Project Report

Knowledge Distillation on Large Language Models

Hussein Barakat
University of Waterloo
hbarakat@uwaterloo.ca
Student ID: 21038582

Marcus Huang
University of Waterloo
m43huang@uwaterloo.ca
Student ID: 21061943

1 Introduction

The recent triumph of deep learning in modeling highly non-linear problems has established neural networks as the cornerstone in artificial intelligence (AI) tasks [1]. This success has motivated both academia and industry to expand and deepen the models to capture higher levels of non-linearity in complex tasks such as natural language processing and visual tasks. As a result, hyper-parameterization (up to billions of parameters) became a common characteristic in the state-of-the-art (SOTA) models in standard AI tasks.

The increase in the size of models has made it difficult to make inferences from large models in industrial/deployment conditions due to high computational costs, memory consumption, and high latency, especially with low-compute resource systems and real-time applications [2]. This has led to various techniques being developed on both software level and hardware level to tune deep learning models to deployment environment via compression and acceleration. To demonstrate the benefits of addressing the issue of over-parameterization, a model such as ResNet 50, which requires more than 95MB of memory for storage and 3.8 billion multiplications, can be compressed using appropriate compression techniques to save more than 75% of parameters and 50% of the inference time [2, 3].

Hardware acceleration techniques mainly aim to improve parallelism and to optimize memory access by adapting domain specialized architectures. Modern architectures have adopted novel Network-on-chip protocols to orchestrate data flow across processing elements. In addition, techniques including Processing-In-Memory and different memory hierarchies were integrated to improve data reusability, and therefore memory access time. Despite the advancement in this research area, sole hardware optimization has a performance upper bound. In addition to the proven potentiality of algorithm-based and co-design solutions to optimize performance [1].

Algorithm-based compression techniques can be classified into four main categories: Pruning, Quantization, low rank factorization and Distillation [4]. Pruning addresses the problem by reducing the number of parameters by removing the redundant parameters with the least contribution to the network's output, therefore having the least impact on the performance. Quantization reduces the precision of the parameters and the activations in the network, which permits exploiting hardware instructions to accelerate the inference. Low-rank factorization structure the matrix reduction process by decomposing the weight matrices into smaller components. Knowledge distillation involves training a small model by distilling the knowledge obtained by a pre-trained larger model. Our project will be focused on conducting an in-depth investigation within the domain of knowledge distillation. The next section will elaborate on the various techniques used in distillation.

2 Background

2.1 Related Work

Utilizing a model to teach another model was discussed initially for ensemble methods [5] as well as in semi-supervised learning paradigm using unlabeled data [6], however, it was formalized as knowledge distillation by Hinton et al. [7] in a teacher-student paradigm. The success of the technique to teach a single small model with comparable results to large model and ensemble specialized models accelerated the research despite the limited theoretical understanding of the matter [8]. Quickly knowledge distillation techniques expanded to different learning configurations, such as teacher-student learning, mutual learning, assistance teaching, life-long learning, self-learning. Data distillation which involves obtaining smaller and comparably effective datasets to reduce training load has also been explored [9]. Summarizing the wide range of the techniques used in distillation, a general KD framework can be developed, comprising three components: knowledge, distillation algorithm, and teacher-student architecture [8].

The type of knowledge to be distilled was a matter of research. While Hinton's paper counted on matching the logits between the teacher and student, diverse types of knowledge were explored such as internal weights [10], internal features [11, 12], gradients (usually via attention maps) [13], sparsity patterns [14], and relational information between input and output [15, 16] [17]. The different approaches to address knowledge might be classified into feature-based, response-based, and relational-based knowledge. Despite the increased understanding of knowledge, unifying the knowledge in a framework is a challenging task as they have interconnecting effects on each other [8].

