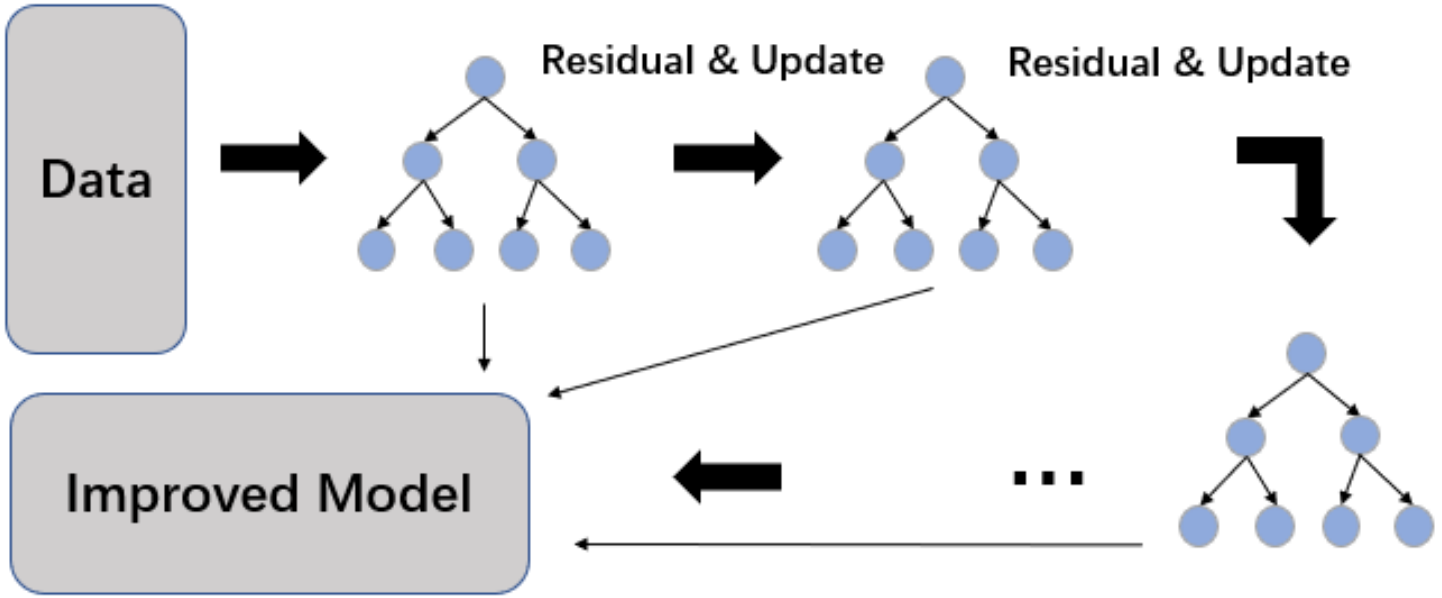
$AVG(word\ Vector)_{col} = (avg_1, avg_2, avg_3)^T$

...

3.2.1 Background Information



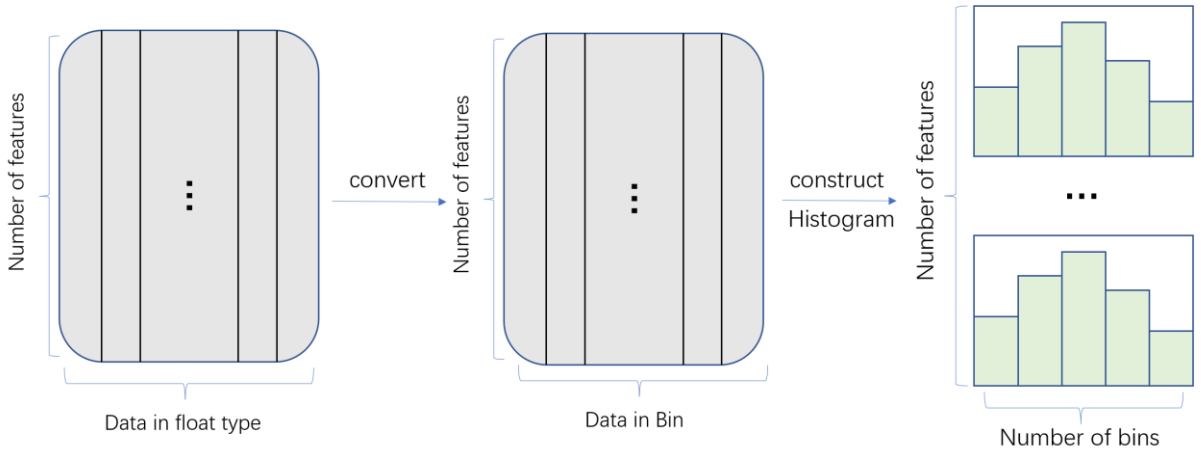
A Simple decision tree



Gradient Boosting Decision Tree

3.2.2 Basic Concepts and Algorithm (1)

(I) Histogram-based Algorithm



Goal: Make the storage of data easier and let the computation more efficient

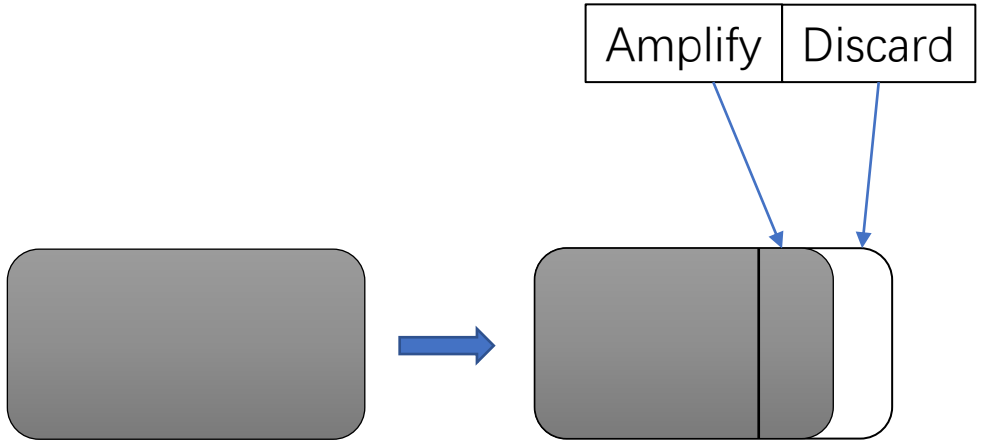
1. Determine the number of bins are required for each feature and assign an integer number to each box.
2. Divide the range of floating points into several intervals, where the number of intervals should equal the number of the bin.
3. Update the sample value in the bin to the new value corresponding to each bin.

The large scale data are simplified. And the features have become discrete and easy to store with low cost of computation.

(II) Gradient-Based One-Side Sampling

A compromise between the learning accuracy and reducing the size of the data set.

Goals: Reduce the samples with the smaller gradients and use the remaining part of the samples to calculate the information gain in the LightGBM.



Size of dataset are reduced!

Since directly regard the information will cost some loss, we would like to discard part of the sample and amplify the remaining part

As a result, the distribution of the original sample data with a lower gradient will be discarded with less cost and the learning accuracy will not be affected so much at the same time.

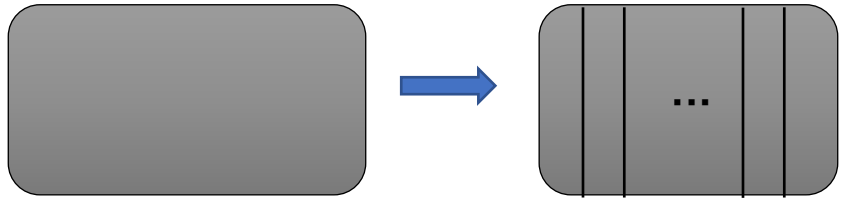
3.2.2 Basic Concepts and Algorithm (2)

(III) Greedy Bundling

Goal: Reduce the number of features by bundling the features to increase computation efficiency.

Find some of the features that are mutually exclusive so that we can bundle them together to reduce the dimension and optimize the time cost for the algorithm.

Number of features are reduced!

