# Large Language Models (LLM)

Christophe Troalen

March 2024

## Introduction

Please note that the explanations provided in this document are not always fully accurate or exhaustive. These notes were taken as quick references during my full-time role from October 2023 to October 2024. They are intended as an informal guide and should be cross-referenced with authoritative sources for precision.

## 1 Fundamental Papers on LLMs

Here is a list of papers that give a good introduction to LLMs:

- *Attention is All You Need* is the must-read paper from Vaswani et al. (Google), published in 06-2017. It presents the transformer architecture and the *self-attention mechanism.* This blog provides a nice overview of the architecture.

- *An Image is Worth 16x16 Words* from Dosovitskiy et al. (Google), 10-2020, applies the logic of transformers to images by enabling positional encoding on *patches.* These models are called *vision transformers* (ViT).

- *LlaMa 2: Open Foundation and Fine-Tuned Chat Models* from Touvron et al. (FAIR), 07-2023, focuses on helpfulness and safety alignment using reinforcement learning from human feedback (RLHF).

- *Training Compute-Optimal Large Language Models* from Hoffmann et al. (DeepMind), 03-2022, introduces the *Chinchilla law*, which provides training guidelines for LLMs (e.g., determining the data required given the model size).

- *DPO: Your Language Model is Secretly a Reward Model* from Stanford, 05-2023, presents an original alignment algorithm simpler than RLHF, as it doesn't require training a separate reward model. It is now widely used alongside fine-tuning methods to align models to *chosen responses* using datasets of chosen/rejected responses for a given input.

- *Neural Machine Translation by Jointly Learning to Align and Translate* from Bahdanau et al. (2016) addresses challenges in RNN architecture for translating longer sequences (context parameter), helping to understand the novelty of transformers.

## 2 Mixture of Experts

OpenAI's GPT-3.5 model, launched in March 2022, outperformed both open-source and proprietary models, likely leveraging a *Mixture of Experts* (MoE) architecture. Mistral attracted attention in December 2023 with the release of Mixtral-8x7B, an efficient open-source model using MoE architecture. Here are some important MoE papers:

- *Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer* (Google, 01-2017)
- *A Review of Sparse Expert Models in Deep Learning* (Google, 09-2022)
- *Megablocks: Efficient Sparse Training with Mixture of Experts* (Stanford, Microsoft, Google, 11-2022)
- *Mixture-of-Experts with Expert Choice Routing* (Google, 02-2022)
- *Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity* (Google, 01-2021)

Instead of relying solely on many experts (essentially FFN layers) during training or inference, another innovative method called *model merging* involves merging the weights of different models directly.

- Phixtral MoE (Maxime Labonne): mergekit
- MoE for clowns
- Huggingface collection of Model Merging
- Linear and Task Arithmetic Merging
- First MoE paper and Second MoE paper
- Google Doc on Model Merging

## 3 Speedup the Inference of LLMs

### 3.1 Speculative Decoding Techniques

Large Language Models (LLMs) are substantial, with small models requiring 7–8 billion parameters, medium models around 45–70 billion, and models like GPT-4 exceeding 1.7 trillion parameters. To optimize latency (the time taken to generate each token), speculative decoding techniques employ smaller draft models to predict outputs for larger target models.

- *MEDUSA: Simple LLM Inference Acceleration Framework with Multiple Decoding Heads* (TogetherAI, 01-2024)
- *EAGLE: Speculative Sampling Requires Rethinking Feature Uncertainty* (Microsoft, 01-2024)

### 3.2 KV Cache Optimization

For throughput optimization, KV cache techniques can reduce memory usage and improve efficiency.

- SGLang (RadixAttention): Introduces KV cache reuse for prompts with identical prefixes, improving efficiency.
- LightLLM: TokenAttention algorithm and Efficient Router scheduling
- *DeepSpeed-FastGen: High-throughput Text Generation for LLMs* (Microsoft, 01-2024)

Several frameworks for model deployment, including *vLLM, TensorRT, and DeepSpeed*, aim to boost LLM inference.

You can try inference with different LLMs using OpenRouter or Perplexity for fast results. For more on performance comparisons, read this blog post comparing A100 and H100 inference.

