Introduction to Neural Networks

Sanjay Singh[⋆†]

[⋆]Department of Information and Communication Technology
Manipal Institute of Technology, MAHE
Karnataka-576104, INDIA
sanjay.singh@manipal.edu

[†]Centre for Artificial and Machine Intelligence (CAMI)
MAHE, Karnataka-576104, INDIA

January 16, 2019

Neural Networks in the Brain

- Human brain "computes" in an entirely different way from conventional digital computers
- The brain is highly complex, nonlinear, and parallel system
- Organization of neurons to perform tasks much faster than computers
- Key features of the biological brain: **experience** shapes the wiring through **plasticity**, and hence **learning** becomes the central issue in neural networks

Neural Networks as an Adaptive Machine

A neural network is a massively parallel distributed processor made up of simple processing units, which has a natural propensity for storing experimental knowledge and making it available for use.
Neural networks resemble the brain:

- Knowledge is acquired from the environment through a learning process
- Inner-neuron connection strengths, known as synaptic weights, are used to store the acquired knowledge

Procedure used for learning: **learning algorithm**. Weights or even the topology can be adjusted.

Benefits of Neural Networks

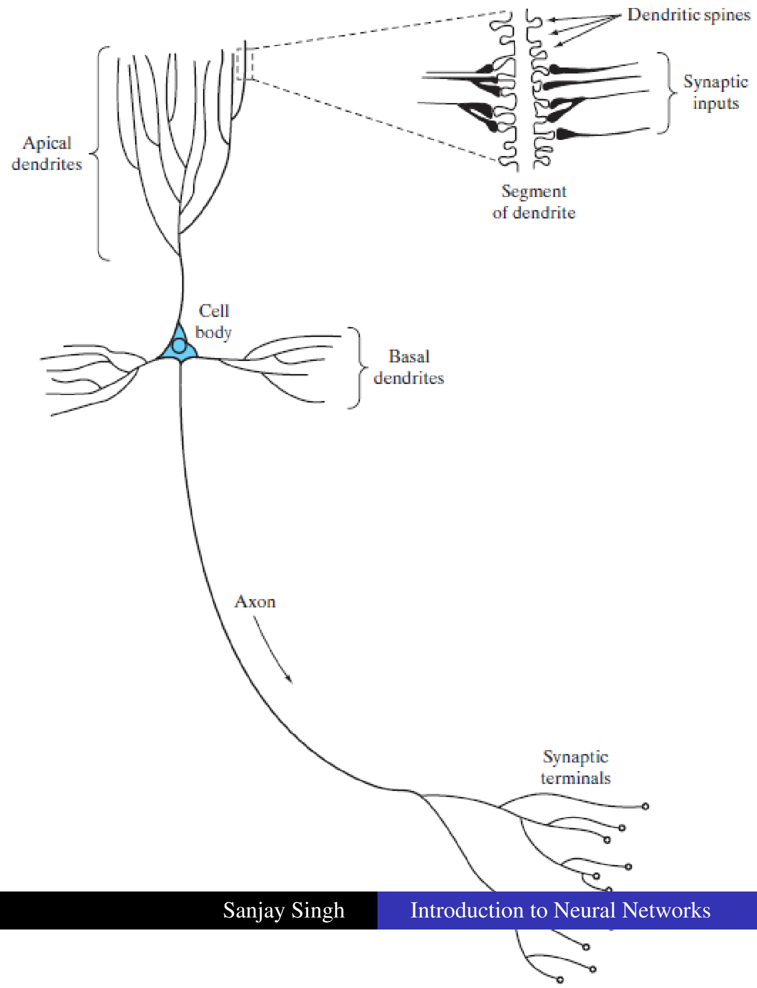
- **Non-linearity**: nonlinear components, distributed non-linearity
- **Input-output mapping**: supervised learning, nonparametric
- **Adaptivity**: either retain or adapt. Can deal with non-stationary environments
- **Evidential response**: decision plus confidence of the decision can be provided
- **Contextual information**: every neuron in the network potentially influences every other neuron, so contextual information is dealt with naturally

Benefits of Neural Networks

- **Fault tolerance**: performance degrades gracefully
- **VLSI implement-ability**: network of simple components
- **Uniformity of analysis and design**: common components, sharability of theories and learning algorithms, and seamless integration based on modularity
- **Neurobiological analogy**: Neural nets motivated by neurobiology, and engineers turn to neural networks for insights and tools.

