ONLY GOD 1402/ 2023

m a n s o o r i @shirazu.ac.ir

Neural Networks & Deep Learning

Introduction to Neural Networks

CSE & IT Department
ECE School
Shiraz University

Biological Inspiration

mansoori@shirazu.ac.ir

- Brain is a highly complex, nonlinear, and parallel computer
    - Simple processing units called neurons
    - Cycle times in milliseconds
    - Massive number of neurons (~ 10^{11} in humans)
    - Massive interconnection (~ 60^{12} connections)

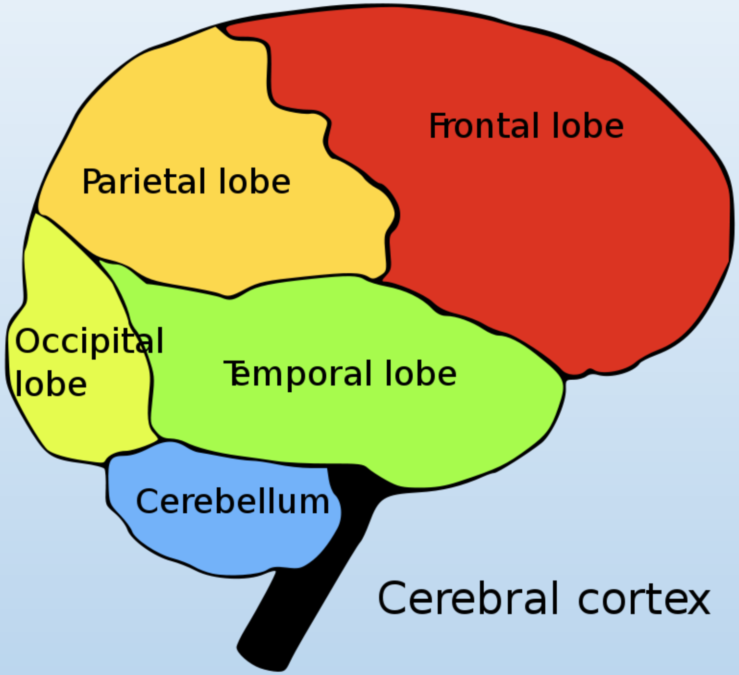
- Brain can perform certain tasks (pattern recognition, perception) many times faster than fastest digital computers

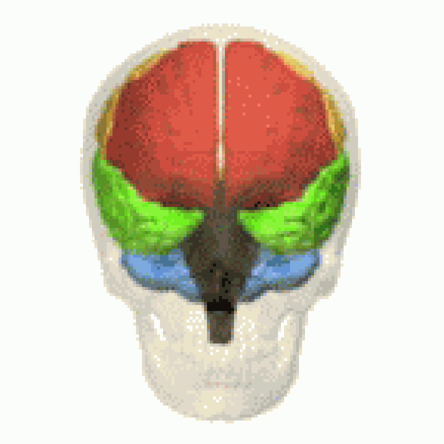
mansoori@shirazu.ac.ir

Natural Intelligence

Structure of human brain

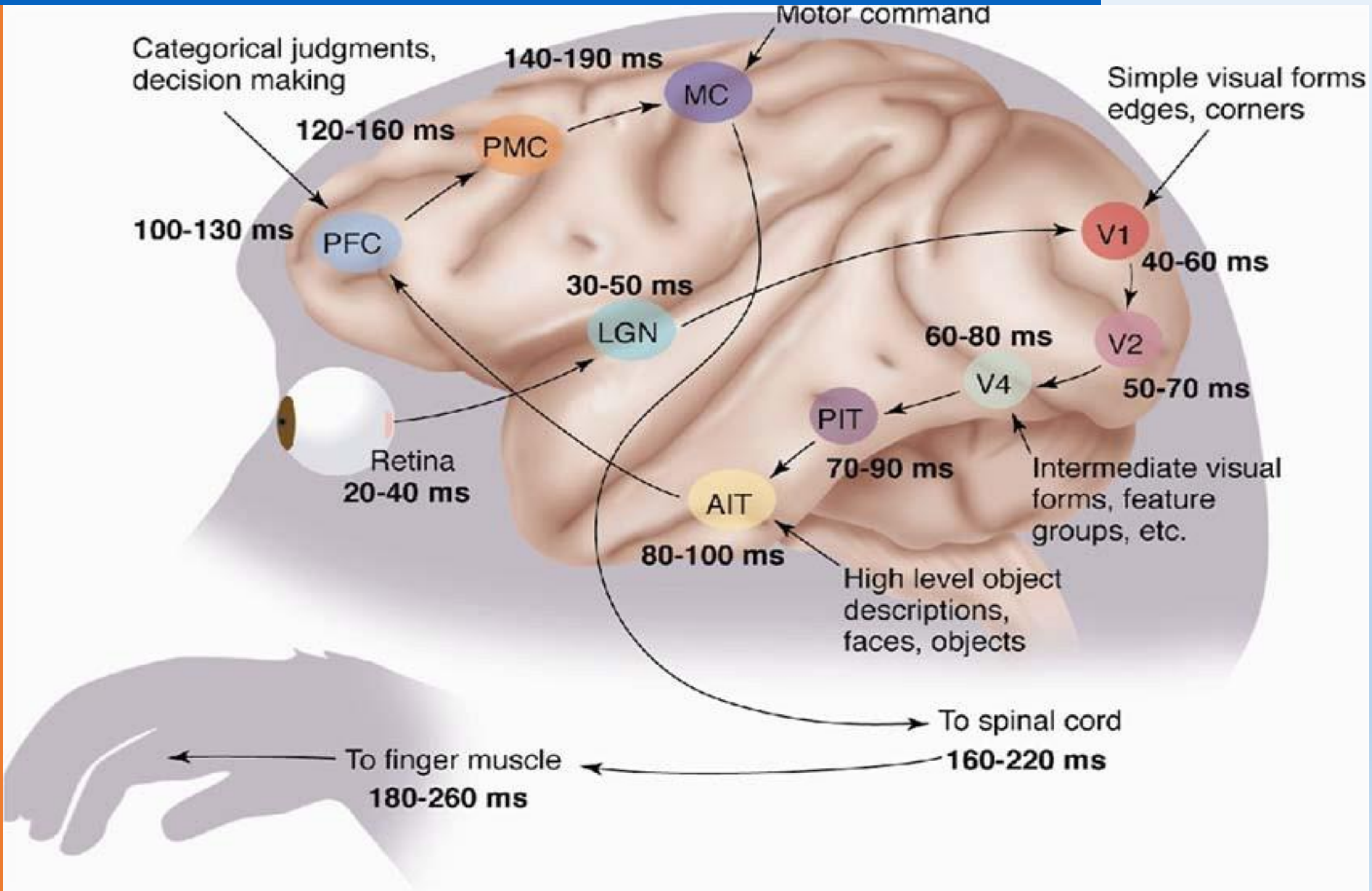
- Frontal lobe
  - Reward, attention, planning, short-term memory, motivation
- Parietal lobe
  - Sensory inputs from skin (touch, temperature, pain)
- Occipital lobe
  - Visual processing center
- Temporal lobe
  - Visual memories, language comprehension, emotion





From Receptor to Effector (~1s)

