Introduction to AI

HISTORY AND FUNDAMENTALS

J. Langley,

Founder, Huntsville AI

CTO, CohesionForce, Inc

January 14, 2026

INTRODUCTION TO J. LANGLEY

Lots of titles:

- Chief Technical Officer
- Chief Archeologist
- Chief Instigator

I've been working with AI since around 2005, with my master's project at Florida Institute of Technology being an NLP based system for recommending web forum channels that best match a user's question.

I believe that the best way to ensure that AI is used for the greater good is to involve the greatest number of perspectives possible in its development, testing, and use.

HELP WANTED

I need some help with finding a way to assess the “AI Literacy” of a group of people. You happen to be a group of people, and also possess a curious mindset that can help.

Here’s the levels that I made up, so I’m hoping for some feedback:

1. UNAWARE - people that use AI, but don’t know it. Email spam filters, product recommendations, voice assistants, map navigation, etc.
2. ADOPTED - people that use available AI tools as part of their daily life or job. Summarizing text, researching questions, planning, etc.
3. ADVANCED - people that are trying the latest and greatest advancements, possibly in new ways.

THANKS!!!

Now back to our regularly scheduled programming.

This is also a reminder for me to stop and ask if there are any AI topics that you really want to cover during this session or the next session.

I can roll in any that are applicable with the material in this presentation, and I can also build the next session based on your input.

POP QUIZ

What is the first mention in writing of a “Thinking Machine”?

- **1950**
- **1945**
- **1936**
- **1920**
- **1863**

POP QUIZ

What is the first mention in writing of a “Thinking Machine”?

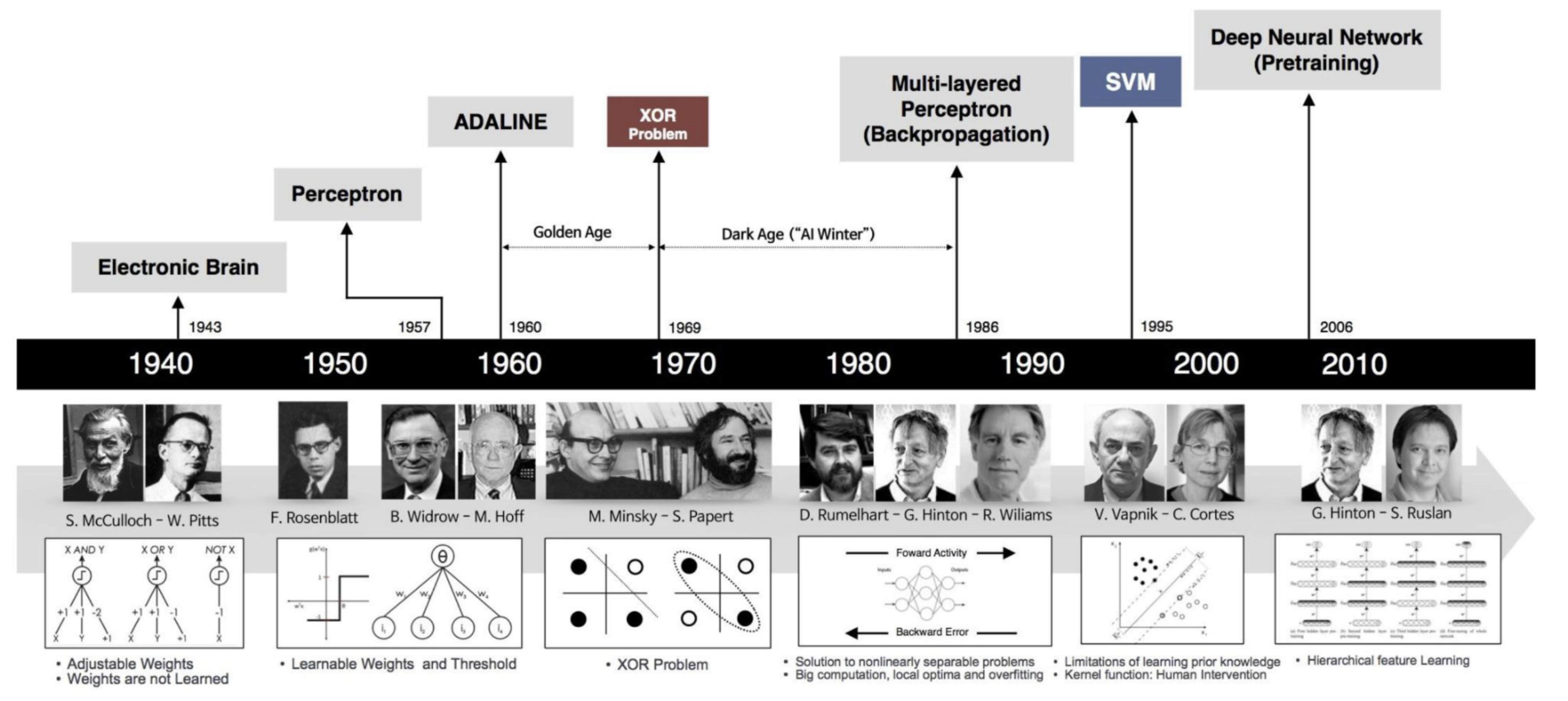
- **1950 - Computing Machinery and Intelligence”, Alan Turing**
- **1945 - "As We May Think", Vannevar Bush**
- **1936 - “On Computable Numbers, with an Application to the Entscheidungsproblem”, Alan Turing**
- **1920 - “R.U.R”, Karel Capek (Rossum's Universal Robots)**
- **1863 - “Darwin Among the Machines”, Samuel Butler**

“Darwin Among the Machines”: The Press, Christchurch, 13 June, 1863.

Our opinion is that war to the death should be instantly proclaimed against them. Every machine of every sort should be destroyed by the well-wisher of his species. Let there be no exceptions made, no quarter shown; let us at once go back to the primeval condition of the race. If it be urged that this is impossible under the present condition of human affairs, this at once proves that the mischief is already done, that our servitude has commenced in good earnest, that we have raised a race of beings whom it is beyond our power to destroy, and that we are not only enslaved but are absolutely acquiescent in our bondage.

HSV.AI

Science Fiction to Deep Learning Timeline



ALAN TURING (1950)

Alan Turing’s "Computing Machinery and Intelligence," published in 1950 in the journal Mind.

The Imitation Game (Turing Test):
Turing proposed the "Imitation Game" (later known as the Turing Test) as a way to operationalize the question, "Can machines think?" The test involves a human evaluator engaging in a text-based conversation with a machine and another human, without knowing which is which. If the evaluator cannot reliably distinguish the machine from the human, the machine is said to exhibit intelligent behavior.

The Turing Test has been a central topic in debates about AI's capabilities and the nature of intelligence.

JOHN MCCARTHY (1956)

The title "Father of Artificial Intelligence" is most commonly attributed to John McCarthy, an American computer scientist. He is credited with coining the term "Artificial Intelligence" in 1956 during the Dartmouth Conference, which is widely regarded as the founding event of AI as a field of study.

We'll run into Mr McCarthy again in a few slides :)

ROSENBLATT'S PERCEPTRON (1958)

The perceptron algorithm was invented in 1957 at the Cornell Aeronautical Laboratory by Frank Rosenblatt, funded by the United States Office of Naval Research.

The perceptron was intended to be a machine, rather than a program, and while its first implementation was in software for the IBM 704, it was subsequently implemented in custom-built hardware as the "Mark 1 perceptron".

This machine was designed for image recognition: it had an array of 400 photocells, randomly connected to the "neurons". Weights were encoded in potentiometers, and weight updates during learning were performed by electric motors.

