Lecture 1

From intelligent machine to machine learning

1. **Intelligent machines, or what machines can do?**
2. **Turing test**
3. **The history of AI: from the “Dark Ages” to deep learning era**
4. **Machine Learning**

1. Intelligent machines, or what machines can do?

- Philosophers have been trying for over 2000 years to understand and resolve two *Big Questions* of the Universe: **How does a human mind work, and Can non-humans have minds?** These questions are still not well answered.
  (Here is one answer https://www.newworldai.com/how-to-create-a-mind-the-secret-of-human-thought-revealed/)

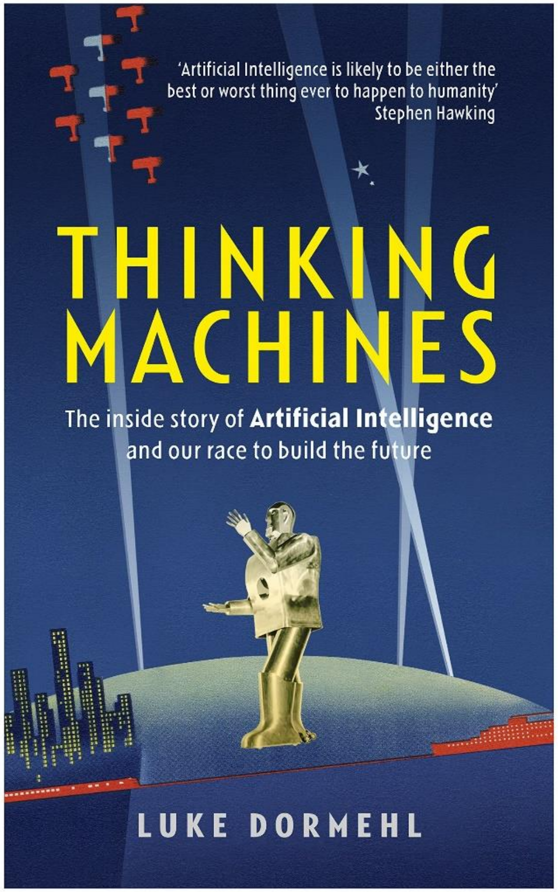
- ***Intelligence*** is the ability to understand and learn things.
  2 ***Intelligence*** is the ability to think and understand instead of doing things by instinct or automatically.

  (*Essential English Dictionary*, Collins, London, 1990)

- In order to think, some*one* or some*thing* must have a brain, or an organ that enables some*one* or some*thing* to learn and understand things, to solve problems and to make decisions. So we can define intelligence as ***the ability to learn and understand, to solve problems and to make decisions***.

- The goal of ***artificial intelligence*** (AI) as a science is to make machines do things that would require intelligence if done by humans. Therefore, the answer to the question *Can Machines Think?* was vitally important to the discipline.

- The answer is not a simple "*Yes*" or " *No* ".

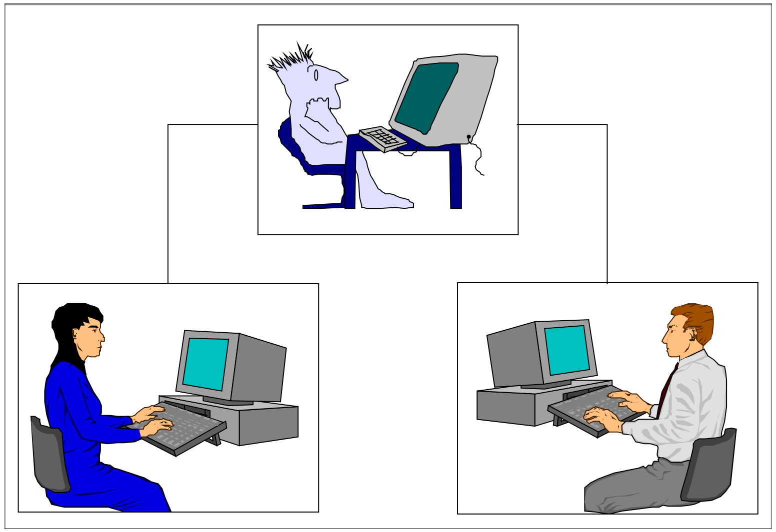


- Some people are smarter in some ways than others.
- Sometimes we make very intelligent decisions but sometimes we also make very silly mistakes.
- Some of us deal with complex mathematical and engineering problems but are moronic in philosophy and history.
- Some people are good at making money, while others are better at spending it.
- As humans, we all can learn and understand, to solve problems and to make decisions; however, **our abilities are not equal and lie in different areas**.
- Therefore, we should expect that if machines can think, some of them might be smarter than others in some ways.

