In his initial experiments trained a tiny the network with only a handful of neurons to predict simple sequences of two symbols 1010101001…ABABBAB

In his paper he first notes a problem with traditional neural networks. They had no memory.

so he borrowed from how we think our minds work - which is to have an ongoing ‘state of mind’ which helps us decide our next actions given what we currently observe.

IMAGE HERE - STATE OF MIND

He added a set of memory neurons he called “state units” to the side of the network, and added back connections from the output to the state units. And connected the state units to the middle of the network to create a loop. Finally he connected the “state unit” to themselves - resulting in a state of mind which depended on the past and could affect the future.

What he called a Recurrent Neural Network

—

Another key innovation was how he set up a prediction problem for the network to learn.

He trained a network by hiding the next letter in a sequence

With this approach the **learning signal** is the difference between the network’s guess of the next symbol and the real data.

(known as SELF SUPERVISED LEARNING - *the data acts as both input and output*)

And critically, after training it, he set up the network to GENERATE data.

By connecting the output of the network back on itself, and kicking it off with a single letter, and it would generate the pattern it learned…(autoregressive, generative network)

And he observed that the network would make mistakes, but those mistakes would go away if it was trained on the pattern longer…

He noticed the learned sequence wasn't just memorized…it generalized to new patterns.

In another experiment where he trained the network on a spatial pattern, a sequence of coordinate points, which traced out a square..

After training a network on this sequence, he fed in a point, and plotted the results, and it would correctly continue the cyclical pattern..

However, when he tried to start it at a NEW point outside this path it learned -, the network would follow the same "cycle pattern" but from a different position, at a different scale. And gradually return to a stable sequence (what he called an ATTRACTOR) -

He writes " when a network learns to perform a sequence, it essentially learns to follow a trajectory through a state space…the learned trajectories tend to be attractors"

He viewed an attractor as the generalized pattern learned by the network which was represented in the connection weights of the inner layers.

Looking back from today, we see the start of something big here, but at the time this was a toy problem. Nobody noticed.

—

5 years later another researcher Jeffrey Elman, picked up on Jordan's research and did this same thing with a slightly bigger network (only 50 neurons) and training it on LANGUAGE.

https://onlinelibrary.wiley.com/doi/pdf/10.1207/s15516709cog1402_1

At first he used 200 short sentences.

-

Interestingly he didn't provide any word boundaries, he simply applied the stream of letters to the network 10 times, and at each step trained it to get closer to making the correct prediction of the next letter.

The first interesting thing he noticed was the network learned word boundaries on its own.

He noted an interesting pattern “at the onset of a new word the chance of error is high, as more of the word is received the error rate declines, since the sequence is increasingly predictable…” at the end of a word the error jumps up again, following the same pattern.

Elman notes that according to Noam Chomsky this shouldn’t be possible….how could this little network understand the words semantically?

Chomsky believed that the human brain contains inborn circuitry that is specifically tailored for comprehending concepts and producing language.

But Elman argued his experiments showed otherwise, no special engineering of network structure was required, everything could be learned from the patterns in language.

He notes “it is therefore an interesting question to ask whether a network can learn any aspects of that underlying abstract structure….”

To test this he set up a bigger network with 362 neurons (150 of them acting as context units where learned patterns could be stored). And he created more data, 10,000 two & three word sentences.

(SHOW “man break car” “girl eat bread”)

He found this larger network could predict the next word dramatically better.

To better understand what his network was doing he probed the internal neurons in the context unit as it was processing words. (which took the form of 150 bit vectors), and then he plotted them and compared the spatial arrangement.

What he found the network would spatially cluster words based on meaning, for example it separated nouns which are inanimate (car, rock) and animate (girl, boy), and within these groups he saw subcategorization, for example the animate objects were broken down into “human” and “non human” clusters, and the “non human” cluster broke down into “large animals” and “small animals”...inanimate objects were broken down into “breakable” and “edible”...

And so he emphasizes that this hierarchical interpretation of meaning was captured in an internal representation the network learned itself.

–

Elman also noted that this approach to training neural networks by hiding “the next event” closely aligns with how humans learn.

He referenced an idea that preverbal children begin the process of learning language, by listening and talking along in their mind with a speaker always guessing the next word - and they can learn from their internal mistakes.
-

In a follow up experiment he pushed this theory further and trained a slightly bigger network on more complex sentences - which required holding information about words across a longer distance. Such as the sentence “boys who mary chases feed cats”,

And initially his networks failed to generate these kinds of sentences correctly…as it should have because, as he noted, linguists claim it shouldn't work unless we taught the network the **rules of grammar first.**

But, then he did something very clever, also in line with his view of how humans learn, he tried to “start simple first” and then got more complex.

He did this with 4 phases of learning.
Where first he only trained it on simple sentences with no complex structure,

And at first he found that the network learned basic grammar patterns on its own.
Starting with whether a word is a noun or a verb, plural or a singular.

then he trained it gradually on the more complex sentences - and this time it worked. He wrote

“When the network was permitted to focus on simpler data first, it was able to learn the task quickly and move on successfully to more complex patterns”

—--

He also had a fascinating insight, since we can represent words as points in high dimensional space, then sequences of words could be thought of as a pathway…
And similar sentences seemed to follow similar pathways….

THOUGHTS FOLLOWS A PATHWAY.