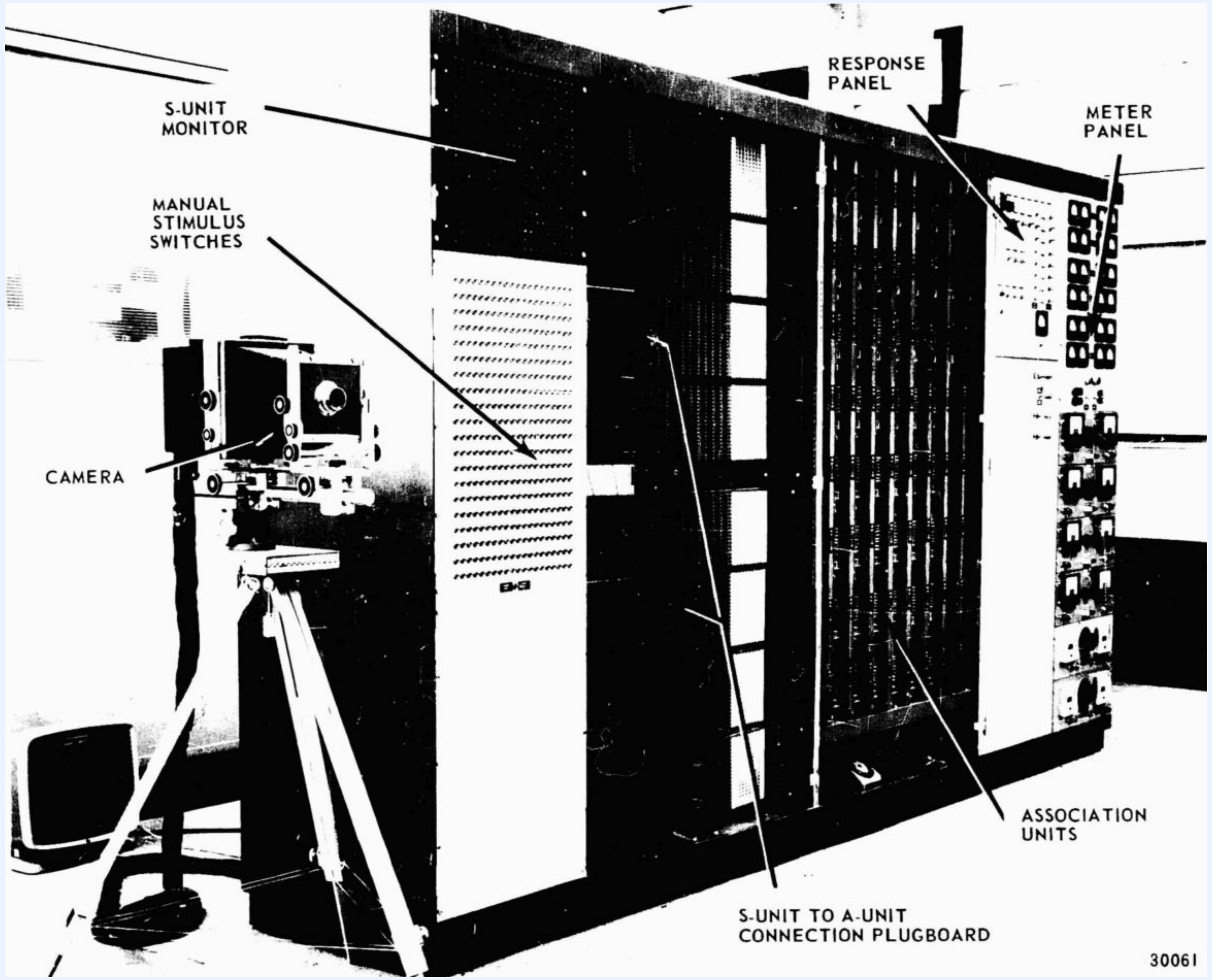
https://homepages.math.uic.edu/~lreyzin/papers/rosenblatt58.pdf

HSV.AI

ROSENBLATT'S PERCEPTRON (1958)

It utilized vacuum tubes as well as resistors, capacitors, and other electronic components.

The perceptron was built with 512 adjustable weights, implemented using motorized potentiometers, which allowed the system to "learn" by adjusting these weights based on input-output relationships.



ROSENBLATT'S PERCEPTRON (1958)

The design of the perceptron was based on how the brain was thought to work. This is the first instance of what would later become neural networks.

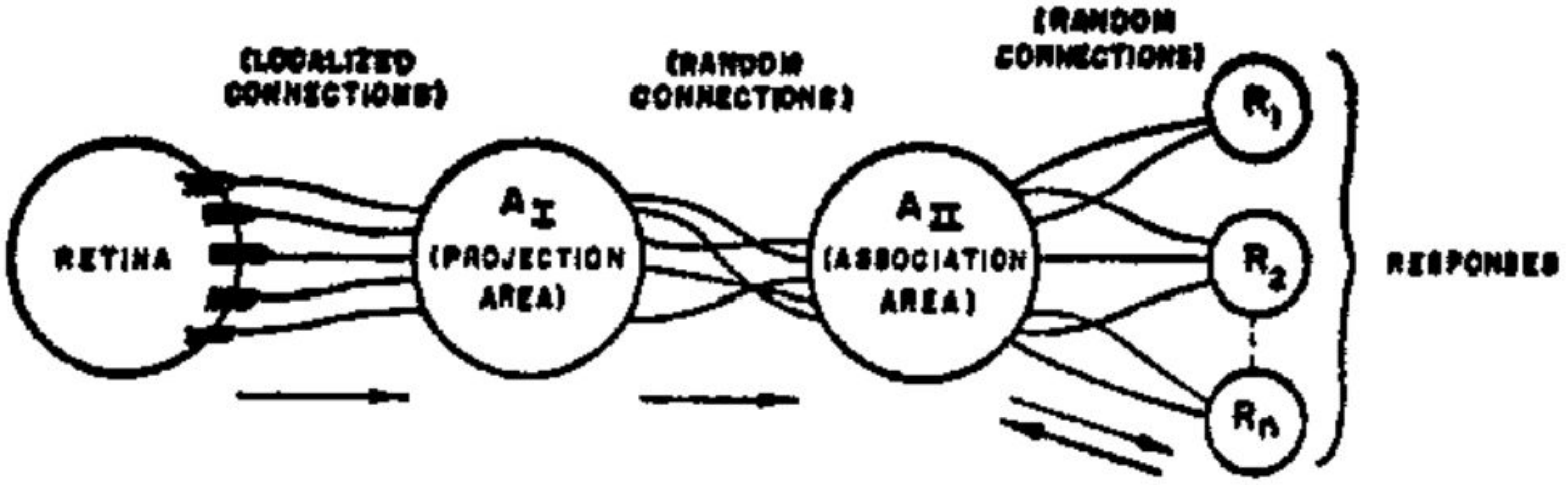


FIG. 1. Organization of a perceptron.

GOLDEN AGE

- General Problem Solver (1959):
  - Developed by Herbert Simon and Allen Newell, it was designed to solve a wide range of problems by breaking them into smaller sub-problems.
- A* Algorithm (1968):
  - Created by Peter Hart, Nils Nilsson, and Bertram Raphael, A* became a foundational algorithm for pathfinding and graph traversal.
- Lisp (1958):
  - Created by John McCarthy, Lisp became the first programming language specifically designed for AI.
- Prolog (1969):
  - Prolog, developed by Alain Colmerauer and Robert Kowalski, became the foundation for logic programming and expert systems.

GOLDEN AGE

- ELIZA (1966):
  - Developed by Joseph Weizenbaum, ELIZA simulated a Rogerian psychotherapist, marking one of the first successful attempts at conversational AI.
- SHRDLU (1968):
  - Terry Winograd created SHRDLU, which could understand and execute commands in a simulated blocks world, advancing NLP and contextual reasoning.
- Automata Theory:
  - Research into automata and formal languages, led by Noam Chomsky and others, influenced AI by formalizing computational processes.
- Logic and Reasoning:
  - Advancements in formal logic, particularly predicate calculus, became integral to AI's ability to model knowledge and reasoning.

GOLDEN AGE CHALLENGES AND LIMITATIONS

Despite these advancements, the golden age of AI faced significant obstacles:

- **Hardware Limitations**: Computers of the time were slow and expensive.
- **Theoretical Challenges**: Early models, like perceptrons, had clear limitations.
- **High Expectations**: Public and academic expectations often exceeded what AI could deliver, leading to disillusionment by the late 1960s and the onset of the first AI winter.

