LLM

Firoz Hasan

Overview of LLM

- LLM is a subset of DL.
- Parts of Generative AI.
- It refers to large general purpose language models that can be pre-trained and then fine-tuned for specific purpose.
- It used to train for solving common language problems.

Conti…

- Large
  - Large training dataset
  - Large number of parameters.
    - Are called Hyper parameters.
    - Are basically the memories and the knowledge that the machine learned from the model training.
    - It defines the skill of a model in solving a problem such as predictive text.

Conti…

- General purpose
  - Common Problem
  - Resource Restriction
- Pre-trained and fine-tuned
  - Pre-trained :for a general purpose with a large dataset.
  - Fine-tuned: it for specific aims with a much smaller dataset.

How much does it cost?

Llama 2 (7b)	~180,000 GPU hours [2]	→	**~10b model**	~100,000 GPU hours
Llama 2 (70b)	~1,700,000 GPU hours		**~100b model**	~1,000,000 GPU hours

Renting

Invidia A100: $1-2 per GPU per hour

⟹ **10b model: $150,000**
100b model: $1,500,000

Buying

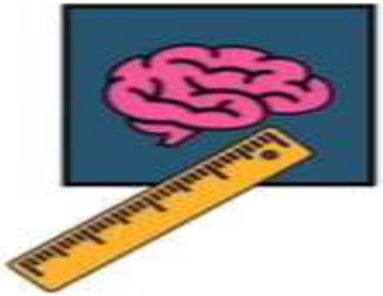
Invidia A100: ~$10,000

⟹ **GPU Cluster**: ~$10,000 x 1000 = **$10,000,000**

Training Energy Cost (100b model): ~1,000 megawatt hour [3]

4 Key Steps

1. **Data Curation**
2. **Model Architecture**
3. **Training at Scale**
4. **Evaluation**



Step 1: Data Curation

The quality of your model is driven by the quality of your data

Step 1: Data Curation

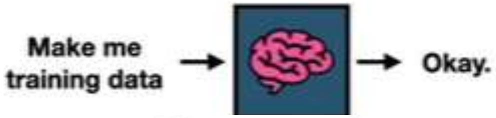
Where do we get all these data?

The internet e.g. web pages, wikipedia, forums, books, scientific articles, code bases, e

Public datasets
- Common Crawl (Colossal Clean Crawled Corpus i.e. C4, Falcon RefinedWeb)
- The Pile[6]
- Hugging Face Datasets

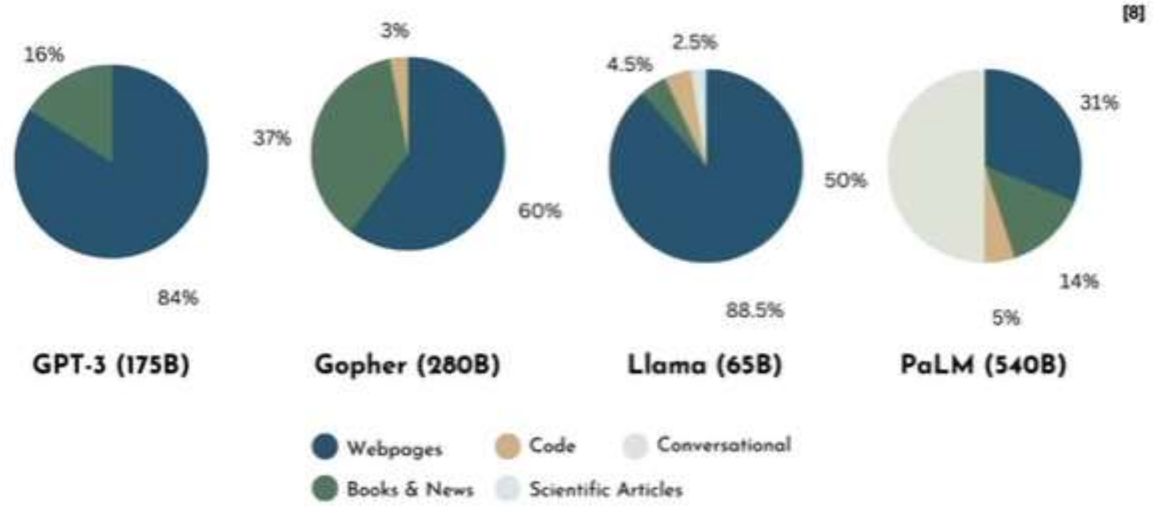
Private data sources e.g. FinPile (BloombergGPT) drawn from Bloomberg archives[1]

Use an LLM e.g. Alpaca - an LLM trained on structured text generated by GPT-3[7]



Step 1: Data Curation

Dataset Diversity



Step 1: Data Curation

How do we prepare the data?

