Dr. Vishwanath Karad
MIT WORLD PEACE UNIVERSITY | PUNE
TECHNOLOGY, RESEARCH, SOCIAL INNOVATION & PARTNERSHIPS

Unit II

Unit II

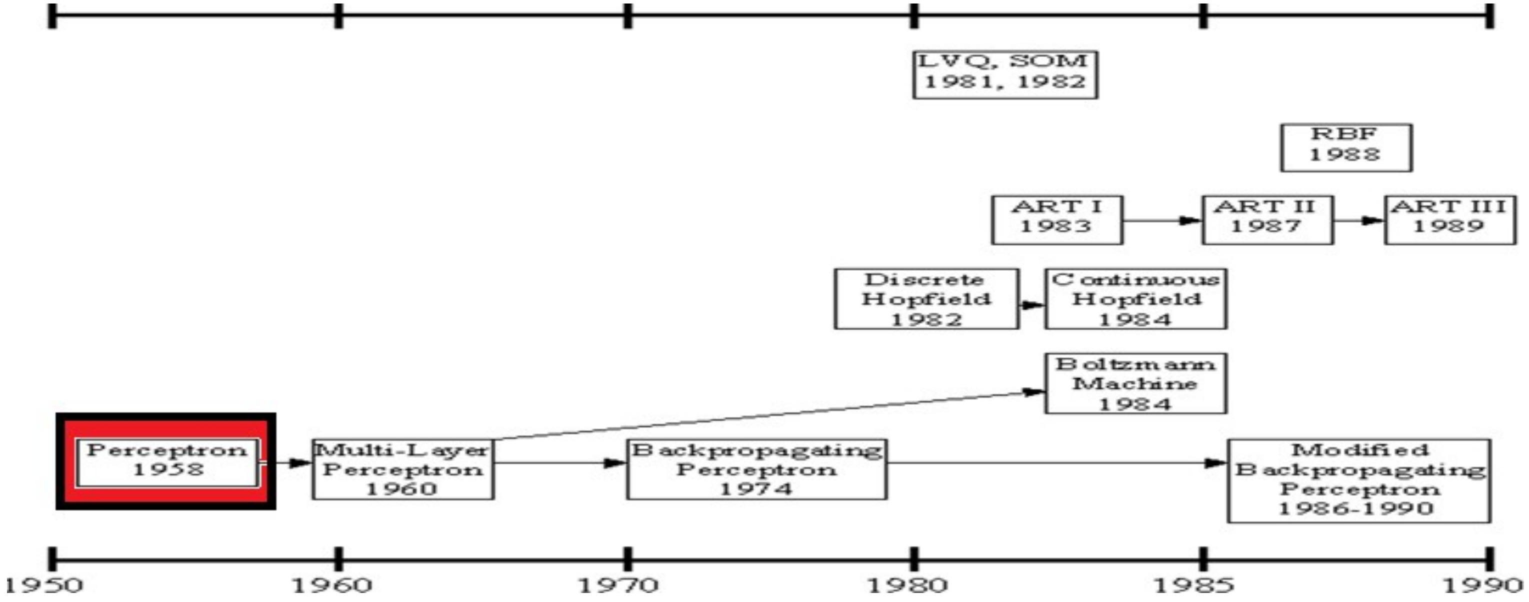
- ➢Machine Learning,
- ➢Deep Learning,
- ➢Reinforcement Learning,
- ➢Natural Language processing,
- ➢Computer Vision,
- ➢ Understanding and summarization of ML and DL based applications.
- ➢AI Tools and Frameworks - TensorFlow, Pandas, NLTK (Natural Language Toolkit)

Artificial Neural Network [ANN]

✓ History of ANN:

➢ History of the ANNs stems from the 1940s, the decade of the first electronic computer.

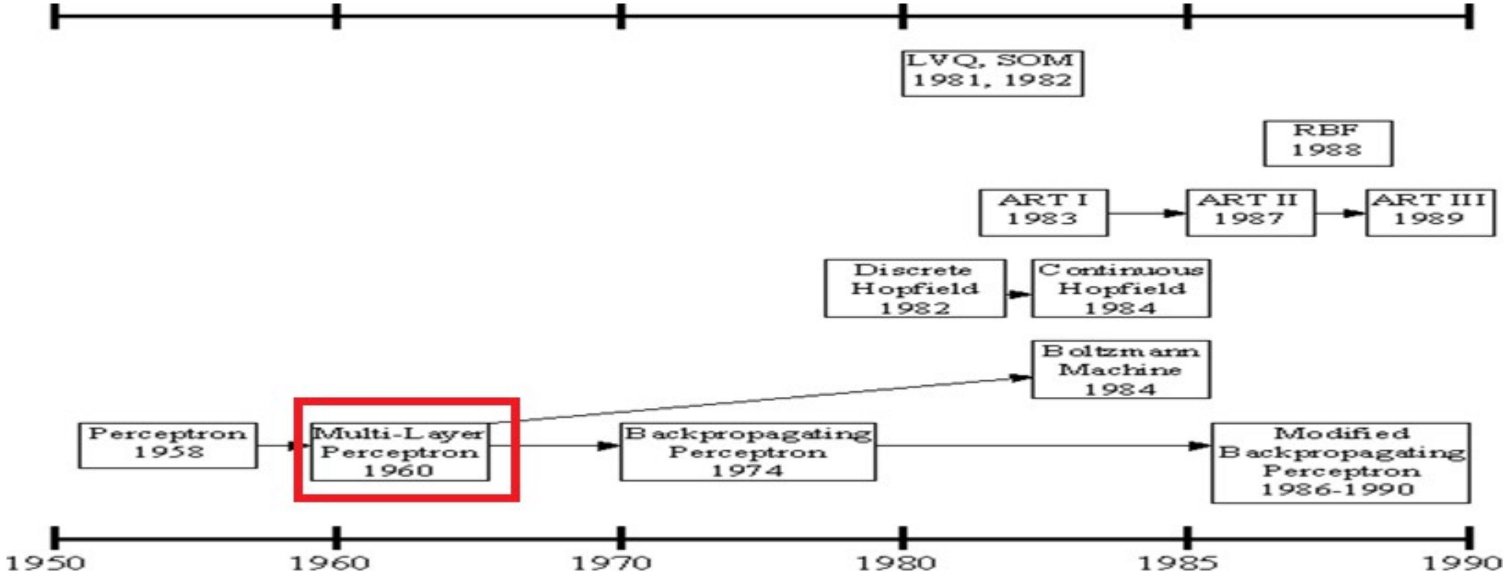
➢ However, the first important step took place in 1957 when Rosenblatt introduced the first concrete neural model, the perceptron. Rosenblatt also took part in constructing the first successful neurocomputer, the Mark I Perceptron. After this, the development of ANNs has proceeded as described in Figure.



Artificial Neural Network [ANN] (Cntd.)

✓ <u>History of ANN (Cntd.):</u>

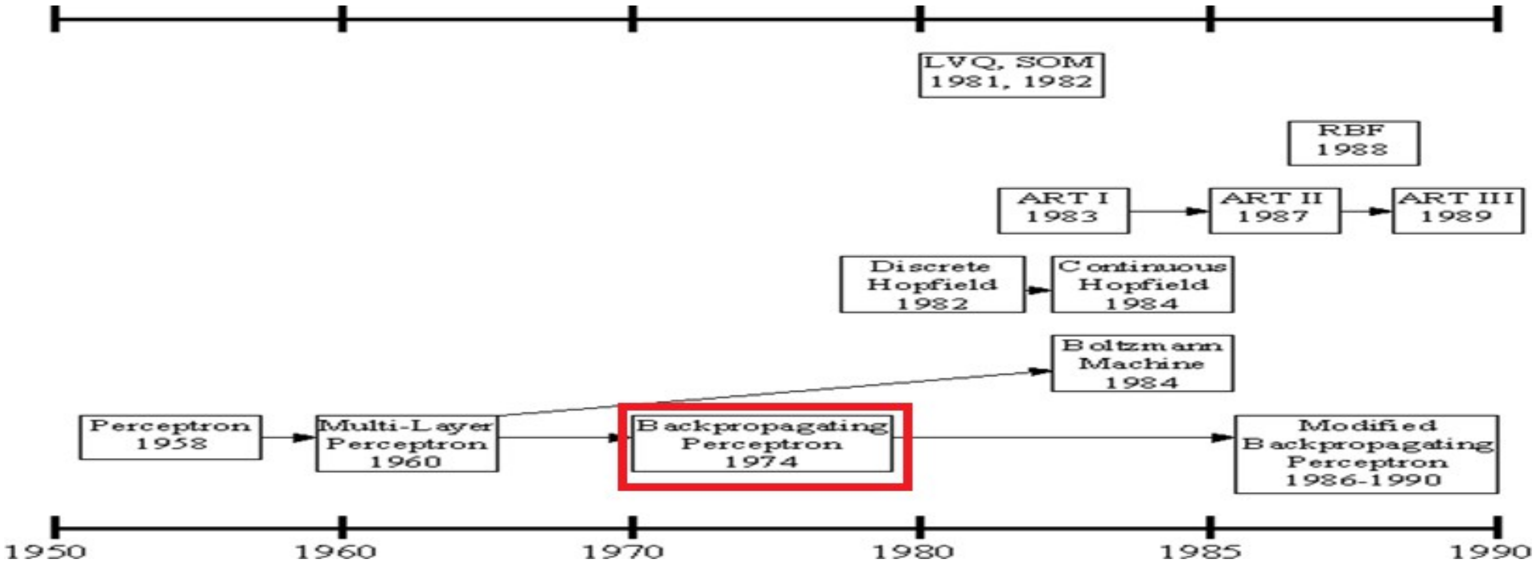
- ➢ Rosenblatt's original perceptron model contained only one layer. From this, a multi-layered model was derived in 1960. At first, the use of the multi-layer perceptron (MLP) was complicated by the lack of a appropriate learning algorithm.
- ➢ In 1974, Werbos came to introduce a so-called back propagation algorithm for the three-layered perceptron network.



Artificial Neural Network [ANN] (Cntd.)

✓ History of ANN (Cntd.):

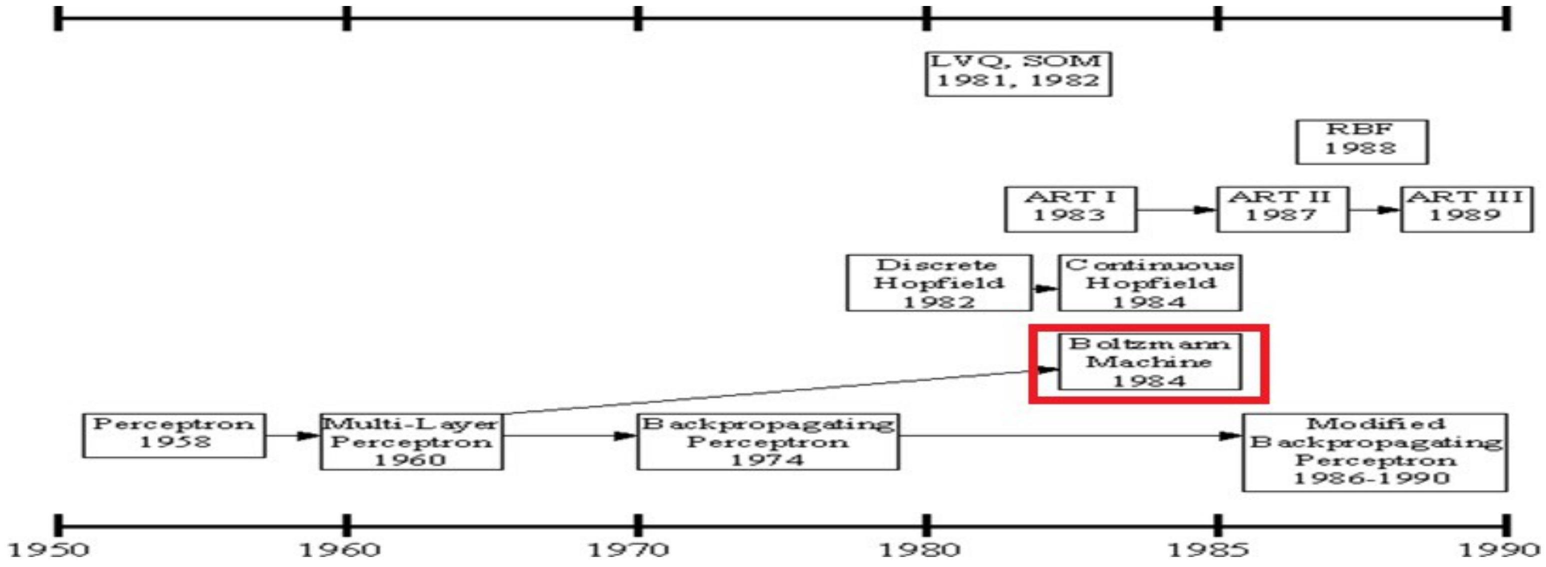
- ➢ in 1986, The application area of the MLP networks remained rather limited until the breakthrough when a general back propagation algorithm for a multi-layered perceptron was introduced by Rummelhart and Mclelland.
- ➢ in 1982, Hopfield brought out his idea of a neural network. Unlike the neurons in MLP, the Hopfield network consists of only one layer whose neurons are fully connected with each other.



Artificial Neural Network [ANN] (Cntd.)

✓ History of ANN (Cntd.):

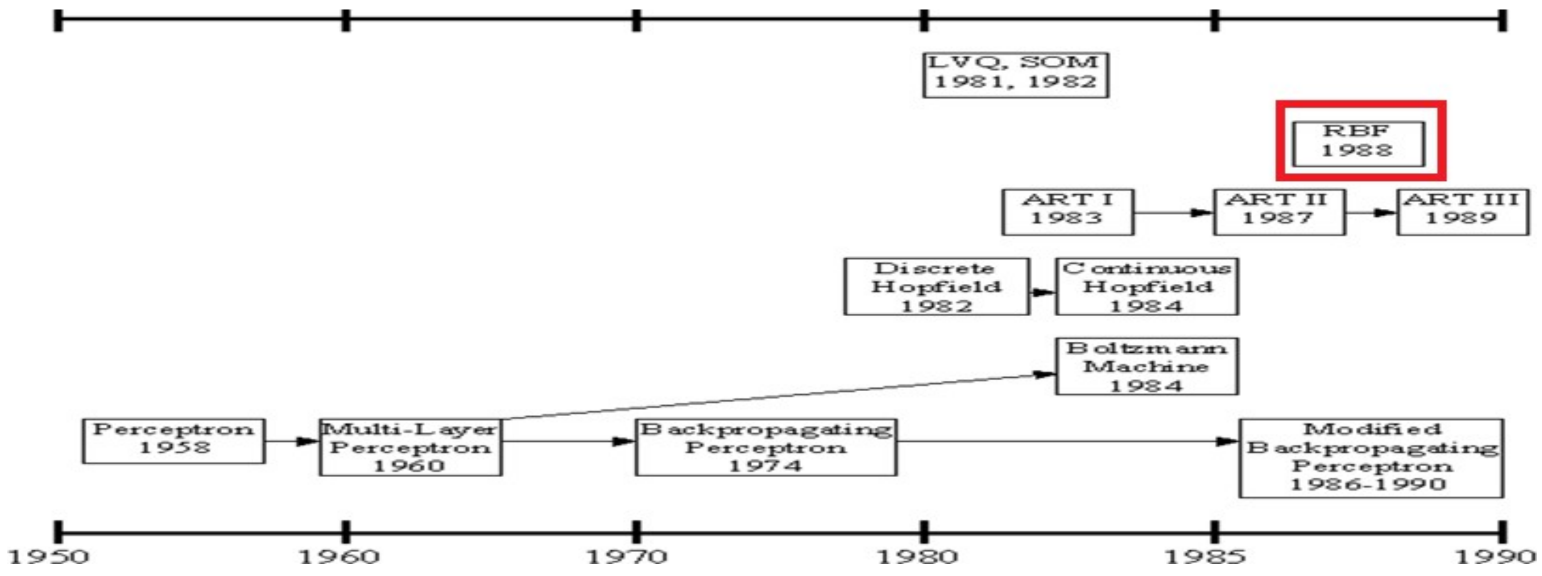
- ➢ Since then, new versions of the Hopfield network have been developed. The Boltzmann machine has been influenced by both the Hopfield network and the MLP.



Artificial Neural Network [ANN] (Cntd.)

✓ History of ANN (Cntd.):

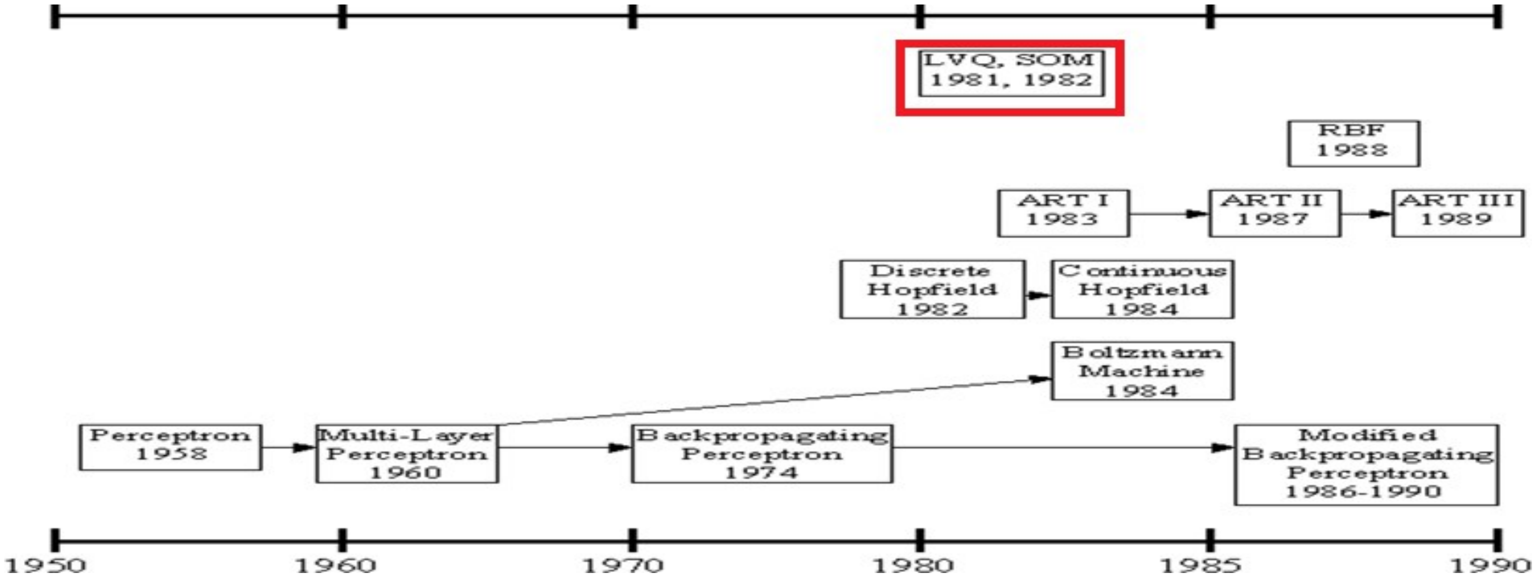
➢ in 1988, Radial Basis Function (RBF) networks were first introduced by Broomhead & Lowe. Although the basic idea of RBF was developed 30 years ago under the name method of potential function, the work by Broomhead & Lowe opened a new frontier in the neural network community.



Artificial Neural Network [ANN] (Cntd.)

✓ History of ANN (Cntd.):

➢ in 1982, A totally unique kind of network model is the Self-Organizing Map (SOM) introduced by Kohonen. SOM is a certain kind of topological map which organizes itself based on the input patterns that it is trained with. The SOM originated from the LVQ (Learning Vector Quantization) network the underlying idea of which was also Kohonen's in 1972.



Artificial Neural Network [ANN] (Cntd.)

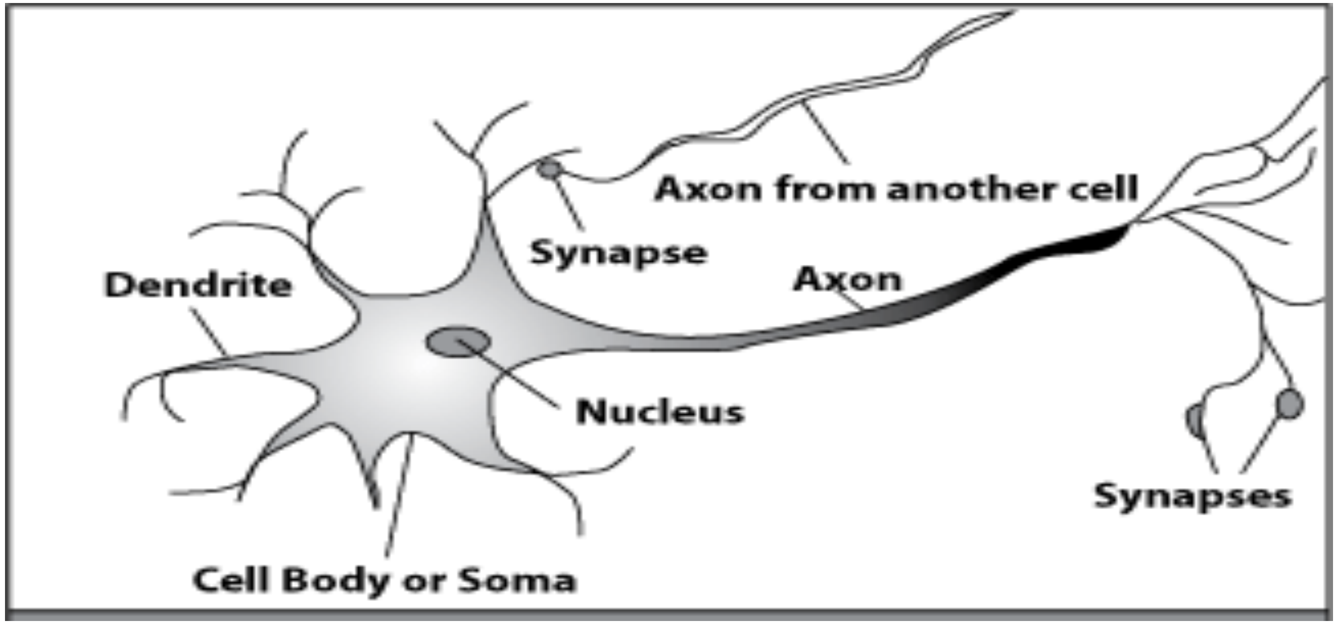
✓ History of ANN (Cntd.):

- Since then, research on artificial neural networks has remained active, leading to many new network types, as well as hybrid algorithms and hardware for neural information processing.

Artificial Neural Network [ANN] (Cntd.)

✓ <u>How Brain works?</u>

- ➢ We have discussed briefly on the basic findings of neuroscience—in particular, the hypothesis that mental activity consists primarily of electrochemical activity in networks of brain cells called neurons. Following showed a schematic diagram of a typical neuron.



Dendrites: Input
Cell body: Processor
Synaptic: Link Axon: Output

Artificial Neural Network [ANN] (Cntd.)

✓ How Brain works? (Cntd.)

- The Brain is A massively parallel information processing system.
- Our brains are a huge network of processing elements. A typical brain contains a network of 10 billion neurons.

Artificial Neural Network [ANN] (Cntd.)