2. Turing Test

- One of the most significant papers on machine intelligence, ***"Computing Machinery and Intelligence"***, was written by the British mathematician ***Alan Turing*** over fifty years ago. However, it still stands up well under the test of time, and the Turing's approach remains universal.
- He asked: ***Is there thought without experience? Is there mind without communication? Is there language without living? Is there intelligence without life?*** All these questions, as you can see, are just variations on the fundamental question of artificial intelligence, ***Can machines think?***

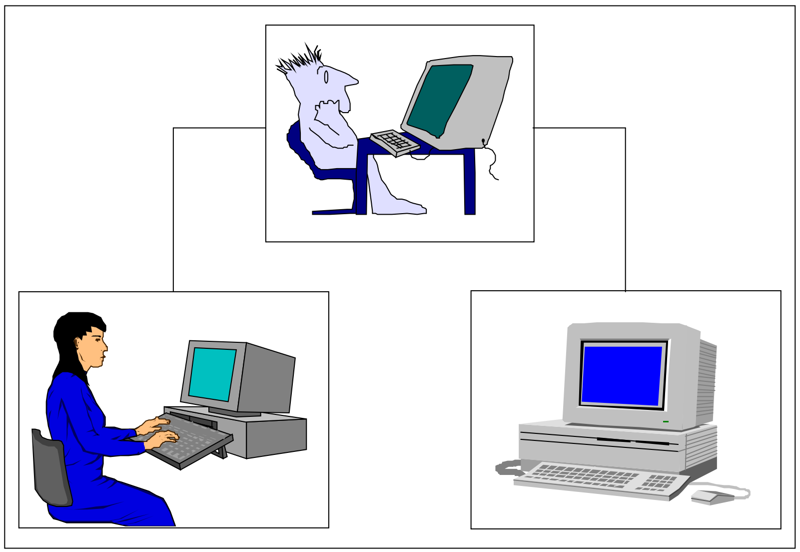
Turing Imitation Game: Phase 1



Turing did not provide definitions of machines and thinking, he just avoided semantic arguments by inventing a game, the ***Turing Imitation Game***.

The imitation game originally included two phases. In the first phase, the interrogator, a man and a woman are each placed in separate rooms. The interrogator's objective is to work out who is the man and who is the woman by questioning them. The man should attempt to deceive the interrogator that *he* is the woman, while the woman must convince the interrogator that *she* is the woman.

Turing Imitation Game: Phase 2



In the second phase of the game, the man is replaced by a computer programmed to deceive the interrogator as the man did. It would even be programmed to make mistakes and provide fuzzy answers in the way a human would. If the computer can fool the interrogator as often as the man did, we may say this computer has passed the intelligent behavior test.

The Turing test has two remarkable qualities that make it universal.

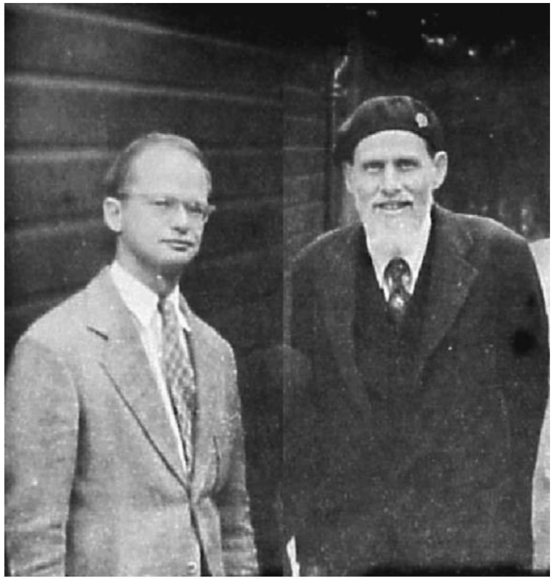
- By maintaining communication between the human and the machine via terminals, the test gives us an objective standard view on intelligence.
- The test itself is quite independent from the details of the experiment. It can be conducted as a two-phase game, or even as a single-phase game when the interrogator needs to choose between the human and the machine from the beginning of the test.

