The History of Artificial Intelligence

From Leonardo da Vinci to ChatGPT and AI-Enabled Nobel Prizes

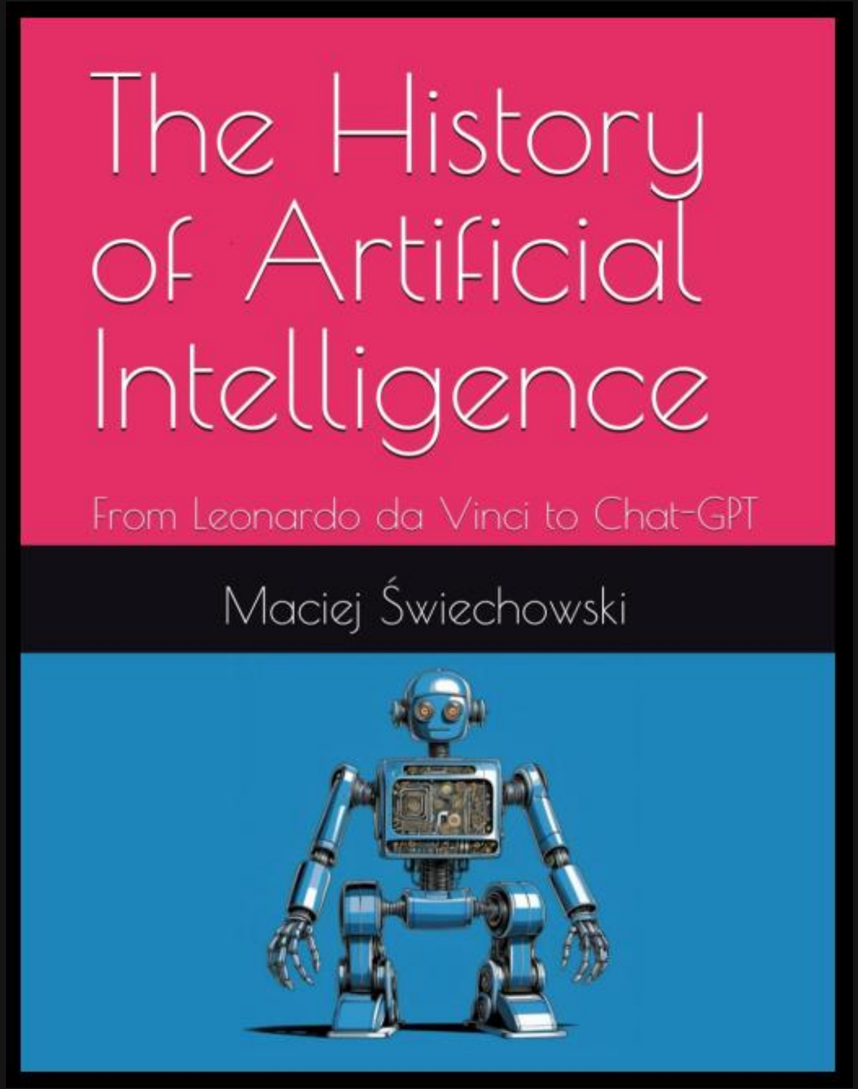
Maciej Świechowski

This talk vs. the book

This webinar will focus only on **selected concepts** and will show them to **tell a specific story**.

The rest will be only outlined.

In the book, there are many equations and mathematical details. In this talk, the concept will be presented more generally without strict technical background.



About the speaker

BSc. and MSc. in computer science, Warsaw University of Technology

PhD in artificial intelligence, 2015, from Systems Research Institute, Polish Academy of Sciences

Research practicum at University of New South Wales, Australia

Current work:

- Chief Scientist at QED Software
- CTO at QED Games
- Teacher at Warsaw University (MIMUW)
- Vice-president at Information Technologies for Psychiatry Foundation

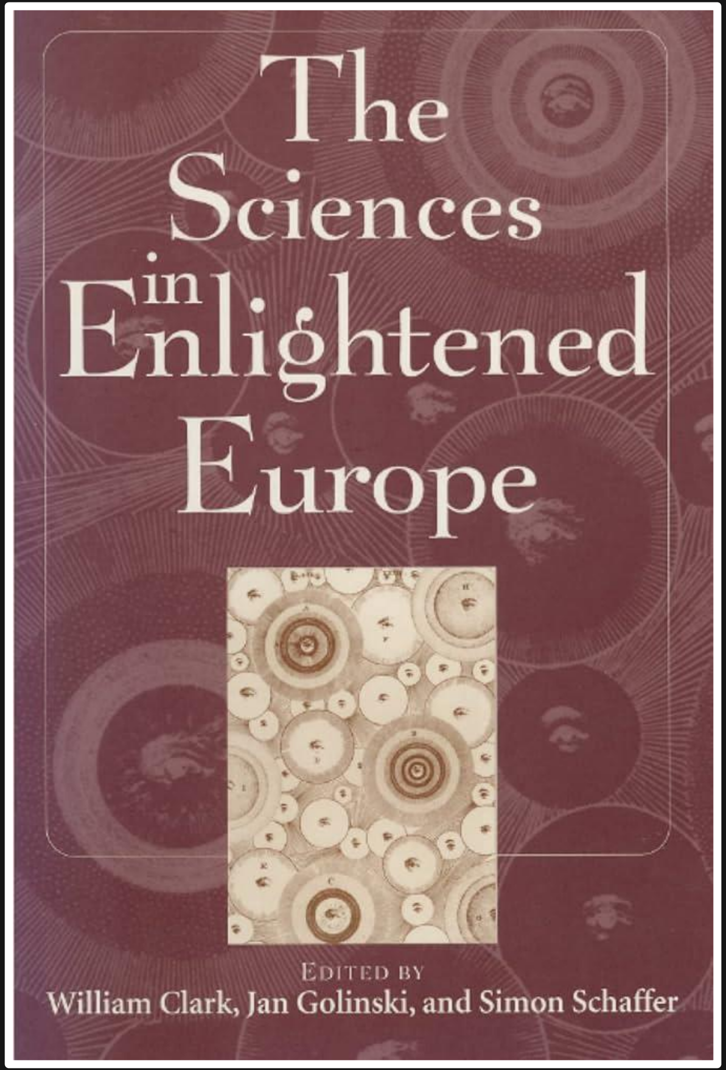
Introductionary stories

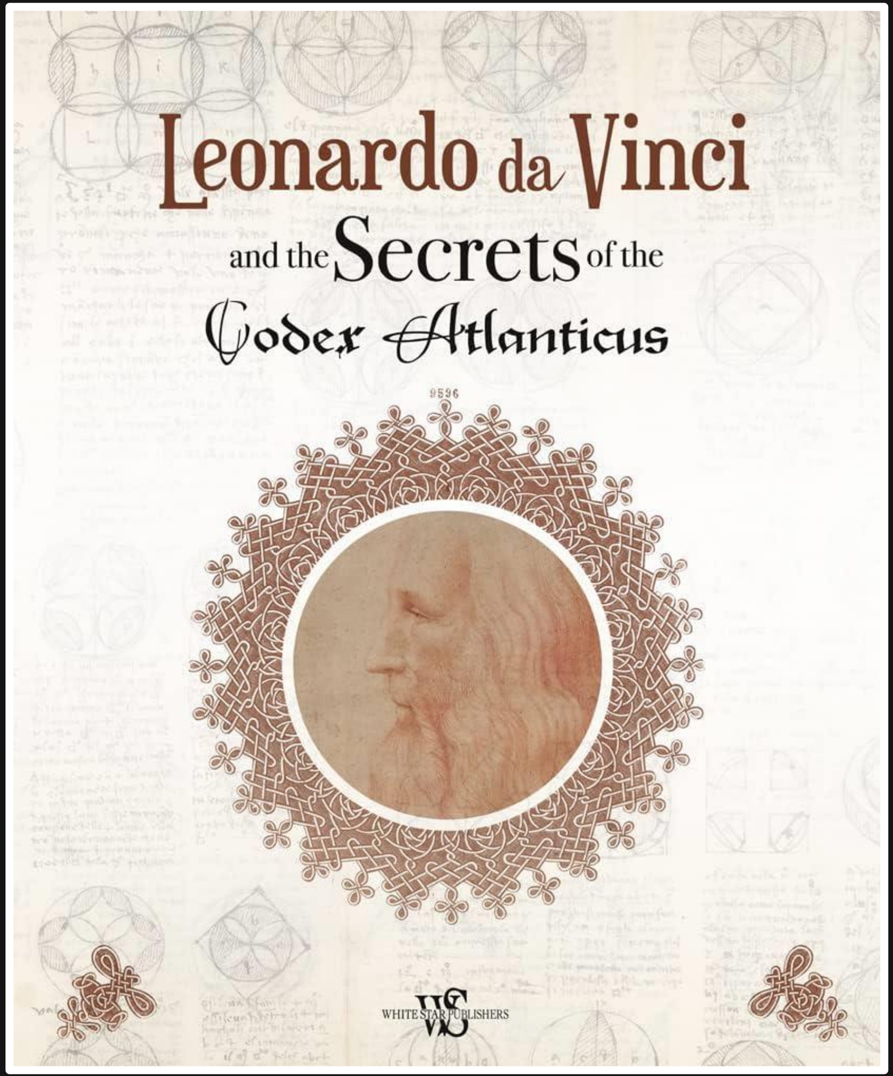
Before I will define Artificial Intelligence

Story 1

Dreams and Inspirations

People dreamed of autonomous machines





(1478 ~ 1519) Leonardo da Vinci’s Automata

Self-propelled cart

Mechanical Knight (~1496)

- could perform basic tricks such as to stand-up and sit-down
- was able to move its arms (in 4 degrees of freedom), legs (in 3d of freedom), head and jaw
- cranks and cables

Mechanical Lion (early 1500s)

- believed to be constructed for the French king Francois I
- could allegedly walk and move its head without human supervision
- required winding up beforehand

[Replica in Da Vinci Museum, Florence]

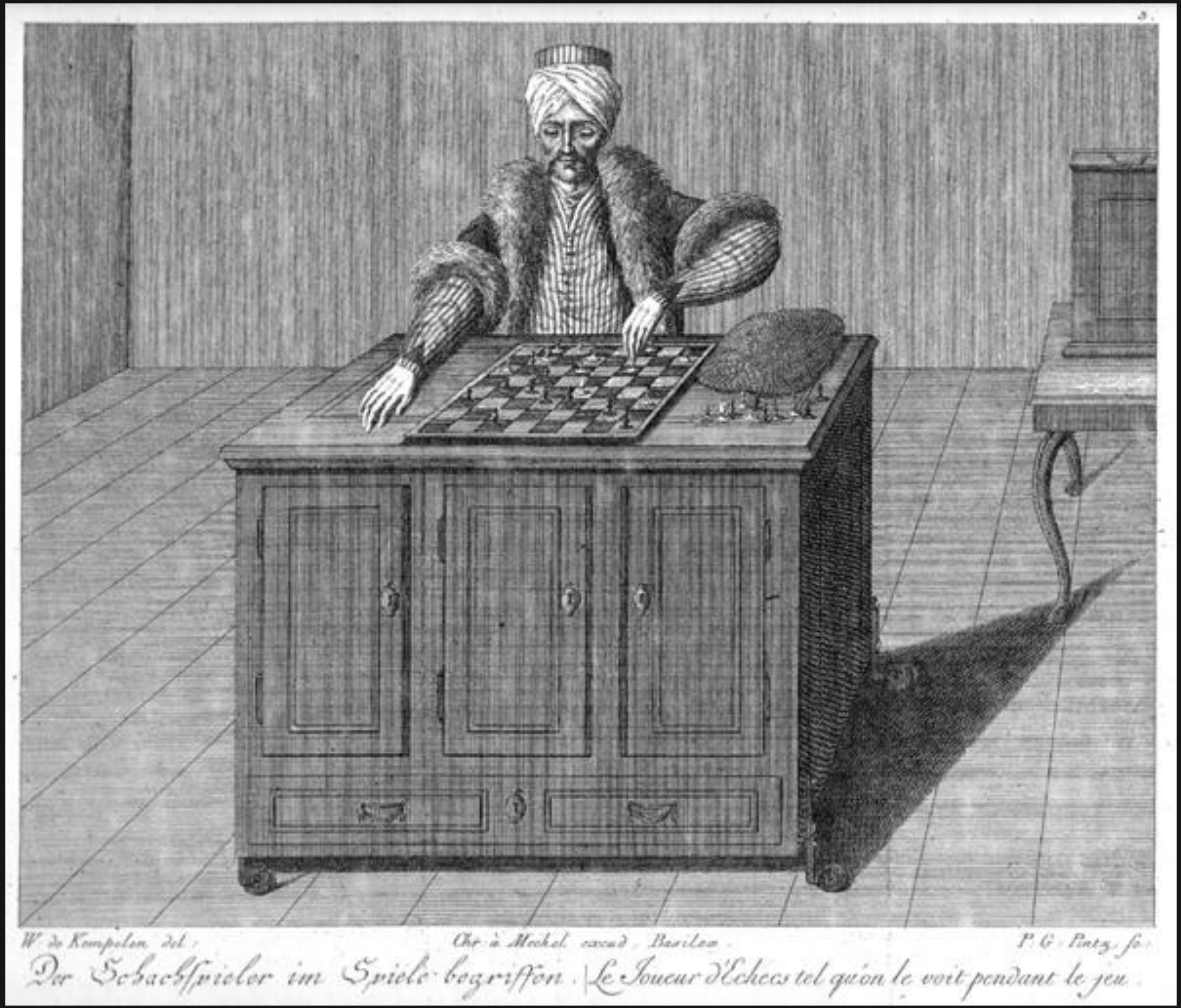
(1770) The Mechanical Turk

1770: Wolfgang von Kempelen presents a chess-playing machine at Schönbrunn Palace, Vienna.

Cabinet + chessboard + upper body mannequin

It could play chess.

Before each game Von Kempelen would open the cabinet and provide a demonstration.

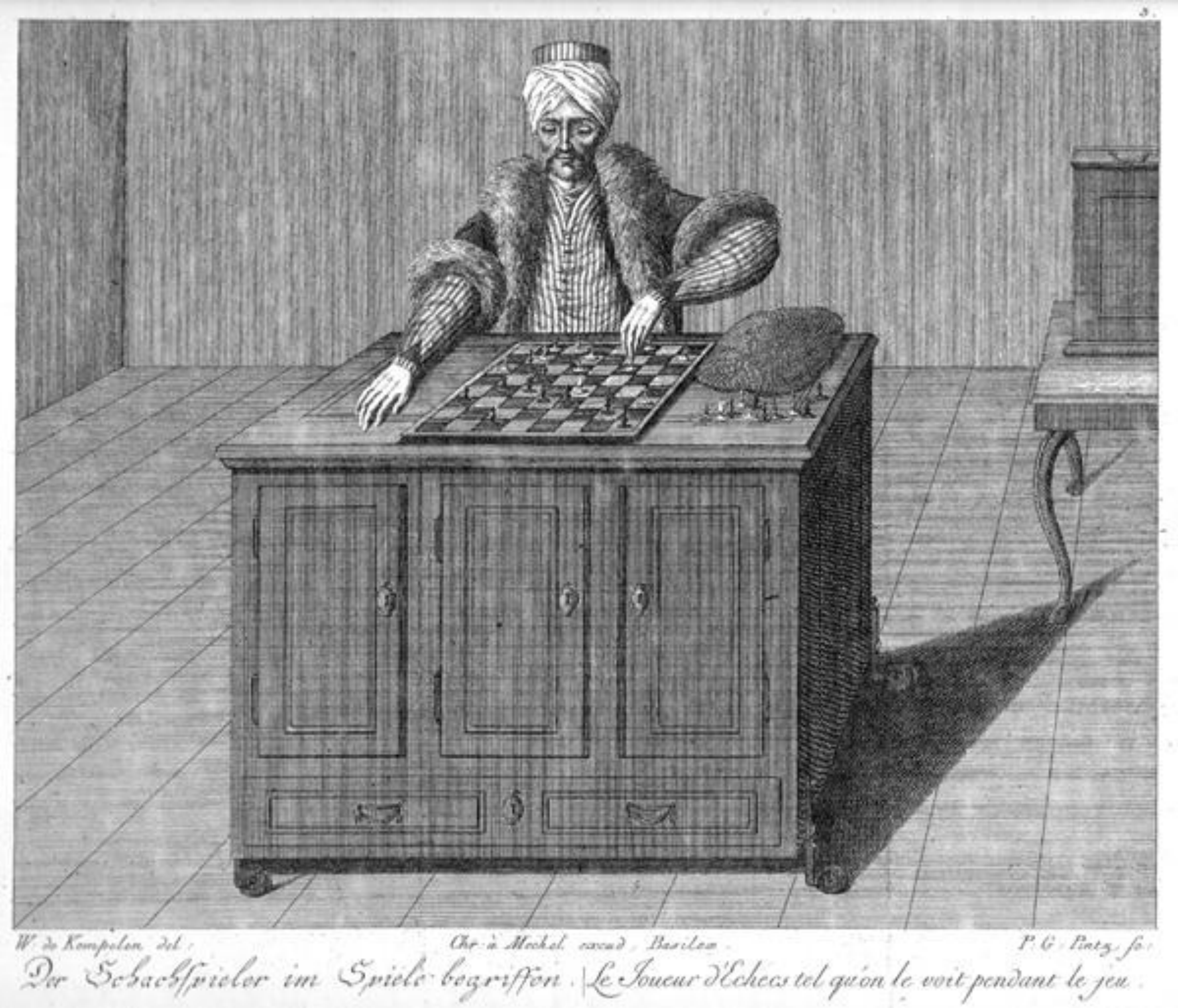


(1770) The Mechanical Turk

The Turk was an excellent chess player, initially defeating all players it faced.

