Large Language Models

A Conceptual Overview

Pranay Kotasthane 03 July 2024

When asked to generate a random number between 1 and 100, what did earlier LLMs produce?

Takeaways

You will get (some) answers to these questions

1. Are LLMs merely “stochastic parrots”?
2. What does "AI bias" actually mean?
3. In what way are LLMs "intelligent"? Are we any different from LLMs?
4. Are LLMs getting us towards AGI?

Agenda

What we'll learn

1. Neural Networks
2. Transformers
    1. Word Vectors
    2. “Transforming”
    3. What’s inside a Transformer?
        1. Attention
        2. Feed-forward
    4. How to train?
3. Can LLMs understand?

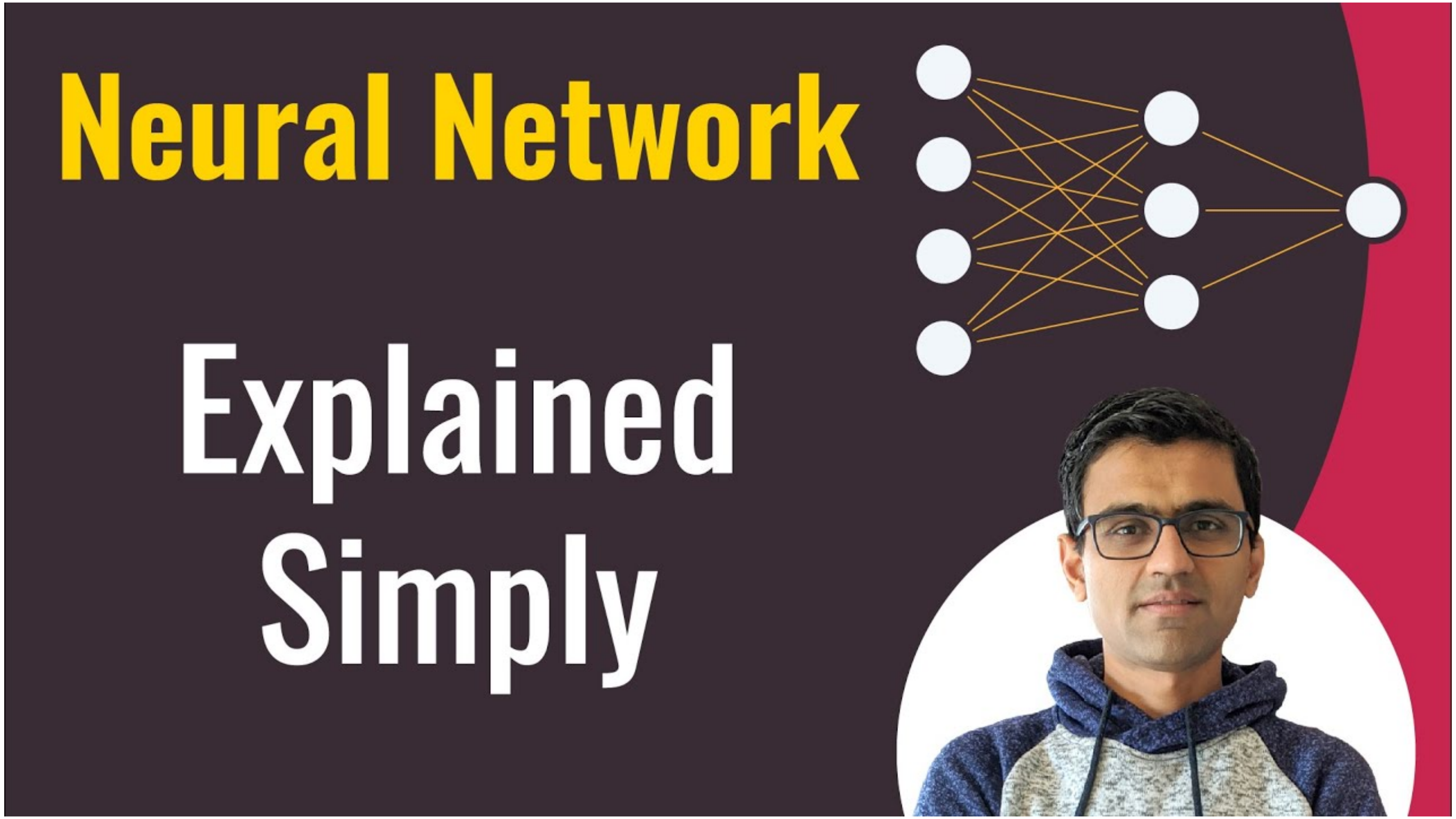
Generative Pre-trained Transformers: An Instrumental Definition