As early distillation settings counted on offline learning scheme, offline configuration usually ignored the phase of training the teacher model and considered this part an inevitable paid overhead before distillation. Such drawbacks encouraged researchers to address online learning methods where both teacher and student are updated simultaneously [18]. In such setting multiple neural nets work cooperatively where any one of them can be the student model [8]. Therefore, the parallel computing opportunity could reduce the learning time for teacher and student, but mutual learning techniques were significantly limited with high-capacity teacher settings. Therefore, a third configuration was explored: distilling knowledge within the same model, which was named self-learning [19]. It has proposed a training setting where knowledge is distilled from deeper layers to shallow layers in the same model during training. Combinations between the schemes are applicable within the knowledge transfer framework [8].

Despite theoretical work of Ba and Caruana, 2014, which suggests that shallow networks can learn the same representation learnt by deep neural nets [20]. Experiments have shown that learning limits were usually achieved because of the capacity gap between the teacher model and the student mode [8][19]. This contradiction reflects the subtlety of knowledge distillation setting design on the level of student architecture and learning algorithm used. From this conclusion, our project aims to design a knowledge extraction-transfer framework and test the framework to distill a large pre-trained teacher model.

2.2 Preliminaries

In this section, we describe the formulation of our project. The experiments examine the effects of different knowledge distillation techniques building upon the model that has achieved the SOTA performance in the domain of applications of KD on LLM, named TinyBERT [22].

2.2.1 Structure of TinyBERT

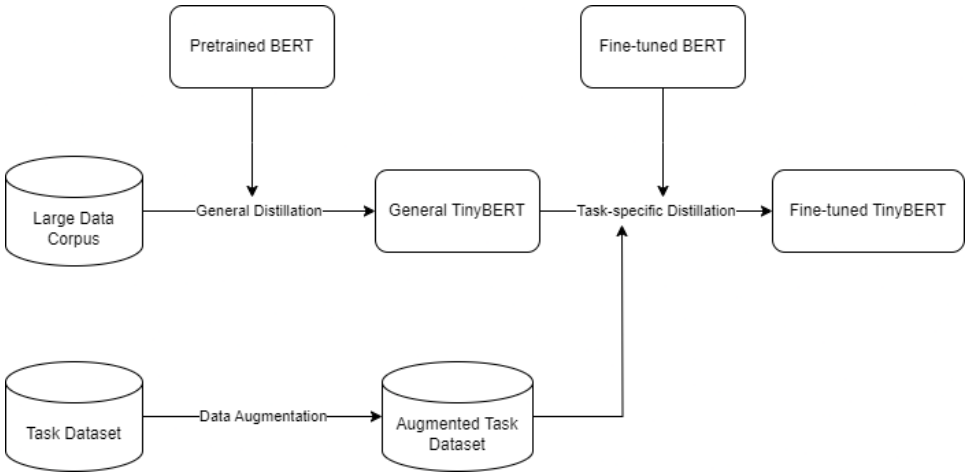

Figure 1: Illustration of TinyBERT Learning

TinyBERT introduced a novel distillation method for Transformer-based models, specifically involving applying knowledge distillation techniques to different parts of the model. Transformer-layer distillation can be further specified in 2 components: attention-based distillation which encourages the student model to match the multi-head attention in the teacher network to facilitate the transfer of linguistic knowledge from the teacher model, while on the other hand, the hidden states-based distillation is attempting to match the hidden states of student and teacher networks. For the embedding-layer distillation, TinyBERT utilized similar techniques in the hidden states-based distillation. Lastly, the prediction-layer distillation aims to match the logits between the student and teacher networks at the prediction layer.

2.3 Our Experiments

As a result of unforeseen delays in securing computational resources, we have had to deviate from our original plan of fully replicating TinyBERT and conducting a comprehensive evaluation on various downstream tasks to facilitate the comparison of effectiveness between various knowledge distillation techniques through contrasting the performance impacts. Instead, to accommodate the need to gain insights within a condensed timeframe and with limited computing resources, we have revised our goals into a less ambitious ones to better align with the situation at hand while hopefully still being able to gain the same valuable insight. With the renewed targets in mind, we have reconstructed experiments to focus on the general distillation stage of the two-stage learning framework with a reduced dataset and used the magnitude of the losses as a proxy measure of the effectiveness of the methods.

