Neural Networks and Learning Systems – I

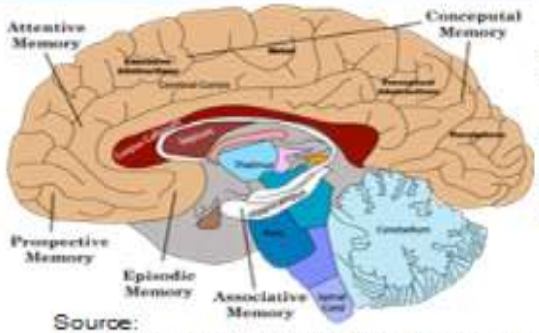
Introduction

Artificial neural networks are motivated by the brain, the way computation works.

BRAIN

Massive highly complex, non-linear, parallel and distributed information processing system

The Grand Problem: Model and engineer associative memories



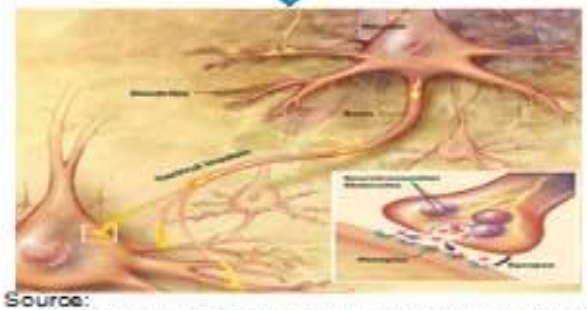
Source:
http://animalia-life.club/other/human-brain-parts-memory.html

Human Brain

- ~10^{10} neurons
- Each neuron connected to roughly 10000 neurons, ~ 1.5 kg adult
- 10^{-16} J/operations/s, 20-40W
  ~ 20% of body power
- Ion transport phenomenon
- Operating temperature 37 ± 2°C
- ~7.5 hrs. good sleep.

Silicon Equivalent

- ~1 million+ transistors
- ~ tolerable 50W
- ~ 2 G/s firing
- Electron-hole transport phenomenon
- Operating temperature 15-85 degrees
- Heat sink/fans
- No sleep required



Source:
https://upload.wikimedia.org/wikipedia/commons/3/30/Chemical_synapse_schema_cropped.jpg

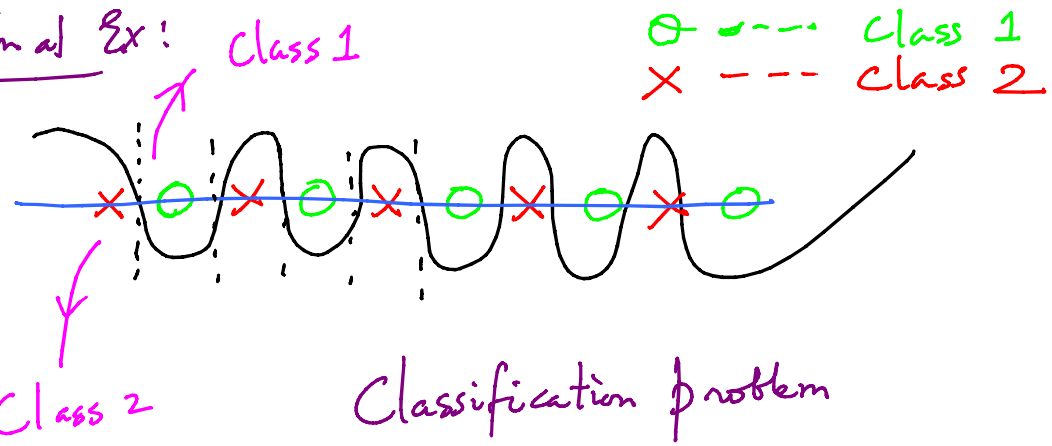
computation, communication and control !

Neural networks : Introduction

Neural network is a machine designed to model the way the brain performs a particular task/function of interest.
We will be focussing on 'learning'

Roots stem from 70 + years of knowledge and experience in signal processing (non-linear signal processing)

Motivational Ex:



Classification problem

Neural network capabilities

1) Non linearities : Special kind of distributed non-linearity (better generalization / abstraction ~ reality)

2) Input / Output mappings : Non parametric statistical inference tool

3) Adaptivity : Adjust quickly to changing / non-stationary environment

4) Stability - Plasticity Dilemma (Grossberg)

a) Long time constants are reqd to ignore spurious disturbances.

b) Short time constants to respond to meaningful changes.

