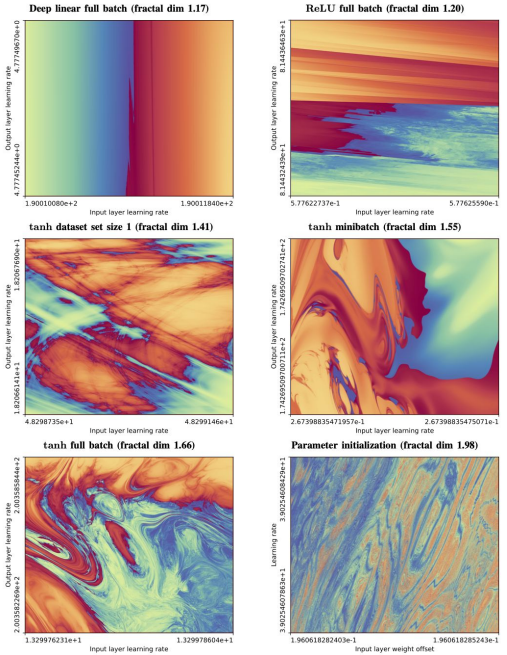
A Quick Run-Through of Transformer Architecture

Agenda - Schedule

1. Recurrent Neural Networks (revisited)
2. Attention Mechanism
3. Transformers and the AI Boom



"The boundary of neural network trainability is fractal" (i.e. neural networks are incomprehensibly complex)

Agenda - Goals

- Understand how LLMs work at a high level
- Big idea behind Attention
- Why Attention sparked the modern AI Boom

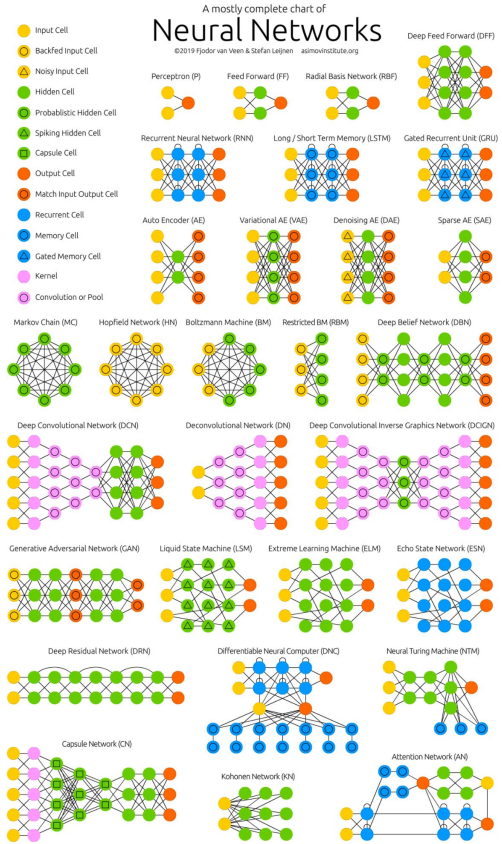
Text Prediction via Neural Networks

RNNs, CNNs, & LSTMs

Like we established previously, there are many different types of **NN architectures**.

Each major NN involves learning some new concept which gives it an “edge” when predicting a specific type of task.

Since we are just getting started with the idea of vanilla neural nets, **let’s keep our discussions of these frameworks shallow.**



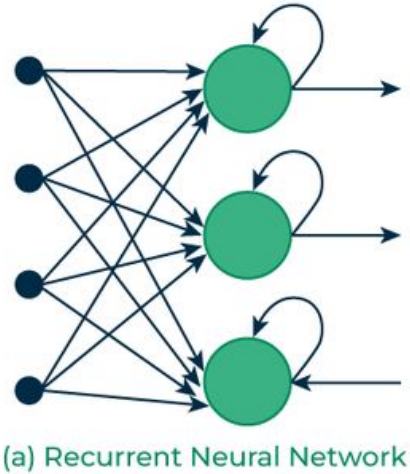
Recurrent Neural Networks - LSTM

Often used to analyze **time series & textual data**.

