SysEng 5112 /EE 5370
Introduction to Neural Networks and Applications

Week 1: Introduction

Cihan H Dagli, PhD

Professor of Engineering Management and Systems Engineering
Professor of Electrical and Computer Engineering
Founder and Director of Systems Engineering Graduate Program

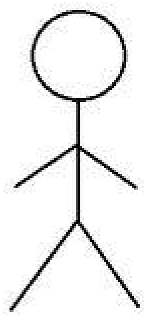
dagli@mst..edu

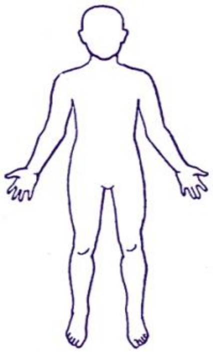
MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY
Rolla, Missouri, U.S.A.

SYSTEMS ENGINEERING
Research Center
Department of Defense UARC

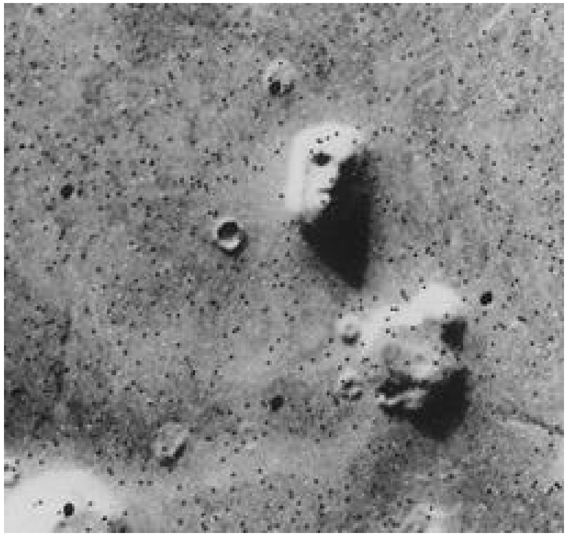
Smart Engineering Systems Lab
Engineering Management and Systems Engineering Department