2.3.1 Dataset

In the general distillation, we set the maximum sequence length to 128 and used a subsample of English Wikipedia as the text corpus. The dataset was tokenized twice using BERT tokenizer and TinyBERT tokenizer while the identicality of the two tokenized sequences was asserted.

2.3.1 Settings

The conducted experiments aimed to compare the quality of the distillation process between two different configurations; the first one is the distillation setting presented by Hinton et al. [7] based on logit loss difference, which will be referenced in the rest of the report as logit loss distillation; and the second setting is by calculating four different losses through the different hidden state representations. The four losses in the second setting are Embedding loss, Hidden loss, Attention Loss, and Logit loss. This setting will be mentioned in the rest of the report in the name of multi-Loss distillation mechanism. Figure 2 depicts the different losses in the inference phase.

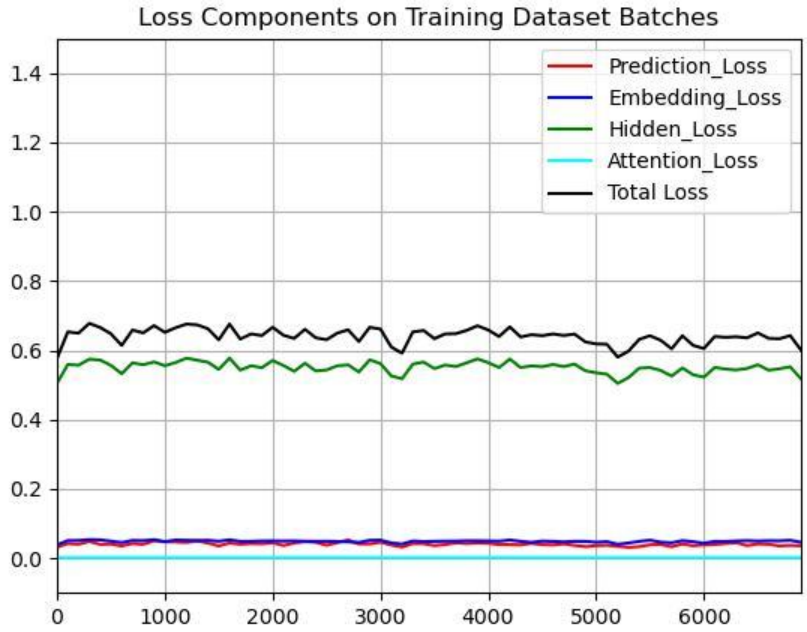

Figure 2: Loss Components on Training Dataset Batches

The general architecture of the two distillation settings included passing the same data through TinyBERT as a student model and BERT as a teacher model and extract the representation of the same data between the two encoders and calculating the loss between these representations. The two distillation settings were instantiated from the same initial parameters for the student model. Forward hooks were used on both models to read the hidden representations in the forward pass. Due to the difference in the dimension of the latent space between BERT (768) and TinyBERT (312), a set of linear layers (with no activation functions) were used to scale up the hidden representations of TinyBERT to match the size of BERT's hidden representations.

The difference between the two distillation techniques was quantified using the loss of the logit layer on both training dataset and test dataset as it will be depicted and discussed in the next section. In addition, the average gradient through the different layers was evaluated through training phase to evaluate the quality of learning process. As gradient decays in the earlier layers with the increase of the model's depth, comparing the gradient average between the two distillation schemes with the same initialized model can indicate the propagation of information across the different layers.