—-

It's useful to pause and consider your own mind as a next event predictor, which is always on a 'pathway' of thought, at many levels

—

But for well over a decade none of this research on language models saw the light of day…

—-

It wasn't until 2011, when an important confluence of researchers pushed this experiment ahead on a much larger model and made a bold observation in terms of the connection between **prediction and intelligence.**

"generating text with RNNs", **(hinton, sutskever…)**
http://www.cs.toronto.edu/~bonner/courses/2014s/csc321/readings/sutskever2011icml.pdf

They write..
"The goal of the paper is to demonstrate the power of large RNNs trained on the task of predicting the next character in a stream of text. "

Interestingly, was the (mundane sounding) application they mentioned…

"this is an important problem because a better character-level prediction could improve compression of text files. "

(since…the patterns in the data are compressed into the network weights - and so after training you could throw away the data and reconstruct it when needed)

But then they go on to make a fascinating claim...

More speculatively, achieving the limit in text compression requires an understanding that is "equivalent to intelligence".

This was in line with one theory that biological brains, at their core, are prediction machines.

- If intelligence is the ability to learn…
- **This views learning as the compression of experience into a predictive model of the world.**

They did three separate experiments, they trained a much larger network, with thousands of neurons and millions of connections, to do next letter prediction on the entire English wikipedia, then the collection of NYT articles, and also on a large collection of academic papers in machine learning…

As early researchers had done, after training they had models generate language by feeding their output back into the input, and kicked off the process by providing some starting text prompt, for example: “the meaning of life is…”

“The meaning of life is the tradition of the ancient human reproduction,,,

And to their surprise the output made sense and quote ”had a significant amount of interesting high-level linguistic structure.” Beyond learning words it absorbed sentence structure…(it didn’t spit out memorized data but new sentences)

-

And it was clear the network absorbed the “style” of it’s input. The NYT trained network looked more like a news report….and the academic paper model spit out technical jargon including references in the correct format.

But beyond a few words the pathway of thought fell ‘off course’ and veered into the nonsensical …

“: it is less favorable to the good boy for when to remove her bigger. In the show’s agreement unanimously resurfaced….”

And so clearly learning was happening, but they were hitting the capacity of the network to maintain coherent context over long sequences.

at the end of their paper they claim that. “If we could train much bigger networks with millions of neurons and billions of connections it is possible that brute force along would be sufficient to achieve an even higher standard of performance…”

—

few took this line of research seriously…

But A few dedicated researchers continued to push this effort ahead.

Another key figure is Andrej Karpahy who wrote a blog post titled “the unreasonable effectiveness of recurrent neural networks”. He did the same experiment again but this time on a bigger network with more layers.

His results were even better, more plausible. Notably he trained them on all of shakespeare and noted that “I can barely recognize these from actual shakespeare”

And when he trained on mathematics papers he noted you get “plausible looking math, it’s quite astonishing”

And just like the early researchers he noted how it learned in phases “the model first discovers the general word-space structure and then rapidly starts to learn words, first starting with the short words and then the longer ones. Topics and themes that span multiple words star to emerge only much later”

SCREENCAP HIS BLOG

And he notes “what’s beautiful about this is that we didn’t have to hardcode any of this, the network decided what was useful to keep track of…this is one of the cleanest and most compelling examples of where the power in deep learning is coming from.”

–

This was more evidence that setting up a system with a broad goal of “learning to speak”, would result in a system which could be retasked on arbitrary narrow goals we might ask of it…

—

More evidence came in 2017 when a team of researchers (at a speculative lab called Open AI) built on Karpathy’s work, and setup a larger recurrent network, and trained it on a massive set of 82 million amazon reviews - this training process took a month. The largest model to date.

When they probed neurons in this network, they found neurons deep in the network which had learned complex conceptual concepts.

For example they reported the discovery of “a single neuron within the network that directly corresponds to sentiment” (how positive or negative the input sounded)

They showed that neuron as it processed text, perfectly classified the sentiment of the text

SHOW SCREENSHOT HERE

–

This was striking because at the time this was something industry used commonly and required specialized systems trained on that one task. requiring a dataset of millions of reviews WITH a human label (positive or negative).

And then training a network on this narrow task (using supervised learning)

–

But in this case they didn't do any of that work, the sentiment neuron emerged out of the process of learning to predict the next word.

And to show this network had a good internal model (or understanding) of sentiment, they had the network generate text, and while doing so they forced the sentiment neuron to be a positive vs negative value, and it spit out positive and negative reviews which were entirely artificial, but indistinguishable from a human review.

–
They write
"It is an open question why our model recovers the concept of sentiment in such a precise, disentangled, interpretable, and manipulable way."
–

And that was just one neuron, the network was full of these representations of abstract concepts it learned from the data, as a result of trying to predict it.

SHOW INDIVIDUAL LEGOS HERE

for future directions of their work they mentioned the key next step: data diversity.

They write
"it is unrealistic to expect a model trained on a corpus of Romance and Fantasy books to learn an encoding which preserves the exact sentiment of a review on Amazon"

—

But simply going bigger with more data was hitting a practical limit at the time…

there was still a key problem with RNN's, which is that they processed data serially, one word at a time. And all of the context had to be squeezed into the fixed internal memory (or, context units) –

-

this was a bottleneck, limiting the ability for the network to handle context over long sequences of text. Too much information is being compressed into a fixed length vector. And so meaning gets squeezed out

BOTTLENECK IMAGE

Practically it was apparent when generating long statements with a RNN, it might make sense for a while but after a few sentences it would drift off into gibberish.