Human Brain

- Santiago Ramon Y Cajal, a spanish neuroanatomist who introduced neurons as a fundamental unit of brain function
- Neurons are slower than silicon logic gates: 10^{-3} s per operation as compared to 10^{-9} s of silicon chips
- Why does still brain so fast?
- Huge number of neurons and interconnections: 10^{10} neurons and 6×10^{13} connections in human brain
- Highly energy efficient: 10^{-16} J per operation per seconds vs 10^{-6} J per operation per seconds in modern computers



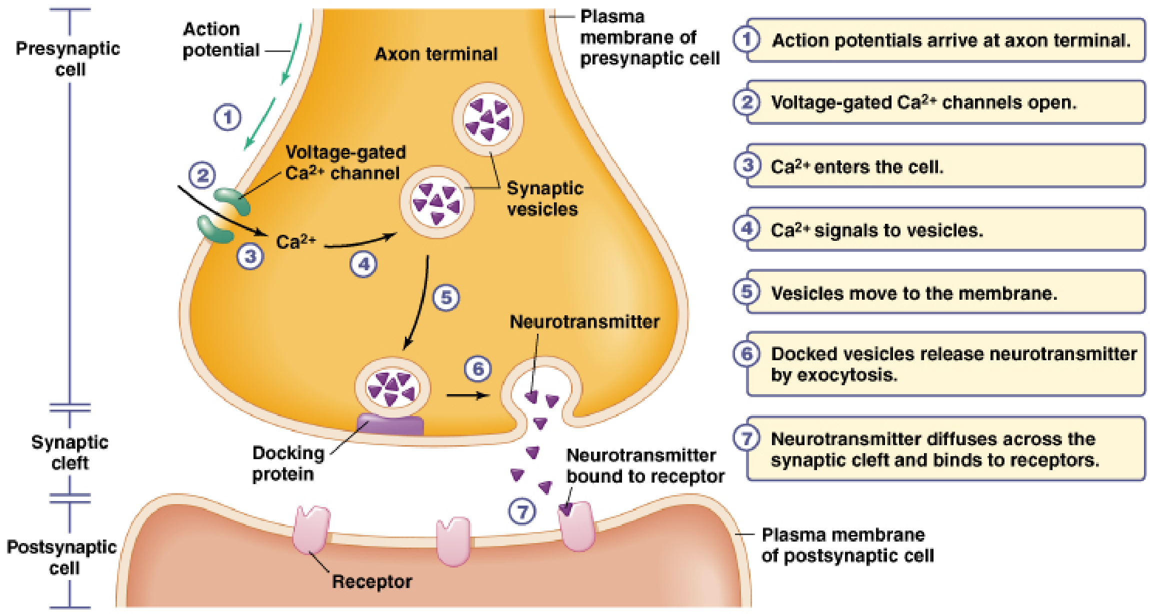


Figure 2: working of a synapse

Some terminologies

- Synapse: where two neurons meet
- Presynaptic neuron: source
- Postsynaptic neuron: target
- Neurotransmitters: molecules that cross the synapse
- Dendrite: branch that receive input
- Axon: branch that sent output (spike or action potential traverses the axon and triggers neurotransmitter release at axon terminals)

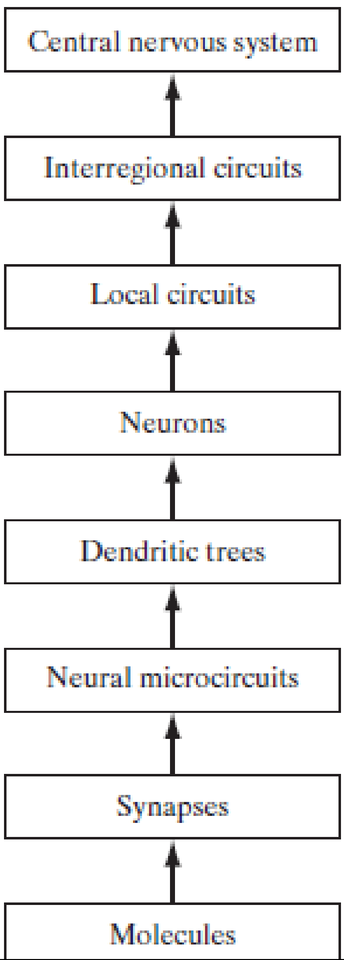


Figure 3

- Molecules,Synapses
- Neural microcircuits refers to an assembly of synapses organized into pattern of connectivity to produce functional units
- Dendritic trees- group of microcircuits
- Local circuits-made up of neurons with similar of different properties-perform operations characteristics of a localized region in brain
- Inter-regional circuits-made up of pathways, columns, topographic maps from different parts of the brain

