Large Language Model Lifecycle Tips

Start-to-finish LLMs

Shaan Khosla

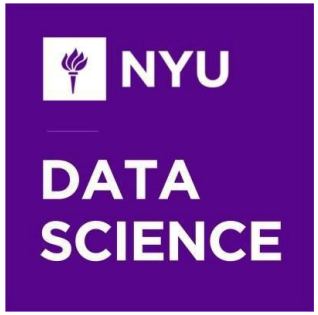
Data Scientist, Researcher, Student

Pearson

My name is **Shaan Khosla**

shaan-khosla/

- Data Scientist at Nebula
- Current Data Science masters student at NYU
- Various research with LLMs, topic models, and legal NLP
- Writer of *Let's Talk Text* newsletter



Pearson

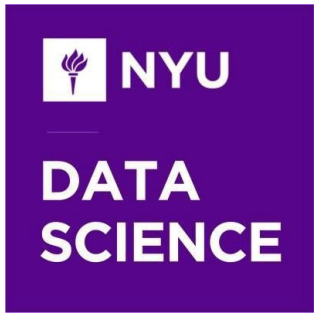
Resources

Check out the slides and code!


github.com/shaankhosla/catalyst

Add me on LinkedIn


shaan-khosla/



Pearson

No Escaping ChatGPT

I am giving a 30 minute presentation where I will cover practical LLM tips over the full NLP lifecycle. These include topics such as efficient training practices, validation methods, and productionization considerations to ensure your design is optimized for implementation within your real-world natural-language application. What are some of the key topics I should discuss?

1. Efficient training practices: Discuss techniques for reducing the number of training samples needed, such as transfer learning and data augmentation. Also, cover the use of hardware accelerators, such as GPUs and TPUs, to speed up training time.
2. Validation methods: Explain the importance of using a variety of evaluation metrics, such as accuracy, F1-score, and perplexity, to ensure your model is performing well on different aspects of the task. Also, cover the use of techniques such as cross-validation and hold-out sets to prevent overfitting.
3. Productionization considerations: Discuss the importance of model optimization for deployment, including techniques for reducing the model size, as well as the trade-offs between model accuracy and inference speed.
4. Handling large datasets: Discuss the importance of distributed training and data parallelism in handling large datasets
5. Handling real-world NLP applications: Discuss the importance of handling rare and out-of-vocabulary words, as well as handling variations in text input, such as typos and slang.
6. Monitoring and maintenance: Discuss the importance of monitoring the model's performance in production and the importance of regular retraining and updating the model to improve its performance.
7. Explain the importance of considering ethical and bias implications of the model in the real world.

- Data augmentation
- Training
- Productionization considerations
- Evaluation
- Ethical considerations
- Explainability
- Monitoring

Data Augmentation Using LLMs

- Back translation with translation models

- Synonym replacement using [MASK]

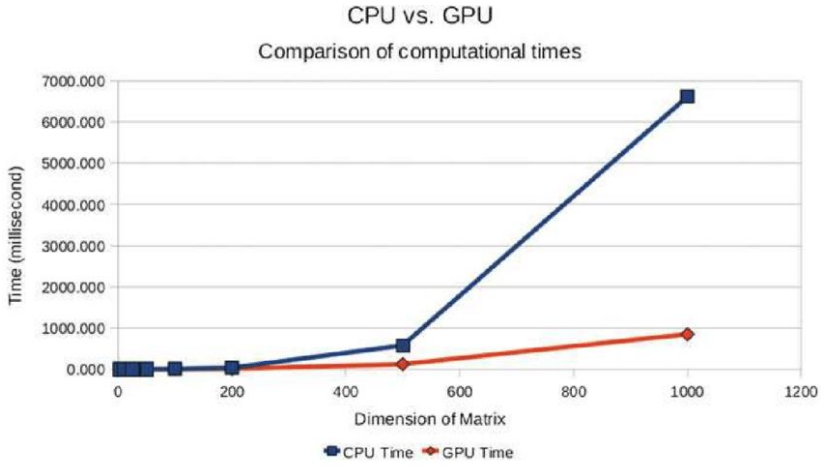
- Text generation with generative models

If only to avoid making this type of film in the future. This film is interesting as an experiment but tells no cogent plot or any other interesting story.

Overall, this is an entertaining film but is in most cases not good.

