LLM4All

Low Cost LLM

Iakovos Evdaimon, Yanzhu Guo, Michalis Vazirgiannis

École Polytechnique

October 11, 2023

Contents

Introduction

Introduction - LIX contributions so far

- BERTweetFR [1], a LLM for French tweets
- JuriBERT [2], a model for French legal text
- SUMM'RE project on meeting summarisation [3]
- BARThez [4], a 165M parameters LLM for French that excels at generative tasks
- AraBART [5], a 139M monolingual pre-trained model for Arabic language
- GreekBART [6], a 181M parameters LLM for Greek language
- FrugalScore [7]: A data-free Knowledge Distillation approach for NLG evaluation metrics (retaining 96.8% of the performance, running 24 times faster, and having 35 times less parameters than the original metrics).

Issues with LLMs I

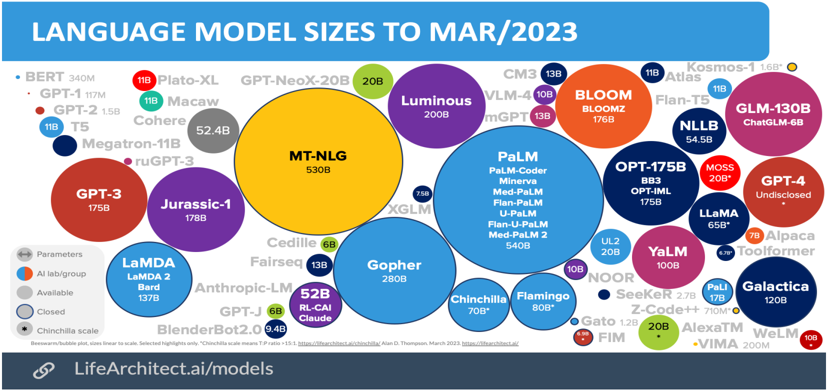

Figure: Overview of popular LLMs in terms of their parameters' number

Issues with LLMs II

- i.e., inference on BLOOM-176B, need 8x80GB A100 GPUs (\$15k each).
- fine-tune BLOOM-176B, need 72 of these GPUs![1]
- Need to reduce these requirements while preserving the model's performance.
- Possible solutions: *knowledge distillation*, *quantization*

[1] `https://cloud.google.com/tpu/docs/bfloat16`

Knowledge Distillation

Knowledge Distillation I

- A small model (the student) is trained to mimic the predictions of a much larger pre-trained model (the teacher) [8]–[11]

- In distillation, knowledge is transferred from teacher model to the student by minimizing a loss function
- Target: student learns the distribution of class probabilities of the teacher model.

Knowledge Distillation II

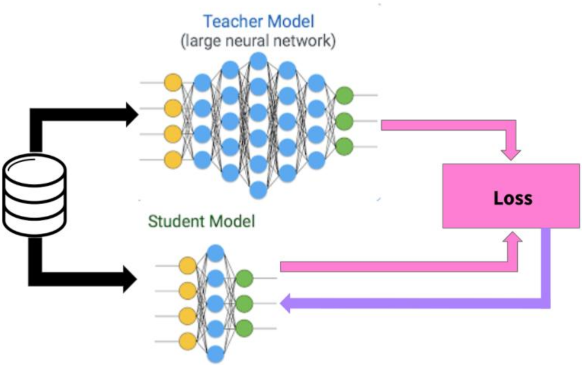

Figure: A simplified representation of Knowledge Distillation framework (Source: `https://towardsdatascience.com/knowledge-distillation-simplified-dd4973dbc764`)

Why Knowledge Distillation?

- Faster Inference and Lower Latency. Distilled models allow for quick decision-making, enhancing user experience.
- Distilled models often generalize better to unseen data due to the regularization effect of distillation [12].
- Improved Performance on Small Devices with limited computational resources. Knowledge distillation enables IoT devices to perform complex tasks.
- Reducing the carbon footprint

Types of Knowledge I

Response-Based Knowledge

- Refers to the neural response of the last output layer of the teacher model.
- The main idea is to directly mimic the final prediction of the teacher model.

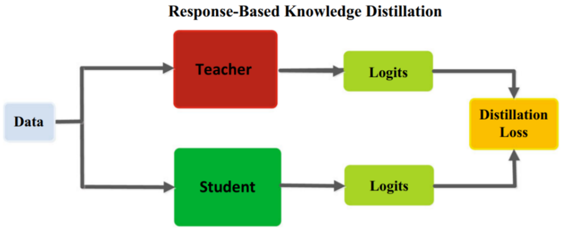

Figure: Representation of Response-Based Knowledge

Types of Knowledge II

Feature-Based Knowledge

- The main idea is to directly match the feature activations of the teacher and the student.