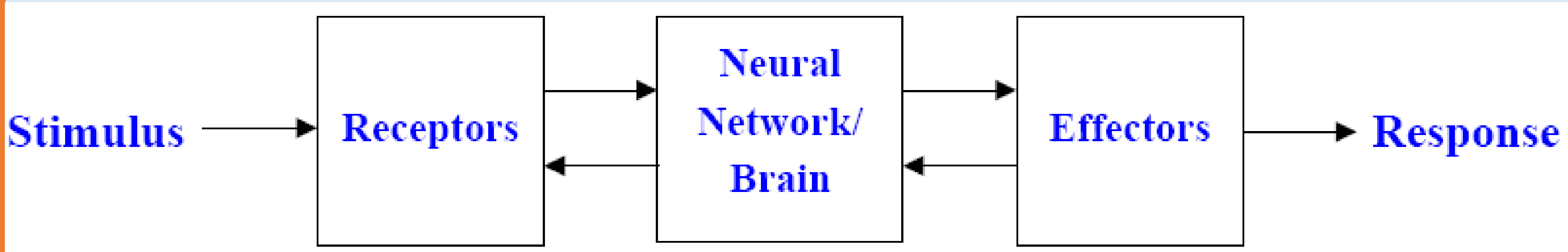
mansoori@shirazu.ac.ir



mansoori@shirazu.ac.ir

Human Nervous System Operation

- May be broken down into three stages:



- Receptors
  - Collect information from the environment (e.g. photons on the retina)
  - Convert stimuli from human body or environment into electrical pulses as information conveyer to brain

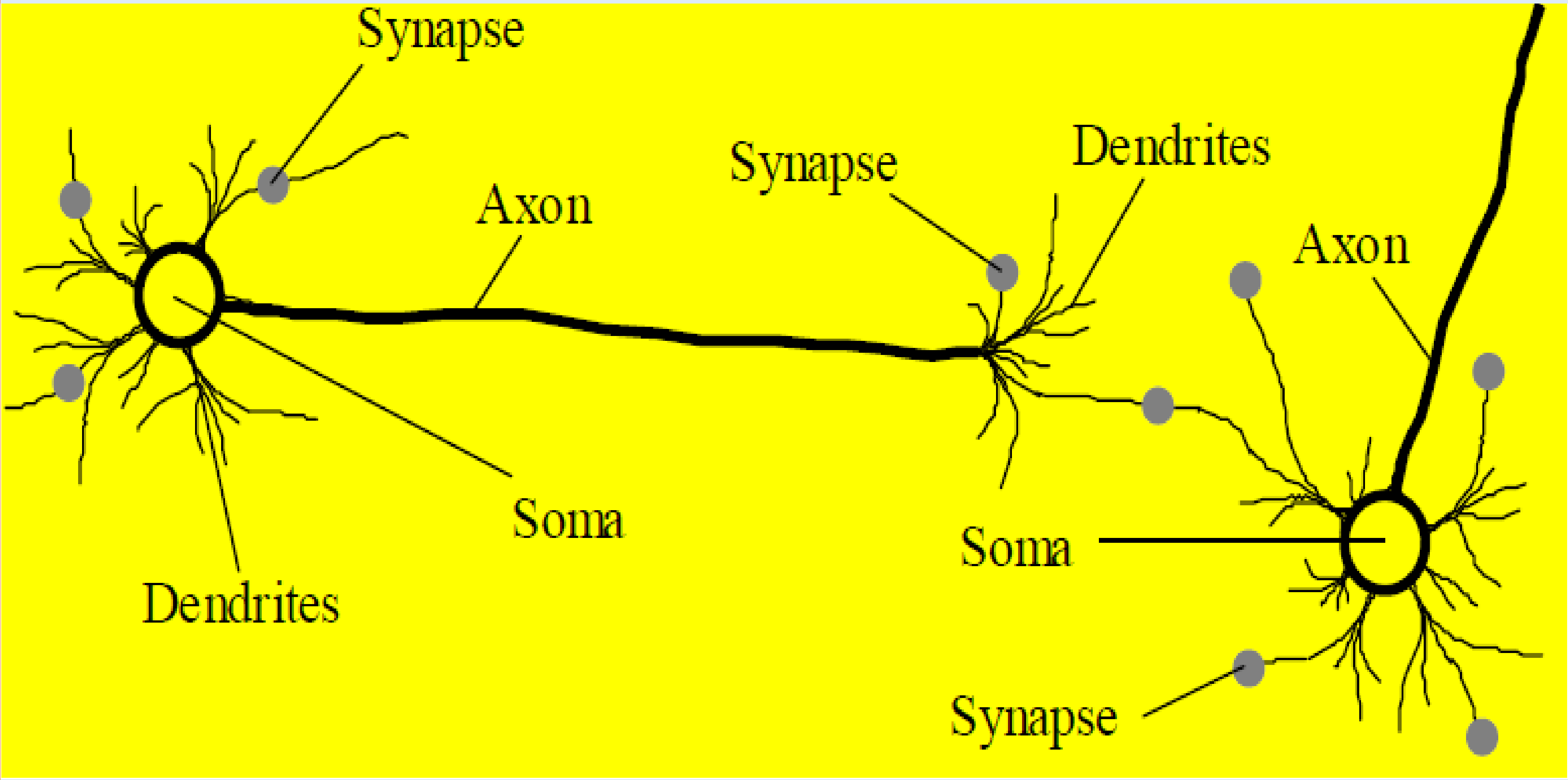
Organization of Human Nervous System

mansoori@shirazu.ac.ir

- Neural network
  - Represents the brain which receives information and makes appropriate decisions
- Effectors
  - Convert electrical impulses generated by the neural network into responses as system outputs
  - Generate interactions with the environment (e.g. activate muscles)
- Arrows
  - Represent the flow of information/activation (feed-forward and feed-back)

mansoori@shirazu.ac.ir

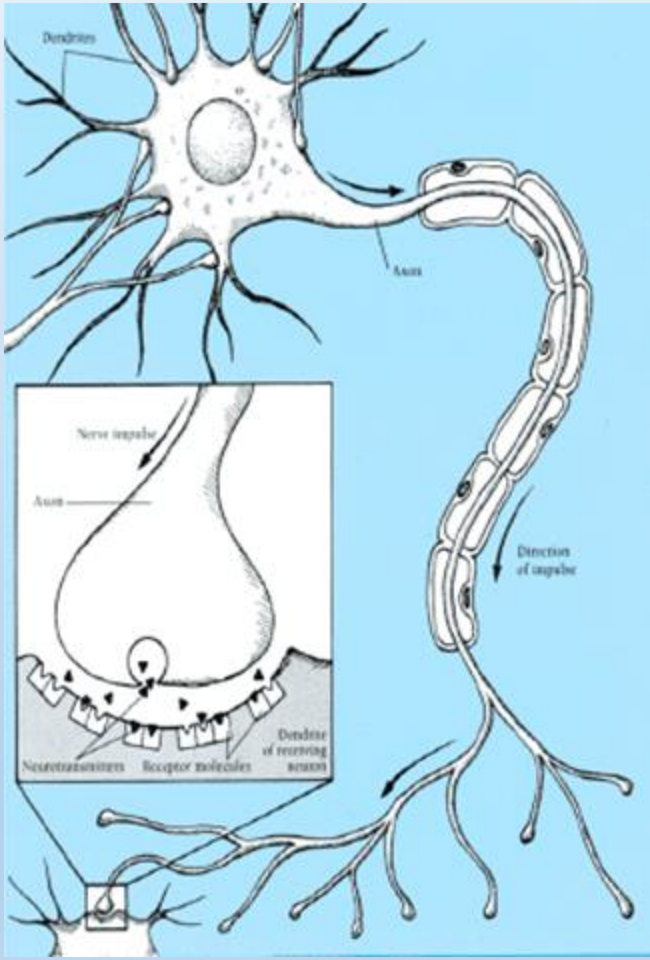
Biological Neuron



mansoori@shirazu.ac.ir

Biological Neuron

- Dendrites
  - Receptive zones that receive activation from other neurons
- Cell body (soma)
  - Processes incoming activations and converts them into output activations
- Axons
  - Transmission lines that send activation to other neurons
- Synapses
  - Allow weighted transmission of signals (using neurotransmitters) between axons and dendrites



mansoori@shirazu.ac.ir

Brain: Amazingly Efficient Computer

~ 10^{11} neurons

~ 10^{4} synapses per neuron

~ 10 spikes go through each synapse per second

~ 10^{16} operations per second

~ 25 Watts (Very efficient)