5) Evidential response

What patterns to select? Confidence about the decision?

Reject ambigious patterns.

6) Contextual Information

Knowledge representation by the 'structure' and 'activation' of the neuron.

7) Fault Tolerance

The learning system must not be vulnerable to catastrophic failures.

Performance must degrade gracefully under worst case/ spurious inputs.

E.g.; Do humans/living beings code non-linearly?

8) VLSI Implementation

Massively parallel & distributed structures for Computation purposes

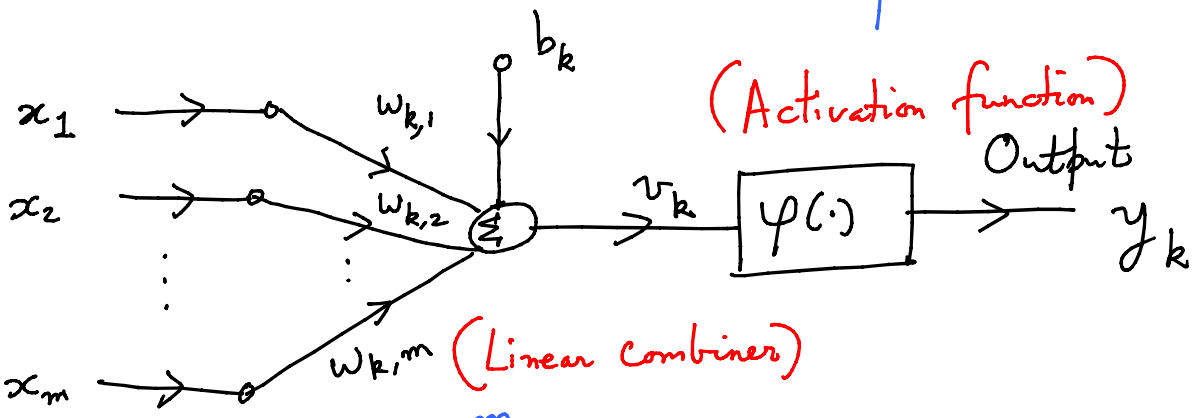
9) NNs are basically information processors

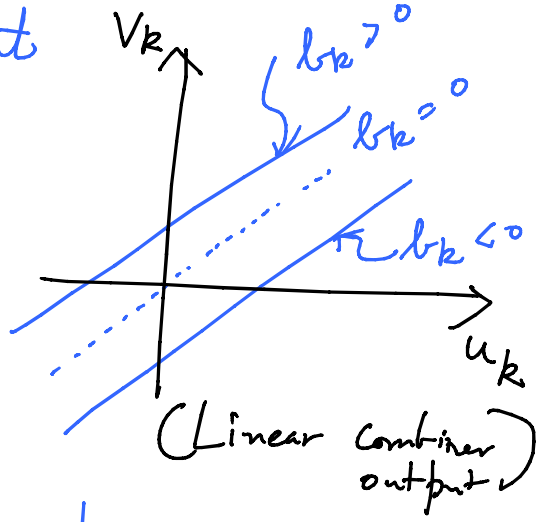
10) Neuro biological analogy

(Sensing, Sig. Processing, Coding, Communication, Inference, Control ...)

Models of a neuron

A neuron is an information processing element





$$u_k = \sum_{j=1}^{m} w_{k,j}\, x_j \quad ; \quad v_k = u_k + b_k$$

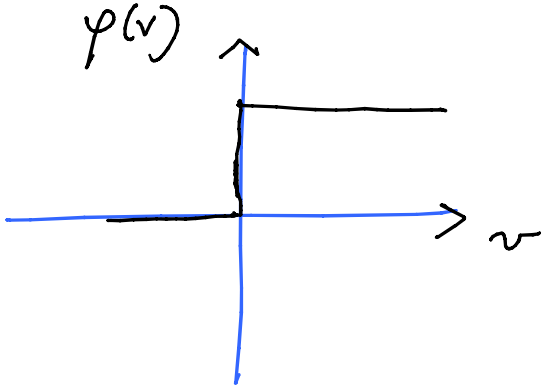
$$y_k = \varphi(u_k + b_k) = \varphi(v_k)$$

Types of activation functions

1) Threshold function (Mc Culloch - Pitts model)

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

$$y_k = \begin{cases} 1 & v_k \geq 0 \\ 0 & \text{else} \end{cases}$$



2) **Sigmoid function**

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

(Logistic function)

3) **Signum function**

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

Stochastic neuronal models

Let the 'decision' for a neuron to fire be probabilistic.