3 Main Result

2.3.1 Loss Comparison

Despite the two instances having a similar loss drop in the early phase of the training, the multi-loss training scheme depicted a lower loss while inference on both training and testing datasets. (Figures 3 and 4). The fact that the logit loss is lower in the multi-loss model is an interesting pattern. As in these settings, the logit-loss scheme has a higher potential to overfit the logit layer of the teacher in comparison to the multi-loss scheme. Therefore, a better performance on training data was expected from the logit-loss model. However, for the data on which the models were trained, a consistent lower loss of the multi-loss scheme was remarked without losing the

generality of the model.

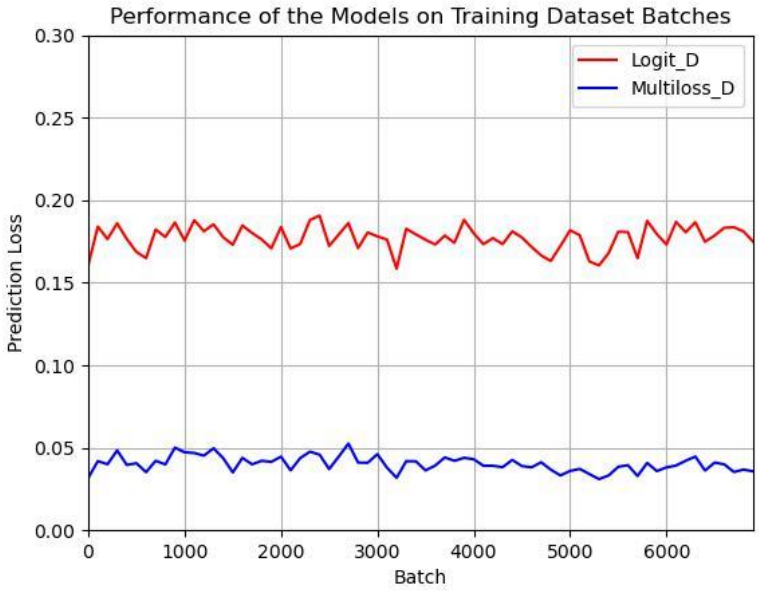

Figure 3: Performance of the Models on Training Dataset Batches

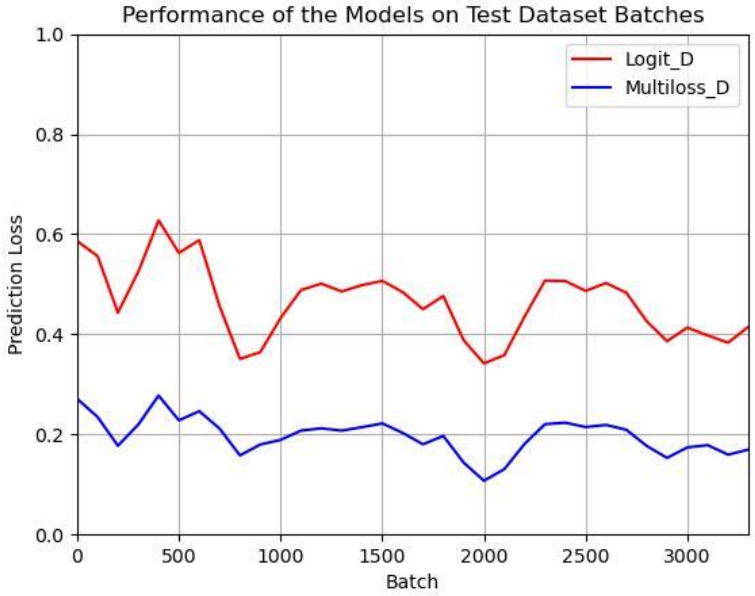

Figure 4: Performance of the Models on Test Dataset Batches

3.2 Gradient Norm Comparison

In order to analyze the impact of the distillation scheme on the training process, gradient norm tracking through the training phase might be an indicator on the quality of learning and information propagation through layers. As Figure 5 depicts, average gradient norm is higher on the multi-loss setting vs logit loss scheme. Given (Figure 3) which describes the different components of the loss, it reflects that the hidden loss is the highest loss among the four losses, we

can conclude that the hidden representation loss was responsible to maintain the training phase dynamic with a relatively high gradient. In contrast, the decay in the average gradient norm in logit-loss settings might indicate the overfitting of the last layers to the logits of the teacher with a limited gradient propagates to train the earlier layers.