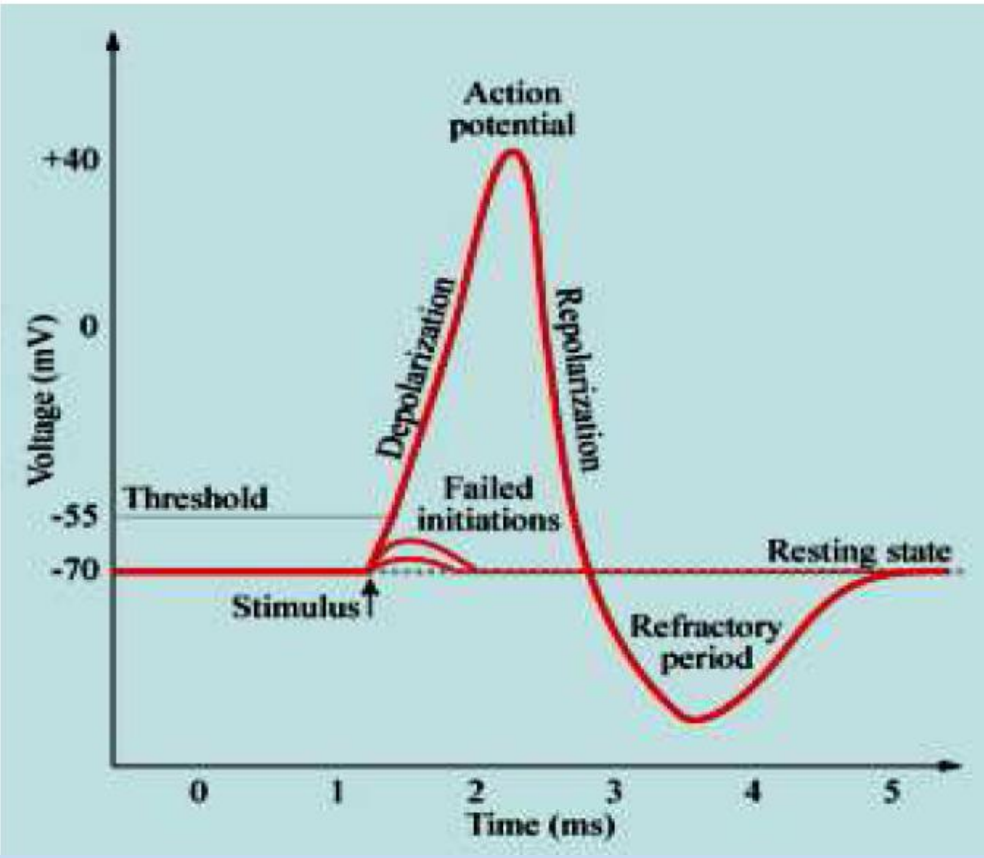
~ 1.4 Kg, 1.7 liters

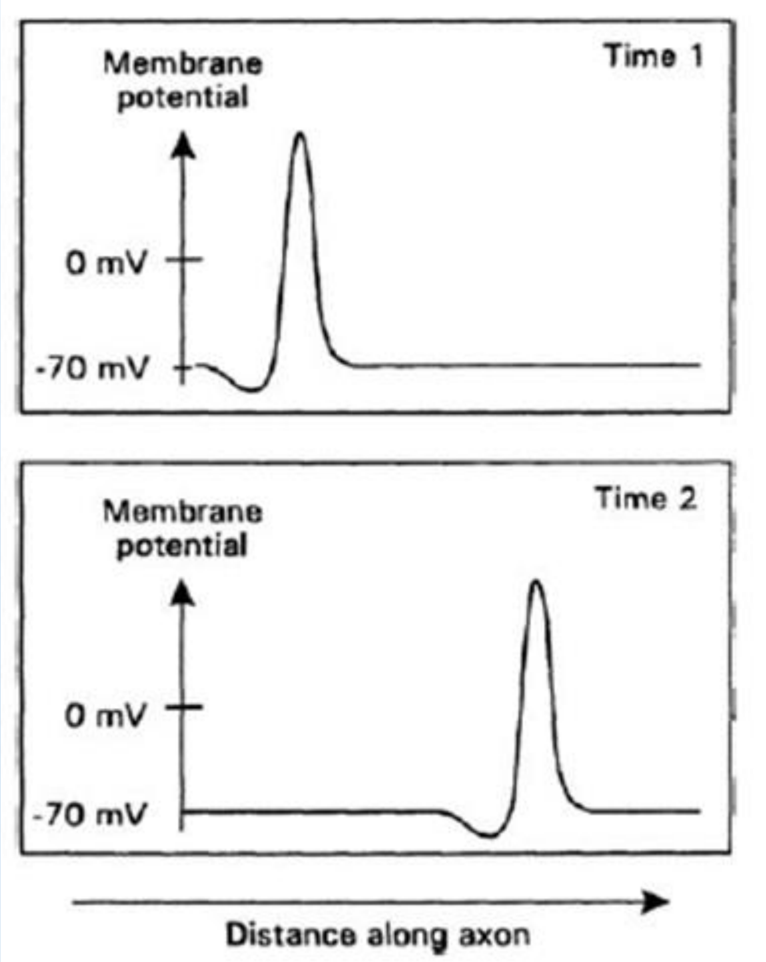
~ 2500 cm^2 (Unfolded cortex)

Action of Biological Neuron

mansoori@shirazu.ac.ir

- Majority of neurons encode their outputs/activations as a series of brief electrical pulses (spikes or action potentials)





mansoori@shirazu.ac.ir

Firing of Neuron

- Behavior is binary (a neuron either fires or not)
- Neurons don't fire if their accumulated activity stay below threshold
- If activity is above threshold, a neuron fires (produces a spike)
- The firing frequency increases with accumulated activity until max. firing frequency reached
- The firing frequency is limited by refractory period of about 1-10 ms

Man vs. Machine (Hardware)

mansoori@shirazu.ac.ir

Features	Human brain	Von Neumann computer
# elements	10^{10} - 10^{12} simple neurons	10^7 - 10^8 transistors 10^4 complex processors
# connections/element	massive	little or no
Switching frequency	slow (10^3 Hz)	fast (10^9 - 10^{10} Hz)
Energy/operation/sec.	10^{-16} Joule	10^{-6} Joule
Power consumption	10 Watt	100 - 500 Watt
Structure	dynamic	static
Reliability of elements	low	reasonable
Reliability of system	high	reasonable
Decision speed/power	high	reasonable