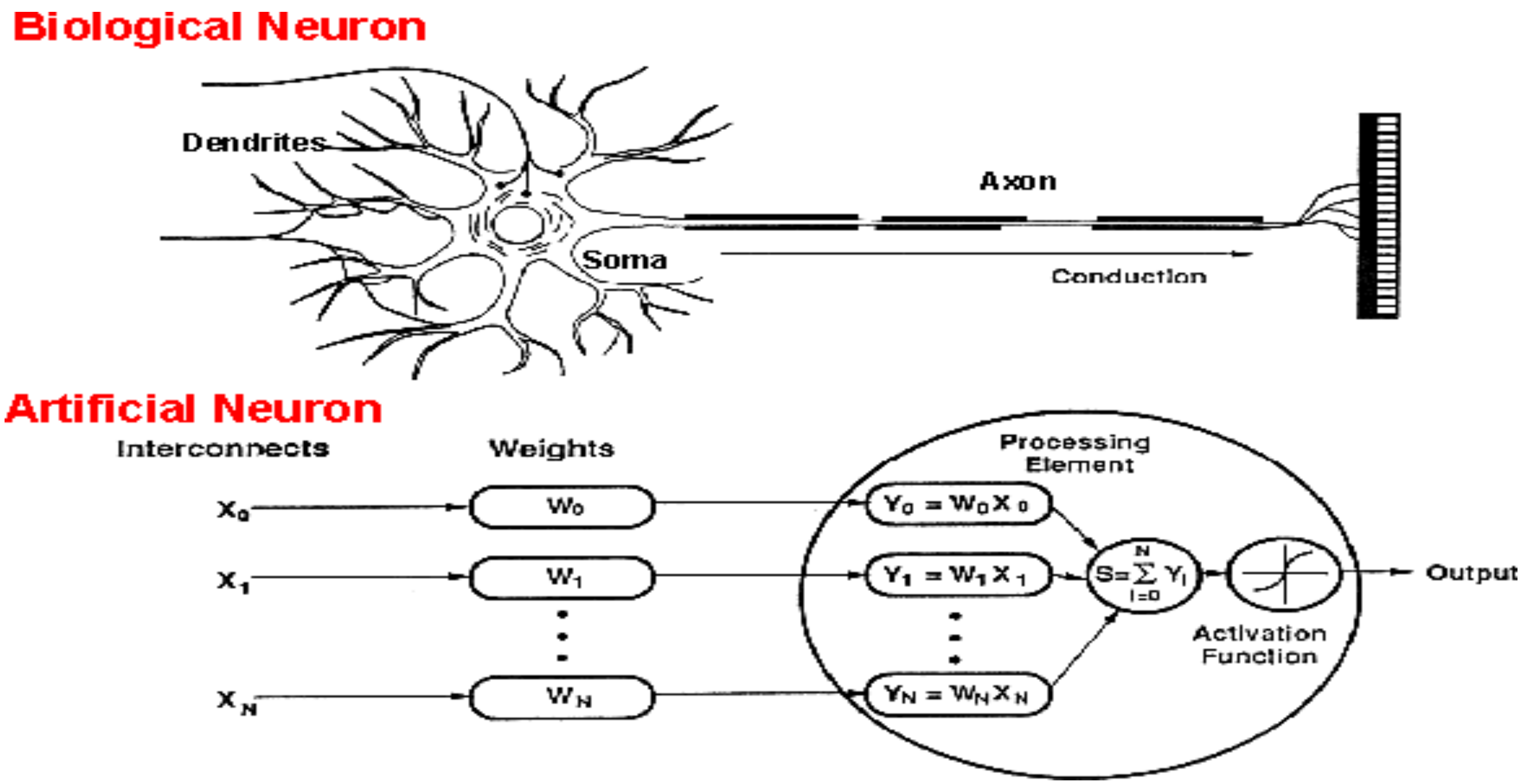
✓ <u>How do ANN works?</u>

- ➢ An artificial neural network (ANN) is either a hardware implementation or a computer program which strives to simulate the information processing capabilities of its biological exemplar. ANNs are typically composed of a great number of interconnected artificial neurons. The artificial neurons are simplified models of their biological counterparts.
- ➢ ANN is a technique for solving problems by constructing software that works like our brains.

Artificial Neural Network [ANN] (Cntd.)

✓ <u>How do ANN works? (Cntd.)</u>

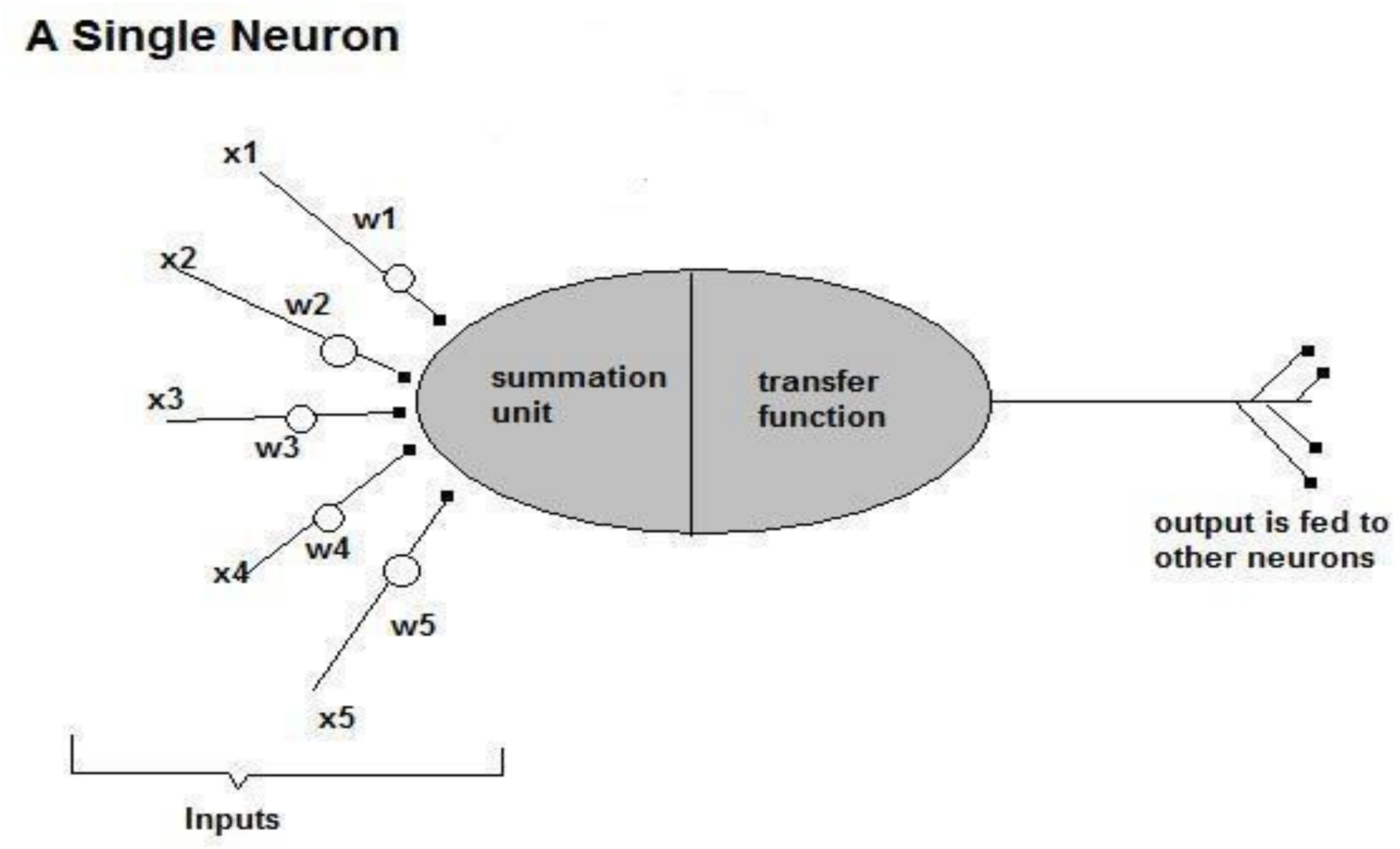
➢ An artificial neuron is an imitation of a human neuron



Artificial Neural Network [ANN] (Cntd.)

✓ <u>How do ANN works? (Cntd.)</u>

➢ Now, let us have a look at the model of an artificial neuron.



Artificial Neural Network [ANN] (Cntd.)

✓ <u>How do ANN works? (Cntd.)</u>

- ➢ Neural networks have lots of pieces, and to refer to them we will need to introduce a variety of mathematical notations. Following Table lists all notations used

Artificial Neural Network [ANN] (Cntd.)

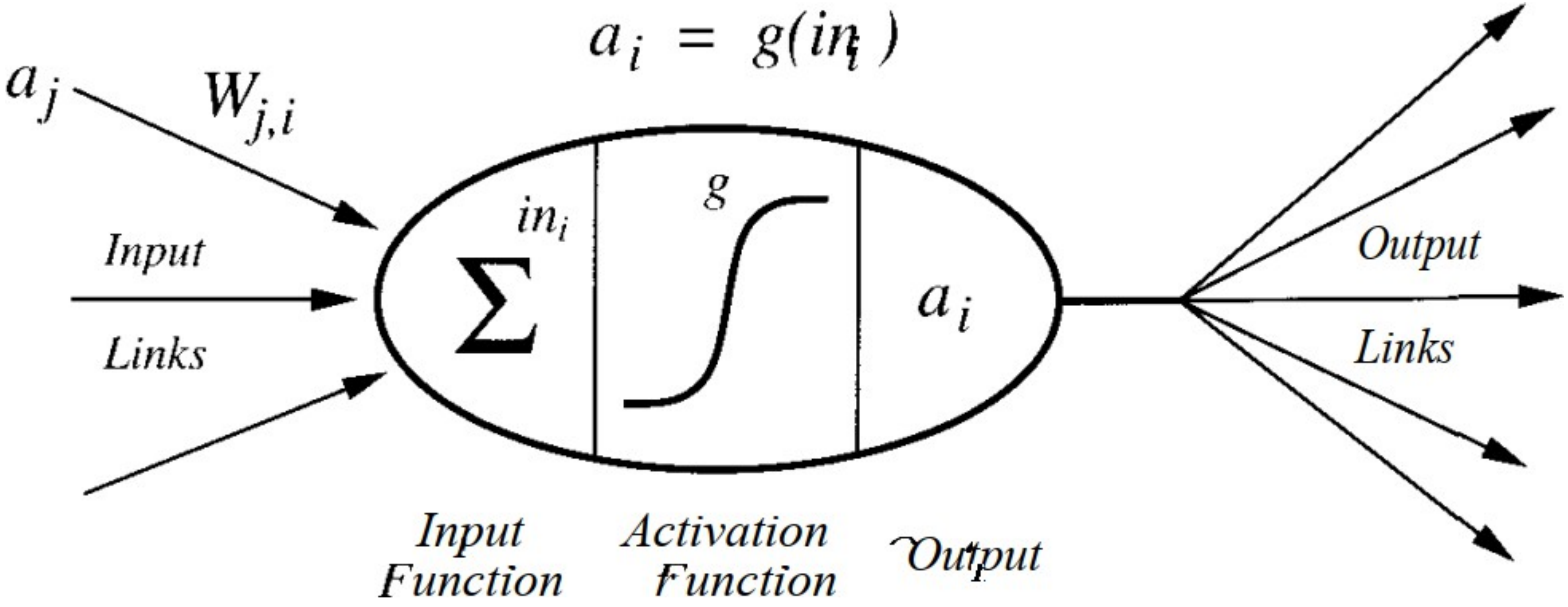
Notation	Meaning
a_i $\mathbf{a}_i$	Activation value of unit i (also the output of the unit) Vector of activation values for the inputs to unit i
g, g'	Activation function Derivative of the activation function
Err_i Err^e	Error (difference between output and target) for unit i Error for example e
I_i $\mathbf{I}$ $\mathbf{I}^e$	Activation of a unit i in the input layer Vector of activations of all input units Vector of inputs for example e
in_i	Weighted sum of inputs to unit i
N	Total number of units in the network
O O_i $\mathbf{O}$	Activation of the single output unit of a perceptron Activation of a unit i in the output layer Vector of activations of all units in the output layer
t	Threshold for a step function
T $\mathbf{T}$ $\mathbf{T}^e$	Target (desired) output for a perceptron Target vector when there are several output units Target vector for example e
$W_{j,i}$ W_i $\mathbf{W}_i$ $\mathbf{W}$	Weight on the link from unity to unit i Weight from unit r to the output in a perceptron Vector of weights leading into unit i Vector of all weights in the network

Figure 19.3 Neural network notation. Subscripts denote units; superscripts denote examples.

Artificial Neural Network [ANN] (Cntd.)

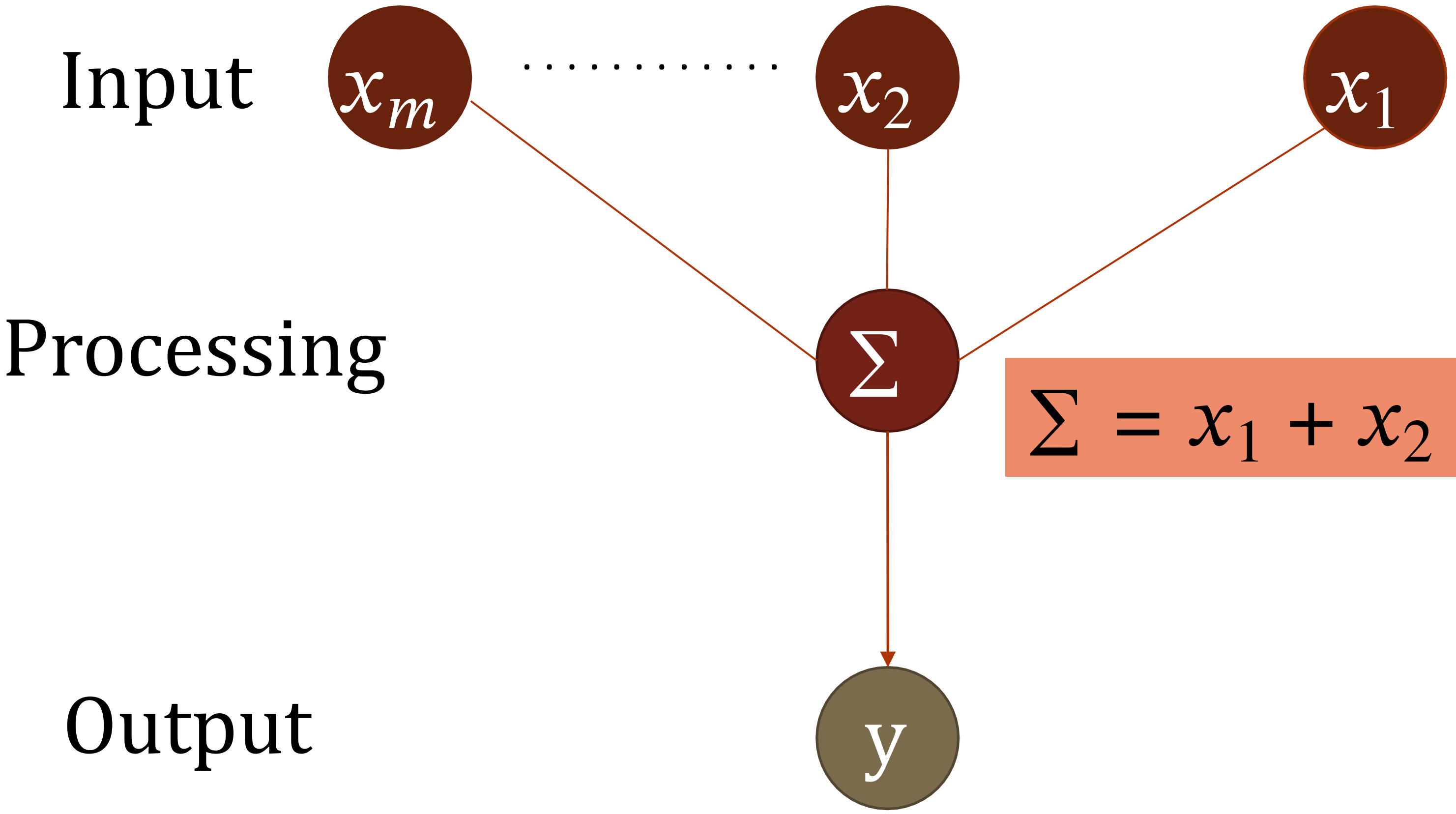
✓ <u>How do ANN works? (Cntd.)</u>

➢ The basic unit of computation in a neural network is the neuron, often called a node or unit. Following figure shows a Unit in Neural network.



Artificial Neural Network [ANN] (Cntd.)

✓ How do ANN works? (Cntd.)



$$\sum = x_1 + x_2 + \quad \ldots . + x_m = y$$

Artificial Neural Network [ANN] (Cntd.)

✓ How do ANN works? (Cntd.)

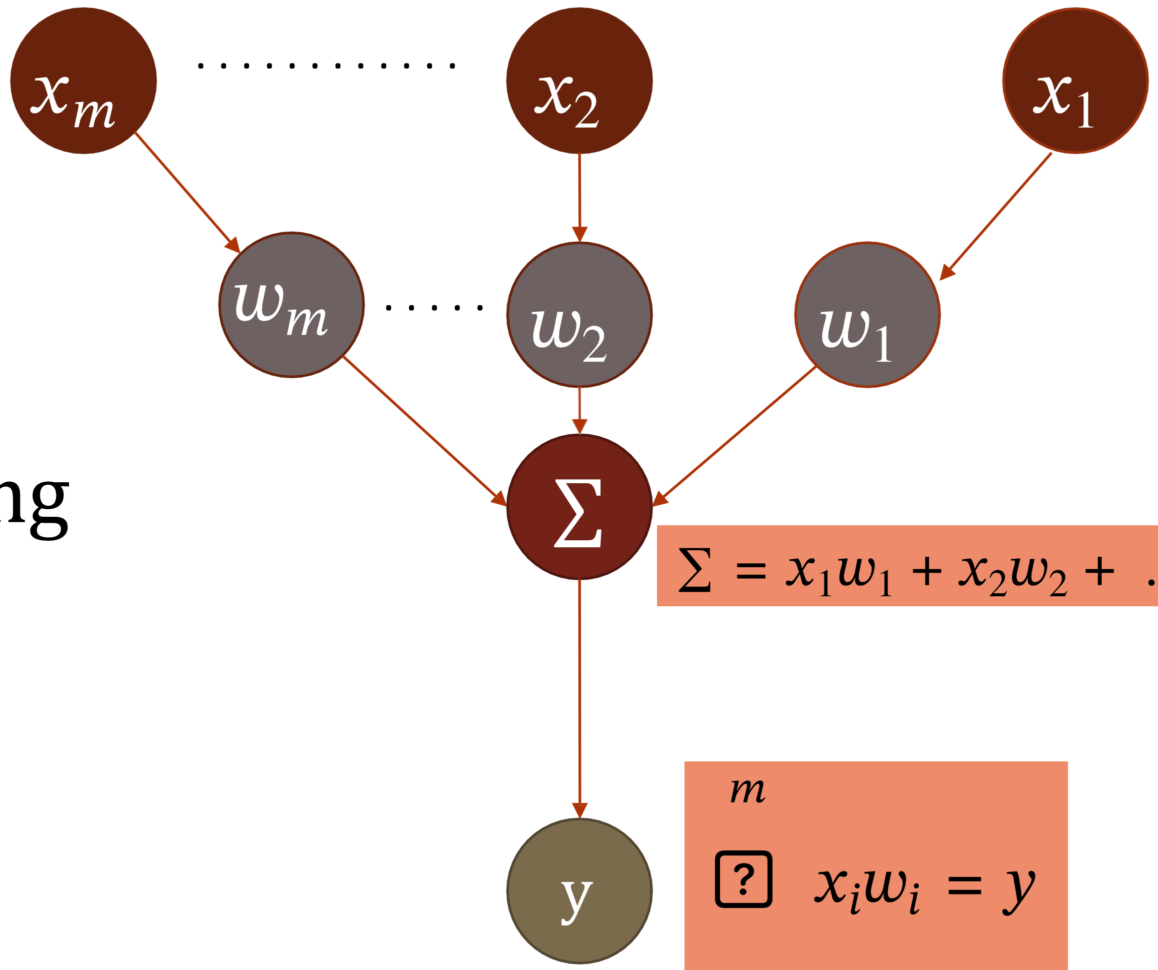
Input

Weights

Not all inputs are equal

Processing

Output



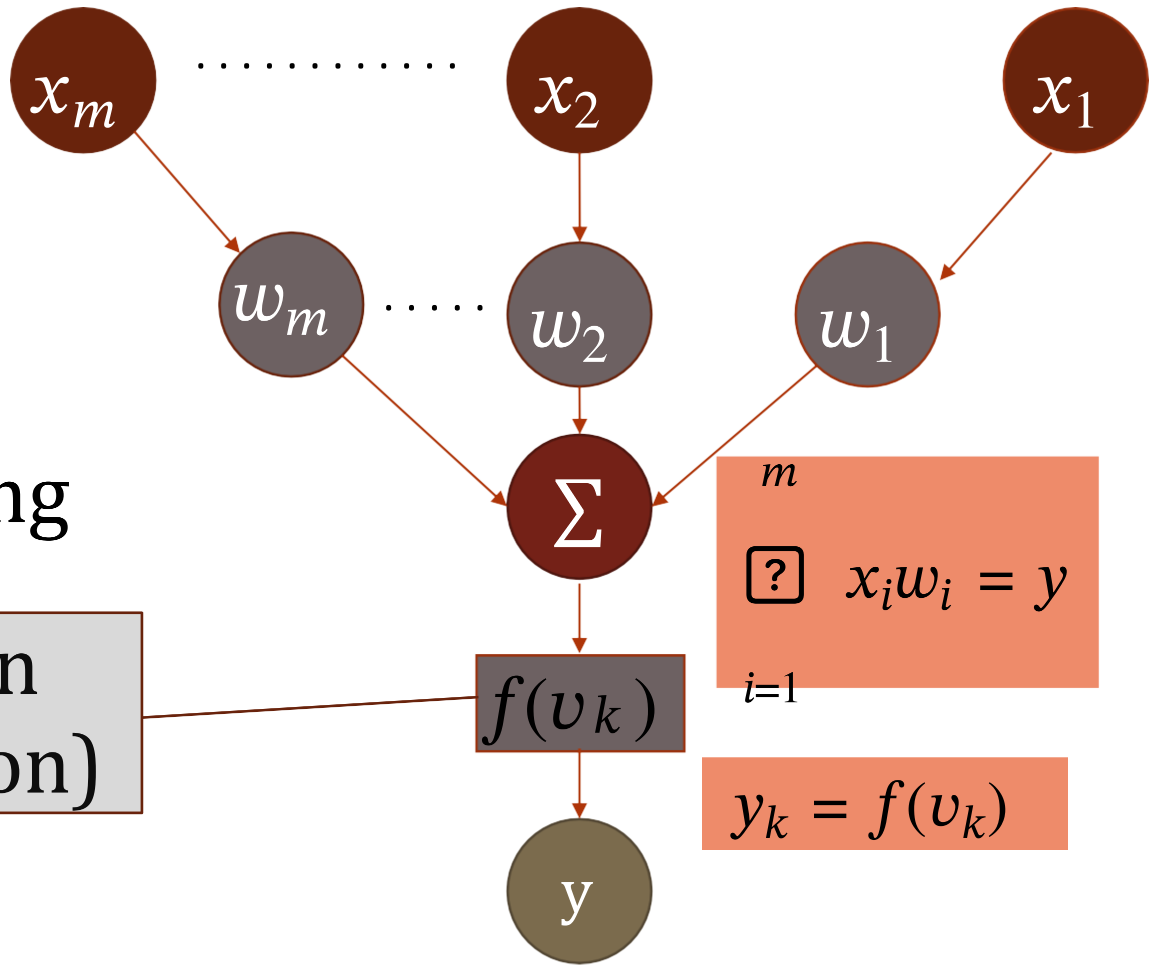
$$\sum = x_1 w_1 + x_2 w_2 + \; \ldots . + x_m w_m = y$$

$$\sum_{i=1}^{m} x_i w_i = y$$

Artificial Neural Network [ANN] (Cntd.)

✓ <u>How do ANN works? (Cntd.)</u>

The signal is not passed down to the next neuron verbatim



$$\sum_{i=1}^{m} x_i w_i = y$$

$$y_k = f(v_k)$$

Artificial Neural Network [ANN] (Cntd.)