This problem led some to dismiss this idea as more of a parlor trick…

MAGIC TRICK IMAGE

Learning long range dependencies was a key challenge faced by the field.

—

An alternate approach to RNN's tried to tackle this problem by simply processing the entire input sequence of text in parallel by passing it through a very wide, fully connected, network.

Parallel processing removed the need for a memory unit. But this requires many layers of depth in order to compensate for the lack of a memory - since it builds up the context of the sentence across multiple layers. This approach is tempting but the resulting network becomes impossible to train.

But in 2017 a groundbreaking paper came out which was focused on the problem of translating between languages. Which offered a solution.to this memory constraint.

The key insight behind their approach was to create a network with a new kind of dynamic layer which could adapt SOME OF its connection weights based on the CONTEXT of the input. Known as a self attention layer.

–

Allowing it to do with one layer what old networks needed several layers to accomplish

–

These self attention layers work by allowing every word in the input to look at & compare itself to every other word, and absorb the meaning from the most relevant words to better capture the context of its intended use in that sentence.

This is done with the addition attention heads which are mini networks inside the layer … (acting as a kind of lens by which words can examine other words)

For example, consider the sentence “the river has a steep bank”

In the self attention layer, the word bank would compare itself to every other word to find conceptual similarities.

This is done by measuring the distance between all the word pairs in a concept space. (distance between their vector representations)

DRAW SENTENCE WITH LINKS

Similar concepts will be closer in this space. And lead to a higher connection WEIGHTING,

For example the word ‘river’ and ‘bank’ both are related in the context of riverbank, and so this would lead to a higher weighting in that context.

This leads to a second operation, a kind of filter.. where a word absorbs meaning from their connections based on the strength of this weighting. (value matrix)

This allows the word bank to adjust its meaning to push towards the concept (or direction) of a riverbank.

And these operations happen across all words at the same time, and so 1 layer allows all words to compare, extract and merge the meaning from all other words!

Leading to a shallower (but much wider) network that WAS practical to train…

And that's why we call them TRANSFORMERS. They take each word and transform its meaning, shaped by the words around it.

–

To get a sense of this parallel process in action let's look at how a transformer network generates music, by predicting the next note. In this visualization each colored line is a different attention head, and the WEIGHT of the line is the amount of attention it gives to each location - notice each attention head looks for different kinds of patterns in the music.

And to select the next note at each step all patterns are taken into consideration.

It's a network architecture that can look at everything, everywhere, all at once. No internal memory needed, its **memory is replaced by self reference within the layer.**

Critically, the "attention is all you need" paper which introduced transformers, was still one foot in the old paradigm, which was a narrow focus only on the problem of translation trained in a supervised way…they were not after a general purpose system which could do anything you asked of it.

—-
But researchers at Open AI saw the result and immediately tried this more powerful transformer architecture on the next word prediction problem at a larger scale not before possible.

—
in 2018 they published a paper which introduced a model called GPT called "Improving Language Understanding by Generative Pre-training" set the entire field on a new course… by moving away from RNN's and using transformers on next word prediction.

In their paper they restated the reasoning behind their experiment,

"We would like to move towards more general systems which can perform many tasks - eventually without the need to manually create and label a training dataset for each one"

They built a much larger network which could capture 100s of words of input (or context) at a time. It had multiple layers, each with a self attention layer followed by a fully connected layer - known as a block.
—

And this time they trained the network using 7000 books from a variety of domains.

The results were exciting. If you prompted this machine with a segment of text it would continue that passage more coherently than before…

But much more importantly….

It showed some capability in answering general questions.

For example you could give it a simple scenario and ask if it was true or false. And more importantly, questions which were not present in the training data.

This is known as zero-shot learning.

This was a remarkable feature. It highlighted the potential of language models to generalize from it's training data, and apply it to arbitrary tasks…

They write, "recent work suggests that task-specific architectures are no longer necessary" - transformers could do anything...this was an still is a shocking claim.

They followed this experiment immediately with GPT2.

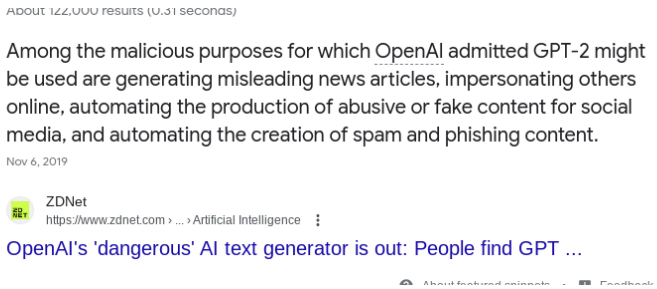
This time they used the same approach but with a dataset scraped from a large portion of the web, 8 million documents. & a much larger network with over 300,000 neurons.

The results surprised the researchers, They tested it on tasks such as reading comprehension, summarization, translation and question answering. The model routinely achieved state of the art performance. better than specialized networks designed ONLY for those specific tasks.

amazingly it could translate languages as good as systems trained only on translation…without any translation specific training!

–

except for a news cycle about its potential misuse use to generate fake news articles or more convincing chat bots…. this development was mostly ignored…even by experts in AI



The problem was, with early experiments GPT2 still drifted off into nonsense after many sentences - it couldn't hold coherence (or context) for long periods of time.

It was still obviously "a trick"...right?