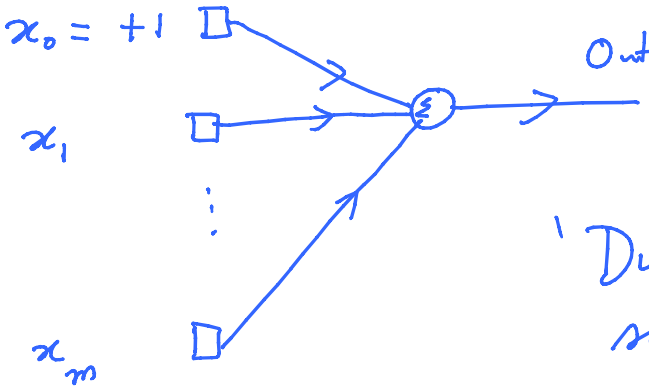
Let 'x' be the state of the neuron

$$x = \begin{cases} +1 & \phi \\ 0 & 1-\phi \end{cases}$$

$$\phi = f(v, T) = \frac{1}{1+\exp(-v/T)}$$

When $T \to 0 \Rightarrow$ Noiseless case $\Rightarrow$ Mc Culloch Pitts model, 'deterministic'

Neural networks as digraphs



'Directed' edges from the source to the destination

Rule 1: A signal flows along a link only in the direction defined by the arrow on the link

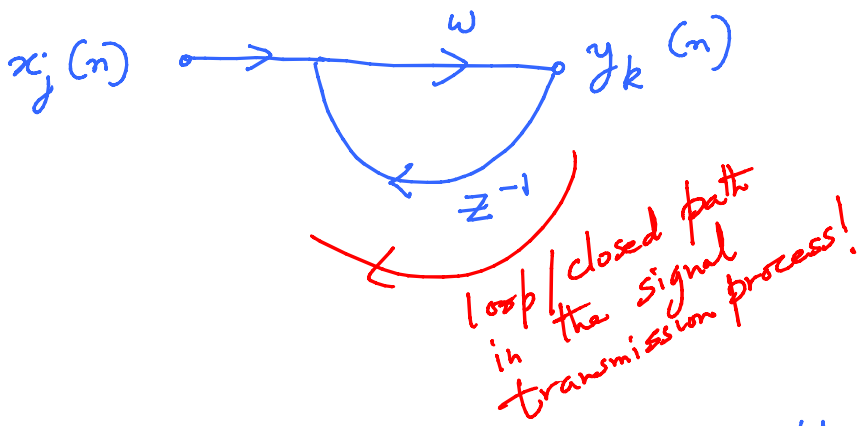
Synaptic Links: node signal x_j $\times$ Synaptic wt w_{kj} to produce y_k

Activation Links: Behavior governed by the non-linear I/o relationship

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

Rule 3 : The signal at a node is transmitted to each outgoing link originating from that node and 'transmission' is entirely independent of the transfer functions of the outgoing links

Feedback



Unit delay system

$$Z(y_k(n)) = \frac{w}{1 - wz^{-1}} Z(x_j(n))$$

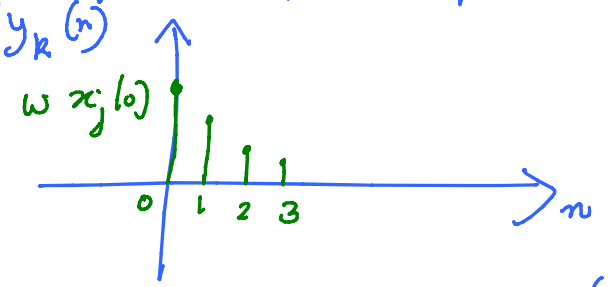
$$= w \sum_{l=0}^{\infty} w^l z^{-l} \; Z(x_j(n))$$

$$y_k(n) = \sum_{l=0}^{\infty} w^{l+1} \underbrace{x_j(n-l)}$$

Sample of the input signal delayed by 'l' discrete time steps

Let us Consider the following cases

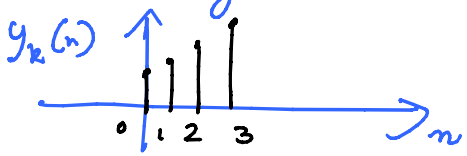
1) If $|w| < 1$, o/p signal $y_k(n)$ is exponentially convergents.