Man vs. Machine (Software)

mansoori@shirazu.ac.ir

Features	Human brain	Digital computer
Data representation	analog	digital
Memory localization	distributed	localized
Control	distributed	localized
Processing	parallel	sequential
Skill acquisition	adaptive-learnable	pre-programming
Fault/error tolerance	fault tolerant	intolerant to errors
Structure	trained	designed
Activation	nonlinear	linear

What are NNs?

mansoori@shirazu.ac.ir

- Artificial neurons are crude approximations of neurons found in brains
  - Physical devices
  - Mathematical constructs
- Artificial NNs (ANN) are networks of artificial neurons as crude approximations to parts of biological brains

Conceptual Definition of NNs

mansoori@shirazu.ac.ir

- An ANN
  - Is a machine, designed to model the brain how performs a particular task or function of interest
  - Is either implemented in hardware or simulated in software
  - Mimics the brain or nervous system, in senses:
    - Structure (simple processing units, massive interconnection, …)
    - Functionality (learning, adaptability, fault tolerance, …)

Pragmatic Definition of NNs

mansoori@shirazu.ac.ir

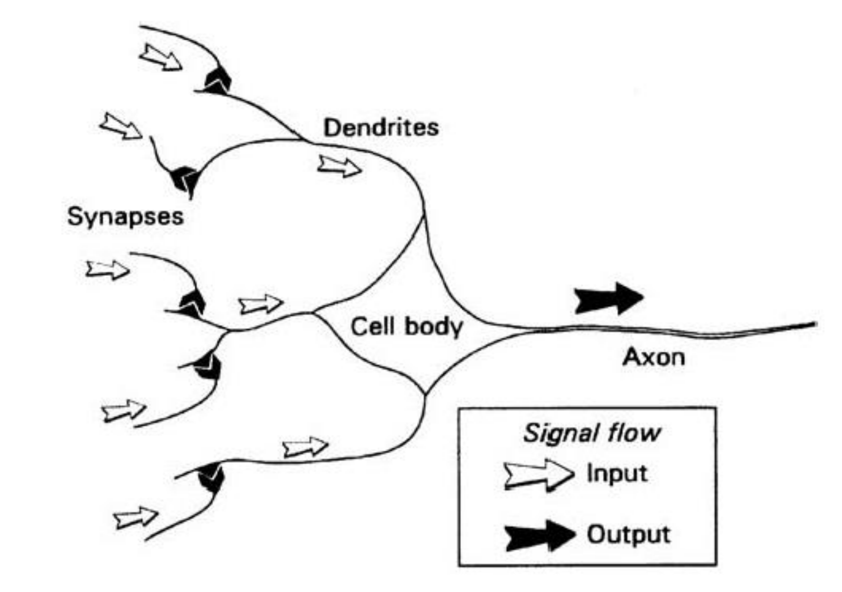
- An ANN
  - Is a massively connected parallel computational system
  - Made up of many simple processing units
  - Has propensity of storing experimental knowledge
  - Can perform tasks analogously to biological brains
  - Resembles the brain in two respects:
    - Knowledge is acquired by the network from the environment through a learning process
    - Inter-neuron connection strengths are used to store the acquired knowledge

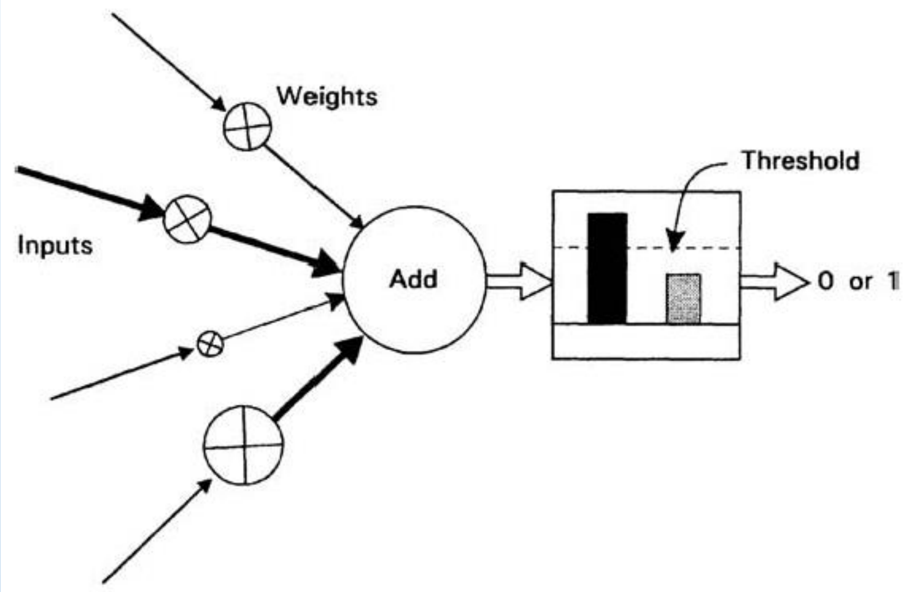
Why NNs?

- They are extremely powerful computational devices
- Massive parallelism makes them very efficient
- They can learn from training data and generalize to new situations
- They are particularly fault tolerant (as "graceful degradation" in biological systems)
- They are robust and noise tolerant
- They can do anything a symbolic/logic system can do, and more

Mathematical Model of a Neuron

mansoori@shirazu.ac.ir

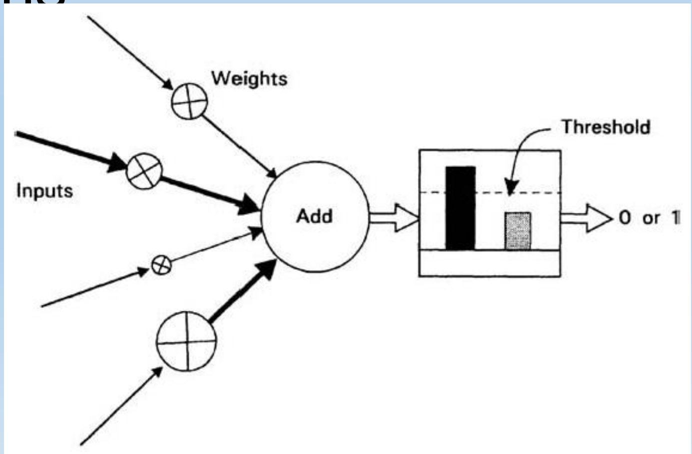




Biological Neural Network	*Artificial Neural Network*
Soma	Neuron
Dendrite	Input
Axon	Output
Synapse	Weight