Classifier-based

Heuristic-based

Quality Filtering - remove "low-quality" text from dataset[8]

De-duplication - several instances of same (or very similar) text can bias model and disrupt training[8, 9]

Privacy Redaction - removal of sensitive and confidential information

Step 1: Data Curation

How do we prepare the data?

Classifier-based

Heuristic-based

Quality Filtering - remove "low-quality" text from dataset[8]

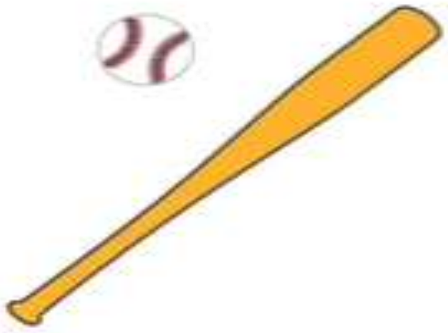
Step 2: Model Architecture

Transformers

Neural network architecture that uses **attention** mechanisms

Attention mechanism - learns dependencies between different elements of a sequence based on position and content [13]

"I hit the baseball with a bat"



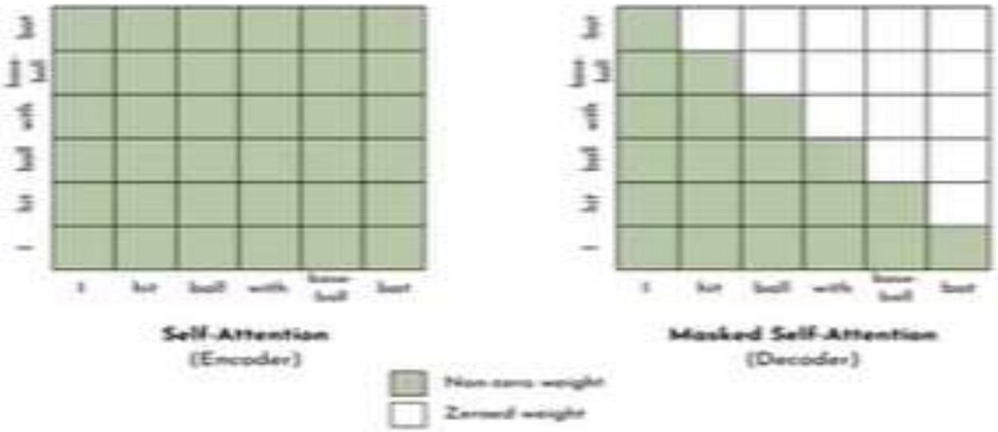
"I hit the bat with a baseball"

Step 2: Model Architecture

3 Types of Transformers [14, 15]

Encoder-only - encoder translates tokens into a semantically meaningful representation |

Decoder-only - similar to encoder but does not allow self-attention with future elements |



Encoder-decoder - combines both and allows cross-attention | *tasks: translation* [13, 15]

Step 2: Model Architecture

Other Design Choices

Residual Connections - allow intermediate training values to bypass hidden layers[14]

[18]