Some notable people it played:

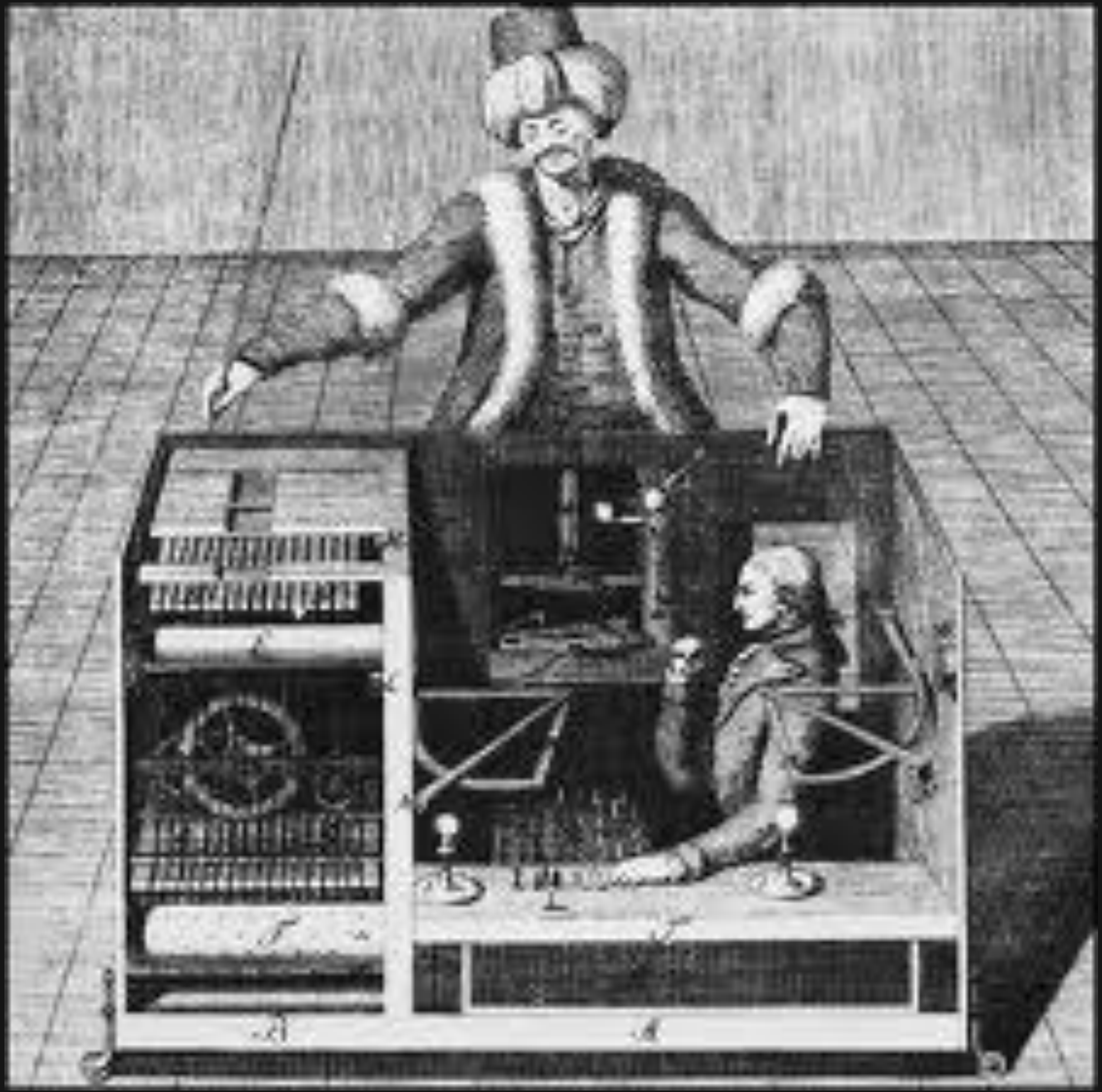
- Emperor Joseph II
- Napoleon Bonaparte
- Grand Duke Paul of Russia
- Benjamin Franklin



(1770) The Mechanical Turk

Eventually the hoax was revealed.

- Small person inside
- Still extremely strong chess player
- Probably switched places during demonstration
- Moved chess pieces using strong magnets and strings



We know a lot about the Turk thanks to essays of Edgar Allan Poe titled “Maelzel’s Chess-Player.

Story 2

Alan Turing

(1950+) Alan Turing's Work

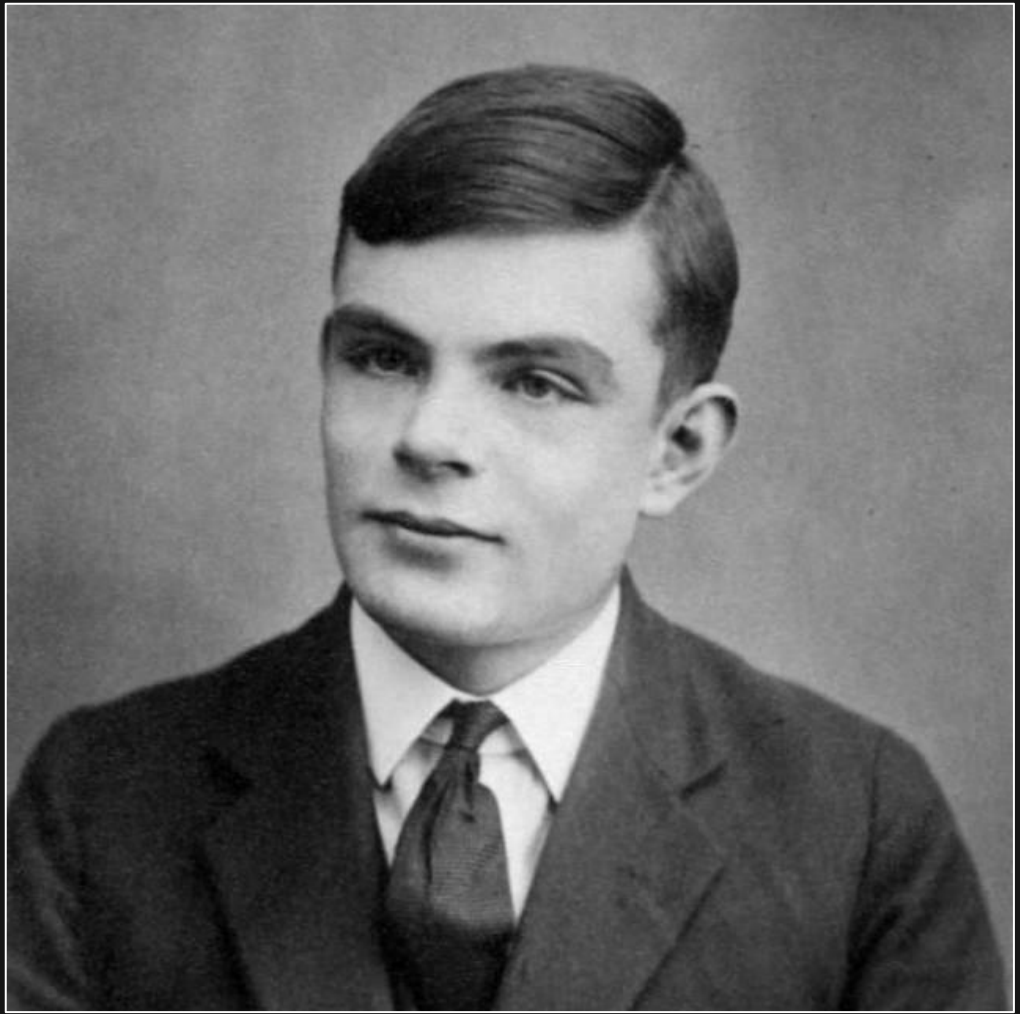
Alan Turing

- 1912 (London, England) – 1954 (Wilmslow, England)

One of the founding fathers of computer science

Notable work:

- theory of computation
- the Turing's machine
- contribution to breaking the enigma code
- imitation game
- the first chess playing program (written on paper)

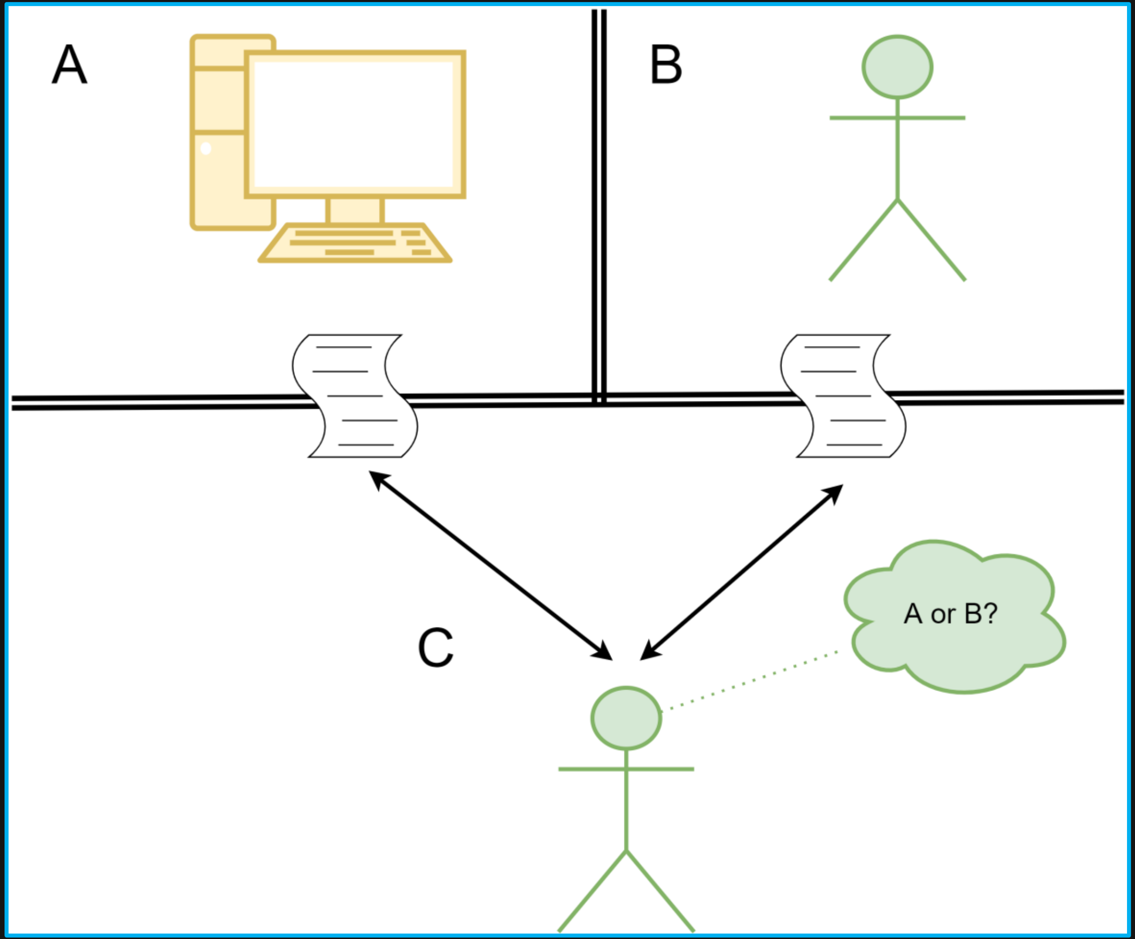


(1950) Imitation Game

1950: in “Computing Machinery and Intelligence” he posed a question “Can machines think?”

Imitation Game

- Later became known as the **Turing Test**
- Three types of participants: A, B, C
- C is an interrogator
- A and B: initially man or woman, later machine or human
- C has to make a judgment



(1950) Imitation Game

The Turing Test has become a big deal, but the point of the original paper was that a general question "Can machines think?" is too meaningless until it is formalized what a machine is and what thinking is. He argued that instead of "thinking" researchers should seek specific activities that require intelligence.

Story 3

Formalization of the Field

(1956) The Dartmouth Workshop

Full name: The Dartmouth Summer Research Project on Artificial Intelligence

20 participants.

The proposal (1955) for the workshop was signed by:

J. McCarthy (Dartmouth College), M.L. Minsky (Harvard University), N. Rochester (I.B.M. Corporation), and C.E. Shannon (Bell Telephone Laboratories)

In the proposal, the term "Artificial Intelligence" was coined.

(1956) The Dartmouth Workshop

Agenda:

- Automatic computers
- How can a computer use a language
- "Neuron nets"
- Theory of computation
- Self-improvement
- Abstraction
- Randomness
- Creativity

(1956) The Dartmouth Workshop

The beginning of the era of symbolic artificial intelligence.

- humans would usually encode knowledge in the form of symbols
- the symbols are manipulated by machines to derive new insights, information, results, also encoded using symbols

Definitions of Artificial Intelligence (AI)

Imitation:

“fooling humans into thinking that they are interacting with another human and not a machine”

From Dartmouth:

“the science of making machines do things that would require intelligence if done by men”

“(…) any intellectual task done by humans with at least the same level of efficacy”

As a branch of Computer Science:

“making computers solve problems without being explicitly programmed to do so”

“solving problems, which are difficult to algorithmize”

“set of certain techniques and methodology of working with them”

AI level – from socio-philosophical point of view

“Narrow AI” (“Weak AI”) – for specific task only

“General AI” (“Artificial General Intelligence”) – the same AI for any task

Superintelligence - when AGI surpasses humans in all tasks => superhuman AI => singularity

AI as computer science field

Three major pillars (ideas) for implementing AI:

1. Expert knowledge

- rules, patterns, computation methods provided by human experts
- solving problems by continuous execution of these, applying their effects, analyzing how they interplay

2. Intelligent search

- iteratively searching an abstract space (of potential solutions or representations translatable to solutions)

3. Statistical learning

- for example: machine learning
- inferences based on empirical data (typically past data)

Machine learning model

ML model – representation (data structure) that is trained using machine learning and after it is ready, used (inferred) on new data.

Main types of machine learning

Supervised learning

- Training using ideal examples (expected outputs – so called “ground truths”)

Unsupervised learning

- Training without examples; usually learns patterns in data

Reinforcement learning

- The method gathers data empirically (on the fly) through by an interaction with environment
- Rewards and sometimes penalties

Also recent soft variants: **semi-supervised learning** and **self-supervised learning**

Types of machine learning tasks

Classification

- assigning labels (one or more) to data
- “is it a cat or dog?” “Is there a car on this picture?”
- related to integer numbers (1: class1, 2: class2, ...) or (0: no, 1: yes)

Regression

- fitting a model, so it can be used to output new data
- related to real value numbers (e.g. predicts an exact price of an item)

Plan:

- The history of neural networks (detailed)
- The history of intelligent game-playing (detailed)
- The history of other AI tools, techniques and events (overview)

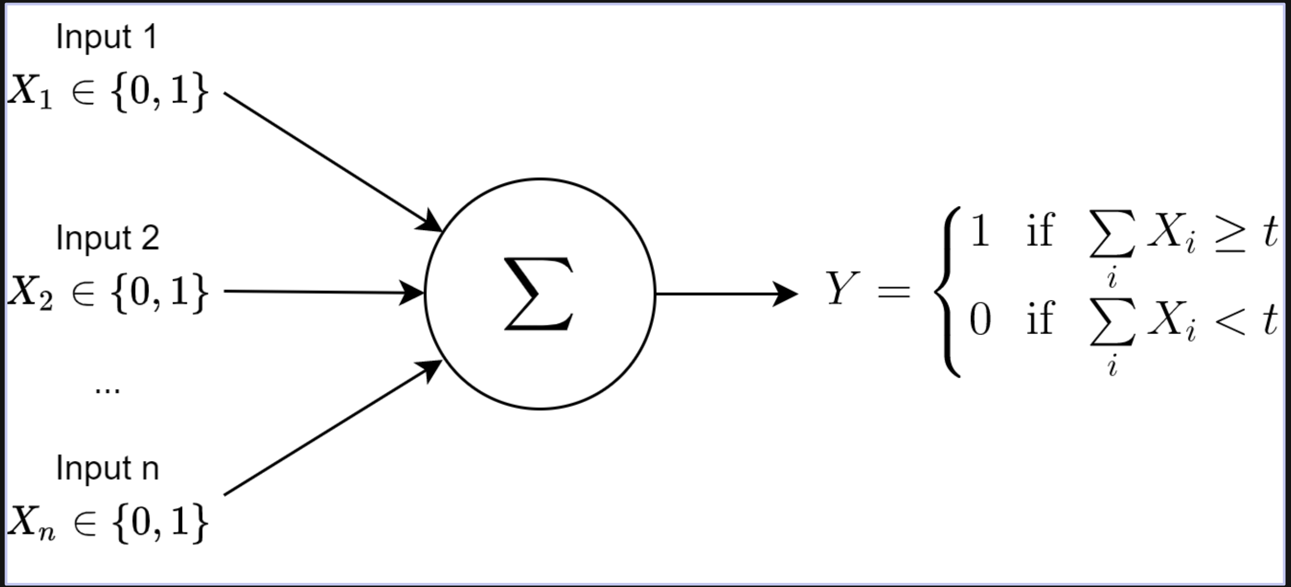
The history of neural networks

(1943) The First Mathematical Model of a Neuron

A paper titled: "A Logical Calculus of Ideas Immanent in Nervous Activity"
by Warren McCulloch (USA, 1898-1969) and Walter Pitts (USA, 1923 – 1969)

Key ideas:

- signal as 0 or 1
- sum of input signals
- representing logical operations
- t: threshold for neuron excitement
- discussions about natural inspirations



(1943) The First Mathematical Model of a Neuron

One neuron has many limitations

X1 input	X2 input	AND	OR	NAND	XOR
0	0	0	0	1	0
1	0	0	1	1	1
0	1	0	1	1	1
1	1	1	1	0 (error!)	0 (error!)

This model still serves as a foundation of most modern neural networks!.