But those researchers now understood this could be solved simply by making everything bigger, especially the context window (or width of the network)

Next they did the same experiment again but they made the network 100x bigger. This model, known as GPT3, had 10's of billions of neurons with 175b connections and 96 layers and a much later context window of around 1000 words. This time they trained it on the entire common web, plus wikipedia as well as several book collections. It was published in July 2020 - what we now refer to as a LARGE LANGUAGE MODEL

Again…It showed increased performance across all measures. Including holding coherence for much longer.

But one capability really jumped out. Once training was complete you could still teach the network new things….

known as IN CONTEXT LEARNING.

In the GPT3 paper researchers showed a simple example where they first gave the definition of a made up word, Gigamuru, and then asked it to use that word in a sentence - which it did perfectly. (this is known as the wug test, and is a key milesone in linguistic development children should aquire by age 7)

-
ADD WUG TEST CLIP HERE

-
Another example was to show the network a few examples of a writing style and ask it to mimic it. Which it could do perfectly.

–
This was just the tip of an iceberg.
–
The key point is we could change the behavior (or goal) of a network WITHOUT changing the network weights. That is, a 'frozen' network can learn new tricks…

–

In context learning works, because it’s leveraging its internal model of individual concepts. Which it can combine, or compose, arbitrarily.

LEGO IMAGES FROM LOGIC VIDEO
-

So you can think of two layers of learning, a “core” in-weight learning which happens during training and then a layer of in-context learning which happens during use (or inference)

LEARNING LAYER IMAGE

(SHOW THE TWEET HERE FROM KARPATHY - “english is the hottest programing language”)

Many pointed out we seemed to have stumbled into a new computing paradigm. Where the computer operates at the level of ‘thoughts’ instead of binary logic operations.

Where a thought is a response to a prompt.

Putting the ‘programming’ of these systems in anyone's hands, via human language.

THE PROMPT IS THE PROGRAM
—

But GPT3 still remained relatively unknown after it’s release.

To enable general use, they took GPT3 and shaped it’s behavior to better follow human instructions, by further training it on many examples of ‘good vs. bad human instruction’ -

This pushed the learning process beyond simply “next word” prediction, and towards “next phrasing”, not only what to say but how to say it.

Known as InstructGPT - which could engage in human conversation much more effectively. Which became the consumer facing product: Chat GPT…

This kicked off the most exciting year of experimentation in Ai history (as 100m + people used the system in public, and reported results in a firehose of surprised)

—

One fascinating observation after its release was the ability to talk to itself…and think out loud.

Because it could “state questions in its output”, which “act as the next instructions” for itself. Creating a reasoning loop.

LOOP IMAGE

A well known paper was published showing that simply adding a phrase such as “think step by step” at the end of your prompt dramatically improved performance of the GPT model.

This kicked off an iterative loop, where the sub-thoughts were written down into meaningful chunks, allowing it to follow a chain of reasoning as long as needed. Resulting in fewer errors.

SHOW EXAMPLES HERE (SCREENSHOT)

—
This led to an explosion of experiments all building on this idea of self talk…

A self driven (or autonomous) Large Langauge Model became known as an Agent -

People then tried to put these Agents in virtual worlds, with no knowledge and gave them tasks (such as “survive in minecraft”) and it would learn to use tools to accomplish this. Talking to itself along the way.

Researchers then applied this “tool use” to the real world, plugging GPTs into external computer systems through API’s - allowing them to make calls, place orders…perform arbitrary tasks.

And finally senses for the physical world such as cameras and actuators….

In fact every task performed by computers could now be re-engineered with an LLM at the core of the process.

–

And when researchers pressed ahead with networks 10x larger on GPT4 & beyond. The trend continued again, better performance on all tasks and improved ability to learn in context.

And today there is a race to build the most capable intelligent agent of them all…the oracle humans have always dreamed of, and feared…

….

Unification:?
This moment marks a possible unification of the field of AI around a single direction. Instead of specialized networks, focused on specific kinds of data (audio, video, text, dna) - all research pointed to treating **all perception as language - that is, a series of information bearing symbols.** And then training networks on prediction powered by networks with self attention. This would lead to more general systems that could then be retasked on any narrow problems.

–
ILLY CLIP HERE - BACH CLIIP HERE TO?
10https://www.youtube.com/watch?v=e8qJsk1j2zE&ab_channel=LexFridman 1h:47min “nobody thought next token prediction was the way forward”
–

It seems that there is something fundamental about the ability to “learn to get better at predicting future perceptions” that is core to learning in biological or artificial neural networks. Imagination is a great survival mechanism because it minimizes surprise.

And since our actions are part of our perceptions, we also learn the result of our actions as a byproduct of this prediction problem.

…

So the question now is, did we invent the core of a tool to end all tools, a potential solution to “mental automation” which is the original dream of Computer Science…

Or is a trick using the largest computer program ever assembled?

Not all agree with this. Some are even insulted…

CHOMSKY CLIP HERE

There is a sharp divide among researchers…and practitioners.

One obvious issue is the gaping holes in the performance of LLM's - often failing at the most rudimentary tasks…

while excelling at advanced problem solving….

A common theme to the failure modes of is the inability for an LLM to break out of strongly learned patterns. But for every "gotcha" example people come up with, others show ways to avoid it with additional prompting. Such as adding "be careful not to get tricked" to a prompt.

Context is everything…this remains an active area of research. With a huge range of uncertainty about the capability of these models.

Even the three godfathers of Deep Learning are not on the same page anymore.