Models that predict the next word iteratively using a neural network trained on a large dataset

The Model

1. Understanding Neural Networks

The building blocks (10 minute video)



Neural Networks

The Building Blocks

- Neural networks are the foundation of modern language models
- They consist of interconnected nodes (neurons) organised in layers
- Imagine your brain as a neural network, with neurons working together to process language
- Each neuron applies a mathematical function to its inputs and passes the result to the next layer
- Neural networks learn by adjusting the strength of connections between neurons
- Parameters = (weights, biases, values)

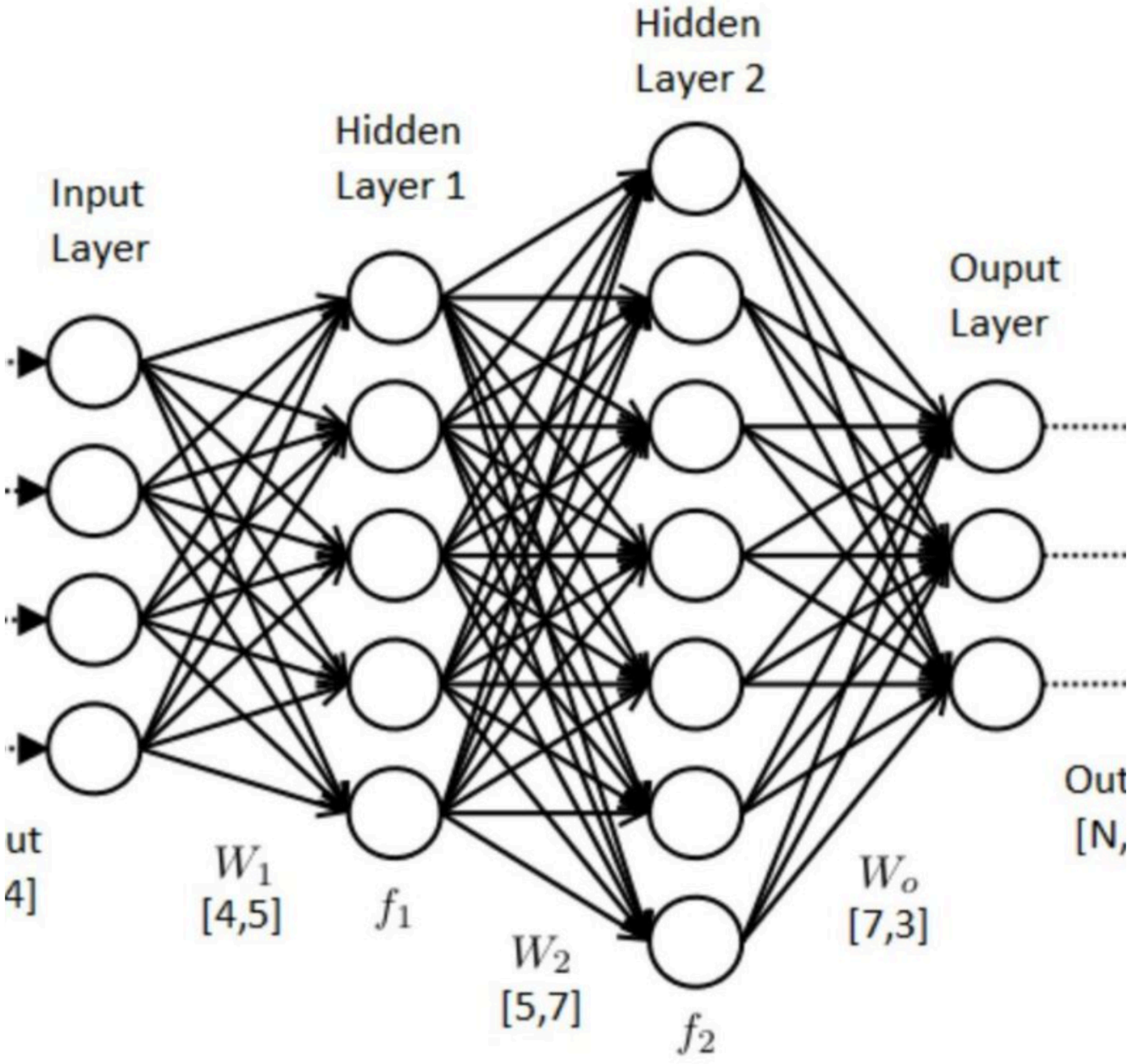
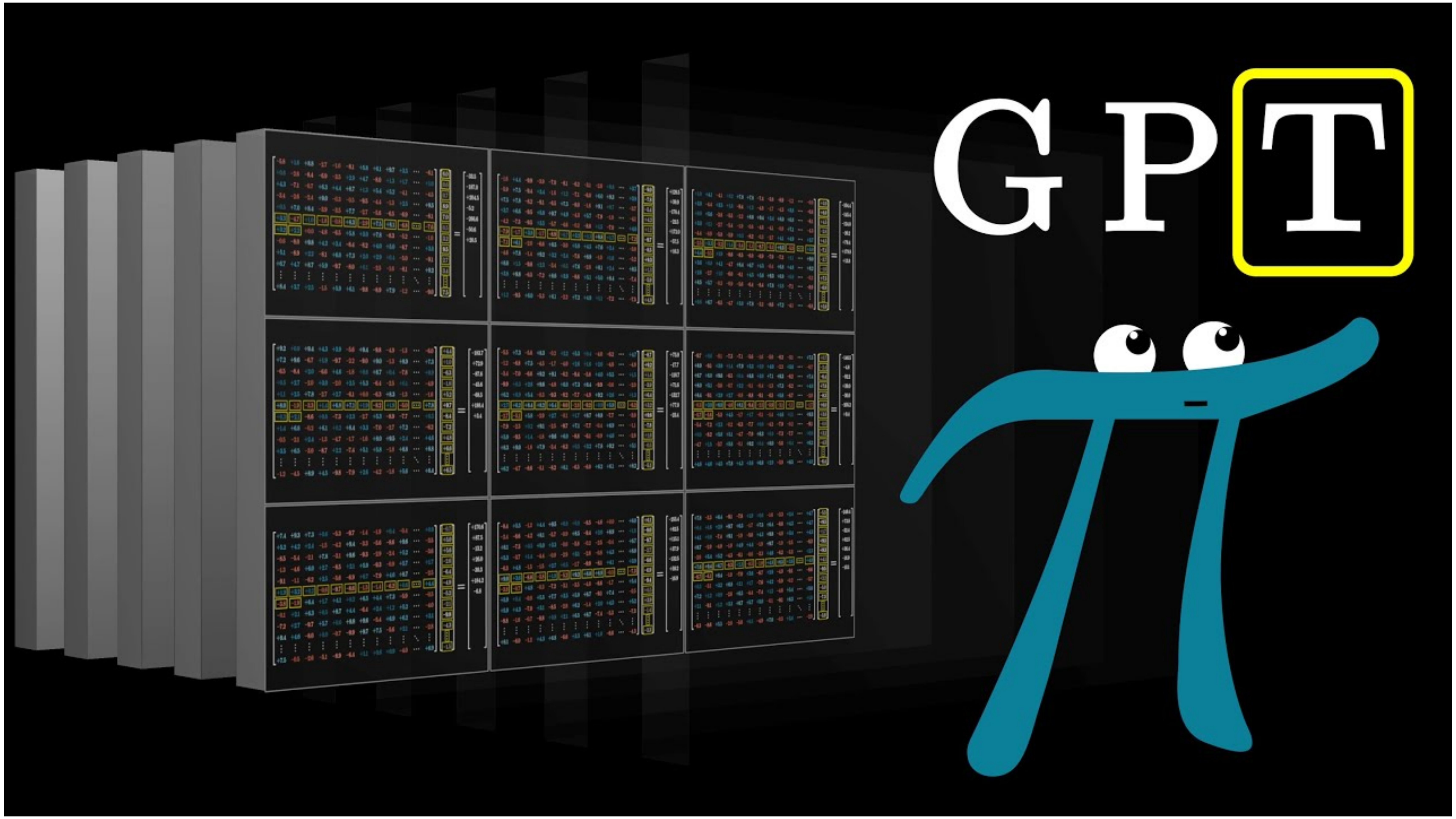