(1944-1958) Logistic Regression

1944: “Application of the Logistic Function to Bio-Assay” by Joseph Berkson

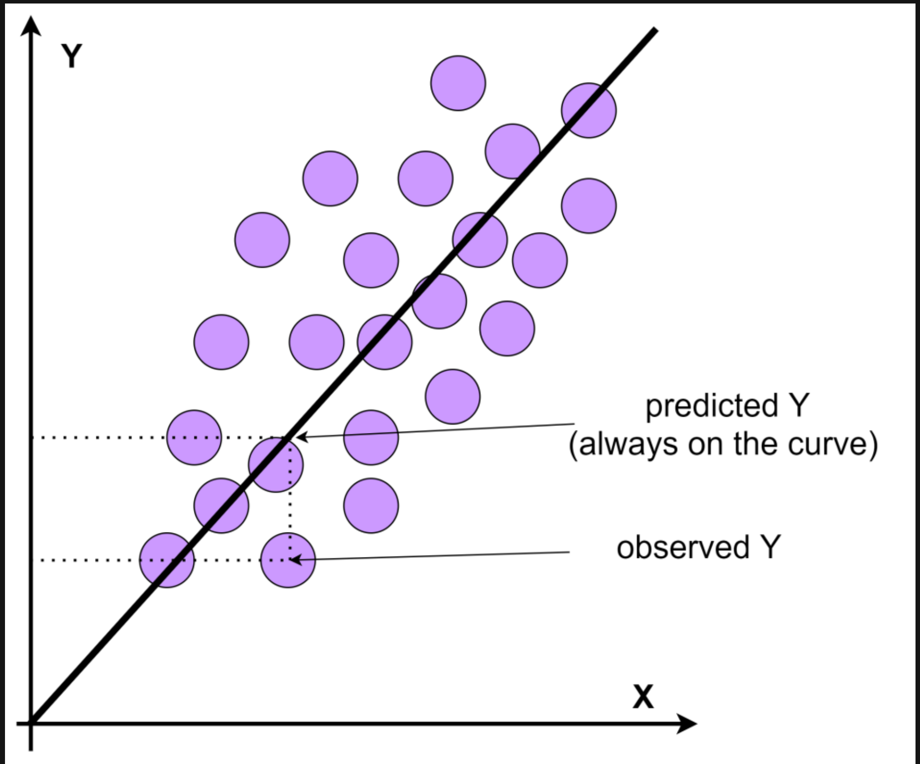
1958: “Two Further Applications of a Model for Binary Regression” by David Cox

Key idea:

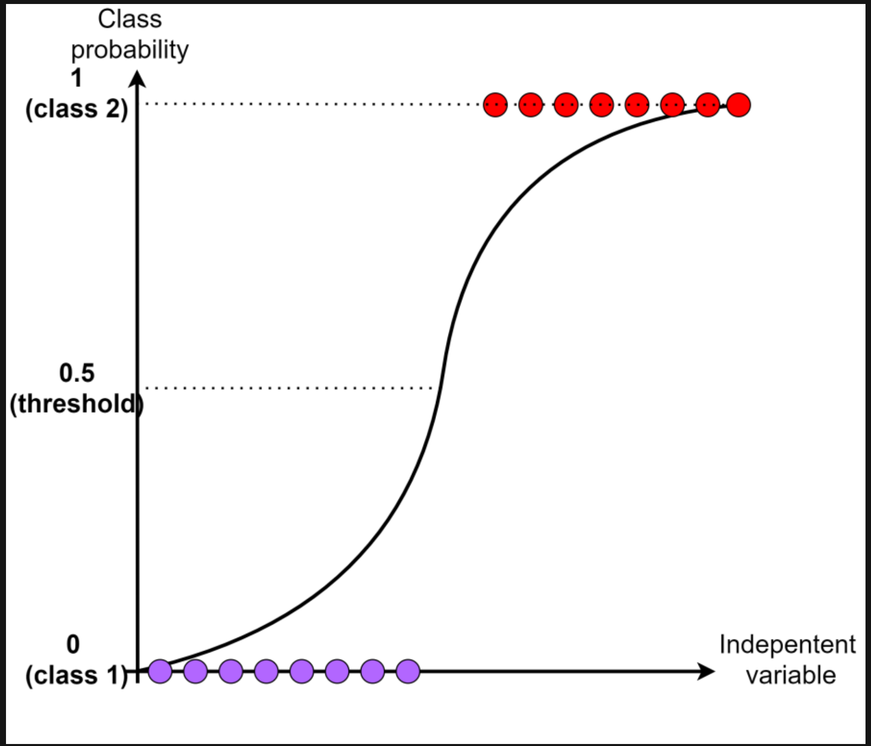
Instead of using just a threshold for a signal, we introduce a function that will separate the input inactive (0) vs. active (1)

The birth of activation function!

(1944-1958) Logistic Regression



Linear regression separating data into classes using a line



Logistic regression – separating data using a logistic function, the output interpreted as probability of classes

(1949) Hebbian Learning

1949: "The Organization of Behavior: A Neuropsychological Theory" by Donald O. Hebb

- Less popular model of a neuron
- One of the key ideas: "neurons that wire together, fire together"
- Famous use in Hopfield Networks (later)

(1958) Perceptron

by Frank Rosenblatt (1928 – 1971)

Key idea: let’s connect McCulloch and Pitts neurons.

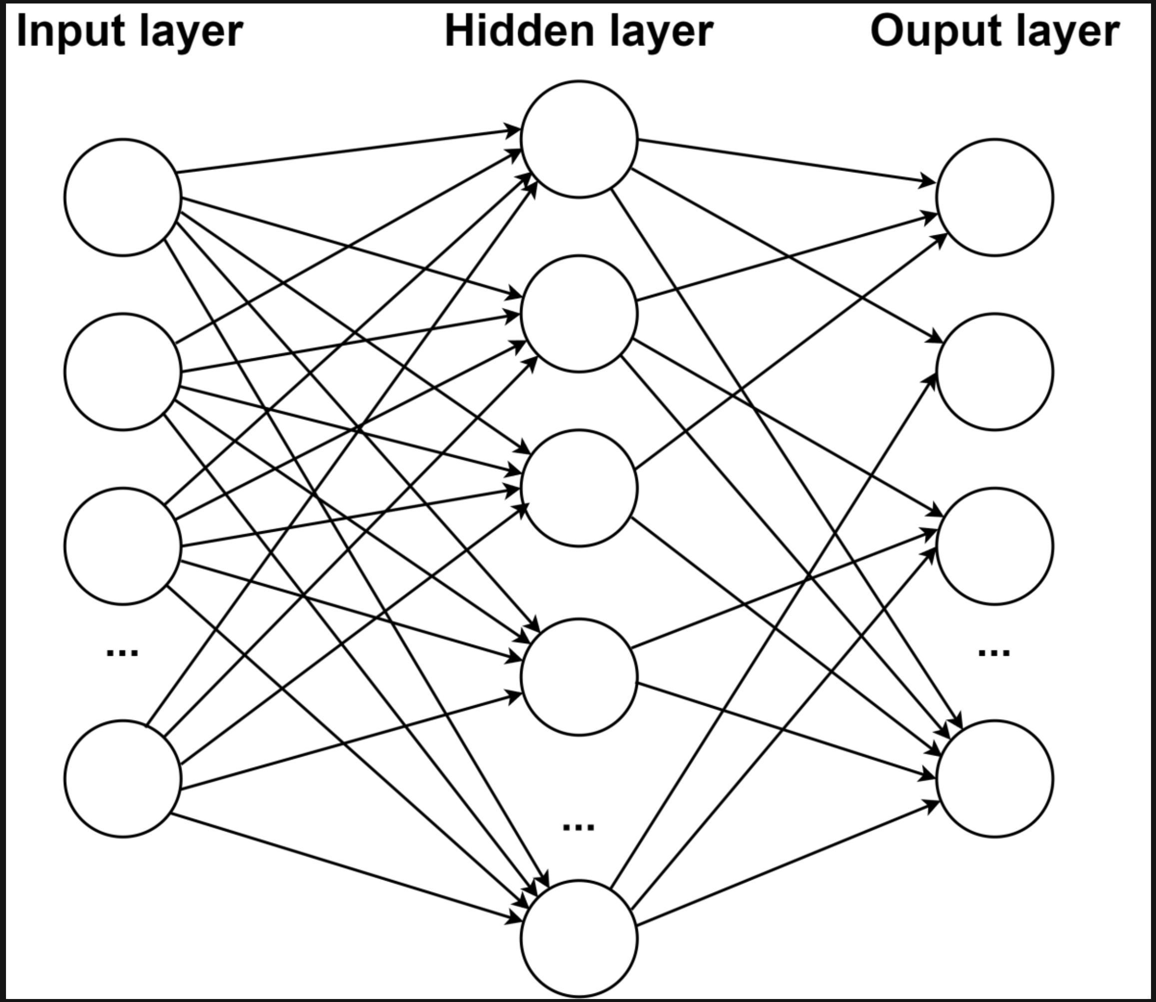
Feed-forward network:

the signal flows from the input in one direction

Perceptron originally had three layers.

Multi-layer perceptron (MLP) – more layers.

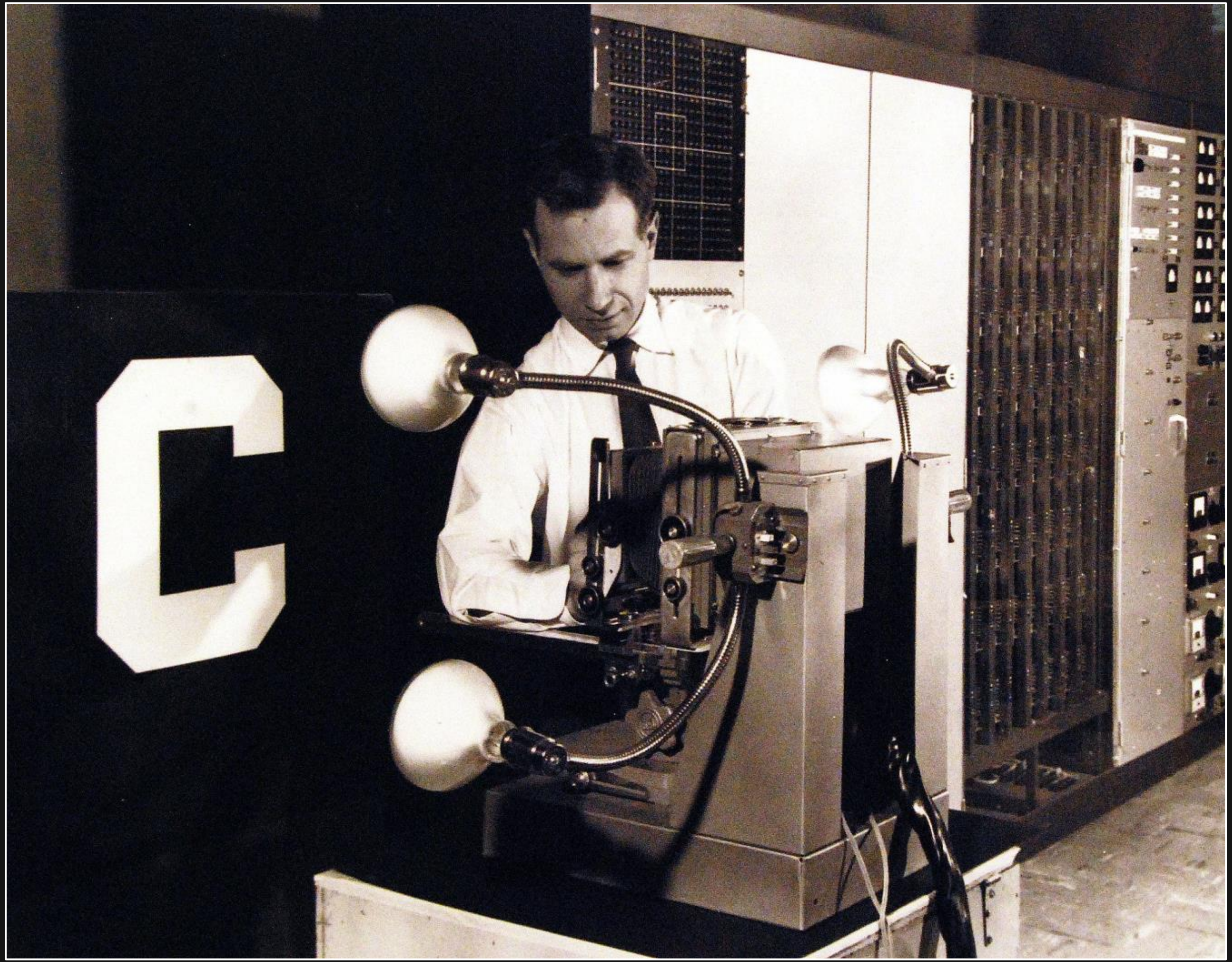
The majority neural networks are MLPs.



(1958) Perceptron as a device

A physical machine.

The goal was to process visual (optical) patterns and simulate human perception.



(~1965) Deep Learning Foundations

The ideas of deep learning:

- large number of layers in neural networks
- the ability to process raw input (features extracted in the network rather than in preprocessing)
- efficient training method, not always from scratch

"Group Method of Data Handling"(GMDH) by Alexey Ivakhnenko (a Soviet and Ukrainian mathematician)

(1982) Backpropagation

A powerful algorithm used to train neural networks in a supervised-learning fashion.

Key idea:

Loss function – returns error between ideal output and current output from the network

Through mathematical operations, we can calculate:

- how each node in the network contributes to the loss
- how each modifiable (learnable) parameter should be changed to minimize the loss

This can be done efficiently starting from the output and going backwards layer by layer.

(1982) Backpropagation

1676: chain rule by Gottfried Wilhelm Leibniz (GER) for derivatives of composite functions

1847: gradient descent, commonly credited to Augustin-Louis Cauchy (FRA)

1960: "Gradient Theory of Optimal Flight Path" by Henry Kelley – similar method to back-propagation

1982: "Applications of Advances in Nonlinear Sensitivity Analysis" by Paul Werbos

- the first implementation of backpropagation in the way it is applied today

1986: "Learning representations by back-propagating errors" by David Rumelhart, Geoffrey Hinton *(2024 Nobel Prize winner)* **and Ronald Williams**

- greatly popularized the algorithm

Non-MLP variants of Neural Networks

1982: Hopfield Networks by John Hopfield *(2024 Nobel Prize winner)*

Recurrent

- Each network was connected to all other networks but not to itself

Based on Hebbian Learning

Associative Memory and Representing Patterns

1985: Boltzmann Machines by David H. Ackley, Geoffrey E. Hinton, and Terrence J. Sejnowski

Many similarities to Hopfield Networks

Foundations for modern Restricted Boltzmann Machines

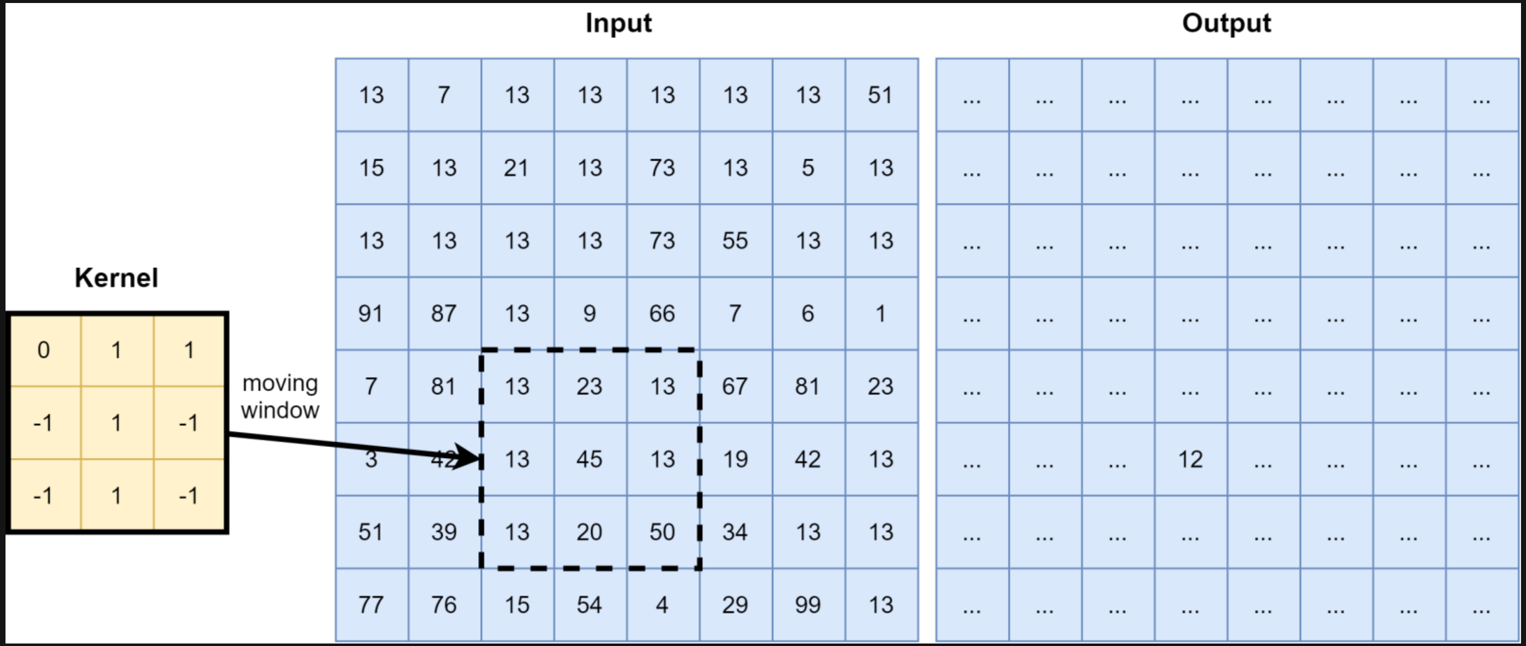
(1989-1998) Convolutional Neural Networks (CNN)

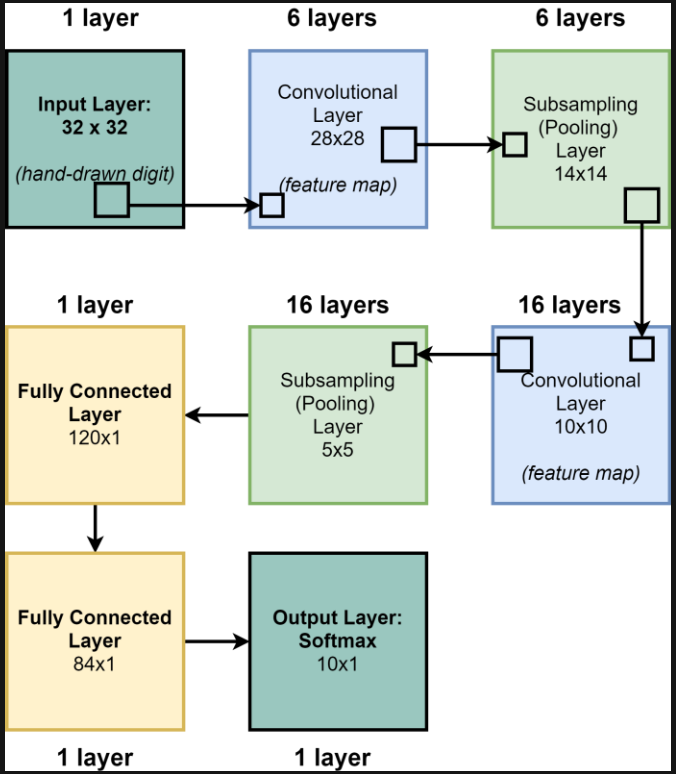
1989 (groundwork): "Handwritten Digit Recognition with a Back-Propagation Network" by Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R.E. Howard, W. Hubbard, and L.D. Jackel

1998: "Gradient-based learning applied to document recognition" by LeCun et. al.

Great for tasks based on processing images.