✓ How do ANN works? (Cntd.)

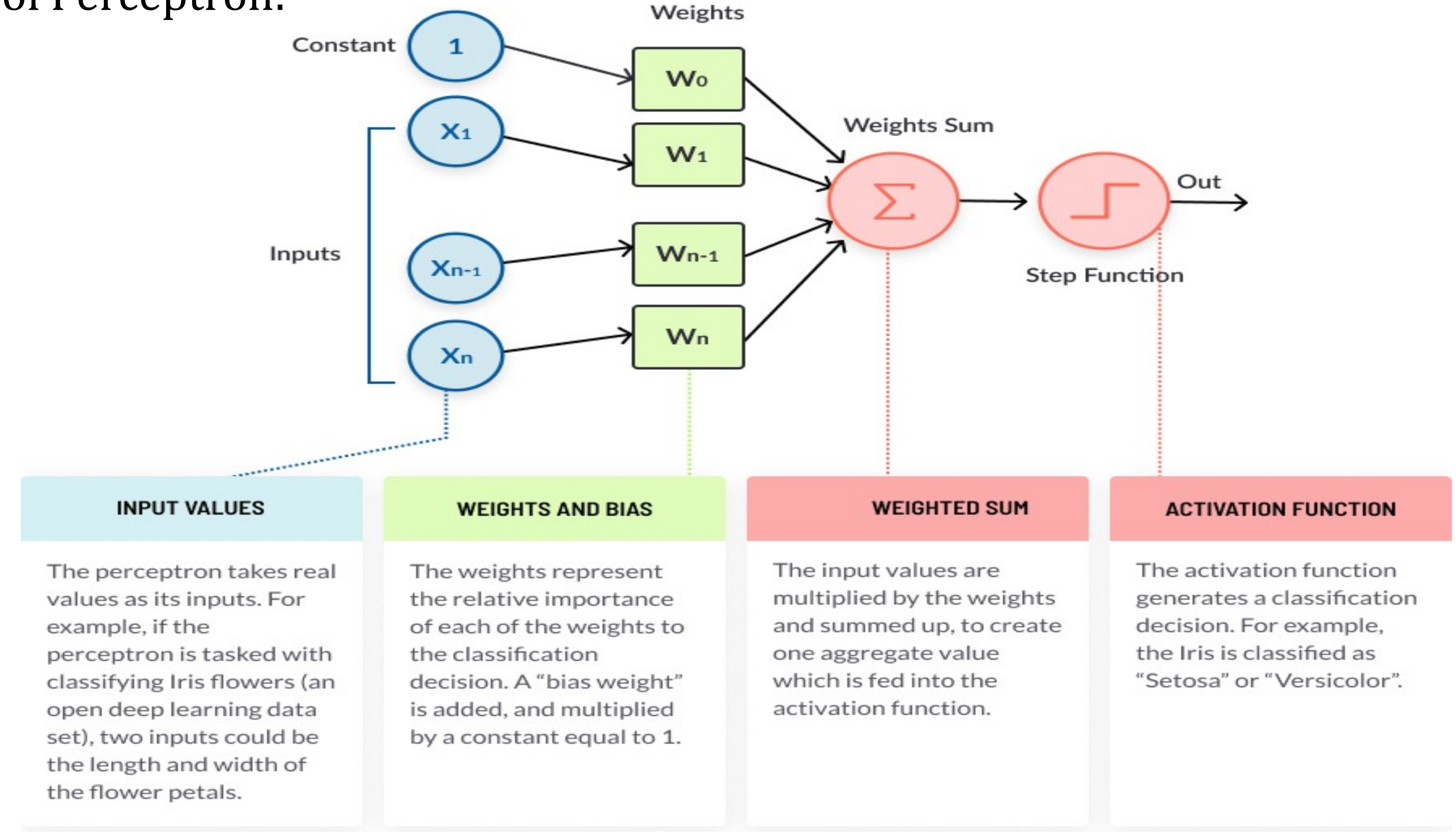
Perceptron:

- A perceptron is a neural network unit (an artificial neuron) that does certain computations to detect features or business intelligence in the input data.
- Perceptron is usually used to classify the data into two parts. Therefore, it is also known as a Linear Binary Classifier. Also, it is used in supervised learning. It helps to classify the given input data.
- Perceptron is a single layer neural network and a multi-layer perceptron is called Neural Networks.
- There are two types of Perceptrons: Single layer and Multilayer.
  - Single layer Perceptrons can learn only linearly separable patterns.
  - Multilayer Perceptrons or feed forward neural networks with two or more layers have the greater processing power.

Artificial Neural Network [ANN] (Cntd.)

✓ <u>How do ANN works? (Cntd.)</u>

Structure of Perceptron:



Artificial Neural Network [ANN] (Cntd.)

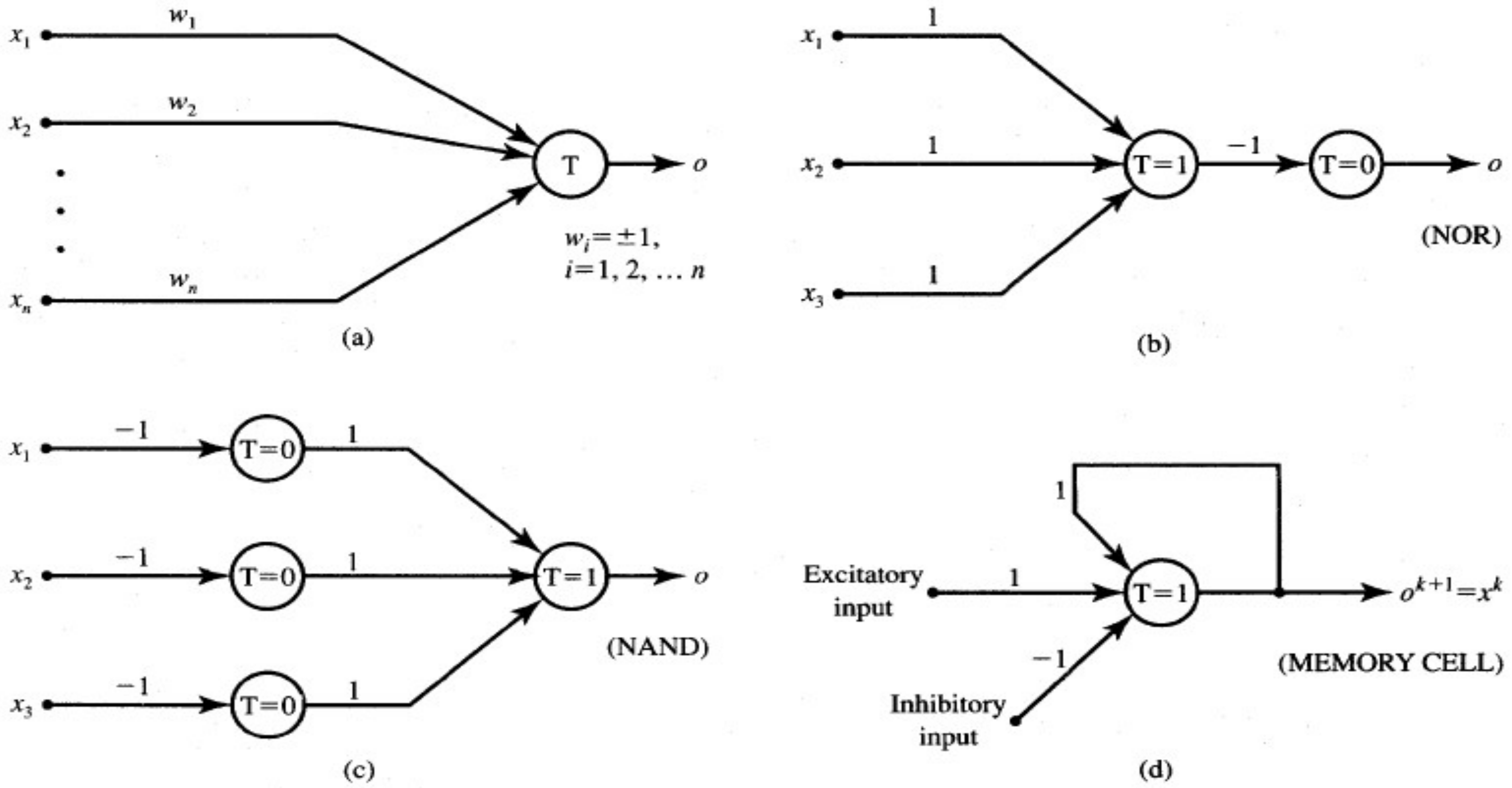
✓ Comparison between brain verses computer

	Brain	ANN
Speed	Few ms.	Few nano sec. massive ||el processing
Size and complexity	10^{11} neurons & 10^{15} interconnections	Depends on designer
Storage capacity	Stores information in its interconnection or in synapse. No Loss of memory	Contiguous memory locations loss of memory may happen sometimes.
Tolerance	Has fault tolerance	No fault tolerance If gets disrupted when interconnections are disconnected
Control mechanism	Complicated involves chemicals in biological neuron	Simpler in ANN

Artificial Neural Network [ANN] (Cntd.)

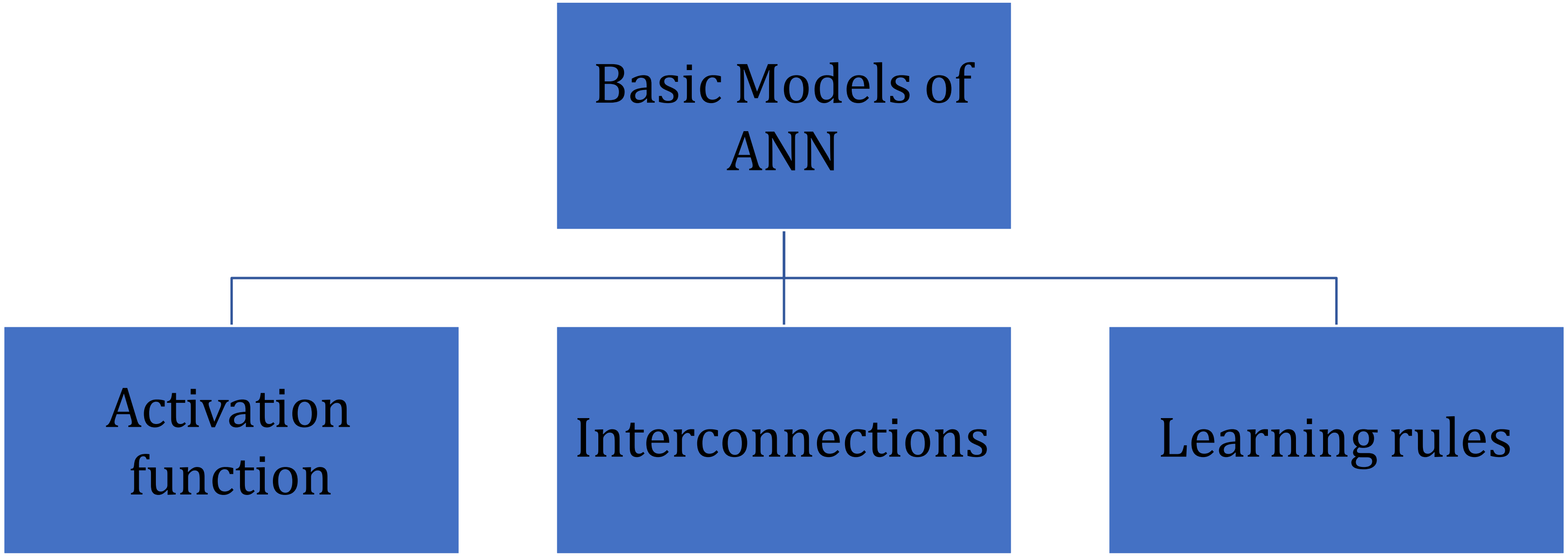
✓ McCulloch-Pitts Neuron Model

$$o^{k+1} = \begin{cases} 1 & \text{if } \sum_{i=1}^{n} w_i x_i^k \geq T \\ 0 & \text{if } \sum_{i=1}^{n} w_i x_i^k < T \end{cases}$$



Artificial Neural Network [ANN] (Cntd.)

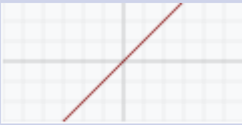
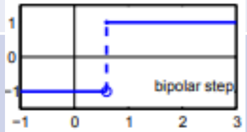
✓ Basic models of ANN



Artificial Neural Network [ANN] (Cntd.)

✓ <u>Activation Function:</u>

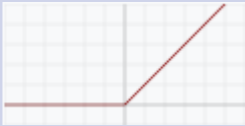
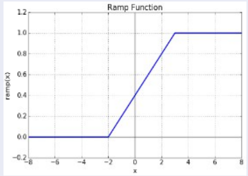
➢ The purpose of the activation function is to introduce non-linearity into the output of a neuron.

Name	Equation	Plot	Range
Linear or Identity Function	$f(x) = x \ for \ all \ x$		$(-\infty, \infty)$
Binary Step function	$f(x) = \begin{cases} 1 & if \ x \geq 0 \\ 0 & if \ x < 0 \end{cases}$		$(0,1)$
Bipolar Step function	$f(x) = \begin{cases} +1 & if \ x \geq 0 \\ -1 & if \ x < 0 \end{cases}$		$(-1, +1)$
Logistic (a.k.a. Sigmoid or Soft step)	$f(x) = \frac{1}{1+e^{-x}}$		$(0,1)$
TanH (a.k.a. Tangent Hyperbolic function)	$f(x) = \frac{2}{1+e^{-2x}} - 1$		$(-1, +1)$

Artificial Neural Network [ANN] (Cntd.)

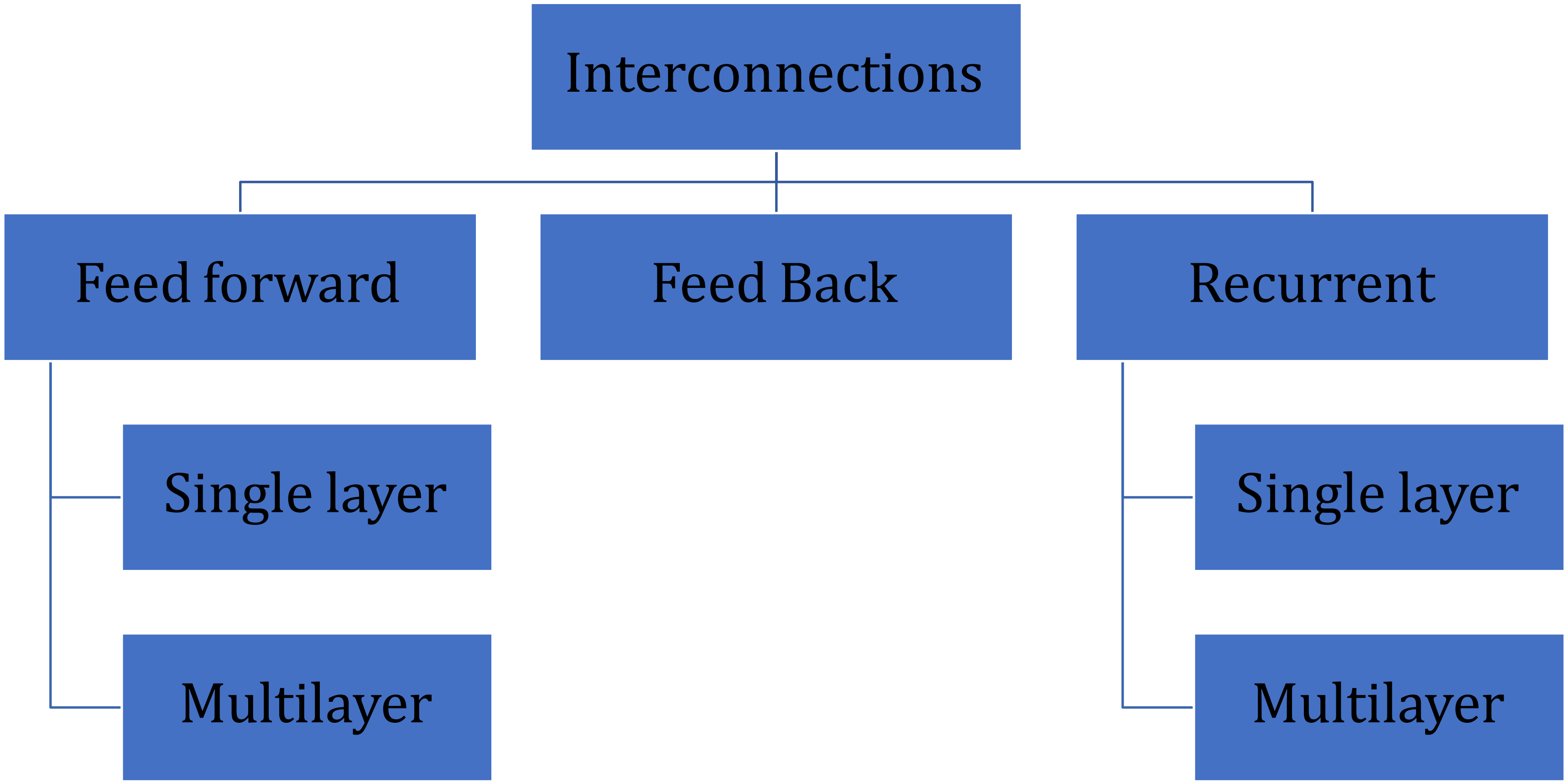
✓ Activation Function: (cntd.)

➢ The purpose of the activation function is to introduce non-linearity into the output of a neuron.

Name	Equation	Plot	Range
Rectified linear unit (ReLU)	$f(x) = \begin{cases} 0 & if\ x \leq 0 \\ X & if\ x > 0 \end{cases}$		$(0, \infty)$
Ramp Function	$f(x) = \begin{cases} 0 & if\ x_i \leq T_1 \\ \frac{x_i - T_1}{T_2 - T_1} & if\ T_1 \leq x_i \leq T_2 \\ 1 & if\ x_i > T_2 \end{cases}$ T_1 and T_2 = definitive cut off points. (In this case T_1=-2 and T_2=3)		(0,1)

Artificial Neural Network [ANN] (Cntd.)

✓ <u>Interconnection:</u>

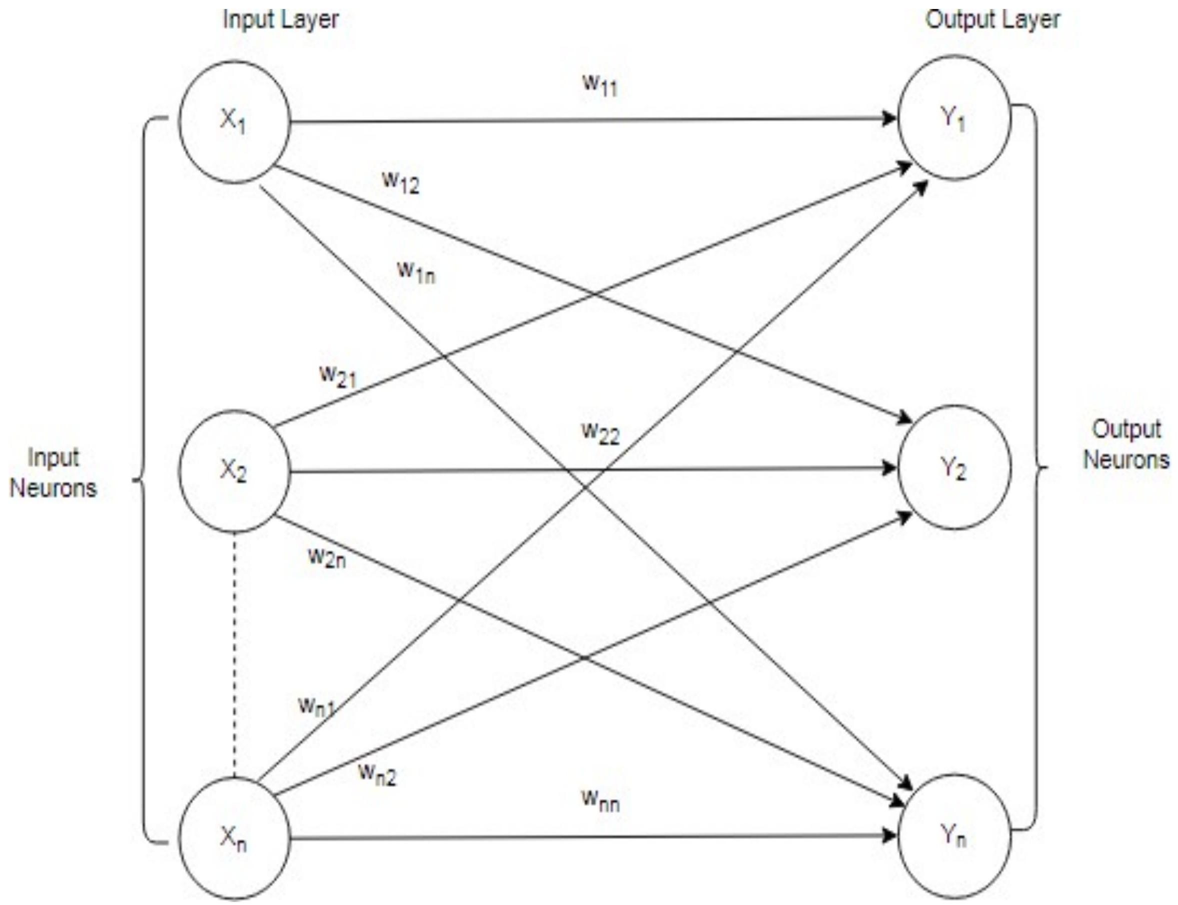


Artificial Neural Network [ANN] (Cntd.)

✓ <u>Interconnection: (Cntd.)</u>

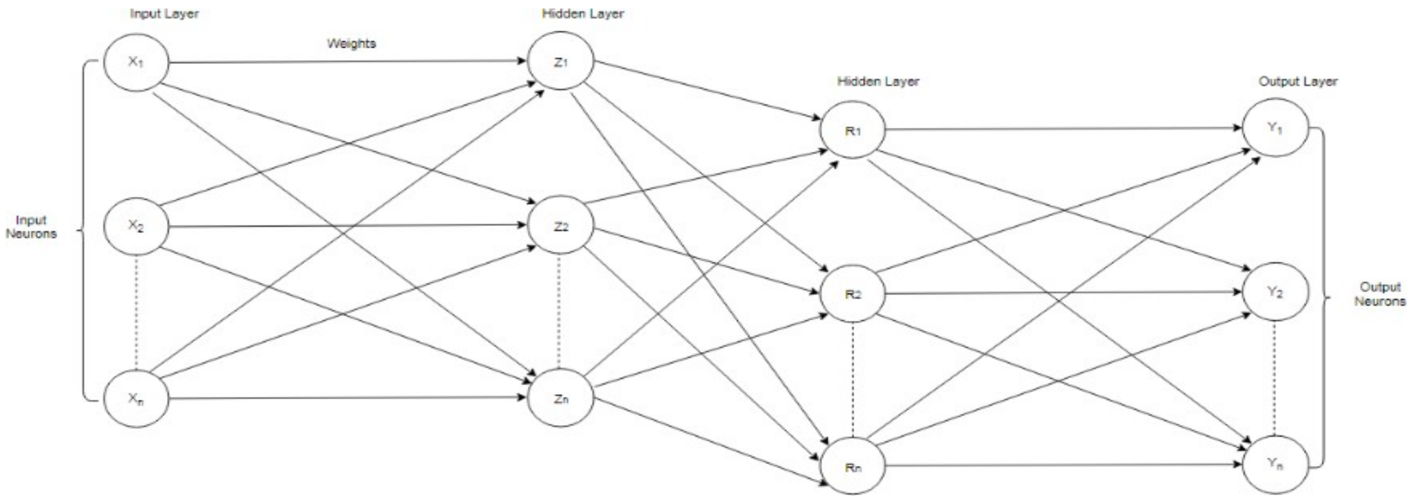
➢ <u>Single Layer Feed Forward Network</u>

- A layer is formed by taking a processing element and combining it with other processing elements. When a layer of the processing nodes is formed, the inputs can be connected to these nodes with various weights, resulting in a series of outputs, one per node. Thus, a single layer feed forward network is formed.



