Advanced Machine Learning T2 Question Paper
Date: 9th Oct 2018

Scope of this exam:

Unit 3: RNN, LSTM, Word Representations, Processing sequences and time series

Objectives of this project

- Develop models that can determine semantic similarity between 2 sequences
- Use distributed word representations and build models using Keras embedding layer
- Using Keras for sequence processing

Problem Statement:

In this project, you are provided with a dataset from Quora that was also launched as a Kaggle competition. Each row in the dataset has a pair of questions: (question1, question2) and you are required to predict an output based on the semantic similarity between the 2 questions. This helps in discovering duplicate questions and handling them suitably. The ground truth is a binary field that is 1 if the 2 inputs are semantically similar (duplicates of each other) and 0 if they are distinct.

You are required to develop a GRU based model that implements the architecture described in the steps below. You are required to train, validate and test your implementation using the dataset provided for this exam.

Detailed Steps:

(a) Obtaining Dataset

1. First run the file: get_api_key.py with Python 3 environment and obtain your API key. The program would ask for user name and password. Your user name is your group number. If your group number is 7, the user name is: g7. The password is pesit.
2. Copy the API key.
3. You are provided with a starter code named: example_kaggle_quora.py. This module will create an instance of API client: client = ApiClient( ). You need to pass your API key as parameter to this: client = ApiClient(auth_key = YOUR_API_KEY)
4. Run the file example_kaggle_quora.py to get an idea about the dataset.
5. The above example file has the necessary comments that serve as the necessary documentation to use the API service.
6. You can obtain data by invoking the step below. num_samples is rate limited to 10000 samples per call.
7. `val = client.get_kaggle_quora_data(num_samples)`
8. You will receive a list of Python dictionariesas the output from this call. An example of a dictionary element is shown below

```
{'id': '4127', 'qid1': '8165', 'qid2': '8166', 'question1': 'Why should I
still vote for Hillary Clinton?', 'question2': "Why shouldn't I vote for
Hillary Clinton?", 'is_duplicate': '0'},
```

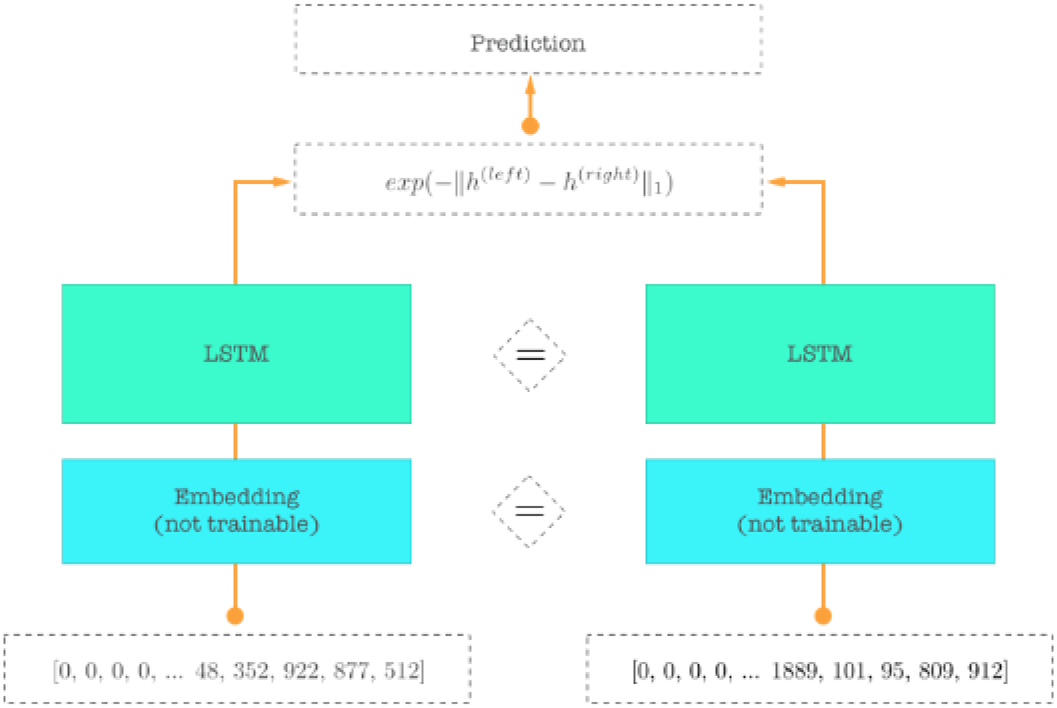
(b) Processing Dataset and building the embedding matrix

9. Obtain 20000 to 50000 training samples by invoking this API as many times as needed. Note that when you want to use more data the training time on your GRU will also be more. So, start with smaller quantities to get your model right and increase it gradually. You can go up to 400K samples for training and keep a part of this for validation.
10. Merge all the lists obtained and obtain all text from question1 and question2 fields. All these sentences together make up your corpus.
11. Build your vocabulary for this corpus. Restrict the size of your vocabulary to say 10000. This is done by making all words to lowercase and also keeping a minimum count of occurrence of words. Eliminate rare words that are below the min count.
12. Obtain the glove vectors for the list of words in the vocabulary using the web service API. Handle rate limits using repeating the calls.
13. Build an embedding matrix using the vectors. Note that the web service will return a zero vector for out of vocabulary words.
14. NOTE: You may need to pad the inputs as they will be of variable size. See pad_sequences of Keras to get an idea and refer to sample code from resources on web to prepare your dataset. See https://blog.keras.io/using-pre-trained-word-embeddings-in-a-keras-model.html for some examples.

(c) Developing the GRU model

15. You should implement an architecture that does the following:
    a. Uses 2 channels: one for processing question1 and other for question2. You may want to implement this by using 2 input layers.
    b. You should pass these through an LSTM and obtain the hidden vectors at the last time step, MAXLEN where MAXLEN is the fixed maximum length of your sequences.
    c. You will now have 2 vectors: one is a representation of question1 and the other is that for question2. Let each of these vectors have a dimension d. You should concatenate these two vectors and pass them through an output stage which should generate a binary output.
    d. As mentioned above, you will need to build a neural network that accepts the concatenated vectors and have a sigmoid output layer.

The diagram below gives the expected architecture. **You can use a single GRU to model both the LSTMs shown in the fig**. **Replace the exp(.) sub system and the prediction sub system in the fig below with your neural network layers that provide a sigmoid output as the final output.**



16. Use 85% of this data for training and remaining for validation
17. Train the model with the dataset given. Adjust the data distribution as necessary so that there is not a high skew between positive classes and negative.

(d) Reporting

18. Report your model details – draw your architecture and mention the details of hyper parameters, number of epochs trained for, dataset size used etc.
19. Report your model training accuracy, losses and validation accuracy and losses
20. Write a report and place it in a zip folder along with your code, any other material like screenshots etc.
21. Send the deliverables to: panantharama@alum.iisc.ac.in by 10th Oct 2018 10 pm

Enjoy the session, best wishes from the faculty!

P.N. Anantharaman, Ambili Rajan