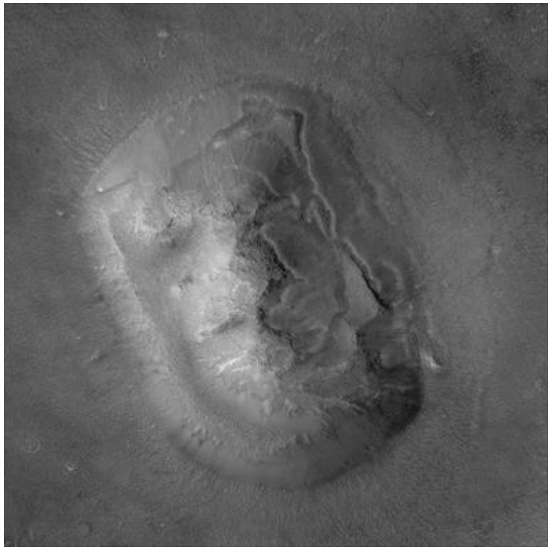
Motivating Example





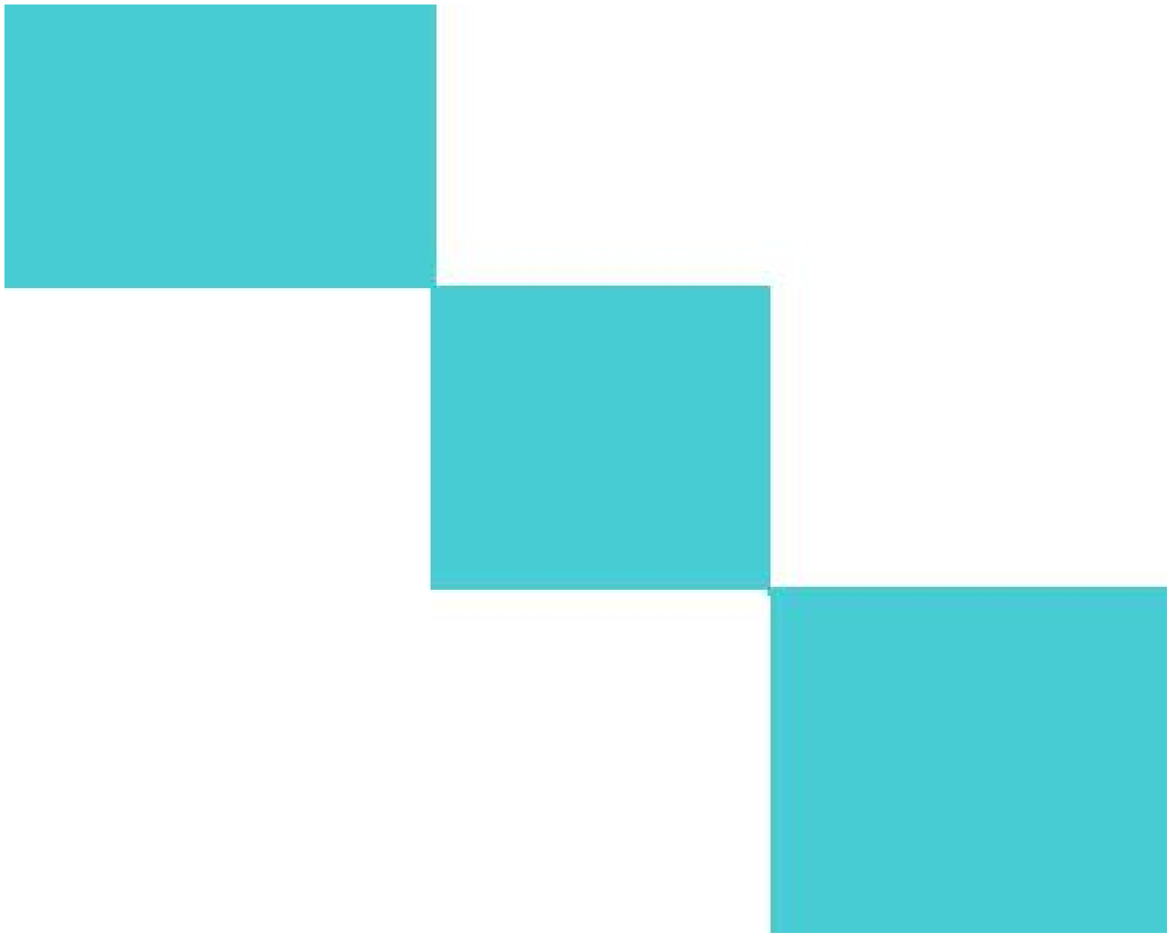
Matching patterns



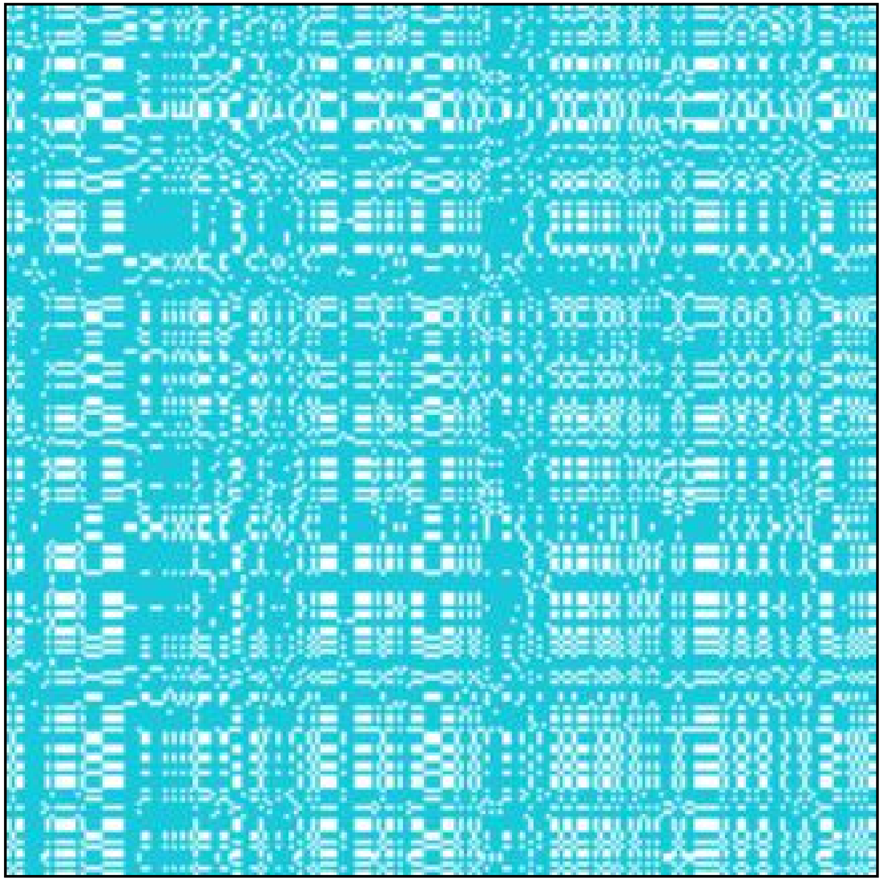


The Face on Mars
Images courtesy NASA JPL

Motivating Example



Motivating Example

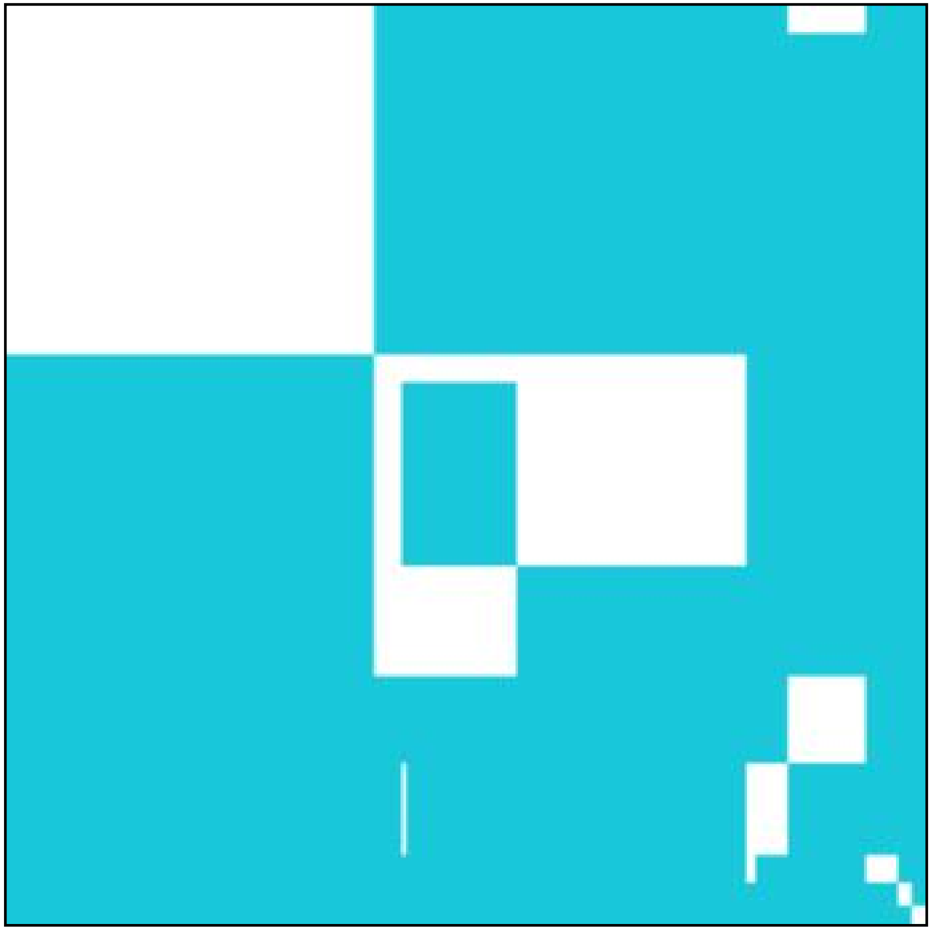


Can you generate the previous picture?

Systems Engineering Research Center
Department of Defense UARC

Missouri S&T
Smart Engineering Systems Lab
Engineering Management and Systems Engineering Department

Motivating Example



If not, can you generate this?

Our Brains and Pattern Matching

An average 3yr old's brain has over a quadrillion (1000 trillion) connections!

- How the words are stored in our brain?
- How do we store and create sound?
- How words are associated with sounds?
- How do we make sounds?
- How do we learn to write letters ?
- Why do we have accents while we speak?
- How do we remember things?
- Why your mom's food is the best?
- What happens when you have epilepsy?
- Why do we need meta facts or examples in explaining abstract terms?
- How do we create associations or mappings?
- How does Alzheimer's disease end life?

