Module 1: A Historical Overview of Artificial Intelligence and Machine Learning

M. Vidyasagar

Distinguished Professor, IIT Hyderabad
Email: m.vidyasagar@iith.ac.in
Website: www.iith.ac.in/∼m_vidyasagar/

IIT Hyderabad

Outline

IIT Hyderabad

Outline

What is Artificial Intelligence?

My Definition: Artificial Intelligence (AI) refers to the technology of enabling computers to perform tasks that had previously been difficult for machines, though quite easy for humans.

This excludes "pure" numerical computations, but includes tasks such as image and speech recognition, game playing, and heuristic (as opposed to strictly logical) reasoning.

AI: Why the Excitement?

AI as a separate discipline has been around since 1950. So why is everyone so excited now?

Some recent advances:

- Image Processing
- Speech Processing
- Natural Language Processing
- Game Playing

Brief overview of each topic is given next.

Advances in Image Processing

- ImageNet: Database of 14 million "hand-curated" images; used as a testbed to benchmark image recognition algorithms.
- Until 2012, best error rates were around 26%.
- In 2012, a "quantum reduction" in error rate to 16% using convolutional deep neural networks (paper by R. Salakhutdinov, G. Hinton, A. Krizhevsky and I. Sutskever). ▸ Link
- Many advances since then (Top 5 Accuracy $\approx 98\%$). ▸ Link

IIT Hyderabad

Advances in Speech and Natural Language Processing

- Interactive Voice Recognition (IVR) systems
  - Restricted vocabulary, structured environment
  - Getting better, but still can't handle unstructured queries
- On-line translation between languages
  - Cannot capture "mood"
  - OK for business contracts, not OK for poetry.
  - Can still produce hilarious results

Advances in Game Playing – 1: Chess

In 1996 Garry Kasporov beat "Deep Blue" built by IBM. In 1997 Kasporov lost a rematch to a bigger computer. ▸ Link

▸ Video (Poor Quality)

Is this an advance in AI – or an advance in sheer computation?

Deep Blue-2 used the *same heuristics* as Deep Blue-1 but had more CPU power.

My view: This was *not* an advance in AI!

But recently there *have* been significant advances.

Advances in Game-Playing – Jeopardy

- IBM's Watson: To play "Jeopardy" ▸ Video
- Watson combined speech recognition, search, and speech generation.
- It competed against humans without any handicap, and defeated the champions.

Advances in Game-Playing – Chess and Go

- Deep Mind (since acquired by Google) Alpha-Go: To play the game of "Go".
- AlphaGo defeated Lee Sidol in 2015 and the top-ranked player Ke Jie in 2017. ▸ Link
- Deep Mind: AlphaZero to play chess, go and Shogi. ▸ Link
- AlphaZero *does not use prior knowledge!*

Success of AlphaZero challenges the widely-held belief that prior knowledge is beneficial.

IIT Hyderabad

What is Behind Current Successes?

- Availability of *massive* amounts of training data.
- Advances in computation, special-purpose hardware such as GPUs and Google's TPUs (Tensor Processing Units).
- Advances in algorithms for machine learning.
- AI is now "democratized" – a lot of open source software is available (e.g., Tensor Flow).

Clouds on the Horizon

All is not what it is claimed to be!

- Results in image processing are rather fragile – slight changes in images lead to massive degradation in performance
- Many results are also not reproducible.
- IBM's Watson was touted for personalized medicine and other medical applications
- It sometimes gave incorrect advice.
- Google tried to commercialize AlphaGo's "framework" as a universal problem solver.
- Neither Watson nor AlphaGo has been a commercial success.

Current Challenges in AI

- Current use of massive computational resources shuts out all but a few enormous-sized companies from doing research in AI.
- This is especially true of "deep learning" research.
- The research community is beginning to recognize that the "mathematical foundations of deep learning" are not well understood.
- In contrast, the mathematical foundations of "shallow learning" are quite well-understood.
- Main challenge now is to develop a suitable theory of deep learning at the scale at which it is practiced today.
- Watch Dr. Ali Rahimi's lecture slamming a lot of current research in learning as "alchemy." ▸ Video

IIT Hyderabad

Are There Precedents for the Excitement?

This is the third or fourth cycle for AI hype.