(1989-1998) Convolutional Neural Networks (CNN)





(1990) Backpropagation through time

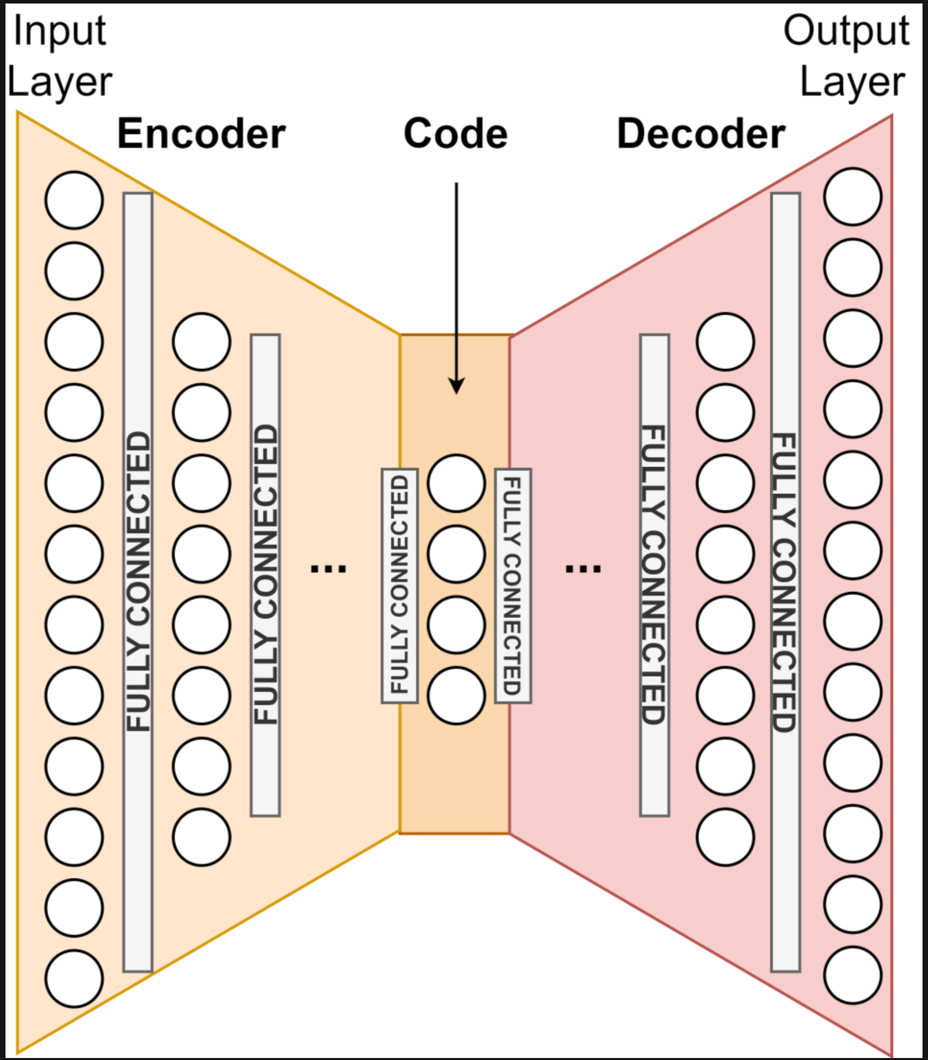
1990 “Backpropagation Through Time: What It Does and How to Do It” by Paul Werbos

Pragmatic method of training recurrent neural networks.

(1991) Autoencoders

1991 “Nonlinear Principal Component Analysis Using Autoassociative Neural Networks”

by **Mark Kramer**

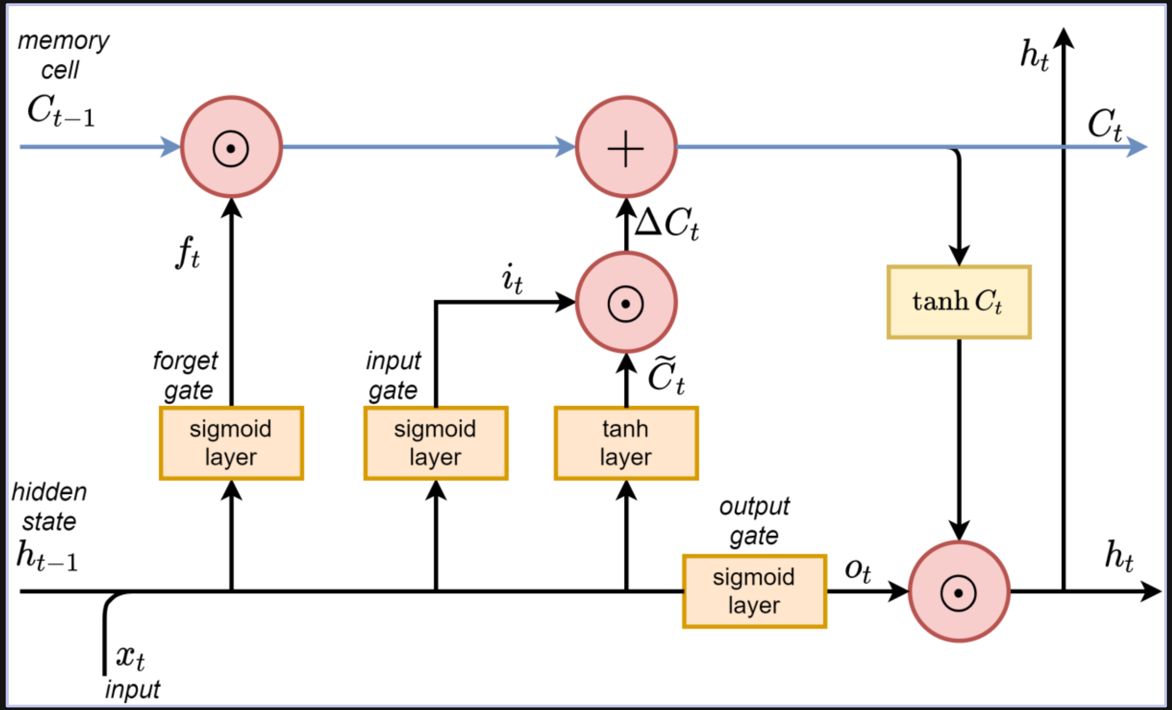


Learning to reconstruct the input.

(1997) Long Short-Term Memory (LSTM)

1997 “Long Short-term Memory”

by Sepp Hochreiter and Jürgen Schmidhuber



Much improved Recurrent Neural Networks

Handle long-term dependencies

Serve as foundation for other architectures

Many practical applications, e.g.: natural language processing, speech recognition, time series prediction, games.

(2005) Graph Neural Networks

2005: "A New Model for Learning in Graph Domains" by M. Gori, G. Monfardini, and F.Scarselli

2005: "Graph Neural Networks for Ranking Web Pages" by F. Scarselli, L.Y. Sweah, M.Gori, M. Hagenbuchner, A.C. Tsoi, and M. Maggini

Can effectively handle graphs as input, e.g.

- Social networks, maps and plans, chemical structures, ontologies, buildings, skeletons, semantic webs, other neural networks

(2012) AlexNet

2012: "ImageNet Classification with Deep Convolutional Neural Networks" by Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton

- **one of the most cited article in computer science**

ImageNet: Large Scale Visual Recognition Challenge

  - in 2012, it contained 14 million images across 22K categories
  - possible by employing the Mechanical Turk crowd-sourcing service

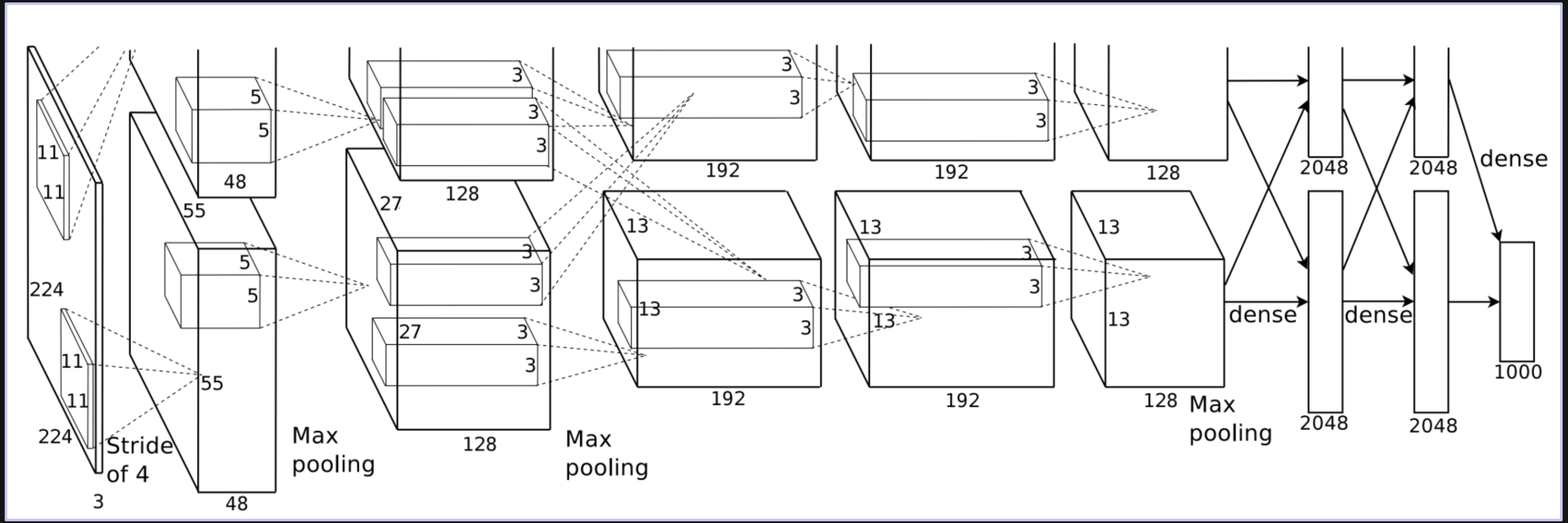
In 2012, AlexNet won by enormous margin.

Since 2012, large NN have won the challenge.

(2012) AlexNet

Some features:

- effective and novel way of training using GPUs
- large scale
- the use of new activation function (ReLU)
- effective prevention of overfitting
- overlapping pooling



(2014) Generative Adversarial Networks (GAN)

Example of generative AI

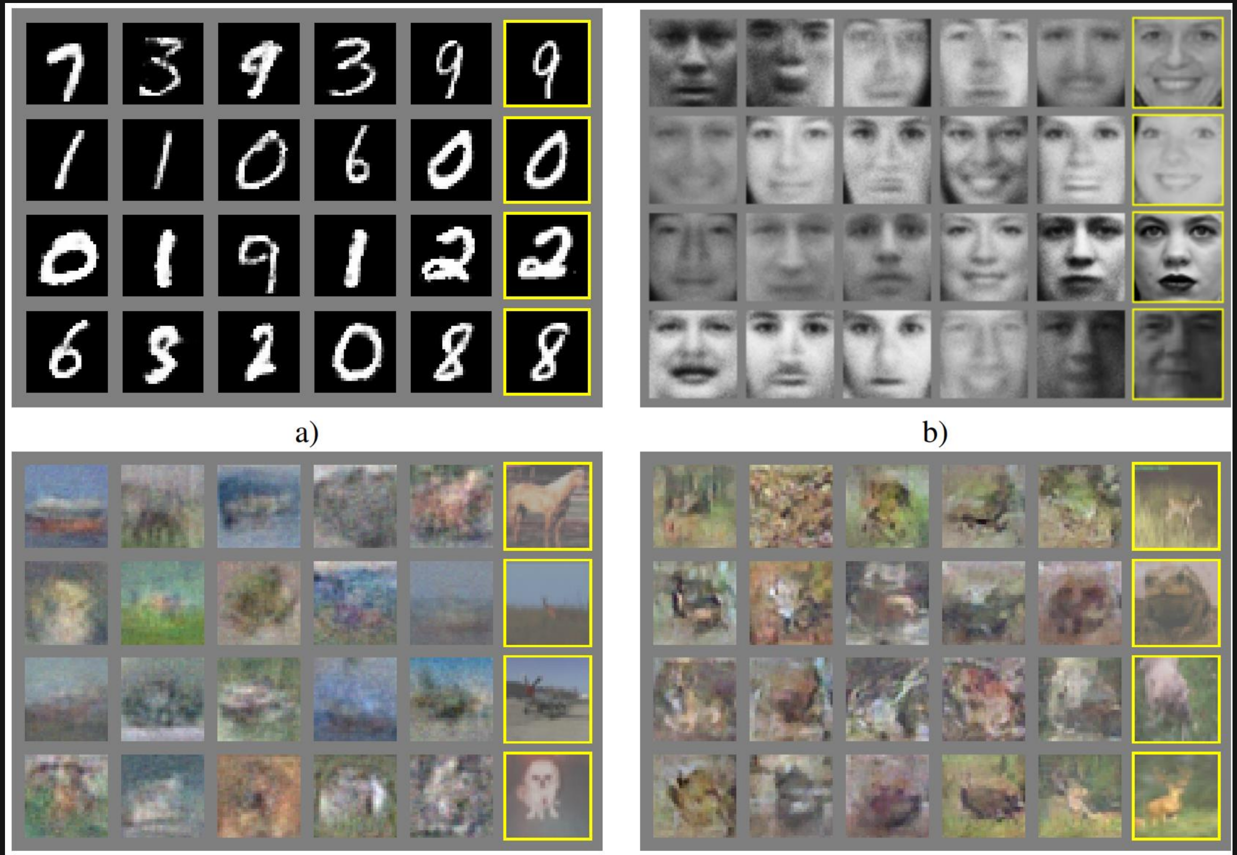
Two networks working in tandem:

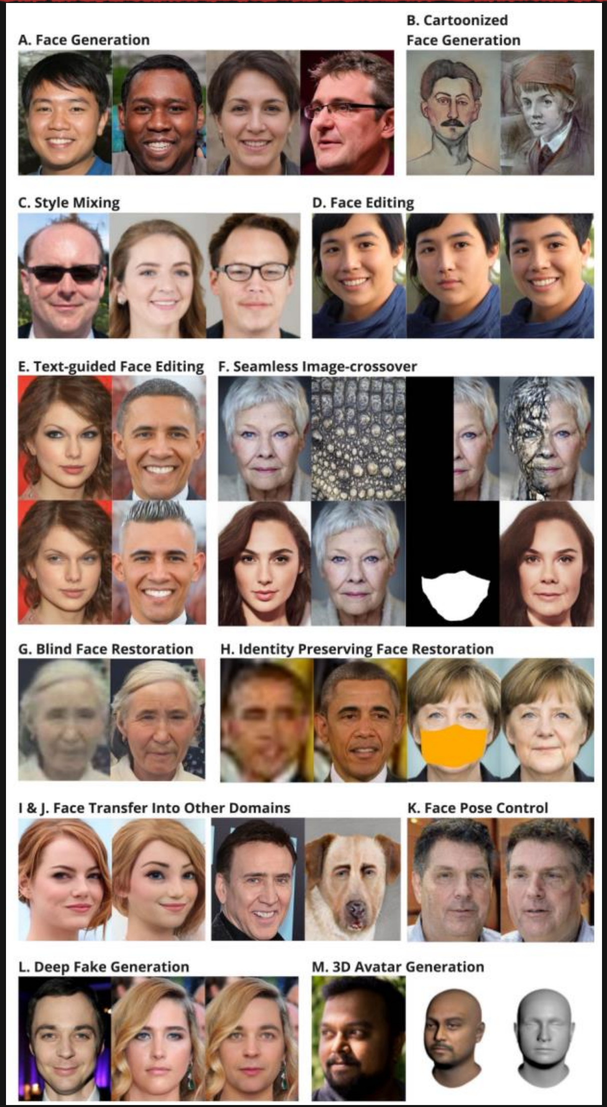
- Generator (like a forger)
- Discriminator (like a policeman)

Generator learns a distribution from which training examples would most likely be sampled.

Discriminator learns to distinguish fake (generated) examples from real ones.

(2014) Generative Adversarial Networks (GAN)





(2014-2015) Further advancements

2014: Adam Optimizer, “Adam: A Method for Stochastic Optimization” by Diederik P. Kingma and Jimmy Lei Ba

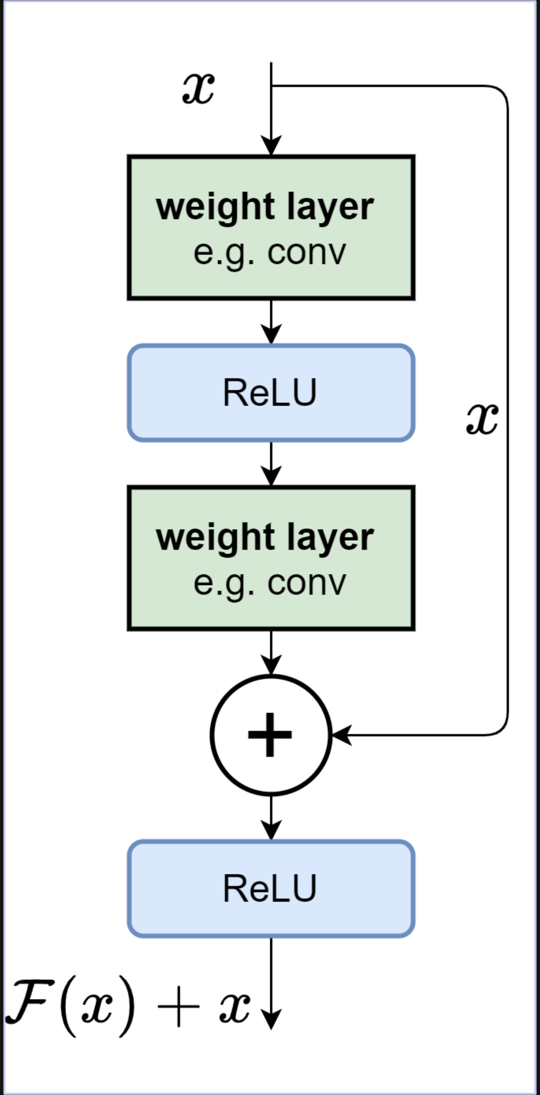
2014: “Sequence to Sequence Learning with Neural Networks” by Ilya Sutskever, Oriol Vinyals, and Quoc V. Le

(2015) Residual Networks (ResNet)

2015 "Deep Residual Learning for Image Recognition"

by K. He, X. Zhang, S.Ren, and J. Sun

- ResNet addresses the so-called vanishing/exploding gradient problem.
- Enable to train extremely large models.



(2016) Google’s Neural Machine Translation

2016 “Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation” by Yonghui Wu et al.