Accomplishes this by feeding the input from the **previous step back into the current step.**

This allows us to incorporate "memory" into our neural network and allows us to model dependent predictors into our neural net.

Often uses the **ReLU activation function**.



Neural network with memory

Limits of RNNs - LSTM

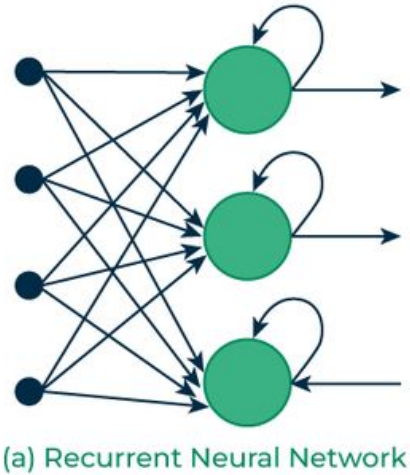
Hard to scale to huge datasets

Slow training: LSTM must process words **one-by-one**. This is visualized here: https://distill.pub/2019/memorization-in-rnns/

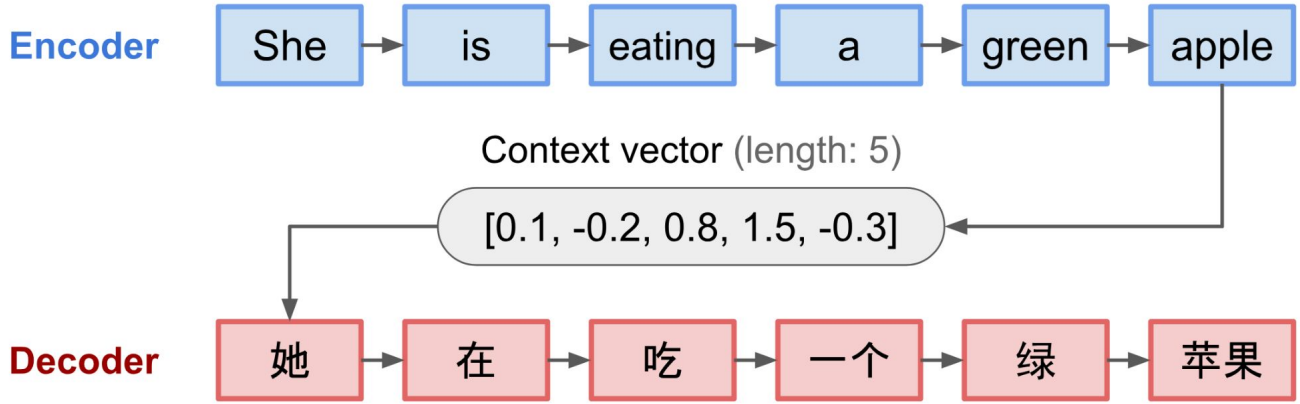
A LSTM network tries to remember everything, but the farther back something happened, the blurrier it becomes (vanishing gradient)

Ex.” The cat the was chased by the dog ran into the house”

The memory of “cat” fades by the time the sequence gets to “ran.”