Artificial Neural Network [ANN] (Cntd.)

- Interconnection: (Cntd.)
  - Multilayer feed forward network
    - A multilayer feed forward network is formed by the interconnection of several layers. The input layer is that which receives the input and this layer has no function except buffering the input signal. The output layer generates the output of the network. Any layer that is formed between the input layer and the output layer is called the hidden layer.
    - Can be used to solve complicated problems.

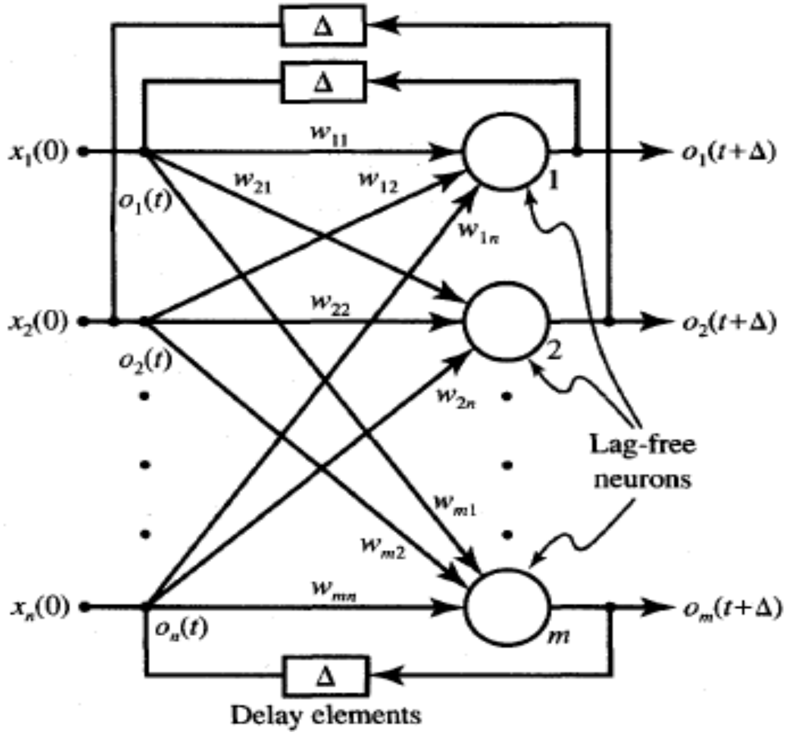


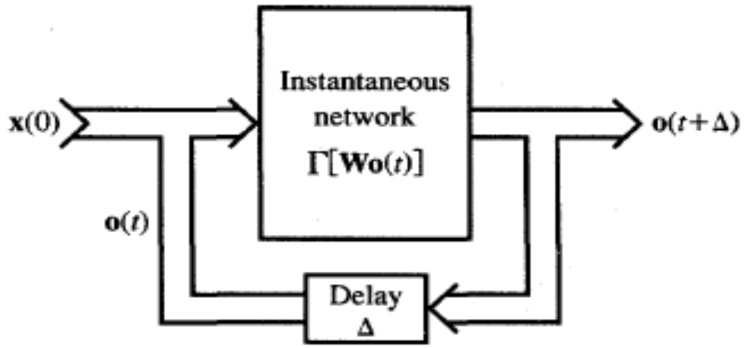
Artificial Neural Network [ANN] (Cntd.)

✓ Interconnection: (Cntd.)

➢ Feedback network

- When outputs are directed back as inputs to same or preceding layer nodes it results in the formation of feedback networks.





Artificial Neural Network [ANN] (Cntd.)

- ✓ Interconnection: (Cntd.)
  - ➢ Recurrent network
    - Feedback networks with closed loop are called Recurrent Networks.
    - Recurrent neural network is a class of artificial neural network where connections between nodes form a directed graph along a sequence. This allows it to exhibit dynamic temporal behavior for a time sequence. Unlike feed forward neural networks, RNNs can use their internal state (memory) to process sequences of inputs.

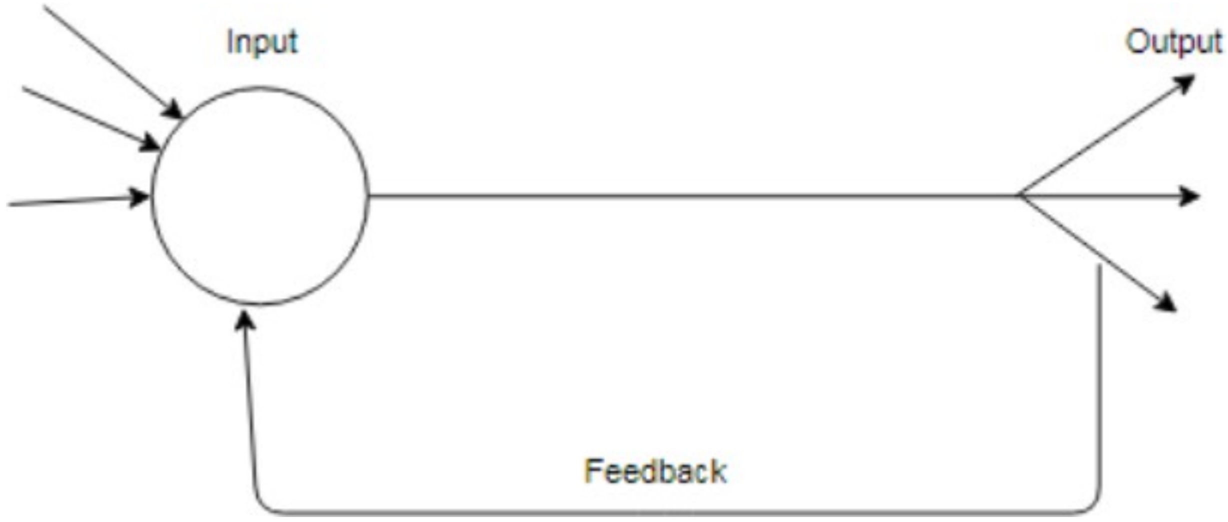
1. Single node with own feedback
2. Competitive networks
3. Single-layer recurrent networks
4. Multilayer recurrent networks
5. Jordan Network

Artificial Neural Network [ANN] (Cntd.)

✓ <u>Interconnection: (Cntd.)</u>

➢ <u>Recurrent network (Cntd.)</u>

- Single node with own feedback



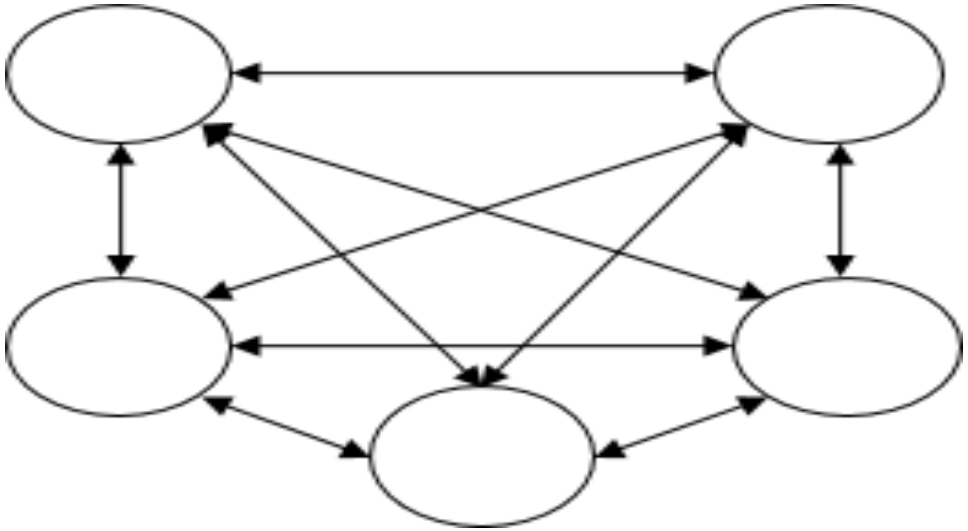
Artificial Neural Network [ANN] (Cntd.)

✓ Interconnection: (Cntd.)

➢ Recurrent network (Cntd.)

- Fully recurrent network (Competitive networks)

It is the simplest neural network architecture because all nodes are connected to all other nodes and each node works as both input and output.



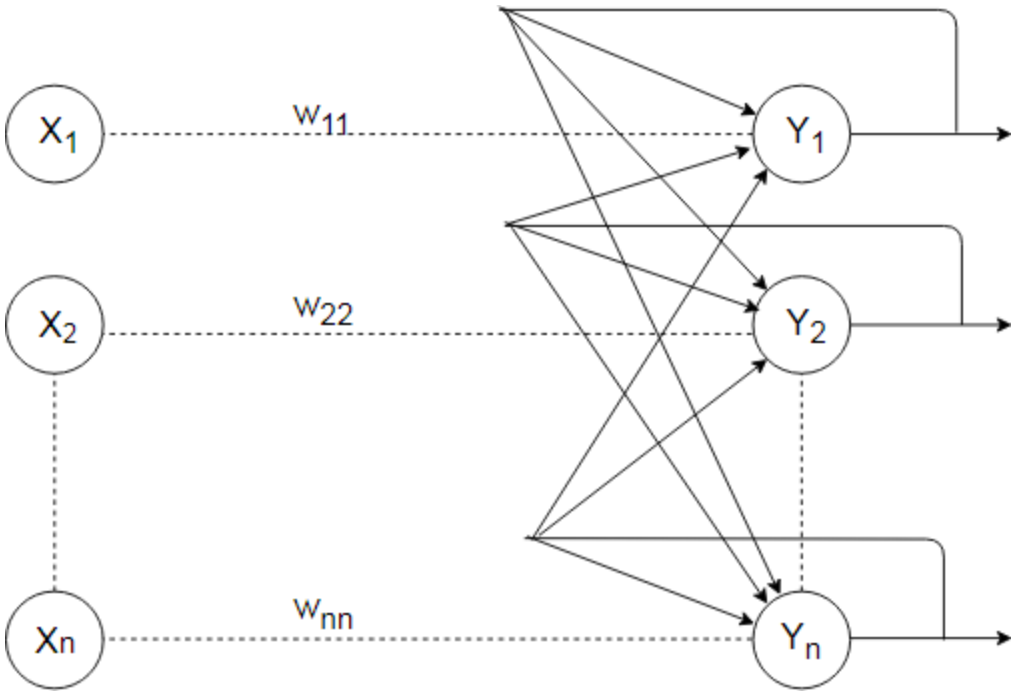
Artificial Neural Network [ANN] (Cntd.)

✓ Interconnection: (Cntd.)

➢ Recurrent network (Cntd.)

- Single-layer recurrent network

This network is single layer network with feedback connection in which processing element's output can be directed back to itself or to other processing element or both.



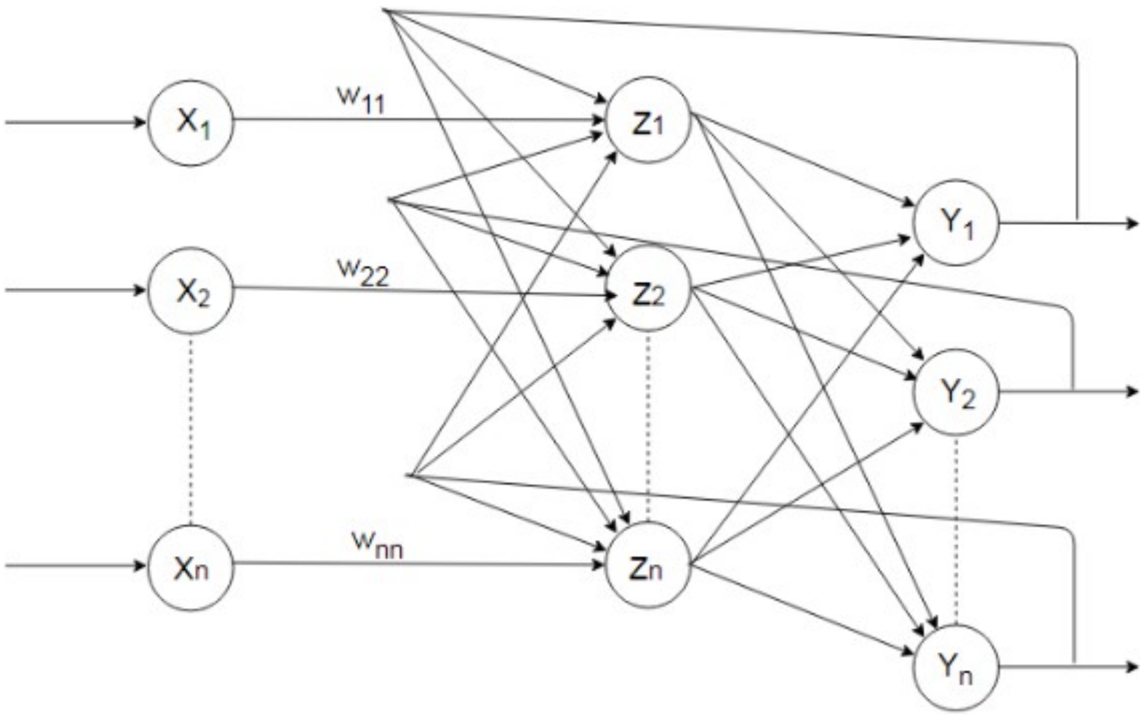
Artificial Neural Network [ANN] (Cntd.)

✓ Interconnection: (Cntd.)

➢ Recurrent network (Cntd.)

- Multi-layer recurrent network

In this type of network, processing element output can be directed to the processing element in the same layer and in the preceding layer forming a multilayer recurrent network. They perform the same task for every element of a sequence, with the output being depended on the previous computations. Inputs are not needed at each time step. The main feature of an Recurrent Neural Network is its hidden state, which captures some information about a sequence.



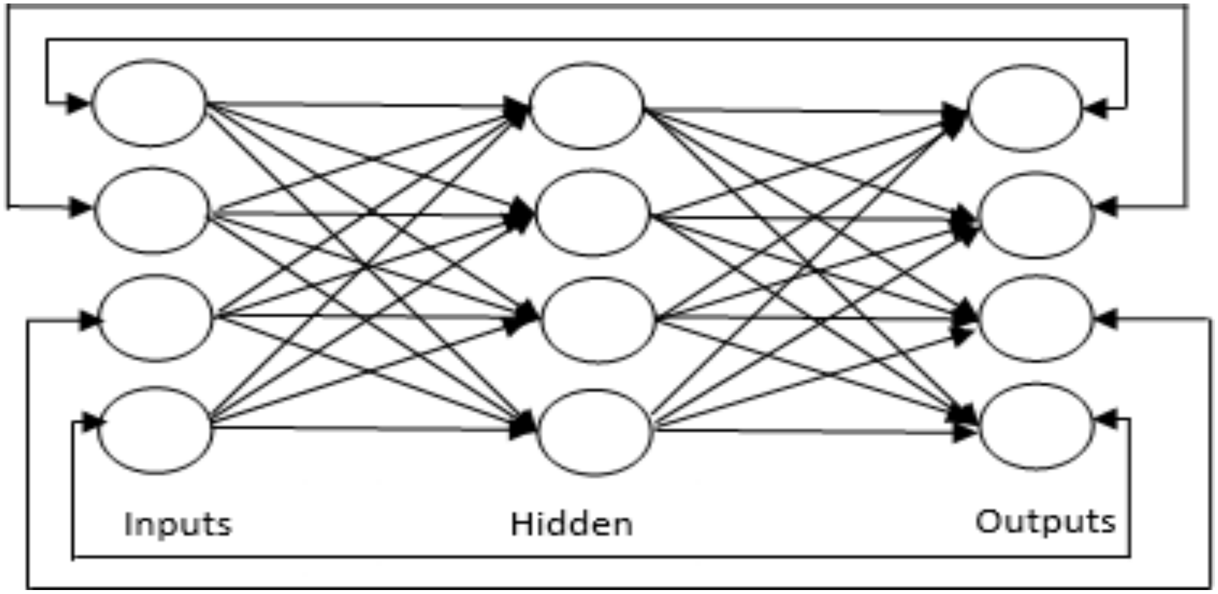
Artificial Neural Network [ANN] (Cntd.)

✓ <u>Interconnection: (Cntd.)</u>

➢ <u>Recurrent network (Cntd.)</u>

- Jordan network

It is a closed loop network in which the output will go to the input again as feedback as shown in the following diagram.



Comparison

Feature	Fully Recurrent Network	Single-layer RNN	Multi-layer RNN	Jordan Network
Recurrent Connections	Between **all nodes**	Within **hidden layer only**	In **multiple hidden layers**	From **output to hidden**
Complexity	High	Moderate	High	Moderate
Feedback Path	Everywhere	Hidden to hidden	Layer-wise recurrent paths	Output to hidden (context)
Typical Use	Theoretical/ complex systems	Simple sequence tasks	Complex temporal modeling	Sequential prediction
Example	Hopfield Network	Elman Network	Stacked LSTM, GRU	Jordan Network

Artificial Neural Network [ANN] (Cntd.)

- Learning:
  - It's a process by which a NN adapts itself to a stimulus by making proper parameter adjustments, resulting in the production of desired response
  - Two kinds of learning
    1. Parameter learning:- connection weights are updated
    2. Structure Learning:- change in network structure

What is Machine Learning?

- **Machine Learning (ML)** is a field of **Artificial Intelligence (AI)** that focuses on building systems that can **learn from data and improve their performance over time without being explicitly programmed**.

Simple Definition:

- **Machine learning is the process by which computers learn patterns from data to make decisions or predictions.**

Types of Machine Learning:

Type	Description	Example
Supervised Learning	Learns from labeled data (input-output pairs)	Predicting house prices, spam detection
Unsupervised Learning	Finds patterns in **unlabeled** data	Customer segmentation, anomaly detection
Reinforcement Learning	Learns by trial and error, receiving rewards/punishments	Game playing, robotics

Basic ML Workflow:

- **Collect Data** – e.g., photos, numbers, text, etc.
- **Train a Model** – Feed the data to an algorithm to learn patterns.
- **Test the Model** – Check how well it performs on unseen data.
- **Make Predictions** – Use the model to predict or classify new inputs.

Machine Learning vs Traditional Programming:

Traditional Programming	Machine Learning
Programmer writes rules	Machine learns rules from data
Fixed behavior	Adaptive behavior
Example: Calculator	Example: Email spam filter

Artificial Neural Network [ANN] (Cntd.)

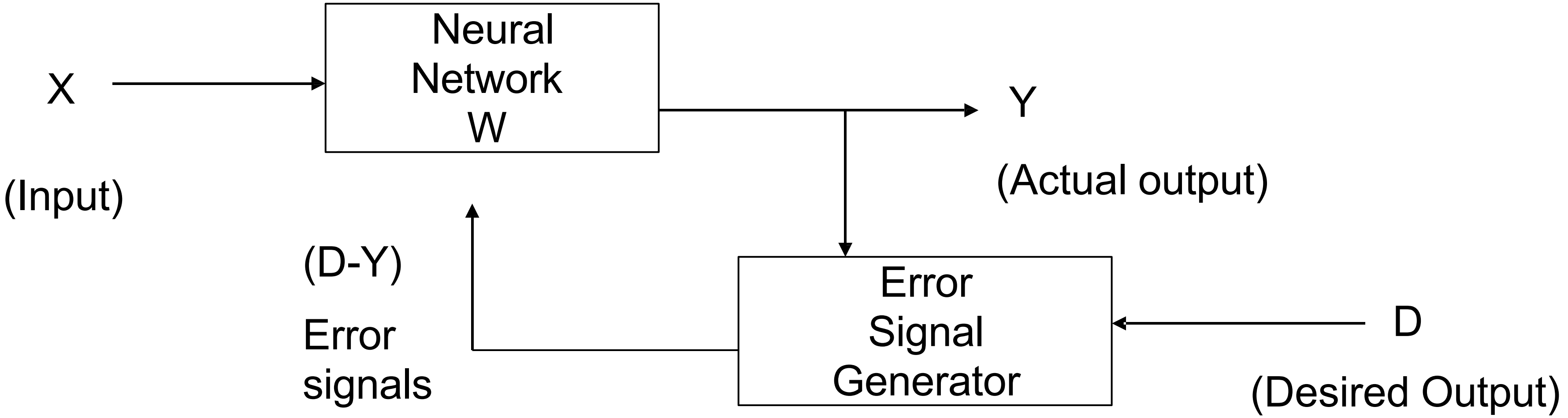
✓ Classification of learning

1. Supervised learning
2. Unsupervised learning
3. Semi-supervised learning
4. Reinforcement learning

Artificial Neural Network [ANN] (Cntd.)

✓ Classification of learning (Cntd.)

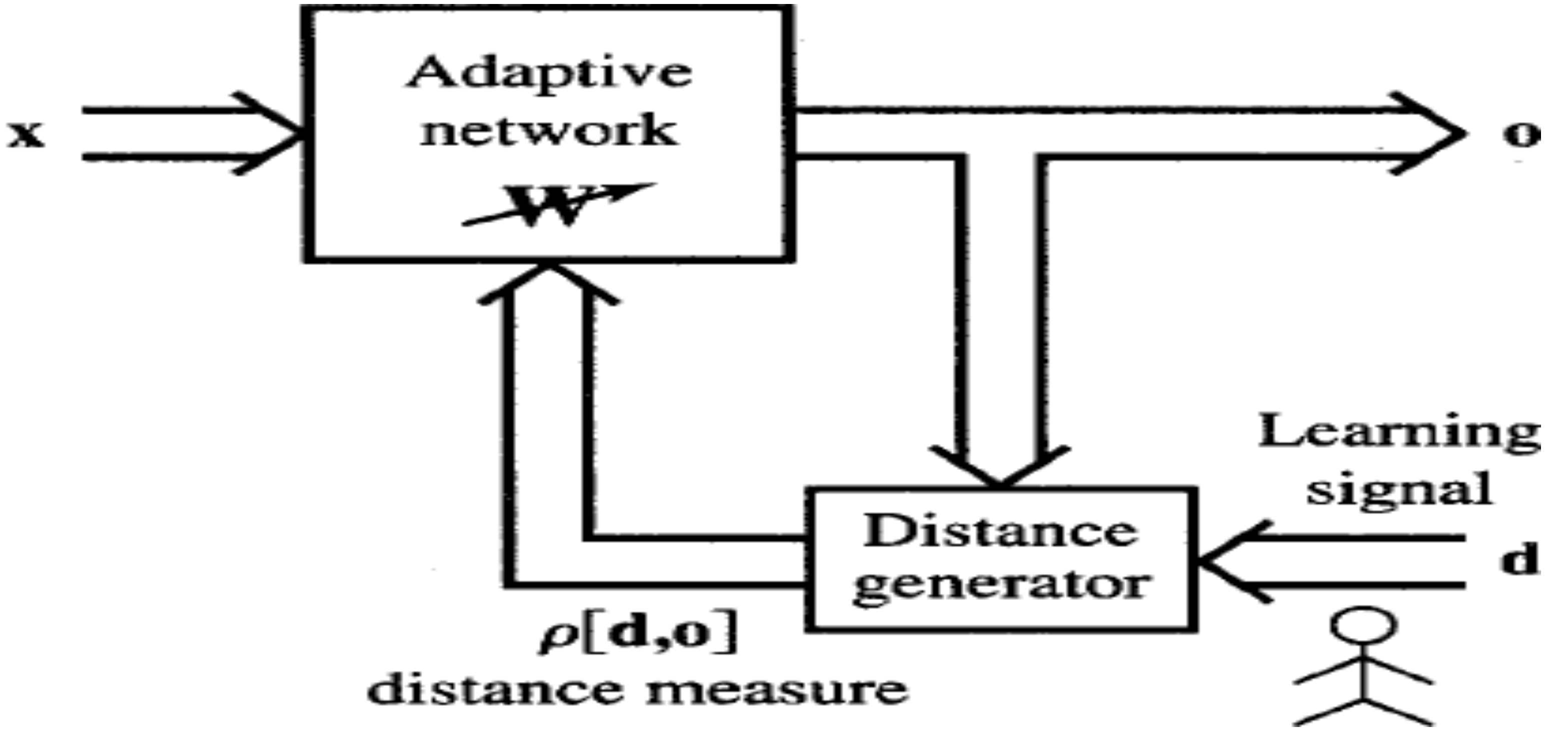
1. Supervised learning
   - Child learns from a teacher
   - Each input vector requires a corresponding target vector.
   - Training pair=[input vector, target vector]



Artificial Neural Network [ANN] (Cntd.)