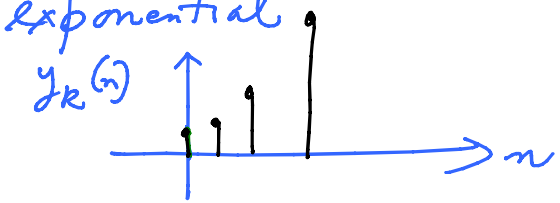


2) If $|w| \geq 1$, $y_k(n)$ is divergent & the system is UNSTABLE.

If $|w| = 1$, divergence is linear

$|w| > 1$, divergence is exponential





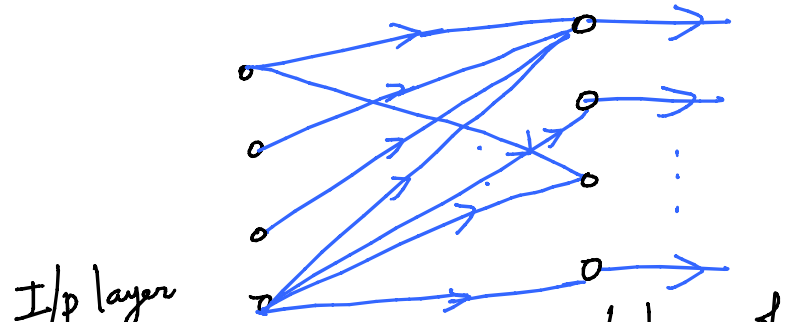
Suppose we let the system to have finite memory

$$y_k(n) \approx w\, x_j(n) + w^2\, x_j(n-1) + \dots + w^N\, x_j(n-N+1)$$

Neural elements have feed back!

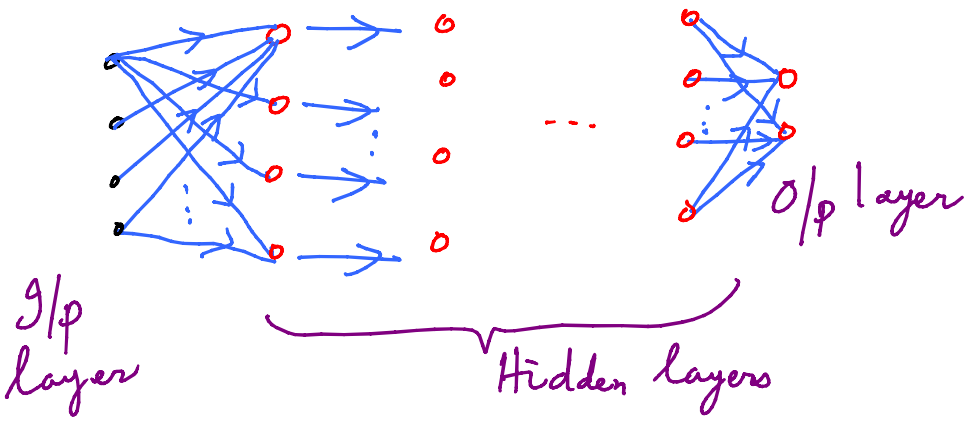
Network Architectures

1) Single-layered feed forward network



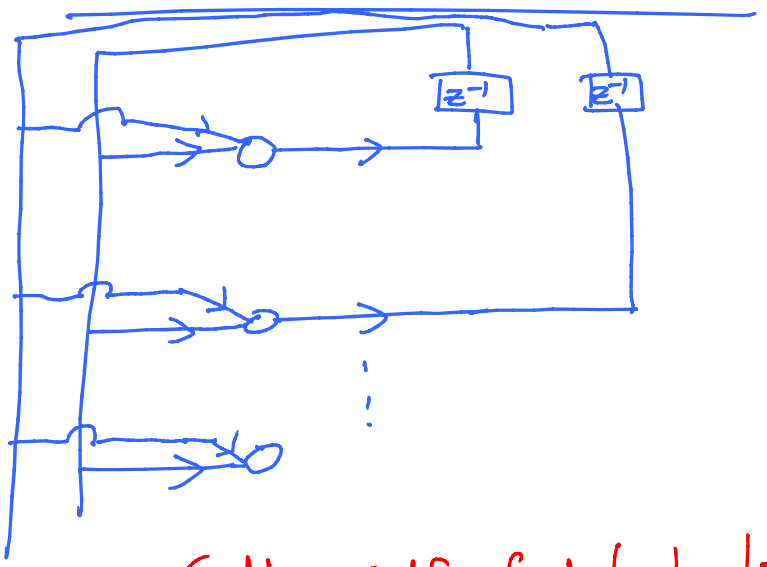
(No computations)