- Turing believed that by the end of the 20th century it would be possible to program a digital computer to play the imitation game. Although modern computers still cannot pass the Turing test, it provides a basis for the verification and validation of knowledge-based systems.
- **A program thought intelligent in some narrow area of expertise is evaluated by comparing its performance with the performance of a human expert.**
- To build an intelligent computer system, we have to capture, organize and use human expert knowledge in some narrow area of expertise.

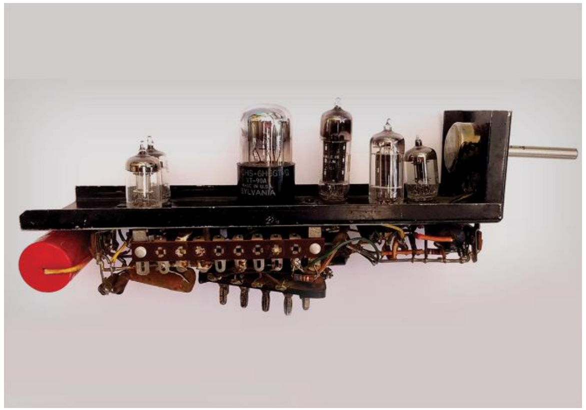
3. The history of artificial intelligence

The birth of artificial intelligence (1943 – 1956)

- The first work recognized in the field of AI was presented by **Warren McCulloch** and **Walter Pitts** in 1943. They proposed a model of an artificial neural network and demonstrated that simple network structures could learn.
- McCulloch, the second "founding father" of AI after Alan Turing, had created the corner stone of neural computing and artificial neural networks (ANN).