✓ Classification of learning (Cntd.)

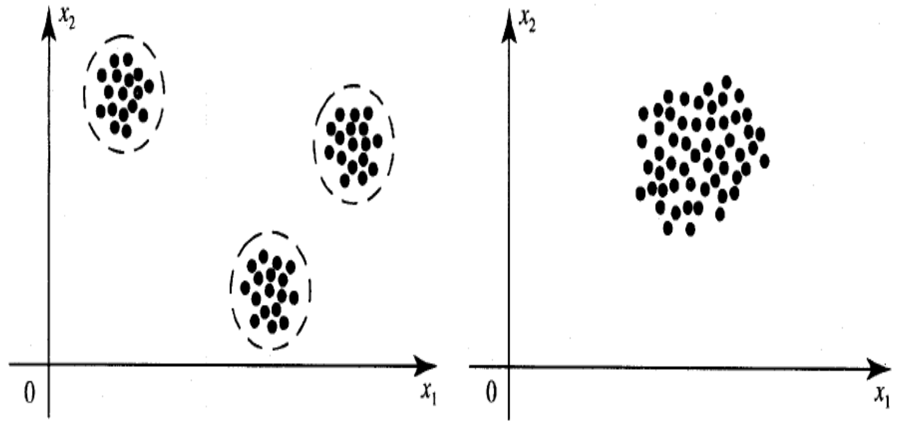
1. Supervised learning (Cntd.)
   - Supervised learning does minimization of error

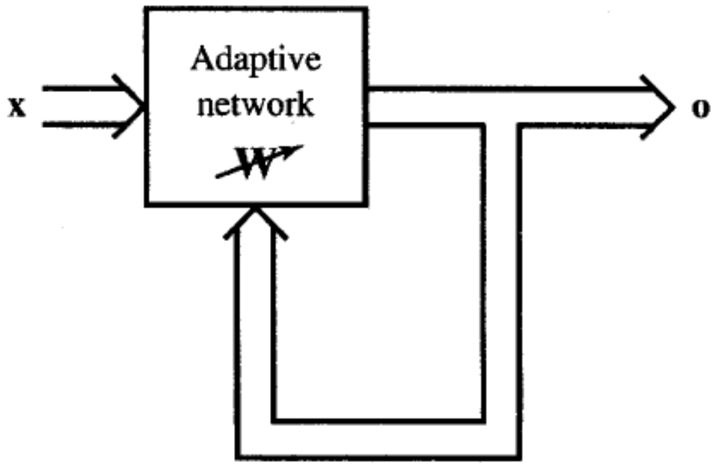


Artificial Neural Network [ANN] (Cntd.)

✓ <u>Classification of learning (Cntd)</u>

2. Unsupervised learning
    - How a fish or tadpole learns
    - All similar input patterns are grouped together as clusters.
    - If a matching input pattern is not found a new cluster is formed





Artificial Neural Network [ANN] (Cntd.)

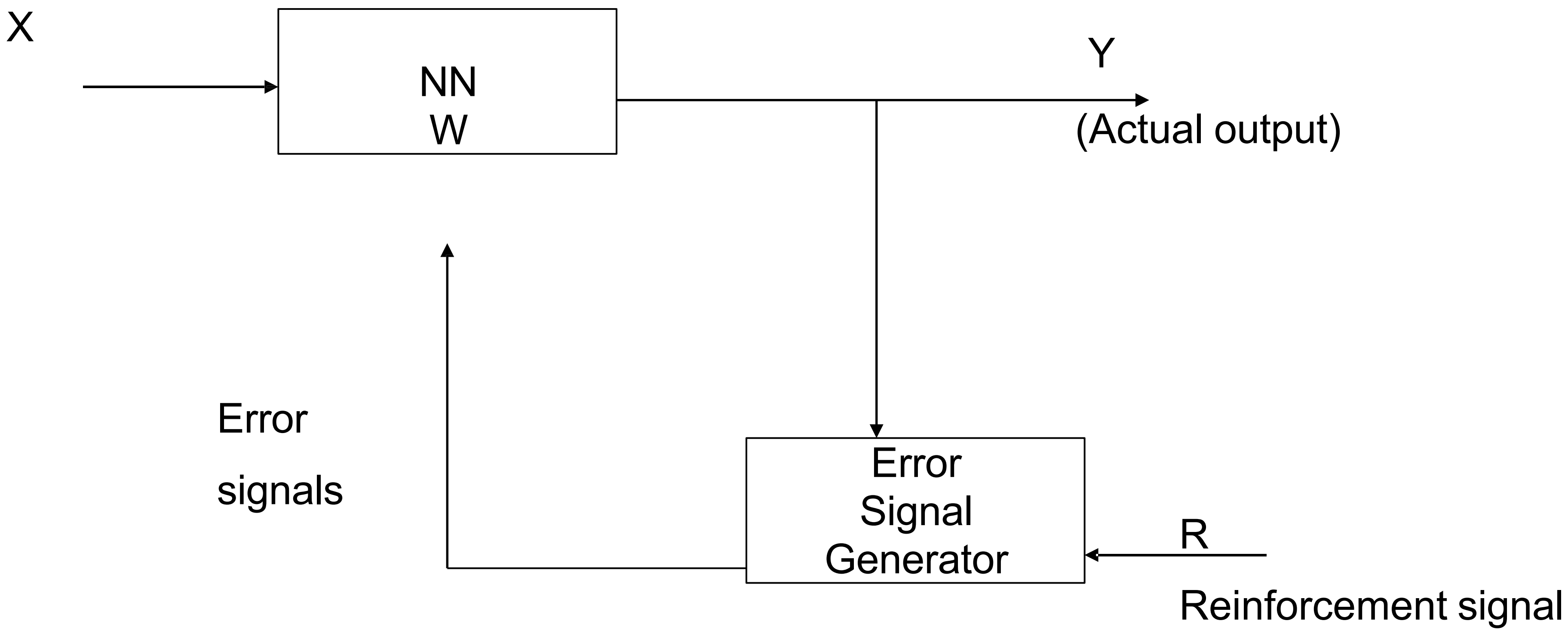
✓ Classification of learning (Cntd)

2. Unsupervised learning (Cntd.)
   - In unsupervised learning there is no feedback
   - Network must discover patterns, regularities, features for the input data over the output
   - While doing so the network might change in parameters
   - This process is called self-organizing

Artificial Neural Network [ANN] (Cntd.)

✓ <u>Classification of learning</u>

3. Reinforcement learning



Artificial Neural Network [ANN] (Cntd.)

✓ Classification of learning

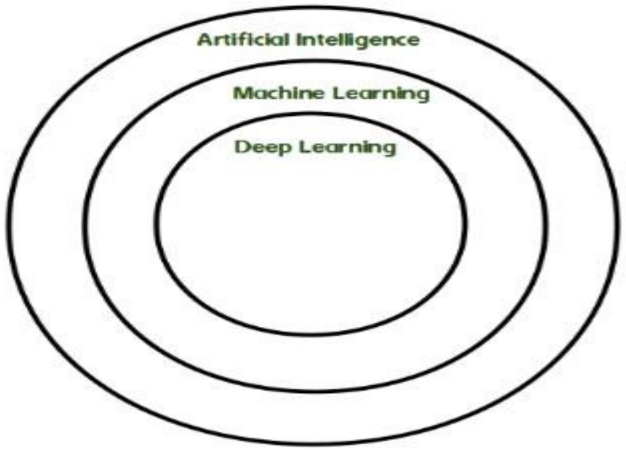
3. Reinforcement learning (Cntd.)
   - When Reinforcement learning is used?
     - If less information is available about the target output values (critic information)
     - Learning based on this critic information is called reinforcement learning and the feedback sent is called reinforcement signal
     - Feedback in this case is only evaluative and not instructive

Artificial Neural Network [ANN] (Cntd.)

- ✓ <u>Example learning algorithms</u>
  - ➢ Supervised:
    - Adaline, Madaline
    - Perceptron
    - Back Propagation
    - multilayer perceptrons
    - Radial Basis Function Networks
  - ➢ Unsupervised
    - Competitive Learning
    - Kohenen self organizing map
    - Learning vector quantization
    - Hebbian learning

Deep Learning

- ✓ Deep learning is a branch of machine learning which is completely based on artificial neural networks, as neural network is going to mimic the human brain so deep learning is also a kind of mimic of human brain. In deep learning, we don't need to explicitly program everything.
- ✓ Definition: Deep learning is a particular kind of machine learning that achieves great power and flexibility by learning to represent the world as a nested hierarchy of concepts, with each concept defined in relation to simpler concepts, and more abstract representations computed in terms of less abstract ones.



Deep Learning (Cntd.)

✓ Architectures :

- ➢ Deep Neural Network – It is a neural network with a certain level of complexity (having multiple hidden layers in between input and output layers). They are capable of modeling and processing non-linear relationships.
- ➢ Deep Belief Network(DBN) – It is a class of Deep Neural Network. It is multi-layer belief networks. Steps for performing DBN :
  1. Learn a layer of features from visible units using Contrastive Divergence algorithm.
  2. Treat activations of previously trained features as visible units and then learn features of features.
  3. Finally, the whole DBN is trained when the learning for the final hidden layer is achieved.
- ➢ Recurrent (perform same task for every element of a sequence) Neural Network – Allows for parallel and sequential computation. Similar to the human brain (large feedback network of connected neurons). They are able to remember important things about the input they received and hence enables them to be more precise.

Deep Learning (Cntd.)

✓ <u>Difference Between ML and DL:</u>

MACHINE LEARNING	DEEP LEARNING
Works on small amount of Dataset for accuracy.	Works on Large amount of Dataset.
Dependent on Low-end Machine.	Heavily dependent on High-end Machine.
Divides the tasks into sub-tasks, solves them individually and finally combine the results.	Solves problem end to end.
Takes less time to train.	Takes longer time to train.
Testing time may increase.	Less time to test the data.

Deep Learning (Cntd.)

- ✓ Advantages :
  - ➢ Best in-class performance on problems.
  - ➢ Reduces need for feature engineering.
  - ➢ Eliminates unnecessary costs.
  - ➢ Identifies defects easily that are difficult to detect.

- ✓ Disadvantages :
  - ➢ Large amount of data required.
  - ➢ Computationally expensive to train.
  - ➢ No strong theoretical foundation.

Deep Learning (Cntd.)

✓ <u>Applications :</u>

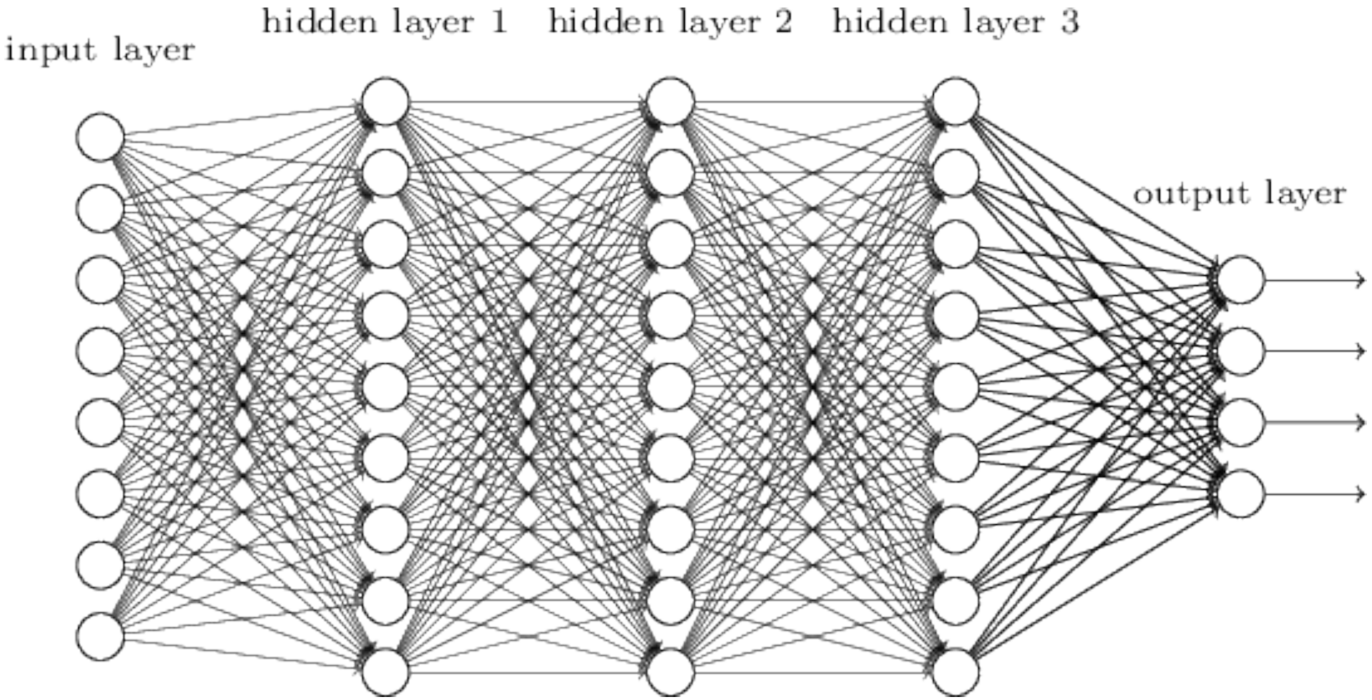
- ➢ Automatic Text Generation
- ➢ Healthcare
- ➢ Automatic Machine Translation
- ➢ Image Recognition
- ➢ Predicting Earthquakes

And many more…

Convolutional Neural Network (CNN)

Smaller Network: CNN

- We know it is good to learn a small model.
- From this fully connected model, do we really need all the edges?
- Can some of these be shared?

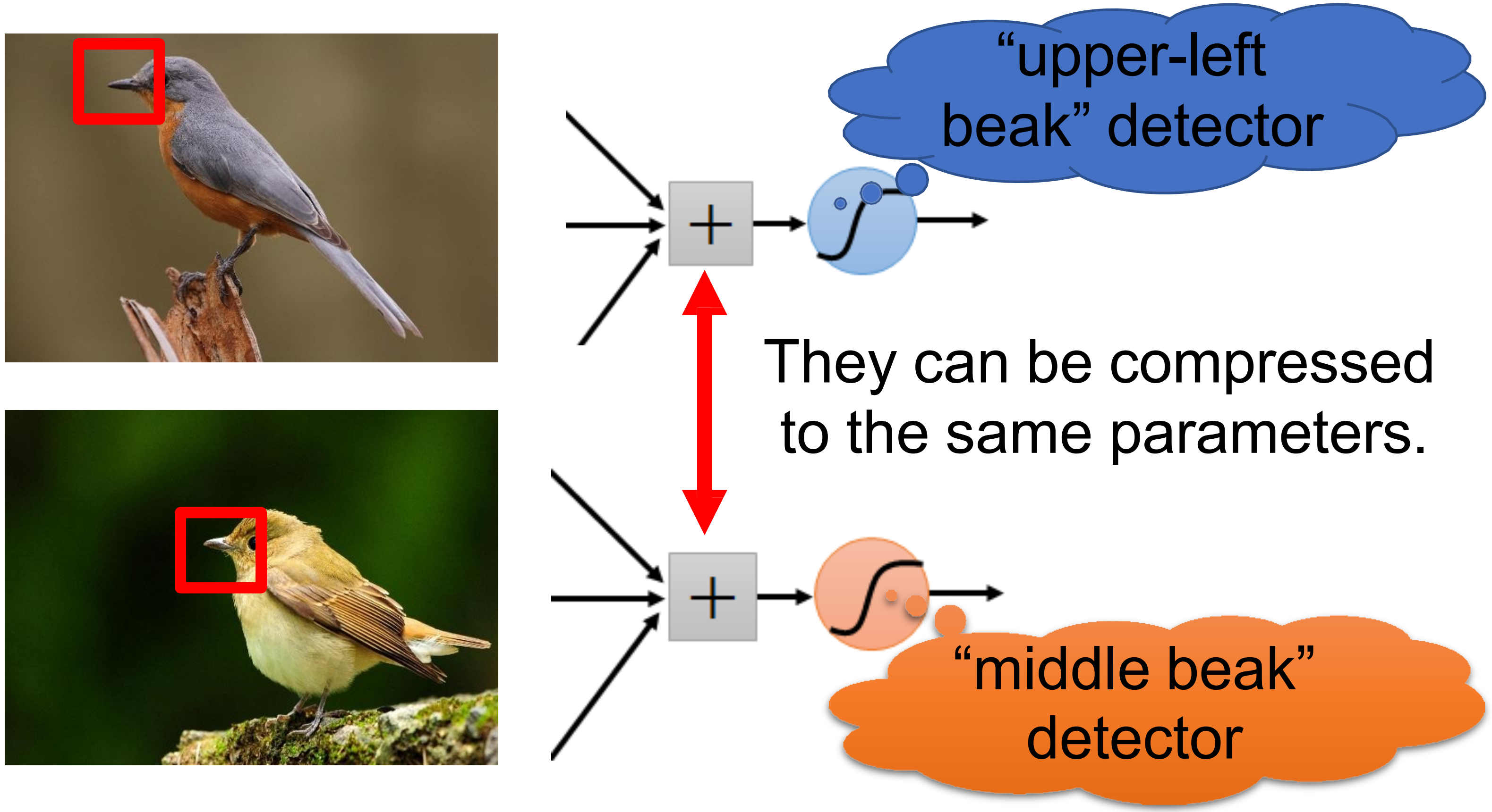


Consider learning an image:

- Some patterns are much smaller than the whole image

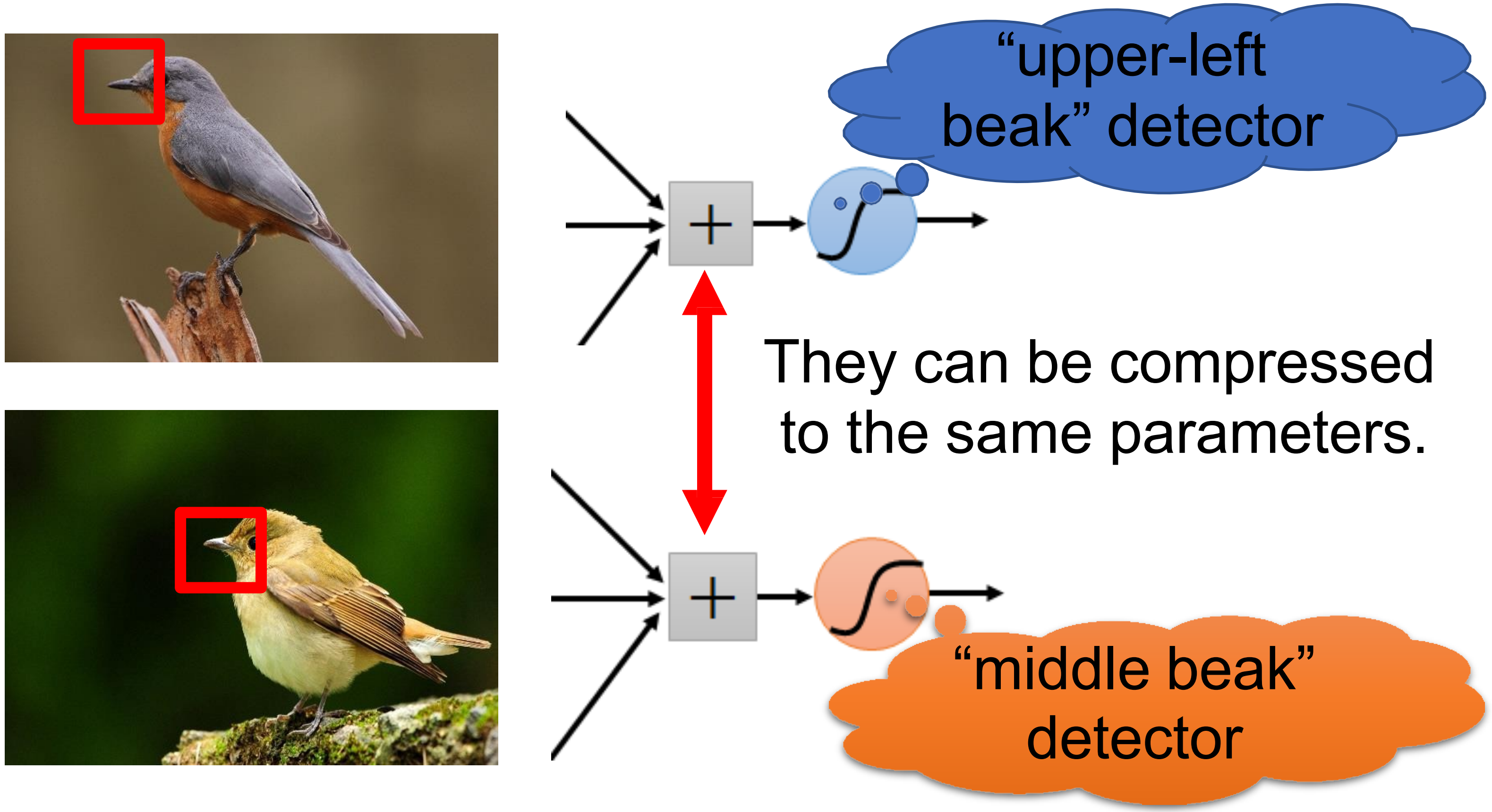
Can represent a small region with fewer parameters

Same pattern appears in different places: They can be compressed!
What about training a lot of such “small” detectors and each detector must “move around”.



Same pattern appears in different places: They can be compressed!

What about training a lot of such "small" detectors and each detector must "move around".



Convolution

These are the network parameters to be learned.

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

Filter 1

1	-1	-1
-1	1	-1
-1	-1	1

Filter 2

-1	1	-1
-1	1	-1
-1	1	-1

⋮ ⋮

Each filter detects a small pattern (3 x 3).

