Steps for building LLM

① Train on large corpus of text data (Raw text)
↳ Regular text w/ out any labelling info

② First stage of LLM (pretraining)
- GPT-3 model is pretrained model which is capable of text completion

③ After obtaining pretrained LLM, we can further train LLM on labelled data → fine tuning

④ Two popular fine tuning categories

→ Instruction fine tuning
- labelled dataset consists of instruction-answer pairs
  - text translation, doctor customer support

→ Classification tasks fine tuning
- labelled dataset consists of text and associated labels,
  - emails → spam vs not spam

Basic intro to Transformers

- Deep neural network arch. most modern LLMs use
  - ↳ 2017 paper introduced
  - ↳ Attention is all you need
- originally proposed for english → german, english → French,
- Decoder

Encoder returns embeddings vector as input to decoder

Embeddings (4)

Encoder (3)

Produces text encoding used by decoder (encoder embeddings)

Preprocessing (tokenization) prepared for encoder

Input text (2)

"Das ist ein Beispiel" (8) The complete output (translation)

Output Layers

Decoder → Generates translated text (one word at a time)

Preprocessing

Input text (6) (Input text is prepared for decoder)

(1) Input text to be translated

English "This is an example"

Simplified Transformer Architecture

"Dss est ein"

Vertical output (5)

Text model completes one word at a time

one output word produced at one time

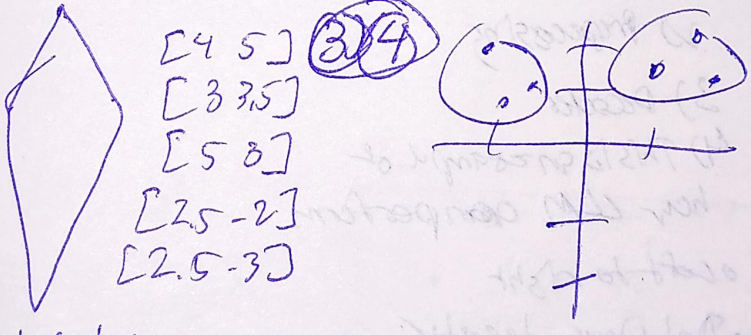
Fine tuning is fun for all

assign a number for each word/text (2)

vector embeddings (2D example) difficult step

learn
Man
woman
apple
banana
orange

[4 5] (3) (4)
[3 3.5]
[5 8]
[2.5 -2]
[2.5 -3]



Convert input text from preprocessing (tokens) into vector embeddings

Semantic meaning is captured between words

Transformer Architecture

Encoder
↳ Encodes input text into vectors

Decoder
↳ Generates output text from encoded vectors

A note on self attention mechanism
- key part of transformers
- Allow model to weigh importance of different words/tokens relative to each other
- Enable model to capture long range dependencies

(4) Later variations of transformer architecture:

BERT
sentiment analysis
spam
"Bidirectional encoder representations from transformers"
↓
Predicts hidden words in a given sentence
1) Receives input where words are randomly masked during training
* This is — of LLM — perform
2) Input
3) Preprocessing
4) Encoder
5) Fill missing words
- Looks both directions

GPT models
"Generative pre-trained transformers"
↓
Generates new word
- This is an example of how LLM can — ?
1) Input
2) Preprocessing
3) Decoder
4) This is an example of how LLM can perform
- Left to right
- Only uses decoder
- Unsupervised learning

(5) Transformers vs LLMs

- Not all transformers are LLMs
- Transformers can also be used for computer vision

Vision transformers → can be used for image classification, like CNN

- Not all LLMs are transformers
- can be LSTM or recurrent NN

Classic look at GPT

(6) Zero-shot vs few shot learning

Z → Zero-shot = Generalize to completely unseen tasks w/ out prior specific examples

- Generative pre-training (GPT)
  - data used is not labelled
  - don't need to provide labels
  - generating next word
  - Transformers + unsupervised pretraining

(OpenAI)

- GPT 3 1.7TB parameters
-
- Few-shot → learning from a limited number of examples which user provides as input