- Late 1950s, early 1960s: Perceptrons, invented by Frank Rosenblatt
- Late 1970s, early 1980s: Rule-based "expert systems"
- Mid 1980s, 1990s: (Artificial) Neural Networks (ANNs)
- Current cycle: Deep learning (DL), reinforcement learning

What can we learn from past failures as well as successes?

My View: Technologies with solid mathematical foundation have survived, others have not.

IIT Hyderabad

Outline

IIT Hyderabad

First Steps in Artificial Intelligence

- Alan Turing proposes the "Turing Machine" (1936) ▸ Link
- Birth of "modern" digital computer ENIAC: (1946)
- John von Neumann proposes the "von Neumann" reprogrammable computer (1945)
- Norbert Wiener writes "Cybernetics" (1948) ▸ Link
- Alan Turing proposes the "Turing test" of machine intelligence (1950) ▸ Link
- Claude Shannon proposes a chess-playing computer (1950) ▸ Link
- John McCarthy coins the phrase "Artificial Intelligence" (1955)

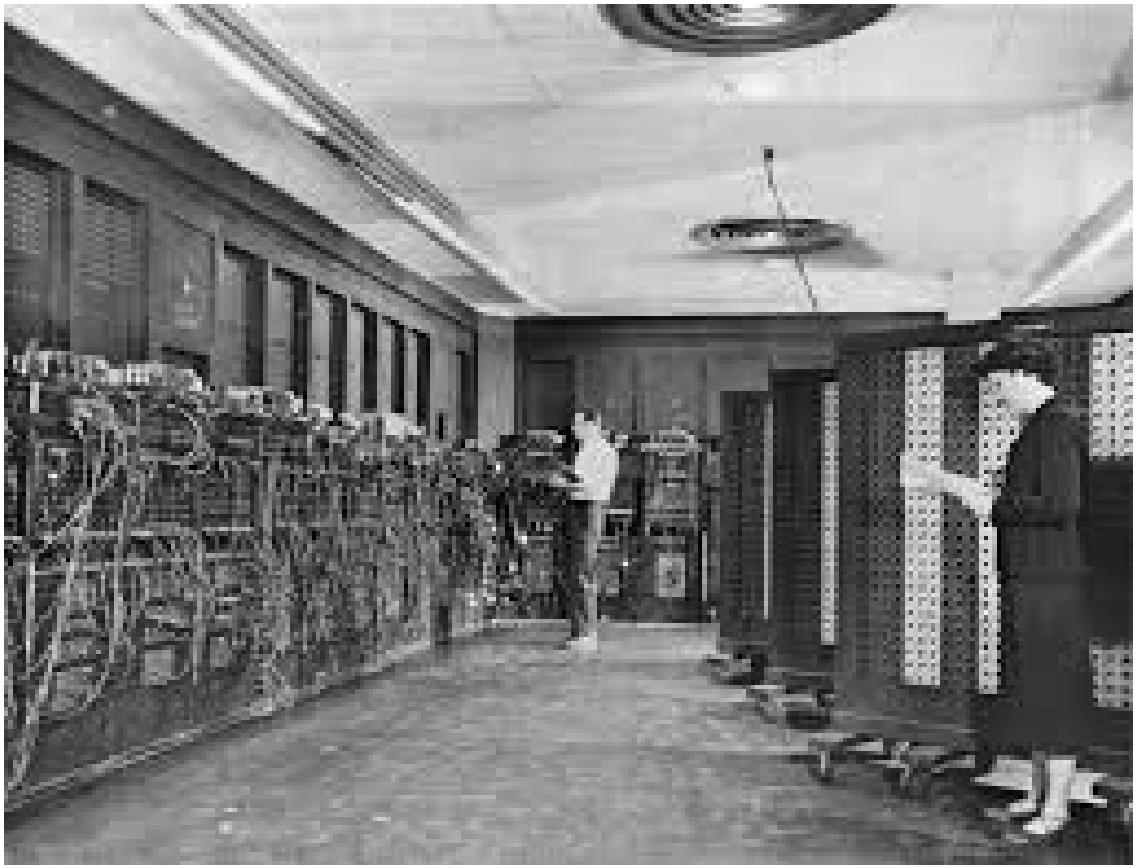
IIT Hyderabad

Alan Turing

The Turing Machine (1936)

- Turing proposed his universal computing framework to solve the "entscheidungsproblem" (the decision problem), in the context of proving the consistency of the axioms of arithmetic.
- Earlier Kurt Gödel showed that the Peano axioms of arithmetic were consistent but incomplete: There exist "true" statements that cannot be proven.
- Turing's approach was simpler and cleaner. (And his conclusions are slightly different.)
- He also introduced the "universal Turing machine" and "decidability" – and showed that the "Halting Problem" was undecidable.