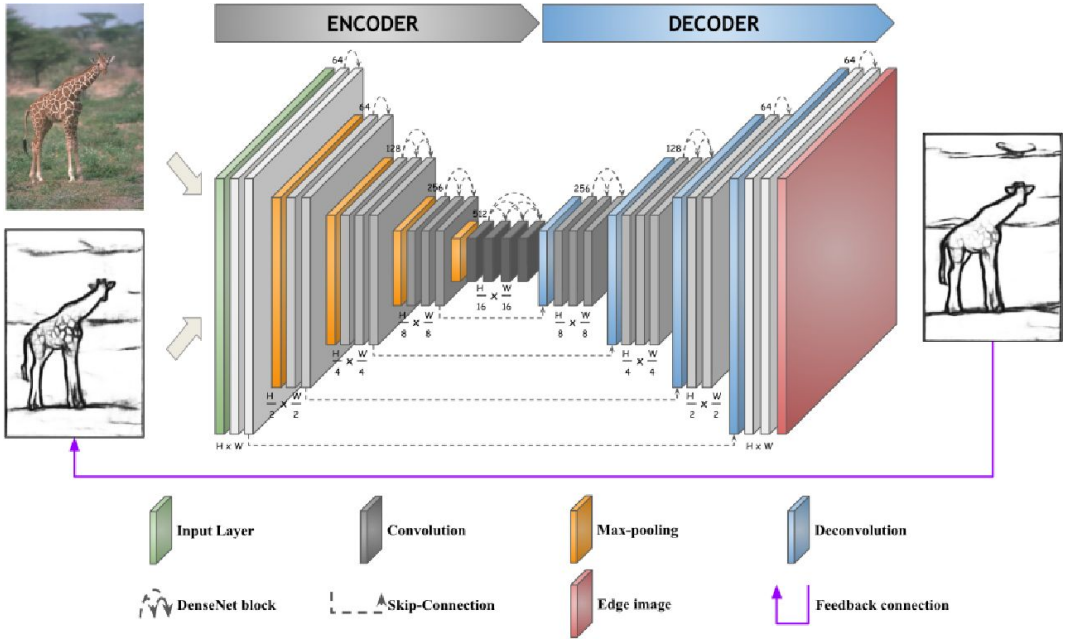


Neural network with memory



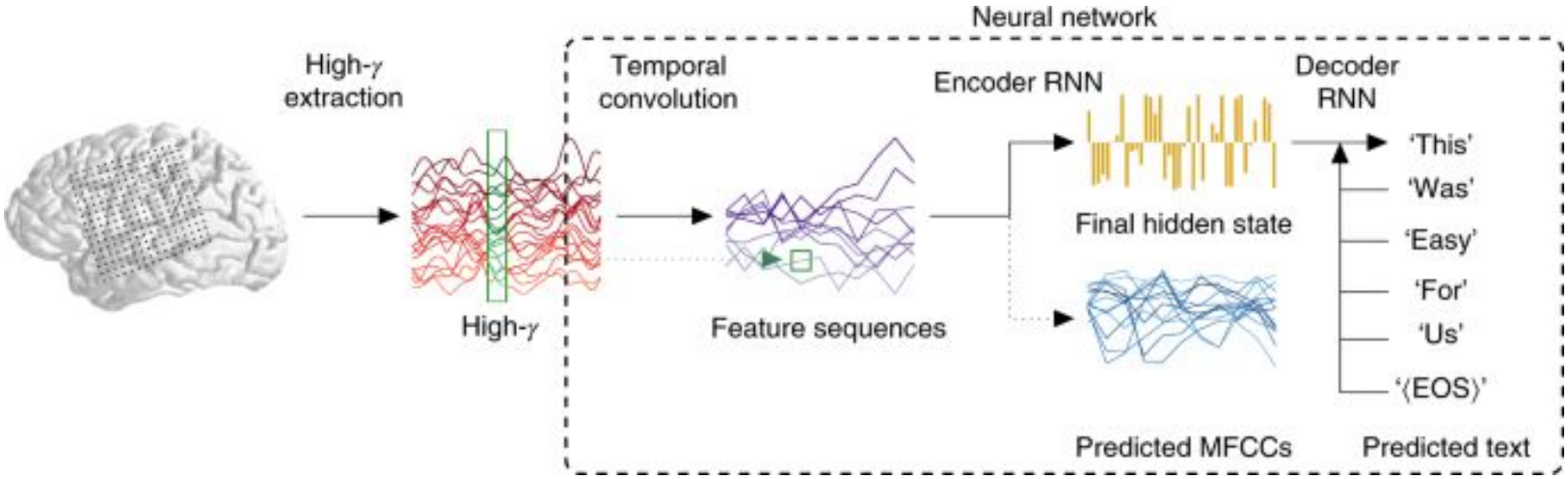
Even though the LSTM architecture is no longer used for text prediction, it gave us the **foundational idea** of the encoder/decoder framework that led to the **transformer innovation**. Attention is All You Need (2017): https://arxiv.org/pdf/1706.03762

In its essence, this idea states that by training our network to encode our input data into a context vector (*something like a vector embedding*), we can then decode it into our language of voice.