Image: Datasciencecentral

2. Transformers

A specific type of neural network (6 min video)



Recap

Transformers

1. Each sentence is divided into tokens (words)
2. Each token is represented using a vector
3. The LLM predicts the next token given a sequence of tokens
4. This means the last word should encode as much context as possible so that it has enough information to generate the next token.
5. This can’t be done with conventional code. You need a neural network trained on ordinary language that can make connections.

Magar Yeh Hoga Kaise?
— Atal Bihari Vajpayee

2a. Word Vectors

‘You shall know a word by the company it keeps.’ (Firth 1957)

1. Words are represented as a long list of numbers.
2. Why?
    1. Think of coordinates for place names on the globe.
    2. Each word vector is a point in an imaginary “word space” with words closely related to each other being placed close together.
    3. You can do mathematical operations on numbers, not words.
3. The n-dimensional space is tough for humans to visualise but not for computers
4. Go to https://bit.ly/catvec

2a. Word Vectors

‘You shall know a word by the company it keeps.’ (Firth 1957)

1. Word2vec project — ingested all of Google News. A neural network was trained to place all co-occurring words close to each other in an n-dimensional space.
2. You can analogise with these numbers!
    1. wv(biggest) - wv(big) ~ wv(smallest)
    2. wv(king) and wv(man) were just as far from each other as wv(queen) and wv(woman) were.
    3. That’s where bias comes in. wv(doctor)-wv(man) ~wv(nurse)!