John von Neumann and ENIAC



An ENIAC installation at Princeton.

Birth of Digital Computation

- "Modern" digital computation was invented as a part of the Manhattan project at Los Alamos.
- John von Neumann (among others) invented the "stored memory" architecture for digital computers, whereby the program to be executed was stored within the computer itself – no need to reconfigure the hardware in order to make execute a different program.

The Idea of Machine Intelligence

Turing's test of machine intelligence (1950): A machine is intelligent if it can fool a human being into thinking that he is interacting with another human, and not a machine, during an interactive session.

Searle's "Chinese Room Argument" (1980): Even if a computer can fool a set of native Chinese speakers by executing very fast operations (for example, translating text between English and Chinese), it still won't "understand" Chinese, nor have a "consciousness." He proposed "strong AI": Building a machine with understanding and consciousness.

My thoughts are more down to earth.

Functional vs. Operational Artificial Ingelligence

What were/are the aims of AI? "Mimicking human abilities" is too vague.

Distinguish between

- *Functional Similarity:* A machine replicates the functionalities of a human at an input-output level.
- *Operational Similarity:* A machine not only replicates the functionalities of a human but also achieves them in the same way that a humans would.

Functional vs. Operational Automation

To illustrate: A "functional" mechanical horse would just pull a tractor, while an "operational" one would have four legs. A "functional" airplane would fly, while an "operational" airplane would flap its wings.

Industrial revolution (automation) focused *entirely* on functional similarity.

Evolution of the Aims of Artificial Intelligence

Early AI focused a great deal on operational similarity.

- Computer vision
- Language processing
- Artificial neural networks
- Deep Blue architecture

Today's AI researchers do not pay much attention to operational AI, and strive to achieve functional AI.

The Turing Test Revisited

Turing's test of intelligence (1950) proposed that a machine can be said to be "intelligent" if a human could not distinguish it from another human. This would be "functional similarity" to a human.

The "Chinese room argument" of John Searle proposed that "merely" replicating human abilities would not be enough, because such a machine would lack consciousness etc.

Turing was arguing for functional similarity while Searle was arguing for operational similarity.

Outline

IIT Hyderabad

The Two Branches of AI

During its history, AI has bifurcated into two main branches:

- Reasoning, exemplified by rule-based expert systems
- Learning and generalization, exemplified by neural networks.

Quite distinct methodologies are involved in the two branches.

Much of current excitement arises from learning and generalization.

There are in turn two broad areas here: Deep Learning (DL) and Reinforcement Learning (RL).

Glimpses are given of each topic.

Rule-Based Expert Systems

Rule-based expert systems (1970s) attempted to capture the ability of humans to reason, and to replicate them in a computer.

"Knowlege engineers" interviewed human experts to distill their knowledge in the form of "If ... then ..." rules.

Example: If it rains, then I will open my umbrella.

Objective: To allow novices to operate at (or near) the level of experts.

Some Common Approaches

Early expert systems relied on two common approaches:

Forward Chaining: Starts from causes and proceeds towards outcomes. Useful for understanding consequences of various events.

Backward Chaining: Starts from outcomes and deciphers what could have caused them. Useful in fault diagnosis, medical diagnosis etc.

Belief Propagation: Another popular approach that allowed the incorporation of uncertainty (probability) and confidence factors.

Outline

What Are Learning and Generalization?

In "reasoning" systems, the output of the program is a truth value (or a confidence factor).

In "learning" systems, the output of the program is either binary (classification), or a real number (regression).

The objective is to "train" the program on a set of "labelled training" data. After that, it is "tested" by being asked to predict the correct label for previously unseen data.

"Generalization" quantifies the quality of the prediction:

- In a classification problem, the fraction of correct labels.
- in a regression problem, a least-squares measure.