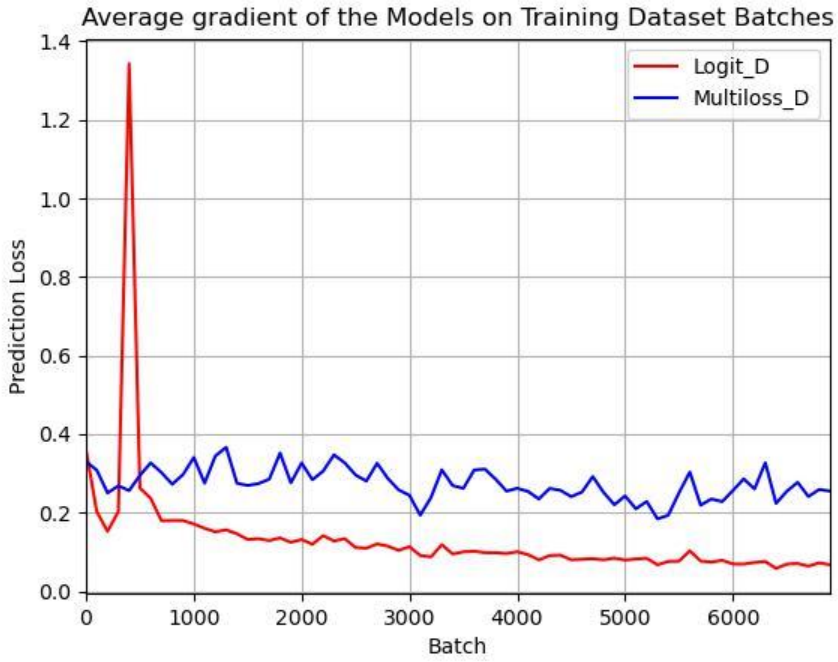

Figure 5: Average gradient Norm of the Models while Training batches

Therefore, given the higher average gradient norm and the lower loss, it can be concluded that the multi-loss distillation scheme outperformed the logit-loss distillation scheme. As it accelerates gradient propagation to the earlier layers of the models, in addition, it is relatively safer since it does not overfit the logits of the teacher if the training is conducted over only a subset of the data used to train the teacher.

4 Conclusion

In conclusion, Knowledge Distillation is a powerful technique to train models via learning from models, either in a student-teacher scheme or in a cooperative learning scheme. The technique allows accelerating the training process and training and developing smaller powerful models.

In this report we examined two distillation settings, multi-loss distillation scheme and Logit-Loss distillation scheme. A comparison was made between the two settings in training phase using average-norm-gradient, and in inference phase using prediction loss on training and testing dataset. It was remarked that the multi-loss scheme maintains the dynamicity of the training process via a higher gradient because of the multiple hidden losses along the teacher model. On the other hand, the logit-loss scheme exhibited a gradual decay in the gradient norm, which might be interpreted as a difficulty in data propagation to the earlier layers. The multi-loss scheme exhibited a lower loss on both training and testing datasets.

It is recommended for future work to expand the results to compare the performance using distillation under different paradigms such as analyzing sparsity patterns [14] and the relationship between the inputs and outputs [15, 16]. Further analysis to understand why this technique works and if it converges to the same model to which it will converge solely by enough data is a valid

promising direction. In addition, combining distillation with other compression techniques (such as quantization) is a third direction to understand the dynamic limitations by which will be imposed by each method.