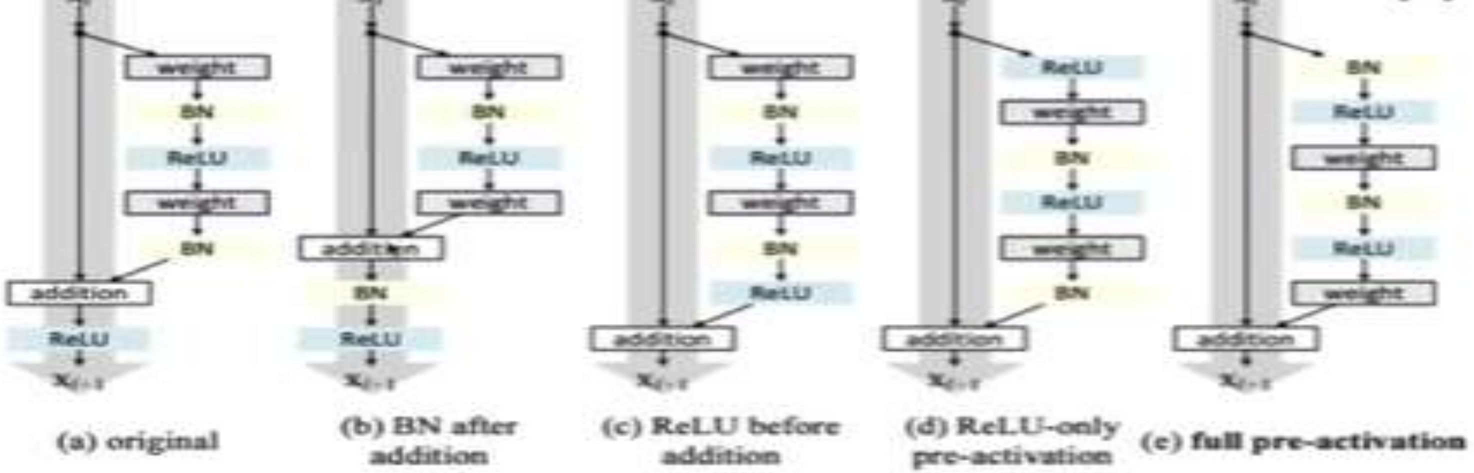
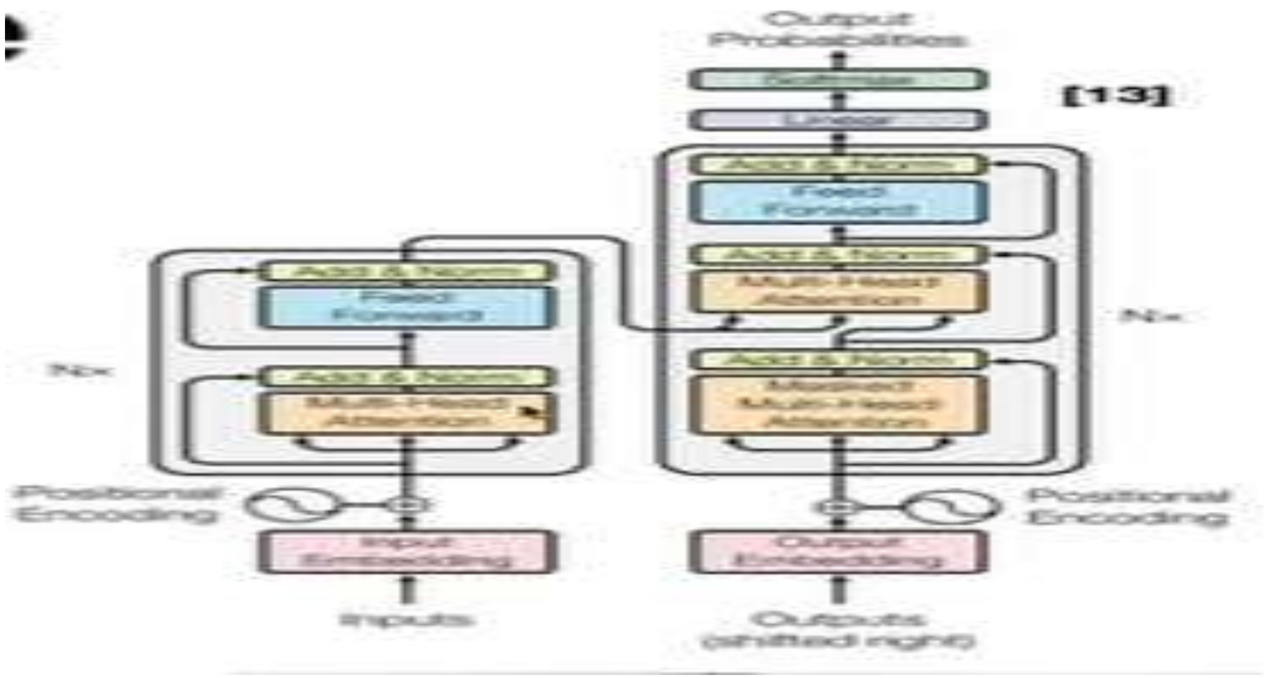


Figure 4. Various usages of activation in Table 2. All these units consist of the same components — only the orders are different.