And at the root of this divide is a philosophical question.
…

One group believes these models trick us into thinking they are smarter than they are…by mirrors reflecting our own thoughts in ways we didn't anticipate.

LECUN CLIP

And then the other side believes that if it looks like rational thought then it IS rational thought….

HINTON CLIP HERE

The line between simulating thought and 'actual' thinking is becoming ever more blurred - or perhaps there is no line?

Whatever the case, we find ourselves in a new phase of the Deep Learning era.
—
an esoteric idea dating back to the birth of computing, to build biologically inspired artificial neural networks which can learn everything through experience. By self wiring into patterns which allow them to process their perceptions effectively, to achieve arbitrary goals.

this was, and still is, something people are struggling to get their heads around…

END VIDEO

TODO

- Check use of LLM vs Network vs. System (when do you switch to LLM?)
- decide on internal model vs. understanding vs knowledge
- Including information theory connection intelligent signal = decreasing entropy over sequence length

https://twitter.com/brandon_xyzw/status/1724293854751318382

—

VISUALIZATIONS:

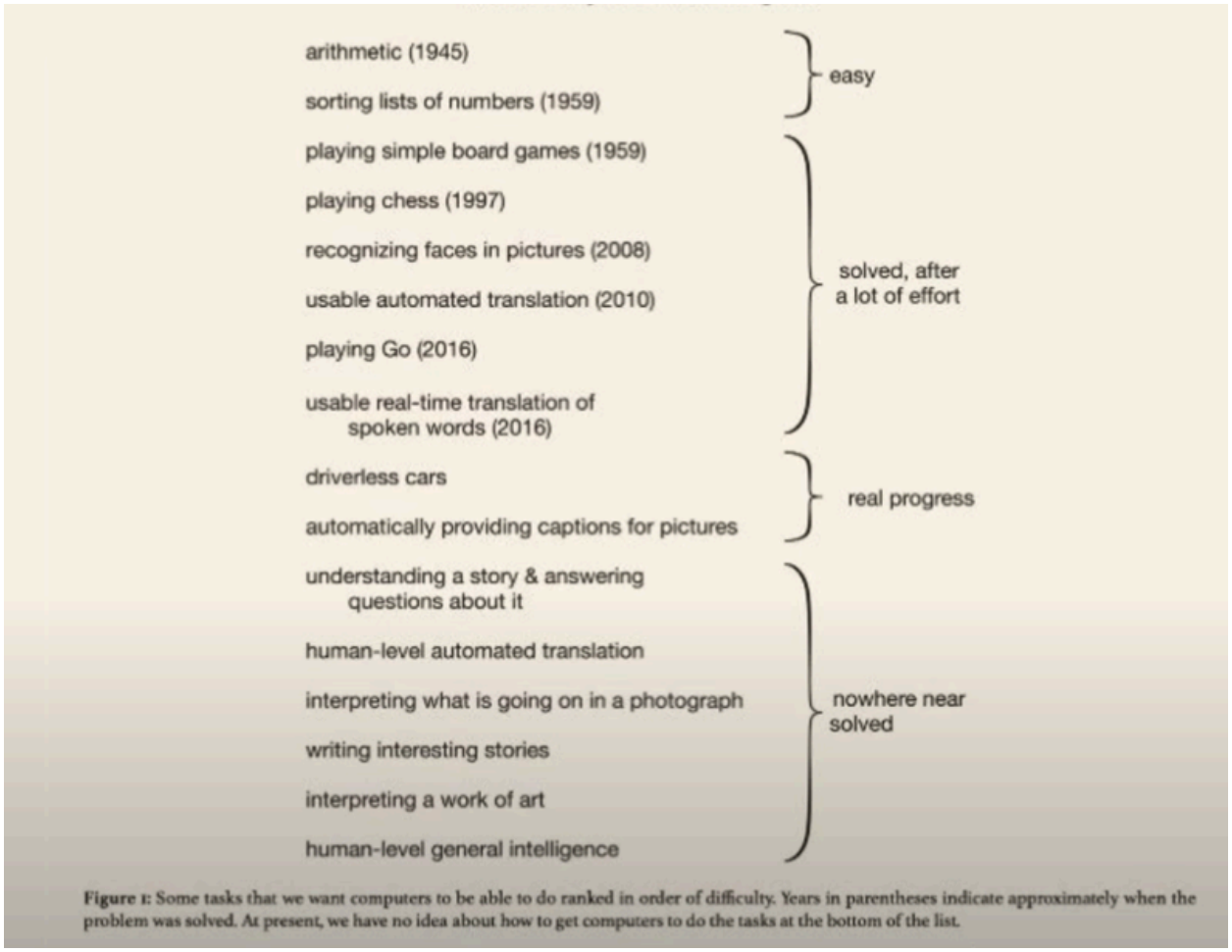
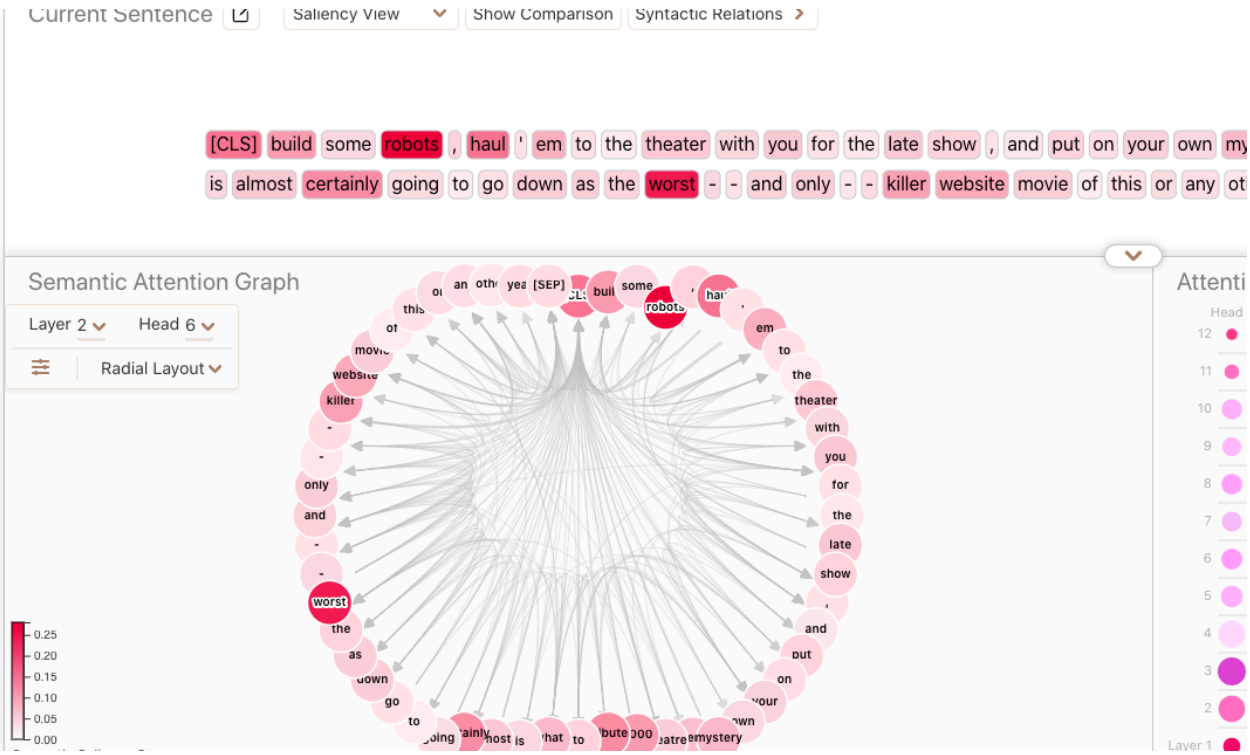