Animal Echolocation

- Bats and Whales use the sound of their calls for navigation and hunting.
- This *biosonar* has only one emitter and two receivers.
- Inputs to the animal's brain
  - the time lag in the reception of the echo by each ear
  - the difference in sound intensity received at each ear
- Outputs of the neural computations
  - The distance of the target, its relative velocity, its azimuth and elevation
  - The identity of the target and its features

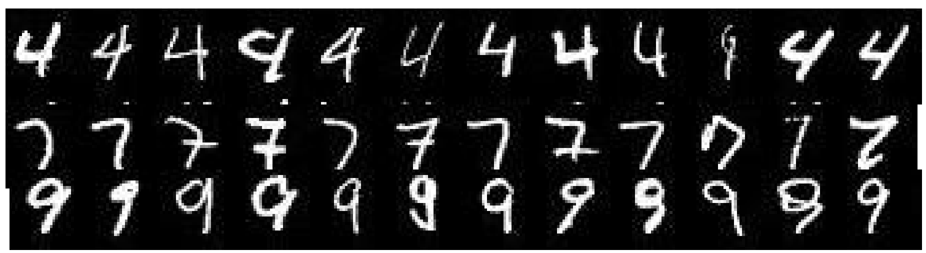
All this from a brain the size of a plum!

What is Machine Learning?

- **Machine learning processes seek to learn and adapt based on past experiences, just like our brains do.**
  - **The goal is to mimic the human brain's ability to *generalize*.**
  - **Develop "Smart" systems**
- **Some problems are very hard to encode**
  - **For example: Handwriting recognition**
  - **We can use lots of handwriting samples and correlate them with the actual text**
  - **The machine learning algorithm learns from the samples to generate a program for recognizing handwritings**

Deciphering Handwriting is Hard

- The USPS Handwritten Digit Data[1] were gathered at the Center of Excellence in Document Analysis and Recognition (CEDAR) at SUNY Buffalo, as part of a project sponsored by the US Postal Service.



- There are totally 9298 handwritten single digits between 0 and 9, each of which consists of a 16x16 pixel image
- This dataset is described in a paper[2] by J.J. Hull

Why Design Intelligent Systems?

- "Intelligent" (or Smart) indicates physical systems that can interact with their environment and adapt to changes through self-awareness and perceived models of the world, based on quantitative and qualitative information.
  - Smart Grid
  - Smart Defense Systems
  - Intelligent Transportation Systems

Smart Engineering Systems

How can we design such systems?

Are there any tools?

Smart Engineering Systems

Tools and techniques:

Artificial Neural Networks

Deep learning

Fuzzy Logic and Systems

Evolutionary Programming

Data Analytics

Machine Learning

Computational Intelligence

这里所说的很重要，对以后的研究

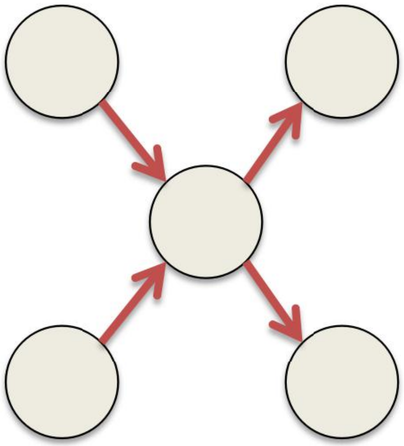
What is an Artificial Neural Network?

- A class of machine learning algorithms originally inspired by the human brain
- Neural networks *model* the way the brain works
  - Only in a very rudimentary sense

A Network of Processing Units

An artificial neural network can be seen as a mathematical model that is designed to recognize patterns and learn "like" the human brain.

- parallel distributed processor made up of simple processing units (*neurons*)
- knowledge stored in the form of interneuron connection strengths (*synaptic weight*)
- learning is performed by a process called a *learning algorithm*



Neural Network Definition

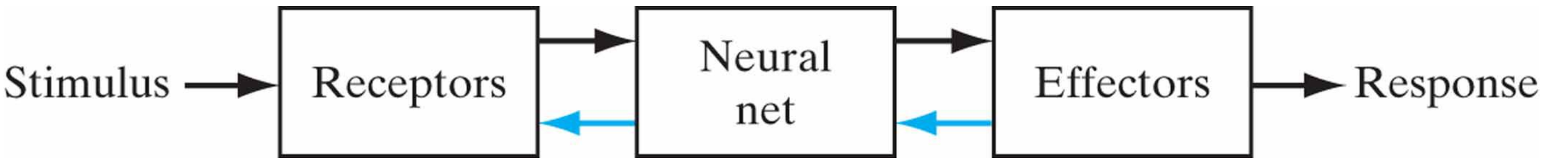
From your textbook[3] –

> **"A neural network is a massively parallel distributed processor that has a natural propensity for storing experimental knowledge and making it available for use. It resembles the brain in two respects:**

1. **Knowledge is acquired through a learning process.**
2. **Inter-unit connection strengths are used to store the knowledge."**

Human Brain

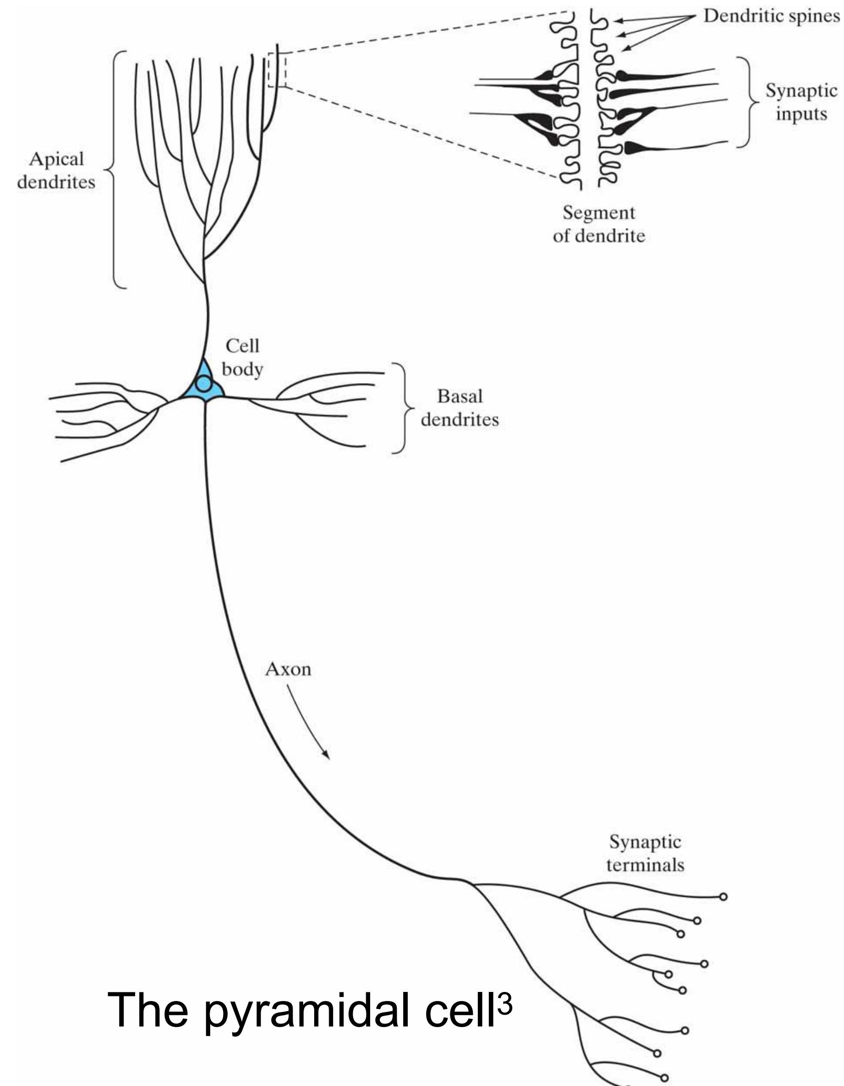
- Block diagram representation of the nervous system.[3]



