IST 597: Foundations of Deep Learning **(Due: 03/30/22)**

Assignment #10000

Instructor: **Dr. Lee Giles** *TA:* **Ankur Mali** *Name: , PSUID:*

Created by Ankur Mali
Source : Thanks and credits to materials from MIT, Google, NYU

Course Policy: Carefully read all the instructions, before you start working on the assignment

- Please typeset your submissions in any LaTeX or word template, give maximum explanation for each sub-problems. Please include your name and PSUID with submission along with date and time on the first page.
- Assignments are due before class at 02:29:59 pm {Please check the due date on the **Official course** webportal} Portal.
- Please avoid single line answers, submissions without any explanations would receive **0** points.
- Late assignments will suffer 50 percent points reductions per day, formula is $x/2$, where x=number of days and counter will start at 02:30:00 pm.
- All source materials must be cited. The University Academic Code of Conduct will be strictly enforced.
- We will be creating Canvas submission page for this. Submit all files on Canvas.
- All queries related to Assignment should have a subject line IST597:Assignment10000 Queries

Problem 1. Compare recurrent and transformers based language model. (11 points)

I would advise to read classical papers on Long short term memory [2] and Transformers [3, 1] to better understand the overall architecture. Language can be treated as bunch of sequential symbols feed into the network (stateful models or recurrent models) or one can break the sequence, by masking the input at random orders (stateless models such as transformers). Important things to remember while working with these models, transformers can scale to billions of parameters and can be trained parallely across clusters, whereas RNNs are sequential in nature (Backprop through time), thus comparatively slower to train. On the other hand RNNs have theoretical guarantee and when number of parameters are identical to transformers ,majority time performance is comparable. Finally, RNNs have generative capability, but this is not true for transformers. So based on use-cases one should choose the appropriate architectures.

Things to do in this Assignment:

- We have provided a codebase (starter code) for RNN and transformers trained on IMDB.
- You will train RNNs from scratch and fine-tune transformers on this dataset.
- Use identical splits (train/val/test) for both architectures.
- Depending on available compute or resources, you can play with any kind of RNN (vanilla RNN, LSTM, GRU etc) and any pre-trained transformer weights (Albert, Bert-small, Bert-Large, Electra, etc)
- Analyze performance for both models with various hyper-parameters (batch sizes, learning rate and optimizer[SGD, ADAMW, RMSPROP, ADAM]) [1].
- Report your findings and also report time taken by your network. Provide reasoning, what is more important overall performance, inference time or interpretability. Also comment on overall observations (difficulties faced during training, model stability, etc).

[1] You have freedom to use any set of hyper-parameters

References

[1] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[2] Hochreiter, S., and Schmidhuber, J. Long short-term memory. *Neural computation 9*, 8 (1997), 1735–1780.

[3] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems 30* (2017).