Question with same meaning ?

Overview

1. Quora is a question-and-answer site where questions are asked, answered, edited, and organized by its community of users. In this platform, everyone could ask others' help for their confuse. Also, they could share their ideas and insights to help others. According to statistical data, over 100 million people visit Quora every month. However, Multiple similar questions means that uses would spend more time to find the best answer and the writer should provide multiple version of answers of similar questions. Therefore, it's a significant but difficult problem for website managers to identify and merge the similar questions.
2. In this project, we would try to solve this natural language processing problem using machine learning.

Data

1. The data includes two files. One is *train.csv*, which includes two questions and the statement whether they are duplicate. The following is the detailed data fields:

- **id –** the id of a training set question pair
- **qid1, qid2 –** unique ids of each question (only available in train.csv)
- **question1, question2 –** the full text of each question
- **is_duplicate –** the target variable, set to 1 if question1 and question2 have essentially the same meaning, and 0 otherwise.

2. The other is *test.csv*, which includes two questions to be tested whether they are duplicate. The following is the detailed data fields:

- **test_id –** the id of a test set question pair

- question1, question2 – the full text of each question

Preparation & Primary Processing

1. Indexing word vectors

   We use the GloVe(Global Vectors for Word Representation) developed by Stanford. This tool transfer the words to the vectors. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space.

2. Removal of duplicates and statistic value of words

   In this part, we first explore the train set about the number of duplicate and non-duplicate question pairs, then remove the duplicated pairs. Finally we count the number of questions which appears more than once.

Dictionary Generation & Embedding layer

1. Dictionary Generation: In this part, we generate a train dictionary. So at first, we tokenize the sentences to get words from the questions in train set. We use the porter stemmer to break down words. For some basic words, we use NLTK stop words to remove them and we would get train dictionary.

2. Preparing embedding layer: We choose the less value from the number of words in the vector dictionary and the number of words in the text data as the number of words. If the word in the text but not in the vector dictionary, then its vector is null value. Otherwise, its vector is the vector in the vector dictionary. Finally we build the embedding layer using the pre-trained word vector.