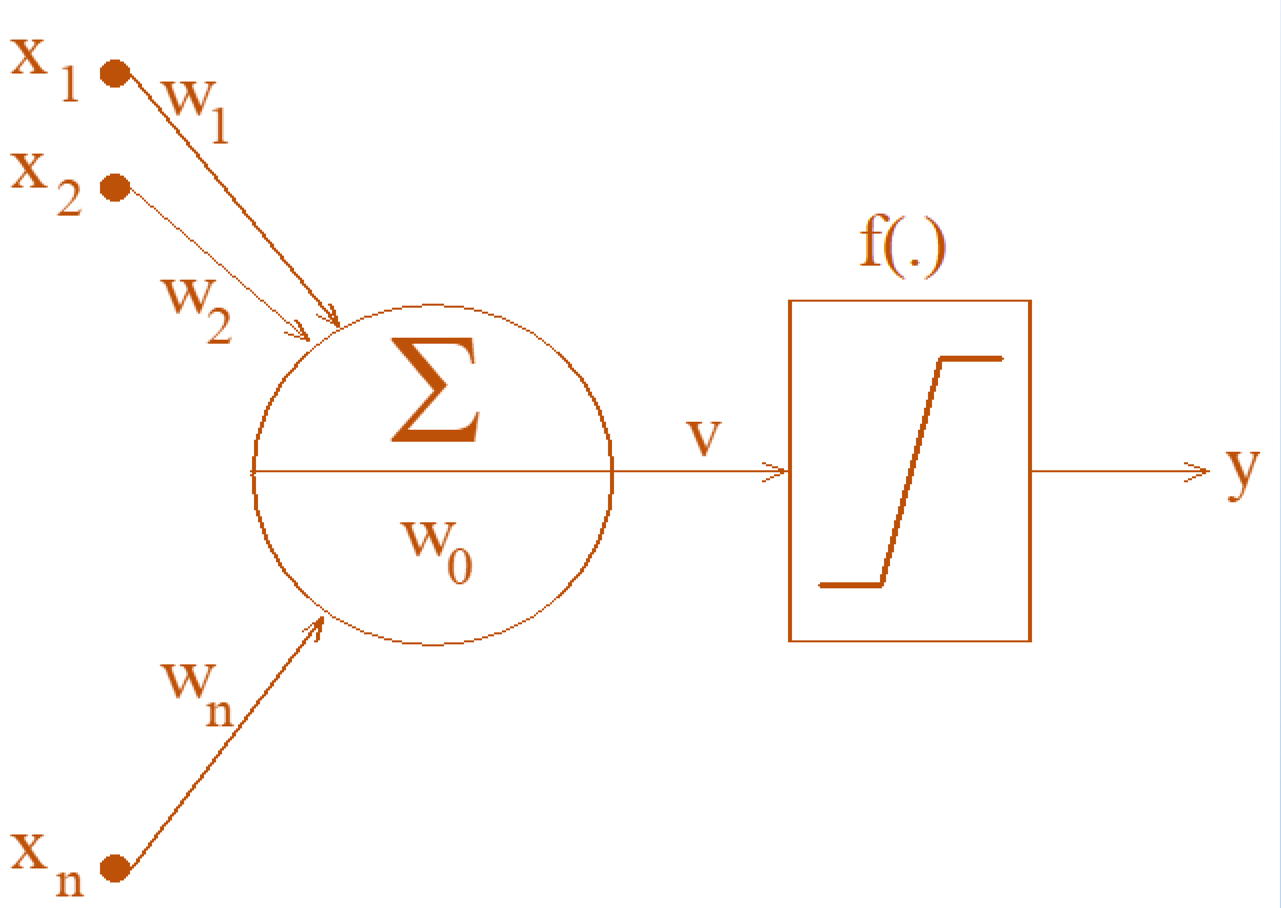
Characteristic of Neuron

- A neuron can receive many inputs
- Inputs may be modified by synaptic weights at receiving dendrites
- A neuron sums its weighted inputs
- An activation function limits amplitude of output to {0, 1}
- A neuron can transmit an output signal
- Output can go to many other neurons



Artificial Neuron

mansoori@shirazu.ac.ir

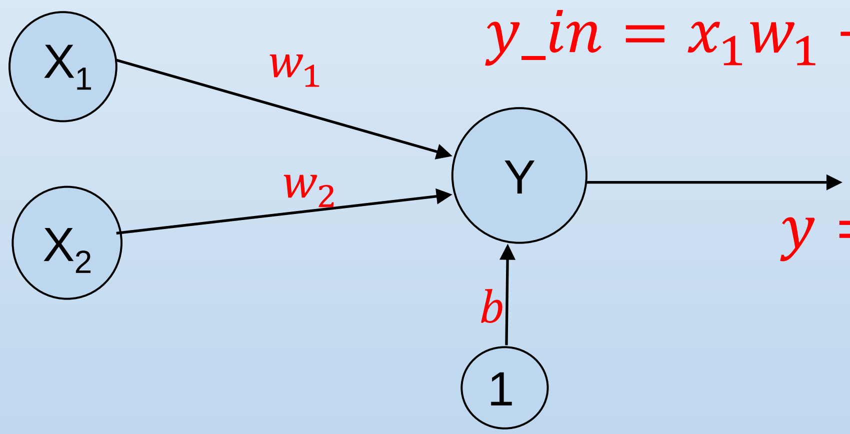


Mathematically, $v = w_0 + \sum_{i=1}^{n} x_i w_i \implies y = f(v)$

$f(.)$: *threshold function*

mansoori@shirazu.ac.ir

Artificial Neural Network



$$y\_in = x_1 w_1 + x_2 w_2 + b$$

$$y = f(y\_in)$$

ANN Features

mansoori@shirazu.ac.ir

- Neurons act nonlinearly
- Information processing is local
- Memory is distributed
- The dendrite weights learn through experience
- The weights may be inhibitory or excitatory
- Neurons can generalize novel input stimuli
- Neurons are fault tolerant and can sustain damage

ANN Categories

- Direction of information (signal) flow
  - Feed-forward network (no feed-back connections)
  - Recurrent network (with at least one feed-back connection)
- Number of layers
  - Single layer network
  - Single hidden-layer network
  - Multilayer network (shallow/deep network)
- Connectivity
  - Fully-connected network
  - Partially-connected network

mansoori@shirazu.ac.ir

ANN Categories

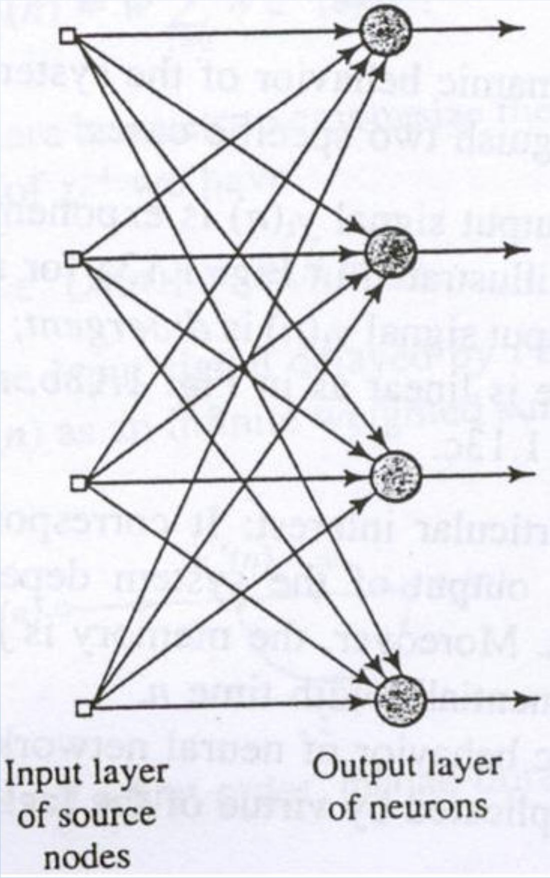
- Activation function
  - Threshold (binary, bipolar)
  - Linear (identity)
  - Nonlinear (sigmoid, radial-basis)
- Learning methodology
  - Supervised learning
  - Unsupervised learning
  - Reinforced learning
- Training method
  - Static
  - Dynamic