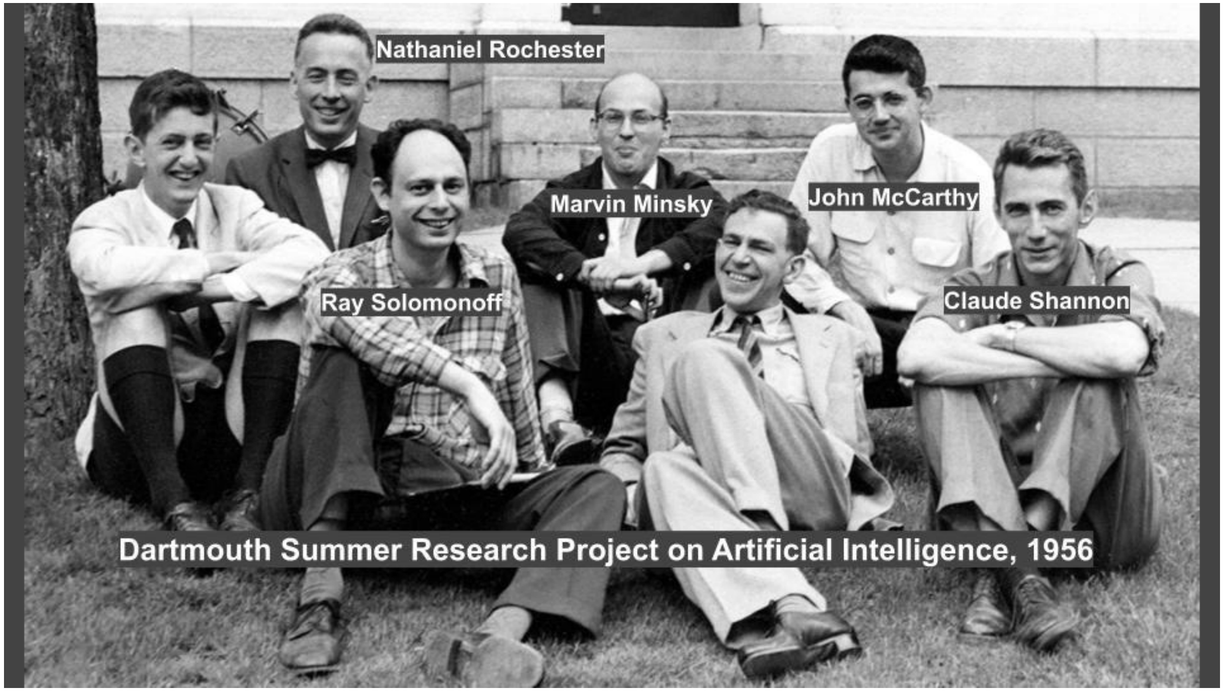


- The third founder of AI was **John von Neumann**, the brilliant Hungarian-born mathematician. In 1930, he joined the Princeton University, lecturing in mathematical physics. He was an adviser for the Electronic Numerical Integrator and Calculator project at the University of Pennsylvania and helped to design the **Electronic Discrete Variable Calculator**. He was influenced by McCulloch and Pitts's neural network model. When **Marvin Minsky** and **Dean Edmonds**, two graduate students in the Princeton mathematics department, built the first neural network computer in 1951, von Neumann encouraged and supported them.



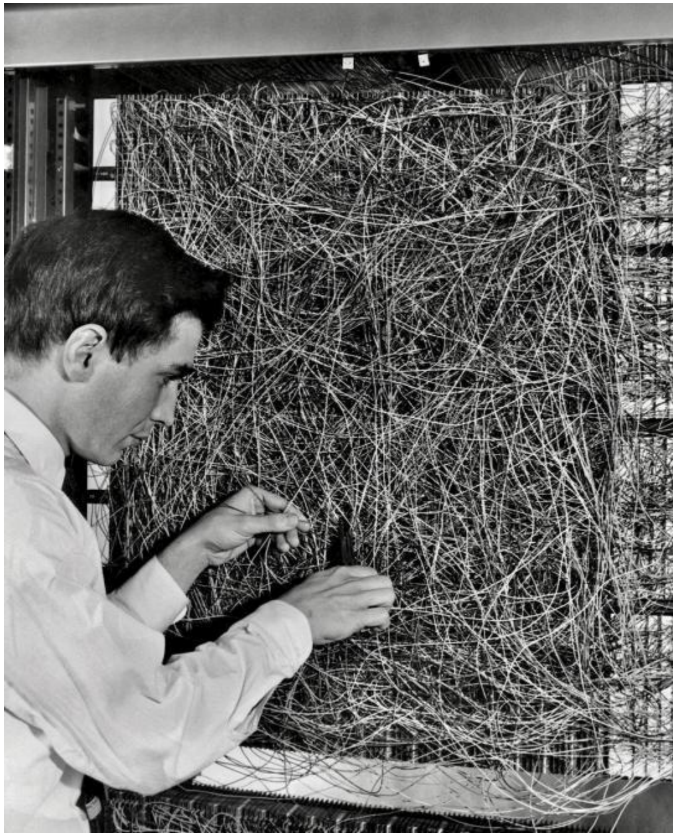
- Another of the first-generation researchers was **Claude Shannon**. He graduated from MIT and joined Bell Telephone Laboratories in 1941. Shannon shared Alan Turing's ideas on the possibility of machine intelligence. In 1950, he published a paper on chess-playing machines, which pointed out that a typical chess game involved about 10^{120} possible moves (Shannon, 1950). Even if the new von Neumann-type computer could examine one move per microsecond, it would take 3 ´ 10^{106} years to make its first move. Thus Shannon demonstrated the need to use heuristics in the search for the solution.

- In 1956, **John McCarthy**, **Martin Minsky** and **Claude Shannon** organized a summer workshop at Dartmouth College. They brought together researchers interested in the study of machine intelligence, artificial neural nets and automata theory. Although there were just ten researchers, this workshop gave birth to a new science called ***artificial intelligence***.



The rise of artificial intelligence, or the era of great expectations (1956 – late 1960s)

- The early works on neural computing and artificial neural networks started by McCulloch and Pitts was continued. Learning methods were improved, and **Frank Rosenblatt** proved the ***perceptron convergence theorem***, demonstrating that his learning algorithm could adjust the connection strengths of a perceptron.



- One of the most ambitious projects of the era of great expectations was the **General Problem Solver (GPS)**. **Allen Newell** and **Herbert Simon** from the Carnegie Mellon University developed a general-purpose program to simulate human-solving methods.
- Newell and Simon postulated that a problem to be solved could be defined in terms of ***states***. They used the mean-end analysis to determine a difference between the current and desirable or *goal state* of the problem, and to choose and apply *operators* to reach the goal state. The set of operators determined the solution plan.

