Artificial Neural Networks

Part 1/3

Slides modified from Neural Network Design
by Hagan, Demuth and Beale

Berrin Yanikoglu
DA514– Machine Learning

Biological Inspirations

Biological Inspirations

Humans perform complex tasks like vision, motor control, or language understanding very well.

One way to build intelligent machines is to try to imitate the (organizational principles of) human brain.

Human Brain

- The brain is a highly complex, non-linear, and parallel computer, composed of some 10^{11} neurons that are densely connected ($\sim 10^4$ connection per neuron). *We have just begun to understand how the brain works...*

- A neuron is much slower (10^{-3}sec) compared to a silicon logic gate (10^{-9}sec), however the massive interconnection between neurons make up for the comparably slow rate.
  - Complex perceptual decisions are arrived at quickly (within a few hundred milliseconds)

- 100-Steps rule: Since individual neurons operate in a few milliseconds, calculations do not involve more than about 100 serial steps and the information sent from one neuron to another is very small (a few bits)

- Plasticity: Some of the neural structure of the brain is present at birth, while other parts are developed through learning, especially in early stages of life, to adapt to the environment (new inputs).

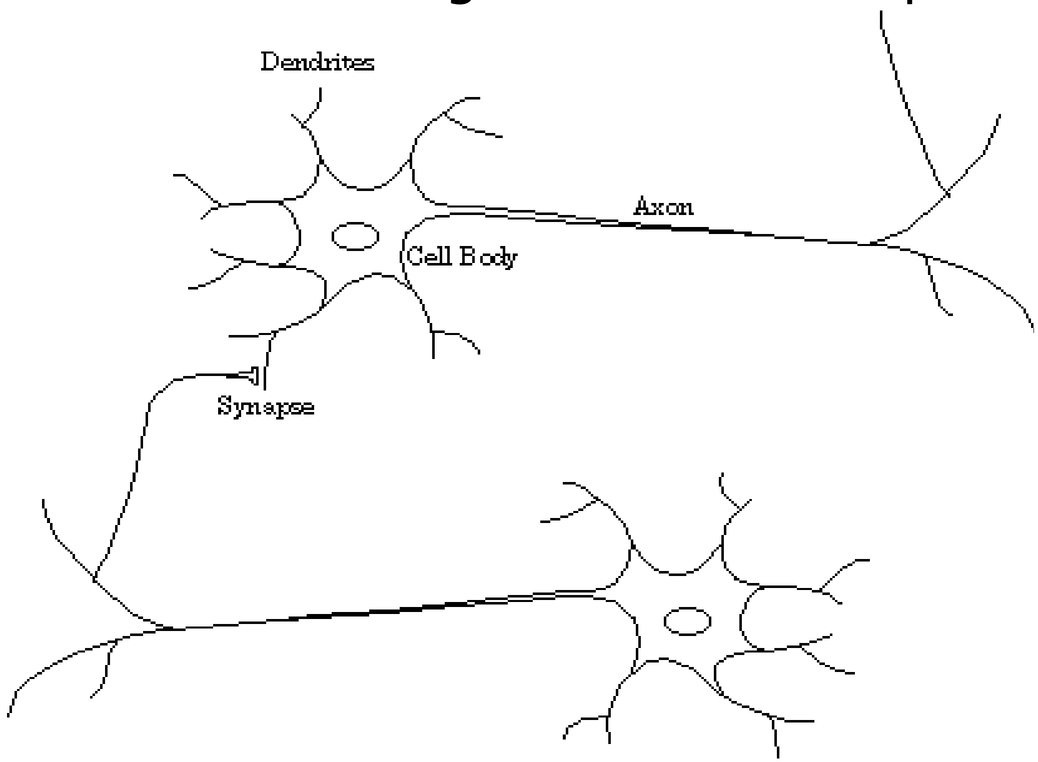
Biological Neuron

A variety of different neurons exist (motor neuron, on-center off-surround visual cells…), with different branching structures.

The connections of the network and the strengths of the individual synapses establish the function of the network.

Biological Neuron

- dendrites: nerve fibres carrying electrical signals to the cell
- cell body: computes a non-linear function of its inputs
- axon: single long fiber that carries the electrical signal from the cell body to other neurons
- synapse: the point of contact between the axon of one cell and the dendrite of another, regulating a chemical connection whose strength affects the input to the cell.