Convolution

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

stride=1

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

Dot product

3 -1

Convolution

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

If stride=2

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

3 -3

Convolution

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

stride=1

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

3	-1	-3	-1
-3	1	0	-3
-3	-3	0	1
3	-2	-2	-1

Convolution

-1	1	-1
-1	1	-1
-1	1	-1

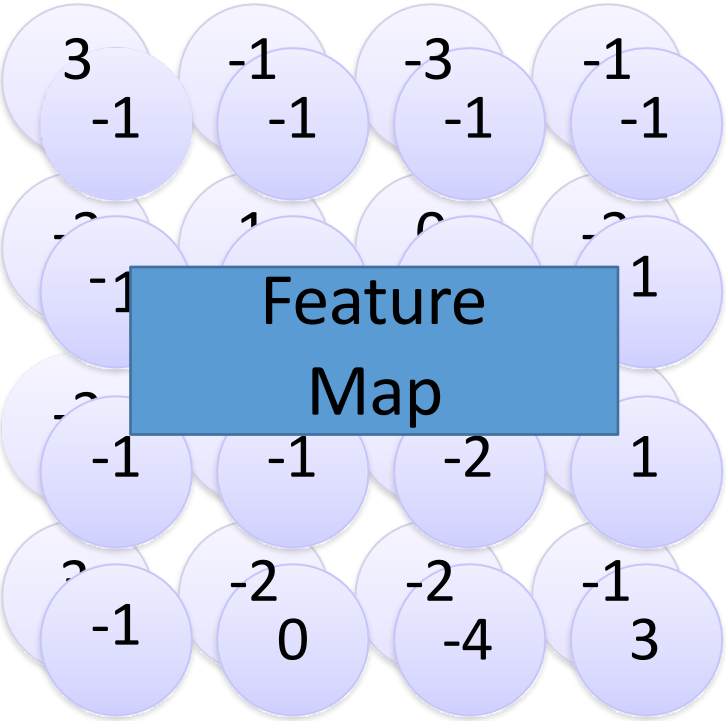
Filter 2

stride=1

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

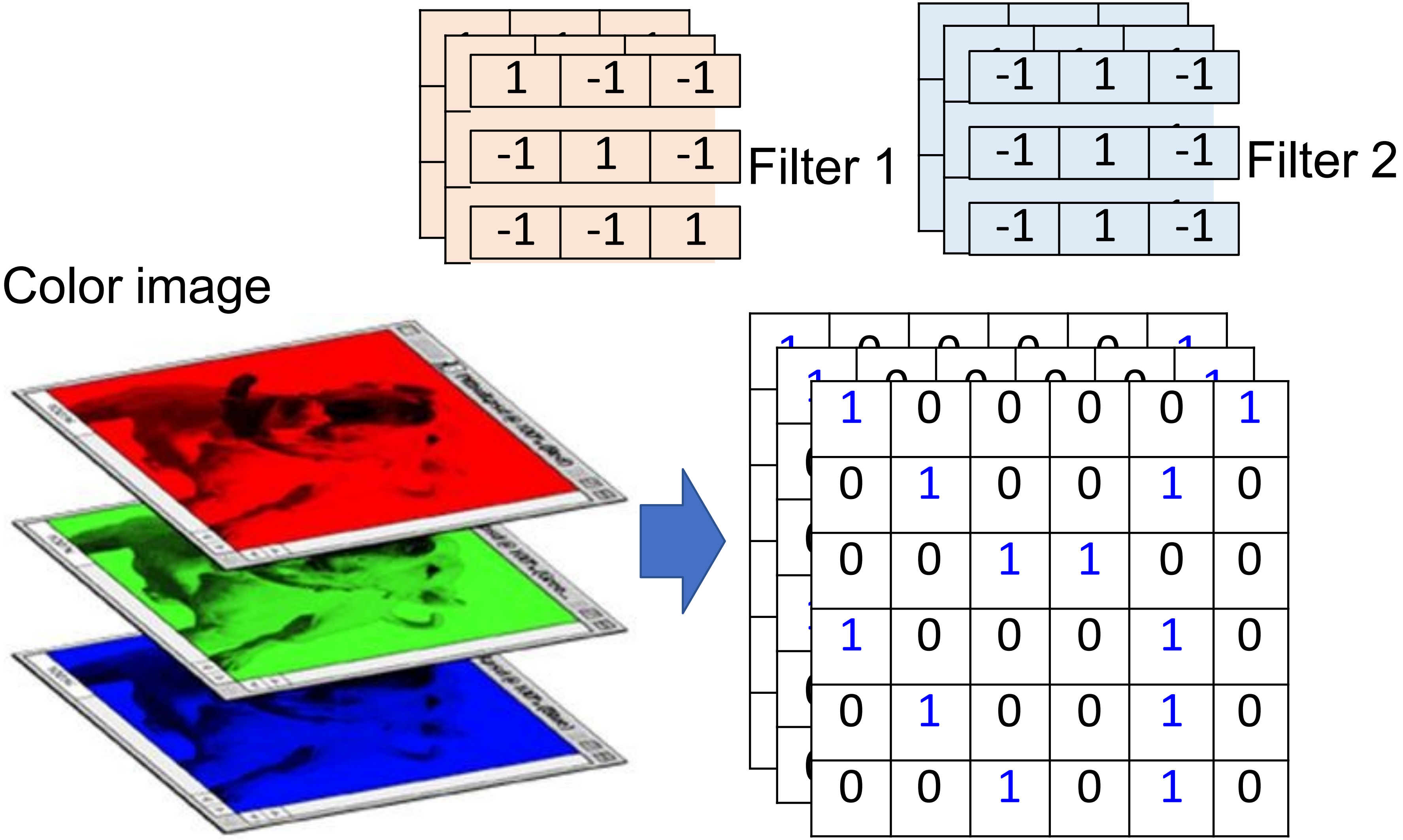
6 x 6 image

Repeat this for each filter

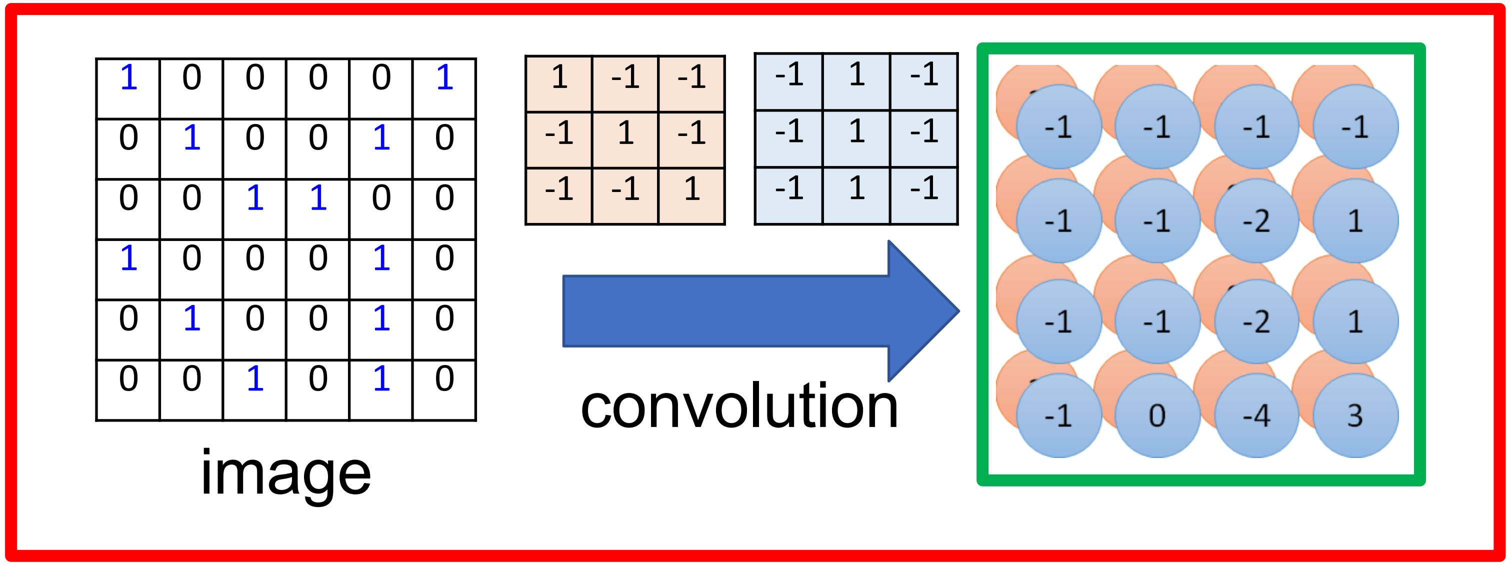


Two 4 x 4 images
Forming 2 x 4 x 4 matrix

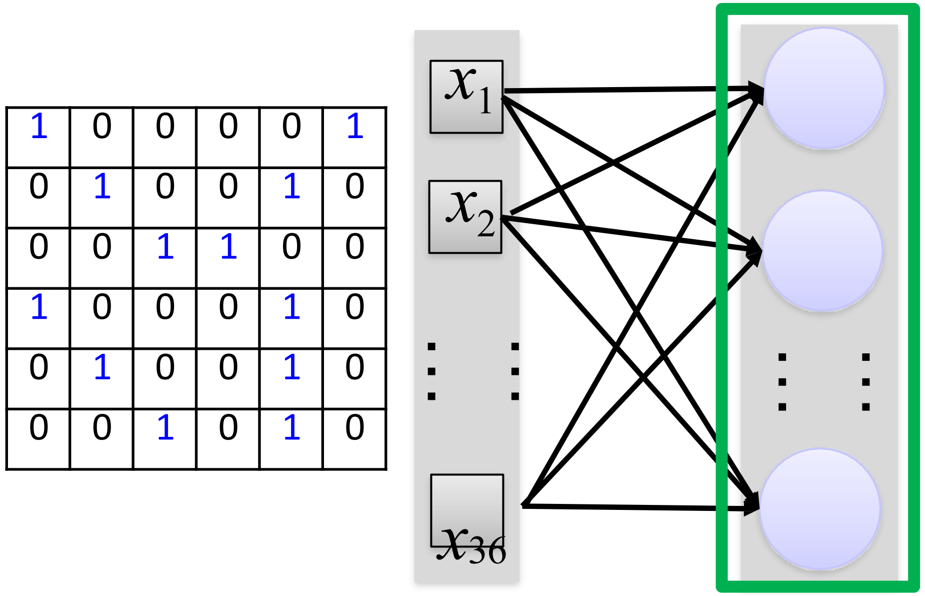
Color image: RGB 3 channels

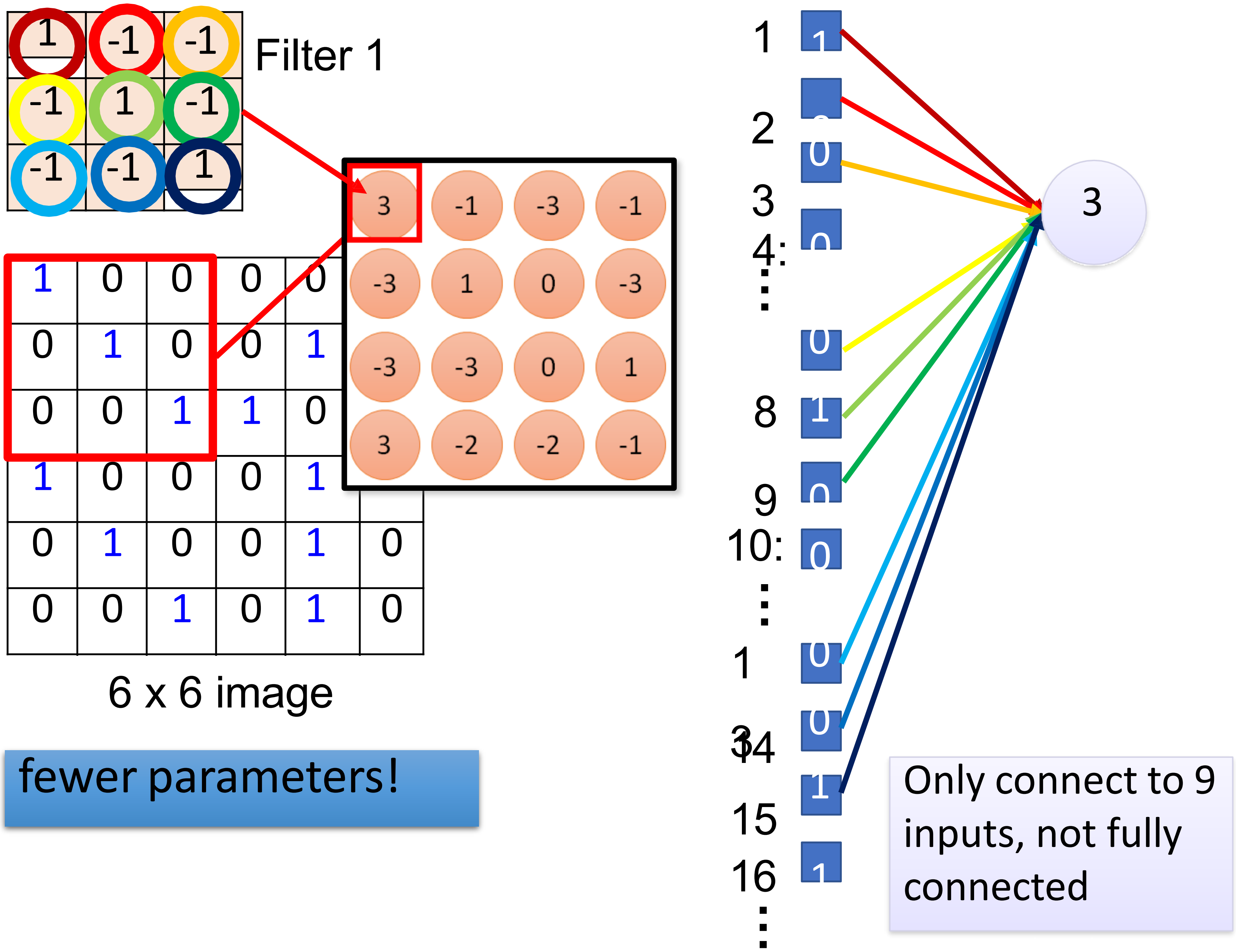


Convolution v.s. Fully Connected

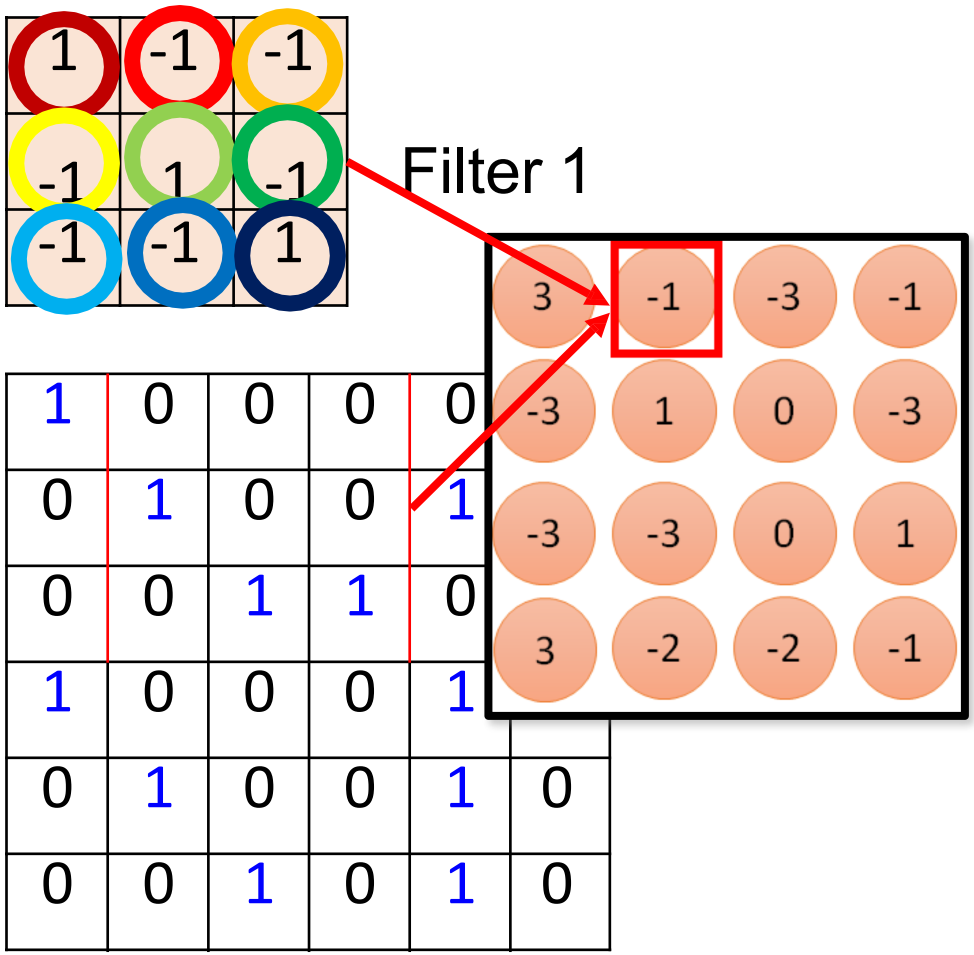


Fully-connected





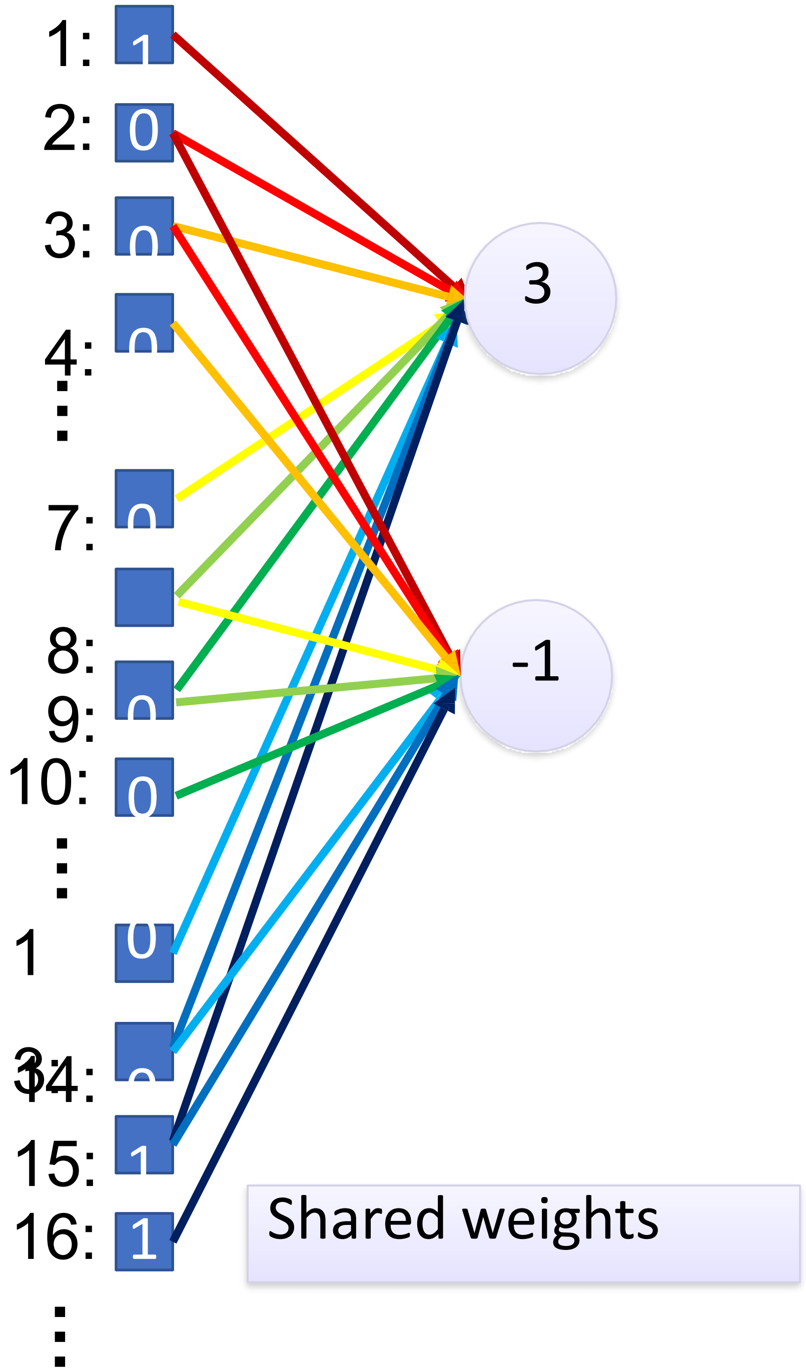
Filter 1
6 x 6 image
fewer parameters!
Only connect to 9 inputs, not fully connected



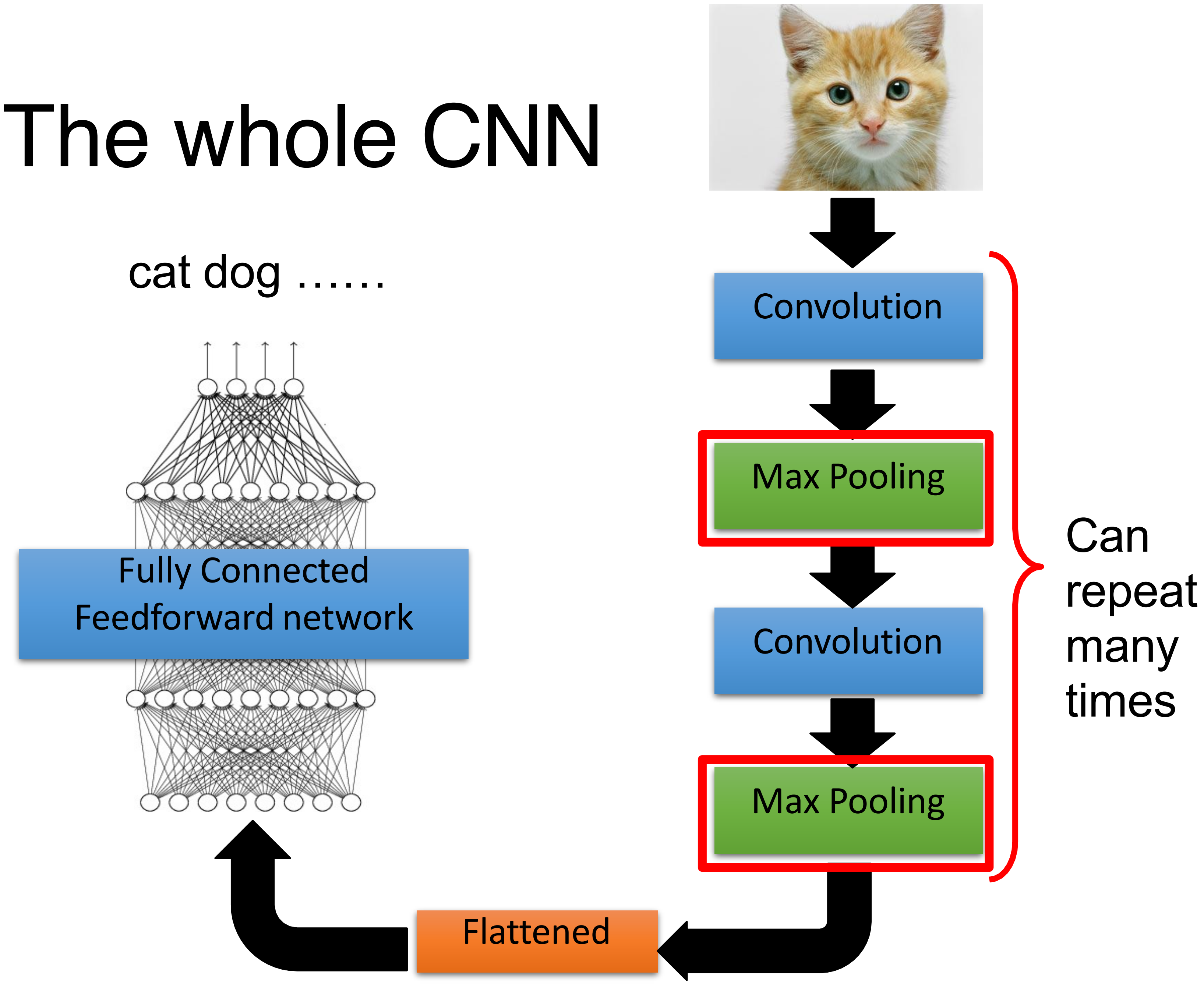
6 x 6 image

Fewer parameters

Even fewer parameters



The whole CNN



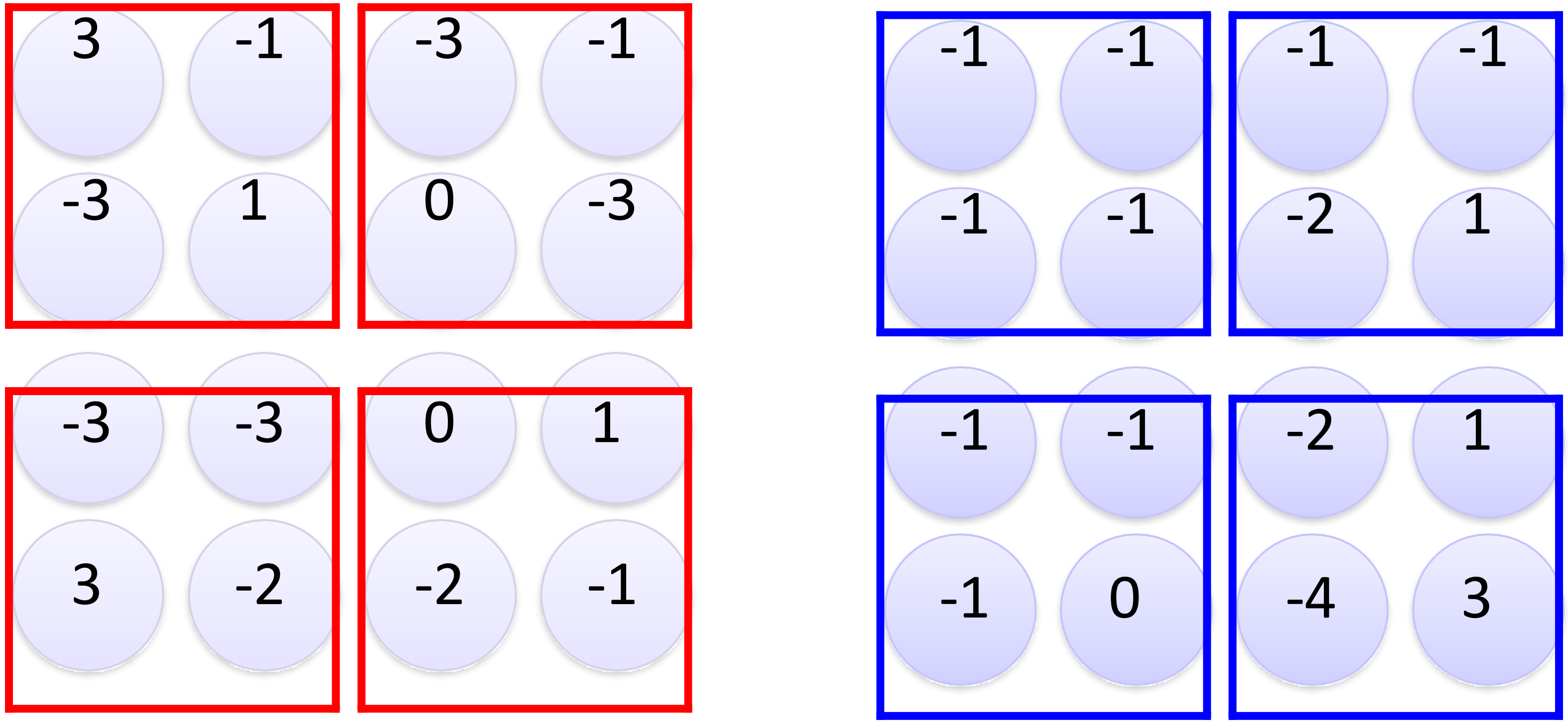
Max Pooling

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

-1	1	-1
-1	1	-1
-1	1	-1

Filter 2



Why Pooling

- Subsampling pixels will not change the object

bird

Subsampling

bird

We can subsample the pixels to make image

smaller

fewer parameters to characterize the image

A CNN compresses a fully connected network in two ways:

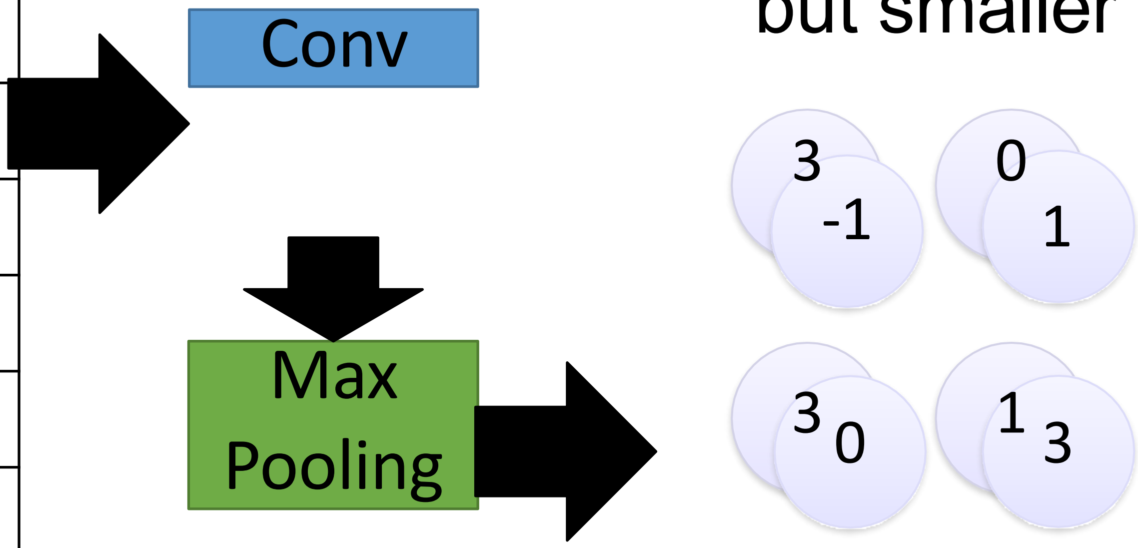
- Reducing number of connections
- Shared weights on the edges
- Max pooling further reduces the complexity

Max Pooling

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

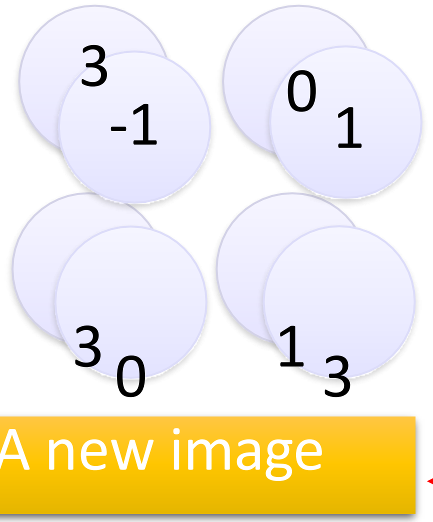
New image but smaller

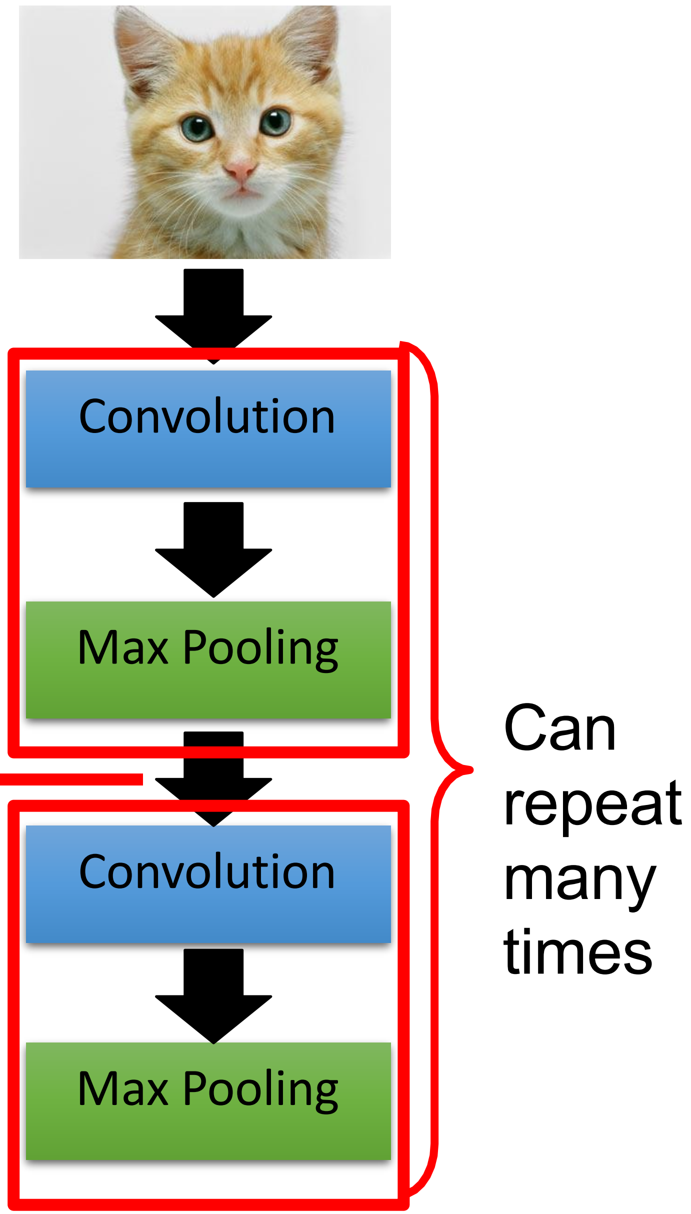


2 x 2 image

Each filter is a channel

The whole CNN

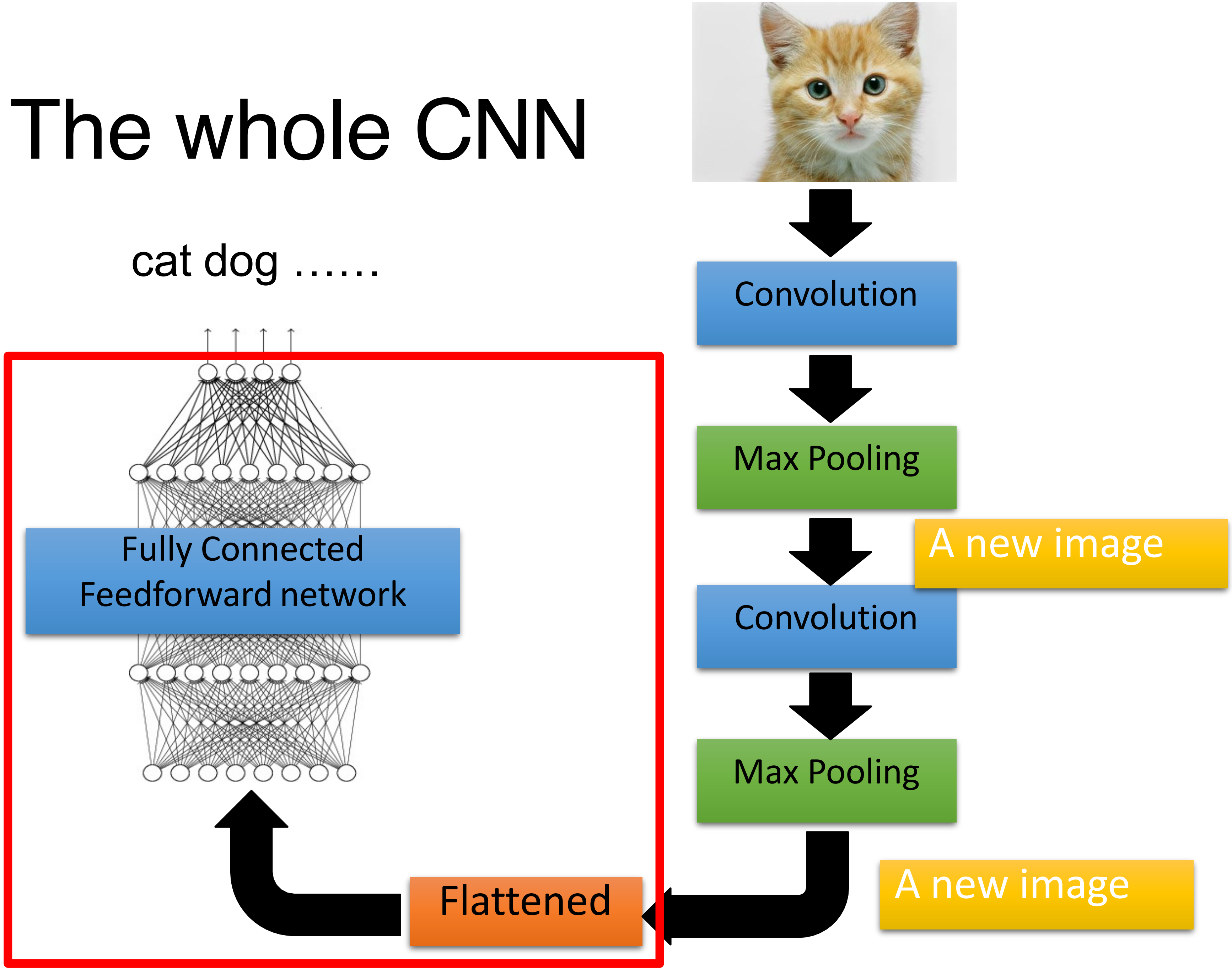




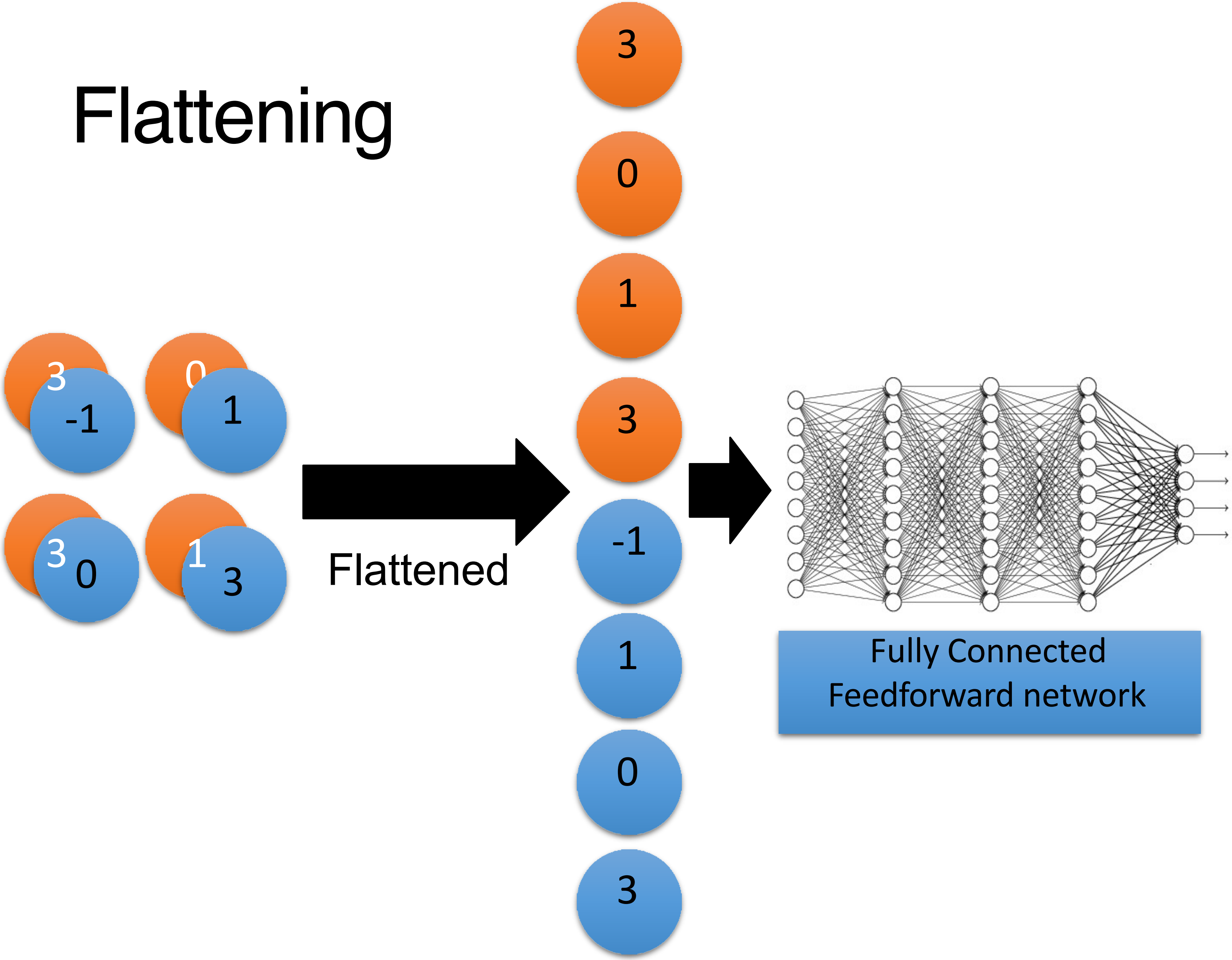
Smaller than the original image

The number of channels is the number of filters

The whole CNN



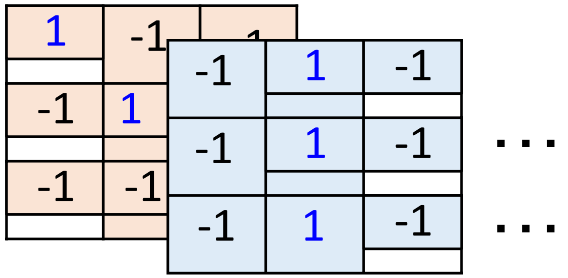
Flattening


3
-1
0
1
3
0
1
3
Flattened
3
0
1
3
-1
1
0
3
Fully Connected
Feedforward network

CNN in Keras

Only modified the ***network structure*** and ***input format (vector -> 3-D tensor)***

```
model2.add( Convolution2D( 25,3,3,
            input_shape=(28,28,1)) )
```



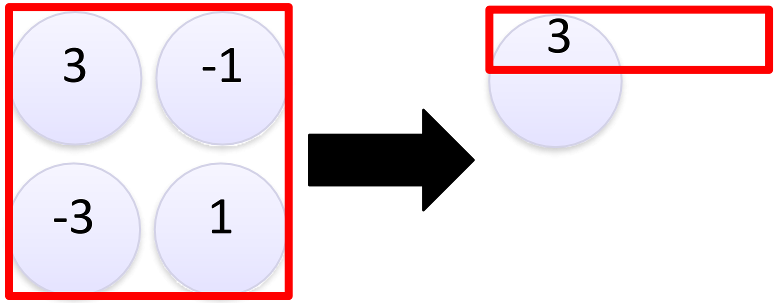
There are 25 3x3 filters.

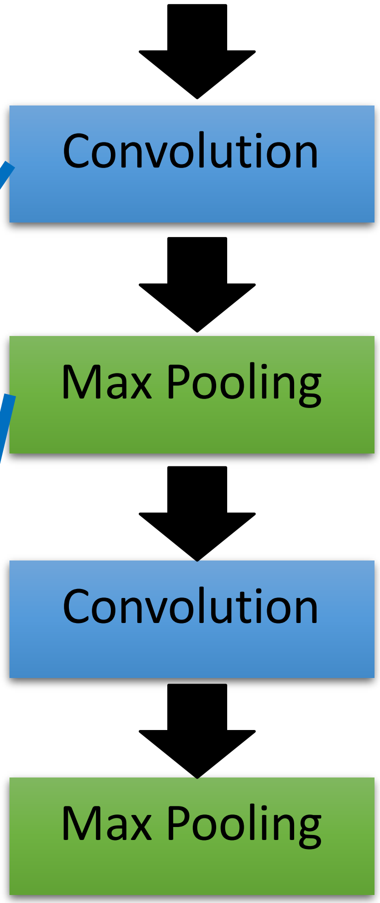
Input_shape = (28 , 28 , 1)

28 x 28 pixels

1: black/white, 3: RGB

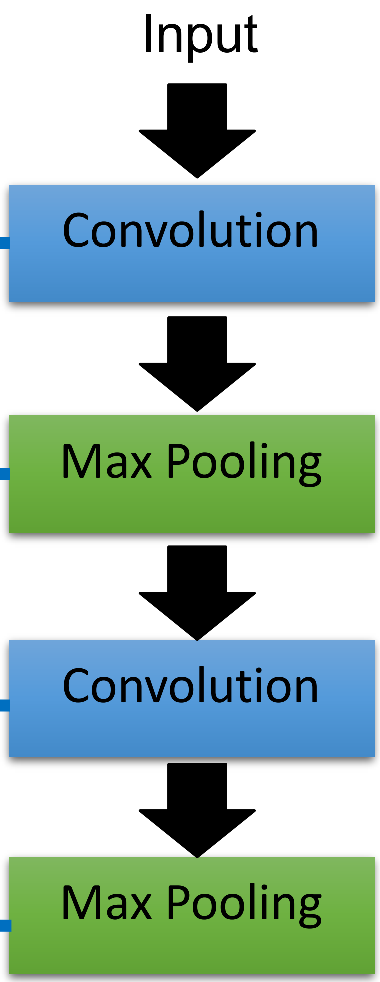
```
model2.add(MaxPooling2D((2,2)))
```





CNN in Keras

Only modified the ***network structure*** and ***input format (vector -> 3-D array)***



1 x 28 x 28

```
model2.add( Convolution2D( 25,3,3,
            input_shape=(28,28,1)) )
```

How many parameters for each filter? 9

25 x 26 x 26

```
model2.add(MaxPooling2D((2,2)))
```

25 x 13 x 13

```
model2.add(Convolution2D(50,3,3))
```

How many parameters for each filter? 225= 25x9

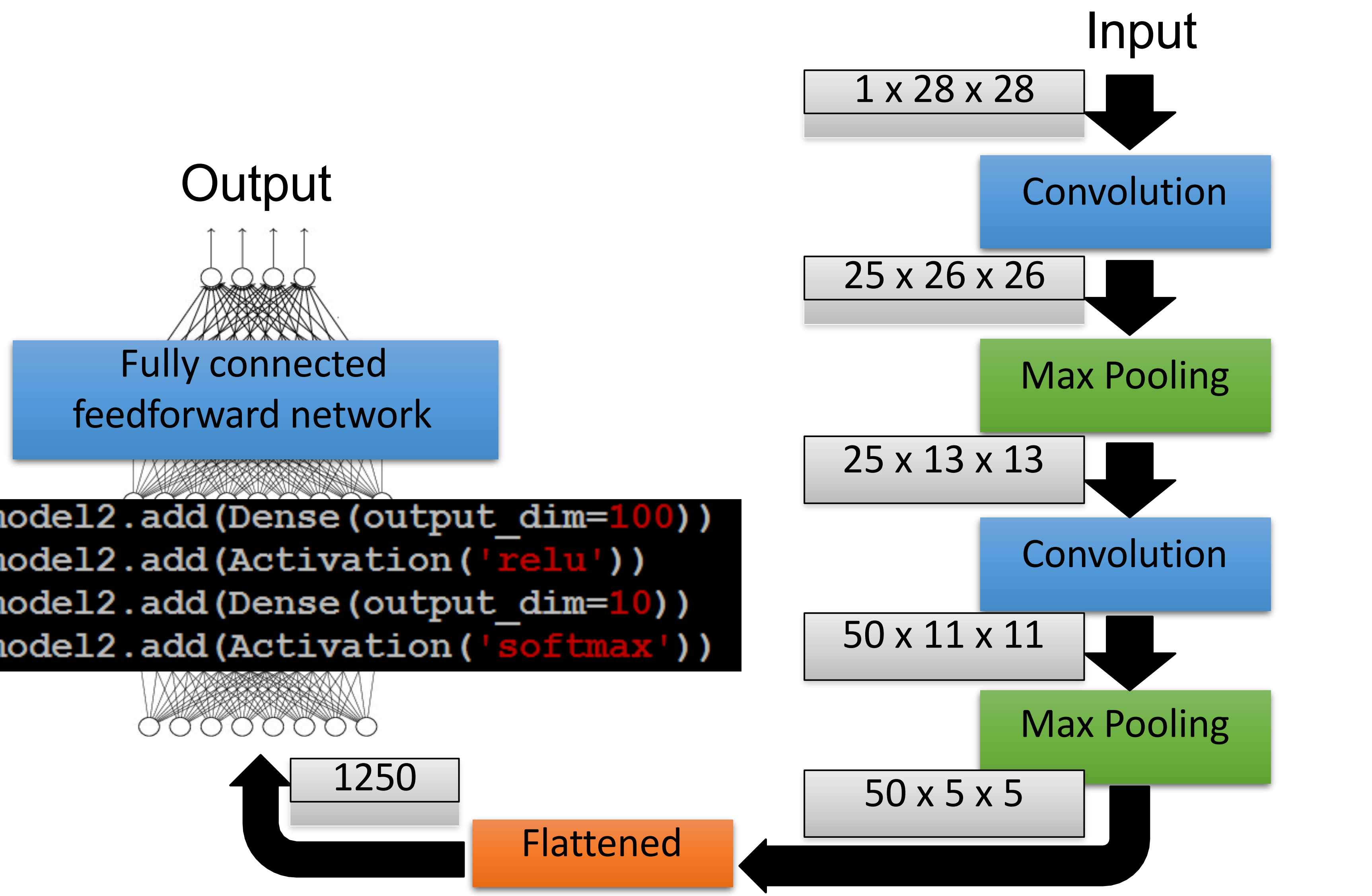
50 x 11 x 11

```
model2.add(MaxPooling2D((2,2)))
```

