

BMDS7010 Supplement: Generative Pre-trained Transformation (GPT)

Learning What ChatGPT is Doing with Stephen Wolfram and Friends

[Stephen Wolfram](#) is a computer scientist, physicist, and businessman. He is known for his work in computer science, mathematics, and theoretical physics.

He has held a **3-hour explanatory walk-through** of Chat GPT [here called What is ChatGPT doing...and why does it work?](#) Its content is indexed below.

===

Wolfram also launched a **six-episode explanatory discussion** of the original talk [here called What Is ChatGPT Doing by Jon Woodard and Giulio Alessandrini](#) – both employees of Wolfram. This is an extended Q&A on Stephen Wolfram's talk so covers neural networks, large language models and the details of how ChatGPT works.

[Episode 1: Introduction](#)

[Episode 2: One Word at a Time](#)

[Episode 3: Neural Networks for Human-Like Tasks](#)

[Episode 4: Training Neural Networks](#)

[Episode 5: Inside ChatGPT](#)

[Episode 6: Why Does It Work?](#)

===

Stephen Wolfram's 3-hour Explanatory Walk-through of ChatGPT

0:09:53 – **Start of presentation, introduction.** The power of ChatGPT was unexpected even to its creators. Why is it a major scientific achievement?

0:12:16 – **Language model definition.** It is trying to continue an initial piece of text and continue it like typical human writing – a reasonable continuation from its prompt in a statistically sensible way. Its writing it one word at a time, what is the next best word from what it has seen from its learning.

0:15:30 – **The “temperature” parameter** – the tweaking of this adjusts the probabilities of inserting a word with 0.8 seeming to work best for essay-type material.

0:17:20 – **Wolfram Desktop demo of GPT2**

He also cites the **article** he wrote to accompany the walk-through at:

<https://writings.stephenwolfram.com/2023/02/what-is-chatgpt-doing-and-why-does-it-work/>

He goes into Wolfram code fairly soon after this and plays with some of the dynamic parameters in the simplified language model (ChatGPT2) being run.

0:18:50 – **generate a sentence with GPT2.** Top 20th-50th words probabilities.

0:25:56 – **unigram model** – looking at probabilities of single letters for specific samples of the English language. Used also for whole books – a larger sample of data, or whole libraries.

0:31:10 – **bi-gram model** – or two-grams – what's the probability for any letter to follow a given letter. e.g. for a letter q, the highest probability is u.

0:33:00 – **n-gram model** – this is the principle of **autocomplete**. You can extend by probabilities of whole words in samples of English text including the whole of the web.

cat for the book flip was generally decided to design of

cat at safety to contain the vicinity coupled between electric public

cat throughout in a confirmation procedure and two were difficult music

cat on the theory an already from a representation before a

0:38:50 – **Why a model is needed?**

You can't sample enough though to make a simple table of probabilities directly so you need a model – you then don't need to have every piece of data as you can fill in the gaps accurately.

0	on gxeetowmt tsifhy ah aufnsoc ior oia itlt bnc tu ih uls
1	ri io os ot timumumoi gymyestit ate bshe abol viowr wotybeat mecho
2	wore hi usinallistin hia ale warou pothe of premetra bect upo pr
3	qual musin was witherins wil por vie surgedygua was suchinguary outheydays theresist
4	stud made yello adenced through theirs from cent intous wherefo proteined screa
5	special average vocab consumer market prepara injury trade consa usually speci utility

0:39:00 – **Definition of a “model”**

0:39:20 – **Early modelling example:** Nice analogy here about a model of Galileo trying to predict how long does it take for a cannonball to hit the ground when dropped from

the Leaning Tower of Pisa – specific times for specific floors (including floor levels that were not measured in real life). In ChatGPT you need a model to find data when it hasn't been measured.