2) Multilayered feed forward n/ws

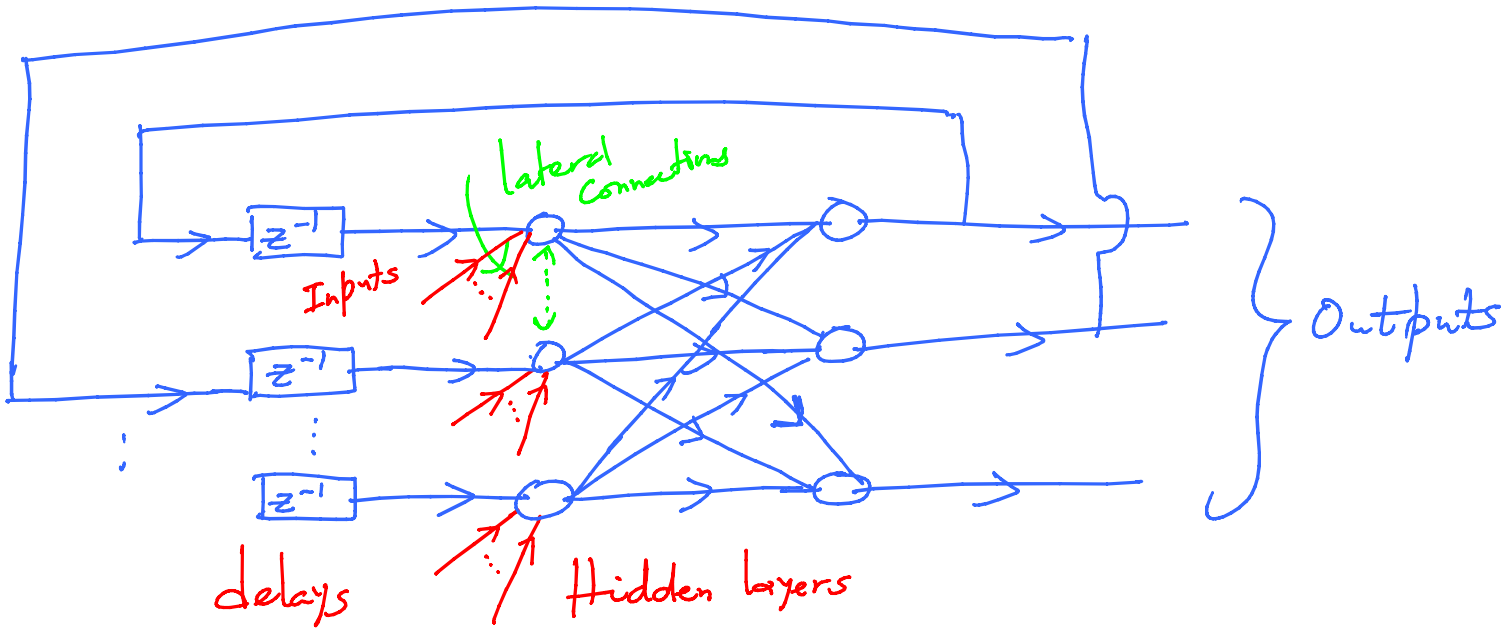


NOTE: The multilayered feed forward n/w can be (a) fully connected (b) partially connected

3) Recurrent networks



(No self feed back loops)



Sketch of a recurrent network

Knowledge Representation

Stored information or models used by a person / machine to interpret, predict and appropriately present to the outside world.

2 facts

a) Prior information (Known state)

b) Observations to probe and learn new information

What is learnt?

a) Biases / weights of the n/w i.e., parameters of the n/w.

How is it learnt?

Through a measure / metric that guides the system to align best to the data

Rules for knowledge representation

Rule 1 : Similar inputs (i.e., patterns drawn from similar classes) should have similar representations inside the n/w & must belong to the same class.