While the 2017 paper initially focused on the task of translating between languages, note that this architecture could be applied to any data of your choice (given the right data transformation).

Think of this as a **universal translation tool** that operates not only between text to text, but between any two mediums (image to text; image to image; audio to image).



In more niche circles, we could even apply the transformer architecture to EEG waves to text (aka mind-reading).

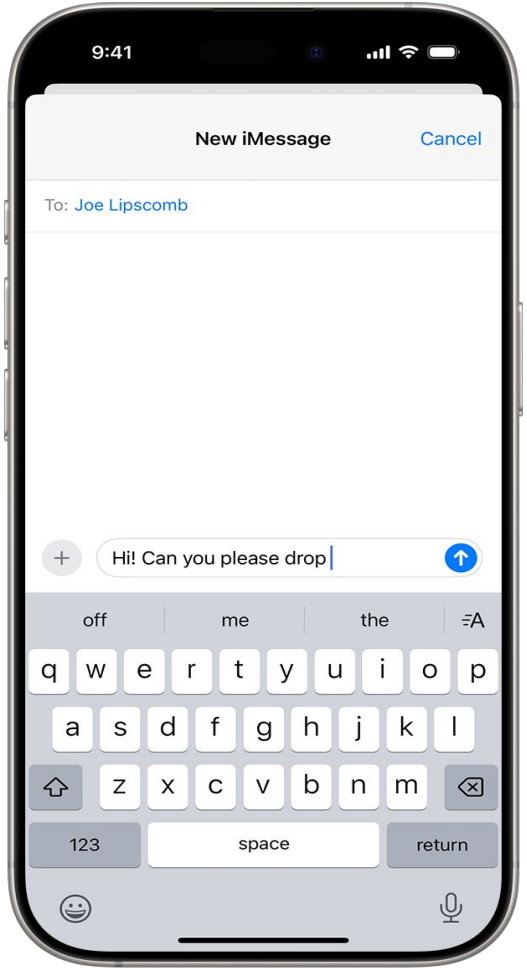
LLMs and Attention

LLMs

For now we want you to think of LLMs as an advanced version of autocomplete, given a prompt.

Rather than providing options of words to complete a phrase, an LLMs output contains a probability distribution of the next likely word.

Instead of always selecting the most probable word however, we can adjust a parameter called **temperature**, such that we see a bit of randomness in the next outputted word.



Transformer Architecture

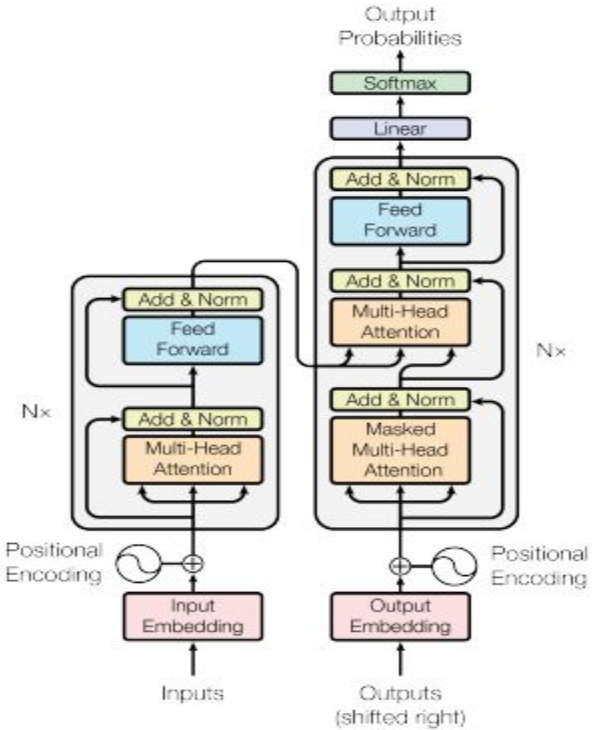
Recall that RNNs must process words one by one.