Artificial Neural Networks

Computational models inspired by the human brain:

- Massively parallel, distributed system, made up of simple processing units (neurons)

- Synaptic connection strengths among neurons are used to store the acquired knowledge.

- Knowledge is acquired by the network from its environment through a learning process

Properties of ANNs

Learning from examples

- labeled or unlabeled

Adaptivity

- changing the connection strengths to learn things

Non-linearity

- the non-linear activation functions are essential

Fault tolerance

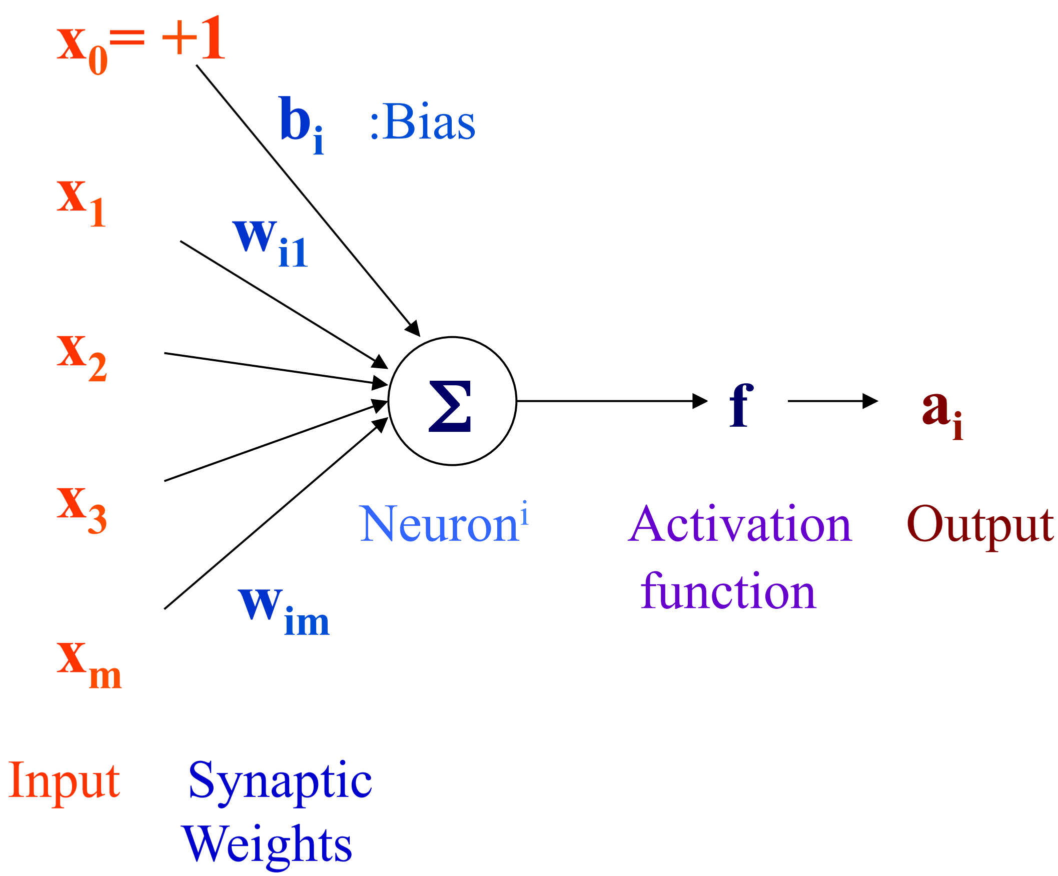
- if one of the neurons or connections is damaged, the whole network still works quite well

Thus, they might be better alternatives than classical solutions for problems characterised by:

- high dimensionality, noisy, imprecise or imperfect data; and
- a lack of a clearly stated mathematical solution or algorithm

Neuron Model
and
Network Architectures

Artificial Neuron Model



Bias

$$a_i = f\,(n_i) = f\,(\sum_{j=1}^{n} w_{ij}x_j + b_i)$$

An artificial neuron:

- computes the weighted sum of its input (called its **net input**)
- adds its bias
- passes this value through an activation function

We say that the neuron "fires" (i.e. becomes active) if its output is above zero.