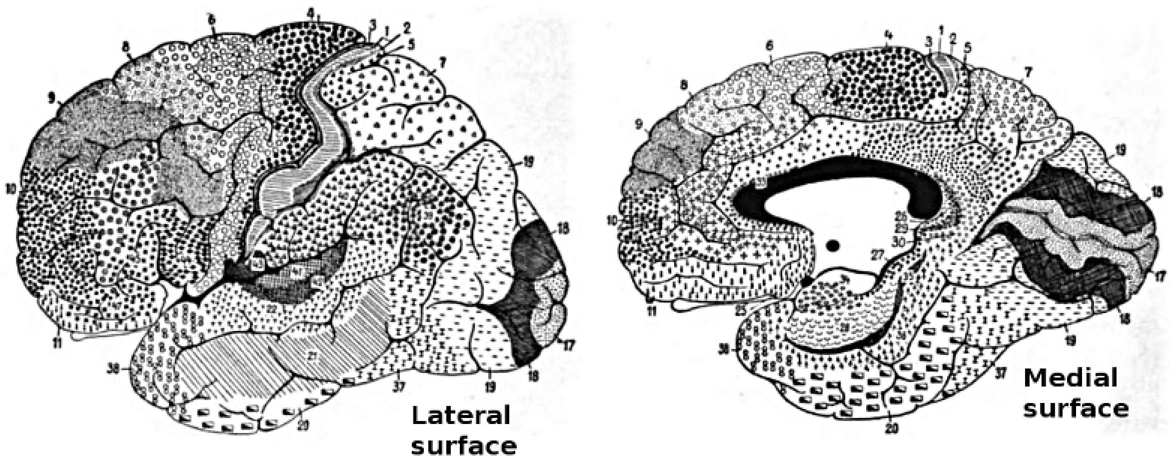
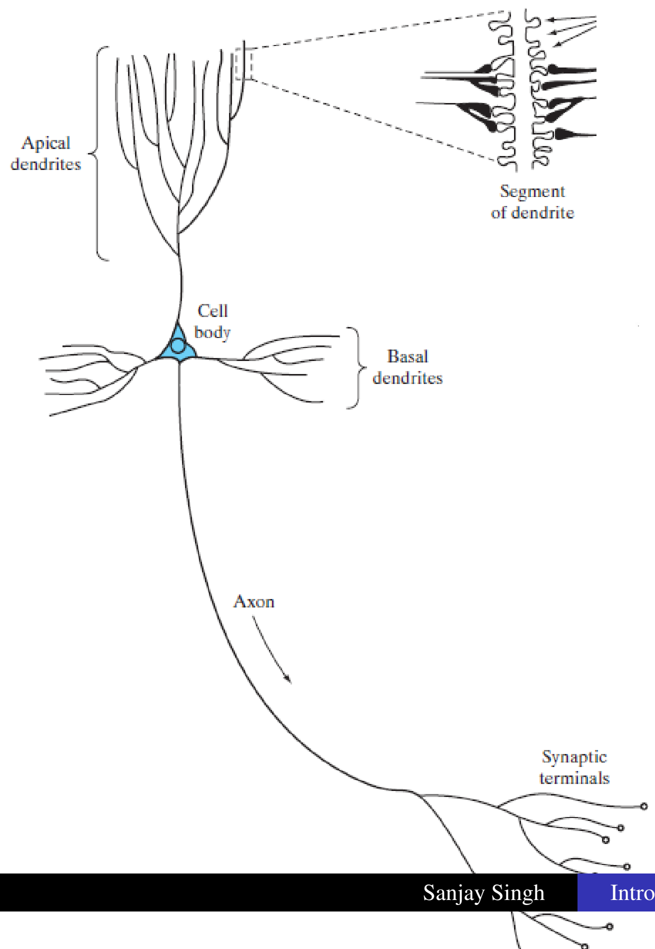
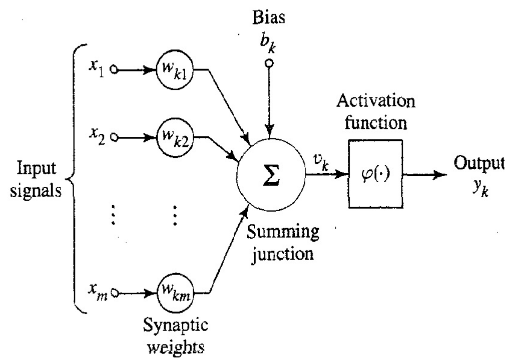


