Project 2

CE/CZ4042: Neural Networks and Deep Learning

Deadline: 13 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 individually.

- 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_A2_report.pdf (report in pdf format); and
    - lastname_firstname_A2_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.

TAs Mr. Kelvin Chan (chan0899@e.ntu.edu.sg), Mr. Shangchen Zhou (s200094@.e.ntu.edu.sg) and Mr. Wenwei Zhang (wenwei001@e.ntu.edu.sg) are in charge of the course project. Please send an Email to arrange meetings with them in case you face issues.

Part A: Object Recognition

The project uses a sample of the CIFAR-10 dataset: https://www.cs.toronto.edu/~kriz/cifar.html

The dataset contains RGB color images of size 32x32 and their corresponding labels from 0 to 9. You will be using the **batch_1** of the dataset, which contains 10,000 training samples. Testing is done on 2,000 test samples. The training data and testing data are provided in files **'data_batch_1'** and **'test_batch_trim'**, respectively. Sample code is given in file **start_2a.py.**

Design a convolutional neural network consisting of:

- An Input layer of 32x32x3 dimensions
- A convolution layer C_1 with 50 channels, window size 9x9, VALID padding, and ReLU activation
- A max pooling layer S_1 with a pooling window of size 2x2, stride = 2, and VALID padding
- A convolution layer C_2 with 60 channels, window size 5x5, VALID padding, and ReLU activation
- A max pooling layer S_2 with a pooling window of size 2x2, stride = 2, and VALID padding
- A fully-connected layer F_3 of size 300 with no activation
- A fully-connected layer F_4 of size 10 with Softmax activation

1. Train the network using mini-batch gradient descent learning for 1000 epochs. Set the batch size to 128, and learning rate α = 0.001.
    a. Plot the (1) training cost, (2) test cost, (3) training accuracy, and (4) test accuracy against learning epochs. One plot for the costs and one plot for the accuracies.
    b. For the first two test images, plot the feature maps at both convolution layers (C_1 and C_2) and pooling layers (S_1 and S_2) along with the test images. (In total one image and four feature maps)
    <u>Note</u>: A feature map with N channels can be viewed as N grayscale images. Do make sure that the pixel values are in the correct range when you plot them.

    (18 marks)

2. Use a grid search ($C_1 \in \{10, 30, 50, 70, 90\}, C_2 \in \{20, 40, 60, 80, 100\}$, in total 25 combinations) to find the optimal combination of the numbers of channels at the convolution layers. Use the test accuracy to determine the optimal combination. Report all 25 accuracies.

    (10 marks)

3. Using the optimal combination found in part (2), train the network by:
    a. adding the momentum term with momentum γ = 0.1,
    b. using RMSProp algorithm for learning,
    c. using Adam optimizer for learning,
    d. adding dropout (probability=0.5) to the two fully connected layers.

    Plot the costs and accuracies against epochs (as in question 1(a)) for each case. Note that the sub-questions are independent. For instance, in (d), you do not need to modify the optimizer.

    (12 marks)

4. Compare the accuracies of all the models from parts (1) - (3) and discuss their performances.

    (10 marks)

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 (Please refer to our given two smaple codes (CNN and RNN) which process text data in terms of character and word respectively). Restrict the maximum length of the characters/word inputs to **100** and the maximum number of training epoch to **250**. Use the **Adam or SGD optimizers** 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.

   (8 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.

   (8 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.

   (8 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.

   (8 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.

   (10 marks)

Hint: Sample code is given in file start_ 2b1.py and start_ 2b2.py