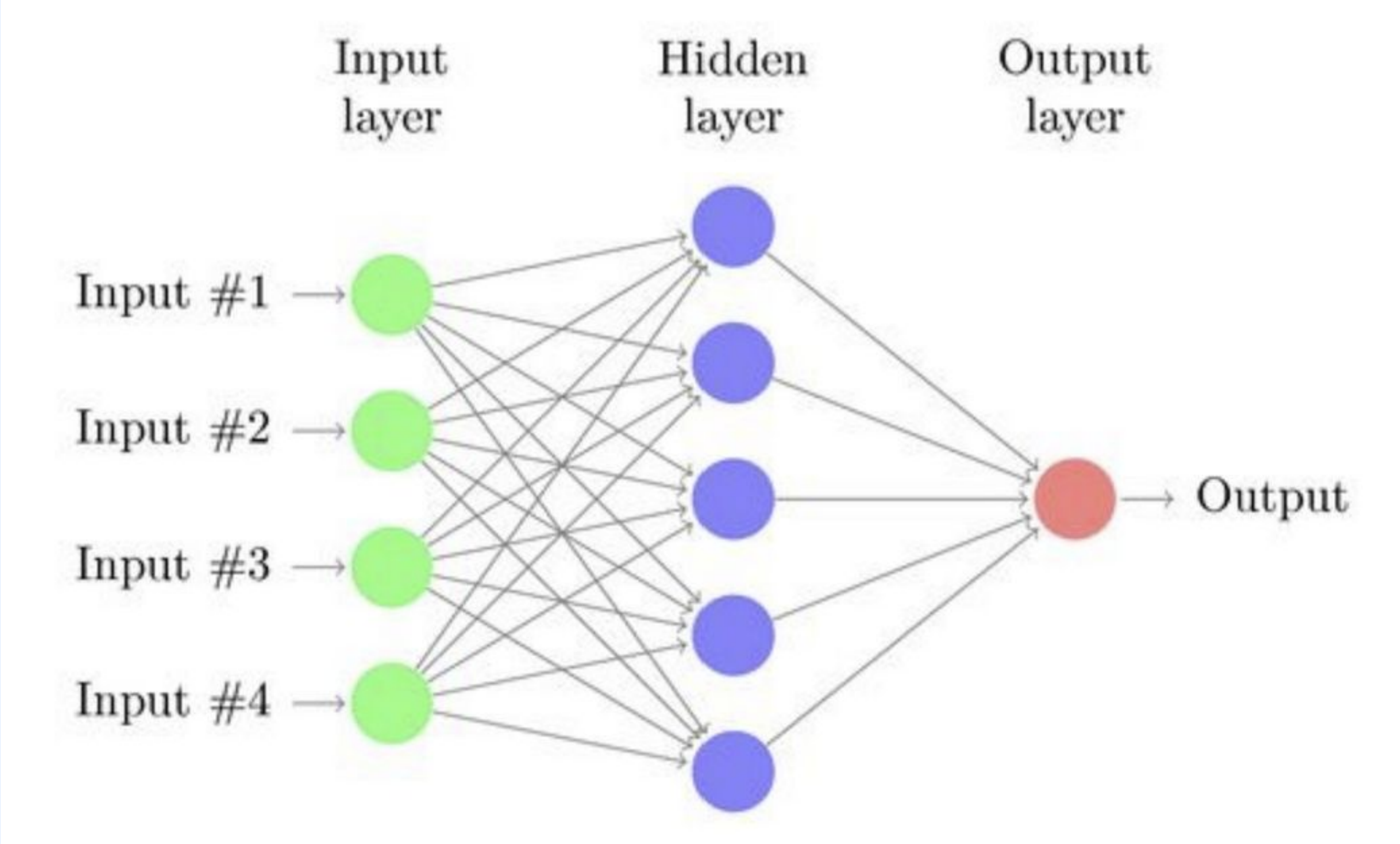
AI WINTER (1969)

Perceptrons: an introduction to computational geometry by Marvin Minsky and Seymour Papert - published in 1969.

It offered a mathematical proof that the perceptron could not approximate an XOR function given an infinite training set.

MULTI-LAYER PERCEPTRONS

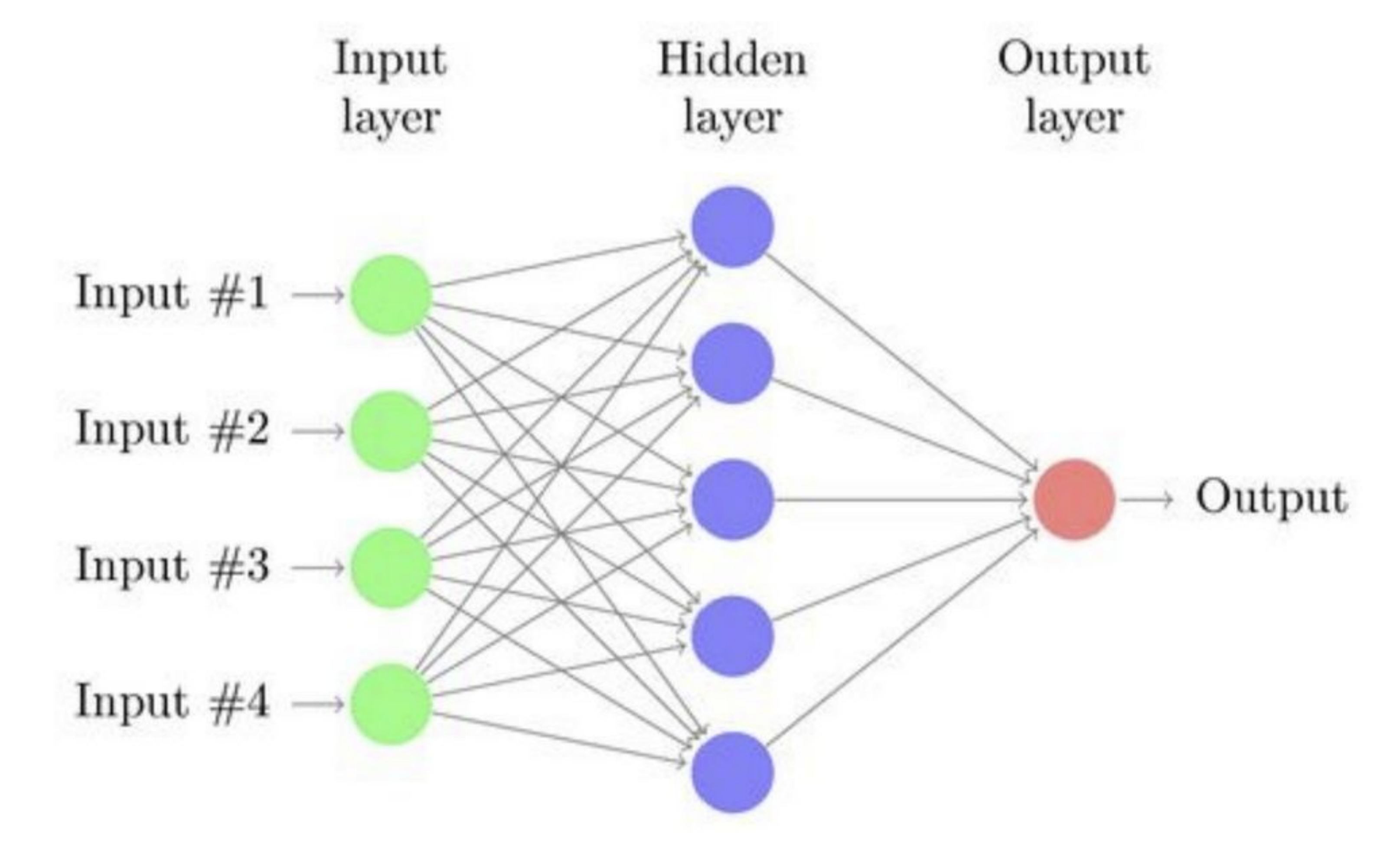
By stacking several layers of perceptrons, researchers were able to overcome the XOR problem.



But it was nearly impossible to train!