Figure 4: Cytoarchitectural map of the cerebral cortex

Models of a Neuron





Models of a Neuron

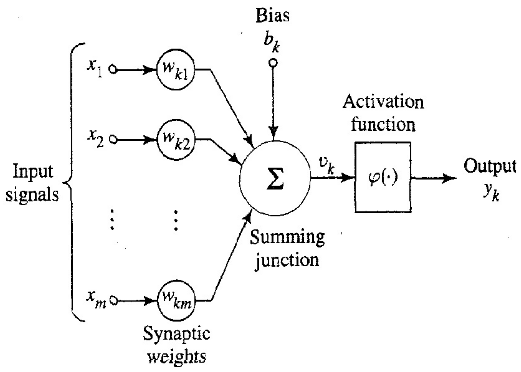


Figure 5: Non-linear model of a neuron

- Set of synapse or connecting links
- Adder
- Activation function to limit the output of a neuron, which is normalized to either [0,1] or [-1,1]

Mathematically neuron k can be described as

- $u_k = \sum_{j=1}^{m} w_{kj} x_j$
- $y_k = \varphi(u_k + b_k)$
- The bias b_k has the effect of applying an affine transformation to the output of the linear combiner

$$v_k = u_k + b_k$$

- v_k is called as induced local field or activation potential

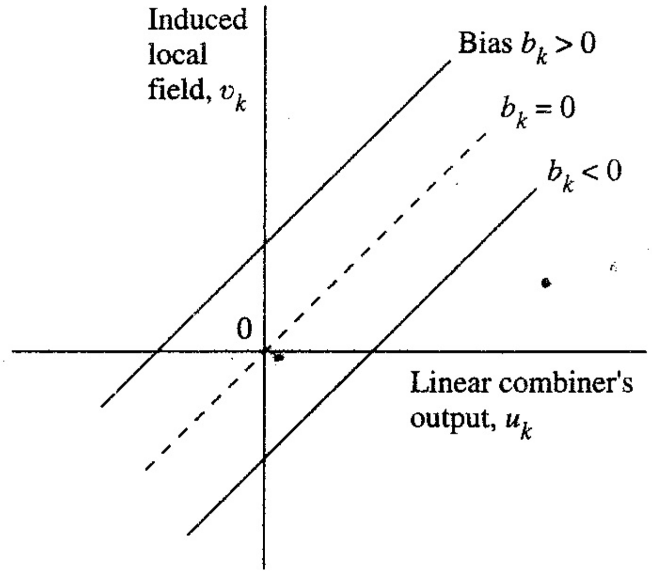


Figure 6: Affine transformation produced by the presence of bias, b_k

Why to bother bias b_k

Bias value allows one to shift activation function to the left or right, which may be critical for successful learning

Figure 7: A neuron without bias.

- output $= sigm(w_0 * x)$
- Change in weight w_0 changes the steepness of sigmoid
- What if you wanted the network to output 0 when x is 2?
- Steepness of sigmoid won't help. What can be done?
- Shift the entire curve to the right
- This is achieved by using bias