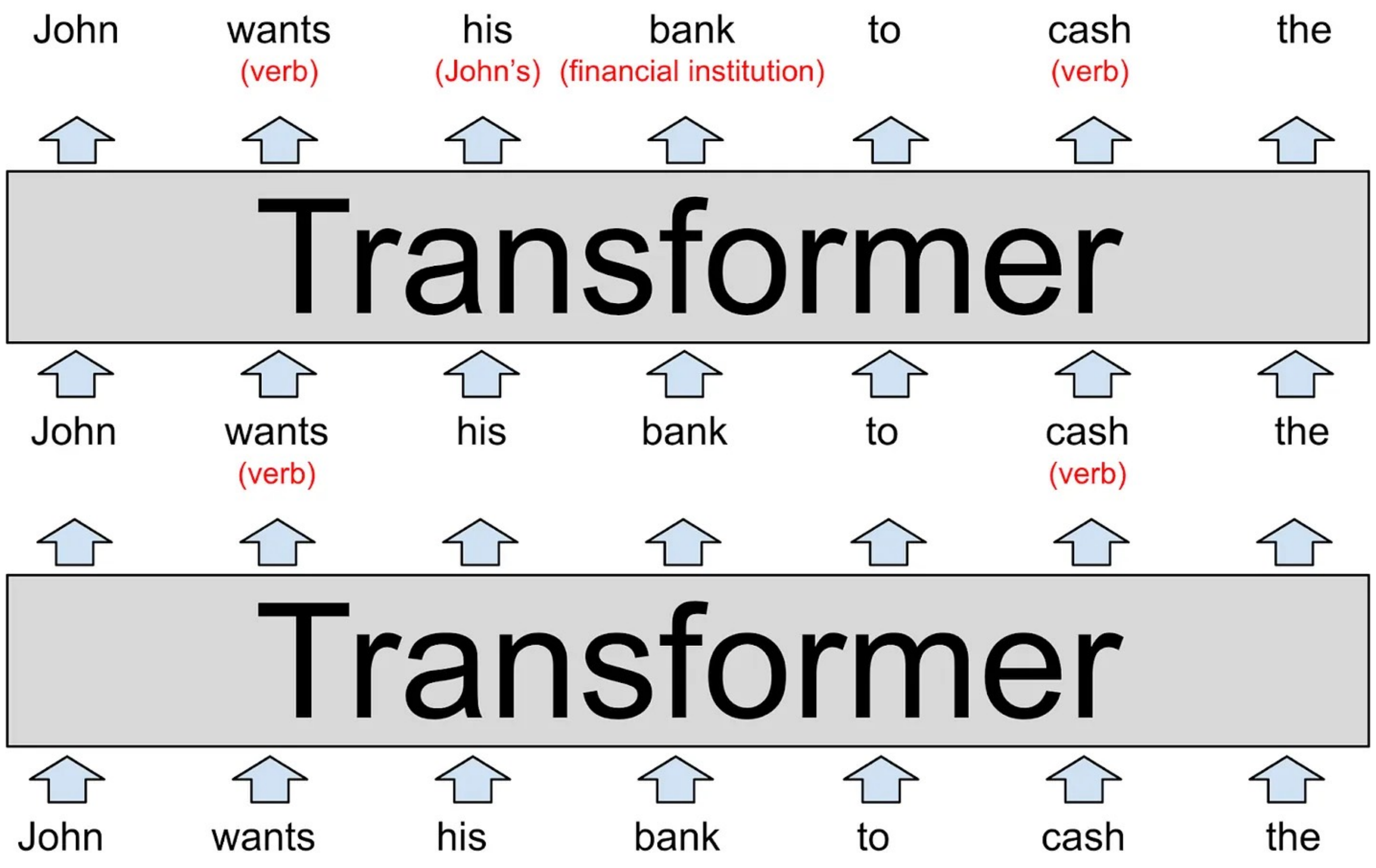
2a. Word Vectors

It Ain't So Easy

1. But same words have different meanings.
2. Natural language is full of complications:
    1. "Anupam asked Satya to take *his* car". His refers to whom?
    2. Same words can have different meanings (Kindle, kindle)
    3. Or the same word can have two closely related meanings - polysemy (eg. The Economist)
    4. Context matters. How to get machines to understand context?

2b. Transform Word Vectors to Predictions

1. GPT has many layers
2. Each layer adds some context by modifying the word vector
3. Aim is to add information to help clarify the meaning of that word and better predict which word might come next.
4. GPT-3 - 96 layers. Each word = 12288 numbers
5. It's a scratch space where notes are taken. By the 60th layer, there will be rich info