Training Setup

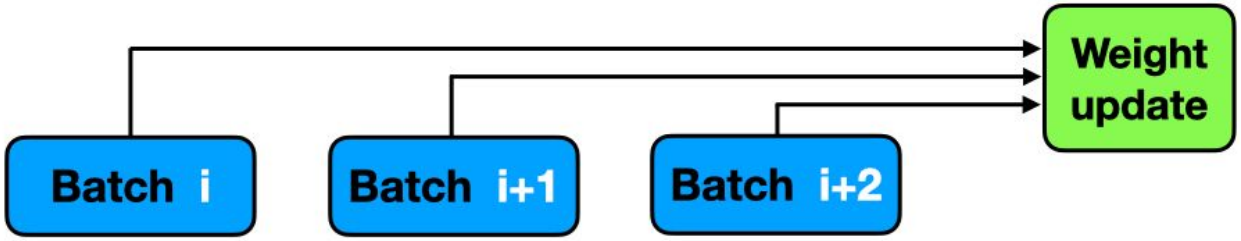
- 25k movie positive and negative movie reviews
- HuggingFace Trainer
- Matrix multiplication chips
  - GPUs
  - TPUs



Pearson

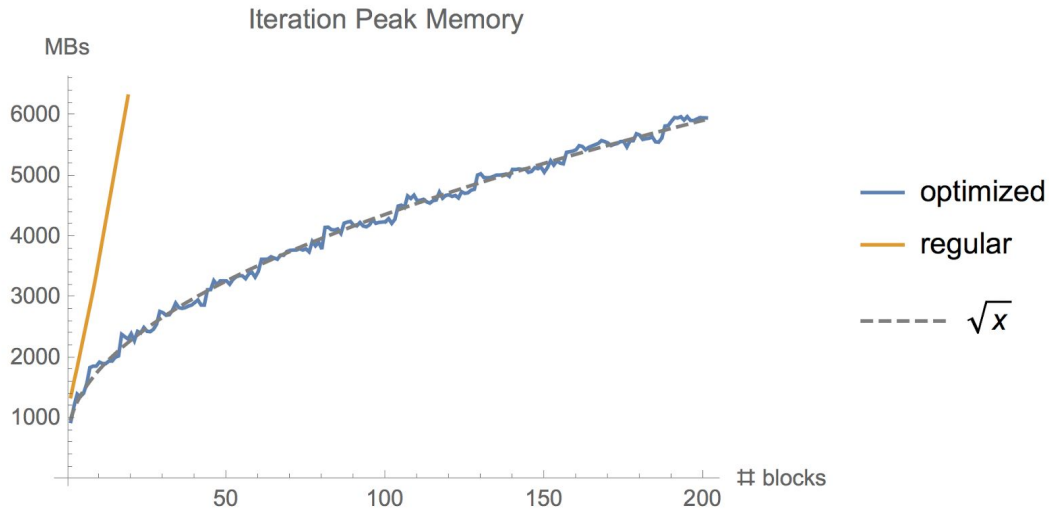
Gradient Accumulation

- Batch size is limited by GPU memory
- Solution
  - Accumulate gradients over N batches
  - Gradient accumulation with N=3 and batch size of 8 is equivalent to batch size of 24



Gradient Checkpointing

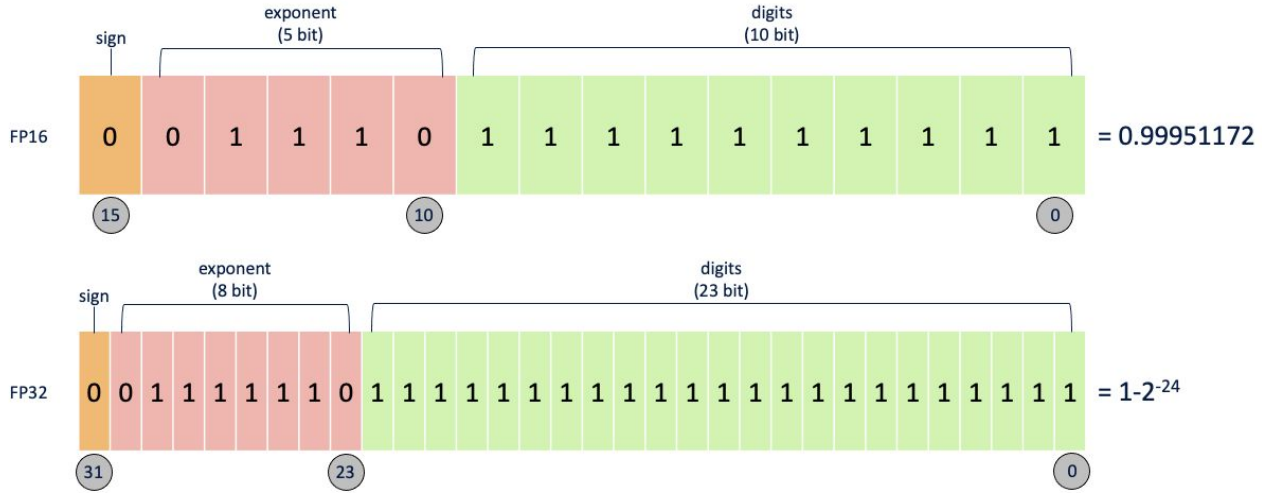
- Normally all the activations from the forward pass are saved to compute the gradients in the backward pass
- Recompute nodes during backpropagation