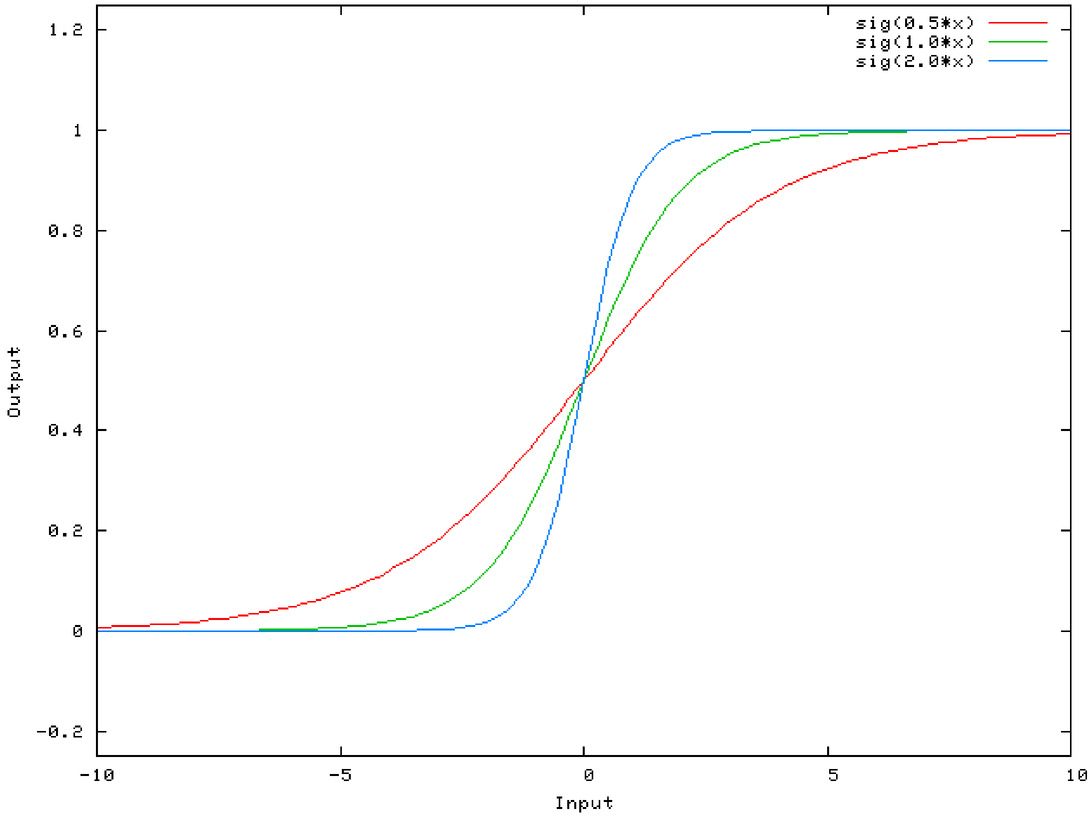


Figure 8: Output of a simple neuron without bias with different weights w_0.

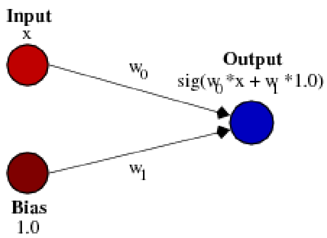


Figure 9: A neuron with bias

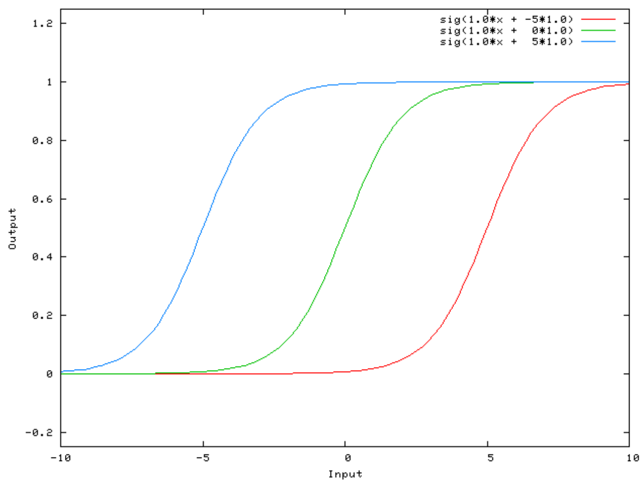


Figure 10: Output of neuron with bias for different weights w_0.