Source: understandingai.org

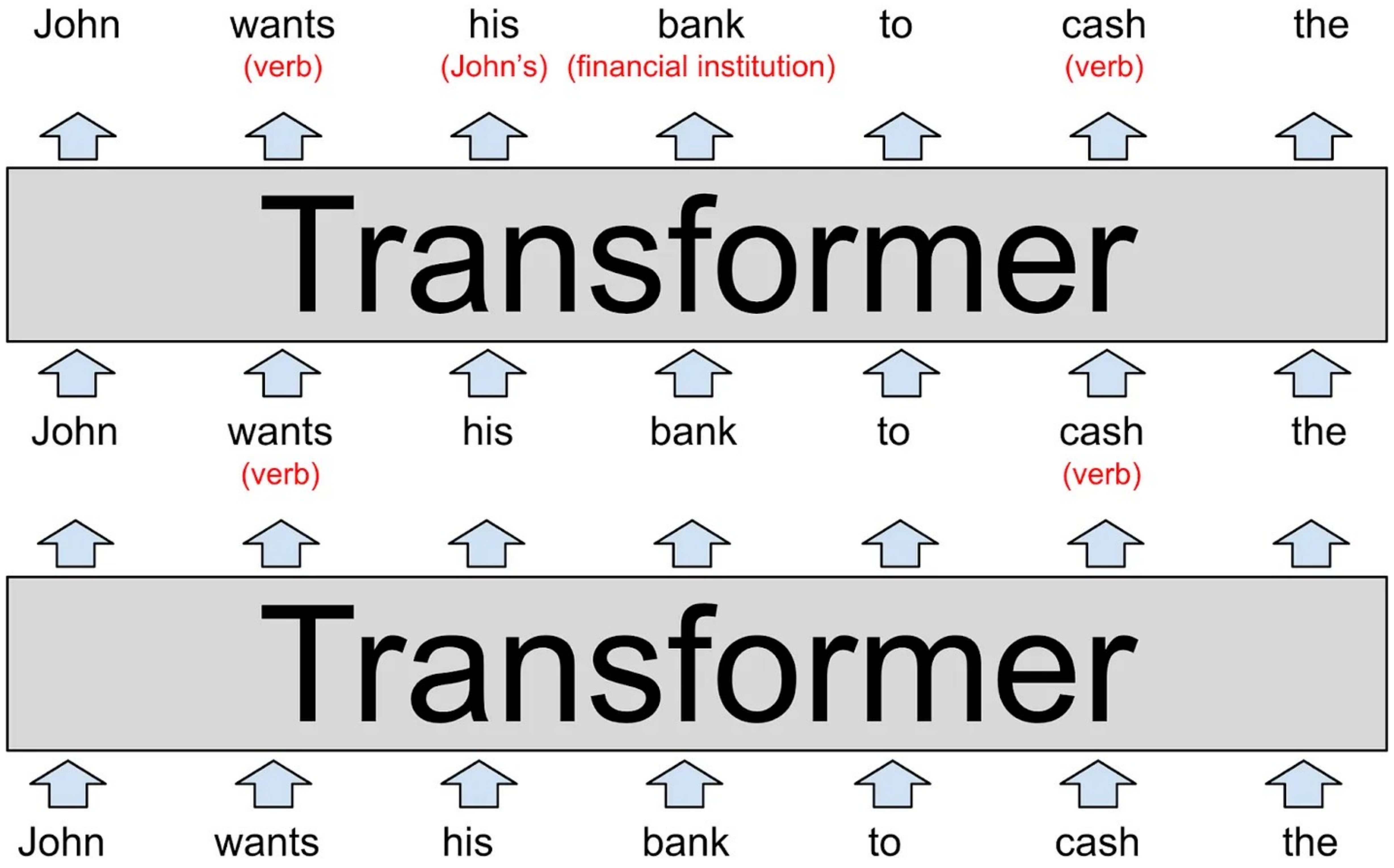
Magar Yeh Hoga Kaise?
— Atal Bihari Vajpayee

2c. What’s Inside the Transformer?

1. Attention Step - Matchmaker
2. Feed-forward Step - Predictor

2c1. Attention

1. Each word searches for context
2. “his” will query “I’m looking for a noun that’s male”
3. “John” will have a key that says “I’m a male person’s name”
4. This information will be stored in the “his” hidden vector