Figure 1: Some tasks that we want computers to be able to do ranked in order of difficulty. Years in parentheses indicate approximately when the problem was solved. At present, we have no idea about how to get computers to do the tasks at the bottom of the list.

Mea



https://poloclub.github.io/dodrio/

One use of attention between RNNs is translation [11]. A traditional sequence-to-sequence model has to boil the entire input down into a single vector and then expands it back out. Attention avoids this by allowing the RNN processing the input to pass along information about each word it sees, and then for the RNN generating the output to focus on words as they become relevant.

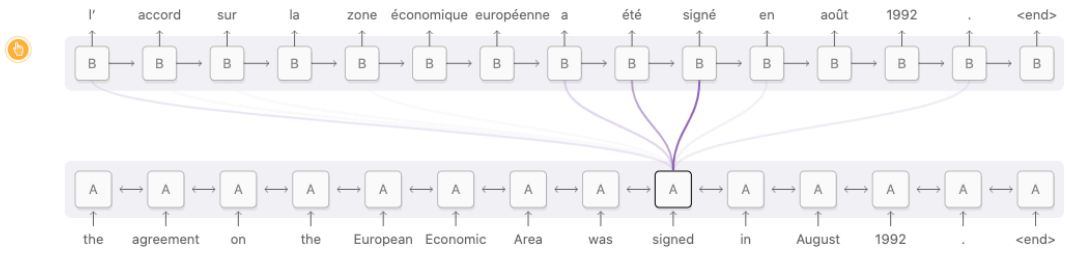
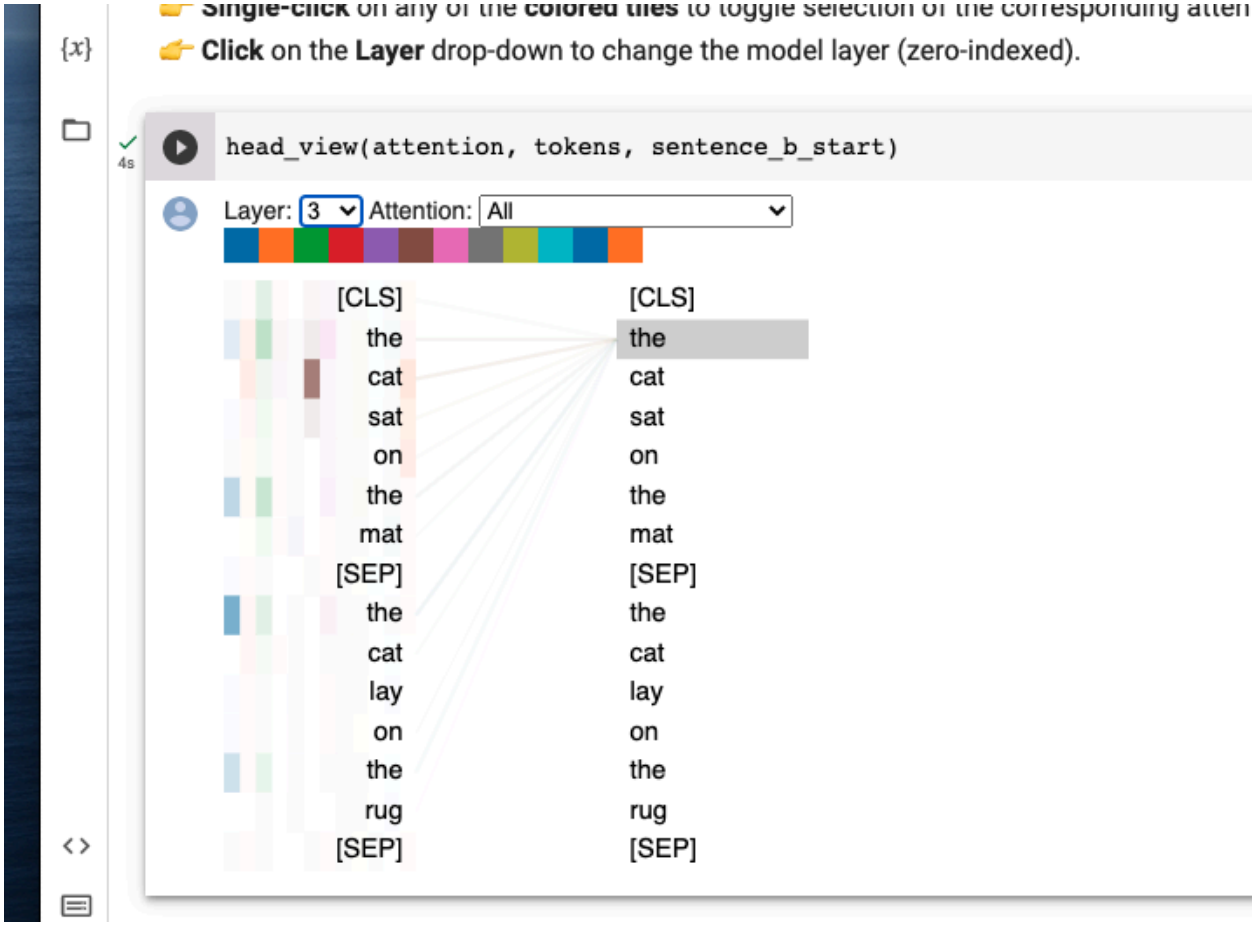
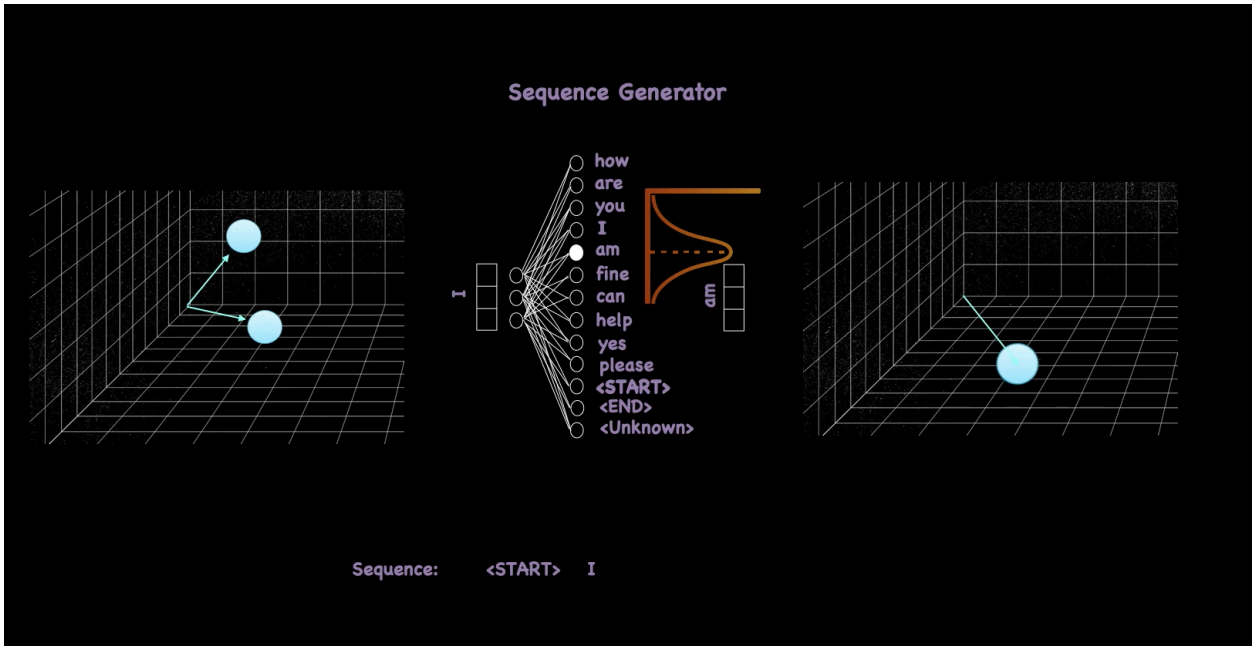