[0:43:55](#) – Printed and handwritten digit recognition tasks

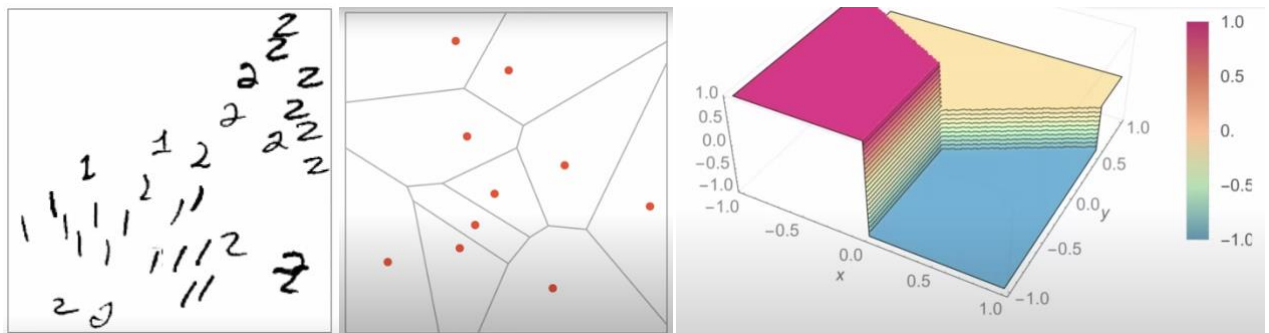


{ 1, 5, 2, 1, 3, 4, 3, 0, 5, 7, 4, 2, 0, 3, 8, 7, 4, 5, 0, 9, 8, 8, 0, 4, 7, 7, 8, 0, 8, 6 }

When we are presented with an array of pixel values, where does it best approximate to? There's no obvious right answer but we estimate it.

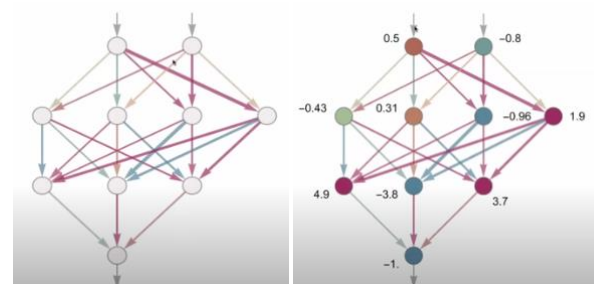
[0:47:40](#) – **Using neural nets to recognize handwritten digits.** This is an idealisation of what's going on in the brain, electrical activity is generated in an elaborate pattern depending on the input.

[0:51:31](#) – **Key idea: attractors** – if we are near the 2 spots we attract to the 2 spots (think of it like water flowing on a range of hills or mountains or “go to the restaurants near to me”).



[0:53:35](#) – **Neural nets and attractors.** In the figure red (excitatory) and blue (inhibitory) flow lines can give massive flexibility to signal representation. The thicker the line the bigger the weight of the connection.

[0:54:44](#) – **Walking through a simple neural net.** The right-hand figure shows the feeding of inputs into the net with representations of particular choices of weights at each neuron. The neural net is just computing the value of the function of x and y – take the weights x values and add a constant offset – outputs a number which gives a thresholding