Plenty of measures :

$$\underline{x}_i = [x_{i,1} \quad x_{i,2} \quad \cdots \quad x_{i,m}]^T$$

$$\underline{x}_j = [x_{j,1} \quad x_{j,2} \quad \cdots \quad x_{j,m}]^T$$

$$d(\underline{x}_i, \underline{x}_j) = \| \underline{x}_i - \underline{x}_j \|$$

$$= \left(\sum_{k=1}^{m} (x_{i,k} - x_{j,k})^2 \right)^{\frac{1}{2}}$$

Another measure (Inner Product)

$$\langle \underline{x}_i, \underline{x}_j \rangle = \underline{x}_i^T \underline{x}_j$$

$$= \sum_{k=1}^{m} x_{i,k} \, x_{j,k}$$

Smaller the Euclidean distance $\Longrightarrow$ Larger the inner product

Suppose $\|\underline{x}_i\| = \|\underline{x}_j\| = 1$

$$d^2(\underline{x}_i, \underline{x}_j) = (\underline{x}_i - \underline{x}_j)^T (\underline{x}_i - \underline{x}_j)$$

$$= \|\underline{x}_i\|^2 + \|\underline{x}_j\|^2 - 2\,\underline{x}_i^T \underline{x}_j$$

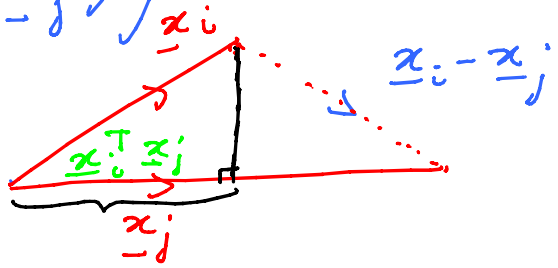
$$= 1 + 1 - 2\,\underline{x}_i^T \underline{x}_j$$

$$= 2 - 2\,\underline{x}_i^T \underline{x}_j$$

$$= 2\left(1 - \langle \underline{x}_i, \underline{x}_j \rangle\right)$$

$d \uparrow$ I.P. $\downarrow$

$d \downarrow$ I.P. $\uparrow$



For stochastic inputs

$$d^2_{i,j} = (\underline{x}_i - \underline{\mu}_i)^T C^{-1} (\underline{x}_j - \underline{\mu}_j)$$

where $\underline{\mu}_i = E(\underline{x}_i)$ $\quad \underline{\mu}_j = E(\underline{x}_j)$

$$C = E\left((\underline{x}_i - \underline{\mu}_i)(\underline{x}_i - \underline{\mu}_i)^T\right)$$

If $\underline{x}_i$ & $\underline{x}_j$ $\in$ same class

$$\underline{\mu}_i = \underline{\mu}_j = \underline{\mu}$$

$$d^2_{i,j} = (\underline{x}_i - \mu)^T C^{-1} (\underline{x}_j - \mu)$$

Rule 2 : Items to be categorized as separate classes should be given widely different representations within the n/w.

Rule 3 : If a particular feature is important, there should be large # neurons to represent it.

Rule 4 : Prior information & invariances should be built into the design of the n/w.
Help simplify the design of the learning machine & improve the accuracy.

Rule 4 is important for the following reasons.

1) Biological systems such as the visual/auditory n/w are known to be highly specialized.

2) If we have a NN with a highly specialized structure, we need smaller # of free parameters to learn this structure as opposed to a general n/w which could be fully connected.

$\Rightarrow$ Smaller data set for training $\Rightarrow$ faster learning

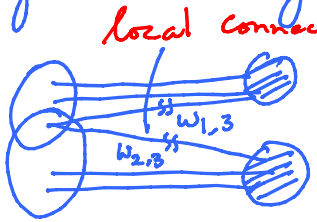
Rule 5 : One has to build connections within the n/w so as to accelerate the rate of information transmission.

Building prior information into neural network

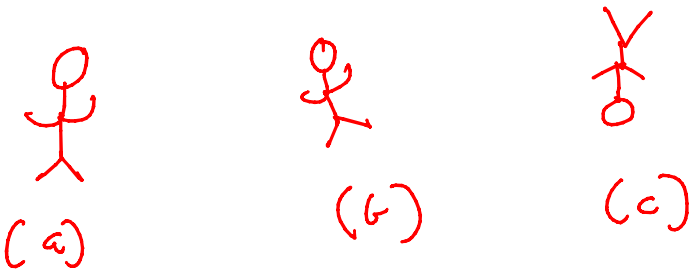
Some adhoc procedures

1) Restricting the network architecture through local connections.