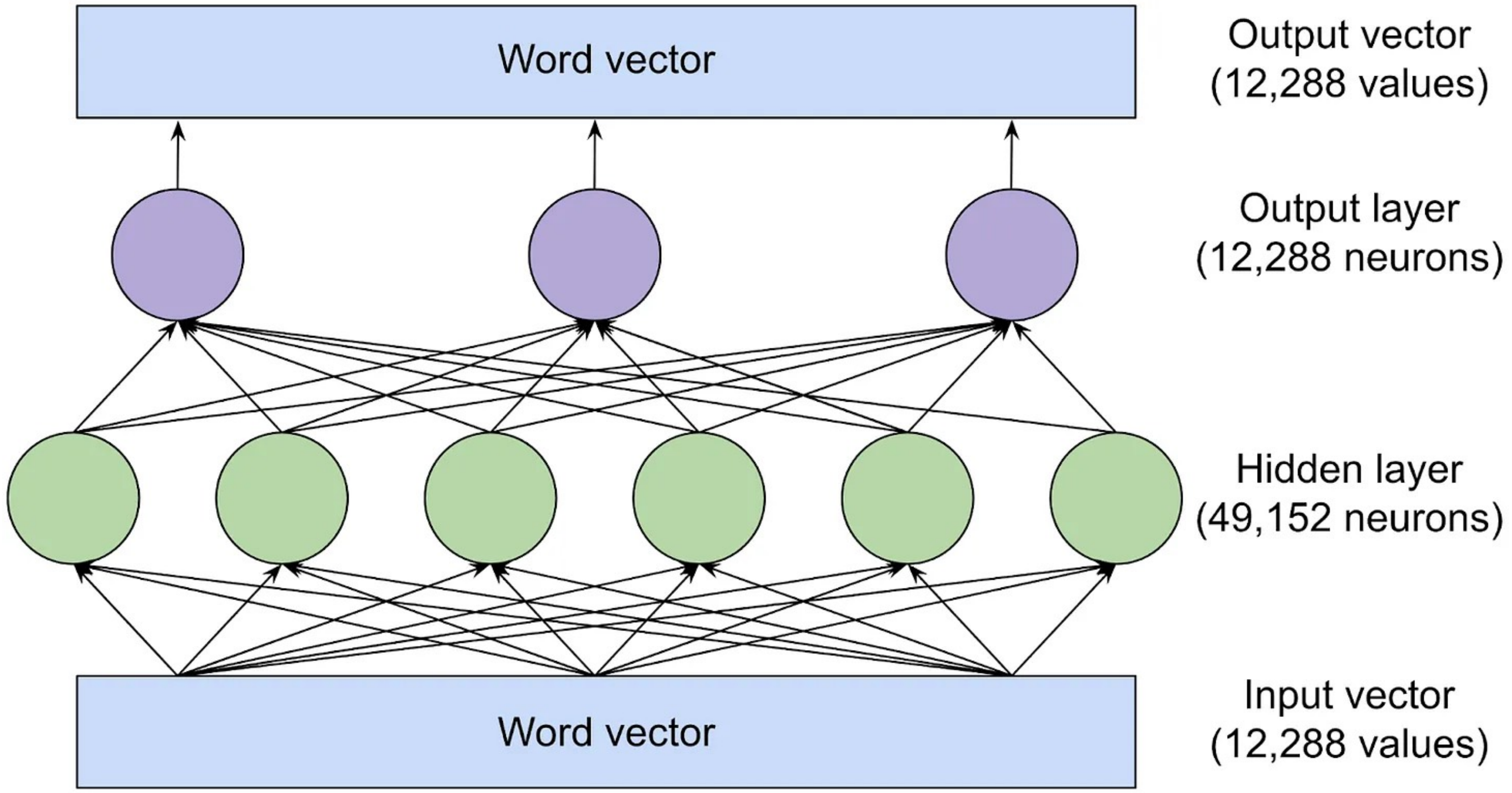
Source: understandingai.org

2c1. Attention

1. Each attention layer has several attention heads. Each head is doing a specific matchmaking
    1. Some are looking for nouns
    2. Some are resolving homonyms etc.

2c2. Feed-forward Step

1. Predict the next word separately based on the model connections
2. It is analogising just as we saw in the word2vec example.
3. If you ask “what’s the capital of India?”, subsequent layers will first predict “India”, and then use the same vector that converts countries to capitals.



Bring’em Together

Attention + Feed Forward

1. Attention is adding richness to the context that’s already in the prompt
2. Feedforward is helping the model “remember” information that’s not in the prompt but is based on training data

2d. How to Train these Transformers?

1. Killer feature - don’t need humans to label content
2. Give a Wikipedia text, it will predict and keep adjusting the parameters. After training with billions of words, it can reasonably reason.
3. Much like what we learned in the neural networks example on Koala, forward passes and backward passes happen.
4. All of this can happen in parallel using GPUs.

3. So can LLMs “Understand”?

1. Some say it’s a stochastic parrot
2. Empirically, bigger models show better reasoning
3. They can identify some abstract features (Anthropic)
4. Language is predictable
5. Far from AGI: scaling has added emergence until now but it is running out of data

References

- understandingai.org
- 3blue1brown YouTube Channel
- TowardsDataScience
- chatGPT
- Claude.ai