Types of activation function

- Threshold Function

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

- Piecewise-Linear Function

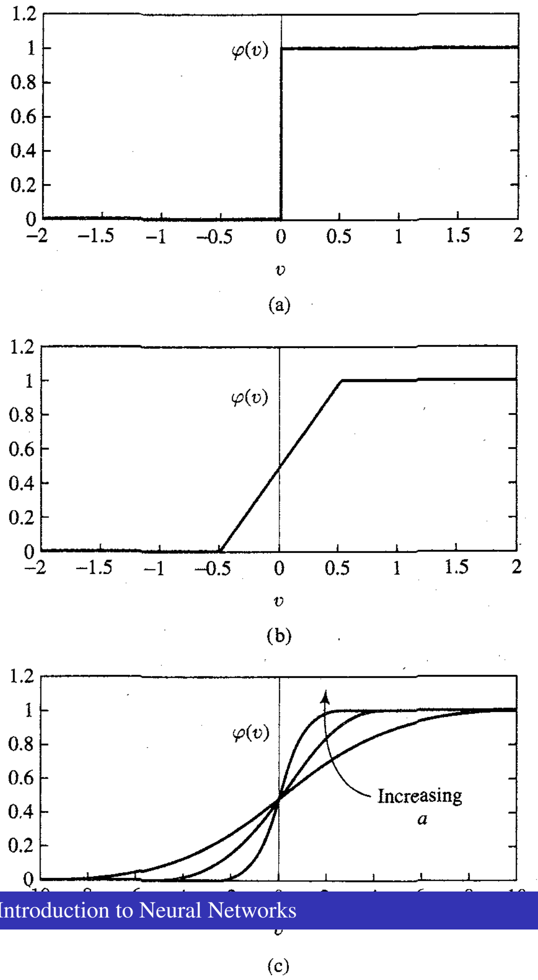
$$\varphi(v) = \begin{cases} 1 & v \geq \frac{+1}{2} \\ v & \frac{-1}{2} < v < \frac{+1}{2} \\ 0 & v \leq \frac{-1}{2} \end{cases}$$

- Sigmoid Function

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

- Signum Function

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



Stochastic Model of Neuron

- Activation function of McCulloch-Pitts model is given probabilistic interpretation
- Neuron is permitted to reside in only one of two states :+1,-1
- Decision for a neuron to fire (i.e.,off to on) is probabilistic
- If x denote the state of neuron, and $P(v)$ probability of firing, then

$$x = \begin{cases} +1 & \text{with probability } P(v) \\ -1 & \text{with probability} 1 - P(v) \end{cases}$$

- $P(v) = \dfrac{1}{1 + exp(-v/T)}$, T is a pseudo-temperature that is used to control noise level and therefore uncertainty in firing
- T controls the thermal fluctuations representing the synaptic noise
- When $T \rightarrow 0$, the stochastic neuron reduces to a noiseless i.e., deterministic

Neural Networks as Directed Graphs

- Block diagram representation of neuron can be further simplified using signal-flow graphs
- Signal-flow graphs with well-defined set of rules were developed by Mason[1]
- Signal-flow graphs provide a neat method for the portrayal of flow of signals in a neural network
- A signal-flow graph is a n/w of directed links (branch) that are interconnected at a point called nodes
- A typical node j has an associated node signal x_j

[1] Wikipedia. *Signal-flow graph — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Signal-flow_graph&oldid=723888623. [Online; accessed 9-Jan-2019]. 2019.

Rules of Signal-flow Graph

Rule 1 A signal flows along a link only in the direction defined by the arrow on the link
- Synaptic links, and
- Activation link

Rule 2 A node signal equals the algebraic sum of all signals entering a node via incoming links

Rule 3 Signal at a node is transmitted to each outgoing link originating from that node, with transmission being entirely independent of transfer function of outgoing links

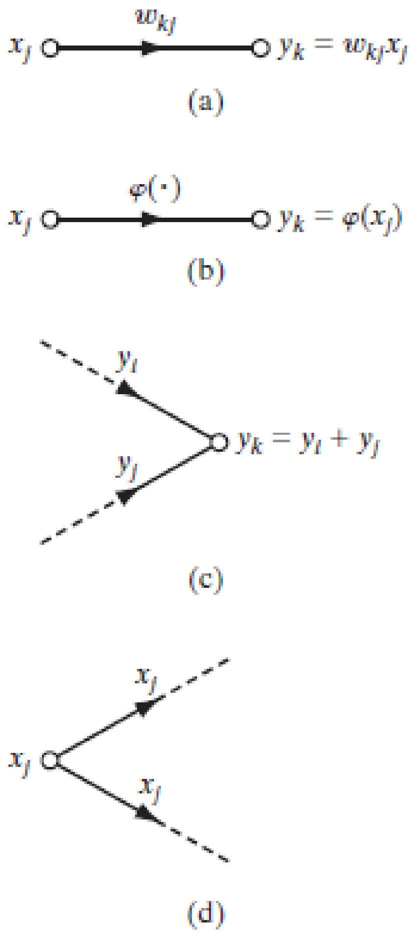


Figure 12: Illustrating basic rules for the construction of signal-flow graph

A neural network is a directed graph consisting of nodes with interconnecting synaptic and activation links, and is characterized by following properties:

- Each neuron is represented by a set of linear synaptic links, an externally applied bias and a possibly nonlinear activation link. The bias is represented by a synaptic link connected to an input fixed at +1
- Synaptic links of a neuron weight their respective input signals
- Weighted sum of the input signals defines the induced local field of the neuron in question
- Activation link squashes the induced local field of the neuron to produce an output.

