INHIBIDISTILBERT: KNOWLEDGE DISTILLATION FOR A RELU AND ADDITION-BASED TRANSFORMER

Tony Zhang & Rickard Brännvall
Department of Computer Science, RISE Research Institutes of Sweden, Luleå, Sweden
rickard.brannvall@ri.se

ABSTRACT

This work explores optimizing transformer-based language models by integrating model compression techniques with inhibitor attention, a novel alternative attention mechanism. Inhibitor attention employs Manhattan distances and ReLU activations instead of the matrix multiplications and softmax activation of the conventional scaled dot-product attention. This shift offers potential computational and energy savings while maintaining model effectiveness. We propose further adjustments to improve the inhibitor mechanism's training efficiency and evaluate its performance on the DistilBERT architecture. Our knowledge distillation experiments indicate that the modified inhibitor transformer model can achieve competitive performance on standard NLP benchmarks, including General Language Understanding Evaluation (GLUE) and sentiment analysis tasks.

Introduction. Transformer-based language models have revolutionized natural language processing (NLP), achieving state-of-the-art performance across a wide range of tasks, from machine translation to sentiment analysis (Vaswani et al., 2023). However, the computational and energy demands of these models, particularly those arising from the self-attention mechanism, pose significant challenges for deployment in resource-constrained environments. The self-attention mechanism, while highly effective, relies heavily on matrix multiplications, which are computationally expensive and energy-intensive. As the scale of transformer models continues to grow, so does their environmental impact, with studies estimating that training a single large model can emit as much carbon as five cars over their lifetimes (Strubell et al., 2019). This has spurred research into more efficient alternatives, including model compression techniques such as knowledge distillation (Sanh et al., 2020) and alternative attention mechanisms, like ReLUFormer (Shen et al., 2023) or Linformer (Wang et al., 2020). Another alternative is inhibitor attention (Brännvall, 2024), which was introduced as a means to avoid using the softmax function and matrix multiplications.

The motivation for this work is driven by the potential advantages of inhibitor attention over conventional dot-product-based attention under low-bit precision quantization. Scaled dot-product attention relies on floating-point matrix multiplication and Softmax activations, which can become challenging when quantized, leading to precision loss. This work takes a first step towards inhibitor transformer model compression by demonstrating that it can be trained via knowledge distillation to perform well on NLP benchmark tasks. Conducted as a Master Thesis project during the fall of 2024, it faced several resource limitations (e.g., access to powerful GPUs). Therefore, it is presented here as a Tiny-paper contribution to the SLLM workshop to invite collaborators.

Inhibitor attention. The conventional scaled dot-product attention is replaced according to

$$S = \frac{QK^T}{\sqrt{d_k}} \quad \Longrightarrow \quad Z_{ij} = \sum_k \frac{\gamma}{\sqrt{d_k}} |Q_{ik} - K_{jk}| \tag{1}$$

where Q, K, V are the same query, key, and value matrices of (Vaswani et al., 2023) and d_k is the size of the latent dimension. The attention head output is then similarly replaced

$$H = \text{softmax}(S)V \quad \Longrightarrow \quad H'_{ik} = \eta \sum_j (V^+_{jk} - \bar{Z}_{ij})^+ + (V^-_{jk} + \bar{Z}_{ij})^- \tag{2}$$

such that the attenuating effect of the softmax function is instead obtained by ReLU applied separately to the positive and negative parts of V, thresholded by the inhibitor attentions score $\bar{Z}_{ij}$, which here has been centered and then shifted by an amount δ (see appendix). Here, $(x)^+ = \max(x, 0)$ and $(x)^- = \min(x, 0)$. In this work, we introduce learnable scalar parameters γ, η, and δ.

Experiments. Our experiments were based on the DistilBERT (Sanh et al., 2020) paper, but instead of using a full-sized BERT model as the teacher, we used the smaller pre-trained DistilBERT model for computational convenience and simpler alignment. The weights were also initialized from the teacher model. We only discuss task-agnostic KD and refer to the Appendix for supplementary information on the experiments, including results for task-specific KD and hyperparameter listings.