- However, GPS failed to solve complex problems. The program was based on formal logic and could generate an infinite number of possible operators. The amount of computer time and memory that GPS required to solve real-world problems led to the project being abandoned.
- In the sixties, AI researchers attempted to simulate the thinking process by inventing *general methods* for solving *broad classes of problems*. They used the general-purpose search mechanism to find a solution to the problem. Such approaches, now referred to as ***weak methods***, applied weak information about the problem domain.

- By 1970, the euphoria about AI was gone, and most government funding for AI projects was cancelled. AI was still a relatively new field, academic in nature, with few practical applications apart from playing games. So, to the outsider, the achieved results would be seen as toys, as no AI system at that time could manage real-world problems.

Unfulfilled promises, or the impact of reality (late 1960s – early 1970s)

The main difficulties for AI in the late 1960s were:

- Because AI researchers were developing general methods for broad classes of problems, early programs contained little or even no knowledge about a problem domain. To solve problems, programs applied a search strategy by trying out different combinations of small steps, until the right one was found. This approach was quite feasible for simple **toy problems**, so it seemed reasonable that, if the programs could be "scaled up" to solve large problems, they would finally succeed.

- Many of the problems that AI attempted to solve were **too broad and too difficult**. A typical task for early AI was machine translation. For example, the National Research Council, USA, funded the translation of Russian scientific papers after the launch of the first artificial satellite (Sputnik) in 1957. Initially, the project team tried simply replacing Russian words with English, using an electronic dictionary. However, it was soon found that translation requires a general understanding of the subject to choose the correct words. This task was too difficult. In 1966, all translation projects funded by the US government were cancelled.

- In 1971, the British government also suspended support for AI research. Sir **James Lighthill** had been commissioned by the Science Research Council of Great Britain to review the current state of AI. He did not find any major or even significant results from AI research, and therefore saw no need to have a separate science called “artificial intelligence”.
- https://www.youtube.com/watch?v=03p2CADwGF8

The technology of expert systems, or the key to success (early 1970s – mid-1980s)

- Probably the most important development in the seventies was the realization that the domain for intelligent machines had to be sufficiently restricted. Previously, AI researchers had believed that clever search algorithms and reasoning techniques could be invented to emulate general, human-like, problem-solving methods. A general-purpose search mechanism could rely on elementary reasoning steps to find complete solutions and could use weak knowledge about domain.

  When weak methods failed, researchers finally realized that the only way to deliver practical results was to **solve typical cases in narrow areas** of expertise, making large reasoning steps.

DENDRAL

- DENDRAL was developed at Stanford University to determine the molecular structure of Martian soil, based on the mass spectral data provided by a mass spectrometer. The project was supported by NASA. Edward Feigenbaum, Bruce Buchanan (a computer scientist) and Joshua Lederberg (a Nobel prize winner in genetics) formed a team.
- There was no scientific algorithm for mapping the mass spectrum into its molecular structure. Feigenbaum's job was to incorporate the expertise of Lederberg into a computer program to make it perform at a human expert level. Such programs were later called ***expert systems***.

- DENDRAL marked a major “paradigm shift” in AI: a shift from general-purpose, knowledge-sparse weak methods to domain-specific, knowledge-intensive techniques.
- The aim of the project was to develop a computer program to attain the level of performance of an experienced human chemist. Using heuristics in the form of high-quality specific rules, rules-of-thumb, the DENDRAL team proved that computers could equal an expert in narrow, well defined, problem areas.
- The DENDRAL project originated the fundamental idea of expert systems – ***knowledge engineering***, which encompassed techniques of capturing, analyzing and expressing in rules an expert’s “know-how”.

- A 1986 survey reported a remarkable number of successful expert system applications in different areas: chemistry, electronics, engineering, geology, management, medicine, process control and military science (Waterman, 1986). Although Waterman found nearly 200 expert systems, most of the applications were in the field of medical diagnosis. Seven years later a similar survey reported over 2500 developed expert systems (Durkin, 1994). The new growing area was business and manufacturing, which accounted for about 60% of the applications. Expert system technology had clearly matured.

However:

- Expert systems are restricted to a **very narrow domain** of expertise. For example, MYCIN, which was developed for the diagnosis of infectious blood diseases, lacks any real knowledge of human physiology. If a patient has more than one disease, we cannot rely on MYCIN. In fact, therapy prescribed for the blood disease might even be harmful because of the other disease.
- Expert systems can show the sequence of the rules they applied to reach a solution, but cannot relate accumulated, heuristic knowledge to any **deeper understanding** of the problem domain.