Bias

Bias can be incorporated as another weight clamped to a fixed input of +1.0

This extra free variable (bias) makes the neuron more powerful.

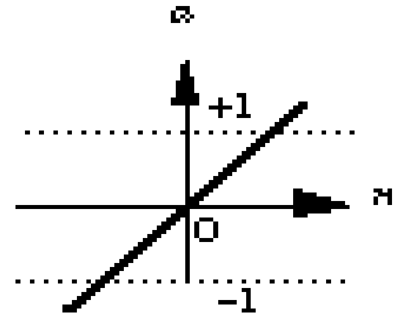
$$a_i = f\,(n_i) = f\,\left(\sum_{j=0}^{n} w_{ij} x_j\right) = f(\mathbf{w}_i . \mathbf{x}_j)$$

Activation functions

Also called the squashing function as it limits the amplitude of the output of the neuron.

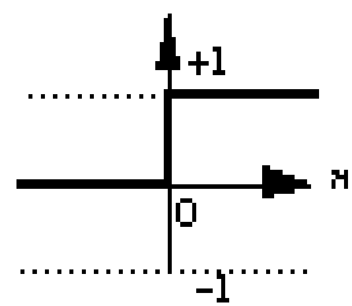
Many types of activations functions are used:

- linear: $a = f(n) = n$

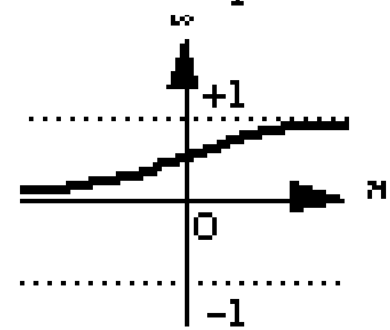


- threshold: (hardlimiting) $a = \{1$ if $n >= 0$

  0 if $n < 0$

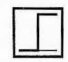
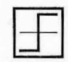
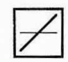
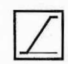
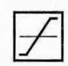
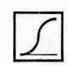
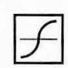


- sigmoid: $a = 1/(1+e^{-n})$



- ...

Activation Functions

Name	Input/Output Relation	Icon	MATLAB Function
Hard Limit	$a = 0 \quad n < 0$ $a = 1 \quad n \geq 0$		hardlim
Symmetrical Hard Limit	$a = -1 \quad n < 0$ $a = +1 \quad n \geq 0$		hardlims
Linear	$a = n$		purelin
Saturating Linear	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n \leq 1$ $a = 1 \quad n > 1$		satlin
Symmetric Saturating Linear	$a = -1 \quad n < -1$ $a = n \quad -1 \leq n \leq 1$ $a = 1 \quad n > 1$		satlins
Log-Sigmoid	$a = \dfrac{1}{1 + e^{-n}}$		logsig
Hyperbolic Tangent Sigmoid	$a = \dfrac{e^{n} - e^{-n}}{e^{n} + e^{-n}}$		tansig
Positive Linear	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n$		poslin
Competitive	$a = 1$ neuron with max n $a = 0$ all other neurons	C	compet

Artificial Neural Networks

A neural network is a massively parallel, distributed processor made up of simple processing units (artificial neurons).

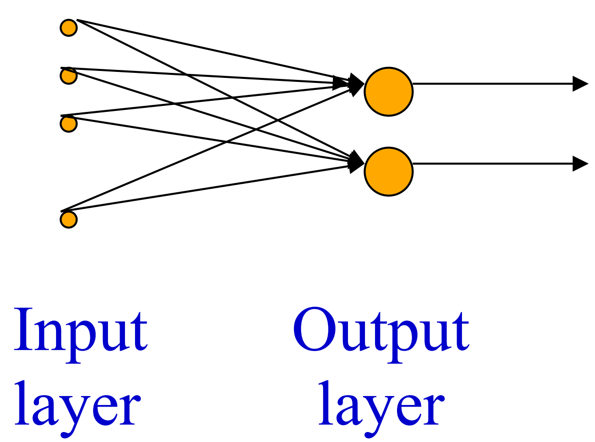
It resembles the brain in two respects:

- Knowledge is acquired by the network from its environment through a learning process
- Synaptic connection strengths among neurons are used to store the acquired knowledge.

