ECE 157B/272B Homework 4: Text Classification

Part 1: LSTM Models

Due date: Wednesday, March 2nd, 2021, 11:59PM

Introduction

In the last several lectures, we have seen how convolutional neural networks (CNN) can be applied to various image-based machine learning tasks. If we view the CNN as the fundamental building block for image-related applications, one question naturally comes into our mind-what is the fundamental building block for solving text-based problems?

This homework will also be done through Google Colab. It is split into 2 major parts (separate submissions).

Part 1

In this part we will explore Recurrent neural networks (RNN), more specifically, Long short-term memories (LSTM) in the similar way as we did for CNN. RNN is a special type of neural network architecture designed for processing sequential data with dynamic temporal information, such as texts, speeches, etc. In this assignment, you will use tensorflow library to build a complete pipeline for data preprocessing, model building and training, result evaluation and inference. You will also utilize free GPU/TPU resources available on Google Cloud for speeding up the training. You can find two relevant tensorflow tutorials here(basic classification) and here(classification with RNN).

Data Set

We will use Keras utility function: *get_file* to download the dataset from this URL and cache it on the file system allocated by the Colab session.

The dataset contains 16000 programming questions from Stack Overflow. Each question (E.g., "How do I sort a dictionary by value?") is labeled with exactly one tag (Python, CSharp, JavaScript, or Java).

Here's the code snippet for loading the dataset from URL:

```
from tensorflow.keras import utils

data_url = \
    'https://storage.googleapis.com/download.tensorflow.org/data/stack_overflow_16k.tar.gz'
dataset = utils.get_file(
    'stack_overflow_16k.tar.gz',
    data_url,
    untar=True,
    cache_dir='',
    cache_subdir='/PATH/TO/DIRECTORY/ON/COLAB/FILESYSTEM') # Specify download directory
```

Part 1: LSTMs

All Students (157B/272B)

1. Read this blog on LSTMs. It will fill conceptual gaps.
2. (2 pts) Download the Stack Overflow dataset to a directory of your choice and inspect the folder content. "Inspect" means go the left-hand side and open the file browser of the Collab environment until you can find the data. What is the path structure (as in, what subfolders exist in the directory you made)?
3. (3 pts) Split the data into train, validation, and testing for the RNN network. You can prepare the dataset with the Keras preprocessing module.
   - (1 pts) Explain how `preprocessing.text_dataset_from_directory` finds labels for the samples.
   - (1 pts) What is the options and purpose of the input variable `label_mode`?
   - (1 pts) Perform the split
4. (7 pts) Standardize, tokenize, and vectorize the data with Keras TextVectorization layer
   - (2 pts) Write a standardization function so that all text input to the model is consistent. The standardization is up to you, but a common starting point is to lowercase all words.
   - (2 pts) EXPLAIN your chosen standardization. Why do you think your choices will help for a dataset from StackOverflow?
   - (1 pts) Train your TextVectorization layer on the training dataset with a max vocabulary size of 10,000 and output sequence length of 250. Make sure to use your standardization function!
   - (2 pts) Describe the preprocessing step of the raw sentence data and the label. (Hint: describe in your own words how the Keras TextVectorization layer works)
5. (5 pts) Visualize data and pre-processing
   - (1 pts) Show one sentence example from the training dataset. Are you able to tell which tag (python, java, javascript, csharp) this question belongs to?
   - (1 pts) What is the vector encoding for the sentence example you've printed above?
   - (1 pts) What are the **top** 10 words in your vocabulary set?
   - (1 pts) What are the **bottom** 10 words in your vocabulary set?
   - (2 pts) Do the top 10 and bottom 10 words make sense for our dataset? Explain.
6. (13 pts) Understanding LSTM models
   - (2 pts) What's the Keras Embedding layer layer for?
   - (1 pts) What do *masking* and *padding* mean in this context?
   - (3 pts) Describe in your own words why LSTM is called long short-term memories?
   - (2 pts) What the difference between using Conv1D vs LSTM?
   - (1 pts) Create a lone embedding layer and inspect it by feeding it a random input from the (vectorized) dataset. What is the input shape? What is the output shape?
   - (4 pts) Create a lone Keras LSTM layer and inspect it by feeding it the random output of the embedding layer above.
     - (1 pts) What shape of the input does it require? What is the output shape?
     - (1 pts) Set *return_sequences* and *return_states* to be True. What is the output shape now?
     - (2 pts) Explain the difference between *return_sequences* and *return_states*.
7. (12 pts) Build and train your LSTM model.

- (3 pts) Write a function that can construct an LSTM model, starting with an Embedding layer. The variable inputs should include `num_lstm_layers, num_dense_layers`. Other things you might include are dropout layers or regularizers for LSTM/Dense. Remember to end with a softmax dense layer with 4 neurons for the language classes.
- (3 pts) Write a function to train your model(s). NOTE: When training your models, use `tf.keras.callbacks.ModelCheckpoint` to save the *best* model during training, since the LSTMs can be unstable as they learn.
- (2 pts) Start with the basic model with one LSTM layer. Plot the training and validation accuracy versus epochs, and training and validation loss versus epochs. What do you observe?
- (2 pts) Add several more LSTM layers to your model. How does it affect the training accuracy? Compare to the single layer results.
- (2 pts) Evaluate both versions on the test data, and print your metrics. Include a confusion matrix. Which languages are the most difficult to differentiate?

8. (3 pts) Export the model to take raw strings as inputs (Hint: compile the TextVectorization layer and the model together to a new model). Take the following questions as input, what are the predicted tags?

   (a) "how do I extract keys from a dict into a list?"

   (b) "debug public static void main(string[] args) ..."

   Do you agree with the tags predicted by your model?

9. (2 pts) What is the conceptual difference between image classification and text classification in terms of feature extractions? (Think spatial vs temporal.) Please elaborate.

Grad/EC (272B)

1. Read the original transformers paper in preparation for next week.
2. Read the BERT paper in preparation for next week.

What to Turn In

Save your Google Colab notebook as ***.ipybn*** file and submit it to Gradescope: `https://www.gradescope.com/courses/351582`.

The Colab notebook should contain the following parts for grading:

- Answers to all the questions and diagrams/plots listed in the above questions
- Code for answering those questions that can be run to reproduce those results if needed

Note: Do not close the cell outputs after running the code cells, so that all logging and plots will be saved in the notebook for grading.

Good Luck!
ECE 157B TAs