Just like the other types of NN, this particular NN is known as the transformer

The attention mechanism looks at all words and chooses which words are important

It learns to “pay attention” dynamically

LLM visualization: https://bbycroft.net/llm

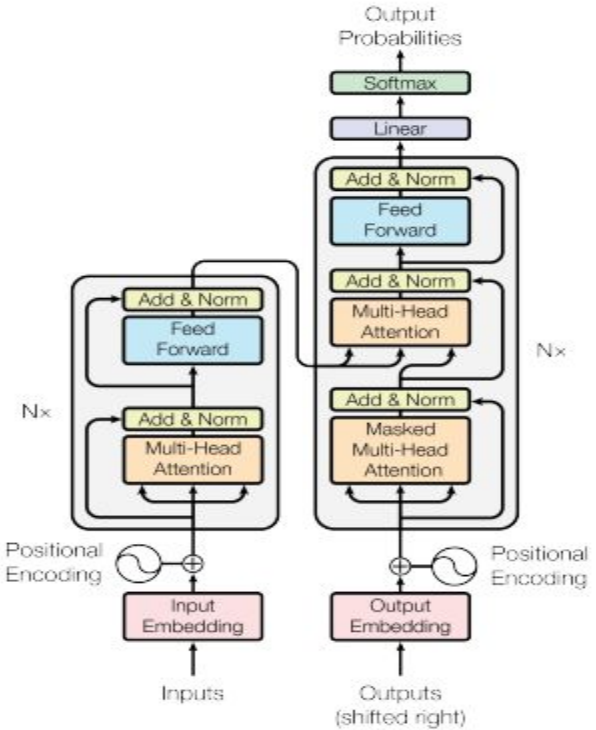


How Attention works

Every word “attends” to all previous words in a sequence of text

Assigns importance scores (attention weights)

