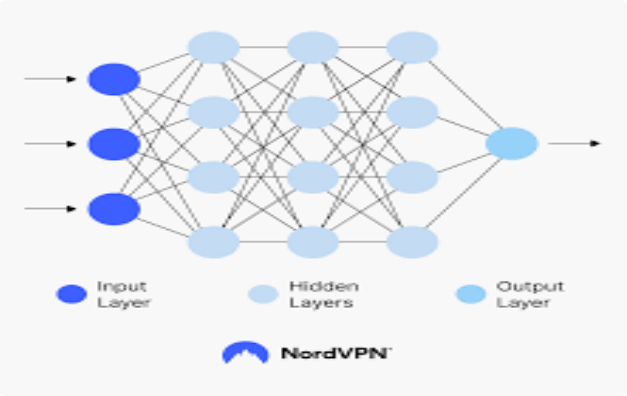
Chapter 1: Introduction to LLMs



LLM (Large Language Model) - A neural network is designed to understand, generate and respond to human-like text.

- The models have billions of parameters, and wide range of NLP (Natural Language Processing) tasks: Question answering, translation, sentiment analysis and much more.

Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}
GPT-3 Small	125M	12	768	12	64
GPT-3 Medium	350M	24	1024	16	64
GPT-3 Large	760M	24	1536	16	96
GPT-3 XL	1.3B	24	2048	24	128
GPT-3 2.7B	2.7B	32	2560	32	80
GPT-3 6.7B	6.7B	32	4096	32	128
GPT-3 13B	13.0B	40	5140	40	128
GPT-3 175B or “GPT-3”	175.0B	96	12288	96	128

Difference between LLMs and NLP models

LLMs	**NLP models**
Can do a wide range of NLP tasks. “Can write email with custom instructions and task that trivial for modern life	Designed for specific tasks like language translation etc.