BACKPROPAGATION (1986)

Geoff Hinton, along with David Rumelhart and Ronald Williams, published a paper entitled "Learning representations by back-propagating errors"

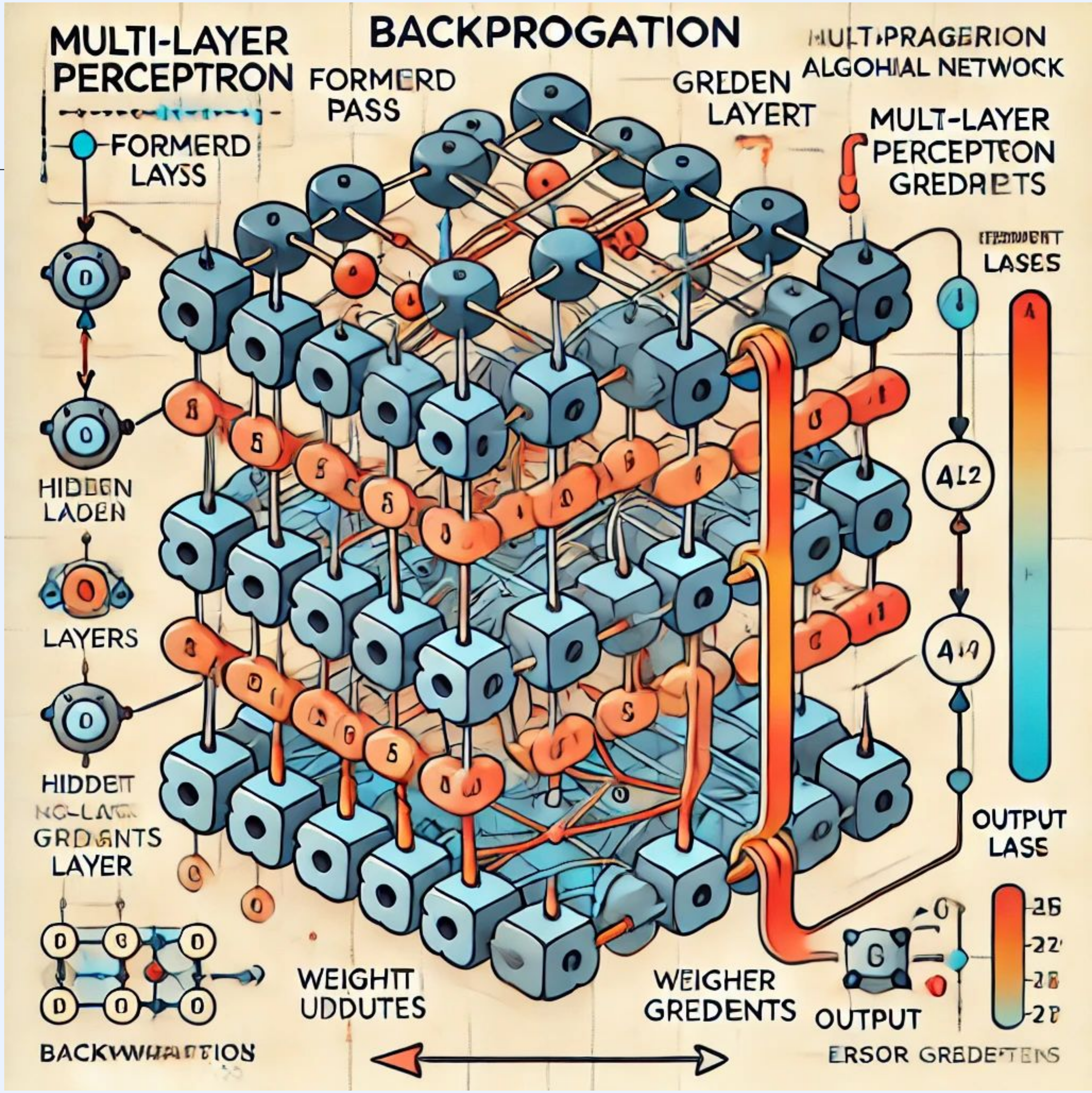


Loss is computed at the output based on difference between it and the known answer.

Weights are updated moving backward through the neural network.

Caution - there's a lot of math here

BREAK



MULTI-LAYER PERCEPTRON
BACKPROGATION
FORMERD PASS
GREDEN LAYERT
ALGOHIAL NETWOCK
FORMERD LAYSS
MULT-LAYER PERCEPTEON GREDRETS
LASES
HIDDEN LADEN
LAYERS
GRDENTS LAYER
OUTPUT LASE
WEIGHTT UDDUTES
WEIGHER GREDENTS
OUTPUT

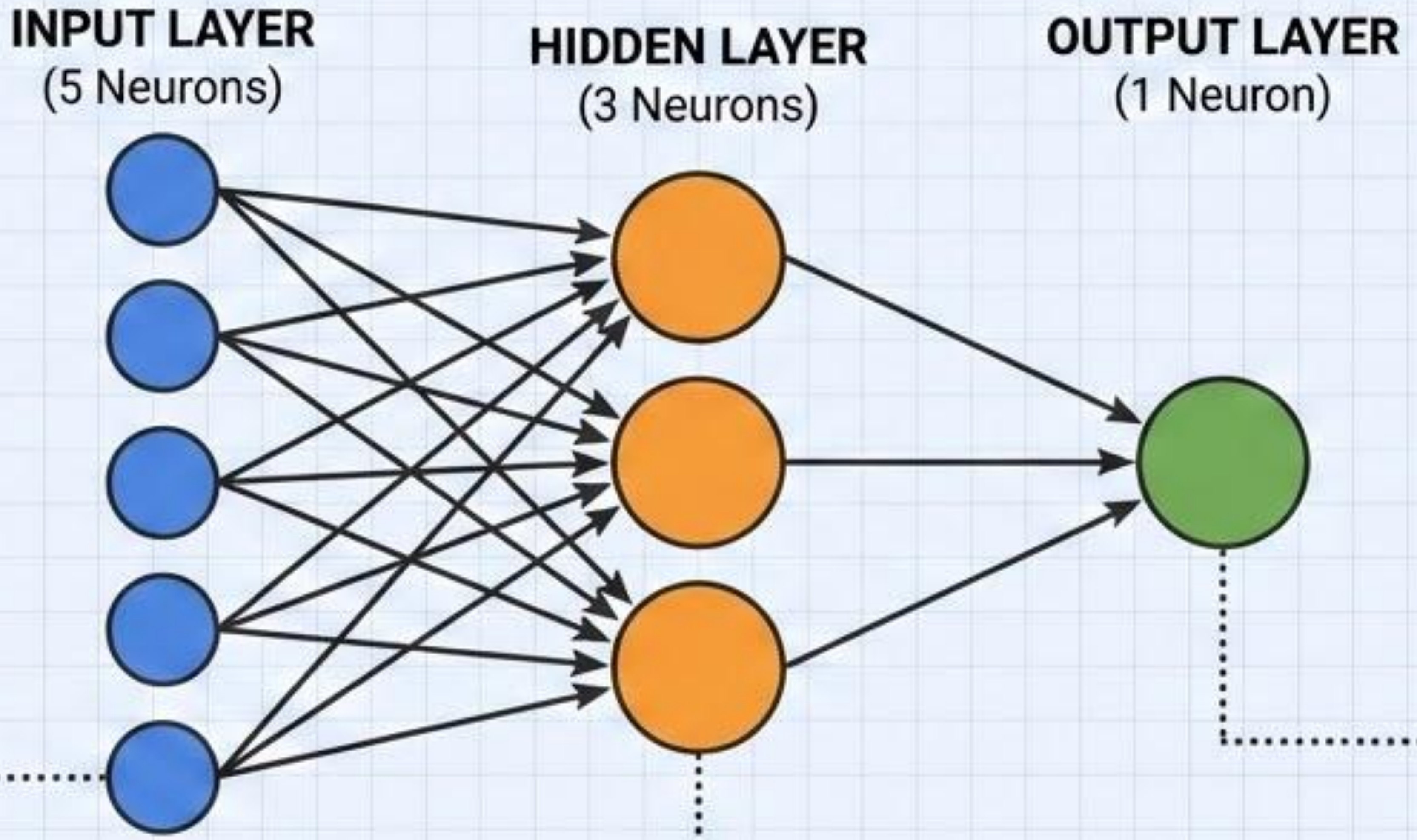
HSV.AI

UNIVERSAL APPROXIMATION THEOREM (1989)

The **Universal Approximation Theorem** is a fundamental result in the theory of neural networks. It states that a feedforward neural network with at least one hidden layer, a sufficient number of neurons, and a suitable activation function can approximate any continuous function to any desired degree of accuracy, given appropriate weights and biases.