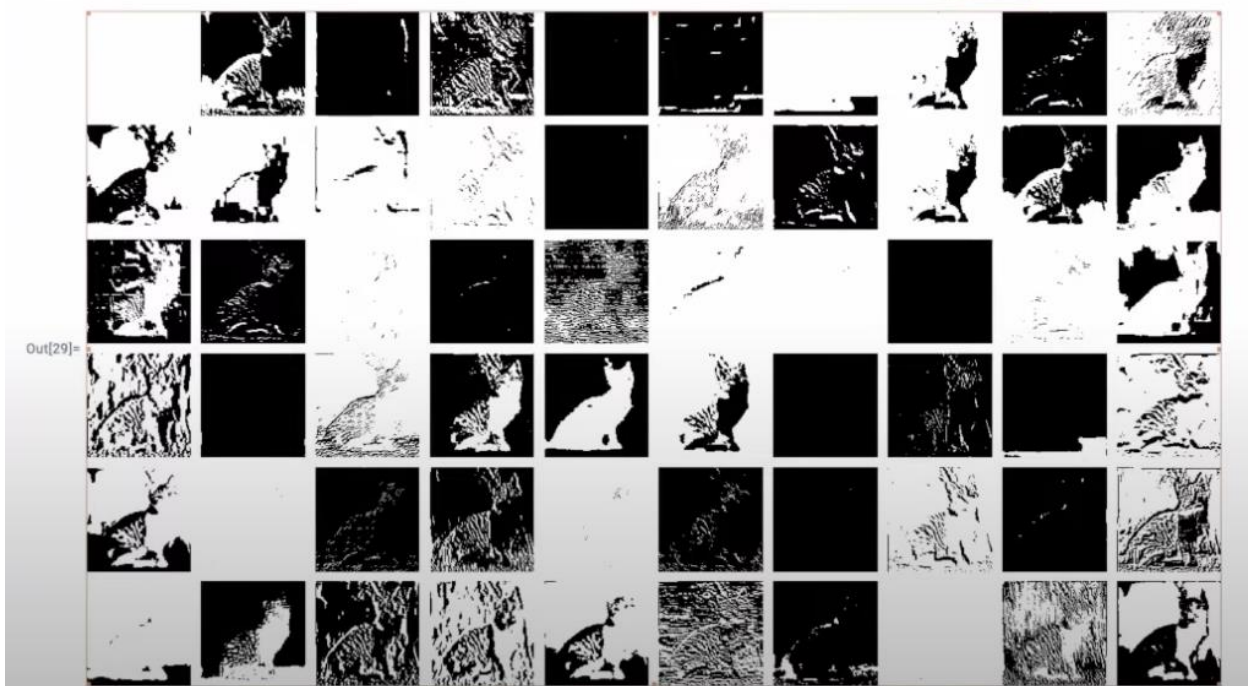


function. There are a few different activation functions and they control whether the output fires into the next layer.

[1:01:50](#) – **What's going inside a neural net during cat classification?** In the example of image analysis of a cat – the software is abstracting different aspects of the image in a similar way to the way the first step in visual processing in brains works.

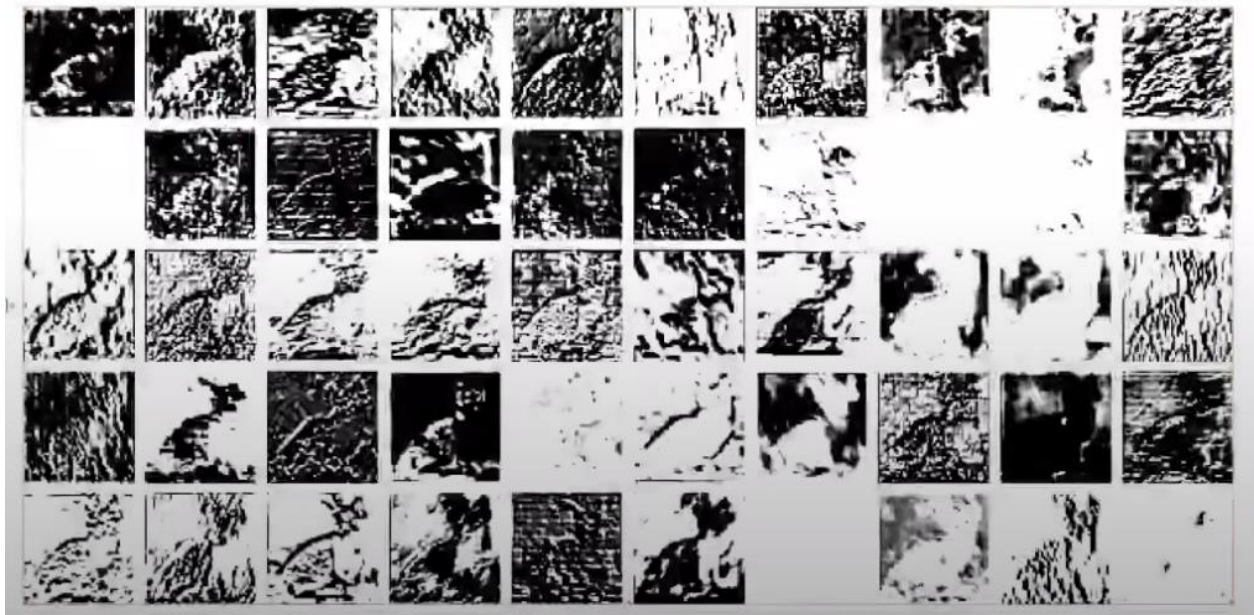


```
In[29]:= GraphicsGrid[Partition[Image /@ Take[NetModel["Wolfram ImageIdentify Net V1"], 1][], 10]]
```