- Neuron is the fundamental computational unit in the brain
- Neuronal processing speed is slow: 10^{-3} s per operation vs. 10^{-6} s in modern computers.
- Compensated by the staggering number of parallel processing units: 10^{10} neurons and 60 trillion connections.
- Energy efficiency of the brain: 10^{-16} J per operation per second vs. 10^{-6} J per operation per second.

A Typical Neuron

- **Soma**: Cell body
- **Axon**: the transmission line
- **Dendrites:** the reception zones
- **Synapse:** functional units that facilitate interaction between neurons
- **Synaptic transmission:** process of information transfer between neurons



The pyramidal cell[3]

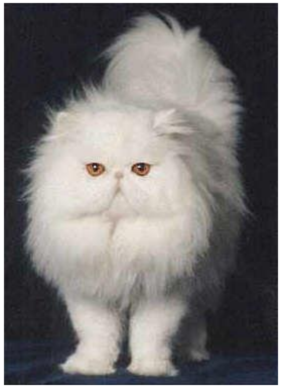
Synaptic Plasticity

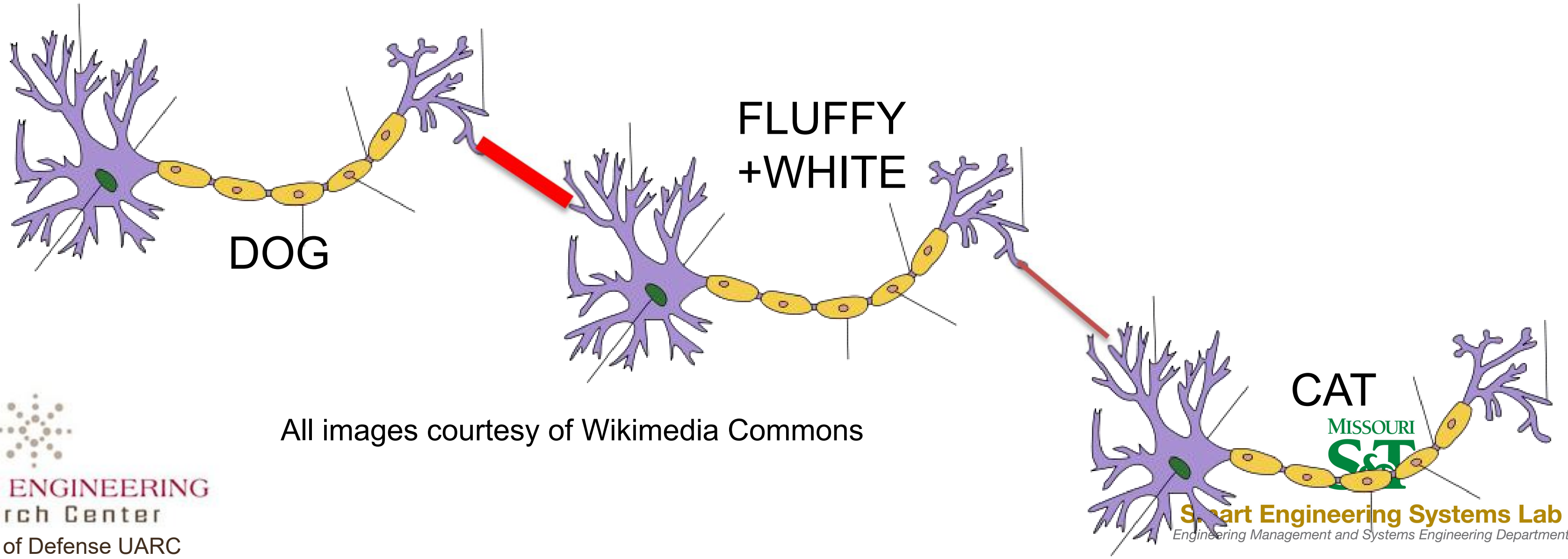
DOG

DOG

DOG

CAT





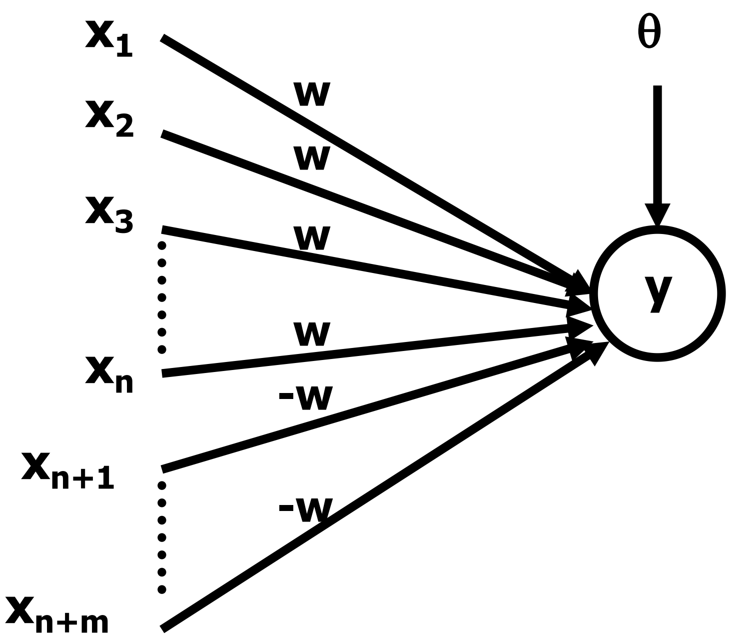
All images courtesy of Wikimedia Commons