Generally credited to papers by George Cybenko (1989) and Kurt Hornik et al. (1989)

NEURAL NETWORK: FEEDFORWARD ARCHITECTURE



1. DATA FLOW: Input features (x_1, x_2, x_3, x_4, x_5) are fed into the network.

2. PROCESSING (Weighted Sum & Activation): Each hidden neuron computes a weighted sum of inputs plus a bias ($w_1x_1 + ... + b$), then applies an activation function (e.g., ReLU) to introduce non-linearity.

3. PREDICTION: The output neuron processes the signals from the hidden layer to produce a final prediction or classification ($\hat{y}$).

HSV.AI

LEARNING BEFORE "DEEP" (1998)

Gradient Based Learning Applied to Document Recognition - Yann Lecun -

CNN from Yann Lecun (AT&T Bell Labs) could recognize handwritten digits.

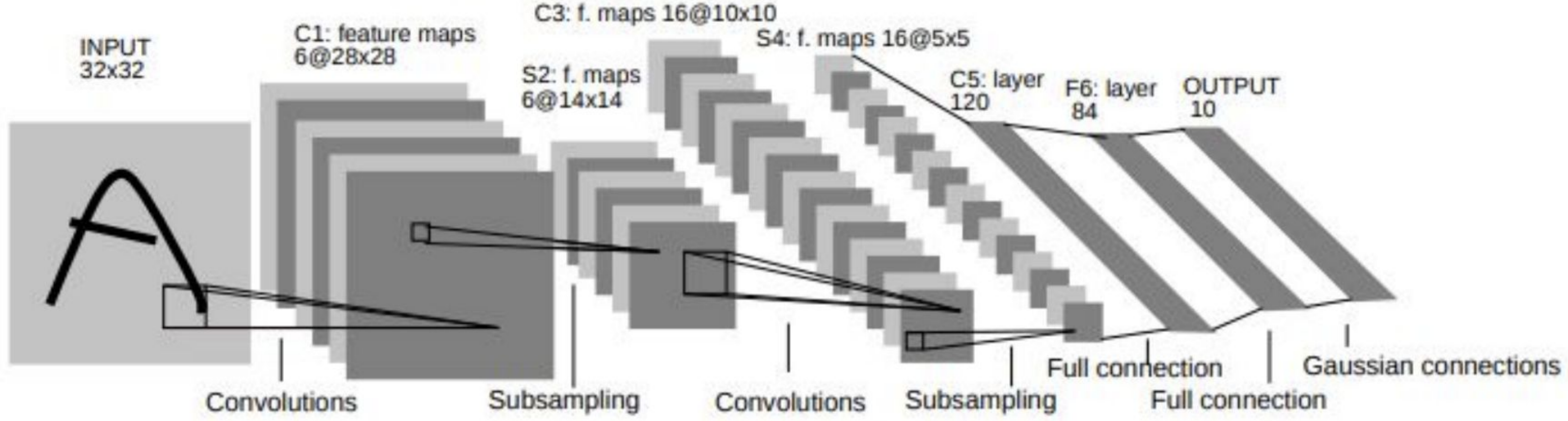


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

HSV.AI

ROLLING IN THE DEEP (2006)

Deep Learning (2006) - Again with Geoff Hinton. The idea was to train a simple 2-layer unsupervised model like a restricted boltzman machine, freeze all the parameters, stick on a new layer on top and train just the parameters for the new layer. Using this strategy, people were able to train networks that were deeper than previous attempts, prompting a rebranding of ‘neural networks’ to ‘deep learning’.

ACCELERATED BY HARDWARE AND DATA

It takes a lot of data to train a deep neural network:

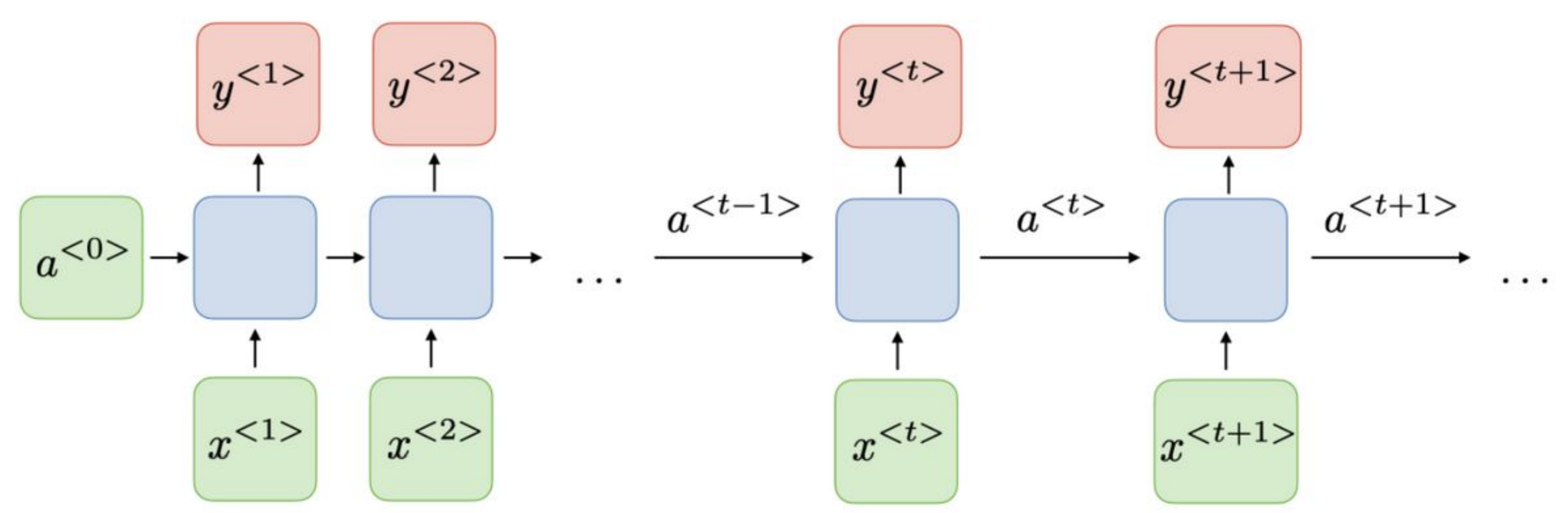
- Imagenet (2009) - millions of labeled images created and published by Fei-Fei Li at Stanford
- MNIST (1999) - Handwritten digits
- Google House Numbers from street view (2014) - created by an intern at Google (Ian Goodfellow)
- Flickr 30k Image dataset (2014)

You also need some good hardware at doing simultaneous calculation:

- GPU - used for multi-core floating point calculation

RECURRENT NEURAL NETWORKS (2010 - 2017)

Recurrent neural networks, also known as RNNs, are a class of neural networks that allow previous outputs to be used as inputs while having hidden states. They are typically as follows:



For each timestep t, the activation $a^{<t>}$ and the output $y^{<t>}$ are expressed as follows:

$$\boxed{a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)} \quad \text{and} \quad \boxed{y^{<t>} = g_2(W_{ya}a^{<t>} + b_y)}$$

where $W_{ax}, W_{aa}, W_{ya}, b_a, b_y$ are coefficients that are shared temporally and g_1, g_2 activation functions.

Credit - Stanford

ADVANCEMENTS OF DEEP LEARNING

Major leaps began to happen in the 2010's as the capability of these networks began to reach human level.

Alexnet (2012) - Won the Large Scale Visual Recognition Challenge(LSVRC) with an error rate 10% lower than the previous year. Used dropout to reduce overfitting and a rectified linear activation unit (ReLU)

Generative Adversarial Networks (2014) - Ian Goodfellow -

Gated Recurrent Units (2014) - this is when Siri, Alexa, Google Voice began to actually understand my accent.

“Langley Theory of AI” - they tried something and it somehow worked. Then they had to write a paper to figure out why.