Concept of Transformer Architecture

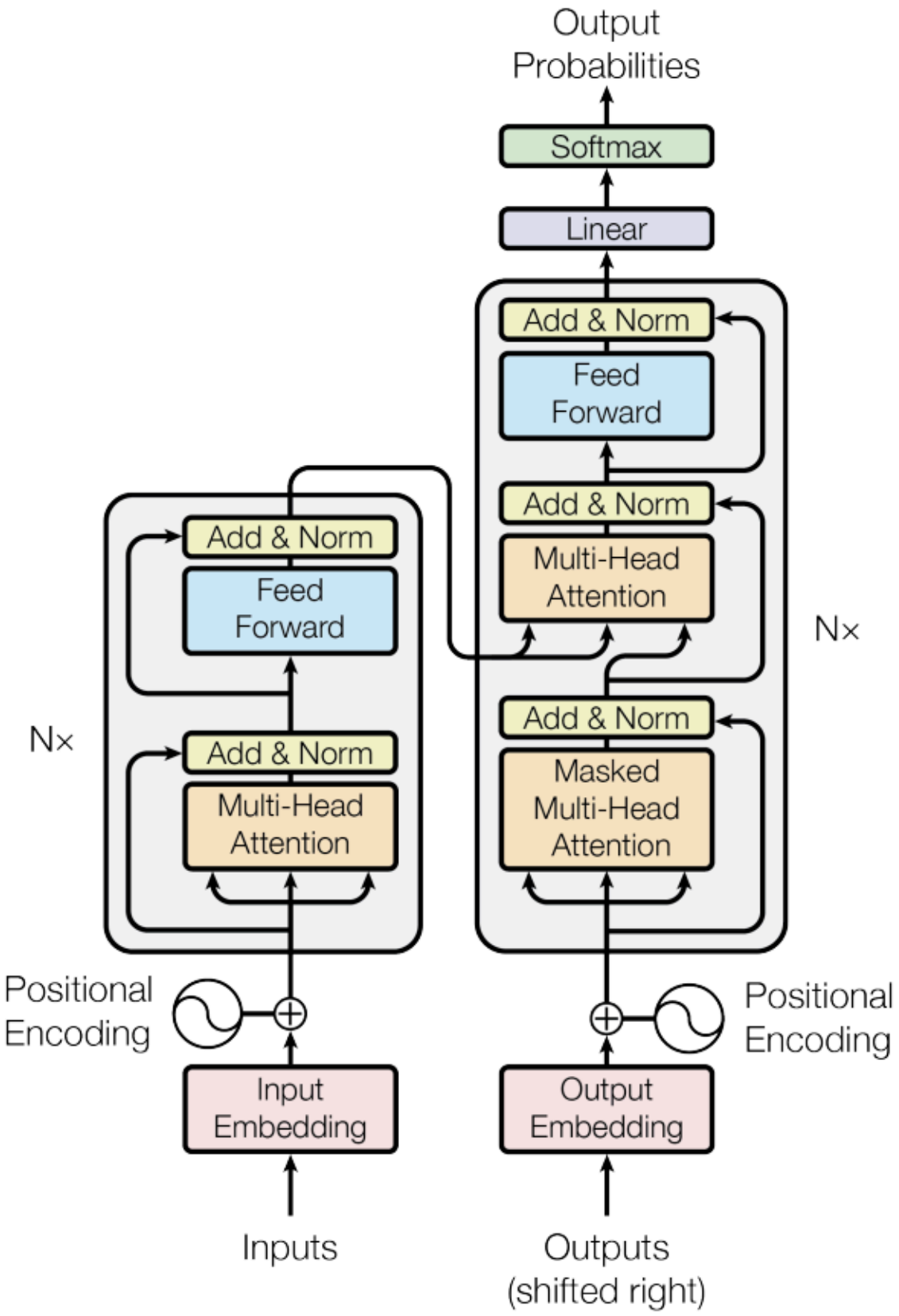


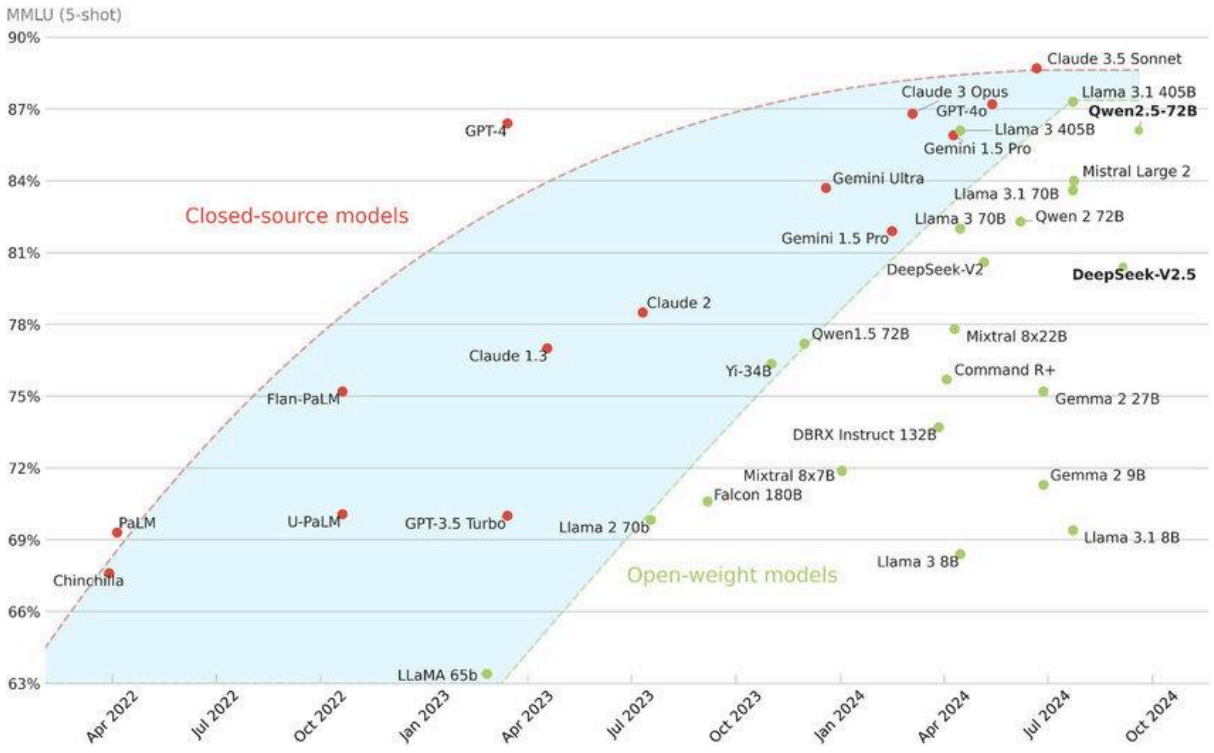
Figure 1: Flow of LLMs (Vaswani et al., 2022)