- Expert systems have difficulty in **recognizing domain boundaries**. When given a task different from the typical problems, an expert system might attempt to solve it and fail in rather unpredictable ways.
- Heuristic rules represent knowledge in abstract form and lack even basic understanding of the domain area. It makes the task of **identifying incorrect, incomplete or inconsistent knowledge** difficult.
- Expert systems, especially the first generation, have **little or no ability to learn** from their experience. Expert systems are built individually and cannot be developed fast. Complex systems can take over 30 person-years to build.

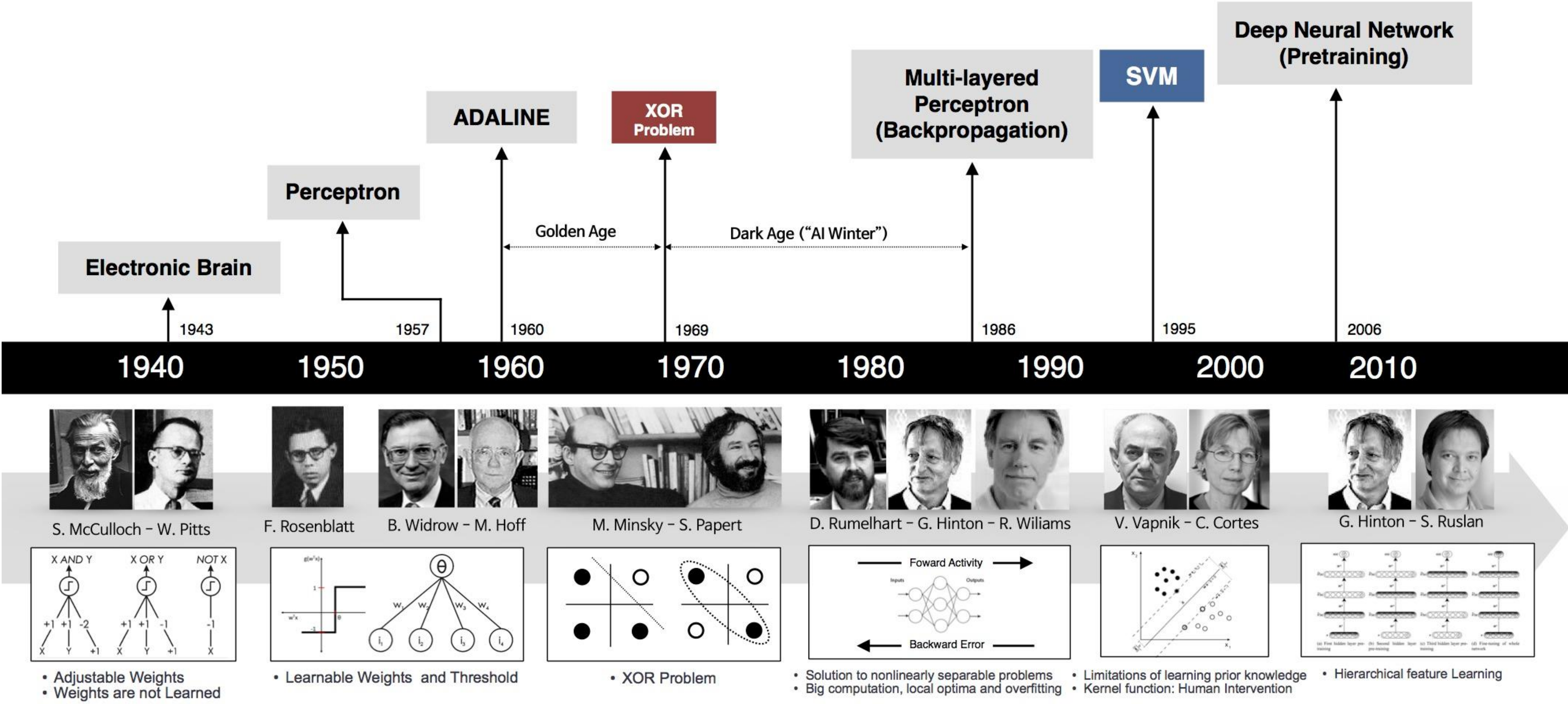
How to make a machine learn, or the rebirth of neural networks (mid-1980s – onwards)

- In the mid-eighties, researchers, engineers and experts found that building an expert system required much more than just buying a reasoning system or expert system shell and putting enough rules in it. Disillusions about the applicability of expert system technology even led to people predicting an *AI "winter"* with severely squeezed funding for AI projects. AI researchers decided to have a new look at **neural networks**.