THE TRANSFORMER (2017 - PRESENT)

Introduced by: Vaswani et al. in the paper "Attention is All You Need"

- **Self-Attention Mechanism**: Models relationships between all tokens in a sequence simultaneously, eliminating the need for sequential processing.
- **Positional Encoding**: Adds positional information to input embeddings since Transformers do not process data sequentially.
- **Feedforward Networks**: Applied after the self-attention mechanism within each layer.
- **Multi-Head Attention**: Allows the model to focus on different parts of the input simultaneously.
- **Parallelism**: Processes entire sequences at once, significantly speeding up training compared to RNNs.

THE TRANSFORMER (2017 - PRESENT)

Advantages:

- Handles long-range dependencies effectively.
- Scales well with increased data and compute.
- Allows for parallel processing, leading to faster training.

BERT (2018)

- Bidirectional Encoder Representations from Transformers.
- Pretrained on massive text corpora using a masked language modeling objective.
- Fine-tuned for downstream tasks like sentiment analysis, question answering, and named entity recognition.

GPT (2018–2023+)

- Generative Pre-trained Transformer.
- Introduced by OpenAI, focused on autoregressive language modeling (predicting the next token).
- GPT models (GPT-2, GPT-3, GPT-4) scaled up significantly, achieving remarkable results in text generation.

FINE TUNING

Fine-tuning a large language model (LLM) involves adapting a pretrained model (e.g., GPT, BERT) to perform specific tasks or align its behavior with particular requirements. This process leverages the general knowledge learned during pretraining and refines it using task-specific data.

Fine-tuning can also be applied to other deep learning networks, such as CNNs to be able to classify specific images.

REINFORCEMENT LEARNING WITH HUMAN FEEDBACK (RLHF)

Description:

- Combines reinforcement learning with human-labeled data to guide the model's behavior.

Pretraining:

- The model is pretrained on a large corpus of data using standard objectives (e.g., predicting the next word).

Fine-tuning:

- The model is fine-tuned using human feedback to optimize for specific goals.

Reward Modeling:

- Human preferences are collected by showing multiple outputs to humans and asking them to rank the responses.
- A reward model is trained on this data to predict the quality of a response.

Policy Optimization:

- The model is fine-tuned using reinforcement learning, optimizing for the reward model's scores.

GENERAL LLM THOUGHTS

There was actually a time (2019 - 2020) when you could build your own, or fine tune existing models on specific data.

Model Size	Approximate Parameter Count
Small (gpt2)	~117 million (or ~124 million in some sources)
Medium (gpt2-medium)	~345 million (or ~355 million in some sources)
Large (gpt2-large)	~762 million (or ~774 million in some sources)
Extra Large (XL) / Full (gpt2-xl)	~1.5 Billion (most frequently cited as 1.5 billion, or 1542M/1558M)

INSTRUCT MODELS (FROM GPT TO GPT-INSTRUCT

Paper - "Training language models to follow instructions with human feedback" - https://arxiv.org/pdf/2203.02155

Large language models can be "prompted" to perform a range of NLP tasks, given some examples of the task as input. However, these models often express unintended behaviors such as making up facts, generating biased or toxic text, or simply not following user instructions.

This is because the language modeling objective used for many recent large LMs — predicting the next token on a webpage from the internet — is different from the objective "follow the user's instructions helpfully and safely".

INSTRUCT MODELS (FROM GPT TO GPT-INSTRUCT

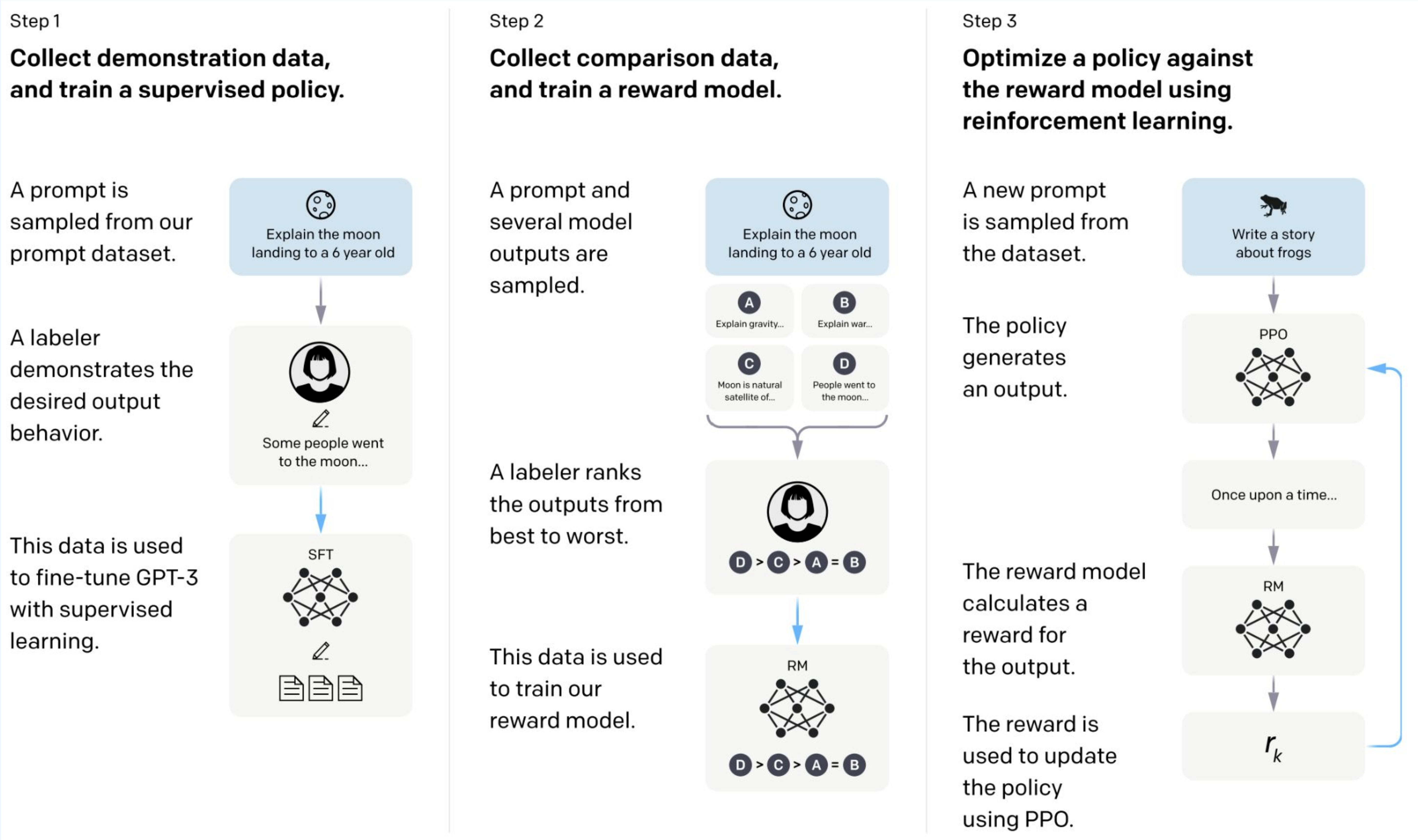
Let's talk a bit about "Alignment"

From the Paper:

We want language models to be helpful (they should help the user solve their task), honest (they shouldn't fabricate information or mislead the user), and harmless (they should not cause physical, psychological, or social harm to people or the environment).

Can you think of areas where alignment would mean something different?

INSTRUCT MODELS (FROM GPT TO GPT-INSTRUCT



INSTRUCT MODELS (FROM GPT TO GPT-INSTRUCT

Step 1: Collect demonstration data, and train a supervised policy. Our labelers provide demonstrations of the desired behavior on the input prompt distribution (see Section 3.2 for details on this distribution). We then fine-tune a pretrained GPT-3 model on this data using supervised learning.

Step 2: Collect comparison data, and train a reward model. We collect a dataset of comparisons between model outputs, where labelers indicate which output they prefer for a given input. We then train a reward model to predict the human-preferred output.

Step 3: Optimize a policy against the reward model using PPO. We use the output of the RM as a scalar reward. We fine-tune the supervised policy to optimize this reward using the PPO algorithm (Schulman et al., 2017).

