Assignment 1

Report Submission Due: 22 March, 2024 (before 11:59 pm)

Submission: Submit to NTULearn (Assignments ⟶ Assignment 1) with subject "CE7455-Ass1-YourName".

Topic: Deep Learning models for Sentence Classification

Objective: The goal of this assignment is to develop a deep learning model that can accurately classify questions into predefined categories using the TREC question dataset. You will experiment with various neural network configurations to find the optimal configuration, optimal input/output formulation, model architecture, including different RNN types, bidirectionality, multiple layers, and additional features.

Code base:
https://drive.google.com/file/d/1n-XzV64cRZklZI1r-MdquybqQyTtkfdO/view?usp=sharing

In the code base, you are given a basic RNN model to classify the type of the questions in TREC dataset. You are required to develop your code based on this example.

Tasks:

1. **Configuration Optimization**: Read, understand, and reimplement the example in the code base. Adjust the code according to the following requirements.

    a. Implement the "pack padded sequence" function in the RNN library of pytorch. **Report** the result under the default setting in the code base and **Discuss** what is the benefit of this function.

    b. Keep the "pack padded sequence" function, conduct experiments with different configurations below and **Report** the best configuration as the combination of the following four options which gives the highest performance on the validation set. **Report** the final accuracy on both the validation set and the test set under the best configuration.

        i. Optimizers such as Adam, etc

        ii. Learning rates

        iii. Batch sizes

        iv. Sizes of the hidden embedding (the size of the input embedding is fixed at 100 as shown in the code base)

    c. Keep the "pack padded sequence" function and the best configuration, implement some regularization techniques. **Describe** all the regularization methods you choose and **why**. **Report** the accuracy results on the validation and test sets after applying all the regularization techniques.

2. **Input Embedding**: Based on the best configuration and regularization you tried for Task 1, change the randomly initialized input word embeddings to pre-trained word2vec embeddings as the input initialization, and keep the other settings unchanged. **Report** the accuracy on the validation set and **compare** the performance with its counterpart (in Task 1) without word2vec initialization. **Discuss** what the difference indicates.

    a. You can refer to https://radimrehurek.com/gensim/intro.html#installation on how to install gensim to in order to work with word2vec

    b. Pretrained word2vec models can be downloaded following this link https://radimrehurek.com/gensim/models/word2vec.html#pretrained-models. Use “**word2vec-google-news-300**” as the pretrained word2vec embeddings.

3. **Output Embedding**: The original code base uses the final hidden representation as the sentence embedding to be fed into the final classifier. **Describe** other option(s) to compute the sentence embedding besides taking the final hidden representation. With the best configuration (and regularizations) you have for Task 1 and the incorporation of word2vec embeddings, change the sentence embedding computation to the best option(s) you provide above. **Report** the accuracy on the validation set and **compare** the performance with the one in Task 2. **Discuss** what the difference indicates.

4. **Architecture Optimization**: The original code base uses the most simple RNN architecture. Now consider more complex architectures in the following. **Run** experiments by replacing the original RNN with each of the following architectures. **Report** the accuracy on the validation set and **compare** these results. **Discuss** your findings.

    a. GRU with a single hidden layer

    b. LSTM with a single hidden layer

    c. Bidirectional simple RNN with a single hidden layer (consisting of a forward pass and a backward pass)

    d. Simple RNN with 2 hidden layers

5. **Critical Thinking**: Based on the best setting and architecture you find from all the above 4 tasks, **describe** one more modification (other than the above optimization options) you can think of that could potentially further improve the performance. **Conduct** experiments, **report** the accuracy on the validation set and **discuss** your findings. (Hint: additional component to the model, integration with different

models, etc.)

Submission:

- **Report:** The report should contain the answers to all the above tasks, including experimental results, comparisons, analysis and explanations. The format is freestyle. Try to be concise and **not more than 4 pages**.

- **Code**: Submit your complete script in ipython notebook.