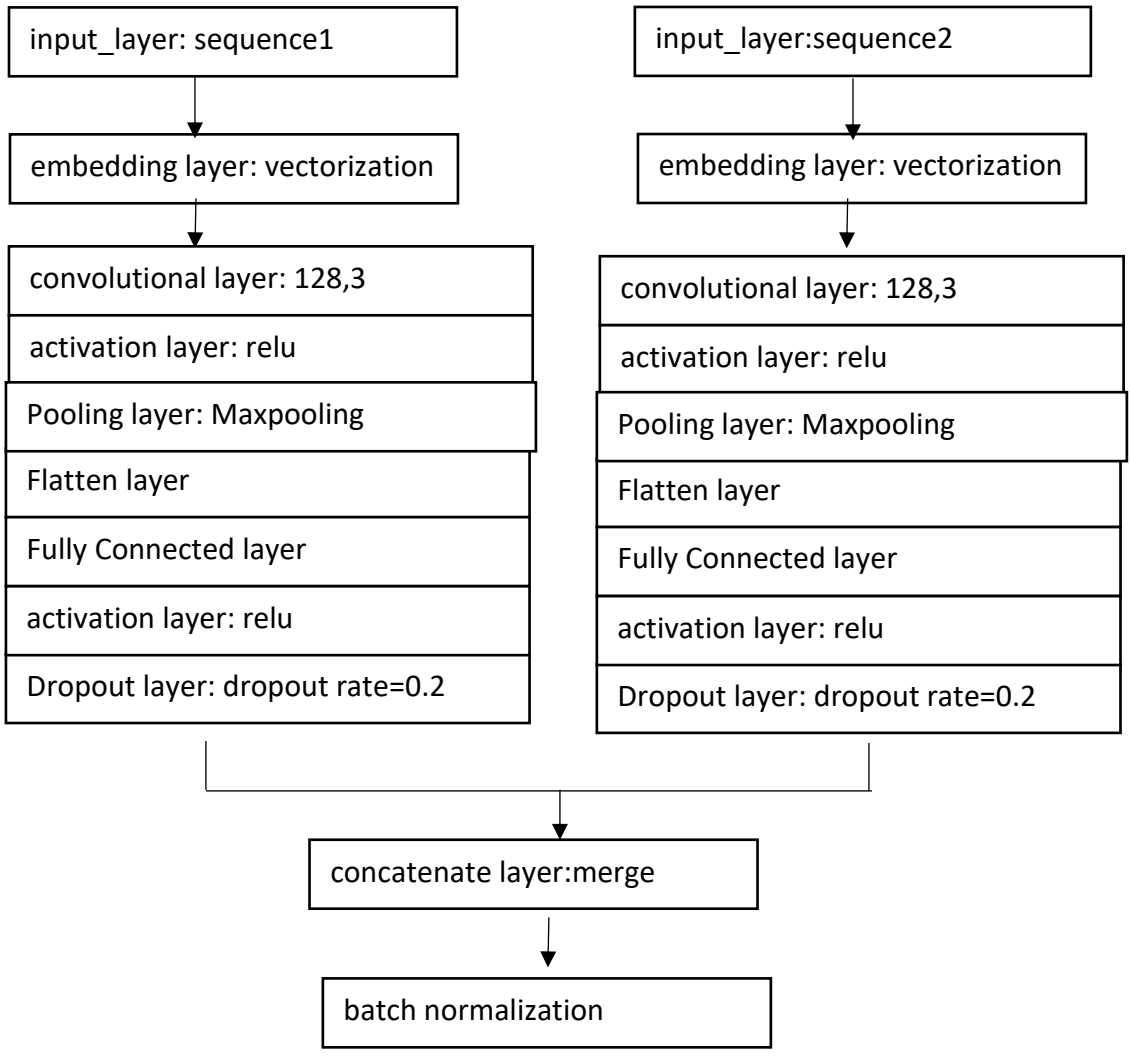
Training the model

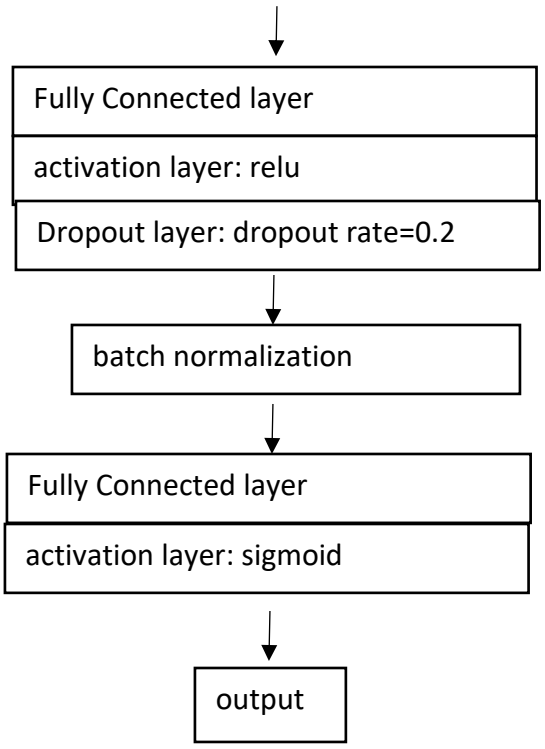
1. In this project, we use three neural network structures to train the model. Before introducing them, we would introduce the layers in the neural network.

- **embedding layers:** embedding layer is the first network layer of the model. It's often used to build the model about the text data.
- **fully connected layers:** In the whole neural network, the fully connected layers play as the classifier, which connect every neuron in one layer to every neuron in another layer and achieve the activation of the neutrons in neural network. It's the same as the traditional multi-layer perceptron neural network.
- **activation layers:** Activation Layer is used to apply activation on input and corresponding derivative on epsilon. There are several activation function and we use rectified linear unit in this project.
- **flatten layers:** Flatten Layer is used to flatten a matrix which has higher dimension to a second dimension matrix. Its method is keeping the size of the first dimension and then flatten all other data into the second dimension.
- **Lambda layers:** Lambda layer can encapsulate any expression into an object of the network layer. The parameter is an expression, and generally it's a function.
- **Convolutional layers:** The convolutional layer is the core building block of a CNN, which apply a convolution operation to the input, passing the result to the next layer.
- **Pooling layers:** Pooling is an operation to the features in CNN, always after the convolutional layers. The purpose of pooling is to calculate the local sufficient statistics of the features to decrease the total number of features. It is effective to prevent overfitting and reduce the amount of calculation. In the project, we use global max pooling and global average pooling.
- **Recurrent layers:** recurrent layer is used to construct the neural network about sequence. In the recurrent network, the flow of information is (current input + previous information of hidden layers) -> current hidden layer -> output.