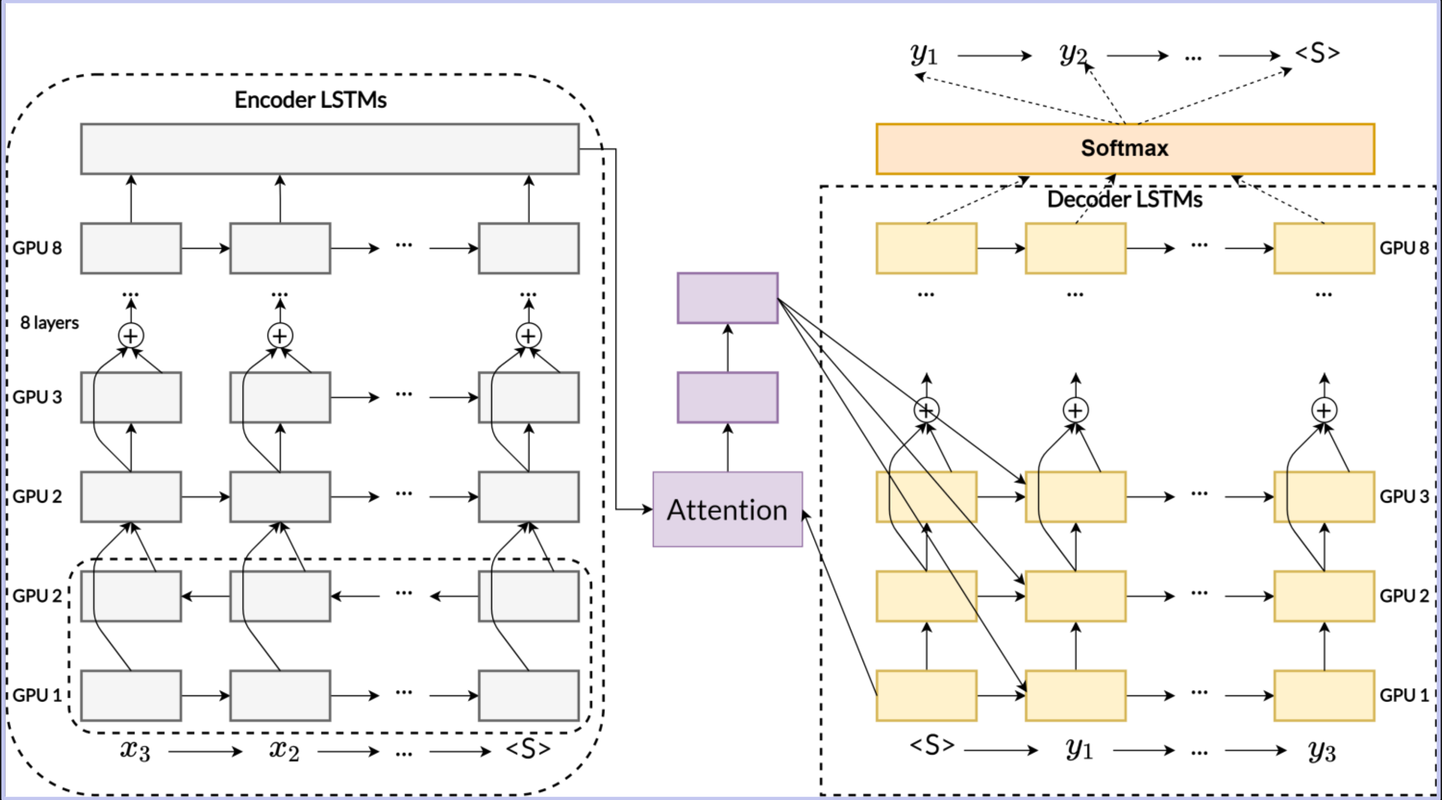
Google Translate works much better since 2016.

This was initially released in secrecy but Japanese and Korean users suddenly noticed the difference in translation quality.

(2016) Google's Neural Machine Translation

Uses:

- large autoencoder network
- encoder and decoder use LSTMs
- end-to-end process
- wordpiece model
- attention mechanism



(2017) Transformers

2017 "Attention is All You Need" by researchers from Google and University of Toronto.

A very powerful model that is currently state-of-the-art

Backbone of all large language models (LLMs) such as Chat-GPT.

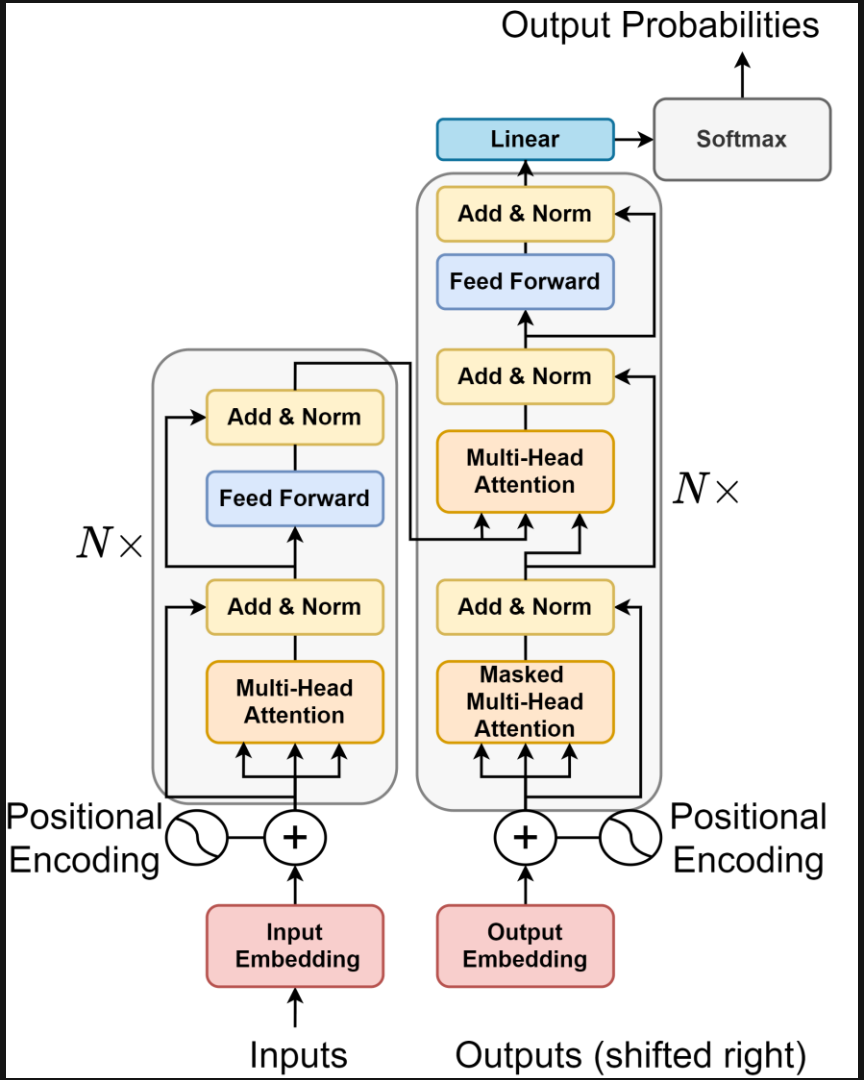
Also used for computer vision (vision transformer).

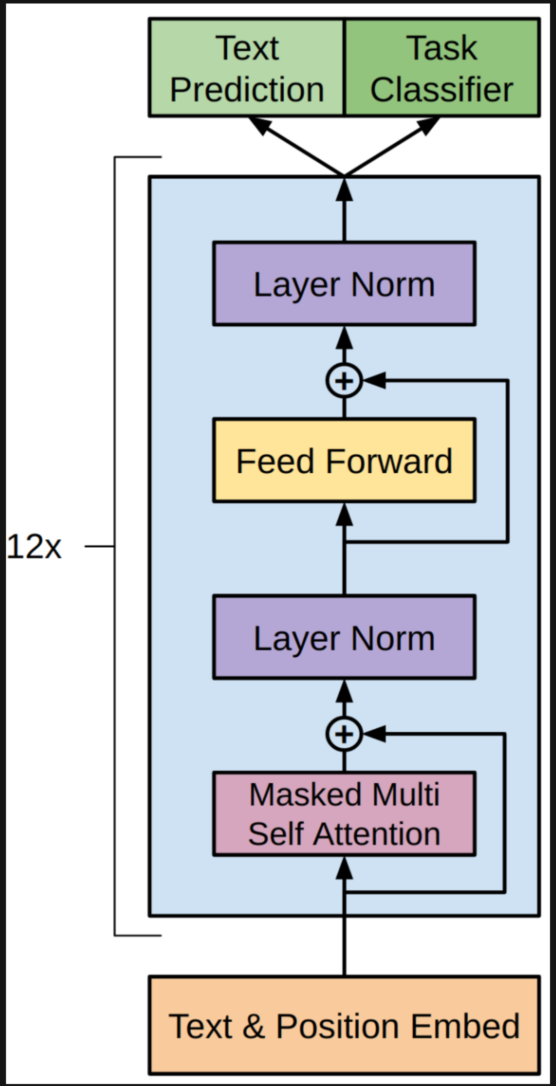
(2017) Transformers

Work with natural language.

Originally contained **encoder** and **decoder** parts but some transformers use only one type.

Attention mechanism focuses on relevant parts of the input.





The history of game-playing AI

(1928/1944) The Min-Max Algorithm & Game Theory

John von Neumann (1903 – 1957)

- Like A. Turing, one of the fathers of computer science
- Known for *Von Neumann architecture*, work at Los Alamos and many other contributions

1928: "On the Theory of Games of Strategy", published in German

1944: "Theory of Games and Economic Behavior" by Oskar Morgenstern and John von Neumann

Game Tree – some terms

Imperfect (incomplete) information vs. perfect (complete information)

Payoff – numerical result assigned to player at the end

- typically **1.0** for win; **0.0** or **-1.0** for loss; 0.5 or 0.0 for draw

Constant sum and zero-sum property

- payoffs sum to a **constant value** or **zero**, respectively

Finiteness

Synchronous vs. asynchronous moves

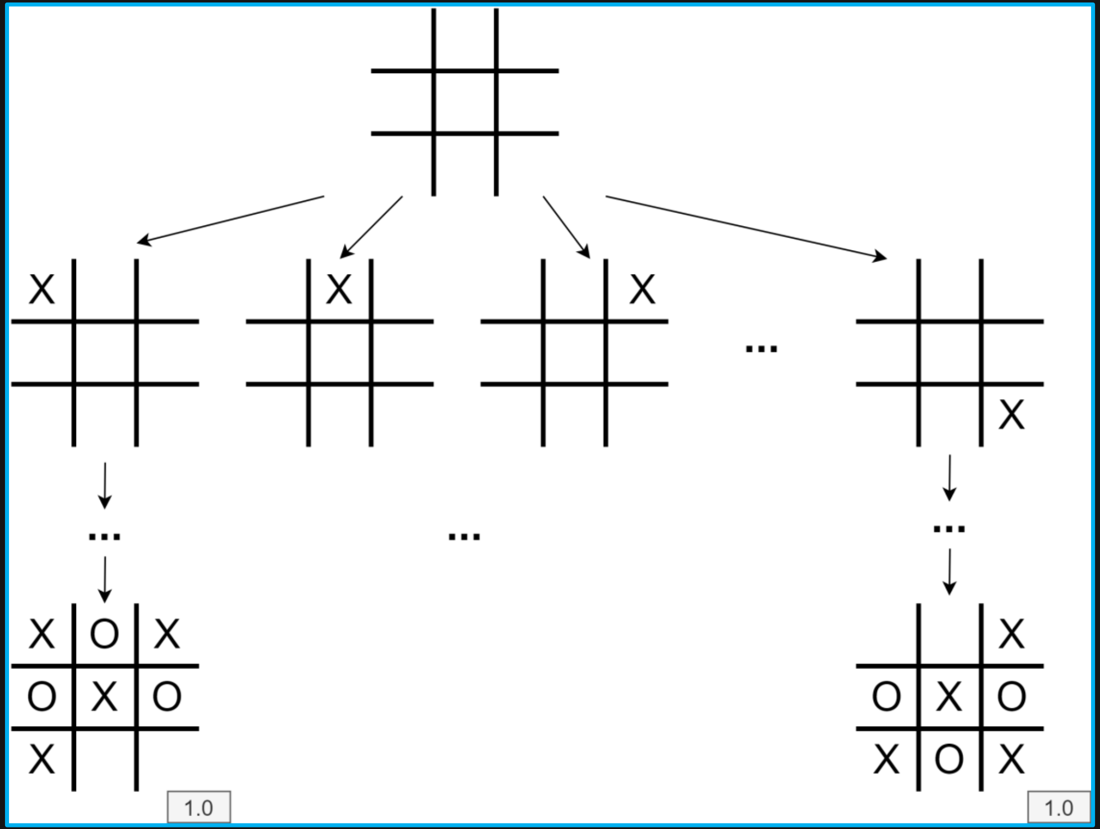
Branching factor

Deterministic vs. non-deterministic

Game Tree

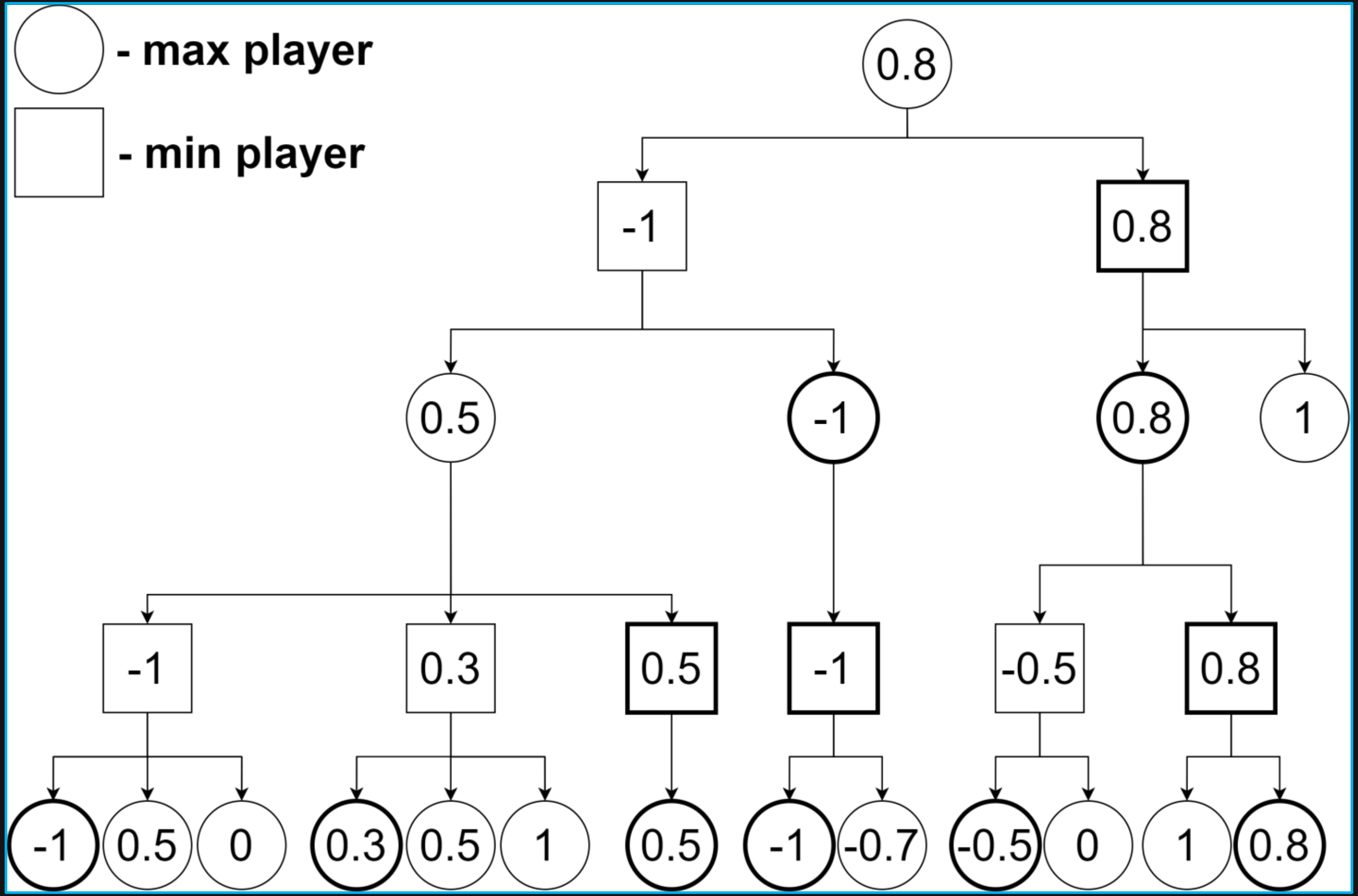
There are many formal ways to represent games such as extensive form or normal form.

Game tree is the most popular way for combinatorial games.



Min-max algorithm

Allows to find a game theoretically optimal value (and thus the best strategy to play too).



Problem:

Requires traversing the entire tree, which is infeasible for non-trivial games.

(1959) The first Checkers Program

1959: **“Some studies in machine learning using the game of checkers” by Arthur Samuel**

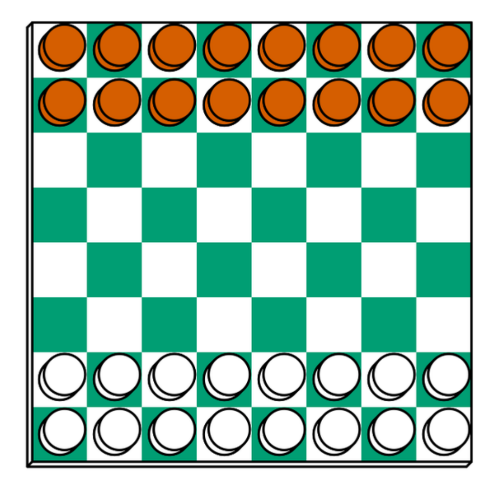
This paper advertised games as a promising benchmark for measuring progress of Artificial Intelligence

- there exist no deterministic algorithm that will guarantee a win, in practical sense (verified later for Checkers)
- there exist a formal definite goal – to win the game
- there exist intermediate goals that help (e.g. the number of pieces of each color on the board)
- the rules of the game are definite, known and formally defined
- there is a background of knowledge (e.g. history of games between human players)
- the game is familiar to a substantial body of people
- game-based research environment is cheap and controllable

(1959) The first Checkers Program

Introduction of a heuristic evaluation function

heuristic(state) -> number

f() = 0.5

The proposed function was a polynomial – linear combination of features.