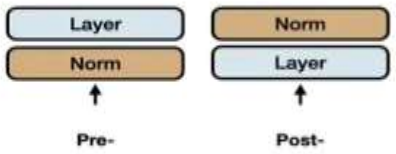
Step 2: Model Architecture

Other Design Choices

Residual Connections - allow intermediate training values to bypass hidden layers

Layer Normalization - re-scaling values between layers based on their mean and standard

Where you normalize



How you normalize

$$y = \frac{x - \bar{x}}{\sqrt{Var(x) + \epsilon}} \times \gamma + \beta$$ **Layer Norm**

$$y = \frac{x}{RMS(x)} \times \gamma + \beta$$ **RMS Norm**

Step 2: Model Architecture

Other Design Choices

Residual Connections - allow intermediate training values to bypass hidden layers

Layer Normalization - re-scaling values between layers based on their mean and standard deviation

Activation Functions - introduce non-linearities into model *e.g. GeLU, ReLU, Swish, SwiGLU, GeGLU*

Step 2: Model Architecture

How big do I make it?

If model is too big or trained too long, it can overfit

If model is too small or not trained long enough, it can underperform

[21]

Parameters	FLOPs	Tokens
400 Million	1.92e+19	8.0 Billion
1 Billion	1.21e+20	20.2 Billion
10 Billion	1.23e+22	205.1 Billion
67 Billion	5.76e+23	1.5 Trillion
175 Billion	3.85e+24	3.7 Trillion
280 Billion	9.90e+24	5.9 Trillion
520 Billion	3.43e+25	11.0 Trillion
1 Trillion	1.27e+26	21.2 Trillion
10 Trillion	1.30e+28	216.2 Trillion

Step 3: Training at Scale

3 Training Techniques

Mixed Precision Training - uses both 32-bit and 16-bit floating point data types[8, 22]

3D Parallelism - combination of pipeline, model, and data parallelism[8]

- **Pipeline Parallelism** - distributes transformer layers across multiple GPUs
- **Model Parallelism** - decomposes parameter matrix operation into multiple matrix multiplies distributed across multiple GPUs
- **Data Parallelism** - distributes training data across multiple GPUs

Step 3: Training at Scale

Training Stability

Checkpointing - takes a snapshot of model artifacts so training can resume from that point [8]

Weight Decay - regularization strategy that penalizes large parameter values by adding a term (e.g. L2 norm of weights) to the loss function or changing the parameter update rule [8, 24]

Step 3: Training at Scale

Hyperparameters

Batch Size: (*Static*) typically ~16M tokens. (*Dynamic*) GPT-3 increased from 32K to 3.2M

Learning Rate: (*Dynamic*) learning rate increases linearly until reaching a maximum value then reduces via a cosine decay until the learning rate is about 10% of its max value [8]

Optimizer: Adam-based optimizers are most commonly used for LLMs [8]

Dropout: typical values between 0.2 and 0.5 [32]

Step 4: Evaluation

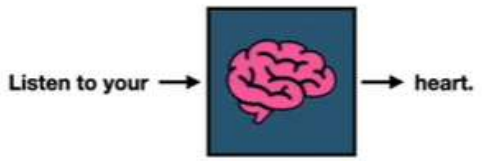
Benchmark Dataset (Open LLM Leaderboard)

Step 4: Evaluation

Multiple-choice Tasks e.g. ARC, Hellaswag, MMLU

Step 4: Evaluation

Multiple-choice Tasks e.g. ARC, Hellaswag, MMLU



Step 4: Evaluation

Open-ended Tasks e.g. TruthfulQA

Human Evaluation - a person scores completion based on ground truth, guidelir

NLP Metrics - quantify completion quality via metrics such as Perplexity, BLEU,

Auxiliary Fine-tuned LLM - use LLM to compare completions to ground truth[30]

What's Next?

Base models are typically a starting point, not final solution

Prompt Engineering

Model fine-tuning

LLM use cases

- Single model can be used different tasks.
- It trained petabytes of data and generative billions of parameters are smart enough to solve different tasks.

Conti…

- ✓it requires minimal field training data to when you tailor then to solve specific problem
- ✓it obtain decent performance even with little domain training data.
- ✓it can be used for
  - ✓few-shot(refers to train a model with minimal data
  - ✓zero-shot(A model can recognize things that have not explicitly been taught in the training before.
- ✓Its performance grow by adding more data and parameter.

Prompt Engineering

- Prompt design is the process of creating a prompt that is tailored to the specific task that the system is being asked to perform.
- Prompt Engineering
  - Process of creating a prompt that is designed to improve performance.
- Generic Language models
  - Predict the next word based on the language in the training data.
- Instruction tuned
  - Trained to predict a response to the instructions given in the input
  - Summarizes the text for example

Conti…

- Dialog-tuned language model
  - It is a special case of instruction tuned where requests are typically framed as a question to chatbot.
  - Tuning. it is a task that you want to perform.

  The process of adapting a model to a new domain or set of customs use cases by training the model or new data.