Pearson

Automatic Mixed Precision

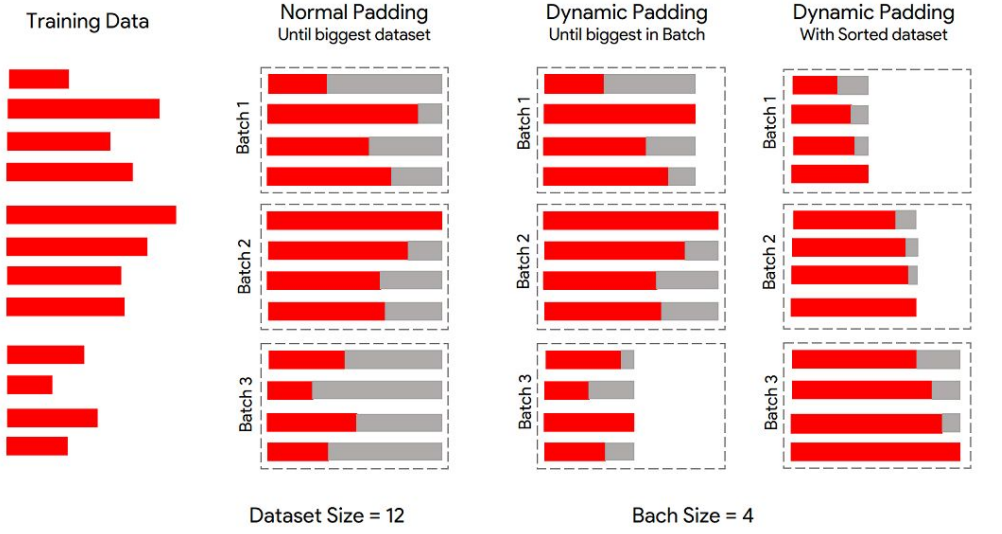
- 32 bits are used to store the weights, activations, and gradients
- Instead, use FP16
- Makes training faster and preserves memory



Pearson

Dynamic Padding & Uniform Length Batching

- All inputs in a batch need to be the same length
- Sequences are padded to the length of the maximum training sample
- [PAD] tokens are still included in all the operations



Pearson

Parallelism: Use more GPUs!

Data Parallelism

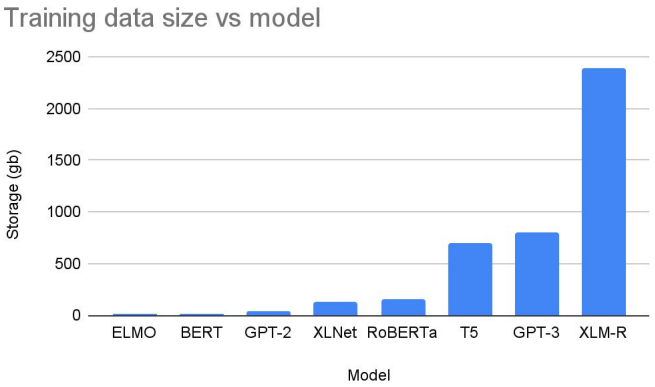
- Each GPU receives a small batch and computes the forward and backward passes
- Gradients are averaged across nodes