- **Concatenate layers:** concatenate layers can concatenate some outputs of a network into a single tensor.
- **Dropout layers:** dropout layer apply the dropout strategy to the input vector of the layer. We often use it to prevent overfitting by set a certain ratio of nodes randomly which we would not update. But we would keep their weights. The ration could set randomly between [0,1].
- **Batch normalization:** Batch normalization is used solve the Internal Covariate Shift problem. We would normalize every dimension in every mini batch.

2. In these models, we have three inputs. The first two is about question1 and the question2 directly, and the third input is the distance feature
3. The structure of the first model we use is the following :



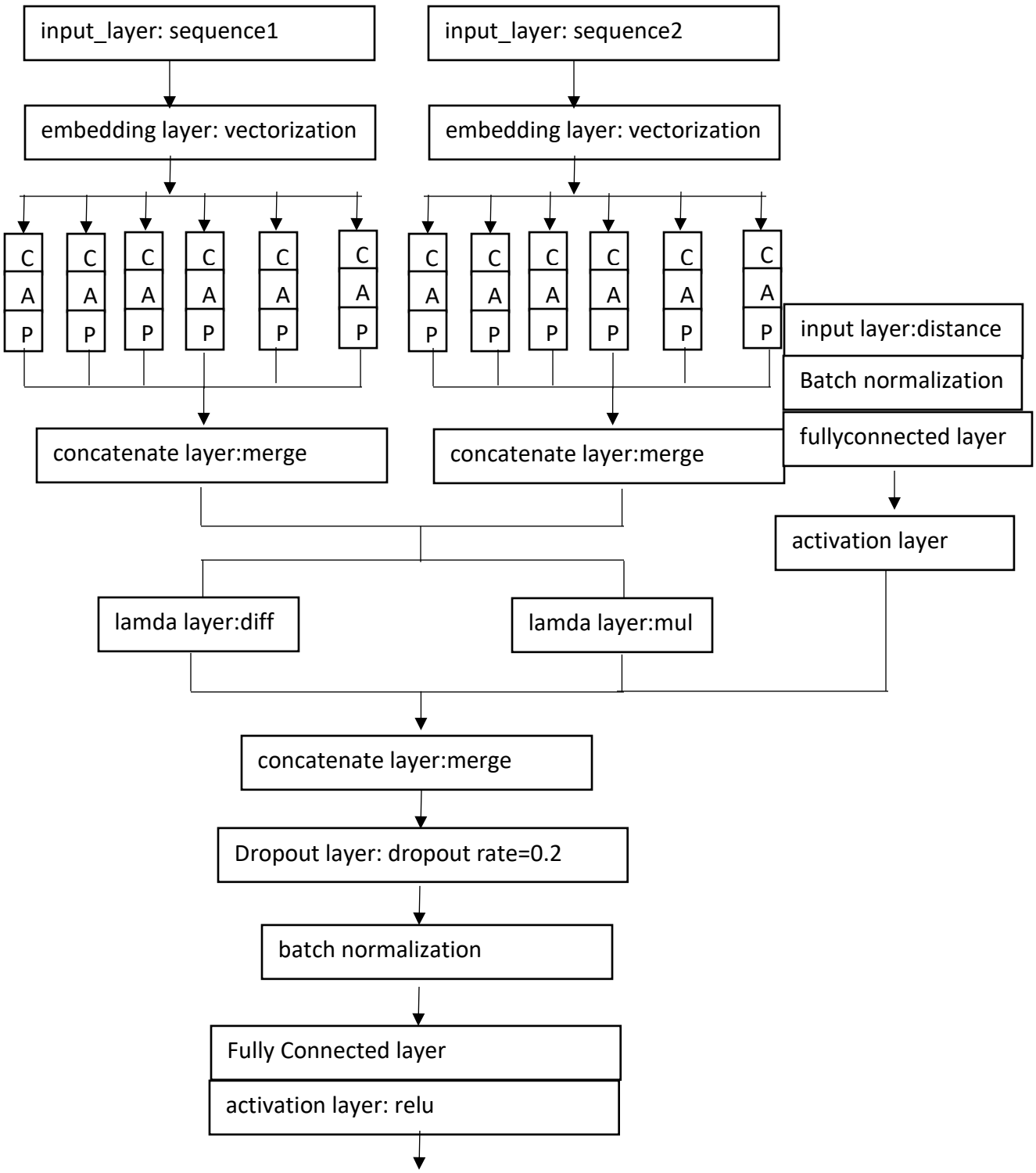


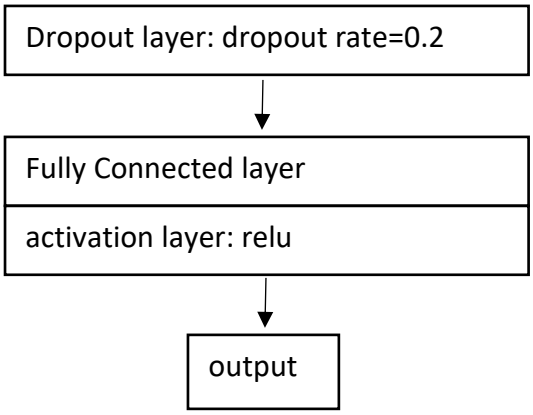
At this model, we just use two features and one convolutional layer, one pooling layer, one flatten layer, one fully connected layer and one dropout layer for each input. Then we concatenate two outputs into one and it would through batch normalization, a fully connected layer and a dropout layer. Finally we do a final batch normalization.

We could see that the accuracy of this model is the following:

```
Train on 319997 samples, validate on 80000 samples
Epoch 1/1
319997/319997 [==============================] - 136s 425us/step - loss: 0.6376 - acc:
0.6168 - val_loss: 0.6072 - val_acc: 0.6484
```

4. The structure of the second model is extended from the first model, which would have more convolutional layers and we add the distance feature into it. Also, we use add lamda layer to encapulate two sequences' features to two function instead of using them directly. The following is the structure:



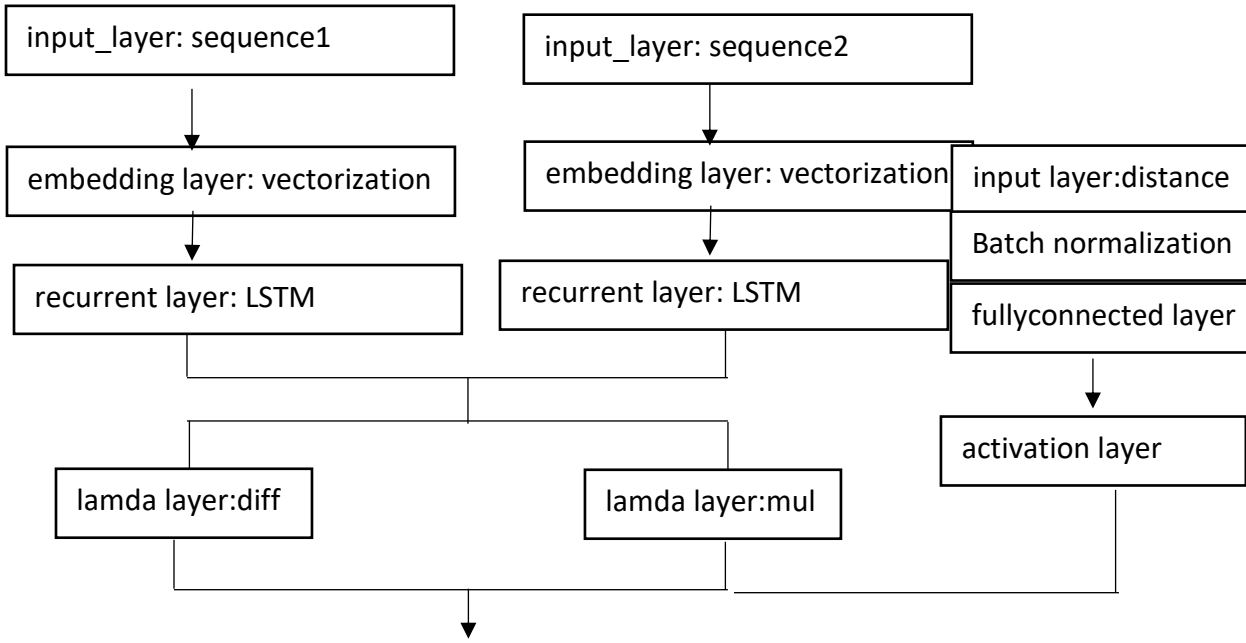


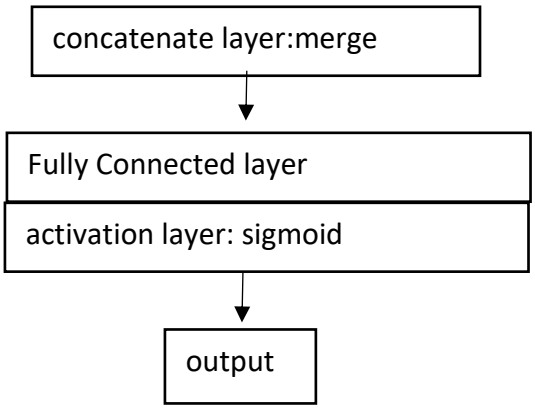
C – convolution layer, A – activation layer, P – pooling layer

We could see that the accuracy of this model is the following, which is much better than the first one:

```
Train on 319997 samples, validate on 80000 samples
Epoch 1/1
319997/319997 [==============================] - 741s 2ms/step - loss: 0.4676 - acc: 0.
7661 - val_loss: 0.4121 - val_acc: 0.8003
```

5. The structure of the third model uses the recurrent layer to achieve the goal. In this model, we would use three inputs. The following is the structure:

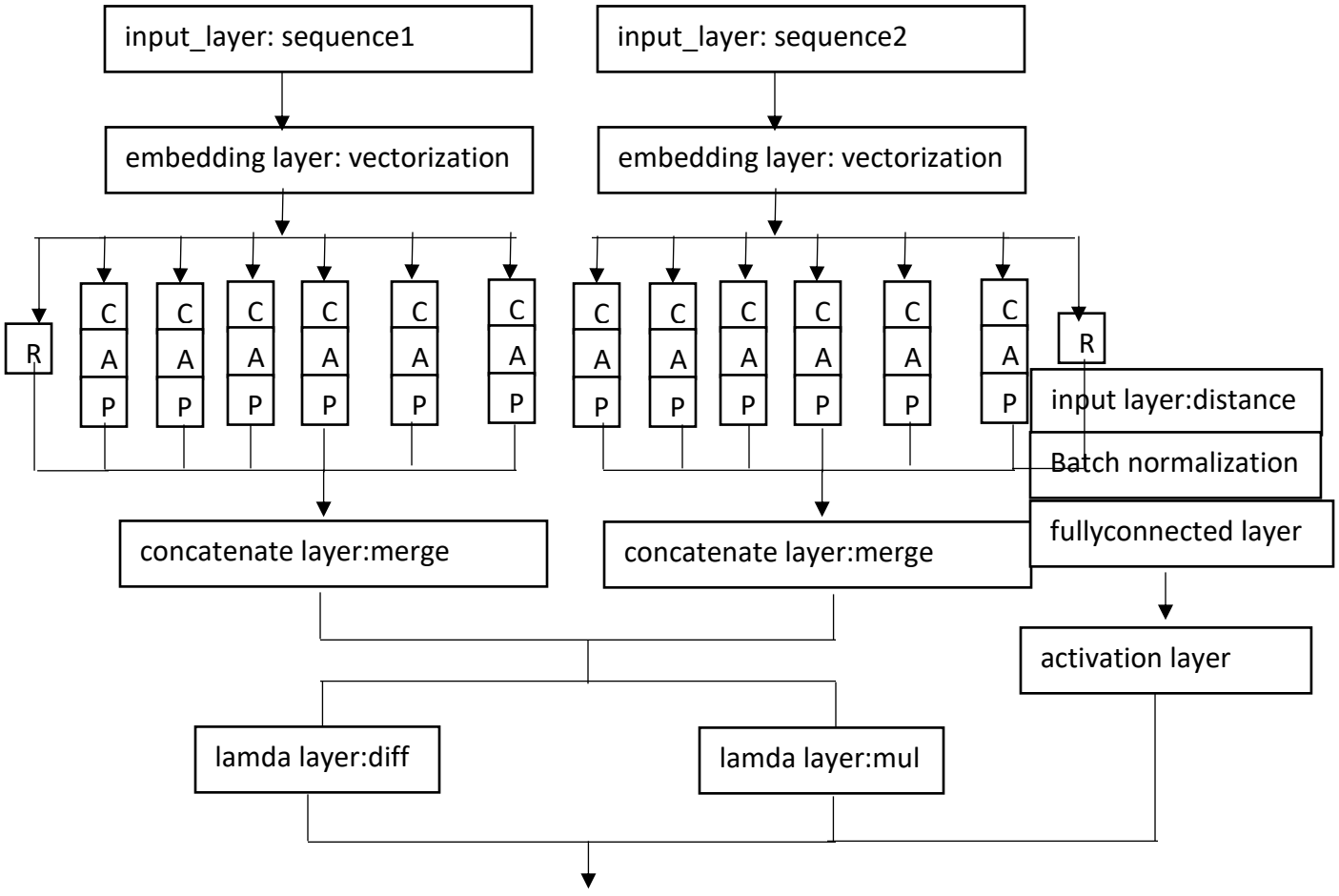


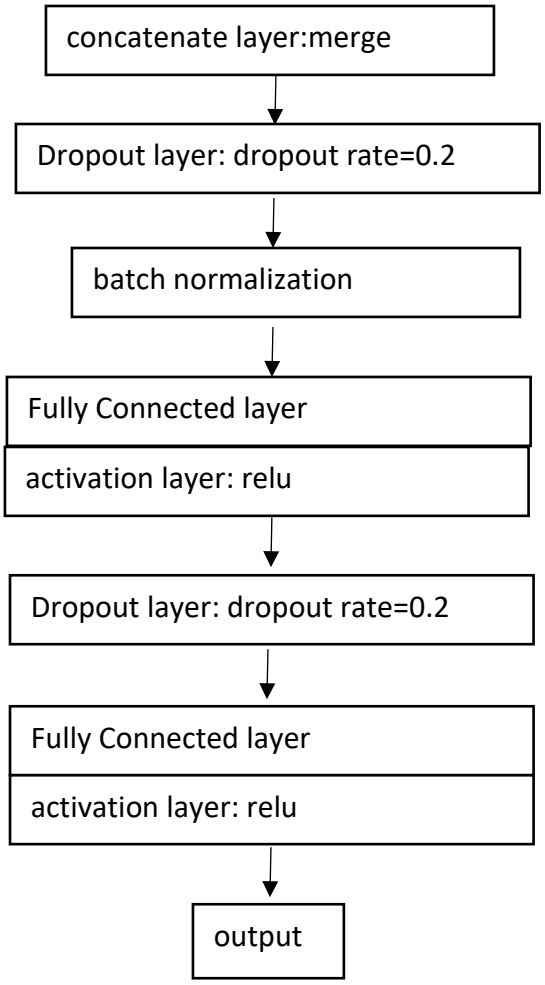


We could see that the accuracy of this model is the following:

```
Train on 319997 samples, validate on 80000 samples
Epoch 1/1
319997/319997 [==============================] - 1320s 4ms/step - loss: 0.4971 - acc: 0
.7375 - val_loss: 0.4530 - val_acc: 0.7711
```

6. The structure of the final model combines the second and the third model, which combines with recurrent layer and convolutional layer, the following is the structure:





C – convolution layer, A – activation layer, P – pooling layer, R – recurrent layer

We could see that it has the highest accuracy:

```
Train on 319997 samples, validate on 80000 samples
Epoch 1/1
319997/319997 [==============================] - 2443s 8ms/step - loss: 0.4558 - acc:
0.7777 - val_loss: 0.3905 - val_acc: 0.8112
```

Summary

In this project, we solve the problem about similar questions in Quora. We tried different ways to train the model and get the best one. For any test set, this model would output the prediction, which improve the efficiency of identifying and concatenating the similar questions.