Combines important info to make better predictions



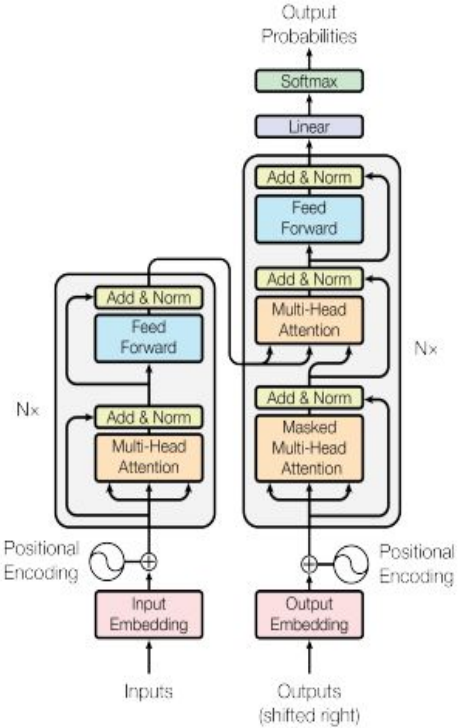
Many heads of Attention

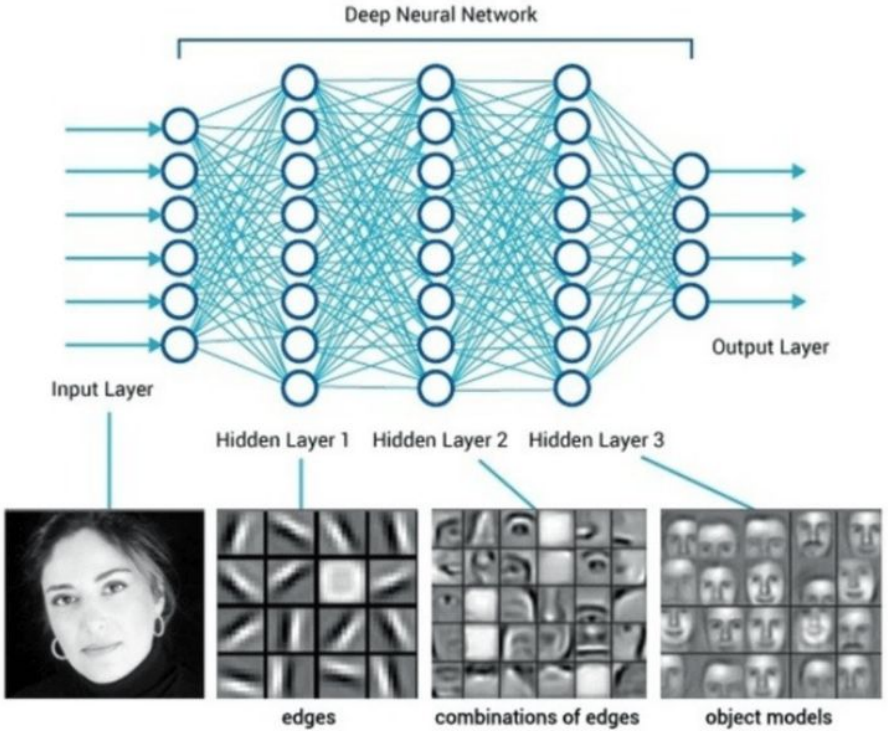
Many heads of attention = many perspectives

Each head of attention focuses differently on grammar, meaning, sentence structure etc.

They create a richer understanding in combination

Think back to image recognition network and how each hidden layer can detect shadows, edges, and eventually facial features