The initial phase involved layerwise training to align the contextual representations between the teacher and student models using 10% of the Wikitext-103 corpus. In this phase, all weights in the student model were frozen except for the weight matrices of the query, key, and value components in the current layer being trained. The Mean Squared Error (MSE) loss function was applied to align the context outputs of corresponding layers in both models. Each layer was trained iteratively from the bottom to the top layer (layer 0 to layer 5). After training a layer for two epochs, its weights were frozen, and the next layer in the sequence was unfrozen.

Following the layerwise training, a full-layer training phase was conducted using 60% of the Wikipedia 20231101 corpus. In this phase, all layers in the student model were unfrozen, allowing parameter updates across the entire network. MSE loss was applied to the hidden states to align the hidden layer outputs between the teacher and student models.

Once the task-agnostic knowledge distillation was completed, the final weights of the inhibitor DistilBERT model were stored and used as the foundation for fine-tuning to more specific NLP tasks.

Table 1: Experiment comparing a pre-trained conventional DistilBert with the Inhibitor alternative pre-trained by task-agnostic knowledge distillation. Each model was finetuned to the GLUE Benchmarks and the IMDB tasks. The performance on each test was averaged over three runs.

Models	**Score**	**CoLA**	**MNLI**	**MRPC**	**QNLI**	**QQP**	**RTE**	**SST-2**	**STS-B**	**WNLI**	**IMDb**
Conv. DistilBERT	77.0	51.3	82.2	87.5	89.2	88.5	59.9	91.3	86.9	56.3	92.82
Inhibi.DistilBERT	74.5	40.0	79.2	86.8	85.4	89.5	59.2	90.2	83.5	56.3	92.81

Results. We evaluate our inhibitor-based DistilBERT model against the conventional scaled dot-product attention DistilBERT on the GLUE benchmark (Wang et al., 2019), which consists of 9 different language understanding tasks. We fine-tuned each model for three epochs using the AdamW optimizer in accordance with standard practices also followed in the original DistilBERT paper. For reference, results for the IMDB sentiment analysis task are also presented.

The performance comparison in Table 1 indicates that a fine-tuned inhibitor DistilBERT achieves competitive accuracy, with a modest 3.2% average drop on GLUE compared to dot-product DistilBERT across the different tasks. Overall, the fine-tuned inhibitor maintained competitive performance across most tasks but faced notable challenges for the CoLA task, which would require a more detailed analysis as to why. We note that a performance drop may be expected as we used the original Distilbert model both as a teacher model and as a benchmark baseline.

Discussion. We demonstrate that inhibitor-based attention mechanisms can achieve competitive performance on NLP benchmarks while relying on simpler arithmetic operations. However, while theoretical analysis suggests potential energy savings, actual measurements on conventional server hardware showed higher energy consumption and lower throughput compared to traditional dot-product attention. This discrepancy underscores the need for specialized hardware, such as custom FPGA designs optimized for ReLU and addition-based operations, to fully realize the theoretical benefits of this novel attention mechanism. The original report can be provided upon inquiry for a detailed analysis of energy efficiency and throughput.

Future work will focus on further optimizations, including specialized hardware implementations and different model compression techniques, such as quantization. Additionally, exploring knowledge transfer from a full-sized BERT model instead of DistilBERT could improve performance. With more powerful GPUs, direct pretraining of Inhibitor Transformers and experiments with larger, modern architectures would be feasible. Additionally, efforts could assess the mechanism's performance in generative language models (GPT family) and Vision Transformers.

REFERENCES

Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. In *Advances in Neural Information Processing Systems*, volume 19, pp. 153–160. MIT Press, 2007.

Rickard Brännvall. The Inhibitor: ReLU and Addition-Based Attention for Efficient Transformers (student abstract). *Proc of the Thirty-Eighth AAAI Conf on Artificial Intelligence (AAAI-24)*, 2024.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020.

