Supplementary Material for: Fast Monte-Carlo Approximation of the Attention Mechanism

Hyunjun Kim and JeongGil Ko

School of Integrated Technology, Yonsei University
hyunjun.kim@yonsei.ac.kr, jeonggil.ko@yonsei.ac.kr

A. Proof for Lemma1 and Theorem 2

We provide formal proofs for Lemma 1 and Theorem 2. The underlying technique for the proof are borrowed from lecture notes on *Randomized Linear Algebra* (Mahoney 2016)

Proof of Lemma 1

Since the variance of $H[i,j]$ is

$$\mathbb{V}[H[i,j]] = \frac{1}{r_i}\sum_{k=1}^{n}\frac{X[i,j]^2 W[k,j]^2}{p(k)} - \frac{1}{r_i}(X[i]W)^2[i,j], \tag{1}$$

The mean Frobenius norm error is evaluated as:

$$\begin{aligned}\mathbb{E}[\|H[i]-X[i]W\|_F^2] &= \sum_{j=1}^{d}\mathbb{E}[(H[i]-X[i]W)^2[i,j]] \\ &= \sum_{j=1}^{d}\mathbb{V}[H[i,j]] \\ &= \frac{1}{r_i}\sum_{k=1}^{d}\frac{X[i,k]^2\|W[k]\|_2^2}{p(k)} - \frac{1}{r_i}\|X[i]W\|_F^2 .\end{aligned} \tag{2}$$

If sampling probability $p(k) = \frac{\|W[k]\|_2^2}{\|W\|_F^2}$, then

$$\begin{aligned}\mathbb{E}[\|H[i]-X[i]W\|_F^2] &= \frac{1}{r_i}\sum_{k=1}^{d}X[i,k]^2\|W\|_F^2 - \frac{1}{r_i}\|X[i]W\|_F^2 \\ &\le \frac{1}{r_i}\sum_{k=1}^{d}X[i,k]^2\|W\|_F^2 \\ &\le \frac{1}{r_i}(\|X[i]\|_2\|W\|_F)^2\end{aligned} \tag{3}$$

Therefore, following the Cauchy-Schwarz inquality yields:

$$\mathbb{E}[\|H[i]-X[i]W\|_F] \le \frac{1}{\sqrt{r_i}}\|X[i]\|_2\|W\|_F \tag{4}$$

Proof of Theorem 2

Given that the approximated output $\tilde{Y}[i]$ is expressed as

$$\tilde{Y}[i] = \sum_{j=1}^{n}A[i,j]H[i], \tag{5}$$

Following Lemma 1, the approximated output Frobenius norm error is

$$\mathbb{E}[\left\|\tilde{Y}[i]-Y[i]\right\|_F] \le \sum_{j=1}^{n}\frac{A[i,j]}{\sqrt{r_i}}\|X[i]\|_2\|W\|_F . \tag{6}$$

Since sample size r_i is determined as

$$\sqrt{r_i} = \frac{n\cdot\max A[:,i]}{\alpha}, \tag{7}$$

Let $\beta = \frac{1}{n}\sum_{j=1}^{n}\|X[i]\|_2$. If $A[i,j] > 0$ for all (i, j) then

$$\begin{aligned}\mathbb{E}[\left\|\tilde{Y}[i]-Y[i]\right\|_F] &\le \sum_{j=1}^{n}\frac{\alpha A[i,j]}{n\max A[:,i]}\|X[i]\|_2\|W\|_F \\ &\le \frac{\alpha}{n}\sum_{j=1}^{n}\|X[i]\|_2\|W\|_F \\ &= \frac{\alpha\beta}{n}\|W\|_F .\end{aligned} \tag{8}$$

Furthermore, recall Markov's inequality with $\gamma \ge 0$ and non-negative real-valued random variable M where

$$\Pr[M \ge \gamma] \le \frac{\mathbb{E}[M]}{\gamma}, \tag{9}$$

Setting a failure probability δ where

$$\delta = \Pr[\left\|\tilde{Y}[i]-Y[i]\right\|_F > \frac{\alpha\beta\gamma}{n}\|W\|_F] \tag{10}$$

Based on Markov's inequality and Equation 8,

$$\delta \le \frac{\mathbb{E}[\left\|\tilde{Y}[i]-Y[i]\right\|_F]}{\frac{\alpha\beta\gamma}{n}\|W\|_F} \le \frac{1}{\gamma} \tag{11}$$

which indicates $\gamma = \frac{1}{\delta}$. Therefore, with probability $1-\gamma$,

$$\left\|\tilde{Y}[i]-Y[i]\right\|_F \le \frac{\alpha\beta}{\delta}\|W\|_F . \tag{12}$$

Task	Fine-tuned weights
CoLA	textattack/bert-base-uncased-CoLA
MNLI	ishan/bert-base-uncased-mnli
MRPC	bert-base-cased-finetuned-mrpc
QNLI	textattack/bert-base-uncased-QNLI
QQP	textattack/bert-base-uncased-QQP
RTE	textattack/bert-base-uncased-RTE
SST-2	textattack/bert-base-uncased-SST-2
STS-B	textattack/bert-base-uncased-STS-B
WNLI	textattack/bert-base-uncased-WNLI

Table 1: ID of fine-tuned model weights for BERT

Task	Fine-tuned weights
CoLA	textattack/distilbert-base-uncased-CoLA
MNLI	textattack/distilbert-base-uncased-MNLI
MRPC	textattack/distilbert-base-uncased-MRPC
QNLI	textattack/distilbert-base-uncased-QNLI
QQP	textattack/distilbert-base-uncased-QQP
RTE	textattack/distilbert-base-uncased-RTE
SST-2	avneet/distilbert-base-uncased-finetuned-sst2
STS-B	textattack/distilbert-base-uncased-STS-B
WNLI	textattack/distilbert-base-uncased-WNLI

Table 2: ID of fine-tuned model weights for DistilBERT

B. Transformer Configurations

Our implementation and pre-trained model weights on MCA-BERT, MCA-DistilBERT, and MCA-Longformer is based on the Huggingface Transformers (Wolf et al. 2020) library version 4.10[1], which is open-sourced on GitHub.

Fine-tuning on GLUE Benchmarks

We make use of publicly available fine-tuned model weights from the Huggingface Transformers repository [2] for efficient reproducibility. We report model weight IDs that were used for our experiments in Table 1 and Table 2.

Fine-tuning on Document Classification

On the other hand, fine-tuned weights for our document classification datasets (i.e., AAPD, IMDB, and HND) on Longformer are not found on the public repository. Using `allenai/longformer-base-4096` as a base language model, we fine-tune each dataset using hyperparameter configurations from the original paper (Beltagy, Peters, and Cohan 2020).

References

Beltagy, I.; Peters, M. E.; and Cohan, A. 2020. Longformer: The Long-Document Transformer. *CoRR*, abs/2004.05150.

Mahoney, M. W. 2016. Lecture Notes on Randomized Linear Algebra. *CoRR*, abs/1608.04481.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; Davison, J.; Shleifer, S.; von Platen, P.; Ma, C.; Jernite, Y.; Plu, J.; Xu, C.; Scao, T. L.; Gugger, S.; Drame, M.; Lhoest, Q.; and Rush, A. M. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 38–45. Online: Association for Computational Linguistics.

[1]https://github.com/huggingface/transformers

[2]https://huggingface.co/models