The features were: **32 board positions** and 6 aggregated information including **piece advantage or disadvantage**.

(1959) The first Checkers Program

Context-dependent search depth based on simple conditions (e.g. whether exchange occurs or the next move is a jump)

Catalogue – selected stored board positions, handling symmetries and redundant positions

At the time of publications, contained 53000 records.

Simple learning mechanism of the strengths of the catalogued positions.

Foundational ideas for opening books, transposition tables and endgame databases.

(1969) A* Algorithm

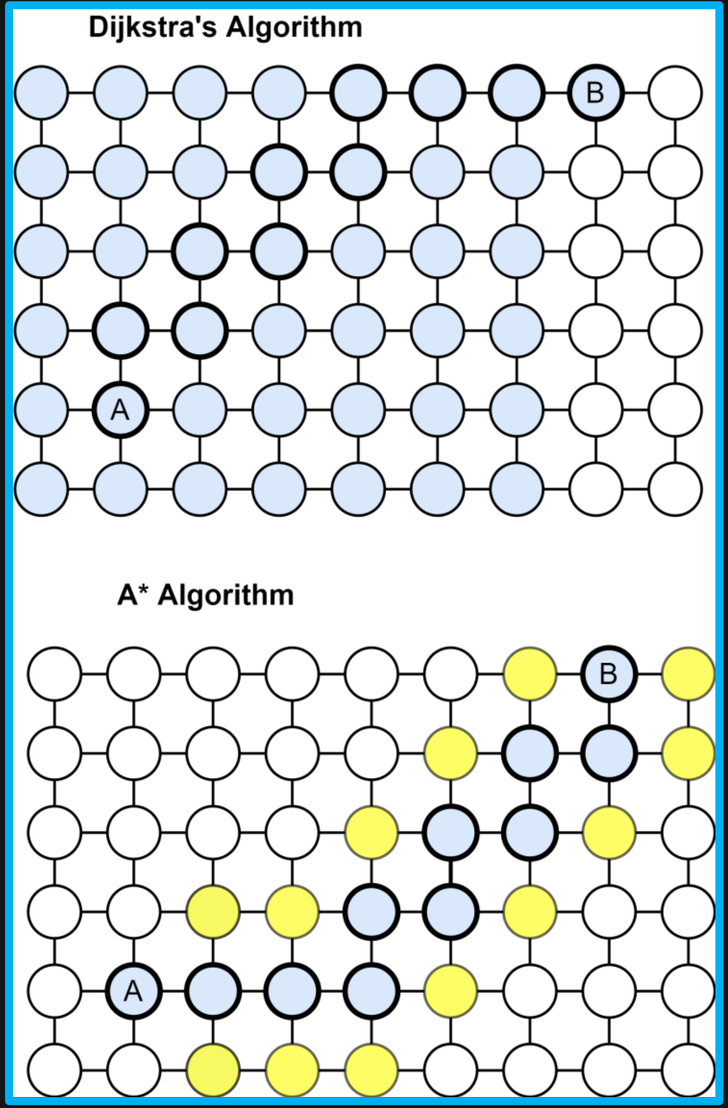
1969: **"A Formal Basis for the Heuristic Determination of Minimum Cost Paths" by Peter E. Hart, Nils J. Nilsson, and Bertram Raphael.**

Very effective algorithm for searching abstract spaces using a distance heuristic.

Used till this day for:

- pathfinding in video games
- Searching game trees of single-player games (aka puzzles)

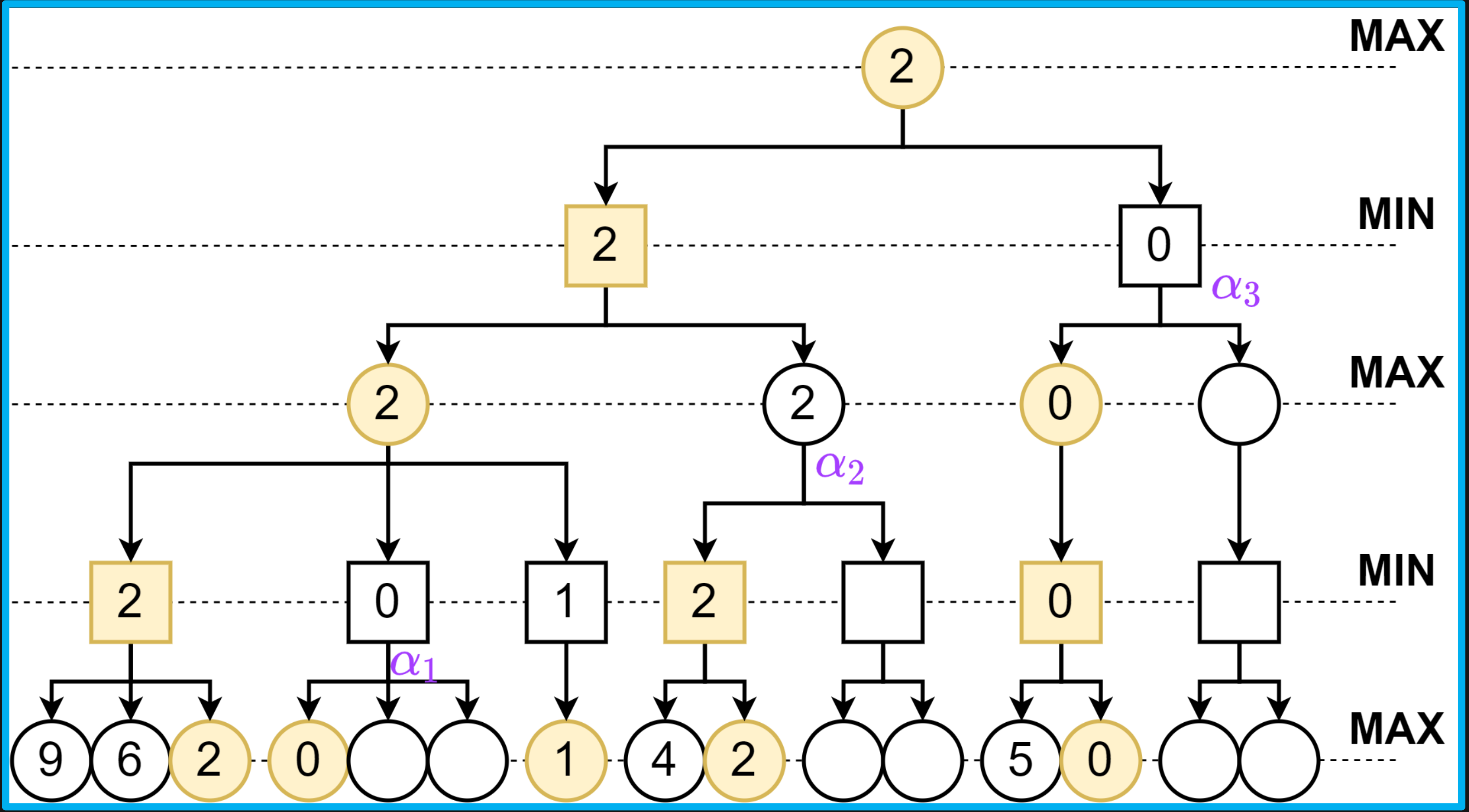
(1969) A* Algorithm vs. exact algorithm



(1975) Alpha-Beta Pruning

1975: "An analysis of alpha-beta pruning" by **Donald Knuth**

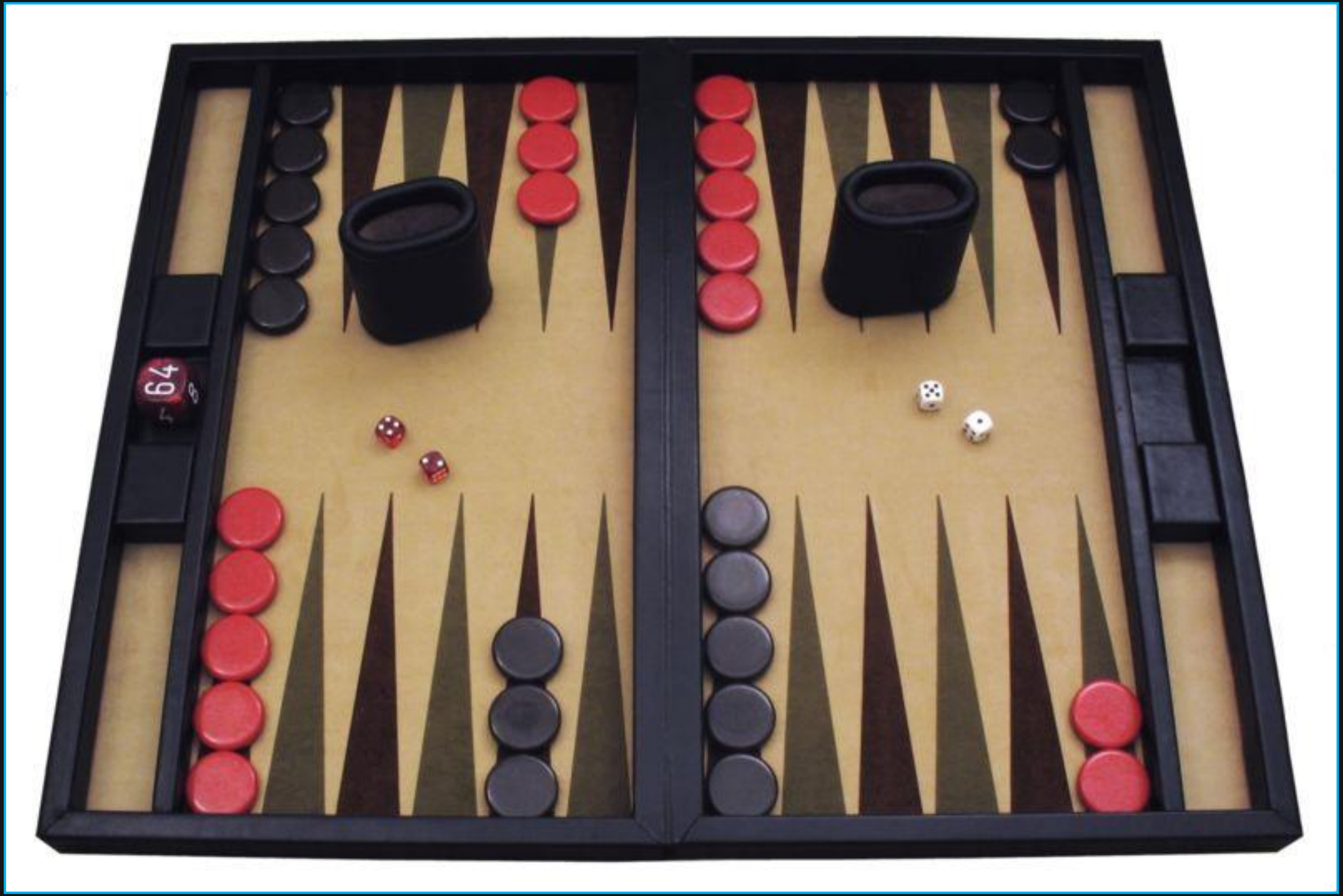
- Optimization of min-max when heuristic evaluation function is available
- Reduces the number of visited nodes



(1992) TD-Gammon

For the game **Backgammon**

Relatively complex game with non-determinism (dice rolls)



(1992) TD-Gammon

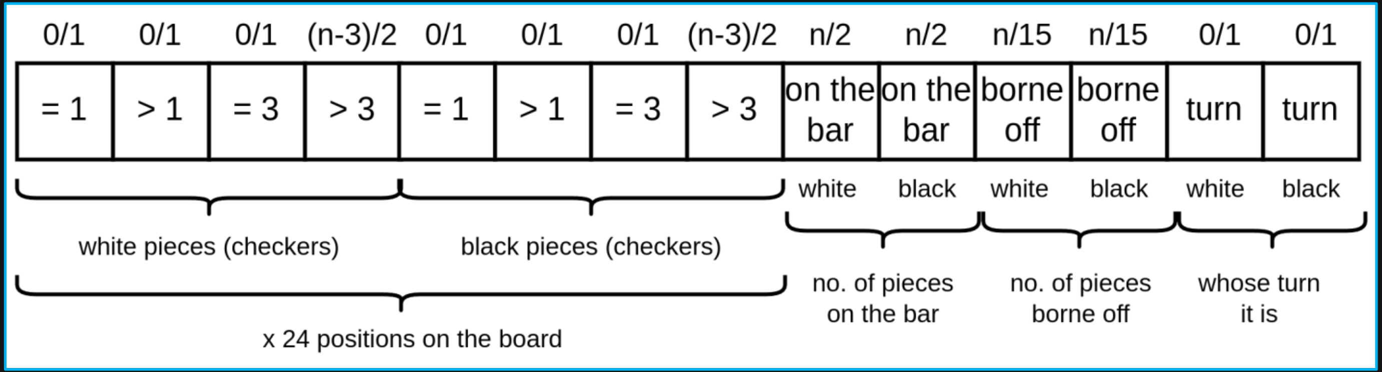
1992: release by **Gerald Tesauro**

1995: paper “Temporal Difference Learning and TD-Gammon”

Purely knowledge-free and learning-based

Used:

- **Neural network model** (fully-connected MLP)
- **TD-learning algorithm** (one of its most famous applications)



(1992) TD-Gammon

TD-Gammon was a strong player but weaker than the top humans.

After 200 000 training games: stopped being a weak player
After 1 500 000 training games: no more improvement

(1992) TD-Gammon

TD-Gammon's exclusive training through self-play (rather than tutelage) enabled it to explore strategies that humans previously hadn't considered or had ruled out erroneously.

Its success with unorthodox strategies had a significant impact on the backgammon community – Tesauro 1995

(1997) IBM Deep Blue

Chess became known as “the drosophila of AI”

Garry Kasparov – Deep Blue

- played between May 3-rd and May 11-th, 1997
- in Equitable Center in New York
- prize money of $1.1 million + $700K for Kasparov just for participation
- huge media coverage

(1997) IBM Deep Blue

Deep Blue won 3 – 1 – 2 against Kasparov

Kasparov accused Deep Blue team of cheating

(1997) IBM Deep Blue

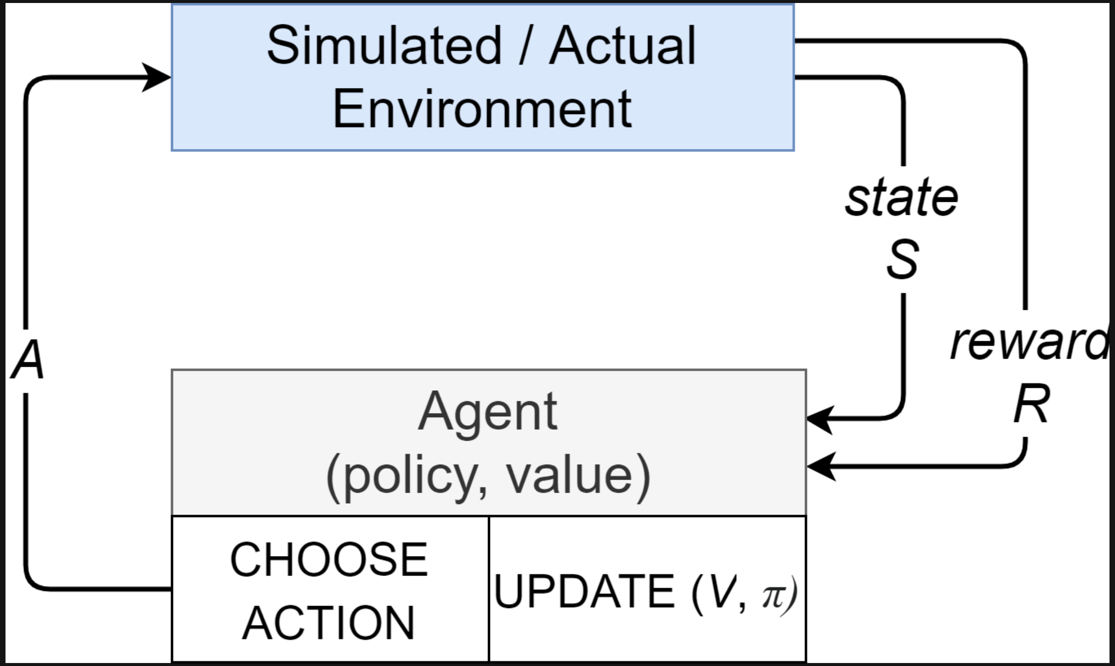
- **Supercomputer, 259-th place at top500**
  - currently at 259-th place: 126458 cores; 3.94PFlop/s; 1.88 MW power consumption
- **Specialized chess computing chips**
- **Depending on the position capable of searching up to 20 moves ahead**
- **Opening book of 4000 states**
- **Zero-windows alpha-beta pruning search**
- **Heuristic evaluation function of 8000 features**
- **Endgame database with each position up to 5 pieces left on the board**

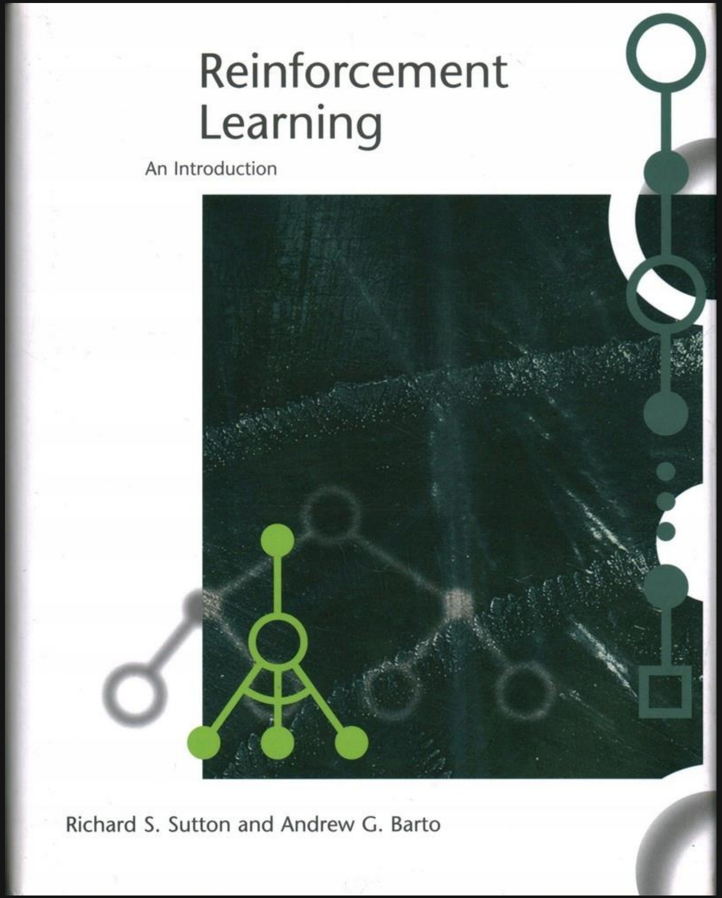
(1997) Logistello

Logistello wins 6-0 against Takeshi Murakami (the 1997 world champion and the then best player) in Othello (Reversi).