Types of Signal-flow Graph

- Complete-it describes not only the signal flow from neuron to neuron, but also the signal flow inside each neuron
- Partially complete-it describes only the signal flow from neuron to neuron, and characterized as follows:
  - Source nodes supply signals to the graph
  - Each neuron is represented by a single node called a computation node
  - Communication links interconnecting the source and computation nodes of the graph carry no weight; they merely provide directions of signal flow in the graph
- A partially complete directed graph defined in this way is called as architectural graph, describing the layout of the neural network.

Graphical Representation of Neural Network

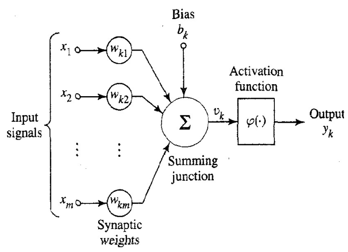


Figure 13: Block diagram representation of a neuron

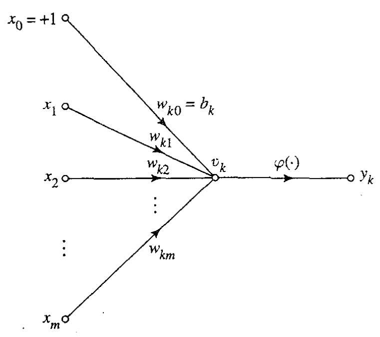


Figure 14: Signal-flow graph representation of a neuron

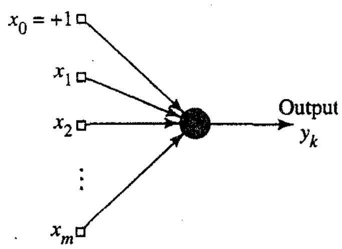


Figure 15: Architectural graph, describing network layout

Connectivity of a neural network is intimately linked with the learning algorithm

- Single-layer feedforward networks: one input layer, one layer of computing units (output layer), acyclic connections
- Multilayer feedforward networks: one input layer, one (or more) hidden layers, and one output layer. With more hidden layers higher-order statistics[2] can be extracted
- Recurrent networks: feedback loop exists

Layers can be fully connected or partially connected. Common notation to express a feedforward network: $m - h_1 - h_2 - q$.

[2] Wikipedia. *Higher-order statistics — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Higher-order_statistics&oldid=745039645. [Online; accessed 10-Jan-2019]. 2019.

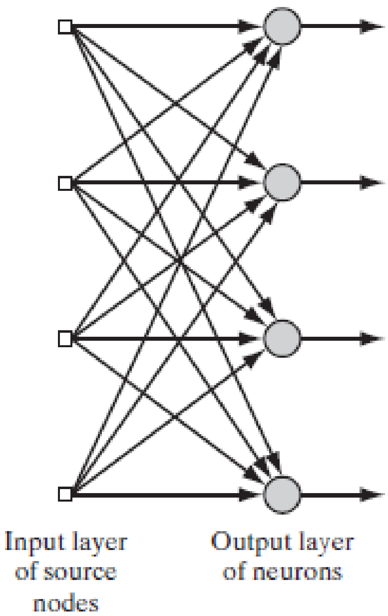


Figure 16: Single layer feedforward network

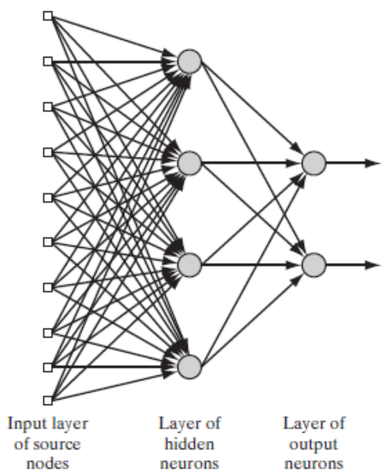


Figure 17: Multilayer neural network

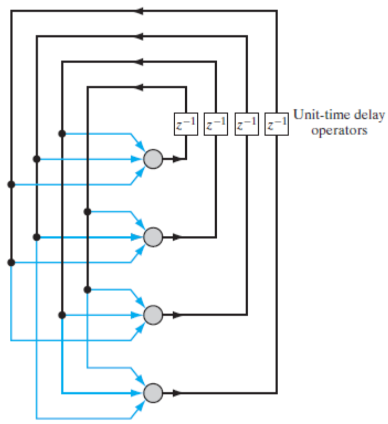


Figure 18: Recurrent network

Knowledge Representation

Knowledge refers to stored information or models used by a person or machine to interpret, predict, and appropriately respond to the outside world.

