Chapter 1

- A brief introduction to the revolution of AI since 2012, and by 2020, AI could generate articles that are indistinguishable from those written by humans.
- The success of ChatGPT was unprecedented and popularized more research into the technology behind it, namely large language models (LLMs).
- However, LLMs have been around for a while now and smaller models are still relevant to this day. LLMs are much more than just a single model and there are many other techniques and models in the field of language AI that are worth exploring.
- Our history of Language AI starts with a technique called bag of-words, a method for representing unstructured text.
- bag of words first step is tokenize the sentence into number of tokens then we append them into a set which has unique words only (no repeated words) then we refer those words into a vector representation to vector the new sentence
- Embeddings are vector representations of data that attempt to capture its meaning. To do so, word2vec learns semantic representations of words by training on vast amounts of textual data, like the entirety of Wikipedia.
- If the two words tend to have the same neighbors, their embeddings will be closer to one another and vice versa.
- There are many types of embeddings, like word embeddings and sentence embeddings that are used to indicate different levels of abstractions (word versus sentence)
- Each step in this architecture is autoregressive. When generating the next word, this architecture needs to consume all previously generated words
- Attention allows a model to focus on parts of the input sequence that are relevant to one another (“attend” to each other) and amplify their signal.
- By adding these attention mechanisms to the decoder step, the RNN can generate signals for each input word in the sequence related to the potential output. Instead of passing only a context embedding to the decoder, the hidden states of all input words are passed
- In attention all you need , The authors proposed a network architecture called the Transformer, which was solely based on the attention mechanism and removed the recurrence network that we saw previously. Compared to the recurrence network, the Transformer could be trained in parallel, which tremendously sped up training
- Now, both the encoder and decoder blocks would revolve around attention instead of leveraging an RNN with attention features .
- The self attention in encoder helps also to use multi tokens in parallel rather than normal RNN which make it faster
- The decoder has an additional attention layer that attends to the output of the encoder.
- The masked self attention layer in the decoder , keep and save data leakage when generating output .
- These encoder blocks are the same as we saw before: selfattention followed by feedforward neural networks. The input contains an additional token, the [CLS] or classication token, which is used as the representation for the entire input. Often, we use this [CLS] token as the input embedding for netuning the model on specic tasks, like classication.
- we will refer to encoder-only models as **representation models** to dierentiate them from decoder-only, which we refer to as **generative models.**Representation models = Focus on *understanding* text (teal color). while Generative models = Focus on *creating* text (pink color).

- Generative LLMs, as sequence-to-sequence machines, take in some text and attempt to autocomplete it. Although a handy feature By fine-tuning these models, we can create instruct or chat models that can follow directions
- generative models being called completion models , The context length represents the maximum number of tokens the model can process .
- e, Llama 2 has been trained on a dataset containing 2 trillion tokens. Imagine the compute necessary to create that model
- we explore the incredible capabilities of LLMs it is important to keep their societal and ethical implications in mind (Bias and fairness - Transparency and accountability - Generating harmful content - Intellectual property - Regulation)
- Unfortunately, there is no single rule to determine exactly how much VRAM you need for a specic model. It depends on the model's architecture and size, compression technique, context size, backend for running the model,
- private llm like openai and anthropic models , A huge benet of proprietary models is that the user does not need to have a strong GPU to use the LLM. The provider takes care of hosting and running the model and generally has more computing available and of course you can't fine-tune your model
- The tokenizer is in charge of splitting the input text into tokens before feeding it to the generative model
-