INSTRUCT MODELS (FROM GPT TO GPT-INSTRUCT

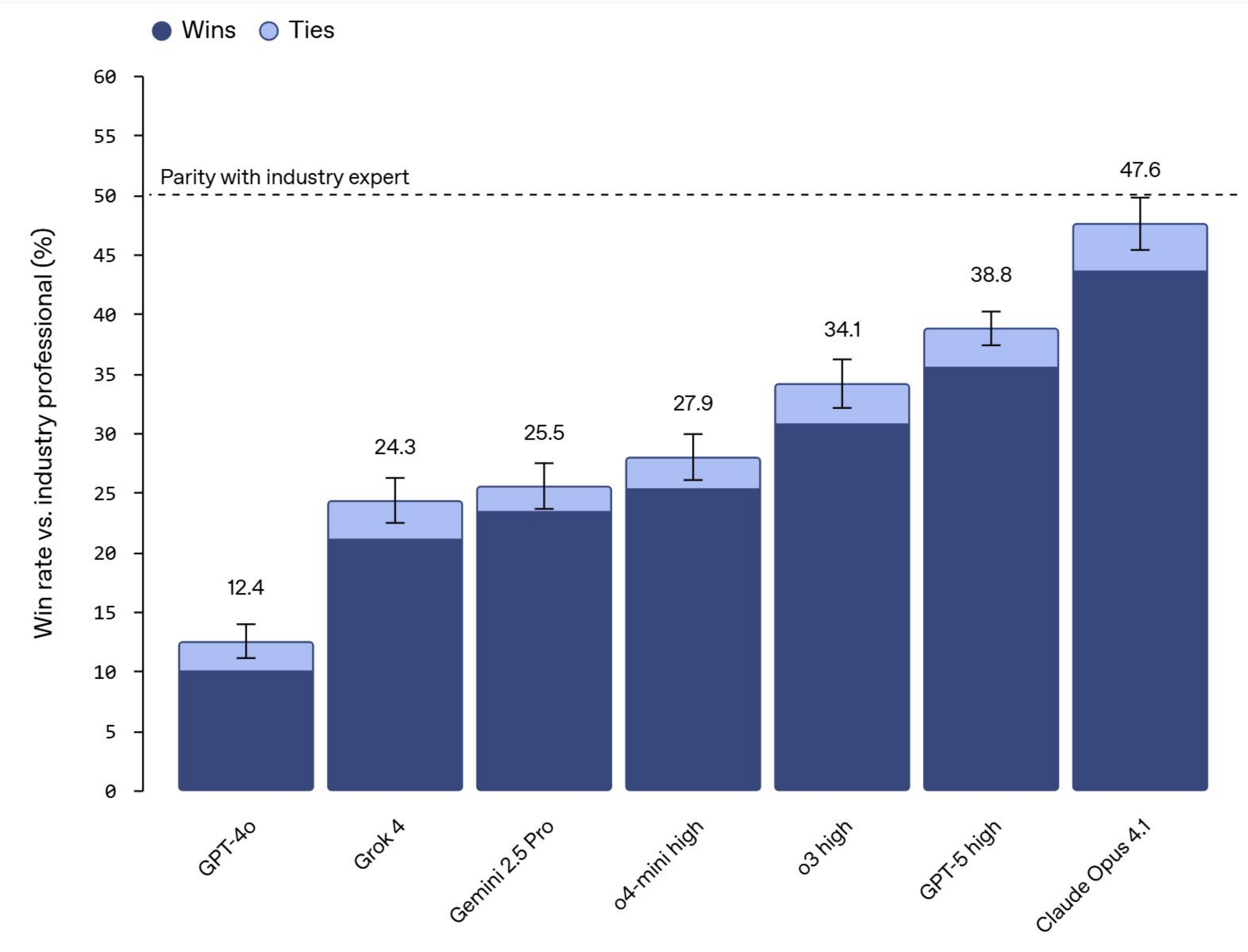
In interacting with our 175B PPO-ptx model, we have noticed it can still make simple mistakes, despite its strong performance on many different language tasks. To give a few examples:

(1) when given an instruction with a false premise, the model sometimes incorrectly assumes the premise is true,

(2) the model can overly hedge; when given a simple question, it can sometimes say that there is no one answer to the question and give multiple possible answers, even when there is one fairly clear answer from the context, and

(3) the model's performance degrades when instructions contain multiple explicit constraints (e.g. "list 10 movies made in the 1930's set in France") or when constraints can be challenging for language models (e.g. writing a summary in a specified number of sentences).

AI AT HUMAN LEVEL



From the Oct 2025 GDPval Paper - https://openai.com/index/gdpval/

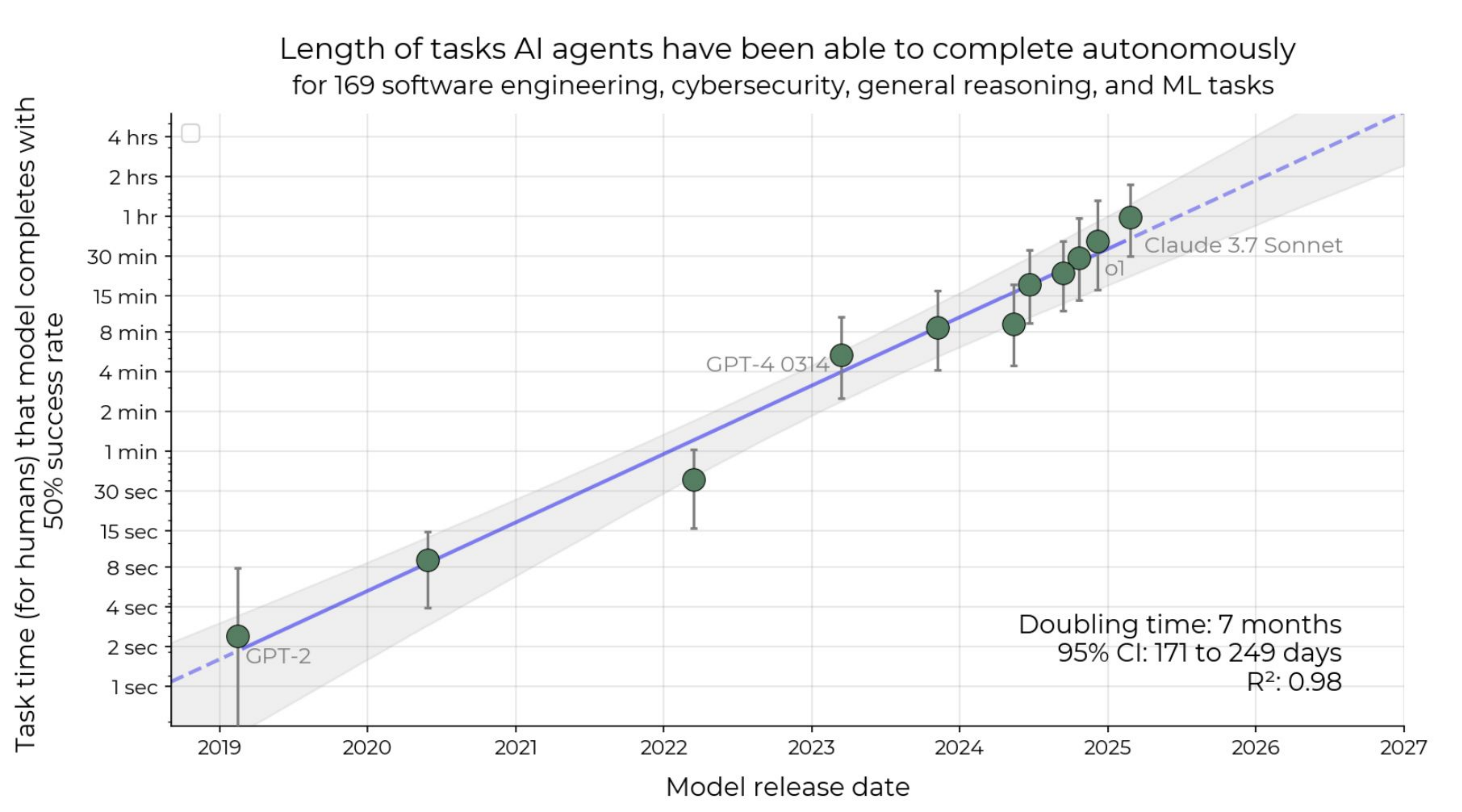
AI AT HUMAN LEVEL

OpenAI released GPT 5.2 in December of 2025

	GPT-5.2 Thinking	GPT-5.1 Thinking
GDPval (wins or ties) Knowledge work tasks	70.9%	38.8% (GPT-5)