Kai Shen, Junliang Guo, Xu Tan, Siliang Tang, Rui Wang, and Jiang Bian. A study on relu and softmax in transformer. https://arxiv.org/abs/2302.06461, 2023.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in NLP. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3645–3650, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1355.

Ashish Vaswani et al. Attention is all you need. https://arxiv.org/abs/1706.03762, 2023.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding, 2019. URL `https://arxiv.org/abs/1804.07461`.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. https://arxiv.org/abs/2006.04768, 2020.

Jiao Xiaoqi et al. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4163–4174, Online, November 2020. Association for Computational Linguistics.

APPENDIX: SUPPLEMENTARY MATERIAL

MORE ON THE METHOD

Conventional transformers Vaswani et al. (2023) utilize the scaled dot-product attention defined as

$$S = \frac{QK^T}{\sqrt{d_k}} \tag{3}$$

$$H = \text{softmax}\left(S\right)V \tag{4}$$

where Q, K, V are query, key, and value matrices. The inhibitor Brännvall (2024) rewrites attention

$$Z_{ij} = \sum_k \frac{\gamma}{\sqrt{d_k}} |Q_{ik} - K_{jk}| \tag{5}$$

where the Manhattan (L1) distance replaces the dot-product (which is related to cosine similarity).

The final output is computed as

$$H'_{ik} = \eta \sum_j (V^+_{jk} - \bar{Z}_{ij})^+ + (V^-_{jk} + \bar{Z}_{ij})^- \tag{6}$$

where we used the notation $(x)^+ = \max(x, 0)$ and $(x)^- = \min(x, 0)$ for the positive and negative ReLU functions, respectively.

To allow for further calibration of the inhibition effect, a shift is applied to the inhibition score by first centering the score and then adjusting it with a shifting parameter δ

$$\bar{Z}_{ij} = (Z_{ij} - \langle Z_{ij} \rangle_j - \delta)^+ \tag{7}$$

where $\langle Z_{ij} \rangle_j$ denotes that the mean is calculated over the axis corresponding to index j. The purpose is to control when values from V can pass through unmodified. This article introduces a new set of learnable scalar parameters for the inhibitor, γ, η, and δ, which are specific to each attention head.

More on the Knowledge Distillation Experiment

This section supplements the main text with more details on the knowledge distillation experiments, including a comparison with task-specific knowledge distillation presented in Table 2 that expands on the comparison between fine-tuned transformers of Table 1 for conventional dot-product attention and inhibitor attention. At the end of this appendix, we have included tables that list the hyperparameters used in our experiments.

Our approach draws inspiration from the original DistilBERT (Sanh et al., 2020), TinyBERT (Xiaoqi et al., 2020), and Greedy Layer-Wise Training (Bengio et al., 2007) papers. For the task-agnostic KD experiments, we used as teacher the smaller pre-trained DistilBERT model for computational convenience and simpler alignment, while a BERT already fine-tuned to the GLUE task was used for the task-specific KD. Weights were initialized from the teacher model.

Experiment 1. We transfer knowledge from a dot-product-based DistilBERT teacher model to an inhibitor-based DistilBERT student model using a **task-agnostic knowledge distillation strategy**:

- **Layerwise Training**: Each layer was trained sequentially using Mean Squared Error (MSE) loss to align self-attention outputs. Only the query, key, and value matrices of the current layer were updated, while all other weights remained frozen. The layerwise training followed a bottom-up approach, freezing all layers except for the current one being trained.
- **Full-Layer Training**: After layerwise alignment, all layers were unfrozen and trained together using MSE loss applied to hidden states to refine the representations.

Experiment 2. Building upon the task-agnostic distillation, we performed **task-specific knowledge distillation** using a fine-tuned BERT model on GLUE benchmark (Wang et al., 2019) tasks. The goal was to transfer task-specific knowledge to the inhibitor model by using the BERT model as a teacher. The loss components consisted of:

- **Soft Probability Distillation Loss**: The distillation loss function uses the teacher model's soft probabilities to encourage the student model to replicate the teacher's predictions.
- **Hidden State Loss**: To help guide the inner layers of the student model toward better alignment with the teacher's representations.