LLM = Deep Learning + Machine Learning + Artificial Intelligence
Generative AI = LLM + Deep Learning
Generative AI - Using deep neural networks to create new content such as text, images and various contents

Applications of LLMs

1. Chatbots/Virtual assistant
2. Machine Translation
3. Novel text generation
4. Sentiment Analysis
5. Content Creation

Closed source and Open-source of LLMs over time



Chapter 2: Stages of Building LLMs

Creation an LLM = Pretraining (1st Stage) + Finetuning (2nd Stage)

Pretraining - Training on a large, diverse dataset

- Dataset to train GPT-3 (Brown et al., 2022)

Dataset	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60%	0.44
WebText2	19 billion	22%	2.9
Books1	12 billion	8%	1.9
Books2	55 billion	8%	0.43
Wikipedia	3 billion	3%	3.4

-

Finetuning - Refinement by training on narrower dataset, specific to a particular task or domain

Reason to Use Fine-Tuning

Fine-tuning is useful when a model needs to perform well in a **specific domain** or on **proprietary company data** that isn't covered in general training. Fine-tuning is most valuable when you need an AI model to perform well on **narrow, domain-specific**, or **organization-specific** tasks, rather than general-purpose use.

For example:

- A **student** using AI to learn general topics (like math, writing, or history) can rely on a **base model** because it already has broad, general knowledge. In this case, fine-tuning is unnecessary or minimal.

- A **manager in a company**, however, may need an AI system to assist with tasks such as analyzing internal reports, generating domain-specific documents, or following company policies. This requires the AI to understand specialized terminology, workflows, or confidential data. Here, **fine-tuning** (based on the pretrained model) helps adapt it to the company's specific needs and ensures better accuracy and relevance.

Pretraining + Finetuning **flows**:

1. Internet text, books, media research articles (Raw unlabeled text, which consist of trillion of words)
2. Pretrained LLM (Foundation model), LLM pretrained on unlabeled text data
3. Classification, summarization, translation and personal assistant (Finetuned LLM) (Labeled Dataset required here) [If use GPT-API, then only this step is needed for development]

Steps for building LLM:

1. Train on a large corpus of text data (raw text)
    - Raw text = regular text without any labeling information.
2. First training stage of LLM (known as pretraining).
    - Create initial pretrained LLM (Base/foundational model)
    - Example: GPT-3 model is a pretrained model which is capable of text completion.
3. After obtaining the pretrained LLM, we can further train LLM on labelled data. (Finetuning)
    - Example get data from JP morgan
4. 2 popular categories of finetuning
    a. Instruction finetuning - Labeled dataset consists of instruction-answer pairs (For example, support different languages for same type of answer)
    b. Finetuning for classification tasks - labeled dataset consists of text if associated labels (For example 10k of emails with spam and no-spam)

Chapter 3: Basic Introduction to Transformers

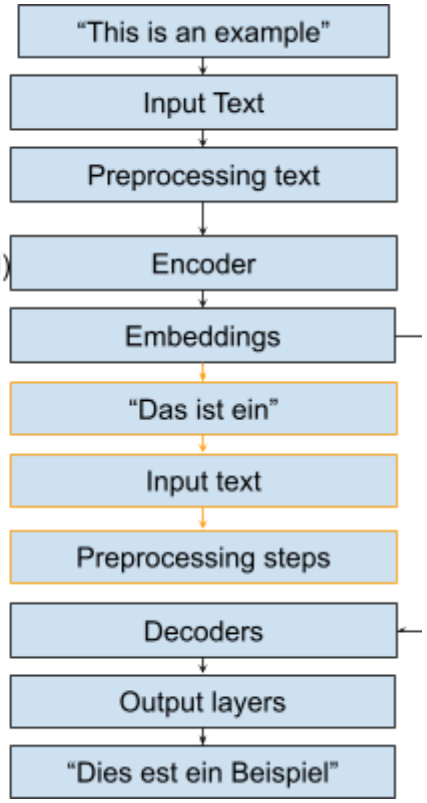
(This chapter is fully based on Attention is all you need (Vaswani et al., 2022))