References

[1] L. Deng, G. Li, S. Han, L. Shi and Y. Xie, "Model Compression and Hardware Acceleration for Neural Networks: A Comprehensive Survey," in Proceedings of the IEEE, vol. 108, no. 4, pp. 485-532, April 2020, doi: 10.1109/JPROC.2020.2976475.

[2] Cheng, Yu & Wang, Duo & Zhou, Pan & Zhang, Tao. (2017). A Survey of Model Compression and Acceleration for Deep Neural Networks.

[3] K. He, “Deep Residual Learning for Image Recognition,” arXiv.org, Dec. 10, 2015. https://doi.org/10.48550/arXiv.1512.03385

[4] Choudhary, T. et al. (2020) A comprehensive survey on model compression and Acceleration -Artificial Intelligence Review, SpringerLink. Available at: https://link.springer.com/article/10.1007/s10462-020-09816-7 (Accessed: 01 October 2023).

[5] Buciluǎ, Cristian, et al. “Model Compression.” Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2006, pp. 535–41. DOI.org (Crossref), https://doi.org/10.1145/1150401150464.

[6] Urner, R., Shalev-Shwartz, S., & Ben-David, S. (2011). Access to unlabeled data can speed up prediction time. In Proceedings of the 28th International Conference on Machine Learning (ICML-11) (pp. 641-648).

[7] Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531.

[8] Gou, J., Yu, B., Maybank, S. J., & Tao, D. (2021). Knowledge distillation: A survey. International Journal of Computer Vision, 129, 1789-1819.

[9] Wang, T., Zhu, J. Y., Torralba, A., & Efros, A. A. (2018). Dataset distillation. arXiv preprint arXiv:18114959.

[10] Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., & Bengio, Y. (2014). Fitnets: Hints for thin deep nets. arXiv preprint arXiv:1412.6550.

[11] Huang, Z., & Wang, N. (2017). Like what you like: Knowledge distill via neuron selectivity transfer. arXiv preprint arXiv:1707.01219.

[12] Kim, J., Park, S., & Kwak, N. (2018). Paraphrasing complex network: Network compression via factor transfer. Advances in neural information processing systems, 31.

[13] Zagoruyko, S., & Komodakis, N. (2016). Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. arXiv preprint arXiv:1612.03928.

[14] Heo, B., Lee, M., Yun, S., & Choi, J. Y. (2019, July). Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 33, No. 01, pp. 3779-3787).

[15] Yim, J., Joo, D., Bae, J., & Kim, J. (2017). A gift from knowledge distillation: Fast optimization,

network minimization and transfer learning. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4133-4141).

[16] Park, W., Kim, D., Lu, Y., & Cho, M. (2019). Relational knowledge distillation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 3967-3976).

[17] Lab, M. H. (2022). Lecture 10 - Knowledge distillation | MIT 6.S965 [Video]. In YouTube. https://www.youtube.com/watch?v=tT9Lnt6stwA&t=93s

[18] Zhang, Y., Xiang, T., Hospedales, T. M., & Lu, H. (2018). Deep mutual learning. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4320-4328).

[19] Zhang, L., Song, J., Gao, A., Chen, J., Bao, C., & Ma, K. (2019). Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 3713-3722).

[20] Ba, J., & Caruana, R. (2014). Do deep nets really need to be deep?. Advances in neural information processing systems, 27.

[21] Peter West, Chandra Bhagavatula, Jack Hessel, Jena D. Hwang, Liwei Jiang, Ronan Le Bras, Ximing Lu, Sean Welleck, and Yejin Choi. 2022. Symbolic Knowledge Distillation: from General Language Models to Commonsense Models.

[22] Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., & Liu, Q. (2020). TinyBERT: Distilling Bert for Natural language understanding. Findings of the Association for Computational Linguistics: EMNLP 2020. https://doi.org/10.18653/v1/2020.findings-emnlp.372