McCulloch Pitts Neuron Model

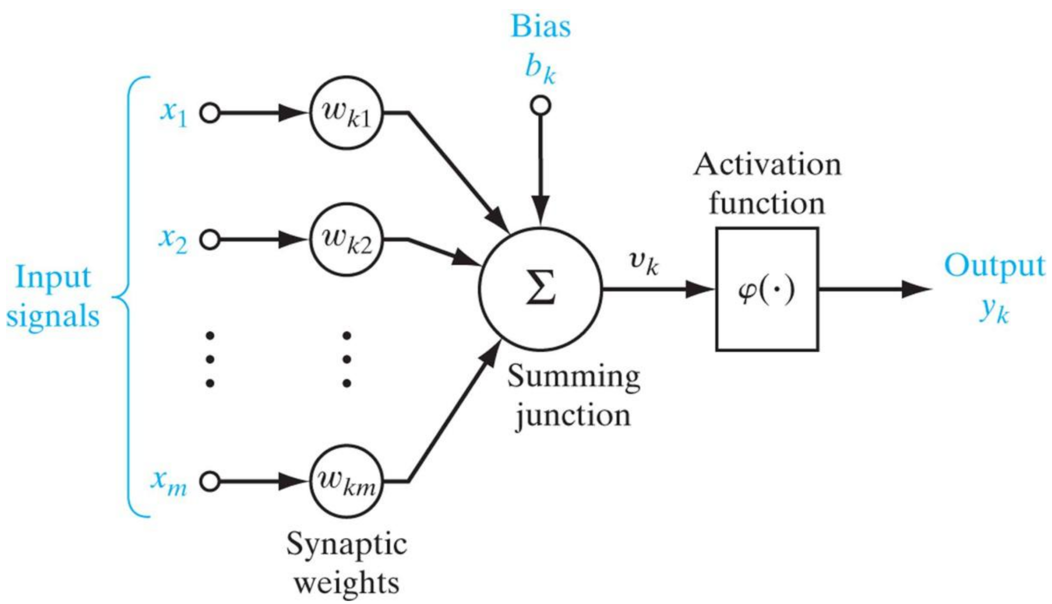
- The first mathematical model describing a biological neuron.
- Very simplistic additive model
- Uses a step function to determine output

$$u = x_1 w + x_2 w + x_3 w + \dots$$

$$y = \begin{cases} 1 \text{ if } u \geq \theta \\ 0 \text{ if } u < \theta \end{cases}$$

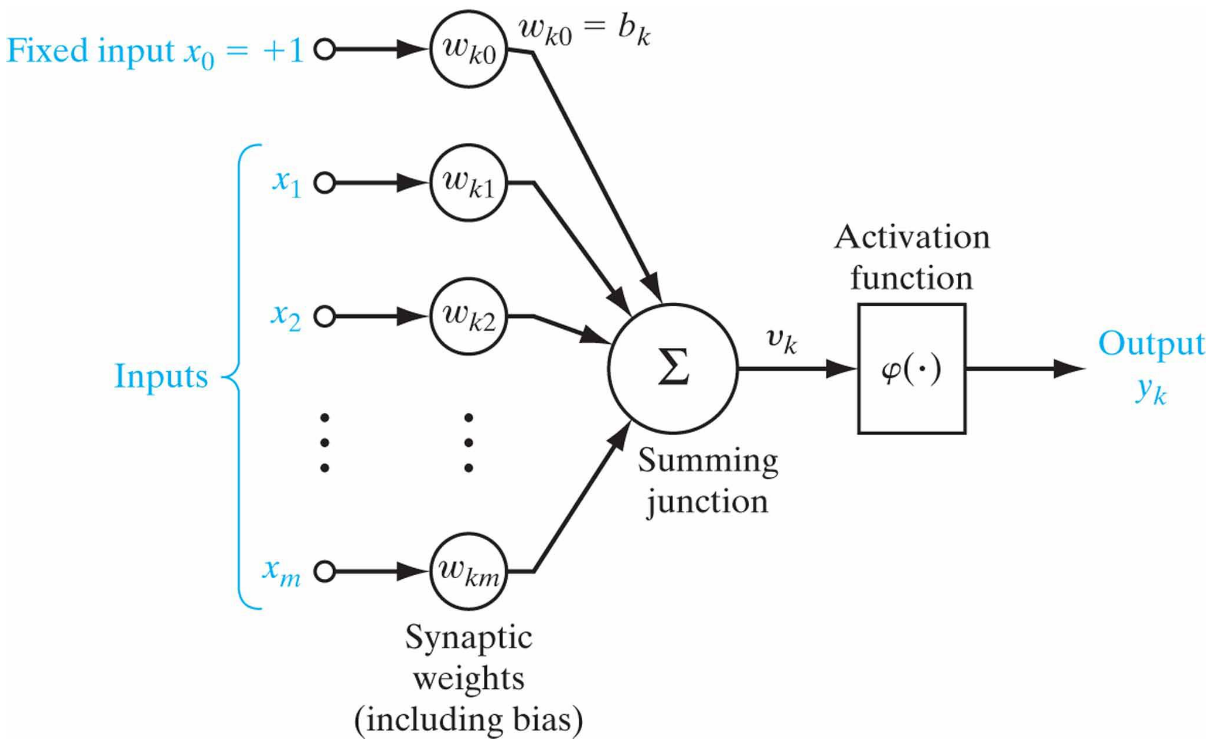


Nonlinear Model of an Artificial Neuron



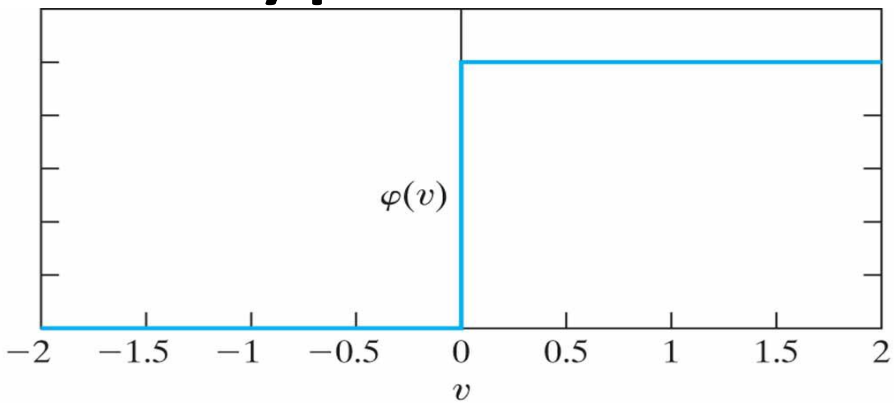
- Input signals: $[x_1, x_2, \ldots, x_m]$
- Synaptic weights: w_{kj}, where k is the neuron index and j is the input index
- Summing junction: $u_k = \sum_{j=1}^{m} w_{kj} x_j, v_k = u_k + b_k$
- Activation function: $\varphi(\cdot)$, Bias: b_k
- Output signal: $y_k = \varphi(u_k + b_k)$