(1998) Reinforcement Learning: An introduction

Reinforcement Learning is currently one of the major approaches *(alongside search and planning ones)* **to creating strong computer game players.**





(2003) Behavior Trees

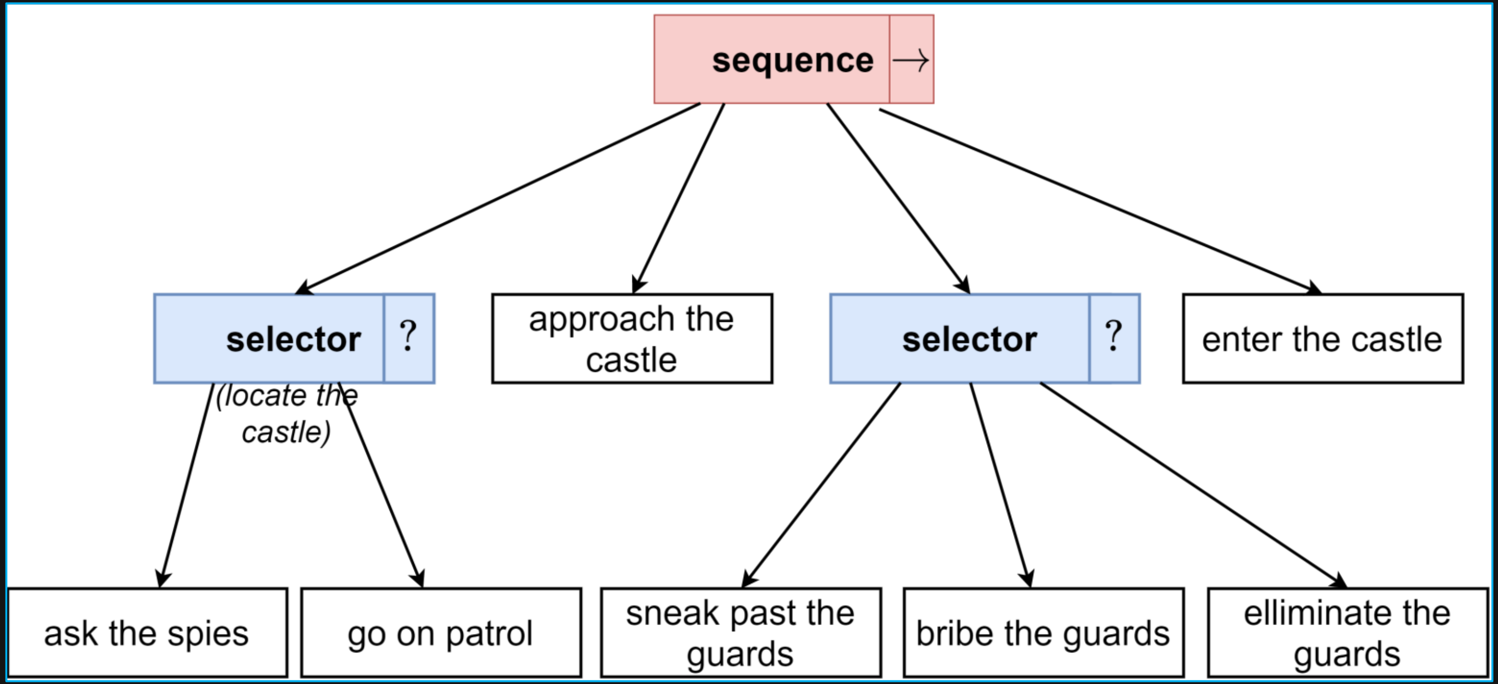
2003: "From Requirements to Design: Formalizing the Key Steps" by Geoff Dromey

Initially applied for controlling robots

One of the most used method to control the behavior of NPCs in video games

Visual scripting

- **no learning involved**



(2006) Monte Carlo Tree Search

2006: "Efficient Selectivity and Backup Operators in Monte Carlo Tree Search" by Rémi Coulom

2006: "Bandit Based Monte-Carlo Planning" by Levente Kocsis and Csaba Szepesvári

Originally proposed for Go

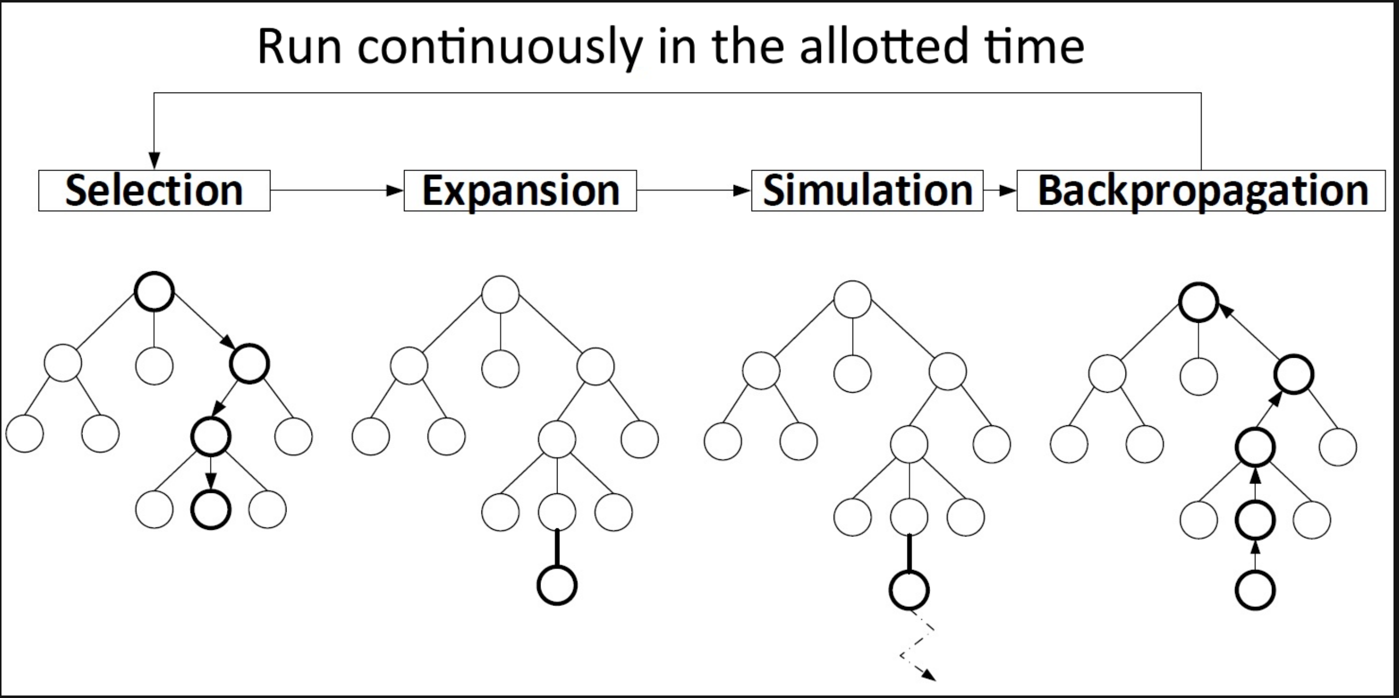
- leap from 14 kyu (weak amateur) to 5 dan (advanced player)

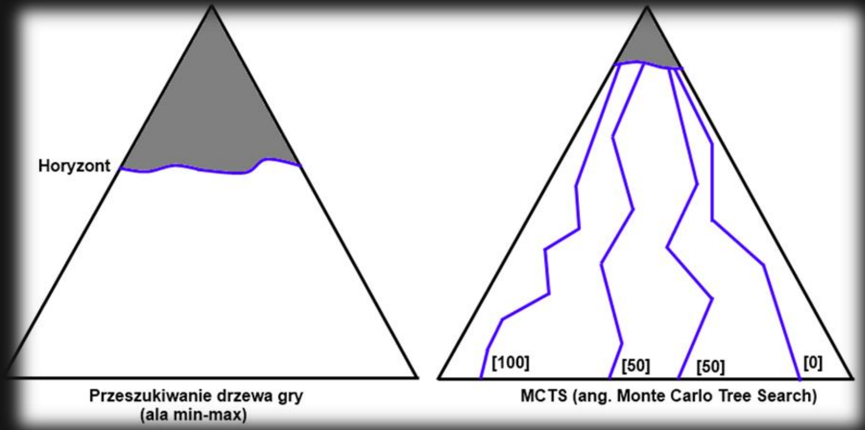
Novel tree search method, which is neither requires heuristic evaluation function nor searching the whole tree.

(2006) Monte Carlo Tree Search

Grows an assymetric tree

“Samples” results of games by random or pseudo-random playouts to the end





(2006) Monte Carlo Tree Search

MCTS/UCT algorithm balances **exploitation** (of the best actions discovered so far) and **exploration** (of unknown lines of play)

Inspired by optimal strategy to play one-armed bandits

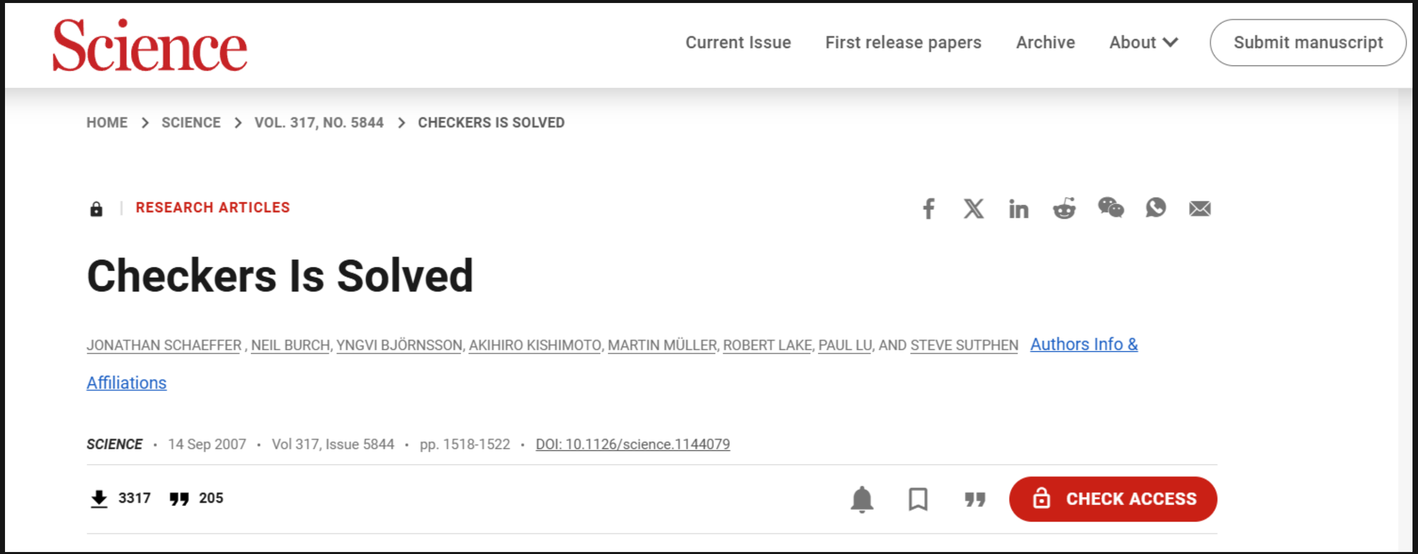
(2007) Checkers is Solved!

It took 18 years of continuous calculations from 1989 to 2007

At peak: 200 processors used simultaneously

Optimal play from both sides ends in a draw

The resulting player - **Chinook** - is unbeatable



(2008) Stockfish

Open source project started by T. Romstad, M. Costalba, and J. Kiiskiin in 2008

https://stockfishchess.org

Arguably the strongest chess player up to date

Estimated ELO rating: 3664

Highest ELO rating ever of a human: 2882 (Magnus Carlsen)

In Top Chess Engine Championship:

- 15x first place (including 2024)
- 5x second place
- never below the second place

(2008) Stockfish

Combines:

- opening books
- complex heuristic evaluation function
- effective iterative deepening search
- end-game database

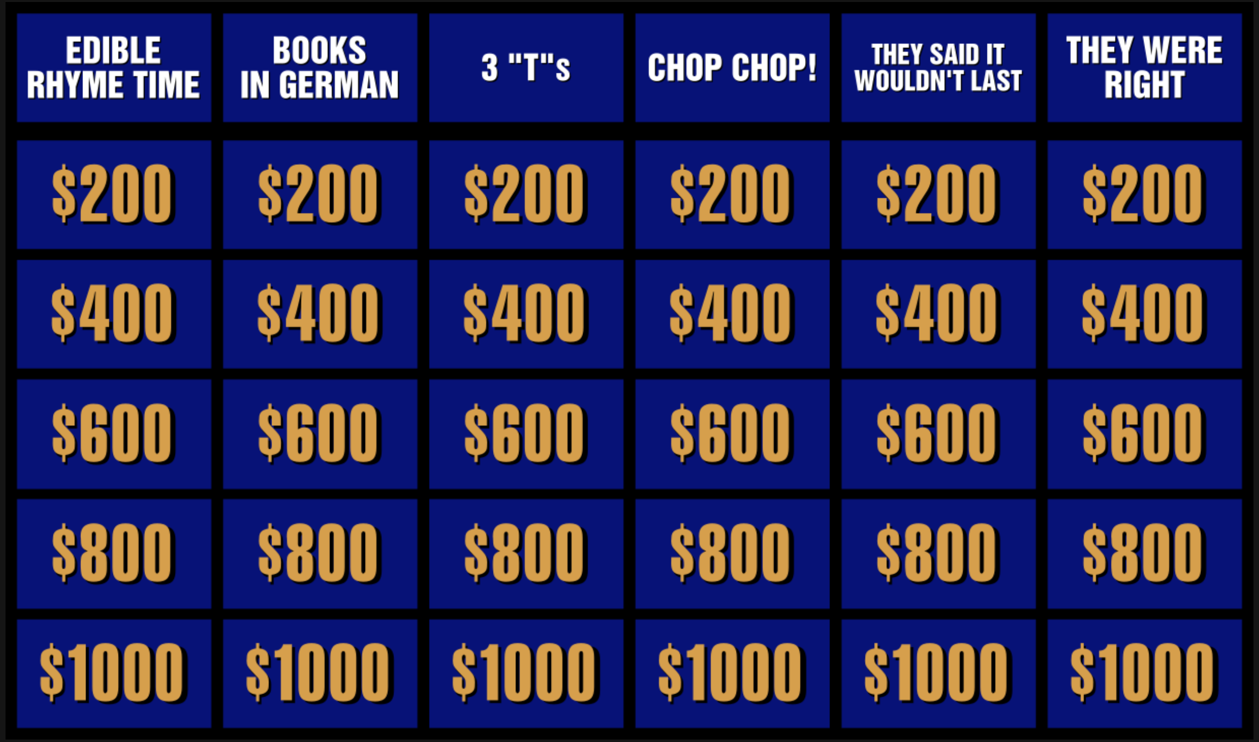
Since 2020, Stockfish uses neural networks

As of 2024: it played over 7 billion chess games across all supported devices

(2011) Watson

Named after the first CEO of IBM

Initially designed to play similar role to Deep Blue but in Jeopardy!



(2011) Watson

Development started in 2006

Feb 14-th 2011, IBM Watson appeared on TV for the first time

Game 1:

- Watson won $35734 (humans: $10400 and $4800)

Game 2:

- Watson won $70147 (humans: $2400 and $21600)

Grand final:

- Watson won $1 million (donated to charity by IBM)

“I for one welcome our new computer overlords”

(2011) Watson

- Record winnings
- Vastly outperformed human players
- Not only answered but also waged well
- Was offline but had access to downloaded 2011 Wikipedia
- The most problem it had with understanding other players' incorrect answers
- The best query-answering system at the time

(2011) Watson

IBM Watson became a commercial product

The first applications of Watson were in:

- Memorial Sloan Kettering Cancer Center as a clinical-decision support tool for cancer
- Cleveland Clinic as a training tool for medical students
- MD Anderson Cancer Center as a clinical-decision support tool for cancer
- New York Genome Center as a genomic-analysis tool for brain cancer
- GenieMD as a consumer app for personalized medical advice
- Mayo Clinic as a clinical-trial matching too

(2013) Deep Q-Learning

2013: "Playing Atari with Deep Reinforcement Learning" by **DeepMind**

A novel combination of deep neural networks and reinforcement learning

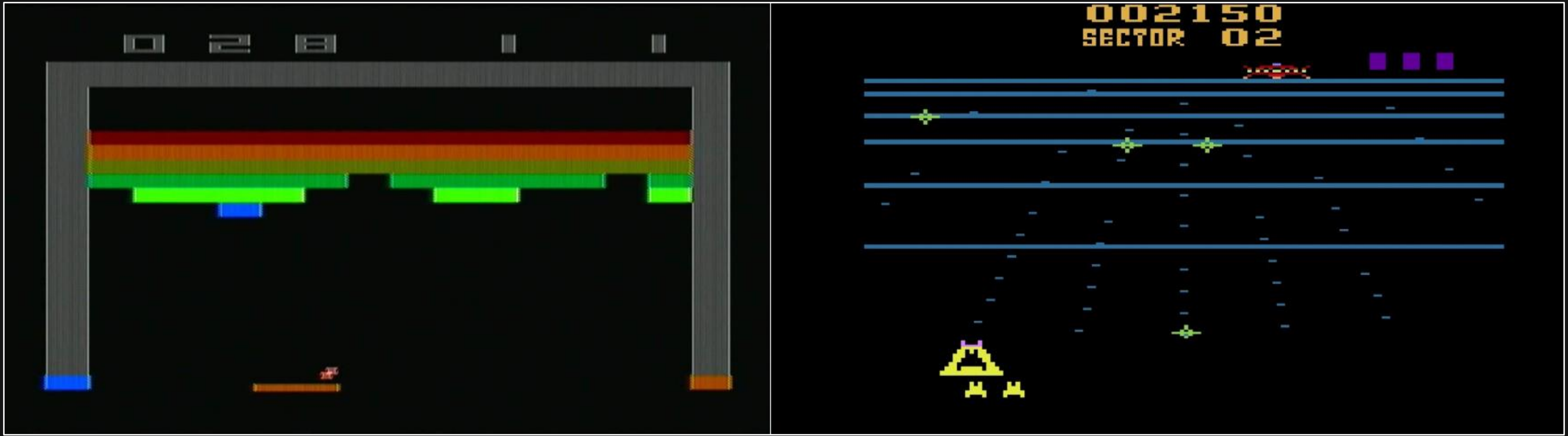
DeepMind is an AI R&D start-up founded by **Demis Hassabis** *(2024 Nobel Prize winner)*, **Shane Legg and Mustafa Suleyman** in 2010 in London.