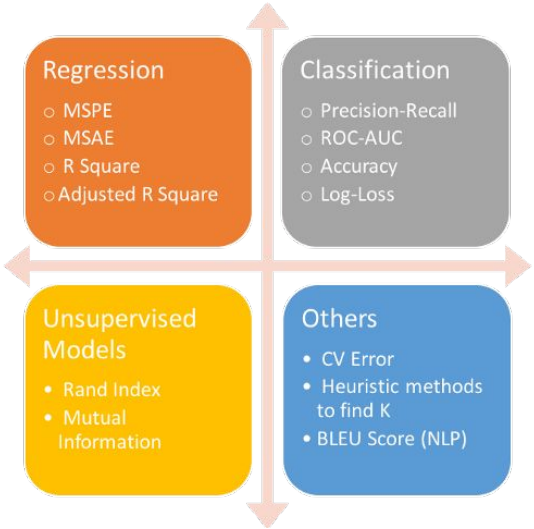
Model Parallelism

- But what if we can't fit even one copy of the model on a device?
  - GPT-3: 175B parameters
  - 175B $\times$ 4 $\approx$ 700GB
- Put different layers of the model on different devices

Evaluation

- Each ML system is different
- You usually don’t have enough labeled data
- Evaluation requires ingenuity and creativity
- Evaluation should reflect user behavior

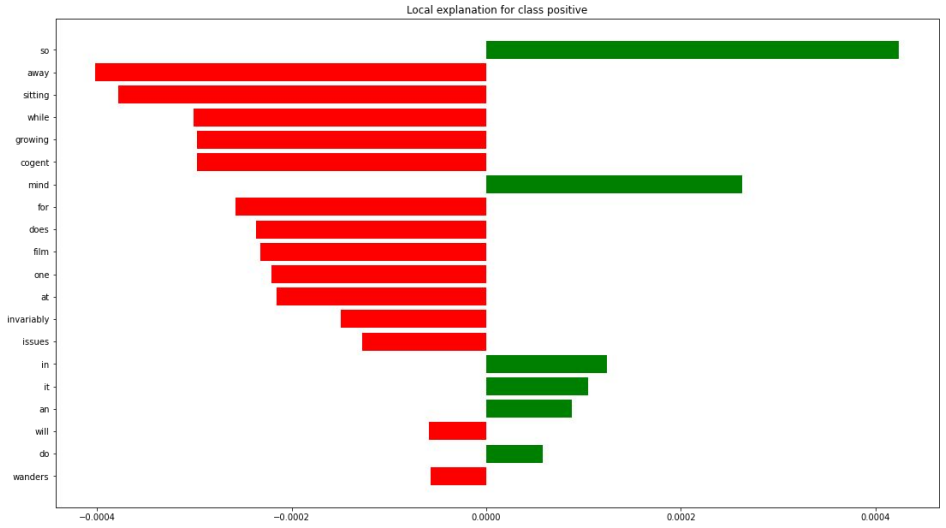


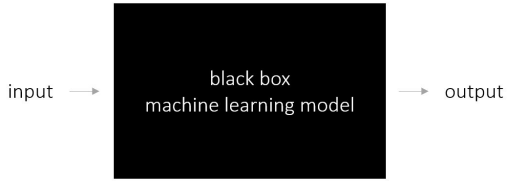


Pearson

Uncovering the Black Box

- Local interpretability
  - Why did the model make a certain prediction for an instance?
- Local Interpretable Model-Agnostic Explanations (LIME)
  - Slightly perturb the input