METR Benchmark

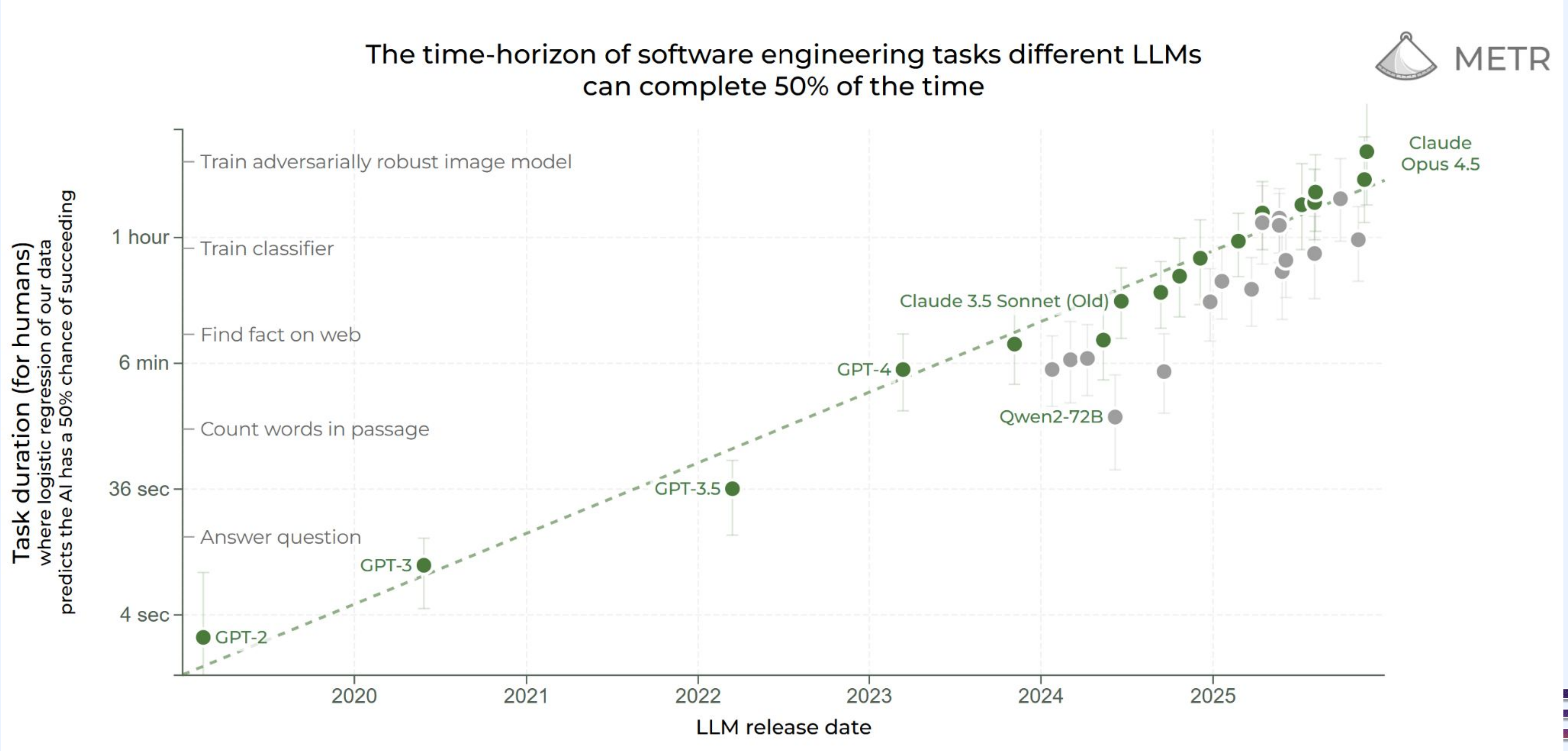
Research paper from METR in March 2025 - https://arxiv.org/pdf/2503.14499



HSV.AI

METR Updated late 2025

Blog from METR with current updates - https://metr.org/blog/2025-03-19-measuring-ai-ability-to-complete-long-tasks/



WHAT DOES HUMAN LEVEL MEAN ANYWAY?

> Aoccdrnig to a rscheearch at Cmabrigde Uinervtisy, it deosn't mttaer in waht oredr the ltteers in a wrod are, the olny iprmoetnt tihng is taht the frist and lsat ltteer be at the rghit pclae. The rset can be a toatl mses and you can sitll raed it wouthit porbelm. Tihs is bcuseae the huamn mnid deos not raed ervey lteter by istlef, but the wrod as a wlohe.

Factually incorrect, but fun meme

CAN MODELS UNDERSTAND?

Climbing towards NLU: On Meaning, Form, and Understanding in the Age of Data

Emily M Bender & Alexander Koller

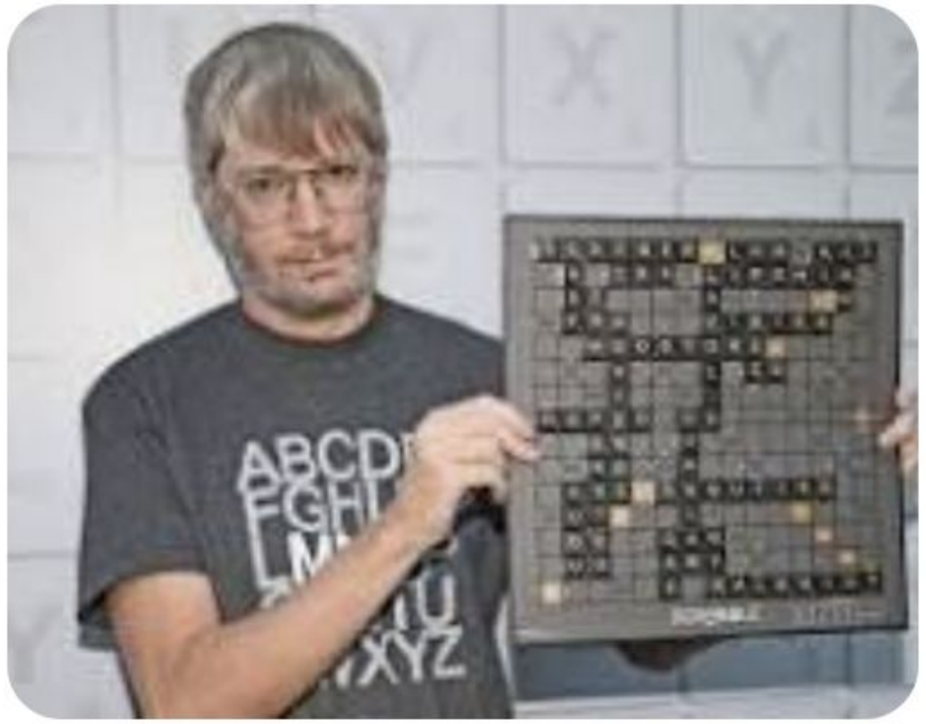
Octopus Thought Exercise:

- Pattern recognition and repitition?
- Understanding?
- Grounding

GOOD ENOUGH FOR SCRABBLE

Nigel Richards

He won the Spanish Scrabble championships, yet he doesn't speak Spanish. Nigel Richards at the World Scrabble Championship in London on October 28, 2018. A New Zealand man hailed as a Scrabble phenom dominated the Spanish World Scrabble Championships – despite reportedly not speaking the language. Dec 11, 2024



CNN https://www.cnn.com

Questions?

COMMENTS, OR CRIES OF HERESY?

J. Langley,

Founder, Huntsville AI

CTO, CohesionForce, Inc

January 14, 2025