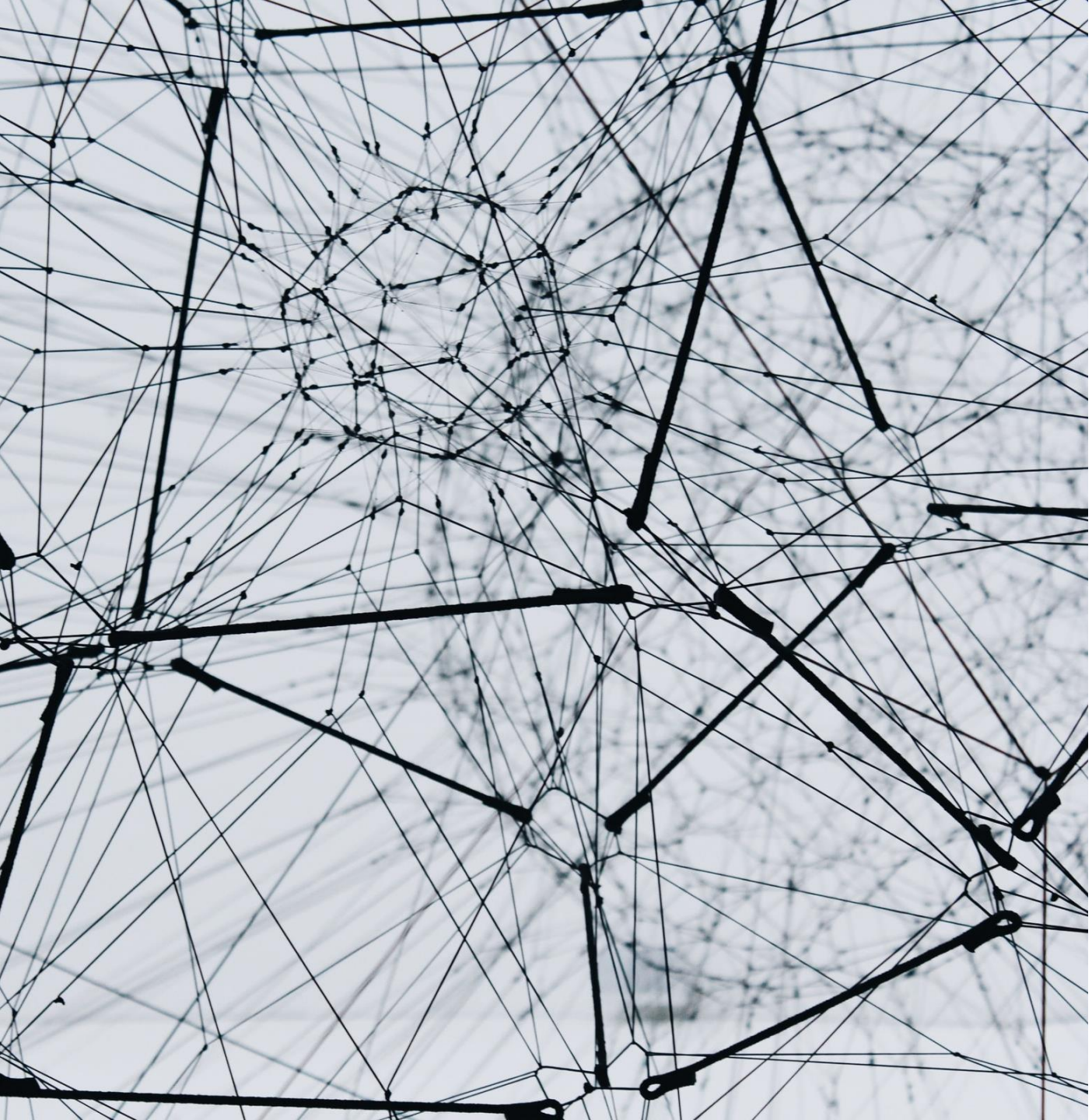


(IV) Merge Exclusive Features

Completes the remaining jobs of the greedy bundling algorithm by Constructing the bundling part.

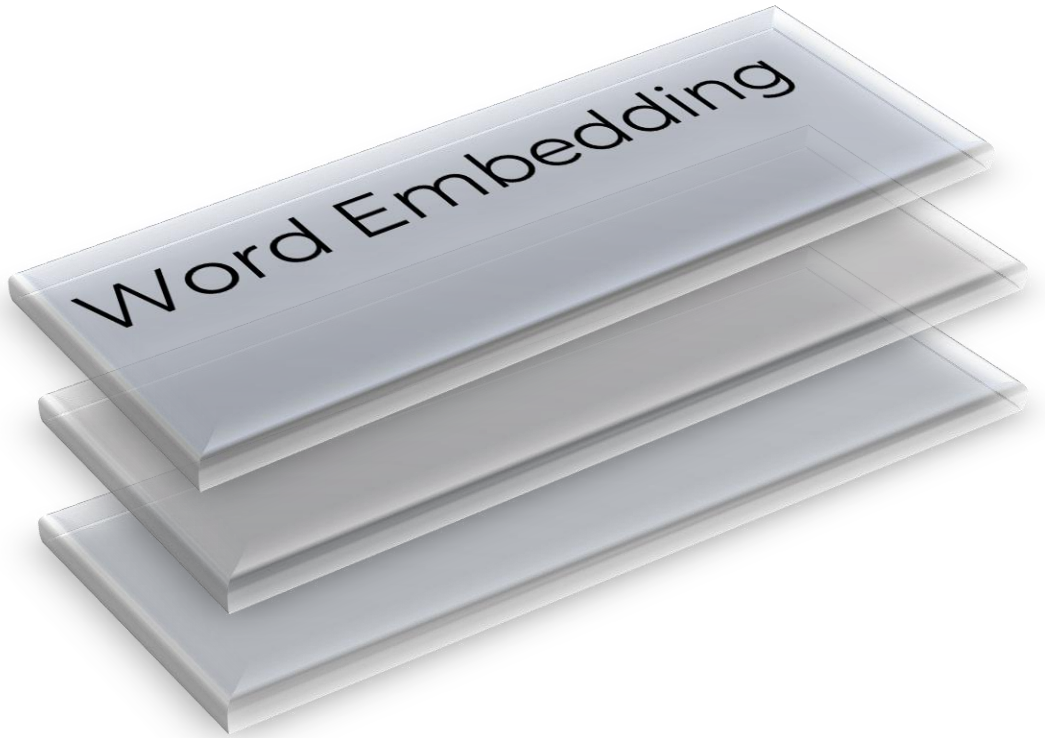
Since we have already transformed the float data into different bins, it is a good idea for us to make the merged features set in the bins to make sure the data in the bundled features can still be recognized.

To achieve this goal, we can add some offset on the original data to ensure that the range of two exclusive features is merged safer.



3.3 Model 2

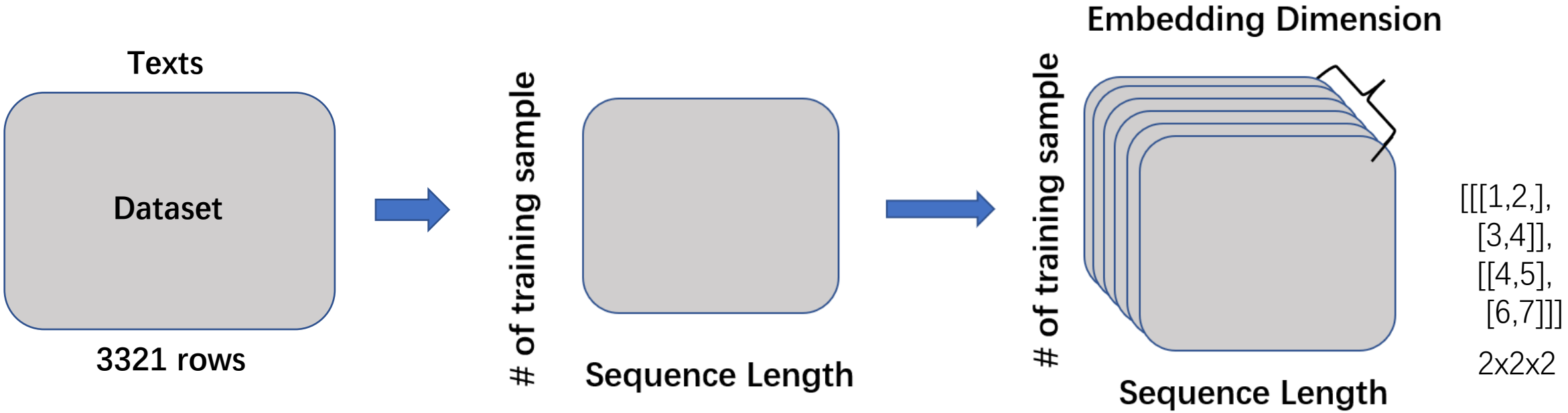
Bidirectional GRU + Attention Context



For Model 2

Word Embedding in Keras

3.1.2 Word Embedding in Keras

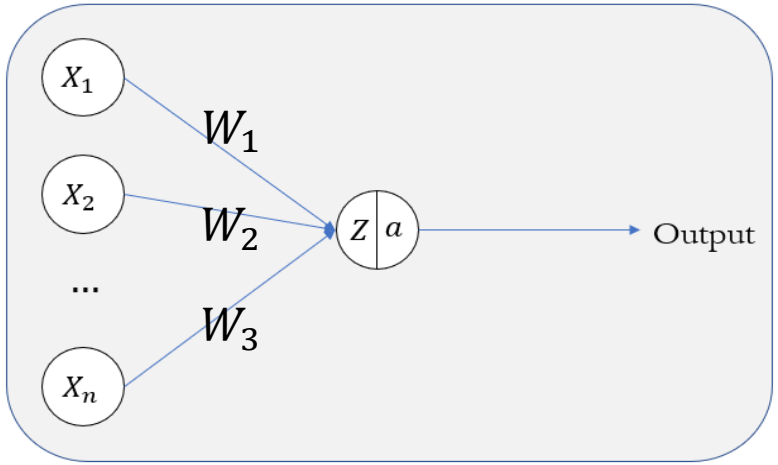


1. Use Tokenizer to change texts to sequence, where the length of sequence is different.
2. Padding, let the sequences to be the same length.

3. Do the same processing for the reverse order text and get the embedding.
   (Due to the bidirectional layer that we add into our model, and we will talk about it later).

3.3.1 Basic Concepts in Neutral Network (1)

(I) Perceptron and Sigmoid



1. Perceptron

Linear Model $Z = \sum_{i=0}^{n} W_i \cdot X_i + b$

Set a threshold u: If the output >u, we label it as 1, otherwise, label it with 0

A Simple classification model is complete!