Most modern LLMs rely on the transformer architecture → Deep neural network architecture introduced in the 2017 paper (Attention is all you need). GPT is based on this architecture for development

Attention is all you need, Purpose - Experiments on two machine translation tasks to show the quality (Translate From English texts to German and French)

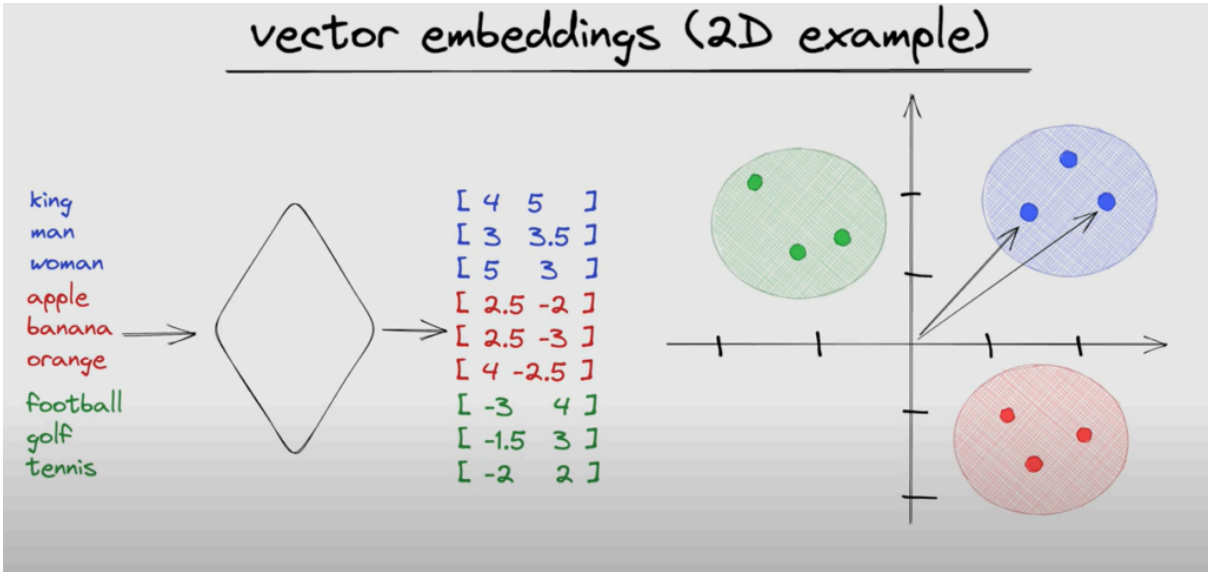
Transformer Architecture

1. Input Text to be translated
2. Input text prepared for encoder (Tokenization)
3. Produces tet encodings used by decoder (Vector embedded)
4. Encoder returns embedding vector as input to decoder
5. Partial output text model completes one word at a time (It should be “Dies est eis Beispiel”)
6. Input text is prepared for decoder
7. Generates translated one word at a time
8. The complete output (Translation)



Transformer Architecture consist of

1. **Encoder Block**
    - Encodes input text into vectors



2. **Decoder Block**
    - Generate output text from encoded vectors

A Note on Self-attention mechanism:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

- Key part of transformers: Allows model to weight importance of different words/tokens relative to each other
- *Attention mechanism Model dependencies between different words*
- Enables model to **capture long range dependencies**

Later variations of transformer architecture

1. **BERT (Bidirectional Encoder Representations from Transformers) -** Predicts hidden words in a given sentences
    - Eg : It will receive input where words are randomly masked during training. →Input text → preprocessing steps → encoder → Replaced hidden words, then output answer
    - It can differentiate the “bank” in two types of meaning (a financial institution or riverbank) by considering the structure of the sentence.
2. **GPT Models (Generative Pretrained Transformers) -** Generates new words
    - Eg: Receive incomplete text → input text → preprocessing text → decoder → Return the complete text.