Diagram derived from Fig. 3 of Bahdanau, *et al.* 2014

https://distill.pub/2016/augmented-rnns/

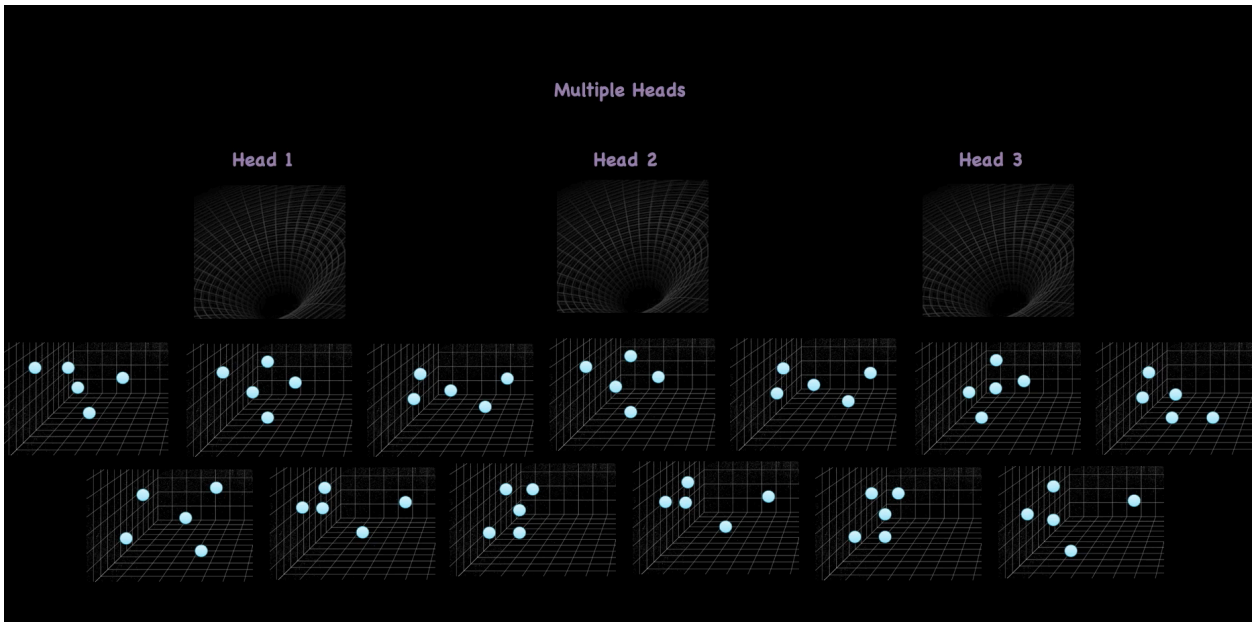


https://colab.research.google.com/drive/1hXIQ77A4TYS4y3UthWF-Ci7V7vVUoxmQ?usp=sharing#scrollTo=twSVFOM9SopW



https://www.youtube.com/watch?v=GTVgJhSlHEk&ab_channel=learningcurve

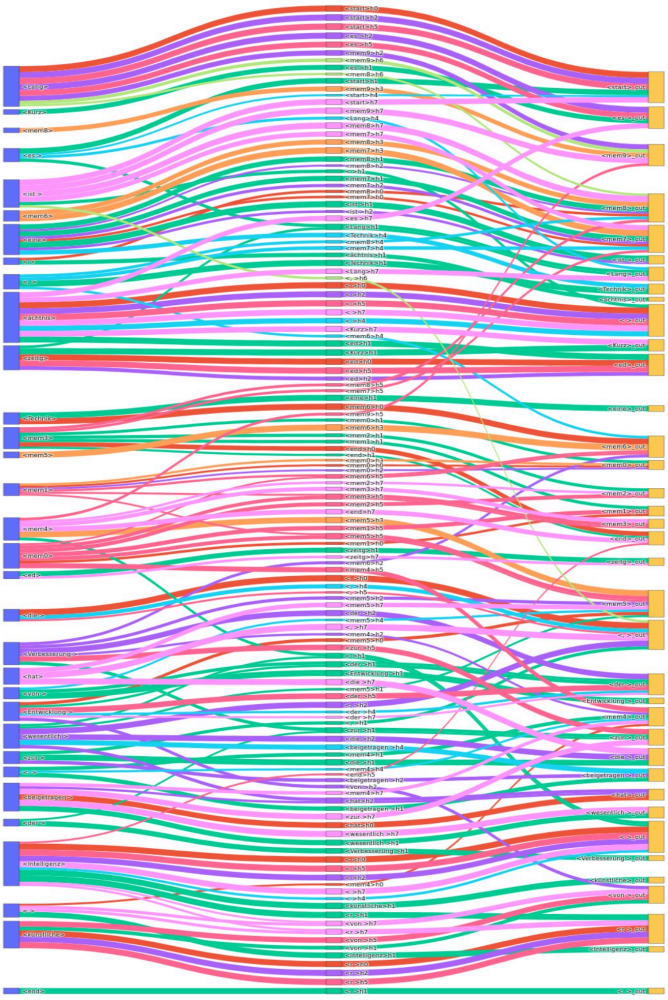
“Attention networks applies a series of transformation to uncover relationships between words in the sentences”



https://www.youtube.com/watch?v=lmepFoddjgQ&ab_channel=learningcurve

https://poloclub.github.io/dodrio/

https://twitter.com/i/status/1475236425578606594



https://www.youtube.com/watch?v=bYmeuc5voUQ&ab_channel=TheResearchandAppliedAISummit-RAAIS **(21 minute mark!!) - here it is**
https://twitter.com/i/status/1073376112724586496

https://twitter.com/ericjang11/status/1475236425578606594