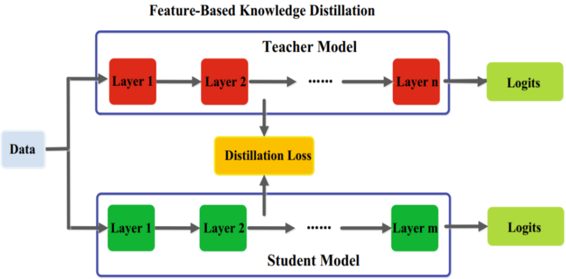

Figure: Representation of Feature-Based Knowledge

Types of Knowledge III

Relation-Based Knowledge

- Both response-based and feature-based knowledge use the outputs of specific layers in the teacher model.
- Relation-based knowledge further explores the relationships between different layers or data samples.

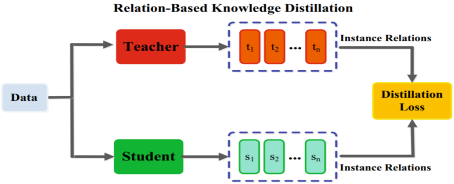

Figure: Representation of Relation-Based Knowledge

Training Process Overview

- Knowledge Transfer: The student model learns from the teacher's outputs, aiming to replicate its behavior.
- Fine-tuning: After knowledge transfer, the student model is fine-tuned on the task-specific dataset.

Knowledge Distillation Techniques I

- **Offline Distillation**: The knowledge is transferred from a pre-trained teacher model into a student model.
  - The teacher is pre-trained before the distillation.
  - The knowledge is extracted from the teacher model in the forms of logits or intermediate features, which are then used to guide the training of the student model.
- **Online Distillation**: Both the teacher and the student are updated simultaneously and the whole knowledge distillation framework is end-to-end trainable.
- **Self-Distillation**: The same networks are used for the teacher and the student models. One way to apply self-distillation is to transfer knowledge from the deeper sections of the network into its shallow sections.

Knowledge Distillation Techniques II

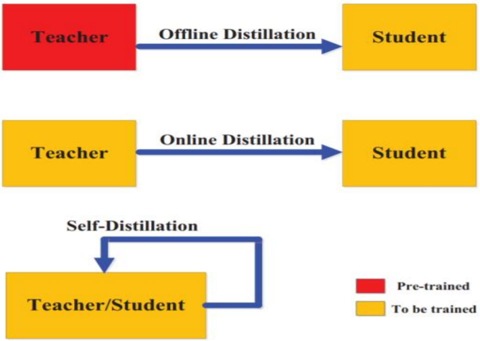

Figure: Knowledge Distillation Methods [13]

State of the Art

- DistilBERT [14]
- Distilled GPT-3.5 for source code summarization [15]
- MiniLLM [16]

DistilBERT

- A 40% smaller BERT model pre-trained leveraging knowledge distillation via the supervision of BERT-*base* model.
- Retaining 97% of BERT-*base* model's language understanding capabilities and being 60% faster.

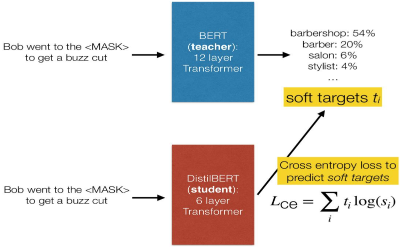

Figure: Illustration of knowledge distillation during the pre-training of DistilBERT

Distilled GPT

- A distilled GPT-3.5 model for a specific task, the source code summarization.
- The model is small enough (350M parameters, so 0.2% of the GPT-3.5 in terms of parameters number) to be run on a single 16 GB GPU (can be run locally).
- Capable of mimicking GPT-3.5 on this task. Based on the conducted human

  evaluation a slight preference favoring GPT-3.5 in direct comparisons by participants (52% GPT-3.5, 46% distilled, 2% undecided)

MiniLLM I

- Support for multiple LLMs (currently LLAMA, BLOOM, OPT) at various model sizes (up to 170B parameters)
- It distills smaller language models from generative larger language models
- It generates precise responses with high overall quality
- Scalable for different model families with 120M to 13B parameters
- A 50% smaller model, in terms of parameters, achieves similar or sometimes better performance than the teacher model.
- In the case of 90% fewer parameters, it exhibits inferior performance than the teacher model but is still comparable.