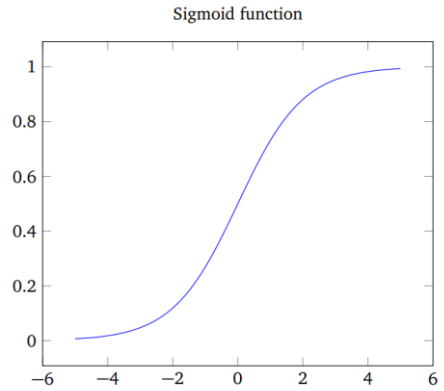
2. Sigmoid

Sigmoid Neutron with activation function

Activation function $a = \sigma(Z) = \frac{1}{1+e^{-Z}}$

E.g. Sigmoid

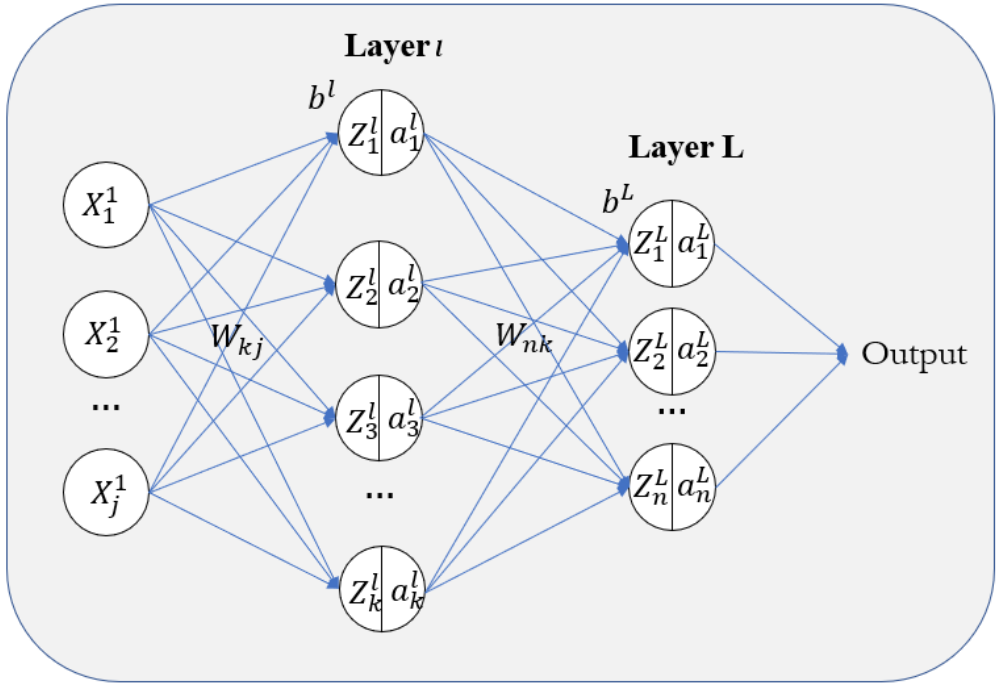
Good Property $\sigma'(Z) = \sigma(Z) \cdot (1 - \sigma(Z))$



$$\Delta \text{output} \approx \sum_j \frac{\partial \text{output}}{\partial w_j} \Delta w_j + \frac{\partial \text{output}}{\partial b} \Delta b$$

3.3.1 Basic Concepts in Neutral Network (2)

(II) Multiple Layer Perceptron



$$C(w,b) \equiv \frac{1}{2n}\sum_x \|y(x)-a\|^2$$

$$Z^l = W^l_{kj} X^1_j + b^l$$

$$a^l = \sigma(Z^l) = \frac{1}{1+e^{-Z^l}}$$

$$Z^L = W^L_{nk} \cdot a^l + b^L \quad \cdots$$

$$a^L = \sigma(Z^L) = \frac{1}{1+e^{-Z^L}}$$

$$Cost = \frac{1}{2}||y - a^L||$$

Feedforward

$$\delta^L = \nabla_a C \odot \sigma'(Z^L)$$

$$\delta^l = ((W^{l+1})^T \delta^{l+1}) \odot \sigma'(Z^l)$$

$$\frac{\partial C}{\partial W^l_{jk}} = a^{l-1}_k \delta^l_j$$

$$\frac{\partial C}{\partial b^l} = \delta^l_j$$

Backward Propagation

$$w_k \to w'_k = w_k - \eta \frac{\partial C}{\partial w_k}$$

$$b_l \to b'_l = b_l - \eta \frac{\partial C}{\partial b_l}.$$

Update the w and b

3.3.2 Recurrent Neutral Network

Why RNN?

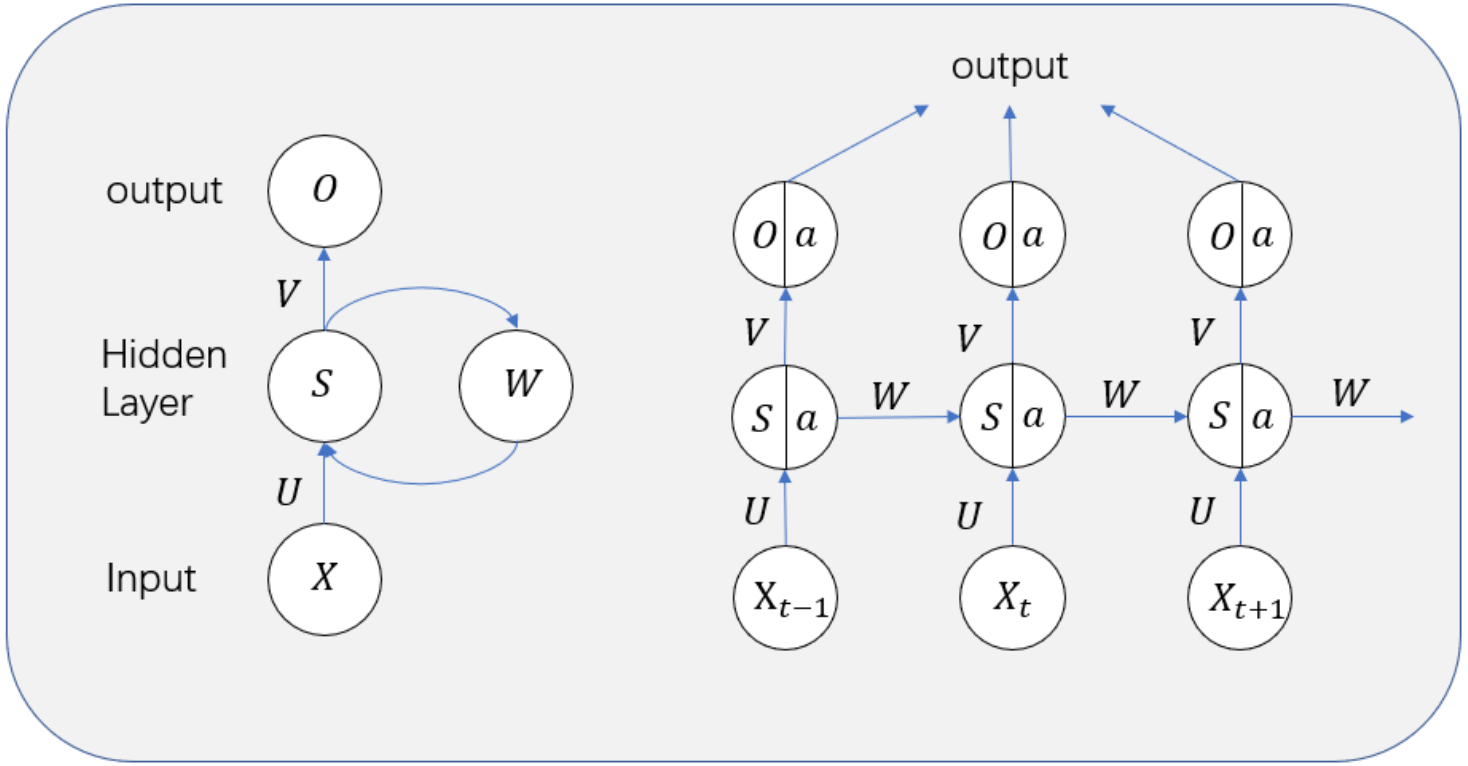
Data with Sequential information

Sequence information in stock price

Stock price of today will affect the next day

Sequence information in Natural language

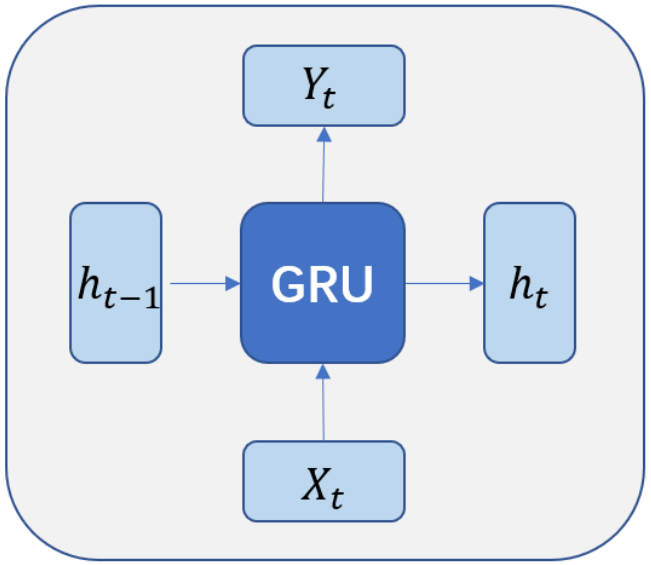
"The patients will take one dose of drag"

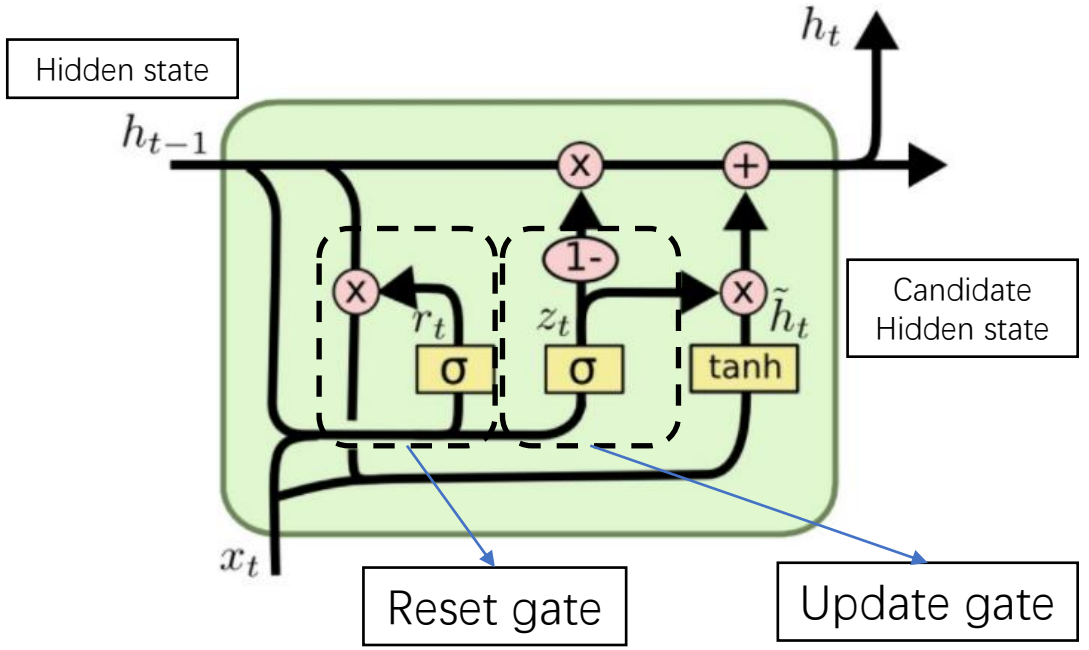


Flatten version of RNN

3.3.3 Gate Recurrent Unit

The purpose of the GRU is to lower down the computation cost but keep a relatively good result.