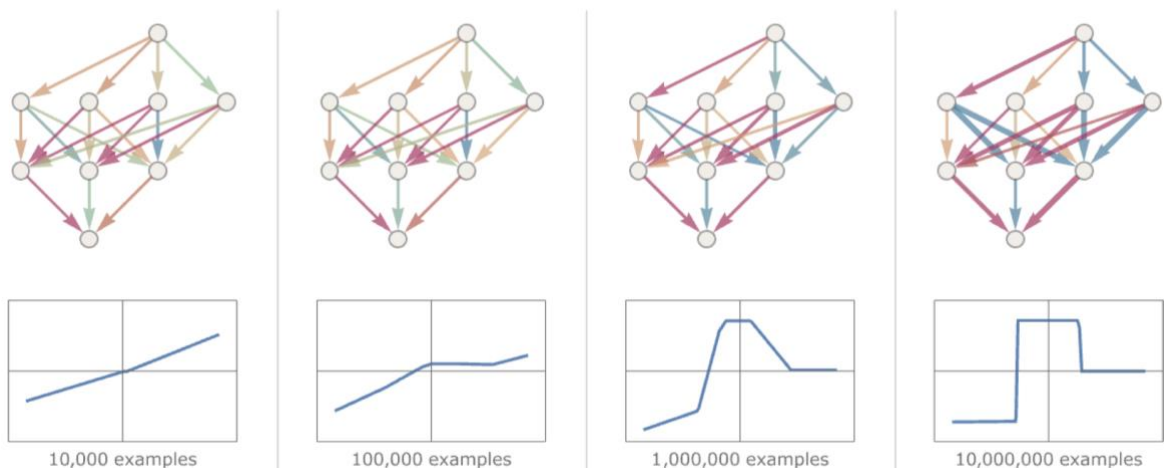
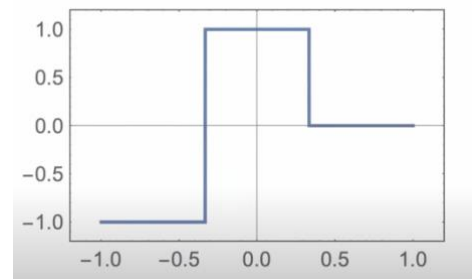
Then ten layers down in the neural net, it is still abstracting features but in ways that humans cannot easily recognise: Its perhaps thinking is it a cat or a dog? Things are much more abstract and so this is why it's often called a "black box" – you can't simply describe or explain what it is doing it discrete ways. This offers a challenge in diagnostic software regulation. There is no simple narrative of how it works. Even humans can't describe recognition easily – for example, explain to your friend how you recognise a

picture is a cat... it has fur, it has pointy ears, it has whiskers, it has feline features etc...



1:06:12 – Training a neural net to correctly compute a function x of a square wave.

This is showing how one can change the weights randomly across increasing numbers of trials in order to compute the function you want. Random inputs will always compute something. For example, if the requirement is to represent a square function (shown on the right) that can be done. The figures below show how random changes of the weights (iteratively) changes the output until it resembles or matches the target output function. The key is to provide examples in learning of the target output – you tweak the weights and below 10 million examples get to the target.