- What information is actually made explicit
- How the information is physically encoded for subsequent use

Knowledge of the world consist of two kinds of information:

- The known world state: what is and what has been known-*prior information*
- Observation (measurements) of the world, obtained by sensors (they can be noisy). They provide examples. Examples can be labeled or unlabeled

Design of Neural Networks

- Select architecture, and gather input samples and train using a learning algorithm (learning phase)
- Test with data not seen before (generalization phase)
- It is data-driven, unlike conventional programming

Problem specification help define the network in the following ways:

- Number of network inputs= number of problem inputs
- Number of neurons in output layer=number of problem outputs
- output layer transfer function choice is partly determined by problem specification of the output

Example

A single-layer neural network is to have six inputs and two outputs. The outputs are to be limited to and continuous over the range 0 to 1. What can you tell about the network architecture? Specifically:

- How many neurons are required?
- What are the dimensions of the weight matrix?
- What kind of transfer functions could be used?
- Is a bias required?

Rules of Knowledge Representation

- R1 Similar inputs from similar classes should produce similar representation inside the network, and should be classified into the same category
- R2 Items to be categorized as separate classes should be given widely different representations in the network
- R3 If a particular feature is important, then there should be a large number of neurons involved in the representation of that item in the network
- R4 Prior information and invariances should be built into the design of a neural network, thereby simplifying the network design by not having to learn them

Building Prior Information into Neural Network Design

Currently there are no well defined rules for doing this, we have some ad-hoc procedures that are known to give good results

1. Restricting the network architecture through the use of local connections known as [3]receptive fields
2. Constraining the choice of synaptic weights through the use of weight-sharing

Due to 2nd point the number of free parameters in the network is reduced significantly

[3] In the context of visual system, the receptive field of a neuron refers to the restricted area on the retinal surface, which influences the discharge of that neuron due to light

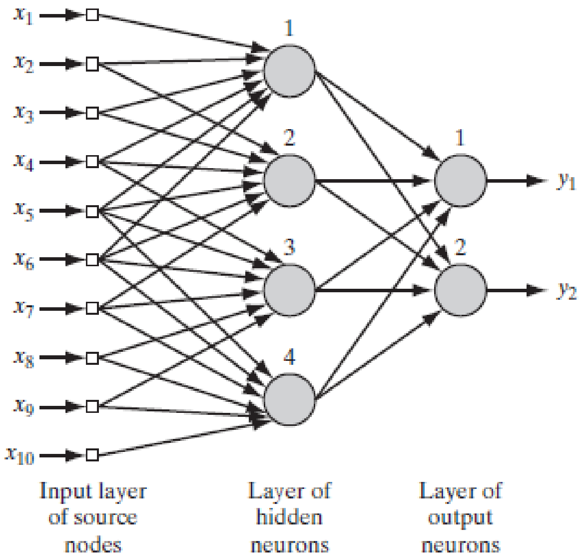


Figure 19: Illustrating the combined use of a receptive field and weight-sharing

- Fig.19 has restricted architecture by construction

- Six source nodes constitute the receptive field for the hidden neuron 1 and so on
- Weight-sharing constraint:use same set of synaptic weights for each neurons in the hidden layer
- We may express the induced local field of hidden neuron s as

$$v_j = \sum_{i=1}^{6} w_i x_{i+j-1} \quad j = 1, 2, 3, 4$$

- Convolutional network

Building Invariances into Neural Network Design

- Invariance by structure
- Invariance by training
- Invariant feature space

Example

An example of the logistic function is defined by

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

whose limiting values are 0 and 1. What is the value of slope at origin

Example

A neuron *j* receives inputs from four other neurons whose activity levels are 10, -20, 4, and -2. The respective synaptic weights of neuron *j* are 0.8, 0.2, -1.0, and -0.9. Calculate the output of neuron *j* for the following situations:

- The neuron is linear
- The neuron is represented by a McCulloch-Pitts model. Assume that there is no bias applied to the neuron.