$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

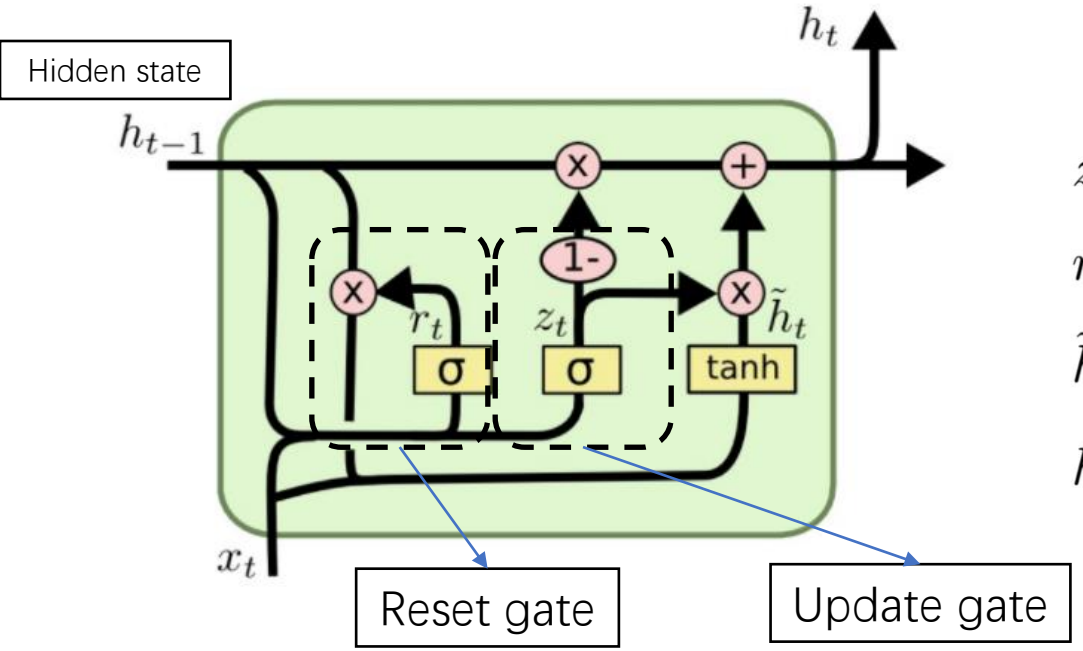
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Reset gate would allow us to control how much of the previous state we might still want to remember, it helps capture short-term dependencies in sequences.

Likewise, an update gate would allow us to control how much of the new state is just a copy of the old state., it helps capture long-term dependencies in sequences

3.3.3 Gate Recurrent Unit



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

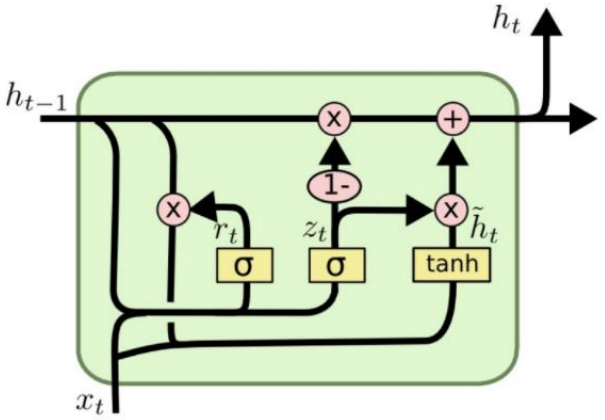
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Fix the vanishing gradient problem

In the traditional RNN model, there may exist exploding gradient problem or vanishing gradient problem if there are too many hidden states in the structure. Gradient vanishing means the gradient will approximately equal to zero, which means the model will no longer be in a learning status. In GRU, since the gates structure can help the model to have a memory of the previous hidden state, the sequence dependency problems that appeared in RNN will be fixed.

3.3.3 Gate Recurrent Unit



Forward Propagation

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$$

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$

$$\tilde{h}_t = \tanh(W_{\tilde{h}} \cdot [r_t * h_{t-1}, x_t])$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

$$y_t = \sigma(W_o \cdot h_t)$$

$$W_r = W_{rx} + W_{rh}$$

$$W_z = W_{zx} + W_{zh}$$

$$W_{\tilde{h}} = W_{\tilde{h}x} + W_{\tilde{h}h}$$

$$y_t^i = W_o h$$

$$y_t^o = \sigma(y_t^i).$$

Cost Function

$$E_t = \frac{1}{2}(y_d - y_t^o)^2$$

$$E = \sum_{t=1}^{T} E_t$$

Backward Propagation Through Time

$$\delta_{y,t} = (y_d - y_t^o) \cdot \sigma'$$

$$\delta_{h,t} = \delta_{y,t} W_o + \delta_{z,t+1} W_{zh} + \delta_{t+1} W_{\tilde{h}h} \cdot r_{t+1} + \delta_{h,t+1} W_{rh} + \delta_{h,t+1} \cdot (1 - z_{t+1})$$

$$\delta_{z,t} = \delta_{t,h} \cdot (\tilde{h}_t - h_{t-1}) \cdot \sigma'$$

$$\delta_t = \delta_{h,t} \cdot z_t \cdot \phi'$$

$$\delta_{r,t} = h_{t-1} \cdot [(\delta_{h,t} \cdot z_t \cdot \phi') W_{\tilde{h}h}] \cdot \sigma'$$

$$\frac{\partial E}{\partial W_o} = \delta_{y,t} h_t$$

$$\frac{\partial E}{\partial W_{zx}} = \delta_{z,t} x_t$$

$$\frac{\partial E}{\partial W_{zh}} = \delta_{z,t} h_{t-1}$$

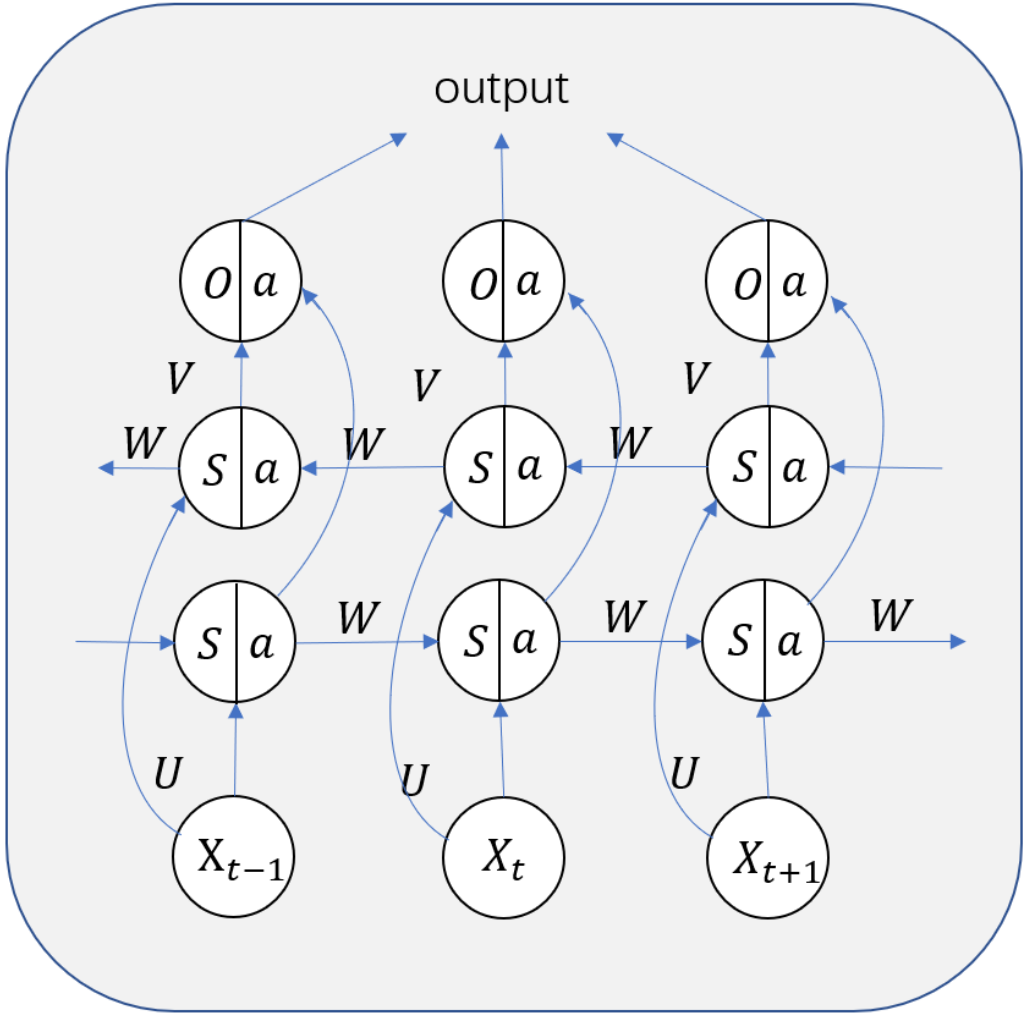
$$\frac{\partial E}{\partial W_{\tilde{h}x}} = \delta_t x_t$$