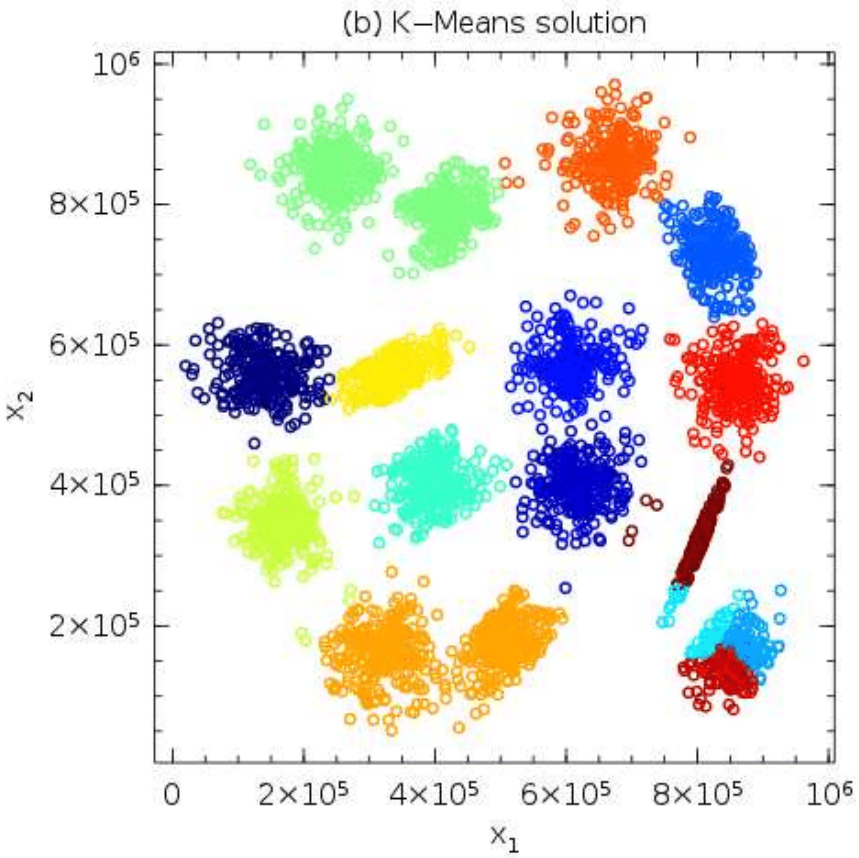
Supervised vs. Unsupervised Learning

The above description is for *supervised* learning. There is also *unsupervised* learning, which the statisticians call clustering.

Clustering Problem: Given a set of vectors, assign them to groups so that each vector is closer to the centroid of its own cluster than to the centroid of any other cluster. (Illustration on next slide.)

Then a new test input is assigned to the cluster whose centroid is closest to it.

Illustration of Clustering



K-Means Algorithm

In general, the clustering problem as stated is *NP-hard*, meaning that no efficient algorithms exist to solve the problem.

The K-means algorithm is a popular heuristic that "approximately" solves the above problem.

Also, how many clusters should be used?

This topic is not discussed further. Check the literature for Kohonen maps, self-organizing networks, etc

Outline

The Perceptron

A perceptron was a device with a binary output invented by Frank Rosenblatt in the 1950s.

It took a set of n real inputs $x_1, \ldots, x_n$, and put out either a -1 or 1 according to

$$y = \text{sign}\left(\sum_{i=1}^{n} w_i x_i - \theta\right) = \text{sign}(\mathbf{x}^\top \mathbf{w} - \theta),$$

where $w_1, \ldots, w_n$ were the weights and θ was the threshold.

Someone called Arthur Samuels built a perceptron that could play checkers (but very little is known).