Acquired by Google in 2014 reportedly for at least 400-500 million USD.

(2013) Deep Q-Learning

Processing raw pixel input

One model to play 57 Atari games

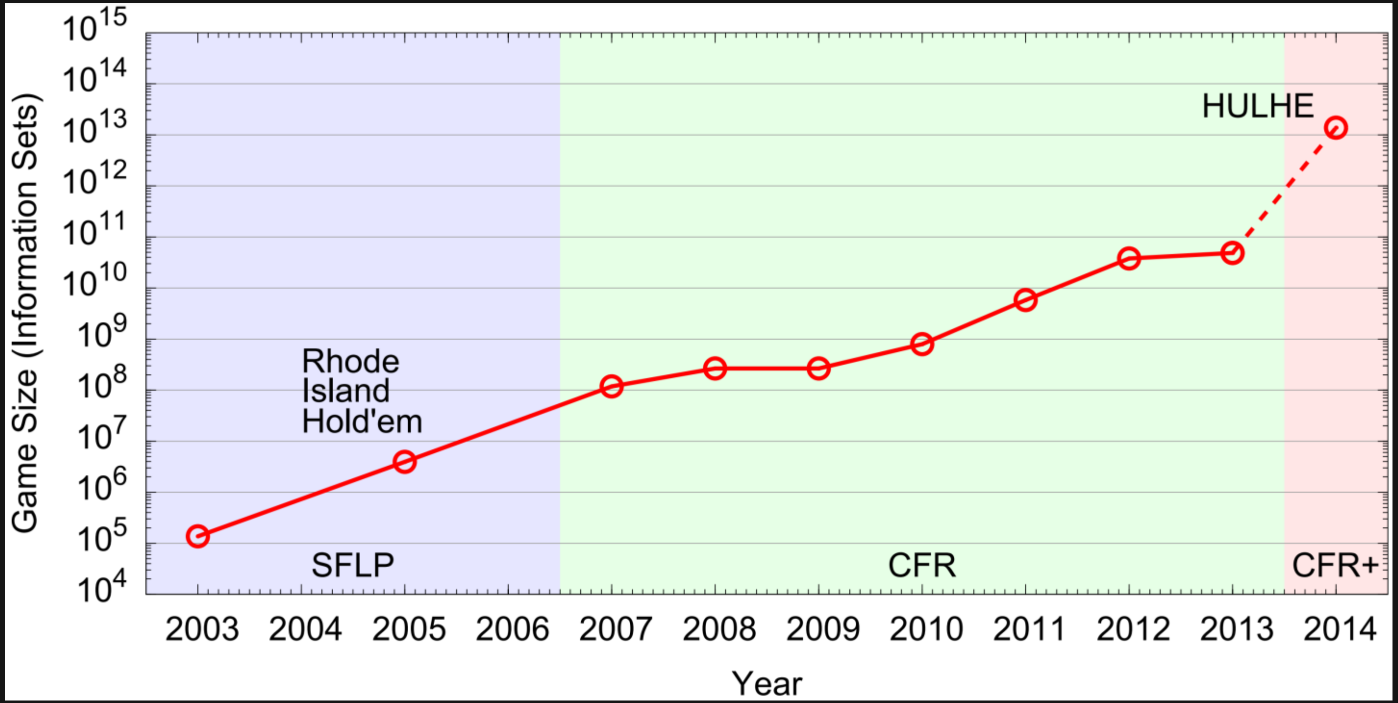


(2015) HULH Poker is Solved

2015: Two-player variant of Texas Hold'em poker with fixed limited bets was weakly solved.

Researchers at University of Alberta

Weakly solved: can play optimally from the beginning but not from any position



(2016) AlphaGo

Developed by IBM DeepMind

One of the most famous breakthroughs in AI up t

- Go had been a notoriously difficult game for computers
- AlphaGo achieved amazing result
- AlphaGo was a proof-of-concept for deep learning
- Foundation for many subsequent approaches including **AlphaZero** and **AlphaFold**

(2016) AlphaGo vs. Lee Sedol

2015: AlphaGo defeats 2012-2014 European Go Champion – Fan Hui.

2016: 5-game match against Lee Sedol, 9-dan, a professional Go player from South Korea, one of the greatest in the history of the game

- the match took place in Four Seasons Hotel in Seol, South Korea, March 9-15-th

Game 1: AlphaGo won (playing White)

Game 2: AlphaGo won (playing Black)

Game 3: AlphaGo won (playing White)

Game 4: Lee Sedol won (playing White)

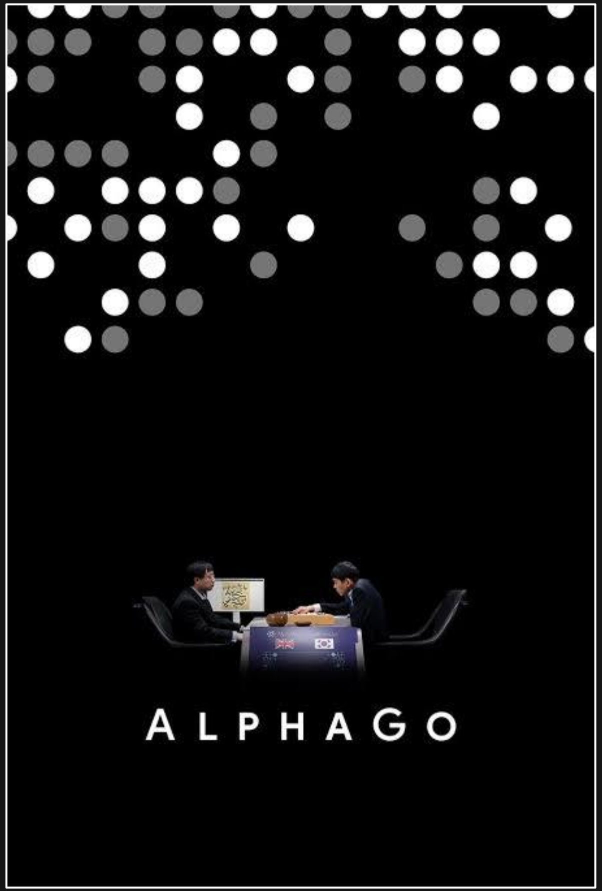
Game 5: AlphaGo won (playing White)

Lee Sedol retired from the game.

(2016) AlphaGo

2016: Mastering the game of Go with deep neural networks and tree search

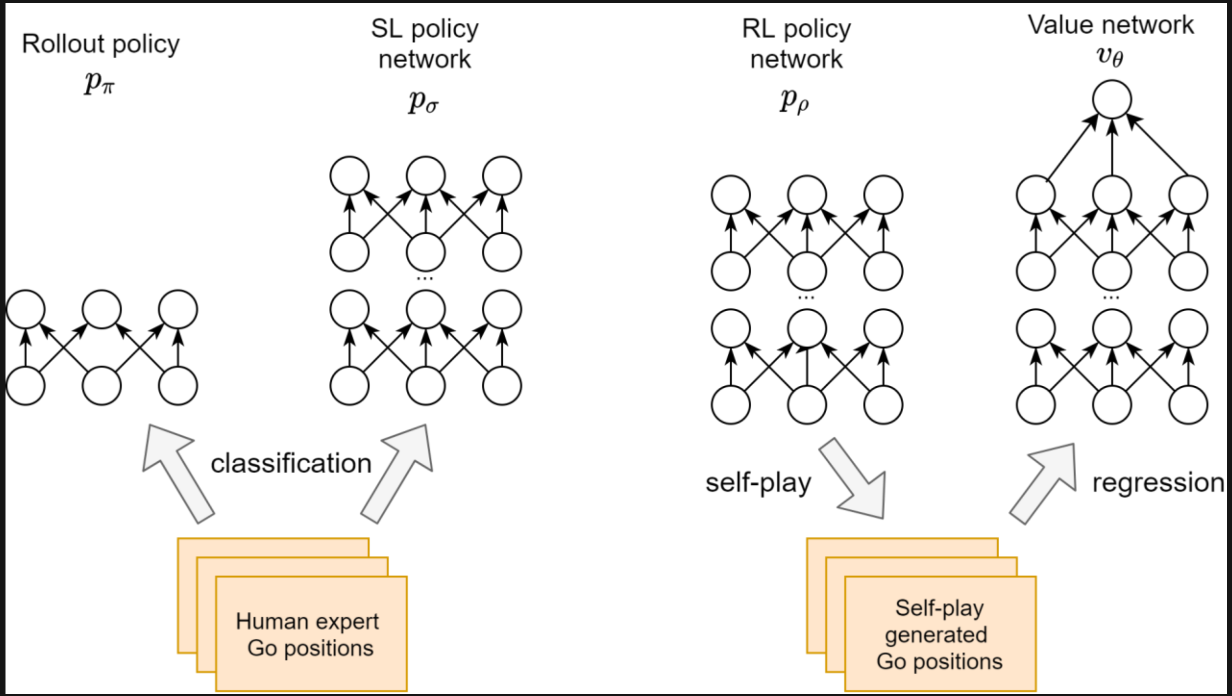
There is even a 2017 documentary by Netflix about AlphaGo



(2016) AlphaGo

AlphaGo combined:

- Supervised learning
- Deep reinforcement learning
- Monte Carlo Tree Search
- Training on 1202 CPUs and 178 GPUs
- Many novel tricks



Later, AlphaZero was able to play two more games and without supervised learning part

(2018) AlphaStar

A bot made to play Starcraft 2, a largely popular e-sport real-time strategy game

Developed by IBM DeepMind

Full game tree size ~ 10^{1685}

Branching factor ~ 10^{50}

(2018) AlphaStar

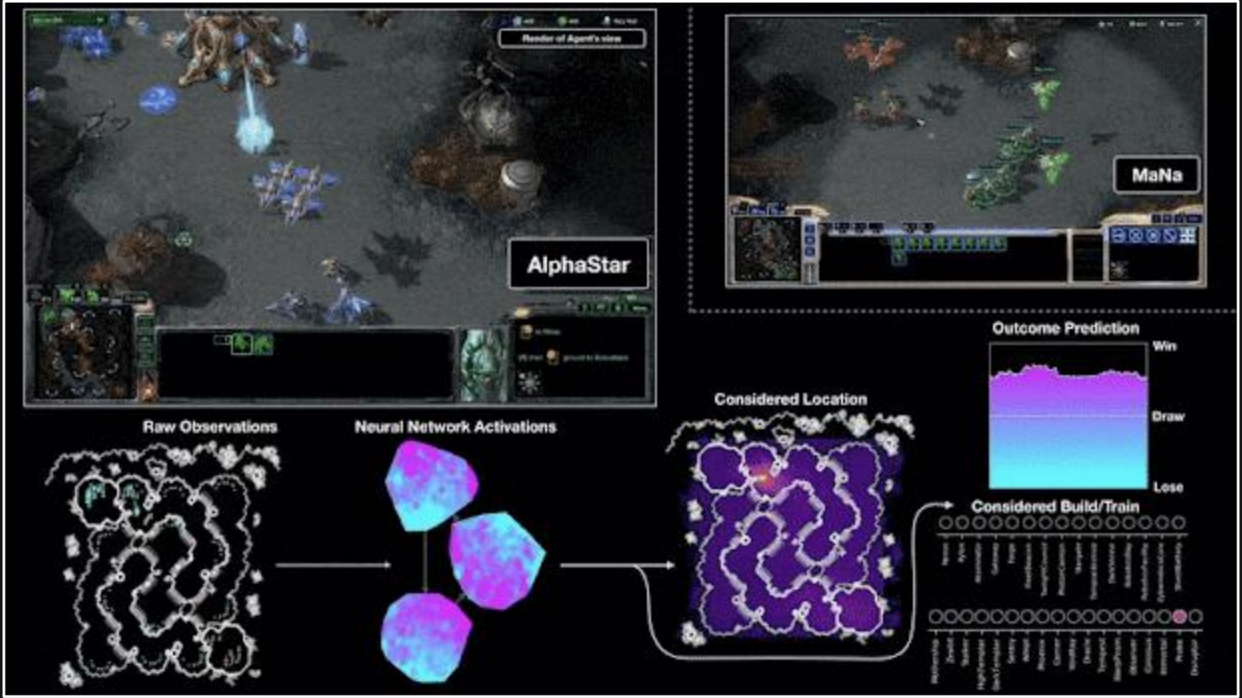
2018: AlphaStar 5 – 0 Dario "TLO" Wünsch (Team Liquid)

19-th Dec, 2019: AlphaStar 5 – 0 Grzegorz "MaNa" Komincz (Team Liquid)

these matches took place at DeepMind's HQ in London

The Guardian named AlphaStar

a "landmark achievement" for the field of AI



(2018) OpenAI Five

Bot developed by **OpenAI** to play **Dota 2**

Novel scale of challenge: Dota 2 is a complex real-time video game for **5 players** in a team

(2018) OpenAI Five

2016: research on this topic starts

2017: OpenAI ready with 1-1 version, where the first player to score 2 kills is the winner

2017: A two-game showcase match against Danil “Dendi” Ishutin during Dota 2 Championship

- both contestants use the same hero: Shadow Fiend
- **the bot wins both games**
- during the event the bot played several other less experienced players and won all the games

2018: OpenAI Five ready to play the regular 5-hero game

2019: OpenAI Five defeats the 2018 Dota 2 Championship winning human team during a live event which was also streamed live at Twitch.tv

It was the first time when a non-human player defeated top human professionals at a complex e-sports game during a live event.

(2018) OpenAI Five

Technology:

- totally self-learned using reinforcement learning (Proximal Policy Optimization (PPO))
- large neural networks
- **it played equivalent of 180 years worth of games every day**
- 1200 categorical and 14534 continuous/boolean values

Potential problems:

- **just a few games against professionals, human players could be surprised**
- **following 2019, OpenAI Five was hosted online; it played 40000 games winning 99.4%** *(meaning it is not unbeatable as it lost 240 games)*

The history of other AI tools, techniques and events

overview

Overview of various other milestones in AI

Year	What happened	Type of significance
1763	Bayes Theorem	mathematical foundation
1901	Principal Component Analysis (PCA)	data science method
1936	IRIS Dataset	important dataset for research
1948	Mathematical Theory of Communication	mathematical foundation
1951	K-Nearest Neighbors	data science / ML method
1950-60	Development of the First Clustering Methods	data science / ML method
1957	Markovian Decision Process (MDP)	mathematical foundation
1959	Programs with Common Sense	logic-based AI
1961	General Problem Solver	logic-based AI / planning
1965	Fuzzy Sets	logic-based AI
1965	Expert Systems	heuristic AI
1966	ELIZA	the first chatbot

Overview of various other milestones in AI

Year	What happened	Type of significance
1969	**Some Philosophical Problems from the Standpoint of Artificial Intelligence**	important article
1970	**Simulated Annealing**	computational intelligence (CI) / metaheuristic method
1971	**STRIPS Planning**	planning AI method
1972-73	**Prolog**	programming language
1974	**The First AI Winter**	low funding & disappointment time
1975	**The Recognition of Genetic Algorithms**	CI / metaheuristic method
1975	**Hierarchical Partial-Order Planning**	planning AI method
1976	**HARPY**	speech recognition system
1977	**EM algorithm and Mixture of Gaussians**	mathematical foundation / ML method
1979	**Stanford Cart**	predecessor of autonomous cars
1979-80	**LISP Machines**	AI-enabling hardware

Overview of various other milestones in AI

Year	What happened	Type of significance
1980	The First AAAI and ICML Conferences	events
1986	Decision Trees	ML method
1986	Artificial Immune Systems	CI / metaheuristic method
1987-95	PROMETHEUS	the first autonomous car project
1987	The Second AI Winter	low funding & disappointment time
1988	Probabilistic Reasoning and Bayesian Nets	ML method
1988	Temporal Difference Learning (TD-Learning)	ML method
1991	Ant Colony Optimization (ACO)	CI / metaheuristic method
1991	Python	programming language
1993	R	programming language
1993	Association Rules Mining	data science method
1995	Particle Swarm Optimization (PSO)	CI / metaheuristic method

Overview of various other milestones in AI

Year	What happened	Type of significance
1995	Support Vector Machines (SVM)	ML method
1995	Artificial Intelligence: A Modern Approach	book
1996	Covariance Matrix Adaptation Evolution Strategy (CMA-ES)	CI / metaheuristic method
1996	Computing with Words	CI / logic-based AI
1997	AdaBoost	ML method
2001	Random Forests	ML method
2002	Neuro-Evolution of Augmenting Topologies (NEAT)	CI + ML method
2002	Designing Sociable Robots	book
2002	Roomba	product
2004	DARPA Grand Challenge (for autonomous cars)	prized competition
2005	Probabilistic Robots	book
2005	General Game Playing	prized competition

Overview of various other milestones in AI

Year	What happened	Type of significance
2007	**GPGPU Programming**	hardware
2009	**Netflix Prize Claimed**	milestone
2010	**Kaggle Platform**	service for competitions
2013	**Word2Vec**	NLP ML method
2015	**The Rise of Explainable AI (XAI)**	ML methods
2015	**Waymo Car fully autonomous drive**	milestone
2015	**Extreme Gradient Boosting (XGBoost)**	ML method
2018	**The First Large Language Models: GPT and BERT**	milestone
2018	**Soft Actor-Critic (SAC)**	ML method
2019	**Emergent Tool Use From Multi-Agent Interaction**	important experiment
2020	**AlphaFold** (in 2024 lead to Nobel Prizes)	milestone
2022	**Chat-GPT**	

(2024) Nobel Prizes

In physics:

John Hopfield: for physics-inspired Hopfield networks

Geoffrey Hinton: for Boltzman machines and life-time deep learning contributions

In chemistry:

David Baker, Demis Hassabis, and John Jumper: for their advancements in protein folding predictions.

Enabled a quantum leap from 100K known structures to over 200 million.

Thank you for your attention!

maciej.swiechowski@qed.pl

https://www.cyberiada.eu

https://www.linkedin.com/in/maciej-%C5%9Bwiechowski-6053a21ab