$$\frac{\partial E}{\partial W_{\tilde{h}h}} = \delta_t (r_t \cdot h_{t-1})$$

$$\frac{\partial E}{\partial W_{rx}} = \delta_{r,t} x_t$$

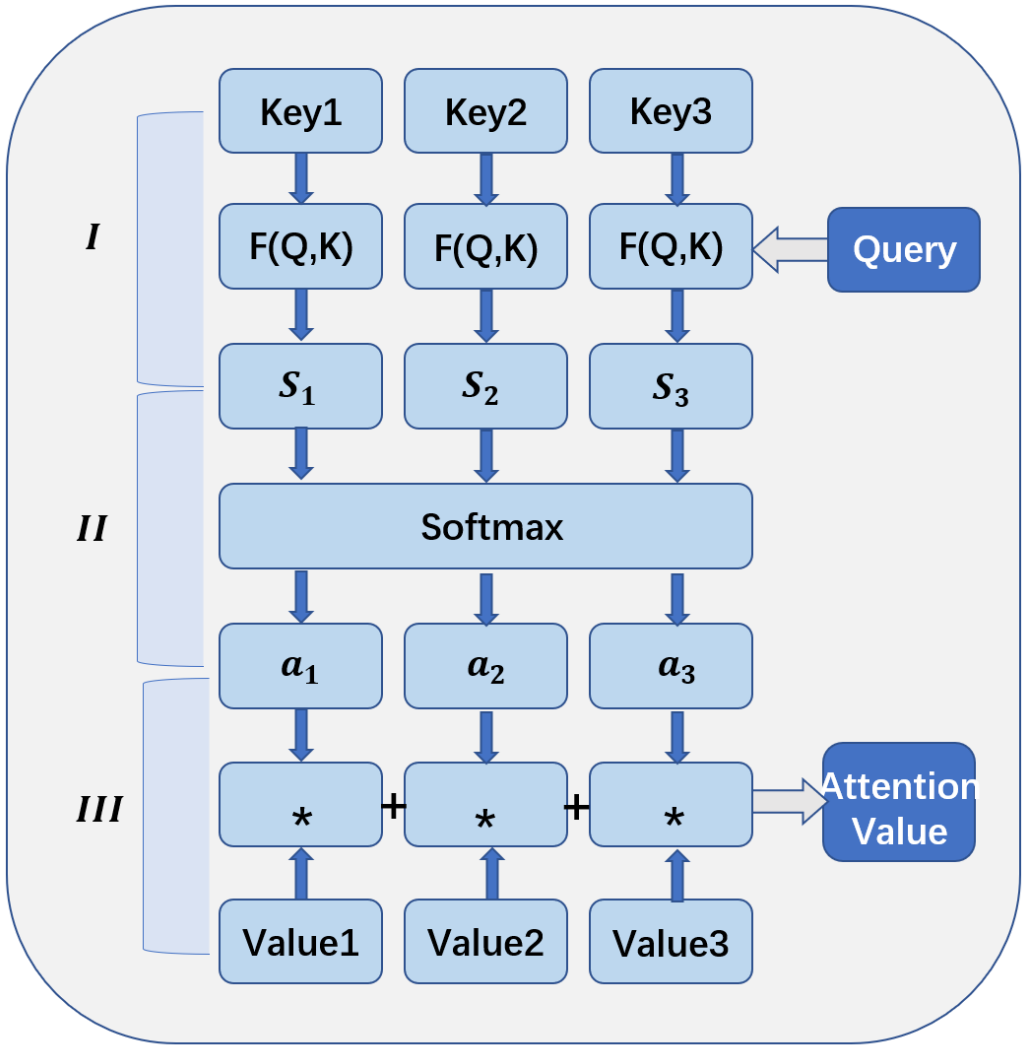
$$\frac{\partial E}{\partial W_{rh}} = \delta_{r,t} h_{t-1}$$

3.3.4 Bidirectional Layer



Apply the original model twice in different directions and combine them together. Traditionally, an RNN model will be applied to follow the exact sequence of the text, while a bidirectional layer adds a path from the end to the beginning of the text so that we can dive deeper into the text by providing a whole picture for the context.

3.3.5 Attention Context



A General Type of Attention Model

Goals: Focus more on valuable information and ignore some useless parts by giving different weights to the data.

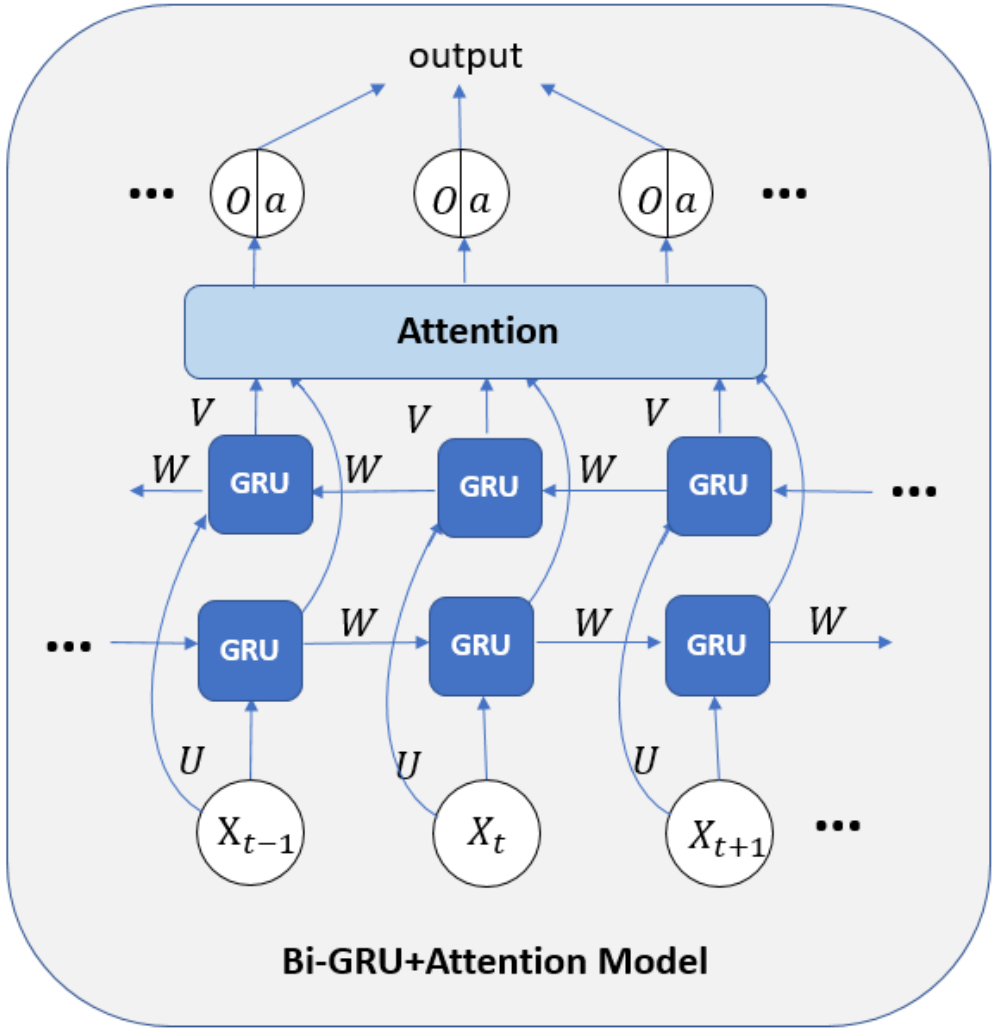
General Steps:

1. Calculate the similarity of query and keys.
2. Normalized the similarity output by using the SoftMax function
3. Calculate the summation of the multiplication of values and the similarity after softmax to get the attention value.

Pros: Give weight to different hidden states, which is very useful for focusing on the more important part of the text and will increase our model performance.