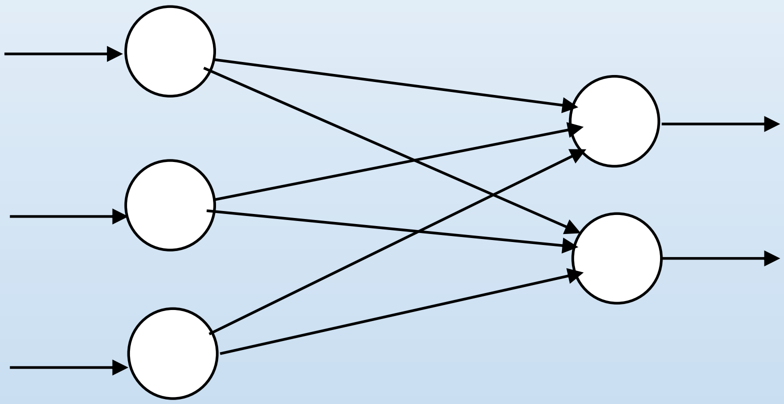
mansoori@shirazu.ac.ir

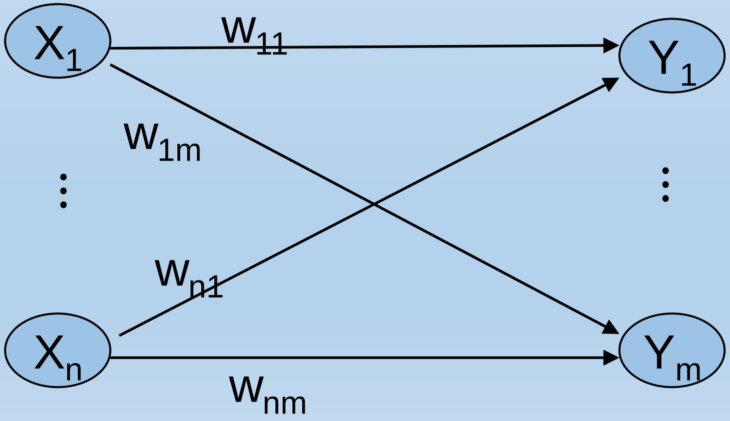
NN Design Decisions

- Architecture: pattern of connections between neurons
  - No. of layers, no. of neurons in each layer, no. of links
  - Connectivity
- Learning algorithm: method of determining the connection weights
  - Supervised, self-organizing, competitive
- Activation function: mapping of neurons' behavior
  - Threshold, linear, nonlinear