Differences between Transformer and LLMs

- Not all transformers are LLM.
- Transformers can be used in computer vision.
- Not all LLMs are transformers
- LLMs can be based on recurrent or convolutional architectures as well

Chapter 4: A closer look at GPT

(This chapter is based on research of GPT-1, GPT-2, and GPT-3)

Difference between ZERO SHOT Learning and FEW SHOT Learning

- **ZERO SHOT:** Ability to generalize to completely unseen tasks without any prior specific examples
    - Eg: Translate English to French:
        - Cheese => ___
- **FEW SHOT:** Learning from a minimum number of examples which the suer provides as input (**GPT-3 use this)**
    - Eg: Translate English to French:
        - Sea otter => loutre de mer
        - Peppermint => menthe poivree
        - Plush girafe => girafe peluche
        - Cheese => ___
- **ONE SHOT:** See one example from the task
    - Eg: Translate English to French:
        - Sea otter => loutre de mer
        - Cheese => ___

***GPT-4 is a combination of Few shot learner and Zero shot learner.**

Aspect	**Zero Shot**	**Few Shot**
Process	**Complete task without examples**	**Complete tasks with examples**

Token - A unit of a text which the model reads. 1 token = 1 word.

Importance of Pretrained models

1. Preparation for further finetuning
2. Many pretrained models are available as open-source models, which can be used as general purpose tools to *write, extract and edit text* which not part of training data

GPT models are simply trained on “next-word” prediction tasks.

Text: Second Law of Robotics: A robot must obey the orders given it by human beings

Generated training examples

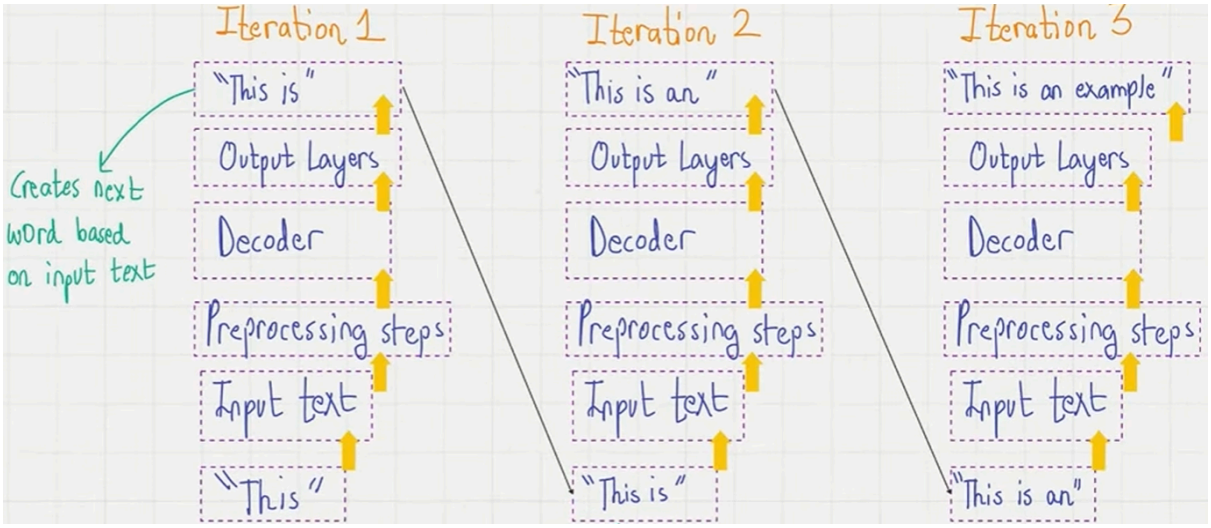
Example #	Input (features)	Correct output (labels)
1	Second law of robotics :	a
2	Second law of robotics : a	robot
3	Second law of robotics : a robot	must
...		

With this training, they can do a wide range of other tasks like translation, correction and spelling and more.