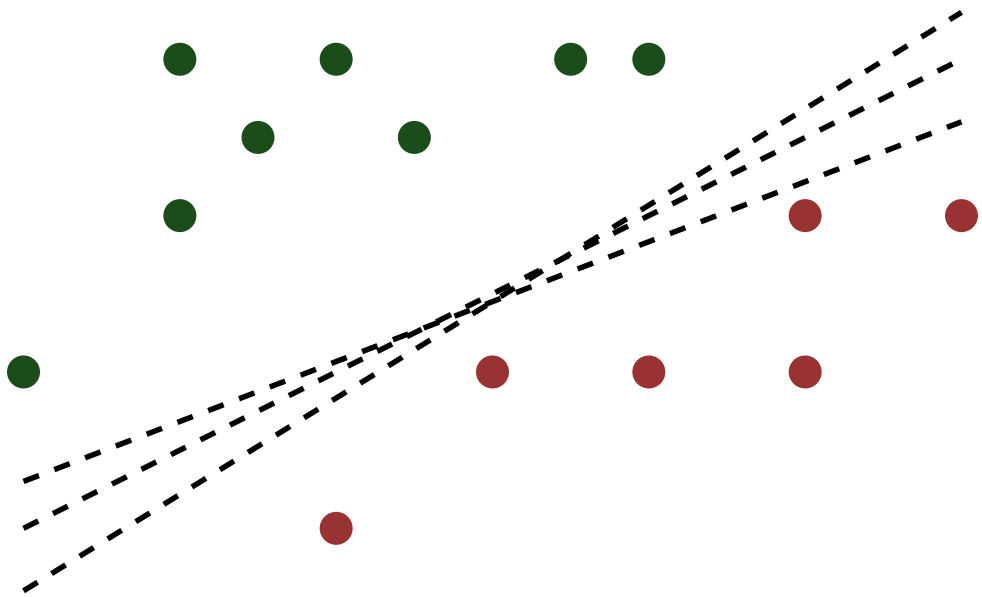
Training a Perceptron

Given a set of "labelled inputs" $\{\mathbf{x}_j, y_j)\}, j = 1, \ldots, m$, where $\mathbf{x}_j \in \mathbb{R}^n$ and $y_j \in \{-1, 1\}$, the "training" of the perceptron consists of identifying a weight vector $\mathbf{w} \in \mathbb{R}^n$ and a threshold $\theta \in \mathbb{R}$ such that

$$\text{sign}(\mathbf{x}_i^\top \mathbf{w} - \theta) = y_i, i = 1, \ldots, m.$$

If such a weight and threshold exist, the data is said to be **linearly separable**. (Illustration on next slide.)

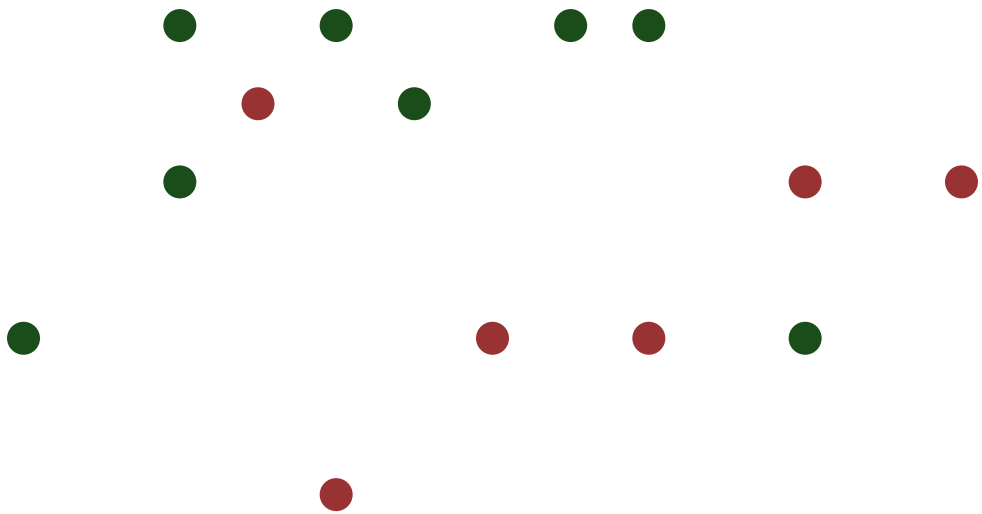
Depiction of a Linearly Separable Data Set



If there exists *one* straight line that separates the two sets, there exist *infinitely many* straight lines!

Which one is "the best"? This leads to Support Vector Machines (SVMs).

Depiction of a Not-Linearly Separable Data Set



No straight line can separate the green and maroon circles!

Can we find a straight line that misclassifies the *fewest number* of points? That problem is also NP-hard, but approximations exist.