Not correct

Not correct

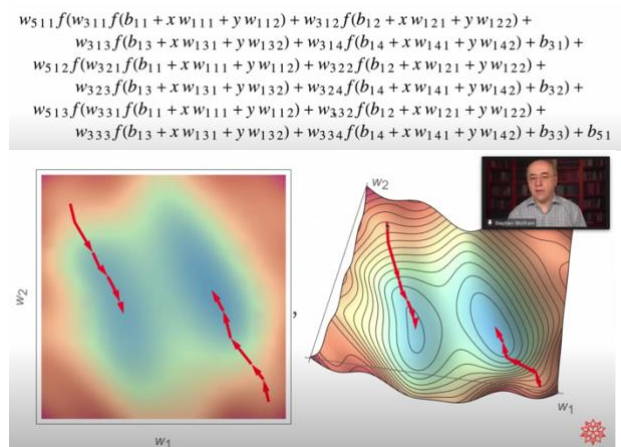
Getting there

The target!

The loss progressively decreases -----> with tweaking weights to reduce the loss to zero

1:09:10 – Measuring “correctness” of neural net with “loss” – as above.

1:10:41 – Reducing “loss” with gradient descent – loss can be computed as a complex algebraic formula, but it is easier to see as an integral plot where the gradient changes and a landscape is produced – you minimise the loss by descending the gradient to the minima point. Note however the process often gets stuck in multiple local minima as shown and as such there needs to be a way of getting out of that.



1:17:06 – Escaping local minima – Note that in higher dimensional space there are many more minima and are distributed in a way that allows escape.

1:21:15 – The generalizability of neural nets - you do not need a different architecture for individual problems.

1:28:06 – Supervised learning – feeding the neural net 10,000 pictures of dogs and 10,000 pictures of cats... annotations to data e.g. titles of audio files can be used as descriptors. If there is no training data, AI won't be able to do a task.

1:30:47 – Transfer learning – this has been important, once you've done a partial learning of features, you can feed that to a second neural net.

1:34:40 – Training LeNet, a handwritten digit recognizer – the handwritten symbol is followed by the label.



[1:38:14](#) – **Embeddings, representing words with numbers** – its best to place each word in some “meaning space” – and have words that mean similar things placed close to each other.

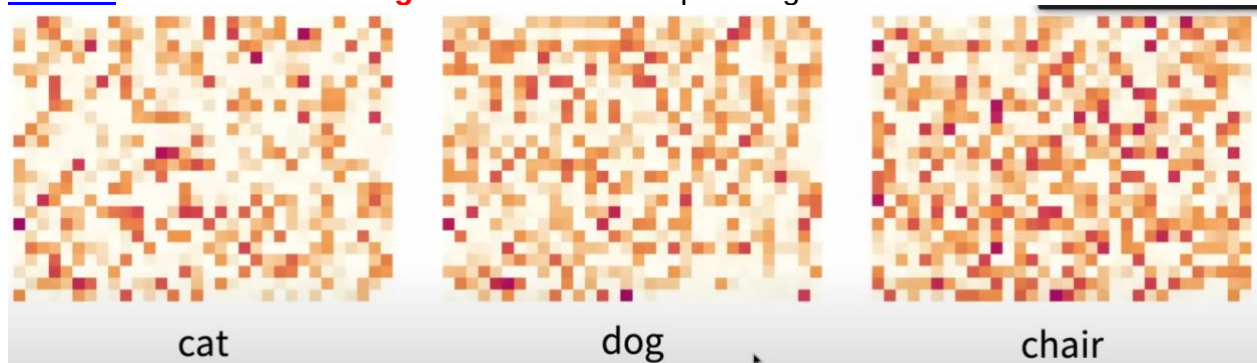


[1:42:12](#) – **Softmax layer** (final layer forcing a decision) and **embedding layer** (penultimate layer making a **feature signature** – the feature vector or “**memory trace**” associated with the input image being recognised).