- By the late sixties, most of the basic ideas and concepts necessary for neural computing had already been formulated. However, only in the mid-eighties did the solution emerge. The major reason for the delay was technological: there were no PCs or powerful workstations to model and experiment with artificial neural networks.
- In the eighties, because of the need for brain-like information processing, as well as the advances in computer technology and progress in neuroscience, the field of neural networks experienced a dramatic resurgence. Major contributions to both theory and design were made on several fronts.

- Grossberg established a new principle of self-organisation (***adaptive resonance theory***), which provided the basis for a new class of neural networks (Grossberg, 1980).
- Hopfield introduced neural networks with feedback – ***Hopfield networks***, which attracted much attention in the eighties (Hopfield, 1982).
- Kohonen published a paper on ***self-organising maps*** (Kohonen, 1982).
- Barto, Sutton and Anderson published their work on ***reinforcement learning*** and its application in control (Barto et al., 1983).

Neural networks to Deep Learning



A BRIEF HISTORY OF DEEP LEARNING

1958
Cornell psychologist Frank Rosenblatt unveils the Perceptron, a single-layer neural network on a room-size computer. →

1969
AI giant Marvin Minsky of MIT cowrites a book casting doubt on the viability of neural networks. They fall out of favor. →

1986
Neural nets pioneer Geoffrey Hinton and others find a way to train multilayer neural networks to correct mistakes. A flurry of activity ensues.

1989
French researcher Yann LeCun, then at Bell Labs, begins foundational work on a type of neural net that becomes crucial for image recognition.

FREDERIC LEWIS—ARCHIVE PHOTOS/GETTY IMAGES, LEFT: ANN E. YOW-DYSON—GETTY IMAGES

1991
German researchers Sepp Hochreiter and Jürgen Schmidhuber pioneer a neural net with memory features, which eventually proves superior for natural-language processing.

1997
IBM's Deep Blue beats **world champion Garry Kasparov** (right) in chess using traditional AI techniques. →

STAN HONDA—AFP/GETTY IMAGES

Mid-1990s
Neural nets fall into disfavor again, eclipsed by other machine-learning techniques.

2007
Fei-Fei Li founds ImageNet and begins assembling a database of 14 million labeled images that can be used for machine-learning research. →

2011
Microsoft introduces neural nets into its speech-recognition features.

2011
IBM's Watson beats two champions at Jeopardy using traditional AI techniques.

CARLOS CHAVARRIA—THE NEW YORK TIMES/REDUX PICTURES

2012
JUNE
Google Brain publishes the "cat experiment." A neural net, shown 10 million unlabeled YouTube images, has trained itself to recognize cats. ←

AUGUST
Google introduces neural nets into its speech-recognition features.

OCTOBER
A neural net designed by two of Hinton's students wins the annual ImageNet contest by a wide margin.

2013
MAY
Google improves photo search using neural nets.

JIM WILSON—THE NEW YORK TIMES/REDUX PICTURES

2014
JANUARY
Google acquires DeepMind, a startup specializing in combining deep learning and reinforcement learning, for $600 million.

2015
DECEMBER
A team from Microsoft, using neural nets, outperforms a human on the ImageNet challenge.

2016
MARCH
DeepMind's AlphaGo, using deep learning, defeats world champion **Lee Sedol** in the Chinese game of go, four games to one. →