2) Constraining the choice of synaptic weights through weight sharing



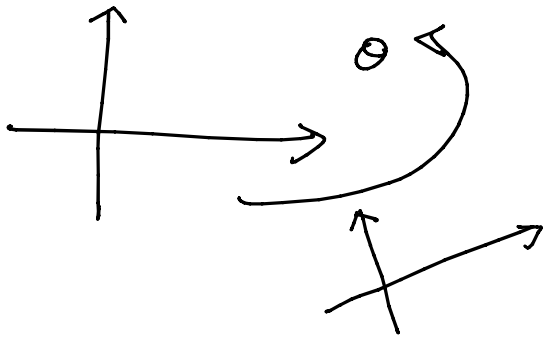
Building invariances into NN design



We perceive (a) – (c) to be the same

How can a neural network take care of this?

$$R = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$



(a) Invariance by structure

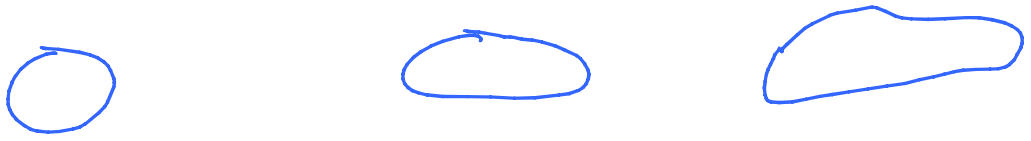
Transformed versions of the same input are forced to produce the same o/p.

Suppose w_{ji} is a synaptic wt. of neuron 'j' connected to pixel 'i'.

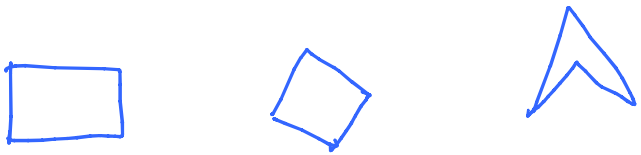
If $w_{ji} = w_{jk}$ $\forall$ pixels 'i' and 'k' lying equidistant from the center of the image

$\Longrightarrow$ Rotational invariance

(b) Invariance by training



Class (a)



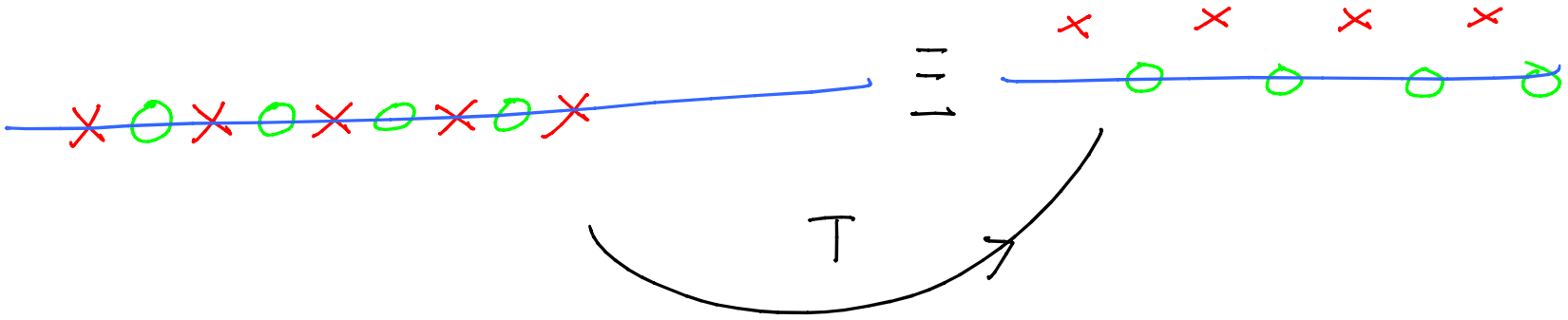
Class (b)

Can we learn different objects / aspects of the same object under transformations?

Invariant feature space

Invariant to transformations of the i/p space.

i.e., Extract those "features" that capture the information content of an input data set



Advantages

(a) # features applied to the n/w may be reduced

(b) Requirements on the n/w design is relaxed.