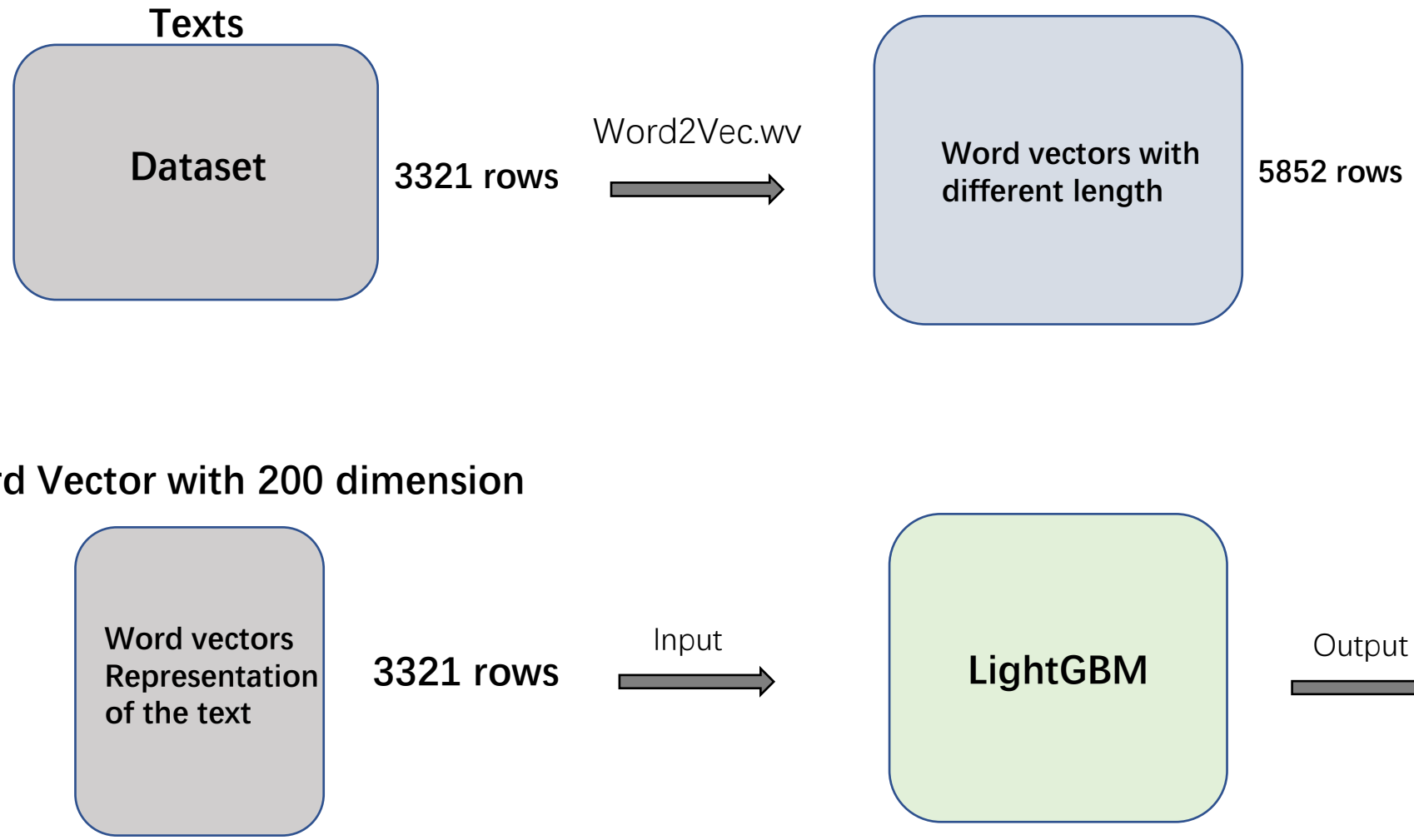
$$Attention(Query, Source) = \sum_{i=1}^{L}{}_{x} Similarity(Query, Key_i) \cdot Values_i$$

3.3.6 Bidirectional GRU with Attention



Part 4 –
Experiments and Results

4.1 Model 1 - Word2Vec+Average feature + LightGBM



4.1 Model 1 - Word2Vec+Average feature + LightGBM

Model Parameters - Word2Vec			
Model	CBOW	Model	Skip-Gram
min_count	1	min_count	1
vector size	200	vector size	200
window	5	window	5
		sg	1
Model Parameters - Average Features			
sentence	train	num_features	200
model	word2vec.wv		

Model Parameters - LightGBM			
boosting_type	gbdt	feature_fraction	0.9
objective	multiclass	bagging_fraction	0.8
num_class	9	bagging_freq	5
metric	multi_error	lambda_l1	0.4
num_leaves	500	lambda_l2	0.5
min_data_in_leaf	100	min_gain_to_split	0.2
learning_rate	0.1	verbose	-1

Training
Model1: Train data length: 2324
Model1: Test data length: 997
Training until validation scores don't improve for 300 rounds
[50] valid_0's multi_error: 0.437312
[100] valid_0's multi_error: 0.418255
[150] valid_0's multi_error: 0.417252
[200] valid_0's multi_error: 0.418255
[250] valid_0's multi_error: 0.412237
[300] valid_0's multi_error: 0.417252
[350] valid_0's multi_error: 0.415246
[400] valid_0's multi_error: 0.416249
[450] valid_0's multi_error: 0.406219
[500] valid_0's multi_error: 0.406219
[550] valid_0's multi_error: 0.402207
[600] valid_0's multi_error: 0.408225
[650] valid_0's multi_error: 0.402207
[700] valid_0's multi_error: 0.410231
[750] valid_0's multi_error: 0.408225
Early stopping, best iteration is:
[491] valid_0's multi_error: 0.398195
all tasks done. total time used:3.629003 s.
auc 0.4502928863833586

4.1 Model 1 - Word2Vec+Average feature + LightGBM

Model Evaluation				
	precision	recall	f1-score	support
0	0.53	0.58	0.55	170
1	0.55	0.40	0.47	146
2	0.44	0.29	0.35	28
3	0.65	0.64	0.64	208
4	0.43	0.36	0.39	72
5	0.82	0.67	0.74	73
6	0.62	0.77	0.69	287
7	0.00	0.00	0.00	3
8	1.00	0.50	0.67	10
accuracy			0.60	997
macro avg	0.56	0.47	0.50	997
weighted avg	0.60	0.60	0.59	997

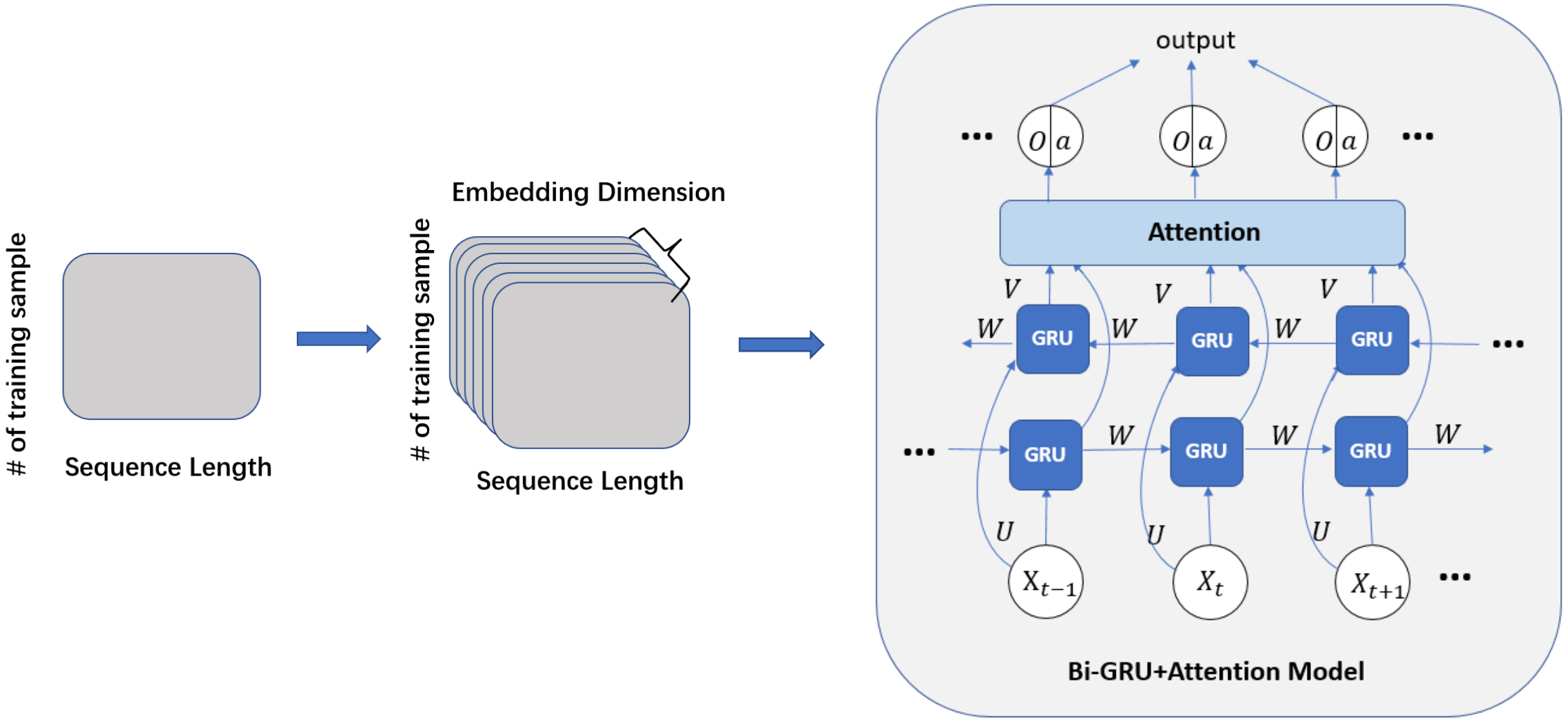
4.1 Model 1 - Results

Model1 - Random 5 Results out of 986 Results

ID	class1	class2	class3	class4	class5	class6	class7	class8	class9
559	0.596669	0.072487	0.000433	0.027216	0.001185	0.016108	0.278105	0.004147	0.003651
560	0.053238	0.002687	0.041341	0.003291	0.815927	0.082539	0.000829	5.35E-05	9.45E-05
561	0.430737	0.00123	0.001359	0.016429	0.539503	0.008062	0.00171	0.000235	0.000736
562	0.000517	7.87E-05	3.95E-05	0.994216	0.000372	0.000246	0.004159	0.000169	0.000202
563	0.022975	0.211937	0.000588	0.107404	0.052235	0.001869	0.598732	0.002087	0.002173
...									
986									

ID	class1	class2	class3	class4	class5	class6	class7	class8	class9
559	1	0	0	0	0	0	0	0	0
560	0	0	0	0	1	0	0	0	0
561	0	0	0	0	1	0	0	0	0
562	0	0	0	1	0	0	0	0	0
563	0	0	0	0	0	0	1	0	0
...									
986									