Alternate Model of a Neuron



- Bias is modeled with an additional synapse
- Bias input: $x_0 = +1$ Bias weight: $w_{k0} = b_k$
- Output of the summing junction: $v_k = \sum_{j=0}^{m} w_{kj} x_j$
- Neuron Output: $y_k = \varphi(v_k)$

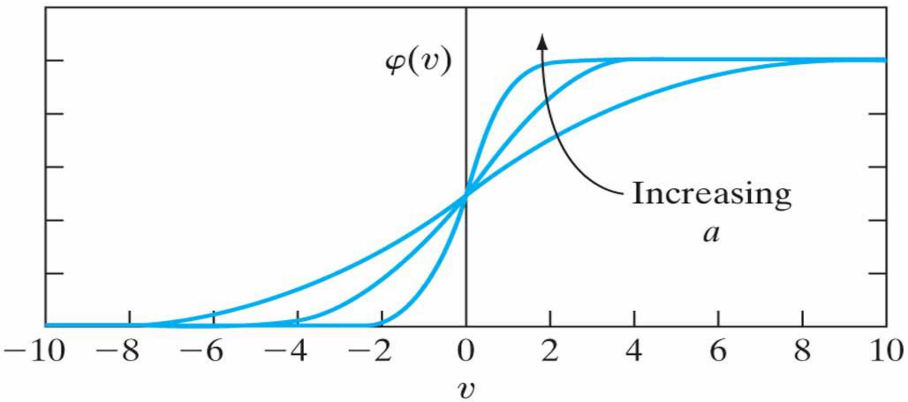
Types of Activation Functions



(a)

(a) Threshold unit or Heaviside function:

$$\varphi(v) = \begin{cases} 1 \text{ if } v \geq 0 \\ 0 \text{ if } v < 0 \end{cases}$$



(b)

(b) Sigmoid Function:

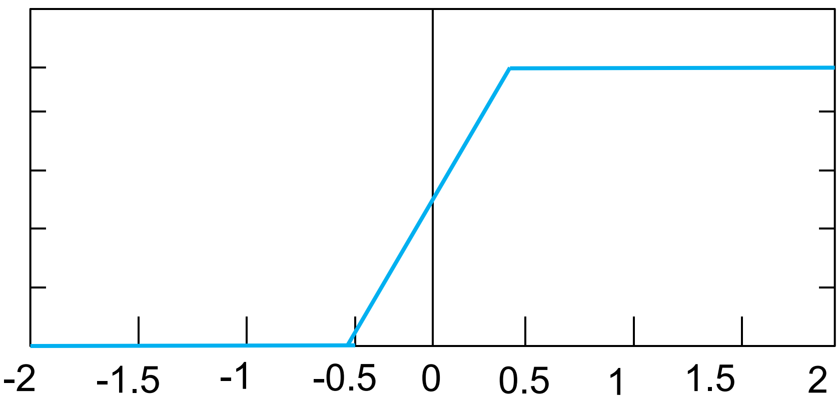
Logistic function,

$$\varphi(v) = \frac{1}{1+\exp(-av)}$$

a is the slope parameter

Activation Functions

(c) Piecewise linear function:



$$\varphi(v)=\begin{cases} 1 \text{ if } v \geq +\frac{1}{2} \\ v \text{ if } +\frac{1}{2} > v > -\frac{1}{2} \\ 0 \text{ if } v \leq -\frac{1}{2} \end{cases}$$

More Activation Functions

- Signum function:

$$\varphi(v) = \begin{cases} 1 \text{ if } v > 0 \\ 0 \text{ if } v = 0 \\ -1 \text{ if } v < 0 \end{cases}$$

- Sign function:

$$\varphi(v) = \begin{cases} 1 \text{ if } v \geq 0 \\ -1 \text{ if } v < 0 \end{cases}$$

- Hyperbolic tangent function: $\varphi(v) = \tanh(v)$

Neural Network Architectures

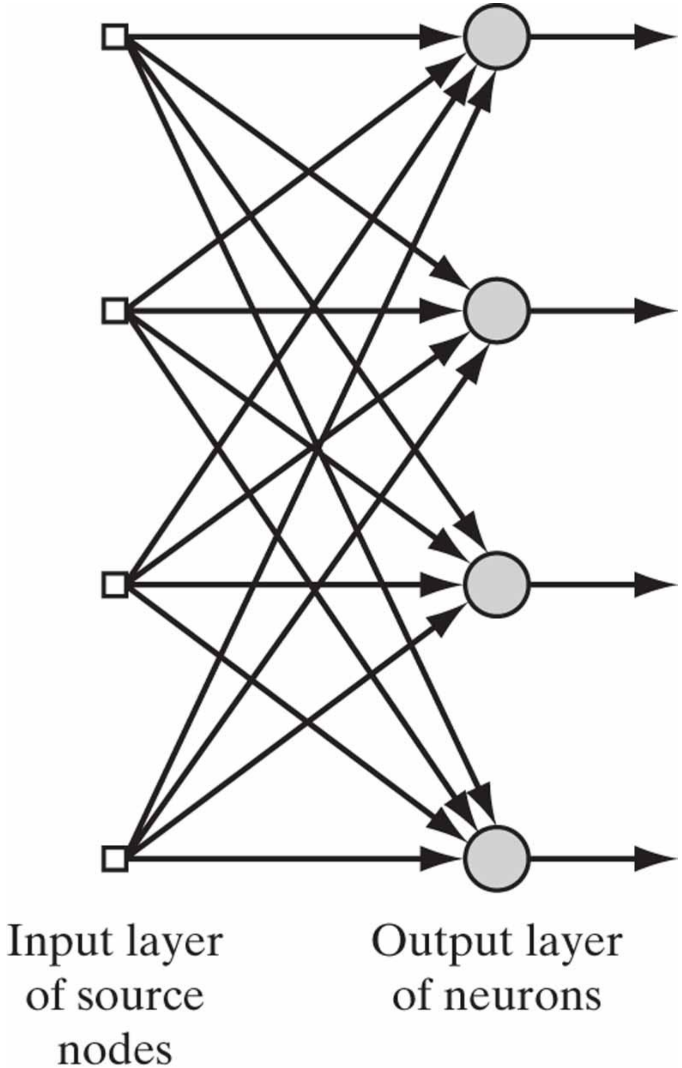
We've seen that neural networks are:

- **An information processing system that has been developed as a generalization of mathematical models of human cognition or neurobiology**
- **Information processing occurs at many simple elements called neurons.**
- **Signals are passed between neurons over connection links.**
- **Each connection link has an associated weight, which typically multiplies the signal transmitted.**
- **Each neuron applies an activation function (usually non-linear) to its net input (sum of weighted input signals) to determine its output signal.**