Different Network Topologies

Single layer feed-forward networks

– Input layer projecting into the output layer

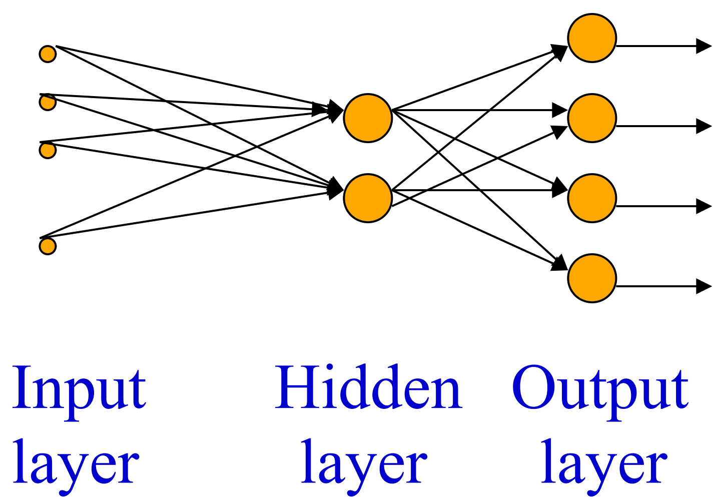


Different Network Topologies

Multi-layer feed-forward networks

- One or more hidden layers.
- Input projects only from previous layers onto a layer.

  typically, only from one layer to the next

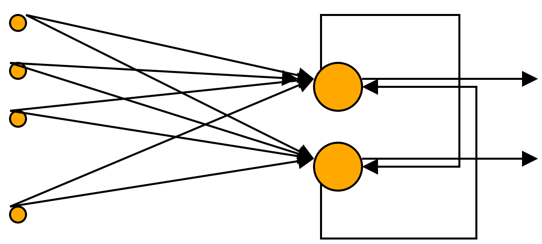


2-layer or
1-hidden layer
fully connected
network

Different Network Topologies

Recurrent networks

– A network with feedback, where some of its inputs are connected to some of its outputs (discrete time).



Input layer

Output layer

Applications of ANNs

ANNs have been widely used in various domains for:

- Pattern recognition
- Function approximation
- Associative memory
- ...

Artificial Neural Networks

Early ANN Models:

- Perceptron, ADALINE, Hopfield Network

Current Models:

- Deep Learning Architectures
- Multilayer feedforward networks (Multilayer perceptrons)
- Radial Basis Function networks
- Self Organizing Networks
- ...

How to Decide on a Network Topology?

- # of input nodes?
  - Number of features
- # of output nodes?
  - Suitable to encode the output representation
- transfer function?
  - Suitable to the problem
- # of hidden nodes?
  - Not exactly known

Multilayer Perceptron

Each layer may have different number of nodes and different activation functions

But commonly:

- **Same activation function within one layer**
  - sigmoid/tanh activation function is used in the hidden units, and
  - sigmoid/tanh or linear activation functions are used in the output units depending on the problem (classification-sigmoid/tanh or function approximation-linear)

Neural Networks Resources

Reference

Neural Networks Text Books

Main text books:

- “Neural Networks: A Comprehensive Foundation”, S. Haykin (very good -theoretical)
- “Pattern Recognition with Neural Networks”, C. Bishop (very good-more accessible)
- “Neural Network Design” by Hagan, Demuth and Beale (introductory)

Books emphasizing the practical aspects:

- “Neural Smithing”, Reeds and Marks
- “Practical Neural Network Recipees in C++”’ T. Masters
- Seminal Paper (but now quite old!):
  – “Parallel Distributed Processing” Rumelhart and McClelland et al.

Deep Learning books and tutorials:

- http://www.deeplearningbook.org/

Neural Networks Literature

Review Articles:

R. P. Lippman, "An introduction to Computing with Neural Nets"' IEEE ASP Magazine, 4-22, April 1987.

T. Kohonen, "An Introduction to Neural Computing", Neural Networks, 1, 3-16, 1988.

A. K. Jain, J. Mao, K. Mohuiddin, "Artificial Neural Networks: A Tutorial"' IEEE Computer, March 1996' p. 31-44.

Journals:

IEEE Transactions on NN

Neural Networks

Neural Computation

Biological Cybernetics

...