4.2 Model 2 - Word Embedding + Bi-GRU + Attention



4.2 Model 2 - Word Embedding + Bi-GRU + Attention

Model Paraneters - Bi-GRU with Attention Model			
NUM_CLASS	9	Epoch	10
VOCABULARY_SIZE	10000	Batch size	32
SEQUENCE_LENGTH	1000/2000	Val_Epoch	3
Embedding_dim	200	Val_Batch size	32
lstm_out	64	recurrent_dropout	0.2
dense	32	dropout	0.2

Bidirectional GRU with Attention Model			
Layer (type)	**Output Shape**	**Param #**	**Connected to**
input_1 (InputLayer)	[(None,1000)]	0	[]
input_2 (InputLayer)	[(None,1000)]	0	[]
embedding (Embedding)	(None, 1000,200)	2000000	['input_1 [0][0]']
embedding_1 (Embedding)	(None, 1000,200)	2000000	['input_2 [0][0]']
input_3 (InputLayer)	[(None,1522)]	0	[]
input_4 (InputLayer)	[(None,347)]	0	[]
bidirectional (Bidirectional)	(None, 1000, 128)	102144	['embedding [0][0]']
bidirectional_1 (Bidirectional)	(None, 1000, 128)	102144	['embedding_1 [0][0]']
concatenate (Concatenate)	(None,1869)	0	['input_3[0][0]', 'input_4[0][0]']
attention_with_context (AttentionWithContext)	(None,128)	16640	['bidirectional [0][0]']
attention_with_context_1 (AttentionWithContext)	(None,128)	16640	['bidirectional_1 [0][0]']
dense (Dense)	(None,32)	59840	['concatenate [0][0]']
concatenate_1 (Concatenate)	(None,288)	0	['attention_with_ context[0][0]', 'attention_with_ context_1[0][0]', 'dense[0][0]']
dense_1 (Dense)	(None,9)	2601	['concatenate_1[0][0]']
Total params: 4,300,009			
Trainable params: 4,300,009			
Non-trainable params: 0			

4.2 Model 2 - Word Embedding + Bi-GRU + Attention

Training for Bi-GRU with Attention Model
Epoch 1/10
- loss: 1.6689 - accuracy: 0.3797
Epoch 00001: val_loss improved from inf to 0.08144, saving model to keras_model
-loss: 1.6689 - accuracy: 0.3797 - val_loss: 0.0814 - val_accuracy: 0.0625
Epoch 2/10
- loss: 1.1092 - accuracy: 0.5929
Epoch 00002: val_loss improved from 0.08144 to 0.06775
- loss: 1.1092 - accuracy: 0.5929 - val_loss: 0.0678 - val_accuracy: 0.1637
Epoch 3/10
- loss: 0.8239 - accuracy: 0.7025
Epoch 00003: val_loss improved from 0.06775 to 0.06573
- loss: 0.8239 - accuracy: 0.7025 - val_loss: 0.0657 - val_accuracy: 0.3670
Epoch 4/10
- loss: 0.6516 - accuracy: 0.7712
Epoch 00004: val_loss improved from 0.06573 to 0.06354
- loss: 0.6516 - accuracy: 0.7712 - val_loss: 0.0635 - val_accuracy: 0.2945
Epoch 5/10
- loss: 0.5466 - accuracy: 0.7983
Epoch 00005: val_loss did not improve from 0.06354
- loss: 0.5466 - accuracy: 0.7983 - val_loss: 0.0696 - val_accuracy: 0.2096
Epoch 6/10
- loss: 0.4893 - accuracy: 0.8118
Epoch 00006: val_loss did not improve from 0.06354
- loss: 0.4893 - accuracy: 0.8118 - val_loss: 0.0704 - val_accuracy: 0.1978
Epoch 7/10
- loss: 0.4372 - accuracy: 0.8223
Epoch 00007: val_loss did not improve from 0.06354
- loss: 0.4372 - accuracy: 0.8223 - val_loss: 0.0731 - val_accuracy: 0.2971
Epoch 8/10
- ETA: 0s - loss: 0.4073 - accuracy: 0.8383
Epoch 00008: val_loss did not improve from 0.06354
- loss: 0.4073 - accuracy: 0.8383 - val_loss: 0.0720 - val_accuracy: 0.1971
Epoch 9/10
- ETA: 0s - loss: 0.3917 - accuracy: 0.8398
Epoch 00009: val_loss did not improve from 0.06354
- loss: 0.3917 - accuracy: 0.8398 - val_loss: 0.0759 - val_accuracy: 0.2041
Epoch 10/10
- ETA: 0s - loss: 0.3667 - accuracy: 0.8452
Epoch 00010: val_loss did not improve from 0.06354
- loss: 0.3667 - accuracy: 0.8452 - val_loss: 0.0765 - val_accuracy: 0.2101

4.2 Model 2 - Results

Model2-1 Random 5Results out of 986 Results Sequence length 1000

ID	class1	class2	class3	class4	class5	class6	class7	class8	class9
559	0.170628	0.147612	0.053016	0.166229	0.09173	0.083567	0.243275	0.016068	0.027874
560	0.126328	0.045024	0.035029	0.05354	0.639337	0.06412	0.021932	0.008378	0.006313
561	0.210251	0.124376	0.042793	0.159376	0.110343	0.110612	0.215928	0.008554	0.017766
562	0.03284	0.018627	0.032981	0.816597	0.027288	0.037492	0.020632	0.004874	0.008668
563	0.166839	0.14777	0.053195	0.16634	0.094276	0.08245	0.246316	0.016318	0.026495

...

ID	class1	class2	class3	class4	class5	class6	class7	class8	class9
559	0	0	0	0	0	0	1	0	0
560	0	0	0	0	1	0	0	0	0
561	0	0	0	0	0	0	1	0	0
562	0	0	0	1	0	0	0	0	0
563	0	0	0	0	0	0	1	0	0

...

986

4.2 Model 2 - Results

Model2-2 Random 5 Results out of 986 Results Sequence length 2000

ID	class1	class2	class3	class4	class5	class6	class7	class8	class9
559	0.141351	0.149183	0.06552	0.175357	0.074423	0.110917	0.225321	0.028615	0.029313
560	0.056723	0.01398	0.121212	0.027628	0.316461	0.045233	0.313854	0.039469	0.065441
561	0.137214	0.150981	0.06545	0.176246	0.073253	0.106498	0.235477	0.028375	0.026506
562	0.140977	0.152256	0.064073	0.17613	0.074437	0.107476	0.228968	0.028351	0.027334
563	0.138861	0.149943	0.065433	0.174805	0.074679	0.10773	0.232379	0.028695	0.027474

...

ID	class1	class2	class3	class4	class5	class6	class7	class8	class9
559	0	0	0	0	0	0	1	0	0
560	0	0	0	0	1	0	0	0	0
561	0	0	0	0	0	0	1	0	0
562	0	0	0	0	0	0	1	0	0
563	0	0	0	0	0	0	1	0	0

...

986

Model Comparision

FINAL REULTS FOR THIS PROJECT

Submission and Description	Private Score	Public Score	Private Rank
Final_Submission_lightGBM.csv a few seconds ago by KaiyangL Final Submission LightGBM	3.59873	1.46629	302/1386

Submission and Description	Private Score	Public Score	
Final_Submission_BiGRUAttention - 1000.csv just now by KaiyangL Final Submission Bi-GRU+Attention - Seq len 1000	2.52648	1.76996	91/1386

The Best Model

Submission and Description	Private Score	Public Score	Private Rank
Final_Submission_BiGRUAttention - 2000.csv just now by KaiyangL Final Submission Bi-GRU+Attention - Seq len 2000	2.36962	1.88202	88/1386

Part 5 – Conclusion

Strength

For the LightGBM model, we use the word vector from the Word2Vec model and do the feature average as the input, we successfully predict the class label. The strength of this model is that only a few seconds are needed when training the LightGBM model and at the same time get a result with relatively high accuracy.