We don’t collect labels for training data, but use the structure of the data itself to make the label. [Auto regressive model: use previous outputs as inputs for future predictions]

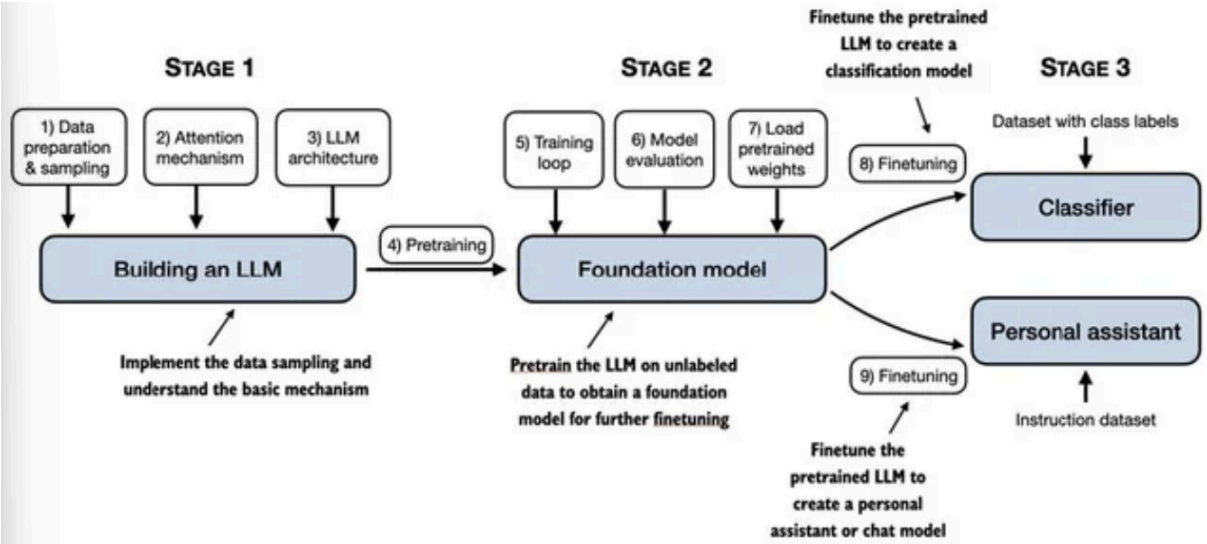
Chat-GPT is Unsupervised ML + Pre-training + Auto Regressive Model
*In GPT, there is no encoder block, only decoder block provided
In original transformer have 6 encoder-decoder blocks
ChatGPT-3 has **96 transformer layers, 175 billion parameters.**

Decoder only architecture



Although trained only for next word prediction, the GPT model can perform other tasks like language translation. (Known as Emergent Behaviour - Ability of a model to perform tasks that the model wasn't explicitly trained to perform.)