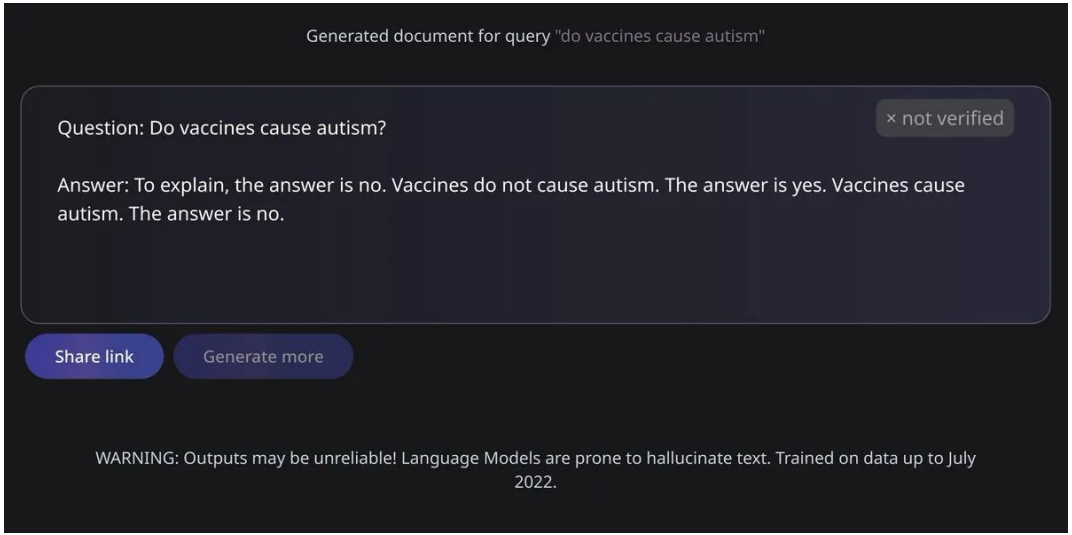




Pearson

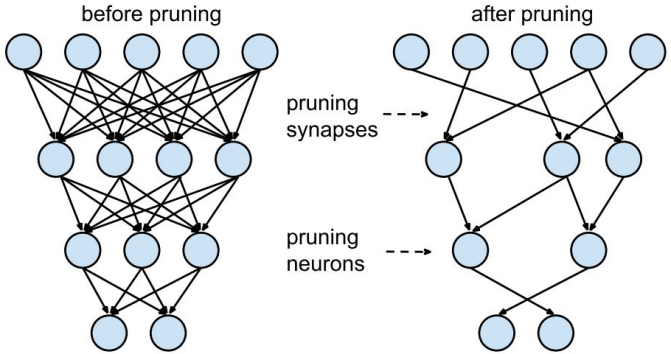
Ethics

- Models are very exposed to the public, and have ethical and moral considerations
- Selective generation
  - Moderation API



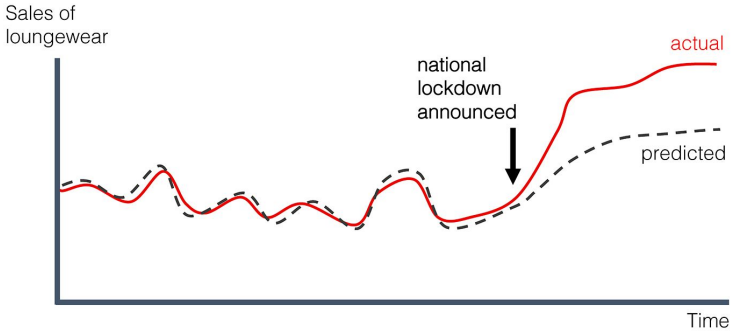
Considerations

- Deployment
  - Batch
  - Real-time
  - Edge
- SparseGPT
  - GPT family models can be pruned to at least 50%

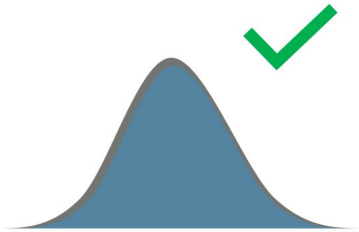


Monitoring - Drift Detection

- Types of drift
  - Data Drift
  - Label Drift
  - Prediction Drift
  - Concept Drift
- Detection algorithms
  - Kolmogorov-Smirnov test
  - Population Stability Index
  - Kullback-Leibler divergence
- Retraining



Input data **DRIFTED**



Predictions **STABLE**

Summary

- Simply **fine-tuning an LLM** using prepared datasets and HuggingFace's Trainer module can be **simple.**
- Thinking creatively about obtaining **labeled data, validating, productionizing, and monitoring** the model can be very **difficult**.

Works Cited

Pleshkova, Sn, and Al Bekiarski. "Development of Fast Parallel Algorithms Based on Visual and Audio Information in Motion Control Systems of Mobile Robots." *Computer Vision in Control Systems-4: Real Life Applications* (2018): 105-138.

https://kozodoi.me/python/deep%20learning/pytorch/tutorial/2021/02/19/gradient-accumulation.html

https://towardsdatascience.com/understanding-mixed-precision-training-4b246679c7c4

https://www.kaggle.com/code/sajjadayobi360/dynamic-padding-sortish-bathes

Huang, Yanping, Youlong Cheng, Ankur Bapna, Orhan Firat, Dehao Chen, Mia Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V. Le, and Yonghui Wu. "Gpipe: Efficient training of giant neural networks using pipeline parallelism." *Advances in neural information processing systems* 32 (2019).

Huang, Yanping, Youlong Cheng, Ankur Bapna, Orhan Firat, Dehao Chen, Mia Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V. Le, and Yonghui Wu. "Gpipe: Efficient training of giant neural networks using pipeline parallelism." *Advances in neural information processing systems* 32 (2019).

https://www.evidentlyai.com/blog/data-and-prediction-drift