### 3.3 Quantization

Quantization techniques can improve throughput by enabling models to run with less memory. A comparison of these techniques can be found here. In general, I found AWQ to perform better than GPTQ.

In 2022, medium-sized models (45–70B parameters) generated around 150 tokens per second for batch size 1, while Groq achieved 500 tokens per second using their proprietary TPU, which is faster than other GPU systems.

For further information on Groq's TPUs and performance benchmarks, refer to the following sources:

- Tensor Streaming Processor paper and related YouTube video.
- Reddit thread on Groq TPUs.

## 4 Improve the Reasoning of LLMs

Reasoning and planning approaches:

- Language Agent Tree Search Unifies Reasoning Acting and Planning in Language Models

- STaR: Bootstrapping Reasoning With Reasoning
- Quiet-STaR: Language Models Can Teach Themselves to Think Before Speaking

To understand the transformers' capabilities, one can perform *prompt-engineering*. An interesting read on this topic: Asimov - The Original Prompt Engineer.

An important aspect is the **context window** of models, which helps maintain large conversations and ensure every important element is in context. Solutions include using vector databases and cosine-similarity metrics, or trained LLMs, to retrieve information.

Since the shapes of all learnable matrix weights are independent of the input token length $n$, we can use an LLM trained on a 2K context length with any size input. However, results may not always be meaningful. A common procedure is to train the model on a 2K context and then fine-tune it on a larger context.

This approach is not directly feasible with the original transformer architecture due to the positional sinusoidal encoding, which lacks "extrapolation" ability. Instead, we can use another positional function, such as Attention with Linear Biases (ALiBi).

If we allow generating sequences of any size, not all tokens in a context of size 100K are relevant to each other. One way to reduce the number of computations is to consider only some tokens when calculating the attention scores. The goal of adding *sparsity* is to make the computation linear to $n$, not quadratic. This is called **sparse attention**. More details can be found here.

## 5 Transformers Aspects to Improve

Positional encoding introduced by Transformers (2017) includes learned absolute positional encoding with sine and cosine positional functions. Other techniques involve learned relative positional encoding. Below are some positional techniques:

- Rotary Position Embedding (RoPE): extending the RoPE, explained here.
- Flash-decoding for long-context inference
- Flash attention: brings down the computation time of attention from quadratic to linear, explained here.

## 6 Multi-modality and Agentic Systems

Insights on the GPT-4V model for image processing can be found here. Some interesting open-source Vision Transformers (ViTs) in late 2023 include CogVLM, Adept Fuyu, MiniGPT, BLIP-2, and LLaVa-1.5. A more recent summary of Vision-Language models is provided in a paper from Meta (May 2024).

There is often a parallel between vision models and software agents, as vision on desktops can be incredibly helpful.

- CogAgent: trained on desktop images.
- Mind2Web: a WebAgent.
- Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks.
- LLM can solve Computer Tasks.

Other techniques combine LLMs and browser software, such as GPT4-V Act, which uses vision with CSSOM (CSS version of DOM).

- VOYAGER: An Open-Ended Embodied Agent with Large Language Models. A Minecraft agent stocking knowledge.

## 7 A Brief Insight into the Biggest Generative Models

The *LLaMa* series, released in February 2023, features models with up to 65 billion parameters and a 2,048-token context window. It introduces three modifications to the traditional transformer architecture: RMSNorm for pre-normalization, rotary embeddings, and the SwiGLU activation function. The subsequent LLaMa2 series, launched in July 2023, scales up to 70 billion parameters and a 4,096-token context window, specifically optimized for dialogue applications.

In comparison, GPT-4, released in March 2023, has 1,760 billion parameters with context windows of 8,192 and 32,768 tokens. The data generated by GPT-4 is often used to fine-tune other models. An example is Orca2, a derivative of LLaMa2 developed by Microsoft in 2023, which trains LLaMa2 base models on high-quality synthetic data produced by GPT-4 (see the LLM index for more model comparisons).

The *Chinchilla law*, introduced in the Chinchilla paper, proposes a rule of thumb for the ideal dataset size given the model's size. It specifies that the number of tokens during training (dataset size including epochs) should be roughly 20 times the number of parameters of the model. BloombergGPT is a model trained on financial data following this rule and performs well, although the reliability of the law is still debatable.