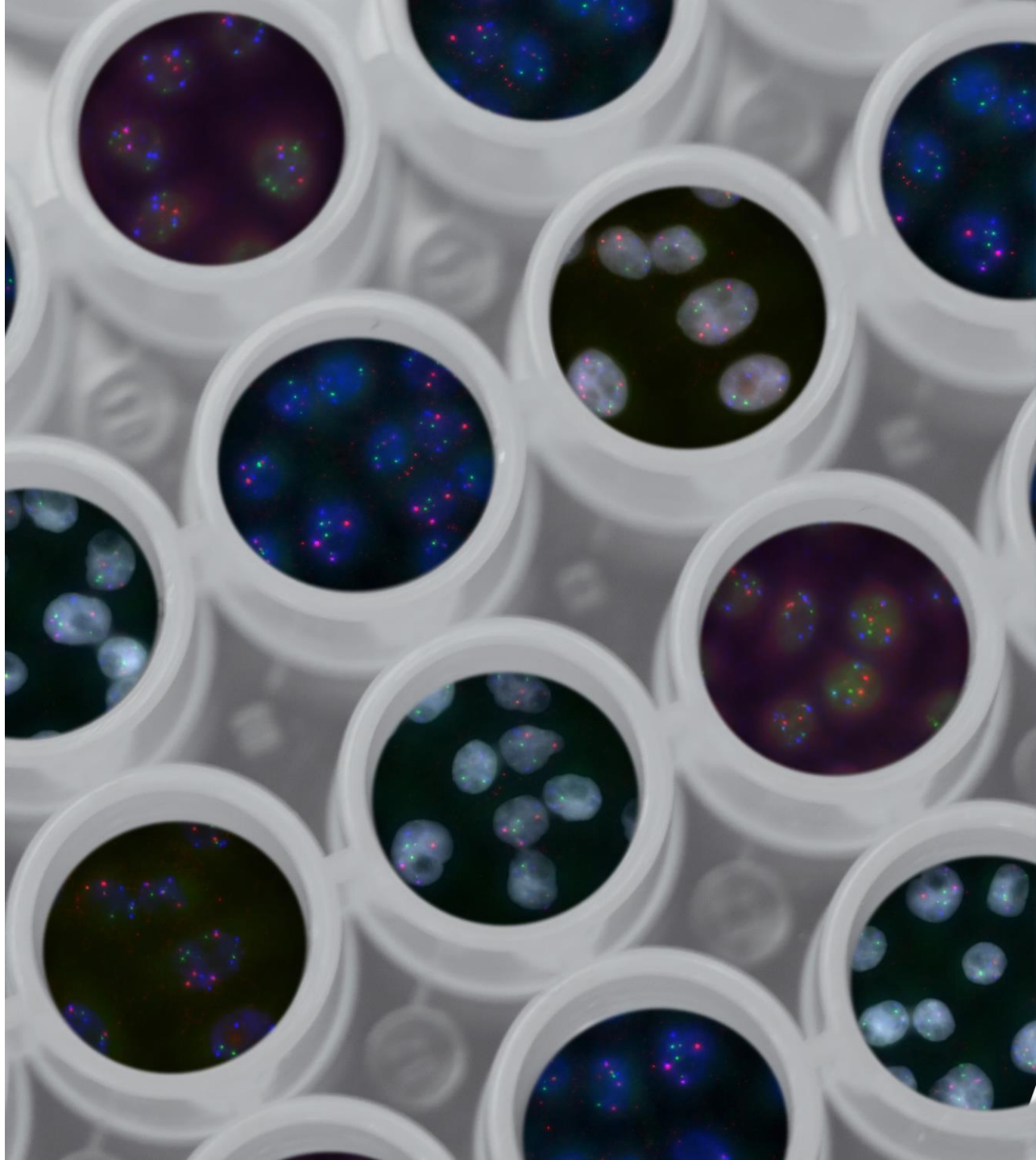
For the Bidirectional GRU model with an attention layer, we use the three-dimensional tensor as the input. The gate structure from GRU will make the balance of the memory and new information mode easier and use fewer computation resources. Also, the bidirectional layer enables us to get the full picture of the sequence from two directions of the texts. Moreover, the adding of attention mechanism greatly improves the performance of our model by paying more attention to the useful information in the texts.

Limitation

The LightGBM model has the highest public score among the three submissions but gets the lowest score in the private score.

To explain it further, the test data sets for public and private scores are different. The result in LightGBM shows that it can learn the text in a public dataset very well due to the potential over-fitting problems, and it cannot handle the text in a private dataset since the lack of generalization ability.

For the Bi-GRU with attentions model, we make full use of the validation set and set the appropriate drop-out parameter to avoid over-fitting. However, for each run of this model, we will need a few hours to get the results. Therefore, those complex models will be greatly restrained by the computation limitation of the computer.



Nowadays, with the great help of machine learning and deep learning model in the classification of clinical documents, especially in dealing with the text that is related to the gene mutation, lots of human efforts will be avoided. In the future, some high-performance model with a lower computational cost is desirable to develop in dealing with the classification of the clinical text so that patients may receive a personalized treatment that benefits from NLP techniques and classification models.

Bibliography

[1] J. Ferlay, M. Ervik, F. Lam, M. Colombet, L. Mery, and M. Piñeros, "Global Cancer Observatory: Cancer Today," International Agency for Research on Cancer, Tech. Rep., 2021. [Online]. Available: https://gco.iarc.fr/today

[2] J. J. Salk, E. J. Fox, and L. A. Loeb, "Mutational Heterogeneity in Human Cancers: Origin and Consequences," *Annual Review of Pathology: Mechanisms of Disease*, vol. 5, no. 1, pp. 51–75, 2010. [Online]. Available: http://www.annualreviews.org/doi/10.1146/annurev-pathol-121808-102113

[3] J. Xu, P. Yang, S. Xue, B. Sharma, M. Sanchez-Martin, F. Wang, K. A. Beaty, E. Dehan, and B. Parikh, "Translating cancer genomics into precision medicine with artificial intelligence: applications, challenges and future perspectives," *Human Genetics*, vol. 138, no. 2, pp. 109–124, 2019. [Online]. Available: http://link.springer.com/10.1007/s00439-019-01970-5

[4] R. Leaman, R. Khare, and Z. Lu, "Challenges in clinical natural language processing for automated disorder normalization," *Journal of Biomedical Informatics*, vol. 57, pp. 28–37, 2015. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1532046415001501

[5] M. Gupta, H. Wu, S. Arora, A. Gupta, G. Chaudhary, and Q. Hua, "Gene Mutation Classification through Text Evidence Facilitating Cancer Tumour Detection," *Journal of Healthcare Engineering*, vol. 2021, pp. 1–16, 2021. [Online]. Available: https://www.hindawi.com/journals/jhe/2021/8689873/

[6] M. D. Yandell and W. H. Majoros, "Genomics and natural language processing," *Nature Reviews Genetics*, vol. 3, no. 8, pp. 601–610, 2002. [Online]. Available: http://www.nature.com/articles/nrg861

[7] S. Velupillai, H. Suominen, M. Liakata, A. Roberts, A. D. Shah, K. Morley, D. Osborn, J. Hayes, R. Stewart, J. Downs, W. Chapman, and R. Dutta, "Using clinical Natural Language Processing for health outcomes research: Overview and actionable suggestions for future advances," *Journal of Biomedical Informatics*, vol. 88, pp. 11–19, 2018. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1532046418302016

[8] H. Farooq, N. Rehmat, S. Kumar, and H. Naveed, "Genetic Mutation Classification using Machine Learning," *Biophysical Journal*, vol. 116, no. 3, p. 292a, 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0006349518328418

[9] M. Singh, A. Murthy, and S. Singh, "Prioritization of Free-Text Clinical Documents: A Novel Use of a Bayesian Classifier," *JMIR Medical Informatics*, vol. 3, no. 2, p. e17, 2015. [Online]. Available: http://medinform.jmir.org/2015/2/e17/

[10] A. Wright, A. B. McCoy, S. Henkin, A. Kale, and D. F. Sittig, "Use of a support vector machine for categorizing free-text notes: assessment of accuracy across two institutions," *Journal of the American Medical Informatics Association*, vol. 20, no. 5, pp. 887–890, 2013. [Online]. Available: https://academic.oup.com/jamia/article-lookup/doi/10.1136/amiajnl-2012-001576

[11] T. K. Ho, "random decision forests," *Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1)*, vol. Volume 1, pp. 278–282, 1995. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/ICDAR.1995.598994

Bibliography

[12] J. H. Friedman, "Greedy function approximation: A gradient boosting machine." *The Annals of Statistics*, vol. 29, no. 5, Oct. 2001. [Online]. Available: https://projecteuclid.org/journals/annals-of-statistics/volume-29/issue-5/Greedy-function-approximation-A-gradient-boosting-machine/10.1214/aos/1013203451.full

[13] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Francisco California USA: ACM, Aug. 2016, pp. 785–794. [Online]. Available: https://dl.acm.org/doi/10.1145/2939672.2939785

[14] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," p. 9.

[15] X. Qin, J. Liu, Y. Wang, Y. Liu, K. Deng, Y. Ma, K. Zou, L. Li, and X. Sun, "Natural language processing was effective in assisting rapid title and abstract screening when updating systematic reviews," *Journal of Clinical Epidemiology*, vol. 133, pp. 121–129, 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0895435621000147

[16] I. Spasic and G. Nenadic, "Clinical Text Data in Machine Learning: Systematic Review," *JMIR MEDICAL INFORMATICS*, p. 19.

[17] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. [Online]. Available: http://dx.doi.org/10.1162/neco.1997.9.8.1735

[18] V. Mnih, N. Heess, and A. Graves, "Recurrent Models of Visual Attention," p. 9.

[19] Y. Bao, Z. Deng, Y. Wang, H. Kim, V. D. Armengol, F. Acevedo, N. Ouardaoui, C. Wang, G. Parmigiani, R. Barzilay, D. Braun, and K. S. Hughes, "Using Machine Learning and Natural Language Processing to Review and Classify the Medical Literature on Cancer Susceptibility Genes," *JCO Clinical Cancer Informatics*, no. 3, pp. 1–9, 2019. [Online]. Available: https://ascopubs.org/doi/10.1200/CCI.19.00042

[20] Y. Wang, S. Sohn, S. Liu, F. Shen, L. Wang, E. J. Atkinson, S. Amin, and H. Liu, "A clinical text classification paradigm using weak supervision and deep representation," *BMC Medical Informatics and Decision Making*, vol. 19, no. 1, p. 1, 2019. [Online]. Available: https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s018-0723-6

[21] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv:1810.04805 [cs]*, 2019. [Online]. Available: http://arxiv.org/abs/1810.04805

[22] Y. Su, H. Xiang, H. Xie, Y. Yu, S. Dong, Z. Yang, and N. Zhao, "Application of BERT to Enable Gene Classification Based on Clinical Evidence," *BioMed Research International*, vol. 2020, pp. 1–13, 2020. [Online]. Available: https://www.hindawi.com/journals/bmri/2020/5491963/

[23] S. Wu, K. Roberts, S. Datta, J. Du, Z. Ji, Y. Si, S. Soni, Q. Wang, Q. Wei, Y. Xiang, B. Zhao, and H. Xu, "Deep learning in clinical natural language processing: a methodical review," *Journal of the American Medical Informatics Association*, vol. 27, no. 3, pp. 457–470, 2020. [Online]. Available: https://academic.oup.com/jamia/article/27/3/457/5651084

Acknowledgement

This project would not have been possible without the support of many people. Thanks to my advisor Dr. Yuhui Deng for holding so many seminars, providing great support, and giving instructions during my final year project. Thanks to my observer Dr. Xiaoling Peng coming for my FYP oral defense. Also, I will give many thanks to my friends in the same group who offer guidance and supports.

Thanks for your attention

Question and Answer