Nonlinear Discriminant Functions

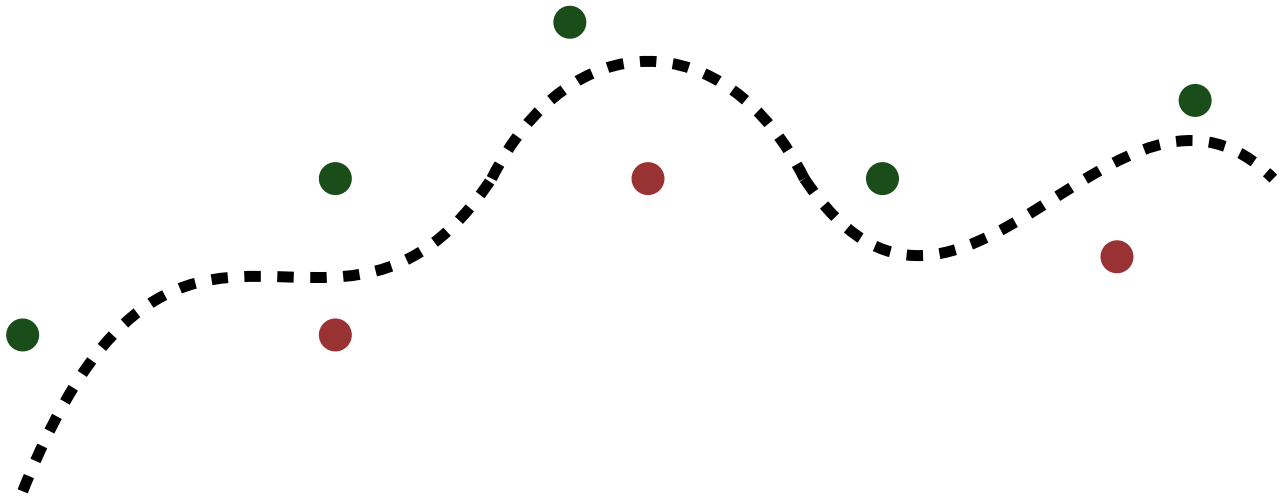
The perceptron is a *linear classifier* because $y = \text{sign}(\Delta(x))$ where the "discriminant function"

$$\Delta(x) = \sum_{i=1}^{n} w_i x_i - \theta$$

is linear in the components of the vector $\mathbf{x}$.

When a data set is not linearly separable, we can use a nonlinear discriminant function, which leads to "kernel-based" classifiers. (Illustration in next slide.)

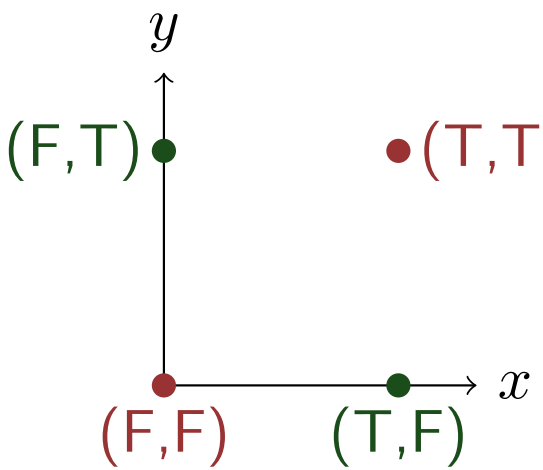
Nonlinear Discriminant Functions Illustrated



No straight line can separate the two sets of circles.

The XOR Counter-Example

In 1969 Minsky and Papert demonstrated that a perceptron could not solve the XOR (exclusive or) problem. No straight line separates the blue dots from the red. So no perceptron can solve the XOR problem.



This finding put perceptrons out of favor for about two decades.

Outline

Multi-Layer Perceptron Networks (MLPNs)