The **feature signature** will be similar across “4’s” but different across “8’s”



[1:46:22](#) – **GPT2 embeddings of words** - examples might be



i.e. not very informative to humans. However, if we did feature reduction we would find the cat one is closer to the dog one than the chair one.

[1:47:40](#) – **ChatGPT’s basic architecture** –millions of neurons, with 175 billion connections between them.

[1:48:00](#) – **Transformers** - the idea came out of so-called transformers developed out of language translation e.g. from English to French. A convolutional neural net or convonet associates at pixel level. In language transformers the algorithm is paying attention to a word and what its precedents are.

[1:52:50](#) – **Attention block** – all neurons are not connected to all others and one can map which are “attentive” and which not – it works out how much attention it should give to words at each different position – feeds to a 12,000 x 12,000 array layer. The pattern looks like nothing in particular, but it actually



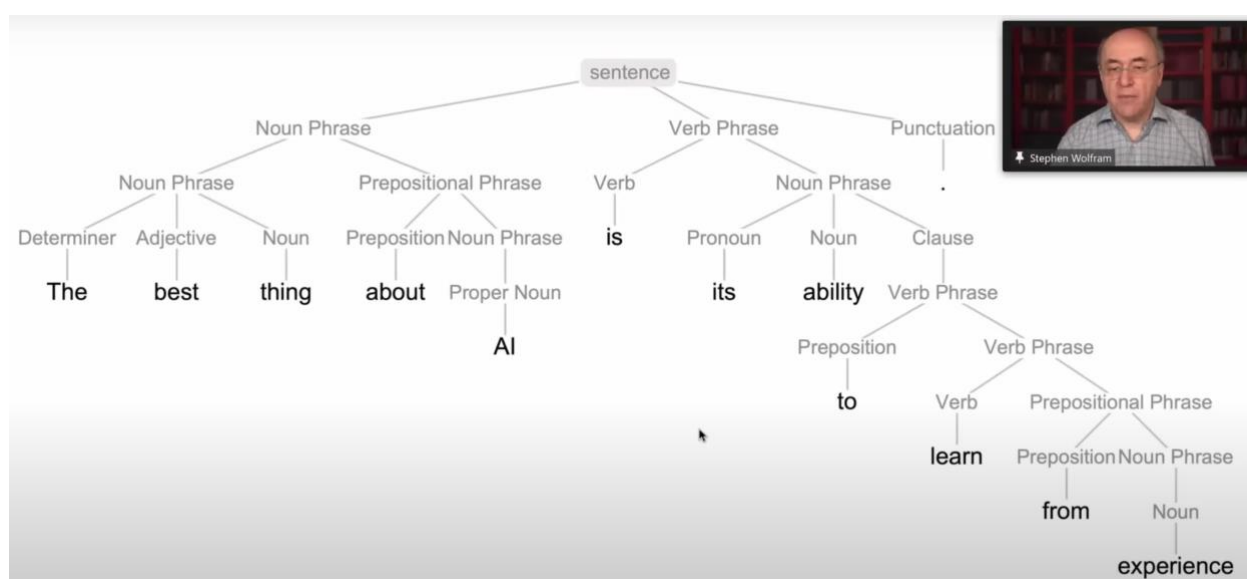
encodes the weights for the neural network that ChatGPT needs in order to continue text.

[1:59:00](#) – **Amount of text training data on the web** – between 6 and 10 billion human-written pages on the web.

[2:03:35](#) – **Relationship between trillions of words and numbers of weights** in the network – for text the number of weights is approximated to the number of training examples in the network. A prompt given to ChatGPT the neural net runs and the text it's got so far in the prompt (as a collection of numbers is the input to the NN) it then ripples through the network layers (about 400 of them) and the output is an array of numbers – they are probabilities for each of the 50,000 words in English – on the basis of that it picks what the next result is. There is a big “outer loop” feedback process as well, where a token from the output is added to the input - so it's not like regular computation which is a fixed set of operations each time. The ChatGPT model does not have inner loops within, it just percolates through once with a complex network. The remarkable thing is that on each percolation it has to use each and every weight (175 Billion – this many mathematical operations) to compute the results. It's probably not necessary to do all of them. But when training it you have to do a reverse version of the forward inference process. It has to be done in reverse in the training. N is pretty large, it often costs hundreds of millions of dollars with current GPUs. In the human brain, most of the brain is both a place of storage and an activity place.