Single-layer Feed-forward NN

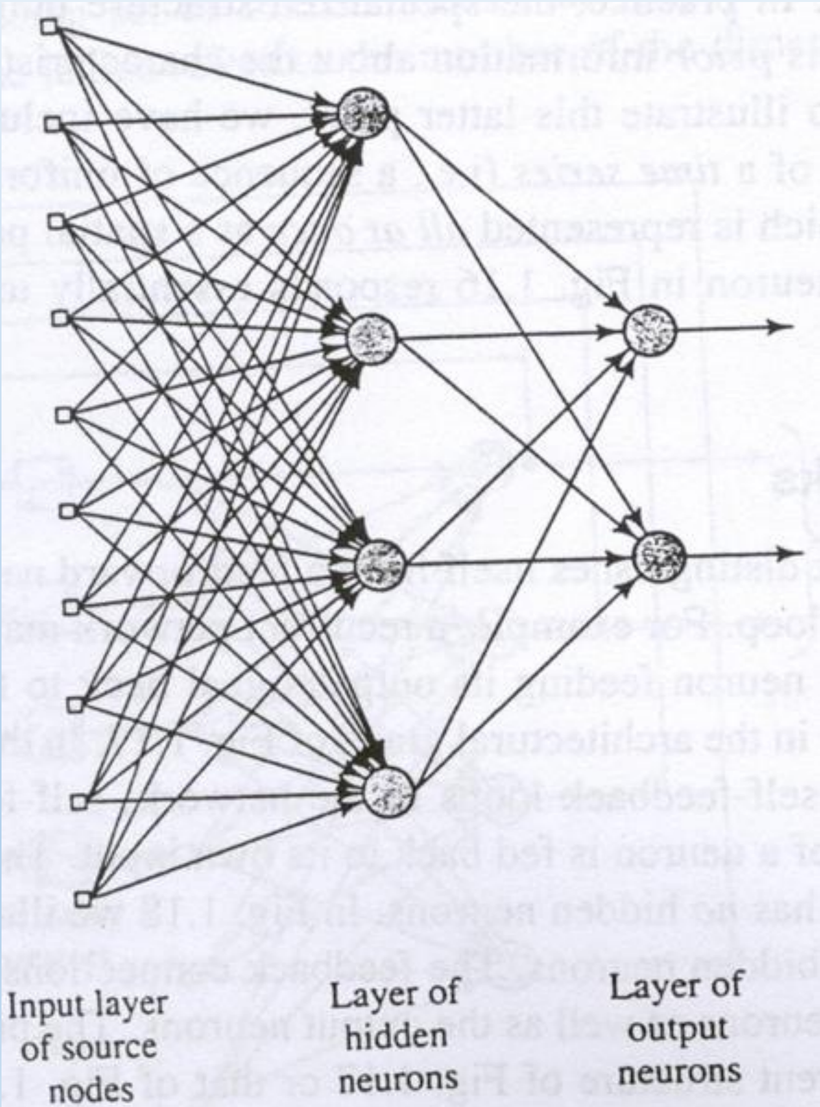


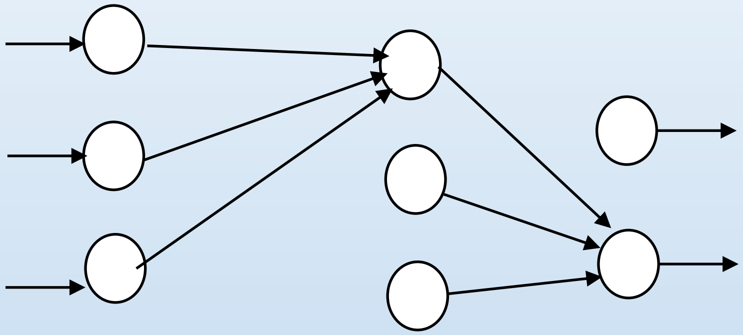


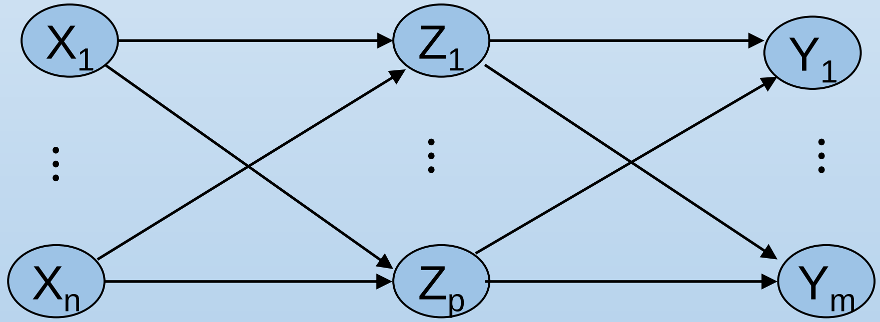


mansoori@shirazu.ac.ir

Multilayer Feed-forward NN



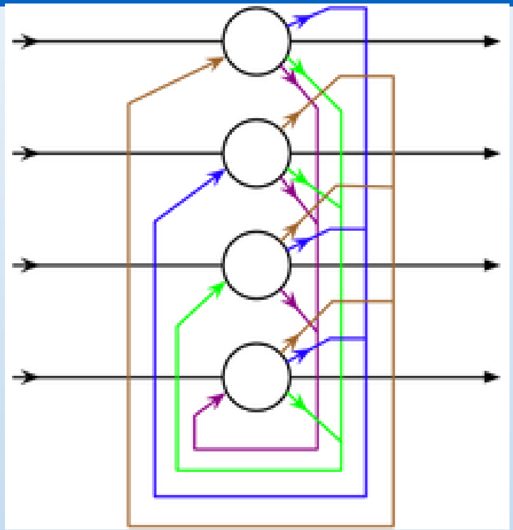


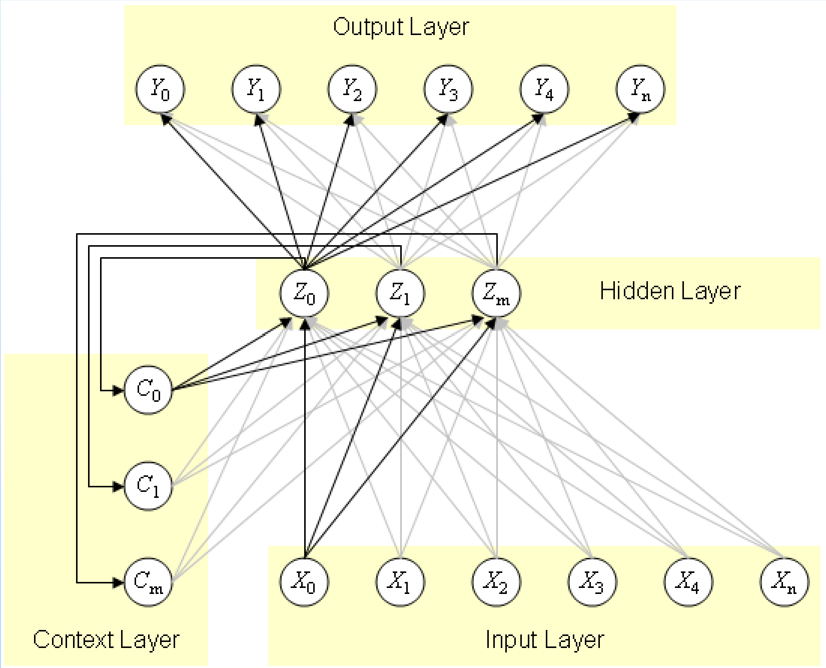


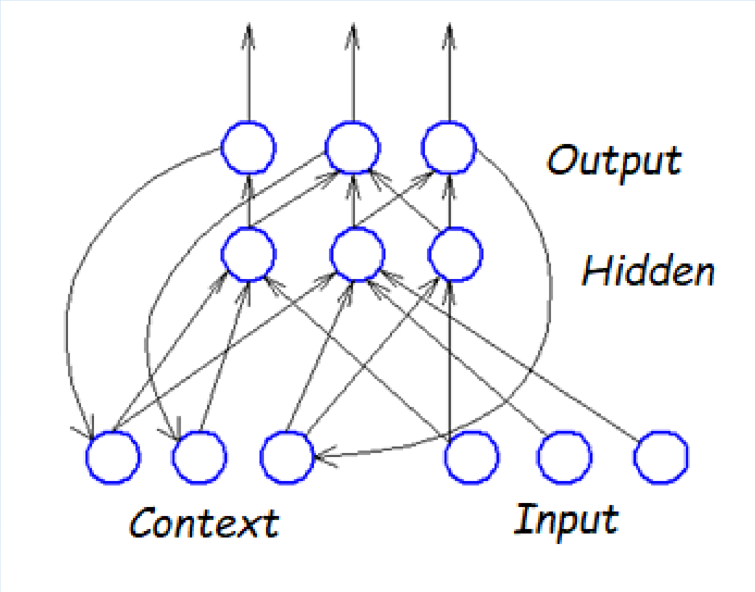
- More powerful
- Harder to train
- Open loop

Recurrent NN

mansoori@shirazu.ac.ir







- More powerful
- Harder to train
- Closed loop

Knowledge Representation in ANN

mansoori@shirazu.ac.ir

- What is knowledge?
  - Information or models used to interpret, predict and appropriately respond to outside world (environment)
- Quality of knowledge representation generally translates into quality of solution
  - Better representation means better solution
- Knowledge representation in NNs is not well-understood
  - There is little theory that relates a given weight to a particular piece of information
- Knowledge is encoded in free parameters of NN
  - Weights and thresholds

Knowledge Representation Rules

mansoori@shirazu.ac.ir

- Rule 1:
  - Similar inputs from similar classes should usually produce similar representations inside the network, and should, therefore, be classified as belonging to the same category
- Rule 2:
  - Inputs to be characterized as separate classes should be given widely different representations in the network
- Rule 3:
  - If a particular feature is important, then there should be a large number of neurons involved in representation of that item in network
- Rule 4:
  - Prior knowledge and invariances should be built into design of network, thus simplifying learning

Knowledge Representation Rules

mansoori@shirazu.ac.ir

- Advantages of building-in prior knowledge
  - Specialized structure
  - Benefits of specialized structure
    - Biologically plausible, less complication, fewer free parameters, faster training, fewer examples needed, better generalization
- How to build-in prior knowledge
  - No hard and fast rules
  - In general, use domain knowledge to reduce complexity of NN based on its performance characteristics
- Built-in Invariances
  - Invariance?
    - Fault tolerance
    - Immunity to transformations
  - Invariance by structure, training, feature space

NN Learning

mansoori@shirazu.ac.ir

- Learning:
  - To acquire and maintain knowledge of interest
- Knowledge of environment that will enable NN to achieve its goals
  - Prior information
  - Current information
- Knowledge can be built into NNs from input-output examples via learning using training algorithms
- The most powerful property of NNs: ability to learn and generalize from a set of training data

mansoori@shirazu.ac.ir

NN Learning

- NNs adapt the weights of connections between neurons so that final output activations are correct
- Three broad types of learning:
  - Supervised Learning (learning with a teacher)
  - Reinforcement learning (learning with environment feedback)
  - Unsupervised learning (learning with no help)
- Most of human/animal learning is unsupervised
- If intelligence was a cherry ice-cream cake,
  - unsupervised learning would be cake
  - supervised learning would be icing on cake
  - reinforcement learning would be cherry on cake

An Example

mansoori@shirazu.ac.ir

- Using NN for signature verification
- Prior knowledge
  - Architecture of NN
- Current knowledge
  - Input-output examples, being used for NN training
- Learning
  - Modification of free parameters (weights, thresholds, …)
- Generalization
  - Using the trained NN for predicting output for unseen input