Chapter 5: Stages of building LLM

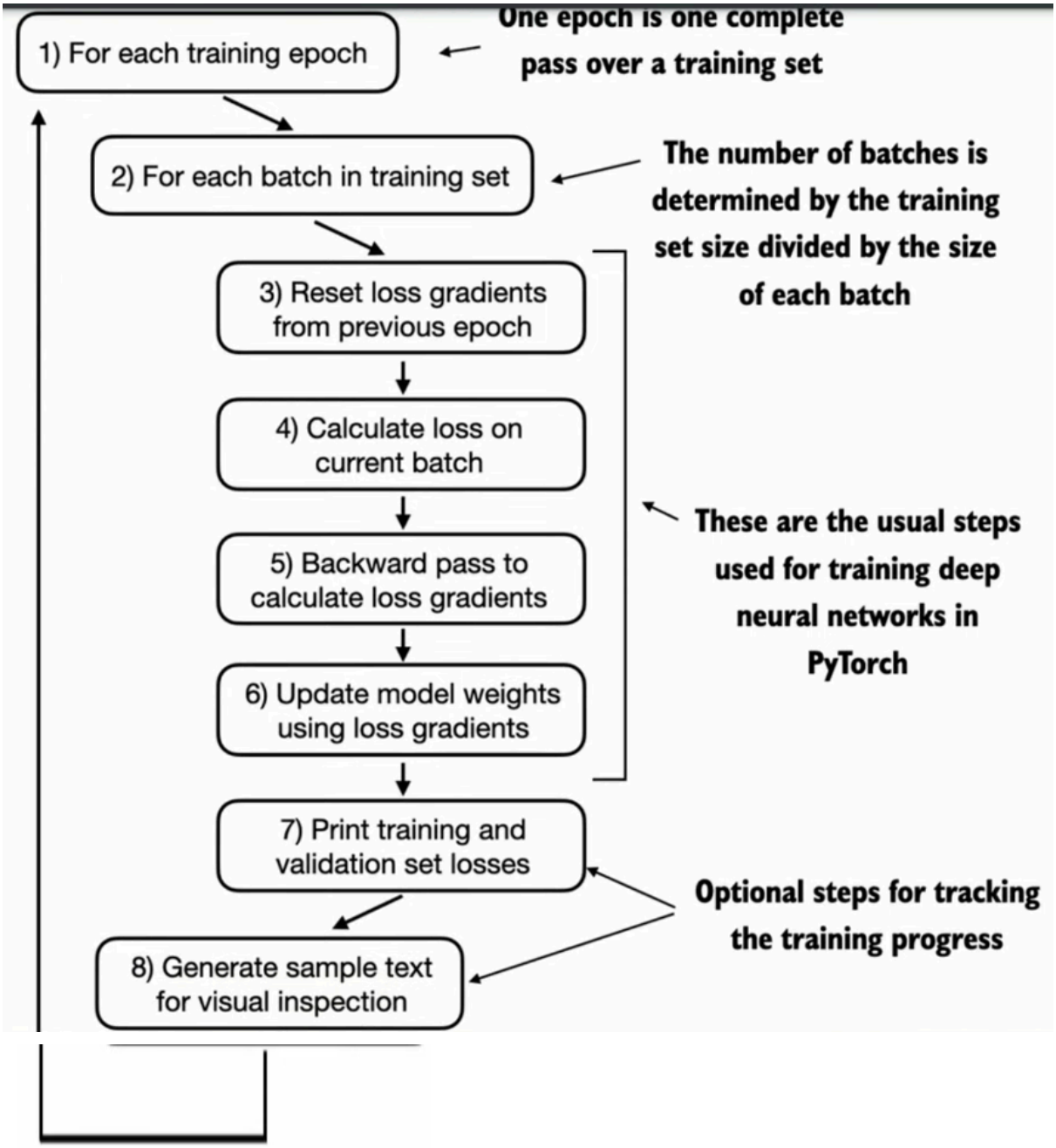


Prerequisite of Stage 1

1. Data preprocessing
2. Attention mechanism (Key query value, how attention shows in Python, Attention score, position of encoding, vector embeddings, Tokenization)
3. LLM Architecture (Where attention had to go)

(After all data had been assembled and finalize the LLM architecture)

Stage 2



RECAP of things learned (Chapter 6 to Chapter 7)

1. LLMs have transformed the field of NLP. They have led to advancements in generating, understanding and translating human language
2. Modern LLMs are trained on 2 main steps
    a. Pretraining on **unlabeled** data (Foundational model), can use very large datasets (in billions)
    b. Finetuning on a smaller, **labeled** dataset (Finetune LLMs can outperform just pretrained LLMs on specific tasks)
3. LLMs are based on transformer architecture
    - Key idea. Attention mechanism gives LLM selective access to whole input sequence when generating output one word at a time
4. Original transformer: Encoder + Decoder
5. GPT: only decoder, no encoder
6. While LLMs are only trained for predicting next word, they show **emergent properties** (Ability to classify, translate and summarize texts)

References

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., . . . Amodei, D. (2022). Language Models are Few-Shot Learners. *arXiv (Cornell University)*, *33*, 1877–1901. **GPT-3** https://doi.org/10.48550/arxiv.2005.14165

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training. **GPT-1**

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners.**GPT-2**

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2022). Attention is all you need. *arXiv (Cornell University)*, *30*, 5998–6008. https://doi.org/10.48550/arxiv.1706.03762