The performance comparison in Table 2 indicates that fine-tuned inhibitor DistilBERT (FT Inhibi.DistilBERT) achieves competitive accuracy, with a modest 3.2% average drop on GLUE compared to dot-product DistilBERT across different tasks. Overall, the fine-tuned Inhibitor DistilBERT (trained by task-agnostic knowledge distillation) maintained competitive performance across most tasks but faced notable challenges in tasks like CoLA, which would require a more detailed analysis.

Task-specific knowledge distillation (KD Inhibi.DistilBERT) lags behind, indicating that improvements in layer alignment and training strategies are needed. Although the task-specific model performs somewhat better on the CoLA benchmark, it shows materially worse results for most other benchmarks, particularly MNLI, MRPC, QNLI, and QQP.

Results from a third experiment on computational efficiency were inconclusive (details can be provided upon inquiry). While our theoretical analysis suggested potential energy savings, the practical experiments showed higher energy consumption and lower throughput on the available computer architecture, indicating the need for further experimentation on more specialized hardware.

Table 2: Supplementary results showing that the performance on GLUE is somewhat weaker for Task-Specific Knowledge Distillation (KD, bottom row) compared to the case with Fine-Tuning after Task-Agnostic Distillation (FT, same as Table 1). Task scores are averaged over three runs.

Models	Score	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	WNLI
Conv. DistilBERT (FT)	77.0	51.3	82.2	87.5	89.2	88.5	59.9	91.3	86.9	56.3
Inhibi.DistilBERT (FT)	74.5	40.0	79.2	86.8	85.4	89.5	59.2	90.2	83.5	56.3
Inhibi.DistilBERT (KD)	68.7	47.5	72.2	77.0	80.0	63.4	47.3	91.0	83.5	56.3

Table 3: Hyperparameters for the layer-wise training.

Hyperparameter	**Value**
Number of Layers	6
Hidden size	768
FFN inner hidden size	3072
Attention heads	12
Attention head size	64
Dropout	0.1
Attention Dropout	0.1
Warmup Ratio	5%
Peak Learning Rate	5e-4
Batch Size	16
Gradient accumulation steps	4
Epochs	2
Learning Rate Decay	Cosine
Adam ϵ	1e-8
Adam β_1	0.9
Adam β_2	0.999

Table 4: Hyperparameters for the full-layer training.

Hyperparameter	**Value**
Number of Layers	6
Hidden size	768
FFN inner hidden size	3072
Attention heads	12
Attention head size	64
Dropout	0.1
Attention Dropout	0.1
Warmup Ratio	5%
Peak Learning Rate	3e-4
Batch Size	16
Gradient accumulation steps	32
Epochs	3
Learning Rate Decay	Cosine
Adam ϵ	1e-8
Adam β_1	0.9
Adam β_2	0.999

Table 5: Hyperparameters for the task-specific knowledge distillation.

Hyperparameter	**Value**
Number of Layers	6
Hidden size	768
FFN inner hidden size	3072
Attention heads	12
Attention head size	64
Dropout	0.1
Attention Dropout	0.1
Warmup Ratio	0
Peak Learning Rate	2e-5
Batch Size	16
Gradient accumulation steps	0
Epochs	3
Learning Rate Decay	Linear
Adam ϵ	1e-8
Adam β_1	0.9
Adam β_2	0.999
Temperature	4
Distillation loss weight	0.5
Hidden state loss weight	0.5

Table 6: Hyperparameters for fine-tuning after task-agnostic distillation.

Hyperparameter	**Value**
Number of Layers	6
Hidden size	768
FFN inner hidden size	3072
Attention heads	12
Attention head size	64
Dropout	0.1
Attention Dropout	0.1
Warmup Ratio	0
Peak Learning Rate	2e-5
Batch Size	16
Epochs	3
Learning Rate Decay	Linear
Adam ϵ	1e-8
Adam β_1	0.9
Adam β_2	0.999