Project 2

CE/CZ4042: Neural Networks and Deep Learning

Deadline: 15 Nov 2019

- You need to complete both parts A and B of the project and submit a project report and source codes online via NTULearn before the deadline.

- Data files for both parts are found in Project 2 folder under Assignments on NTULearn.

- The project is to be done in a group of not more than two students. Both members of the group should submit the project report and codes to NTU Learn, using their individual accounts. The cover page of the report should contain the names of both members.

- The project report would contain sections on (i) introduction, (ii) methods, (iii) experiments and results, and (iv) conclusions. The report should be submitted in the
  - lastname_firstname_P2_report.pdf (report in pdf format); and
  - lastname_firstname_P2_codes.zip (all the source codes in a zip file)

- The assessment will be based on both the project report and the correctness of the codes submitted. You are encouraged to use TensorFlow to complete the project but using other libraries is allowed. Late submissions will be penalized: 5% for each day up to three days.

- This project requires the use of GPU enabled machines. You can access these machines in Software Lab 3 located at N4-B1c-14 on
  - Mondays 12.30pm-5.30pm (Weekly),
  - Tuesdays 8.30am-2.30pm (Odd weeks),
  - Wednesdays 8.30am-12:30pm (Even weeks),
  - Thursdays 10.30am-12.30pm/2.30pm-5pm (Odd weeks), and
  - Fridays 12.30pm-5pm (Even weeks).

  Please contact the lab manager Mr. Goh Tong Hai (THGoh@ntu.edu.sg) for remote access. Note that only 10PCs are allocated for remote access.

- TAs Mr. Kelvin Chan (CHAN0899@e.ntu.edu.sg) and Mr. Wenwei Zhang (WENWEI001@e.ntu.edu.sg) are in charge of the course project. Please see them at MICL (N4-B1c-17) or PDCL (N4-B2a-02) during their office hours: Friday 3pm-4pm, in case you face issues.

Part B: Text classification

The dataset used in this project contains the first paragraphs collected from Wikipage entries and the corresponding labels about their category. You will implement CNN and RNN layers at the word and character levels for the classification of texts in the paragraphs. The output layer of the networks is a softmax layer.

The training and test datasets will be read from ‘train_medium.csv’ and ‘test_medium.csv’ files. The training dataset contains 5600 entries and test dataset contains 700 entries. The label of an entry is one of the 15 categories such as people, company, schools, etc.

The input data is in text, which should be converted to character/word IDs to feed to the networks by using ‘tf.contrib.learn.preprocessing’. Restrict the maximum length of the characters/word inputs to 100. Use the Adam optimizer for training with a batch size = 128 and learning rate = 0.01. Assume there are 256 different characters.

1. Design a Character CNN Classifier that receives character ids and classifies the input. The CNN has two convolution and pooling layers:
    - A convolution layer C_1 of 10 filters of window size 20x256, VALID padding, and ReLU neurons. A max pooling layer S_1 with a pooling window of size 4x4, with stride = 2, and padding = 'SAME'.
    - A convolution layer C_2 of 10 filters of window size 20x1, VALID padding, and ReLU neurons. A max pooling layer $S_2$with a pooling window of size 4x4, with stride = 2 and padding = 'SAME'.

    Plot the entropy cost on the training data and the accuracy on the testing data against training epochs.

    (9 marks)

2. Design a Word CNN Classifier that receives word ids and classifies the input. Pass the inputs through an embedding layer of size 20 before feeding to the CNN. The CNN has two convolution and pooling layers with the following characteristics:
    - A convolution layer C_1 of 10 filters of window size 20x20, VALID padding, and ReLU neurons. A max pooling layer S_1 with a pooling window of size 4x4, with stride = 2 and padding = 'SAME'.
    - A convolution layer C_2 of 10 filters of window size 20x1, , VALID padding, and ReLU neurons. A max pooling layer $S_2$with a pooling window of size 4x4, with stride = 2 and padding = 'SAME'.

    Plot the entropy cost on training data and the accuracy on testing data against training epochs.

    (9 marks)

3. Design a Character RNN Classifier that receives character ids and classify the input. The RNN is GRU layer and has a hidden-layer size of 20.

   Plot the entropy cost on training data and the accuracy on testing data against training epochs.

   (9 marks)

4. Design a word RNN classifier that receives word ids and classify the input. The RNN is GRU layer and has a hidden-layer size of 20. Pass the inputs through an embedding layer of size 20 before feeding to the RNN.

   Plot the entropy on training data and the accuracy on testing data versus training epochs.

   (9 marks)

5. Compare the test accuracies and the running times of the networks implemented in parts (1) – (4).

   Experiment with adding dropout to the layers of networks in parts (1) – (4), and report the test accuracies. Compare and comment on the accuracies of the networks with/without dropout.

   (8 marks)

6. For RNN networks implemented in (3) and (4), perform the following experiments with the aim of improving performances, compare the accuracies and report your findings:
   a. Replace the GRU layer with (i) a vanilla RNN layer and (ii) a LSTM layer
   b. Increase the number of RNN layers to 2 layers
   c. Add gradient clipping to RNN training with clipping threshold = 2.

   (8 marks)

Hint: Sample code is given in file start_project_2b1.py and start_project_2b2.py