IITM-CS7016 : Topics in Deep Learning — Given on: 2nd September

Programming Assignment (Individual) — Due on : September, 22nd, 11:55pm

- The aim of this assignment is to train and test a Transformer Network for Machine Translation Tasks for EN-DE dataset.
- Collaborations and discussions with others are strictly prohibited.
- This assignment is going to be time-consuming. **PLEASE START EARLY.**
- It will be better if you use **Tensorflow/Pytorch** library (Python) for your implementation. If you are using any other languages, please contact the TAs before you proceed.
- Note that you can use the publicly available code for Transformers. Citing the public apis is **COMPULSORY**.
- You have to turn in the well documented code along with a report with **Detailed Observations** and inferences of the results electronically in Moodle.
- Typeset your report in Latex code provided (attached). It is necessary to fill '**Checklist**' in the attached sample report. Reports which are not written using Latex will not be accepted.
- The report should be precise and concise. Unnecessary verbosity, (like **Theory about RNNs**) will be penalized.
- You can find the evaluation pattern (marks distribution) at the last page.
- *You have to check the Moodle discussion forum regularly for updates regarding the assignment.*

1 Task

1.1 Problem Definition

In this assignment you will train and test a small Transformer network on the Machine Translation task for English-German Language.

1.2 Instructions

- Download the EN-DE datset from the link (https://drive.google.com/open?id=1ZLU0bpFzCm1qilR75L8sxvxuh_1eooF_). The dataset contains train split, public test split (ground truth is present) and private test split (ground truth is hidden).
- Create validation split by randomly choosing 5000 samples from train dataset.

- Train a Transformer Model (You can refer/re-use the model given at https://github.com/tensorflow/models/tree/master/official/transformer). The overall structure of the network is as follows:
  (a) **Encoder-Block**: There should be 2 Blocks of Encoder (rather than 6 in vanilla transformer) with 4 attention heads in each layer.
  (b) **Decoder-Block**: There should be 2 Blocks of Decoder with 4 attention heads at each layer of self-attention and Encoder-Decoder attention.
  (c) The $d_{model} = 512$, and $d_k = 128$. Keep the vocab-size for source and destination language as 30000 using BPE.
- The objective function is to minimize the negative log-likelihood. For inference, you should implement a greedy decoding. Additionally, you can also try to implement beam search. Implementation of beam search is optional (no extra credits, only for your interest and curiosity).
- Train the network using Adam using the entire training dataset. Use the valid set for validation.
- Use dropout and layer-normalization as specified in the paper[1]
- You have to run the private_test.en set on the trained model and submit the translated file where each line will correspond to the translation for the respective line in source file.
- The primary evaluation metric for the task is BLEU score [2]. We will evaluate your submitted translated sentences and rank the assignments based on these BLEU scores.

1.3 Report

Prepare a report containing the observations and inferences for the following:

1. Implement and run Transformer and report the best performing parameters [25 marks]
2. BLEU score on public_test.en split. [5 marks]
3. BLEU score on private_test.en split . [10 marks]
4. A plot of the learning curve showing iterations on the x-axis and negative log likelihood over labels on the y-axis. Make a single plot showing both the training loss and the validation loss. [5 marks]
5. Report a table with validation accuracies with different hyperparameter settings (variation in hidden size, dropout/nodropout, different optimizers, learning rate, etc). [15 marks]

[1] https://arxiv.org/pdf/1706.03762.pdf
[2] https://github.com/Maluuba/nlg-eval

6. Visualize the attention plots across layers for Encoder [10 marks]

7. What is the effect if all 8 heads are in one layer at Encoder/Decoder self-attention layers ? [10 marks]

8. Based on the attention plots, is there any specific characteristic that can be associated to that attention head ? [5 marks]

9. What happens if you tie the weights of Key and Value in the attention mechanism ? [5 marks] If you use a seq2seq model based on LSTM/Attention mechanism with Bidirectional LSTM, hidden size=512 , what is the difference in training time for one epoch ? You can use any inbuilt code for Seq2Seq architectures. [10 marks]

1.4 Submission Instructions:

You need to submit the source code for the assignment.

All other supporting files used for generating plots, etc. should also be placed in the zip file. You need a single folder (RollNoTeamMemberA_PA1, *e.g.* CS14B042_PA1) containing the following:

- `train.py`
- `run.sh (containing the best hyperparameters setting)`
- `any other python scripts that you have written`
- `'report.pdf' (in Latex) of the results of your experiments with neatly written answers to the questions mentioned in the above report section`
- `'private_test_translations.txt' should contain the translations on private_test.en( one per line)`

The zip should be named as RollNoTeamMemberA_PA1.zip, (*e.g.* CS14B042_PA1.zip). Note that zip file and folder name **SHOULD BE SAME**.