[2:09:40](#) – **Reinforcement learning from human feedback** was essential to perfect autocompletion tasks – the humans rated what came out – this small bit of human guidance was very significant. This step at the end is also where humans say “do not talk about this subject”. The tweaking by humans can evolve the overall future network. Long prompts, with much information can be made use of.

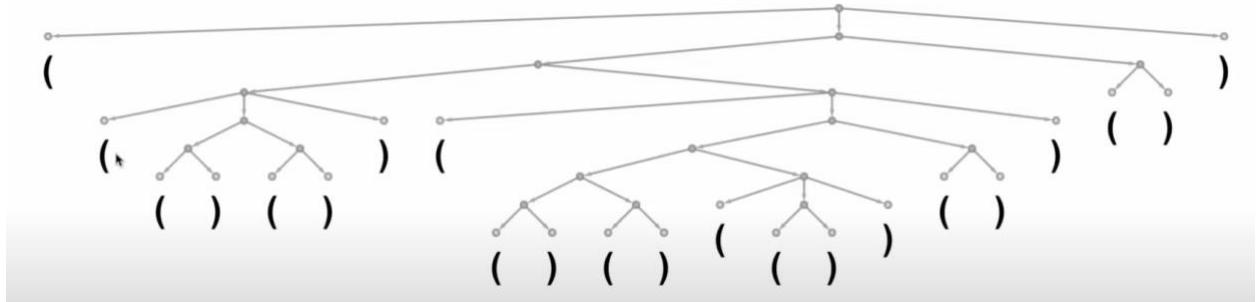
[2:12:38](#) – **Why does ChatGPT work?** It's really telling us a scientific fact – human language has a lot more structure and regularity in it. ChatGPT has learned structure that we didn't know was there.



There are only certain possible forms that correspond to grammatically correct language.

[2:15:50](#) – **ChatGPT learns syntactic grammar**. Syntactic regularity (grammar rules) say you can only put nouns together with verbs in particular ways. ChatGPT has implicitly learnt this by seeing a trillion words of text on the web, mostly which followed correct grammar.

[2:19:30](#) – **ChatGPT's limitation in balancing parentheses** – it sometimes gets the parsing count of clauses wrong.

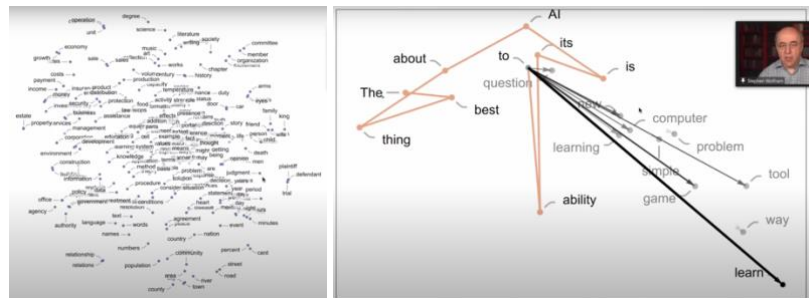


[2:20:51](#) – **ChatGPT learns [inductive] logic based on all the training data it's seen**. Compare the origin of logic by Aristotle. He discovered general meaningful patterns of argument (logic language or forms of syllogism) that could be applied to any argument – this figuring out is a similar “this sentence follows this sentence and so what follows by statistical pattern). Later Boole developed Boolean expressions (AND OR NOT) which ChatGPT doesn't do too well at (as opposed to the reasoning type arguments). ChatGPT has a significant limitation in mathematical computation.