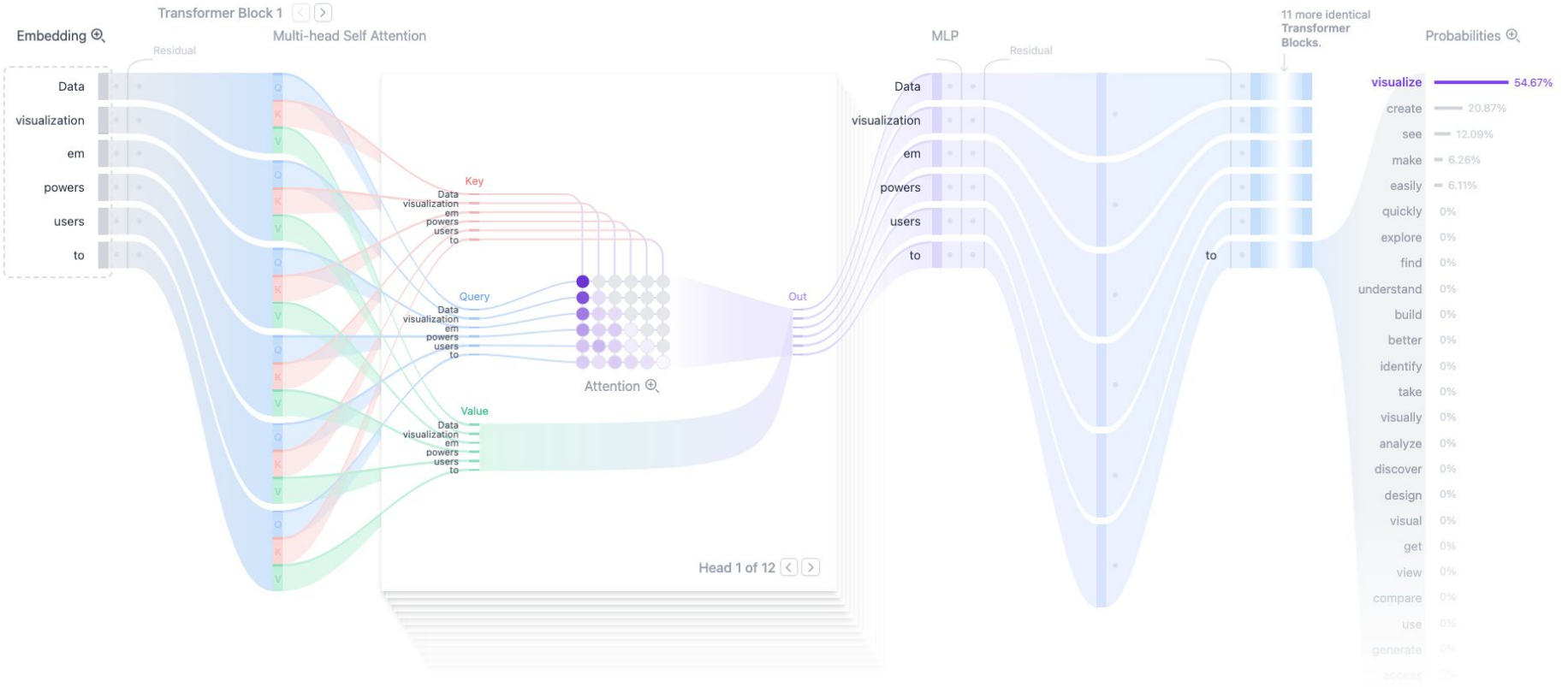




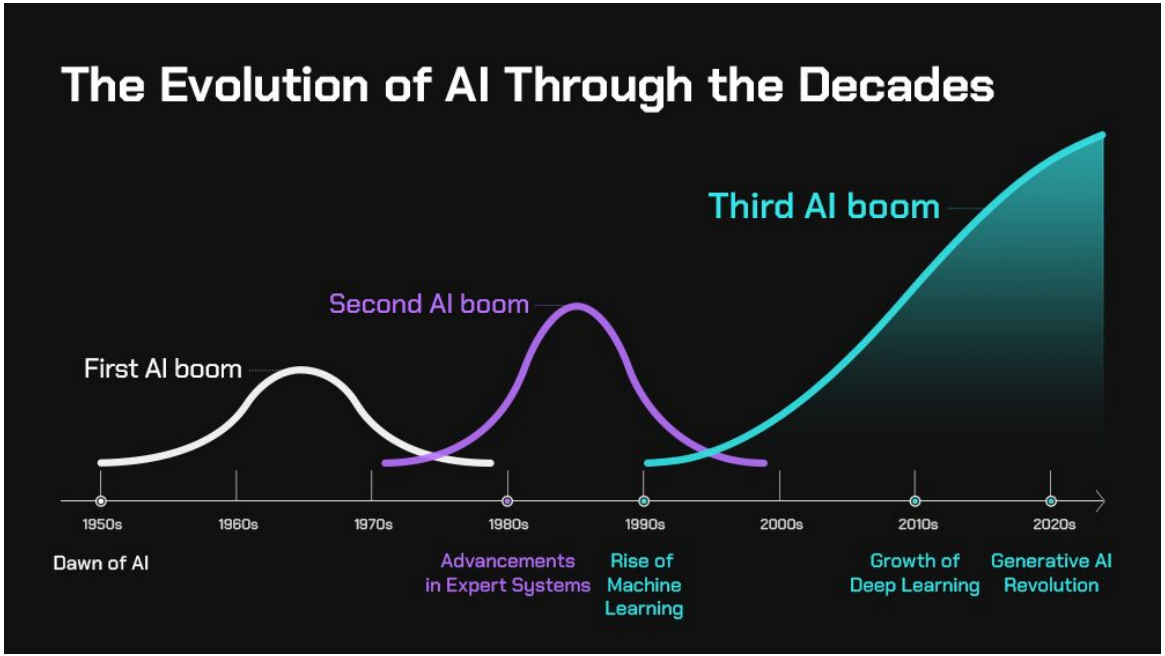
The recognition of these traits were **learned independently by the neural network itself.**

Similarly as the word embedded vectors pass through the transformer architecture, each head of attention learns different patterns within the body of text

Innovations in hardware allow us to run these neural networks. Attention models can process **all words at once.** Without parallel hardware, LLMs would not work!



Let’s visualize this in action:
https://poloclub.github.io/transformer-explainer/



Almost all of the technology behind this third AI boom has come from the transformer architecture! Although originally invented for text, (machine translation) it can be used on image data, audio data, biological data (protein folding) and much more!

In short, transformers aren't just good at reading, they're good at any problem where the model needs to look at a lot of data and decide what's important.