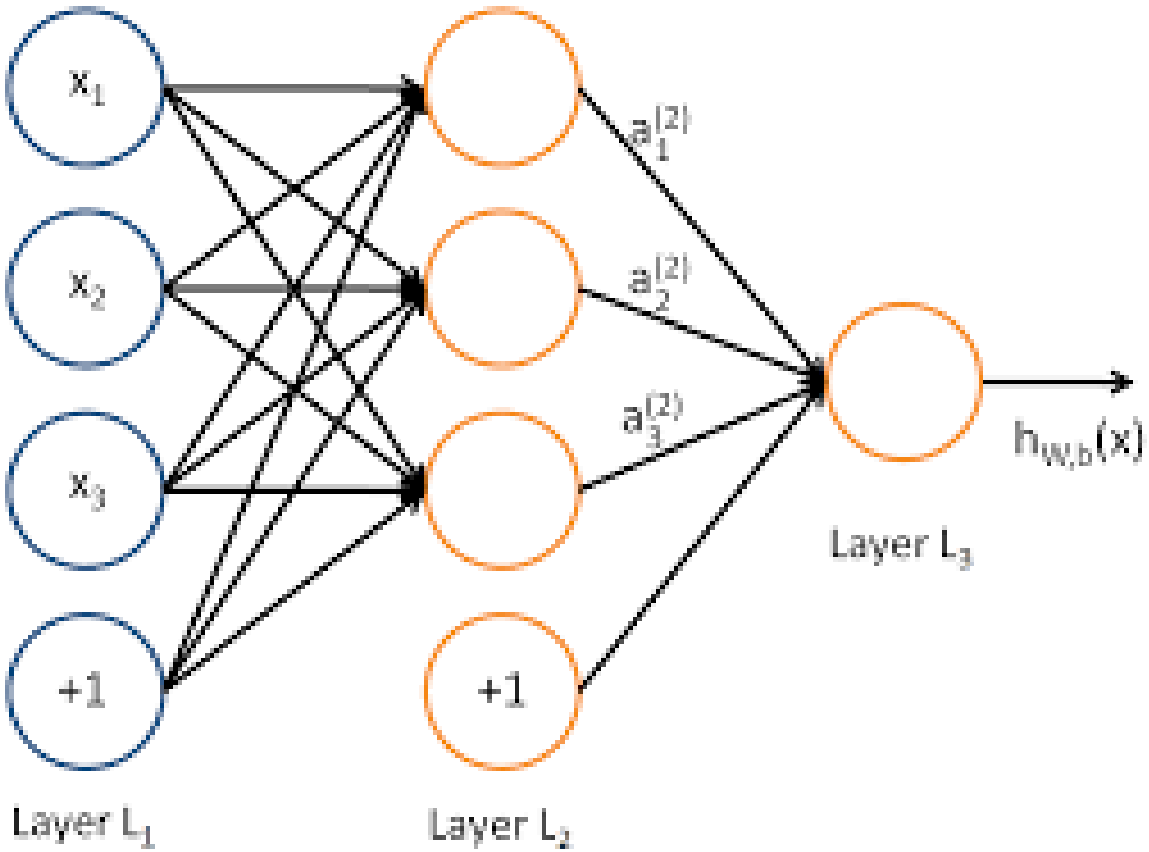
In 1986 neural networks made a spectacular comeback with the invention of MLPNs, essentially perceptrons arranged at multiple "layers." Key paper by Rumelhart, Hinton and Williams. ▸ Link

Advantages of MLPNs:

- A training algorithm, popularly called back-propagation.
  - It is a rediscovery of the "adjoint network" introduced in circuit theory by Director and Rohrer.
- Universal approximation property: *Any* nonlinear function could be approximated arbitrarily closely by a sufficiently rich MLPN.
- A rich theory that explained *why* MLPNs worked. See MV, *Learning and Generalization*, Springer-Verlag, 2003.