MiniLLM II

Model	#Params	Method	DollyEval		SelfInst		VicunaEval		S-NI	UnNI
			GPT4	R-L	GPT4	R-L	GPT4	R-L	R-L	R-L
GPT2	1.5B	Teacher	58.4	27.6	42.9	14.3	48.6	16.3	27.6	34.9
	120M	MiniLLM	44.7	24.6	29.2	13.2	34.1	16.9	25.3	30.1
	340M	MiniLLM	52.2	25.4	40.5	15.6	42.6	17.7	27.4	34.5
	760M	MiniLLM	54.7	26.4	44.6	15.9	45.7	18.3	29.3	37.7
OPT	13B	Teacher	70.3	29.2	56.1	18.4	58.0	17.8	30.4	36.1
	1.3B	MiniLLM	60.7	26.7	47.0	14.8	50.6	17.9	28.6	33.4
	2.7B	MiniLLM	63.2	27.4	52.7	17.2	55.9	19.1	30.7	35.1
	6.7B	MiniLLM	70.8	29.0	58.5	17.5	60.1	18.7	32.5	36.7
LLaMA	13B	Teacher	79.0	29.7	75.5	23.4	65.1	19.4	35.8	38.5
	7B	MiniLLM	76.4	29.0	73.1	23.2	64.1	20.7	35.5	40.2

Table: Evaluation results. GPT4 and R-L stand for the average GPT-4 feedback scores and Rouge-L scores across 5 random seeds (Source [16])

Quantization

Quantization

The number of digits allowed to be used in the *mantissa* governs the **precision** of the value, the *exponent* governs the **range**, e.g., 6.02×10^{23} v.s. $6.022140857 \times 10^{23}$.

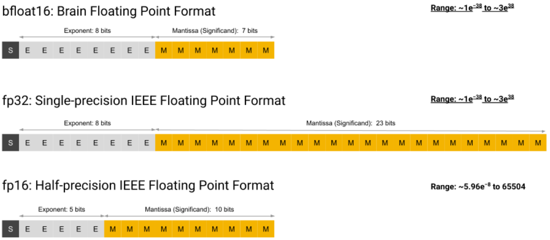

Figure: floating point formats: `bfloat16` (used by BLOOM-176B), `float16` (used by BLOOM-7.1B) and `float32`[3].

Replacing `float32` with `bfloat16` can shorten the time, and use less memory while preserving the accuracy[4] (models are more sensitive to changes in exponent rather than mantissa).

[2] https://cloud.google.com/tpu/docs/bfloat16
[3] https://cloud.google.com/tpu/docs/bfloat16
[4] https://www.cerebras.net/blog/to-bfloat-or-not-to-bfloat-that-is-the-question/

8-bit optimizer

- Stateful optimizers (e.g., Adam, AdamW, and Momentum) maintain gradient statistics over time to accelerate optimization $\Rightarrow$ these optimizer states take 33-75% of the total memory footprint during training!
- For 32-bit states, Adam consumes 8 bytes per parameter. That is 8 GB for a 1B parameter model. 8-bit quantization reduces the cost to 2 GB.

Examples

https://github.com/TimDettmers/bitsandbytes
https://github.com/IST-DASLab/gptq

Model	Inference memory	Fine-tuning memory
T5-11B	22 GB	176 GB
OPT-66B	132 GB	1,056 GB
BLOOM 176B	352 GB	2,800 GB

LLM.int8() 8-bit optimizers

Model	Inference memory	Fine-tuning memory
T5-11B	11 GB	66 GB
OPT-66B	66 GB	396 GB
BLOOM 176B	176 GB	1,056 GB

Figure: VRAM reduction of 8-bit quantization[5].

[5] https://www.youtube.com/watch?v=jyOqtw4ry2w

Moving Towards Data-Centric NLP

Moving Towards Data-Centric NLP

- Enhancing data quality offers an alternative approach to scaling up.
- In low-cost LLMs with limited parameters and resources, high-quality data becomes even more crucial to compensate for model constraints.
- Related work: *"Questioning the Validity of Summarization Datasets and Improving Their Factual Consistency" in EMNLP 2022* [17].

  By filtering out unfactual samples from popular summarization datasets, we improve the performance of abstractive summarization models while reducing training time and lowering the need for computational resources.

Embracing Continual Learning

Embracing Continual Learning

- Data are dynamic and evolve over time; keeping LLMs updated is essential to maintain their relevance and effectiveness.
- Continuously updating models with new data is more cost efficient than retraining from scratch.
- Risks and Challenges: **the Curse of Recursion** [18]

  Typically, training data is sourced from the Internet, which is increasingly populated with machine-generated content. Recursively training LLMs on such data can potentially result in language deterioration and linguistic diversity decrease.

