Homework 3

May 15, 2020

[KSE527/AI502] Deep Learning
TA Sangmook Kim, Sungnyun Kim, Younggyo Seo

HW Description

In this homework, you will study about various types of recurrent neural networks, and implement them on sentiment analysis and subway passenger prediction tasks.

The contents of this homework consist of:

- **Task1** : Implement various types of RNNs for sentiment analysis.
- **Task2** : Implement RNNs for subway passenger task.

After you make a code for each task, you should write a report according to the each problem.

Source Code
Download homework3.zip from KLMS that contains the source code for two tasks.

Before tasks : GPU setting

Because this homework takes a lot of time if you do on CPU, we recommend you to use GPU. If you do not have a GPU machine, please use Google Colab. (Also, the provided code is GPU-based code.)

Before tasks : Change the indent size (For Colab User)

For python, the indent size is very important. If indent is not proper, program will not run because of syntax error. Anyway, the indent size of our code is 4. When you use Colab first, this size might be set 2. Therefore, you have to change the indent size from 2 to 4 on Colab.

Homework 3

May 15, 2020

Task1 : Implement various types of RNNs for sentiment analysis. (10 pt)

RNNs are widely used for natural language processing (NLP) tasks. In this task, you will implement RNNs to conduct one of the simplest NLP tasks, which is sentiment analysis. You should predict a review as positive or negative. Also, after training RNNs, you will compare and analyze the performance of each type of RNN.

Implement the code and answer the questions below. We provide you with skeleton codes for building RNN which will have poor performance. The training / validation performance is no better than random guessing ($50 \sim 60\%$). Please write your own code to answer the questions.

(a) Look at the shape of tensor `hidden` and `embedded`. Have you noticed what is the problem? Explain what is the issue and report the test performance when you fix the issue. (Hint: This is related to the length of sequences. See how sequence is padded. You may use `nn.utils.rnn.pack_padded_sequence`.) (3 pt)

(b) Use different architectures, such as LSTM or GRU, and report the test performance. **Do not** change hyperparameters from (a), such as `batch_size`, `hidden_dim`,. . . . (3 pt)

Now, try to use below techniques to further improve the performance of provided source codes. Compare the test performance of each component with/without it.

(c) For now, the number of layers in RNN is 1. Try to stack more layers, up to 3. (1 pt)

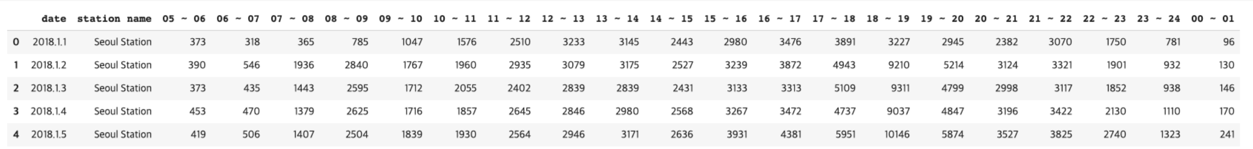
(d) Use bidirectional RNNs. (1 pt)

(e) Use dropout for regularization with stacked layers (recommended: 3 layers and dropout rate 0.5). (1 pt)

(f) Finally, apply all techniques and have an enough time to play with introduced techniques (e.g., changing hyperparameters, train more epochs, try other techniques you know, ...). Report the final test performance with your implementation and hyperparameter choice. Please note that this is not a competition assignment. We will not evaluate your assignment strictly! (1 pt)

Task2: Implement RNNs for subway passenger task.

RNN is often used to predict the next value for time series data. In this task, you will implement a model of LSTM to predict the number of passengers on the Korean subway. You will train the LSTM model using the given data ‘train.csv’ and evaluate the performance using ‘test.csv’. As can be seen in Figure 1, the data consists of date, station name, and number of passengers per hour. We will only briefly cover Seoul Station, Gangnam Station and Yeouido Station. We recommend that you open and check the ’train.csv’ and ’test.csv’ files before writing the code.



	date	station name	05 ~ 06	06 ~ 07	07 ~ 08	08 ~ 09	09 ~ 10	10 ~ 11	11 ~ 12	12 ~ 13	13 ~ 14	14 ~ 15	15 ~ 16	16 ~ 17	17 ~ 18	18 ~ 19	19 ~ 20	20 ~ 21	21 ~ 22	22 ~ 23	23 ~ 24	00 ~ 01
0	2018.1.1	Seoul Station	373	318	365	785	1047	1576	2510	3233	3145	2443	2980	3476	3891	3227	2945	2382	3070	1750	781	96
1	2018.1.2	Seoul Station	390	546	1936	2840	1767	1960	2935	3079	3175	2527	3239	3872	4943	9210	5214	3124	3321	1901	932	130
2	2018.1.3	Seoul Station	373	435	1443	2595	1712	2055	2402	2839	2839	2431	3133	3313	5109	9311	4799	2998	3117	1852	938	146
3	2018.1.4	Seoul Station	453	470	1379	2625	1716	1857	2645	2846	2980	2568	3267	3472	4737	9037	4847	3196	3422	2130	1110	170
4	2018.1.5	Seoul Station	419	506	1407	2504	1839	1930	2564	2946	3171	2636	3931	4381	5951	10146	5874	3527	3825	2740	1323	241

Figure 1: Dataset Sample for Task2

We will implement a model that predicts the number of passengers after an hour with the current number of passengers as input. For example, when we know that the number of passengers from 05 ˜06 in 2018.1.1 Seoul Station is 373, we want a model to predict the number of passengers (318) in 06 ˜07. (It is assumed that the number of passengers before 05 ˜06 is 0.)

You will learn not only to implement the LSTM model, but also to encode the input data of the LSTM model through this task. And you will check how it affects the performance of the model when encoding data with time information added as an input by extending simple encoding. You have to put the appropriate code in ’?’ so that the entire code works fine in the provided code.

(a) In the given code, put the appropriate code in ’?’ from step 1 at problem 1 to problem 3, and in problem 3, and write code to measure RMSE on test data and print RMSE value (In this case, it is calculated based on the full-batch.) (4pt)

(b) In problem 4 of Step 2, write the code to generate the encoding input data with the time information (3pt)

(c) With reference to step 1, train the model by modifying the LSTM model to match the encoding input added to the time information (2pt)

(d) In problem 6 of Step 2, write code to calculate and print the RMSE for the test dataset (0.5pt) and describe whether there was a change in performance compared to the model in Step 1, and if so, why this occurred (0.5pt)

Homework 3

May 15, 2020

About the Submission

- The deadline for submission is **23:59 on 29 May (Fri)**, and late submission is not permitted.
- You have to submit **.zip file** including both **.ipynb file** and **.pdf file**.
  (Please convert .doc file to **.pdf file**)
- File name should be **[hw3]student_ID.zip** (e.g., [hw3]20201234.zip)
  (If you do not keep this naming, there will be a disadvantage.)