IIT Hyderabad

Depiction of a MLPN

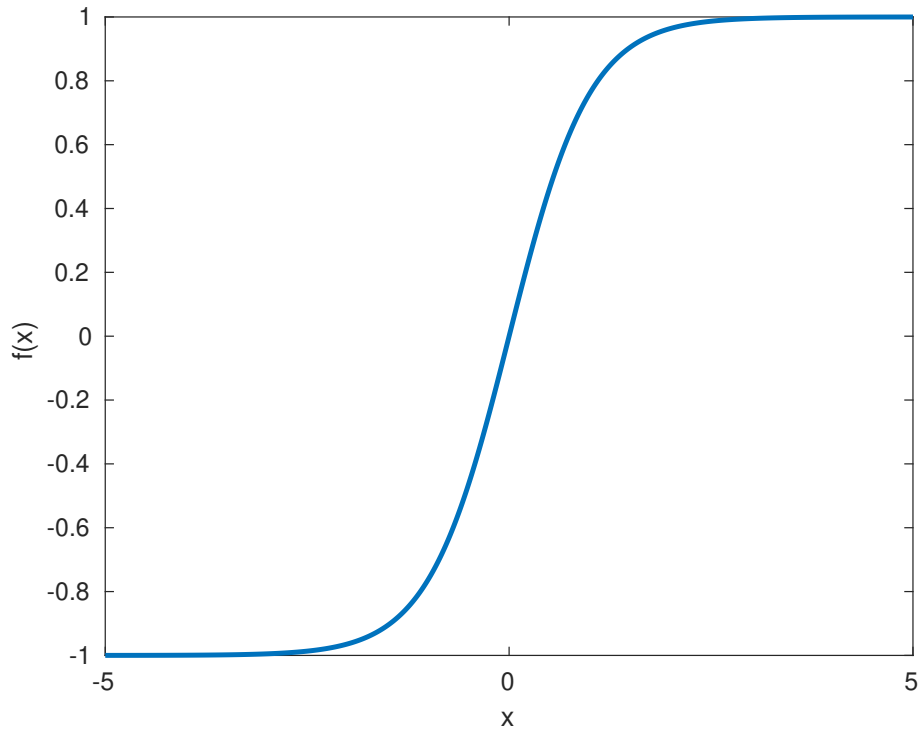


Artificial Neural Networks?

- During the early days, researchers tried to model the human brain using "artificial" neural networks (ANNs).
- Early ANNs used the sigmoidal I/O characteristic, motivated by the human brain. (Graph on next slide.)
- As time went on, the prefix "artificial" was dropped and now people just speak of NNs.
- Other alternatives to the sigmoid were explored, such as ReLU (Rectified Linear Unit).
- Some successes were achieved. But success was limited by non-availability of training data, and CPU speed. The advent of GPUs changed all that.

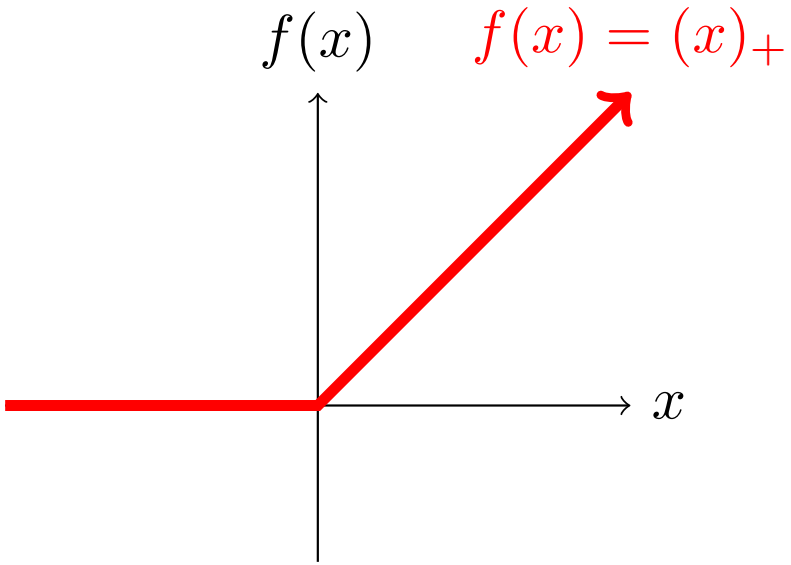
IIT Hyderabad

Sigmoidal Neuron Characteristic



$$f(x) = \tanh x.$$

Rectified Linear Unit



Both the sigmoid and the ReLU (and lots of other functions) have the "universal approximation property": *Any* I/O characteristic can be approximated by a combination of these units.

Training vs. Testing

A NN is "trained" by adjusting its weights and thresholds on a large amount of training data.

Then it is tested on previously unseen "testing" data.

The performance on training data is less important than the performance on testing data.

When there are too many adjustable parameters and too little data, the NN suffers from "over-fitting":

- Excellent performance on training data
- Poor generalization ability on previously unseen data.

Outline

Deep Learning Neural Networks

- Traditional NNs had a few layers (two, three, four ...) and thousands of adjustable parameters. Deep NNs have hundreds of layers and millions of adjustable parameters.
- Unless training data sets were in the high millions or larger, such NNs could suffer from "over-fitting":
- Problems such natural language or speech processing are tailor-made for deep learning.

Current Research in Deep Learning

Standard statistical learning theory tells us that when the number of adjustable parameters is too large compared to the amount of training data available, over-fitting would result.

Interestingly, deep neural networks don't seem to suffer from over-fitting.

Understanding why this is so is an active area of research.

Reinforcement Learning

The topic of reinforcement learning (RL) is quite mathematically advanced.

RL builds on Markov Decision Processes (MDP), where the optimal decision policy is obtained by solving a dynamic programming problem.

In MDP the underlying model is known; in RL it has to be inferred.

Questions?

Questions?