Conclusion

Conclusion

- Large Language Models (LLMs) demand significant resources, leading to high costs in terms of space, GPU usage, and time consumption.
- LLMs also have a substantial energy footprint, contributing to environmental concerns.
- There is a crucial need to produce smaller Language Models that balance resource efficiency and performance.
- Methods for achieving this include:
  1. Knowledge Distillation (KD): Transferring knowledge from large models to smaller ones.
  2. Quantization: Reducing the precision of model weights, saving memory.
  3. Data-Centric NLP.
  4. Continual Learning

References I

[1] Y. Guo, V. Rennard, C. Xypolopoulos, and M. Vazirgiannis, "Bertweetfr: Domain adaptation of pre-trained language models for french tweets," in *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, 2021, pp. 445–450.

[2] S. Douka, H. Abdine, M. Vazirgiannis, R. El Hamdani, and D. R. Amariles, "Juribert: A masked-language model adaptation for french legal text," in *Proceedings of the Natural Legal Language Processing Workshop 2021*, 2021, pp. 95–101.

[3] V. Rennard, G. Shang, J. Hunter, and M. Vazirgiannis, "Abstractive Meeting Summarization: A Survey," *Transactions of the Association for Computational Linguistics*, vol. 11, pp. 861–884, Jul. 2023, ISSN: 2307-387X. DOI: 10.1162/tacl_a_00578. eprint: https://direct.mit.edu/tacl/article-pdf/doi/10.1162/tacl_a_00578/2150534/tacl_a_00578.pdf. [Online]. Available: https://doi.org/10.1162/tacl%5C_a%5C_00578.

References II

[4] M. K. Eddine, A. Tixier, and M. Vazirgiannis, "Barthez: A skilled pretrained french sequence-to-sequence model," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 9369–9390.

[5] M. K. Eddine, N. Tomeh, N. Habash, J. Le Roux, and M. Vazirgiannis, "Arabart: A pretrained arabic sequence-to-sequence model for abstractive summarization," *WANLP 2022*, p. 31, 2022.

[6] I. Evdaimon, H. Abdine, C. Xypolopoulos, S. Outsios, M. Vazirgiannis, and G. Stamou, "Greekbart: The first pretrained greek sequence-to-sequence model," *arXiv preprint arXiv:2304.00869*, 2023.

[7] M. K. Eddine, G. Shang, A. Tixier, and M. Vazirgiannis, "Frugalscore: Learning cheaper, lighter and faster evaluation metrics for automatic text generation," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 1305–1318.

References III

[8] G. Hinton, O. Vinyals, and J. Dean, *Distilling the knowledge in a neural network*, 2015. arXiv: 1503.02531 [stat.ML].

[9] J. Ba and R. Caruana, "Do deep nets really need to be deep?" In *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., vol. 27, Curran Associates, Inc., 2014. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2014/file/ea8fcd92d59581717e06eb187f10666d-Paper.pdf.

[10] A. Kuncoro, M. Ballesteros, L. Kong, C. Dyer, and N. A. Smith, "Distilling an ensemble of greedy dependency parsers into one MST parser," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 1744–1753. DOI: 10.18653/v1/D16-1180. [Online]. Available: https://aclanthology.org/D16-1180.

References IV

[11] C. Bucila, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, August 20-23, 2006*, T. Eliassi-Rad, L. H. Ungar, M. Craven, and D. Gunopulos, Eds., ACM, 2006, pp. 535–541. DOI: 10.1145/1150402.1150464. [Online]. Available: https://doi.org/10.1145/1150402.1150464.

[12] S. Stanton, P. Izmailov, P. Kirichenko, A. A. Alemi, and A. G. Wilson, "Does knowledge distillation really work?" In *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34, Curran Associates, Inc., 2021, pp. 6906–6919. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/376c6b9ff3bedbbea56751a84fffc10c-Paper.pdf.

[13] B. Yu, *Efficient computing of deep neural networks - knowledge distillation*, https://www.cse.cuhk.edu.hk/~byu/CMSC5743/2021Fall/slides/Lec10-KD.pdf, 2021.

References V

[14] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, *Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter*, 2020. arXiv: 1910.01108 [cs.CL].

[15] C.-Y. Su and C. McMillan, *Distilled gpt for source code summarization*, 2023. arXiv: 2308.14731 [cs.SE].

[16] Y. Gu, L. Dong, F. Wei, and M. Huang, *Knowledge distillation of large language models*, 2023. arXiv: 2306.08543 [cs.CL].

[17] Y. Guo, C. Clavel, M. K. Eddine, and M. Vazirgiannis, "Questioning the validity of summarization datasets and improving their factual consistency," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022, pp. 5716–5727.

[18] I. Shumailov, Z. Shumaylov, Y. Zhao, Y. Gal, N. Papernot, and R. Anderson, "The curse of recursion: Training on generated data makes models forget," *arXiv preprint arxiv:2305.17493*, 2023.