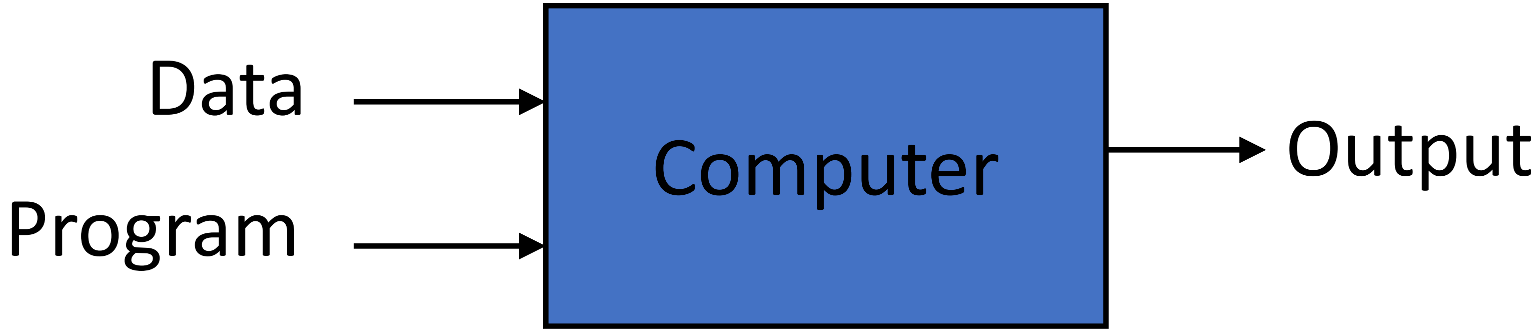
LEE JIN-MAN—AP PHOTO

Source: - fortune.com

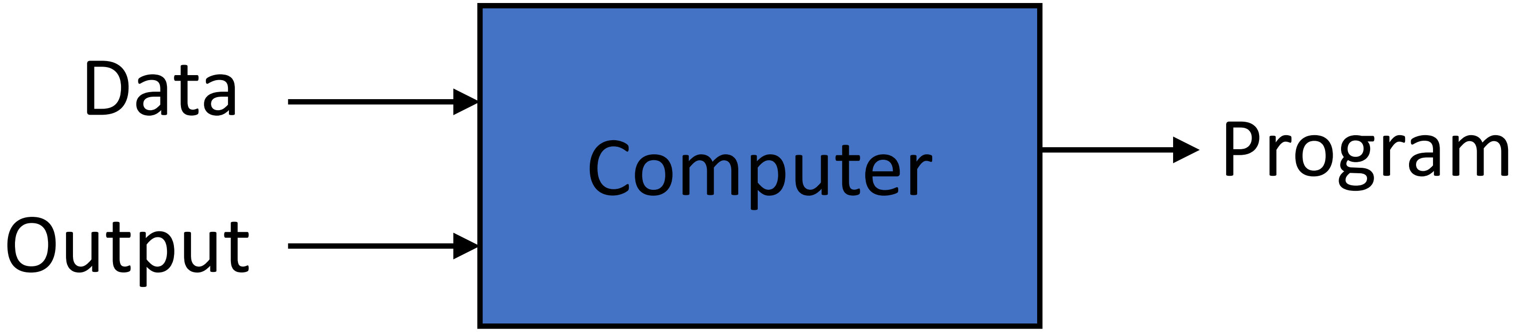
4. So What Is Machine Learning?

- Getting computers to program themselves
- Writing software is the bottleneck
- Let the data do the work instead!

Traditional Programming



Machine Learning



Magic?

No, more like gardening

- **Seeds** = Algorithms
- **Nutrients** = Data
- **Gardener** = You
- **Plants** = Programs

Sample Applications

- Web search
- Computational biology
- Finance
- E-commerce
- Space exploration
- Robotics
- Information extraction
- Social networks
- Debugging
- [Your favorite area]

ML in a Nutshell

- Tens of thousands of machine learning algorithms
- Hundreds new every year
- Every machine learning algorithm has three components:
  - **Representation**
  - **Evaluation**
  - **Optimization**

Representation

- Decision trees
- Sets of rules / Logic programs
- Instances
- Graphical models (Bayes/Markov nets)
- Neural networks
- Support vector machines
- Model ensembles
- Etc.

Evaluation

- Accuracy
- Precision and recall
- Squared error
- Likelihood
- Posterior probability
- Cost / Utility
- Margin
- Entropy
- K-L divergence
- Etc.

Optimization

- Combinatorial optimization
  - E.g.: Greedy search
- Convex optimization
  - E.g.: Gradient descent
- Constrained optimization
  - E.g.: Linear programming

Types of Learning

- **Supervised (inductive) learning**
  - Training data includes desired outputs
- **Unsupervised learning**
  - Training data does not include desired outputs
- **Semi-supervised learning**
  - Training data includes a few desired outputs
- **Reinforcement learning**
  - Rewards from sequence of actions