[2:23:57](#) – **What regularities?**

Stephen Wolfram guesses that ChatGPT has discovered – considering “meaning space” from earlier. ChatGPT navigates the meaning space of words – there does seem to be a flow or direction to meaning space – a

characteristic that joins the best connections by related and likely-to-follow words. e.g see example in figure - “The best thing about AI is its ability to...”



[2:36:20](#) – **ChatGPT possibly discovering semantic grammar** – there is a way to use computational primitives to represent the things we talk about in text. This is in the area of “language design” with ChatGPT discovering those relationships (Stephen Wolfram has been trying to design a computational language for the last 20 years – so has special insight on this). He is now convinced it's do-able.

[2:38:17](#) – **There is a fundamental limit of neural nets is in performing irreducible computations**. It can't run certain things with its structures – it turns out like humans find a lot of things to do by thinking (e.g. run computer code in your head). It does work by seeing previous things. It's a form of human interface. It's good at wrapping in human

compatible language as a medium – ChatGPT can do this well. It wraps specifics with the wrappers.

[2:41:09](#) – **Q&A**

[2:41:16](#) – **Question: “Are constructed languages like Esperanto more amenable to semantic grammar AI approach?”**

[2:43:14](#) – **Question: What’s the best way of prompting?**

[2:45:00](#) – **Question: tension between superintelligence and computational irreducibility. How far can LLM intelligence go?**

[2:52:12](#) – **Question: Why did things start happening in 2012?**

[2:53:22](#) – **Question: Will pretraining a large biologically-inspired language model be feasible in the future?**

[2:55:46](#) – **Question: Is it human-like?**

[2:56:25](#) – **Question: Describe the creativity in AI image generation.**

[2:59:17](#) – **Question: How does ChatGPT avoid controversial topics? (taught through reinforcement learning and a list of controversial words).**

[3:03:26](#) – **Question: How does neural AI nets compare to other living multicellular intelligence** – explain the principle of computational equivalence. Deep discussion on why how we can discover the laws of physics.

[3:04:45](#) – **Question: Human consciousness is a consequence of what?**

[3:06:40](#) – **Question: automated fact checking for ChatGPT via an adversarial network.** Train ChatGPT with WolframAlpha so it will improve (yes up to a point). There will always be unexpected thing.

[3:07:25](#) – **Question: Can ChatGPT play a text-based adventure game?** (He bets it can).

[3:07:43](#) – **Question: What makes GPT3 so good at language?** (there is regularity in language).

[3:08:22](#) – **Question: Could feature impact scores help us understand GPT better?** If you run a neural net you can say how much did some feature affect the output?

[3:09:48](#) – **Question: ChatGPT’s understanding of implications of ambiguous language** (the pattern must have occurred before)

[3:10:34](#) – **Question: the human brain’s ability to learn is dependent on its architecture?** (has to be absolutely). Did biology already discover the patterns in

human language? Did biology have the idea of drones or planes? – probably rediscovered, yes.

[3:13:07](#) – Question: how difficult will it be for individuals to train a personal ChatGPT that behaves like a clone of the user? -don't know, but Stephen is going to try it.

===

Next step (after this) if anyone wants to learn more see:

“The Long History of Neural Nets”

Stephen Wolfram had a 3-hour conversation about the history of neural networks with [Terry Sejnowski](#) called [The Long Story of How Neural Nets Got to Where They Are](#).

Sejnowski is the Francis Crick Professor at the Salk Institute for Biological Studies and director of the Computational Neurobiology Laboratory and is the director of the Crick-Jacobs Center for Theoretical and Computational Biology.

This conversation integrates the “computational” aspect of **artificial neural networks** together with the “biological” (neurophysiological) aspect of **real neural networks in the brain** – this needs to be seen as a recent major scientific milestone.

This talk is quite specialised, but if you are interested in this topic, it is a **goldmine of information** and it puts things in historical context.

A breakdown of the chat is in the YouTube comments.