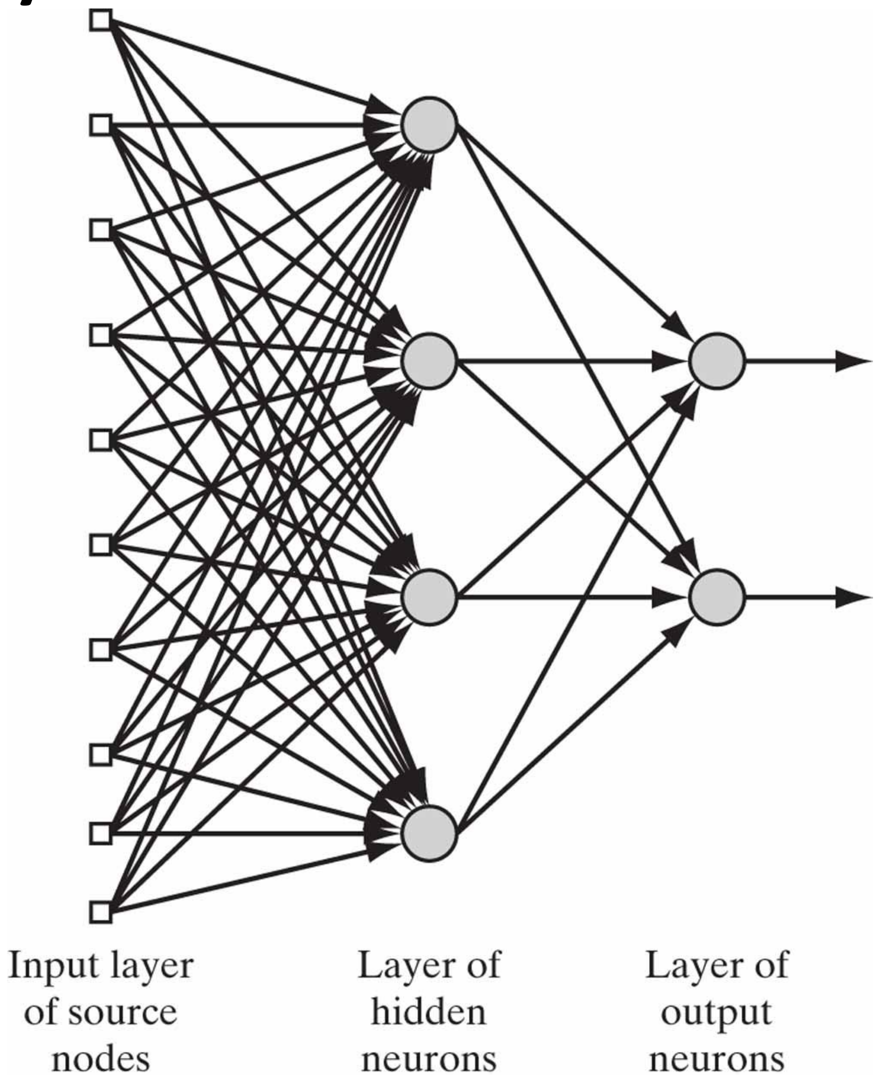
Network Topology

- Architecture refers to the layout of the neurons and their interconnections
- NN architecture or topology is closely related to its learning algorithm
- Three fundamentally different types of topologies commonly occur
  - Single-layer feedforward networks
  - Multilayer feedforward networks
  - Recurrent networks

