Université de Montréal
Faculté des arts et des sciences
Département d'**informatique et de recherche opérationnelle**

IFT6289-H22 Assignment 3: Transformers and Pretrained Language Models

March 19, 2023

- *Due Date:* ***April 7rd, 2023 (23:59 pm, EST timezone)***
- *Assignment 3 should be completed by individuals, which is worth* ***15%*** *of your grade.*
- *Maximum* ***2*** *late days. Late submissions would result in a lower grade.*
- *Please Submit your PDF report and predictions 'FirstName_(MiddleName)_LastName.csv' on Studium.*

In this assignment, you will analyse the design of the Transformer architecture, and fine-tune GPT-2 on the given datasets.

Part 1: Analyzing the Design of Transformer

1. Scaled Dot-Product Attention.

The output of Transformer's self-attention is computed as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

Where d_k is each key vector's dimension.

1. (1pt) Assume that q and k are both d_k-dimensional vectors, and each component of q and k are independent variables with mean 0 and variance 1. Prove that $Var\left(\frac{q \cdot k}{\sqrt{d_k}}\right) = 1$.
2. (1pt) In self-attention, we use different matrices to transform the representation of tokens into Q, K and V. Now, suppose $Q = K$. From the characteristics of matrix $QK^\top$, specify **at least two** consequences of this.

2. Concatenation vs. Summation of Embeddings.

(1pt) Different from the token ordering in RNNs, Transformers naturally lacks the ability to represent token positions, so one needs to use positional encoding or positional embedding as a compensation. Assume we have a token embedding $e_i \in \mathbb{R}^{d_e}$, its corresponding positional encoding vector $p_i \in \mathbb{R}^{d_e}$, and two transform matrices $W_e \in \mathbb{R}^{d_h \times d_e}$ and $W_p \in \mathbb{R}^{d_h \times d_e}$. If we denote concatenation on the first dimension as $[\cdot;\cdot]$, please express $[W_e^\top; W_p^\top]^\top [e_i; p_i]$ by a summation of two terms relating to e_i and p_i respectively.

3. Positional Encoding.

In Vaswani et al. [2017], the sinusoidal positional encoding of the t-th token is as follows:

$$P_{t,(2i)} = \sin(c_i t)$$
$$P_{t,(2i+1)} = \cos(c_i t),$$

where $P_{t,(2i)}$ is the $(2i)$-th component of P_t, and c_i is a constant related to the position in the embedding, which is set to $\frac{1}{10000^{\frac{2i}{d}}}$ in Vaswani et al. [2017].

1. (1pt) Prove that the result of $P_t^\top P_{t+k}$ only depends on k. This indicates that the dot product of the sinusoidal positional encoding can reflect the relative distance between tokens.

2. Now, we dive into more details about the interaction between token embedding and positional embedding in transformers. To make the analysis simpler, we focus on the self-attention mechanism in the first layer of Transformer. The attention score between the i-th and j-th token is as below:

   –

   $$\alpha_{ij} = \frac{1}{\sqrt{d_k}} \left(W^{Q,1}(e_i + p_i)\right)^\top \left(W^{K,1}(e_j + p_j)\right),$$

   where the 1's in the superscripts indicate the first layer.

   (a) (1pt) Expand the expression of the attention score, and explain how does the token embeddings and positional encoding interact with each other?

   (b) (1pt) Based on the result of the previous subquestion, does this form guarantees that sinusoidal positional encoding reflects the relative distance between tokens? Why?

   (c) (1pt) **Brainstorming**: Is this form reasonable? Based on your results, can you think of potential ways to improve Transformers?

Part 2: GPT-2 Fine-tuning

(8 pt) In this part, you will use one of the most popular pretrained language models (PLMs), GPT-2 [Radford et al., 2019], on the given movie review datasets. In this datasets, each sample contains a piece of movie review with its corresponding sentiment label (1 for positive and 0 for negative). In the code base, we provide the complete code for fine-tuning the GPT-2 model.
Note:

- In this part, you **can use as many tricks as you can to improve the performance**.
- You only have to upload the predictions on the test set of the datasets in '.csv' format. Please refer to the 'test_sample.csv' for an example. Note that you should rename the file name to 'First-Name_(MiddleName)_LastName.csv'. E.g. Andrew Yan-Tak Ng should be 'Andrew_Yan-Tak_Ng.csv', Yoshua Bengio should be 'Yoshua_Bengio.csv'.
- If you do not have access to other GPU resources, we encourage you to use Google Colab's GPU for training.
- In your report, please include your analysis of results on the validation set, including but not limited to accuracy, precision, recall, ROC-AUC curves, and F1 score.
- In your report, please introduce how you improved your model and what techniques and tricks were used in your model.
- The ranking board will be publicly available after the submission deadline.

Our grading scheme:

1. Students submit PDF report and their model predictions for the test set, e.g. 'report.pdf' and 'Bang_Liu.csv'.

2. Calculate the number of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) based on the model's predictions and the true labels of the test data.

3. Calculate the F1 score for each student's model using the following formula:
   F1 score = 2 * (Precision * Recall) / (Precision + Recall)
   where Precision = TP / (TP + FP) and Recall = TP / (TP + FN)

4. Rank the students based on their F1 score, with the highest score being the top rank.

5. If there are ties in the F1 score, then we compare their accuracy, then precision, then recall.

6. Your final grade for Part 2 is the summation of report points, base points, and rank points. You will get 4 base points if your accuracy is above 70%. The calculation of rank points is:

   Rank Points = $(1 - \frac{\text{Your Rank}-1}{\text{Number of Valid Submission}}) \times 4$

   For example, you analyze your results well in the report and get full (2) report points. And your predictions achieve over 70% accuracy in our evaluation, so you get full (2) base points. And your F1-score rank is 10 out of 30 valid submissions, so you can get $(1 - \frac{10-1}{30}) \times 4 = 2.8$ rank points. Overall, your final grade for Part 2 is 2 + 2 + 2.8 = 6.8.

Writing your report

1. The template of the assignment 3 report is the same as before, which can be downloaded from StudiUM or slack.

2. Please indicate in your report if you used any open-source codes or materials.

Submission Instructions

Submit your PDF report and 'FirstName_(MiddleName)_LastName.csv' on Studium.

References

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, 2017.