Advantages of ANNs

mansoori@shirazu.ac.ir

- Efficiency
    - Inherent massively parallel
- Robustness
    - Can deal with incomplete and/or noisy data
- Fault tolerance
    - Still works when part of net fails
- User friendly
    - Learning instead of programming

mansoori@shirazu.ac.ir

Drawbacks of ANNs

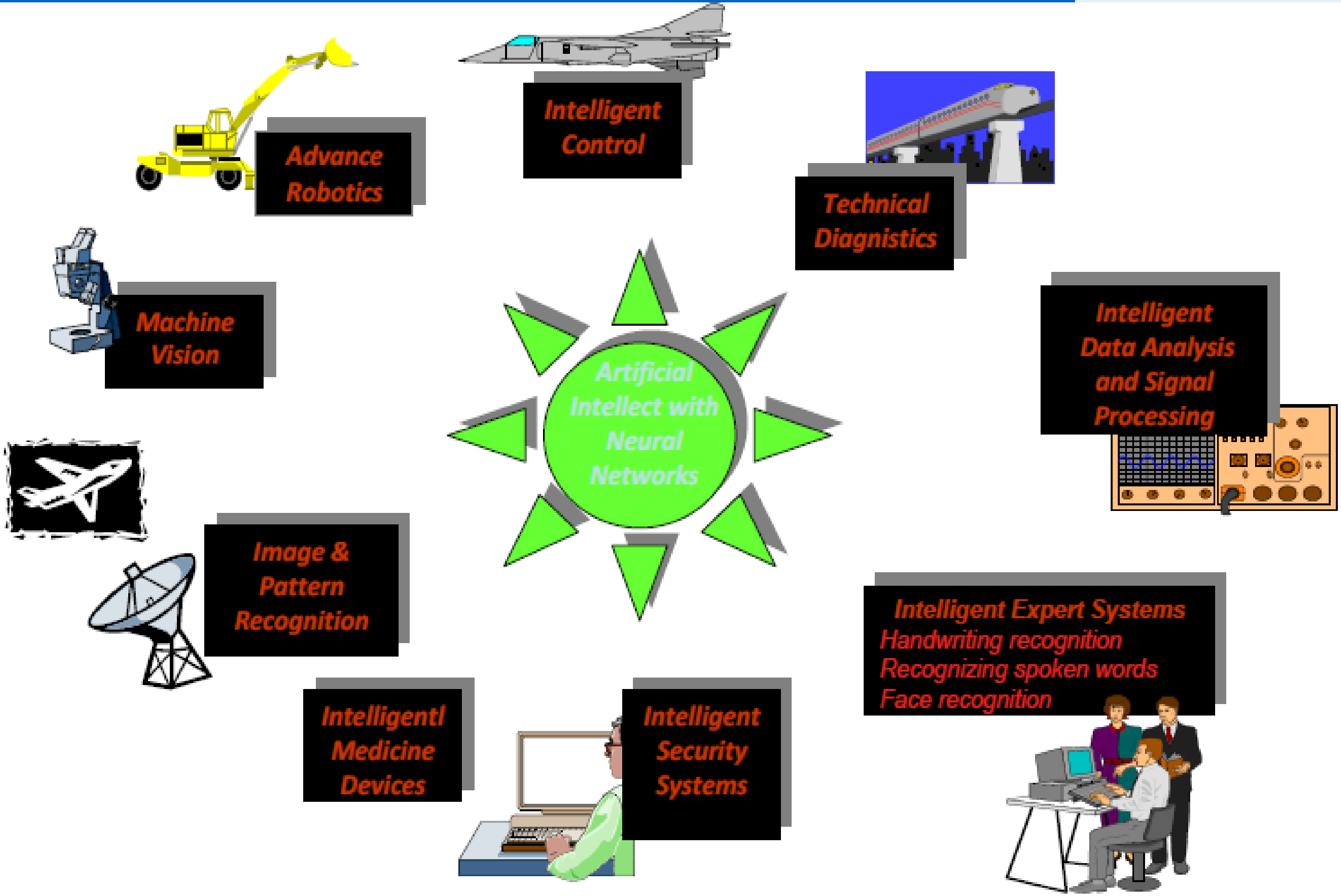
- Difficult to design
  - No clear design rules for arbitrary applications
- Hard or impossible to train
  - When training data are rare
- Difficult to assess internal operation (black box)
  - Difficult to find out what tasks are performed by different parts of the net
- Unpredictable
  - Difficult to estimate future network performance based on current behavior

Real-world NN applications

mansoori@shirazu.ac.ir

- Financial modeling (predicting stocks, currency exchange rates)
- Time series prediction (climate, weather, airline marketing)
- Computer games (intelligent agents, backgammon)
- Control systems (autonomous robots, microwave controllers)
- Pattern recognition (speech recognition, hand-writing recognition)
- Data analysis (data compression, data mining)
- Noise reduction (function approximation, ECG noise reduction)
- Bioinformatics (protein secondary structure, DNA sequencing)

Other Applications of NNs



History of NN

mansoori@shirazu.ac.ir

- 1943 McCulloch-Pitts neuron model
- 1949 Hebbian learning rule
- 1958 Single layer network, Perceptron
- 1960 Adaline
- 1969 Limitations of Perceptrons
- 1982 Kohonen nets, Associative memory,
  Hopfield net
- 1985 ART

History of NN

mansoori@shirazu.ac.ir

- 1986 Back-propagation learning algorithm for multi-layer Perceptron
- 1990s Radial basis function networks
- 2000 Ensembles of NNs, Cascaded Networks
- 2004 Deep Belief Networks
- 2010~ Convolutional Networks, Autoencoders, LSTMs

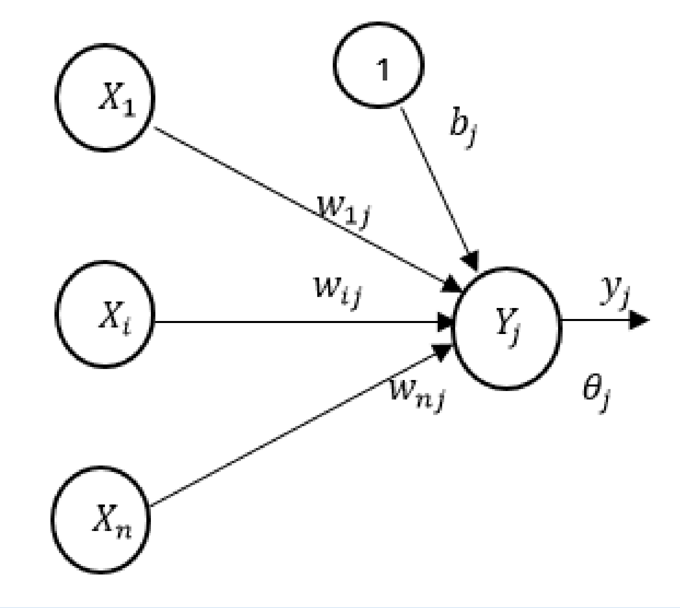
mansoori@shirazu.ac.ir

Some Useful Notations

x_i, y_j : outputs of units X_i and Y_j

w_{ij} : weight on connection between X_i and Y_j

b_j, θ_j : bias and threshold of neuron Y_j



$$W = \{w_{ij}, i = 1, \dots, n \quad j = 1, \dots, m\} = W_{n\times m}$$

$$\vec{w}_{\cdot j} = \begin{bmatrix} w_{1j} \\ \cdot \\ \cdot \\ \cdot \\ w_{nj} \end{bmatrix}$$ vector of weights to Y_j

y_in_j = net input to neuron Y_j

$$y\_in_j = b_j + \sum_{i=1}^{n} x_i w_{ij} = b_j + \vec{x} \cdot \vec{w}_{\cdot j} \quad \Rightarrow \quad y_j = f(y\_in_j)$$

mansoori@shirazu.ac.ir

If the brain were so simple
that
we could understand it

then we would be so simple
that
we couldn't understand it

Lyall Watson