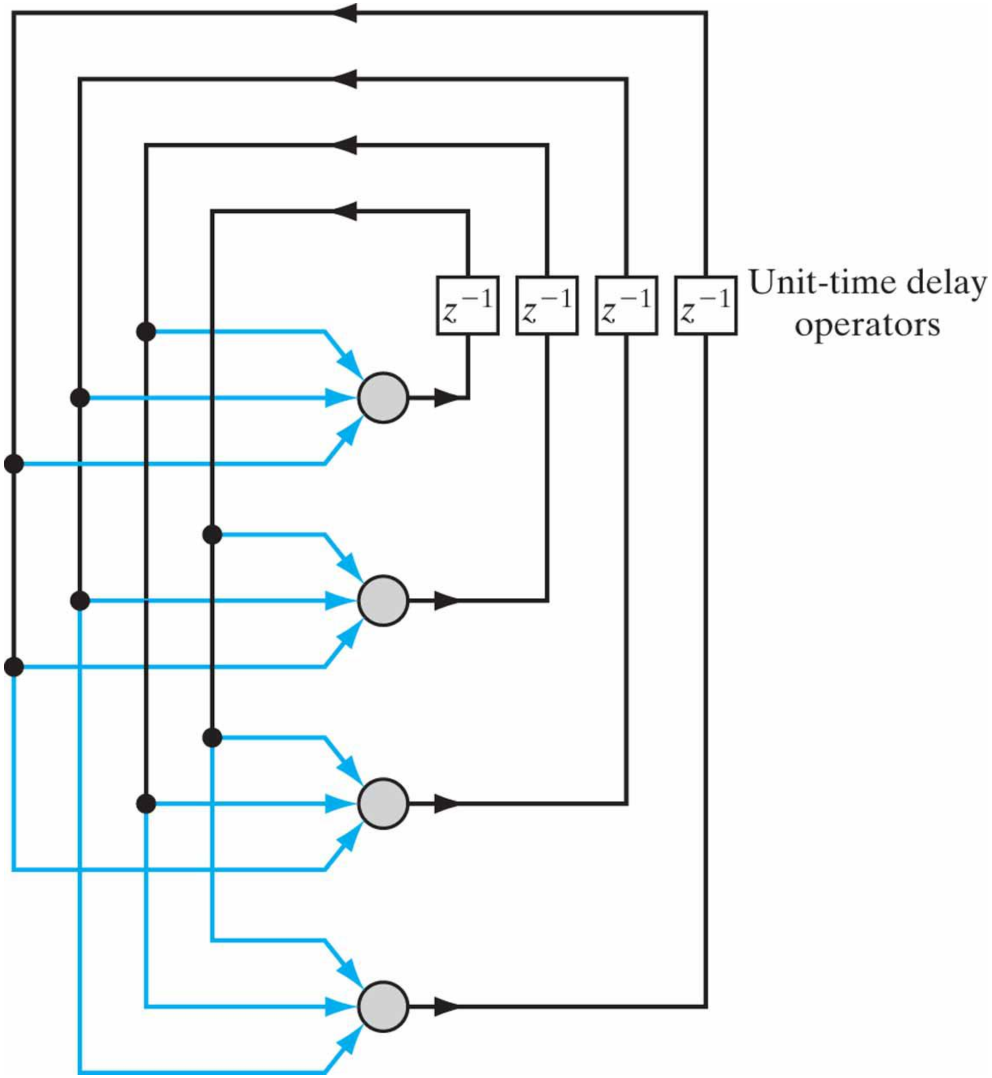
Single-Layer Feedforward Networks



Multilayer Feedforward Networks

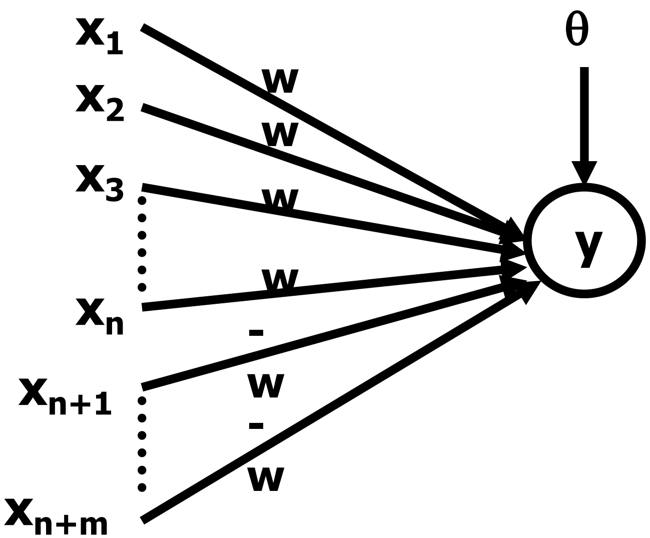


Recurrent Networks



Historical Notes

- **McCulloch and Pitts 1943**
  - Neural networks in principle can compute any arithmetic or logical function



- **Donald Hebb, 1949**
  - **Published the book *The Organization of Behavior***
  - **First presentation of a learning mechanism for modifying synapses**
  - **Hebb's Postulate**

    "When an Axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased."

- **Taylor, 1956**
  – Initiated work on associative memory

- **Kohonen, Nakano, Anderson, 1972**
  – Introduced correlation matrix memory independently in the same year

- **Rosenblatt, 1958**
  – Novel supervised learning method, ‘perceptron’

- **Widrow and Hoff, 1960**
  - Least mean square algorithm
  - Formulated the adaline (adaptive linear element) and madaline (multiple-adaline)

- **Marvin Minsky and Seymour Papert, 1969**

Rosenblatt's and Widrow's Networks cannot Learn Nonlinear Functions; A Dead End Research

- **Von der Malsburg, 1973**
  - Networks Possessing Synaptic Modification Rules Can Modify and Organize Itself .
  - Self organizing maps

- **Werbos, 1974**
  - Developed the Backpropagation Algorithm

- **Little and Shaw, 1975**
  - Probabilistic Model of a Neuron

- **Lee and Lee, 1975**
  - Fuzzy McCulloch-Pitts Neuron Model

Rebirth of Neural Networks Research

- **John Hopfield, 1982**
  - The use of statistical mechanics to explain the operation of a certain class of recurrent networks
- **Kirkpatrick, Gelatt, and Vecchi, 1983**
  - Boltzmann Machine
  - The first realization of a multilayer neural network
  - Proved that Minsky and Papert's speculations were illfounded

- **Rumelhart, Hinton, and Williams, 1986**
  - Independent discoveries of the backpropagation training algorithm
- **Carpenter and Grossberg, 1987**
  - Adaptive Resonance Theory
- **Broomhead and Lowe, 1988**
  - Radial Basis Functions Networks
- **Vapnik, 1990**
  - Invented support vector machines
  - A class of supervised learning algorithms

- **Geoffrey Hinton, 2006**
  - Deep Belief Networks
  - Probabilistic generative models that learn one layer at a time

- **Neural networks today are a well established subject with deep roots in multiple disciplines.**

Useful Properties of ANNs

- Nonlinearity
- Input-output mapping
- Adaptivity
- Evidential response
- Contextual information
- Fault tolerance

What can we do with Neural Networks?

- Classification
- Optimization
- Prediction
- Control
- Function Approximation

Neural Networks in Practice

- **Recognizing patterns:**
  - Facial identities or facial expressions
  - Handwritten or spoken words
  - Medical diagnostics
- **Recognizing anomalies:**
  - Abnormal events in networks (Intrusion detection)
  - Unusual sequences of credit card transactions
  - Unusual patterns of sensor readings in a nuclear power plant
  - Spam detection, malware classification etc.
- **Prediction:**
  - Future stock prices
  - Future currency exchange rates
  - Sales forecasting

The terminator units deployed by Skynet contained a neural net CPU!

A Word of Caution

- **In spite of their capabilities and extensive applications, neural networks are not a cure all.**
- **No free lunch theorem of machine learning**
  - "...all algorithms that search for an extremum of a cost function perform exactly the same, when averaged over all possible cost functions. In particular, if algorithm A outperforms algorithm B on some cost functions, then loosely speaking there must exist exactly as many other functions where B outperforms A..“
- **However, when applied correctly to the correct problem neural networks can prove to be powerful tools in a machine learning engineer’s arsenal.**

About this Course

- **We will learn**
  - About popular neural network models and architectures
  - Learning paradigms and algorithms
  - To build and program neural networks
  - To train them to solve engineering problems such as pattern recognition and function approximation

Software

- For programming assignments use Matlab and the Matlab NN Toolbox
- We will review Matlab next week
- Systems Engineering PhD student Deepak Gottapu, rgrk6@mst.edu will attend all lectures and support the course.

Obtaining Matlab

- **Purchase Matlab student edition and the NN toolbox from**
  - MST Bookstore
  - https://www.mathworks.com/store/link/products/student/new?s_iid=htb_buy_gtwy_cta3

Textbook

Simon Haykin, *Neural Networks and Learning Machines*, 3rd edition, Prentice Hall, Upper Saddle River, NJ, 2008. ISBN 0131471392.

Points Breakdown

Six Homework Assignments..........30%

Two Exams 40%

-- Exam I (in class)..........20%

-- Exam II (take home).....20%

Project.................................... 30%

Class Project

- Individual project applying the techniques learned in class to a real world problem
- Data available from many online databases
- Project deliverables will be part of the homework to help you make regular progress

For Next Tuesday

- **Viewing Assignment**
  - Matlab tutorial videos, 12 units, 120 minutes
  - https://matlabacademy.mathworks.com/?s_tid=acport_tut_sp_til
- **Homework Assignment 1, Part 1**
  - Find one interesting recent application of neural networks and summarize its implementation and purpose.

References

1. http://www.cs.nyu.edu/~roweis/data.html
2. J. J. Hull: A database for handwritten text recognition research," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16, no. 5, pp. 550-554, 1994.
3. All figures adapted from Simon Haykin, *Neural Networks and Learning Machines*, 3rd edition, Prentice Hall, Upper Saddle River, NJ, 2008.