50 x 5 x 5

CNN in Keras

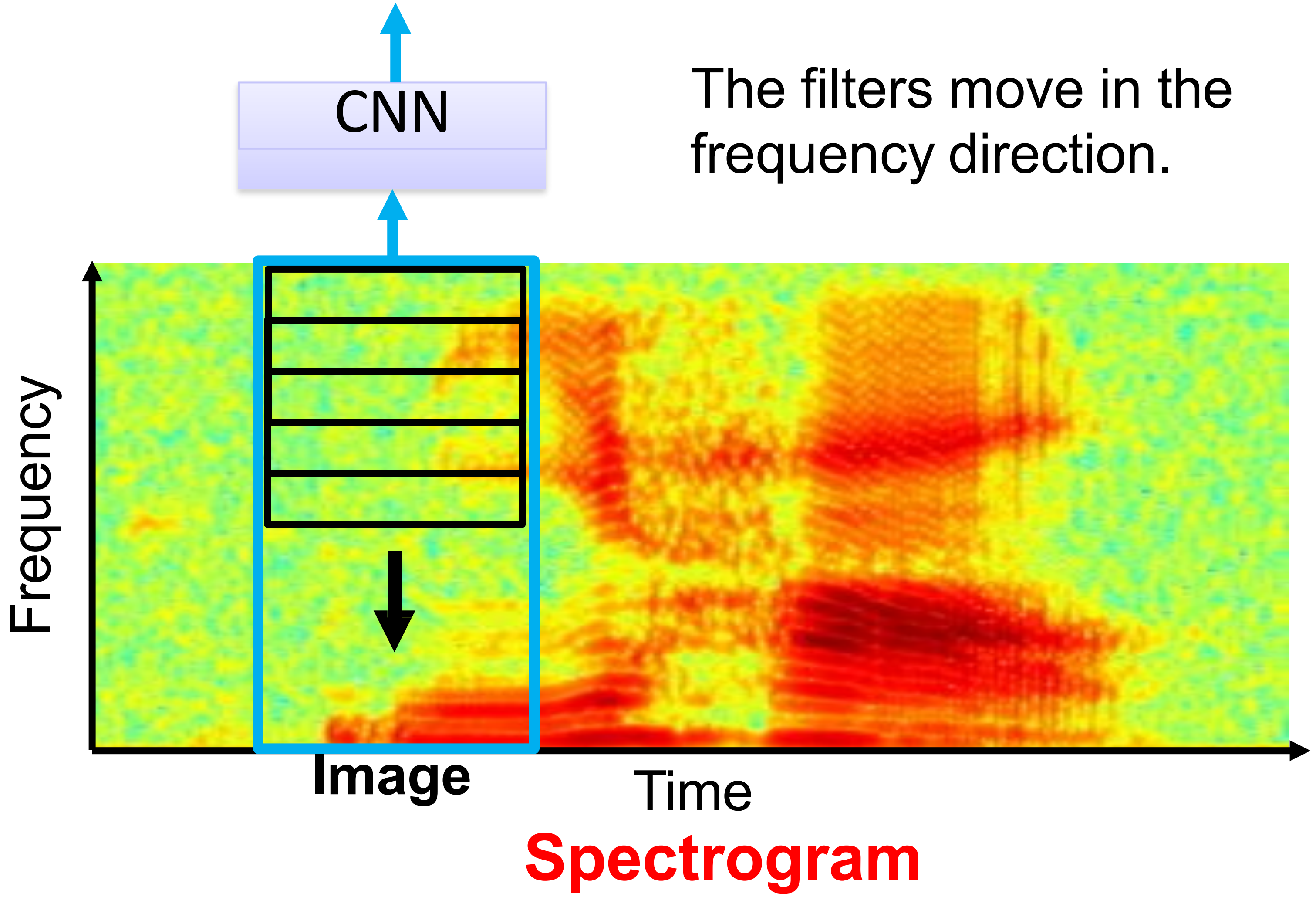
Only modified the ***network structure*** and ***input format (vector -> 3-D array)***




```
model2.add(Dense(output_dim=100))
model2.add(Activation('relu'))
model2.add(Dense(output_dim=10))
model2.add(Activation('softmax'))
```

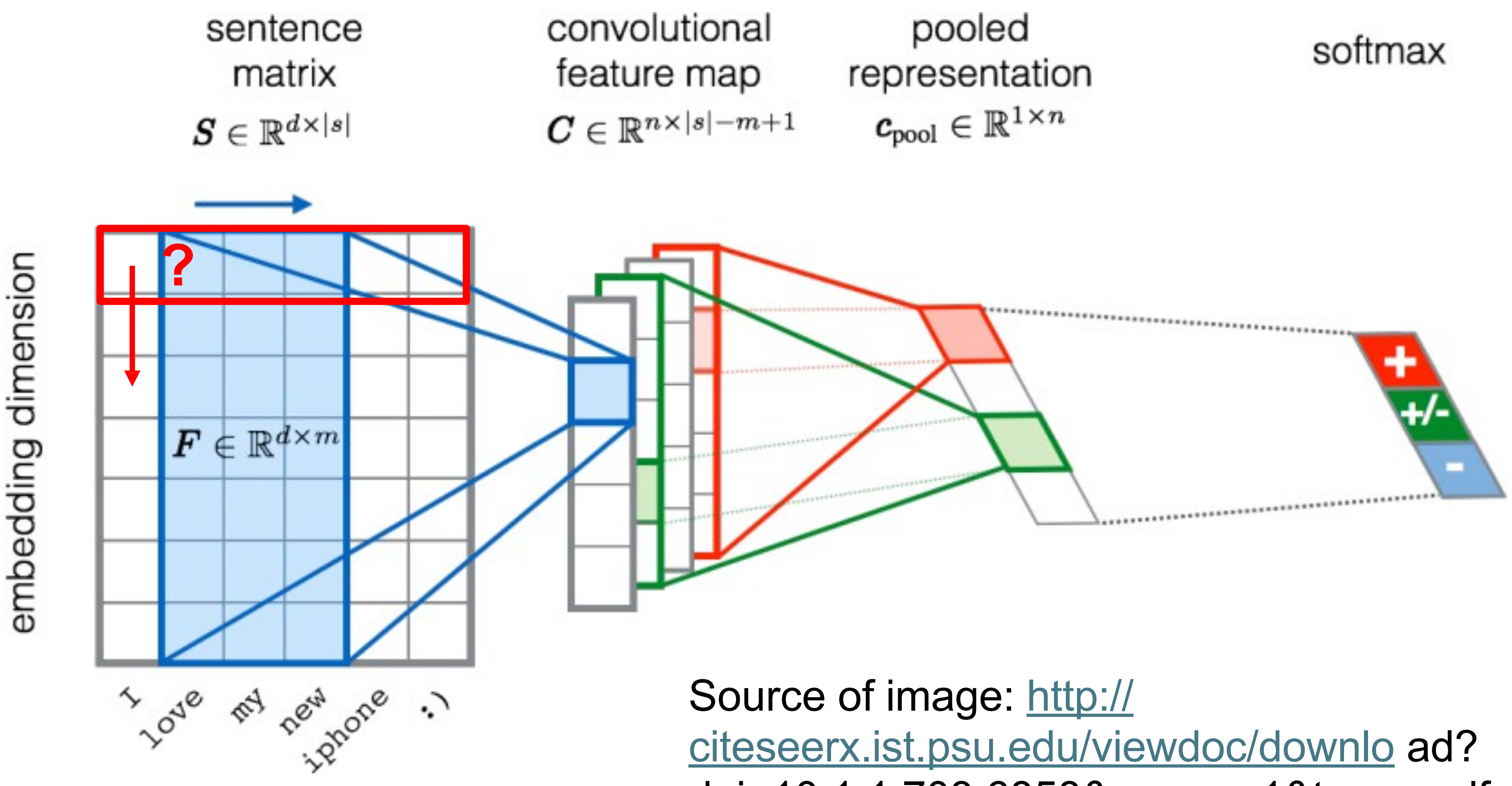
```
model2.add(Flatten())
```

CNN in speech recognition



The filters move in the frequency direction.

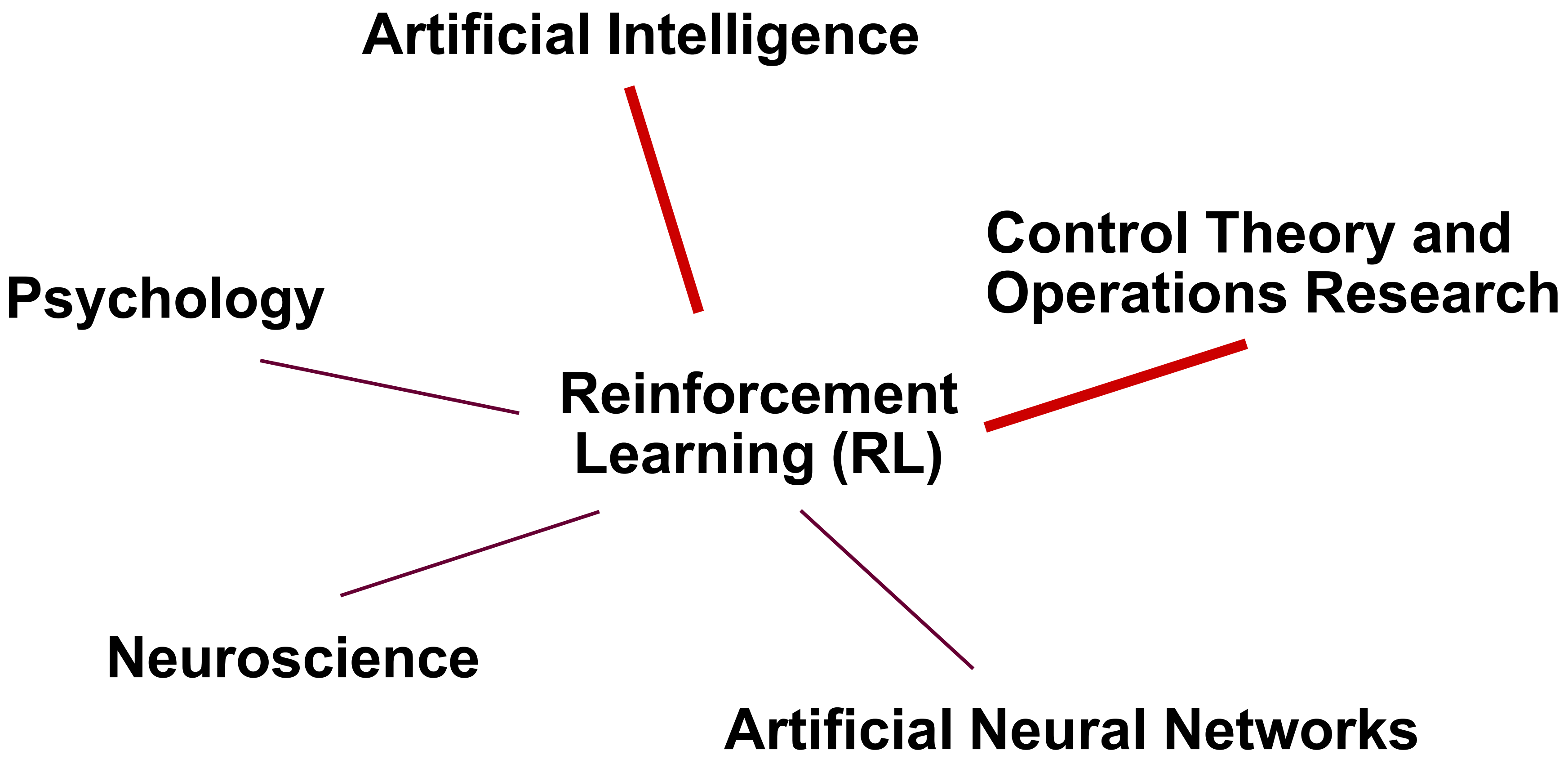
CNN in text classification



Source of image: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.703.6858&rep=rep1&type=pdf

Reinforcement Learning

Learning from Experience Plays a Role in …



Reinforcement learning

- Reinforcement learning (RL) is a machine learning paradigm where an agent learns to make decisions in an environment to maximize a cumulative reward. Unlike supervised learning, RL doesn't rely on labeled data, and unlike unsupervised learning, it has a defined goal (the reward). The agent learns through trial and error, receiving feedback (rewards or penalties) for its actions, and adjusts its behavior to achieve the optimal outcome.

Here's a more detailed explanation:
Core Concepts:

- **Agent:** The entity that learns and makes decisions.
- **Environment:** The world in which the agent operates.
- **Actions:** The choices the agent can make within the environment.
- **Reward:** Feedback from the environment indicating the success or failure of an action.
- **Policy:** A strategy that the agent uses to determine which action to take in a given state.
- **Value Function:** Estimates the long-term cumulative reward for being in a particular state or taking a specific action.

How it works:

- The agent starts in an initial state within the environment.
- The agent takes an action based on its current policy.
- The environment provides feedback (reward or penalty) to the agent based on its action.
- The agent updates its policy based on the feedback, aiming to maximize future rewards.
- This process repeats, with the agent continuously learning and improving its decision-making process.

Key Differences from other ML paradigms:

Supervised Learning:

- RL doesn't require labeled data for training. The agent learns through interaction and feedback.

Unsupervised Learning:

- RL has a clear goal in the form of a reward function, unlike unsupervised learning which explores data without a specific target.

Applications:

Reinforcement learning is used in various applications, including:

- Game playing (e.g., chess, Go)
- Robotics (e.g., robot navigation, manipulation)
- Resource management (e.g., traffic control, energy management)
- Personalized recommendations
- Drug discovery

What is Deep Reinforcement Learning?

- Deep reinforcement learning is standard reinforcement learning where a deep neural network is used to approximate either a policy or a value function
- Deep neural networks require lots of real/simulated interaction with the environment to learn
- Lots of trials/interactions is possible in simulated environments
- We can easily parallelise the trials/interaction in simulated environments
- We cannot do this with robotics (no simulations) because action execution takes time, accidents/failures are expensive and there are safety concerns

Natural Language Processing

Natural Language Processing

- NLP (Natural Language Processing) is a branch of Artificial Intelligence (AI) that enables computers to understand, interpret, generate, and respond to human language.

- NLP is the technology that helps machines **read, understand, and communicate** in **natural human language**, like English, Hindi, or any spoken language.

Motivation for NLP

- Understand language analysis & generation
- Communication
- Language is a window to the mind
- Data is in linguistic form
- Data can be in Structured (table form), Semi structured (XML form), Unstructured (sentence form).

Example Use Cases of NLP

Use Case	Example
🔍 Search Engines	Google understanding your query intent
💬 Chatbots / Virtual Assistants	Siri, Alexa, ChatGPT
📧 Email Filtering	Detecting spam vs. non-spam
😊 Sentiment Analysis	Classifying tweets as positive or negative
🌍 Translation	Google Translate (English ⇄ Hindi, etc.)
🧠 Text Summarization	Summarizing news articles
🎙️ Speech Recognition	Converting voice to text (dictation software)

Why is NLP Challenging?

Human language is:

- **Ambiguous**: "I saw a man on a hill with a telescope." (multiple meanings)
- **Context-sensitive**: "He is *cool*" (could mean calm or stylish)
- **Rich in grammar**: Variations in sentence structure, tenses, and idioms


Levels of NLP
Bhargav Joshi ©
01 PHONOLOGY LEVEL
02 MORPHOLOGICAL LEVEL
03 LEXICAL LEVEL
04 SYNTACTIC LEVEL
05 SEMANTICS LEVEL
06 DISCLOSURE LEVEL
07 PRAGMATIC LEVEL

NLP Involves Multiple Levels

Level	Focus	Example
Phonology	Sounds of language (speech-to-text)	"Can you hear me?"
Morphology	Word formation	“Unbreakable” = un + break + able
Lexical	Meaning of individual words	POS tagging: "run" = verb or noun?
Syntax	Grammar and structure	Sentence parsing
Semantics	Literal meaning of words and phrases	Understanding "bank" = river/finance
Discourse	Meaning across sentences	Resolving “he”, “she”, “it” references
Pragmatics	Intended meaning in context	"Can you pass the salt?" → a request

Levels

1. Phonological Level *(Speech-based NLP)*

- **Deals with:** Sound patterns and pronunciation.
- **Example:** Speech-to-text conversion (e.g., Siri, Google Assistant).
- **Tools Used:** Speech recognition APIs, HMMs, DeepSpeech.

2. Morphological Level

- **Deals with:** Structure and formation of words (prefix, suffix, root).
- **Example:**
    - Word: *unhappiness*
        - Prefix: *un-*
        - Root: *happy*
        - Suffix: *-ness*
- **Use:** Lemmatization, stemming.
- **Tools:** spaCy, NLTK

3. Lexical Level

- **Deals with:** Understanding the meaning and category of individual words.
- **Processes:**
  - Tokenization (splitting text into words)
  - POS (Part-of-Speech) tagging
- **Example:** “Run” can be a **verb** or **noun** depending on context.

4. Syntactic Level (Parsing)

- **Deals with:** Grammar and sentence structure.
- **Processes:**
  - Parsing sentences
  - Analyzing grammar (subject, object, verb)
- **Example:**
  - “The cat sat on the mat.” — Correct syntax
  - “Sat the cat mat on.” — Incorrect syntax
- **Tools:** Dependency parsers, CFGs

5. Semantic Level

- **Deals with:** Literal meaning of words and sentences.
- **Tasks:**
  - Word Sense Disambiguation
  - Named Entity Recognition
- **Example:**
  - “Bank” could mean a financial institution or river bank — depends on context.

6. Discourse Level

- **Deals with:** Meaning across multiple sentences or the whole paragraph.
- **Use:**
  - Resolving references (e.g., pronouns: *he, she, it*)
  - Coherence in conversation
- **Example:**
  - “Ram went to the market. He bought apples.”
    → “He” refers to “Ram.”

7. Pragmatic Level

- **Deals with:** Meaning based on **context and world knowledge**.
- **Use:** Understanding *intended meaning*.
- **Example:**
  - Sentence: “Can you pass the salt?”
    - Literally a question, but **pragmatically** a request.

Summary Table

Level	Focus Area	Example
Phonological	Sounds and pronunciation	Speech-to-text systems
Morphological	Word formation	"Unhappiness" = un + happy + ness
Lexical	Word meaning and category	POS tagging
Syntactic	Sentence grammar	Subject-Verb-Object parsing
Semantic	Literal meaning	“Bank” = financial or river
Discourse	Context across sentences	Resolving pronouns
Pragmatic	Intended meaning from context	“Can you pass the salt?” → request

Common NLP Tools and Libraries

Tool/Library	Purpose
NLTK	Educational NLP toolkit (Python)
spaCy	Fast, production-ready NLP
TextBlob	Simple sentiment analysis
Hugging Face	Transformer-based NLP models
Stanford NLP	Java-based NLP tools

Computer Vision

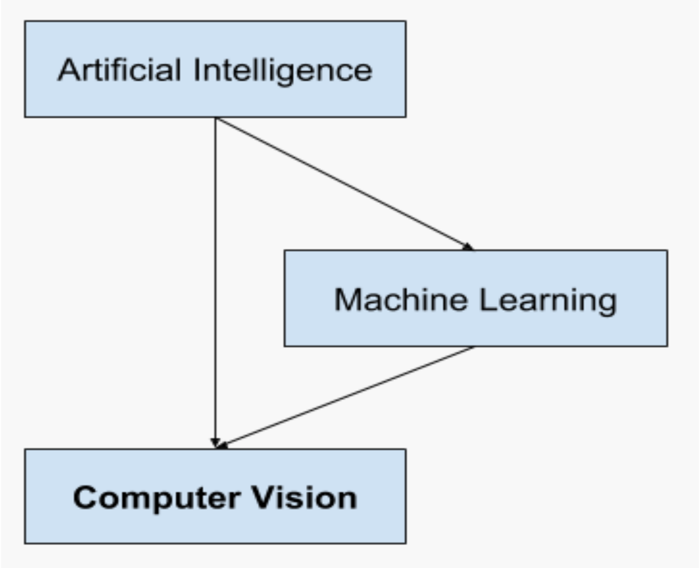
- **Computer Vision** is the field of **Artificial Intelligence (AI)** that enables computers to **see, understand, and interpret images and videos**, much like the human visual system.

Goal of Computer Vision

- To **extract meaningful information** from **visual inputs** (images, video, real-time feeds), and use it for decision-making, classification, segmentation, etc.

Computer Vision

- ✓ Computer Vision, often abbreviated as CV, is defined as a field of study that seeks to develop techniques to help computers "see" and understand the content of digital images such as photographs and videos.
- ✓ At an abstract level, the goal of computer vision problems is to use the observed image data to infer something about the world.
- ✓ It is a multidisciplinary field that could broadly be called a subfield of artificial intelligence and machine learning, which may involve the use of specialized methods and make use of general learning algorithms.



Key Tasks in Computer Vision

Task	Description	Example
📷 **Image Classification**	Predict the category of an image	Cat vs. Dog
🧱 **Object Detection**	Detect and locate objects with bounding boxes	Detecting cars, people in a video
🧩 **Image Segmentation**	Classify each pixel to a class (fine-grained detection)	Tumor area in medical scan
👁️‍🗨️ **Face Recognition**	Identify or verify person identity from face	Face unlock in smartphones
🔁 **Image Captioning**	Generate text describing an image	"A man riding a bicycle on road"
🔍 **Optical Character Recognition (OCR)**	Convert text in images into editable text	Reading text from scanned documents
🛣️ **Scene Understanding**	Understand relationships among objects	Autonomous vehicle navigation

Popular Deep Learning Models in CV

Model	Use Case	Architecture Type
CNN (Convolutional Neural Network)	Basic image classification	Convolutional layers
ResNet	Deep image classification	Residual connections
YOLO (You Only Look Once)	Real-time object detection	Single-shot detection
Faster R-CNN	High-accuracy object detection	Region proposal + CNN
UNet	Biomedical image segmentation	Encoder-decoder (U-shape)
GAN (Generative Adversarial Network)	Image generation, enhancement	Generator + Discriminator

Popular Libraries/Tools

Library	Language	Use for
OpenCV	Python, C++	Image/video processing, computer vision
TensorFlow/Keras	Python	Deep learning model building
PyTorch	Python	Deep learning, especially CV models
Detectron2	Python	Facebook’s library for object detection
MediaPipe	Python/C++	Real-time face, hand, pose detection

Computer Vision

Make computers understand images and video.

What kind of scene?

Where are the cars?

How far is the building?

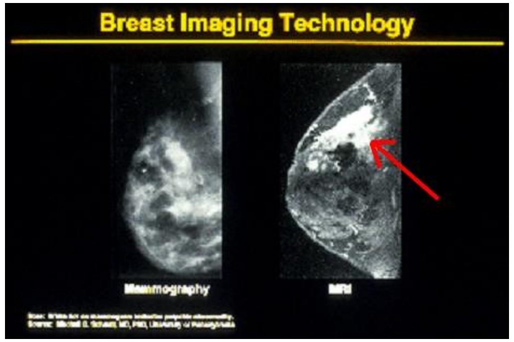
…

Vision is really hard

- Vision is an amazing feat of natural intelligence
  - Visual cortex occupies about 50% of Macaque brain
  - More human brain devoted to vision than anything else

Why computer vision matters

Safety



Health

Security

Comfort

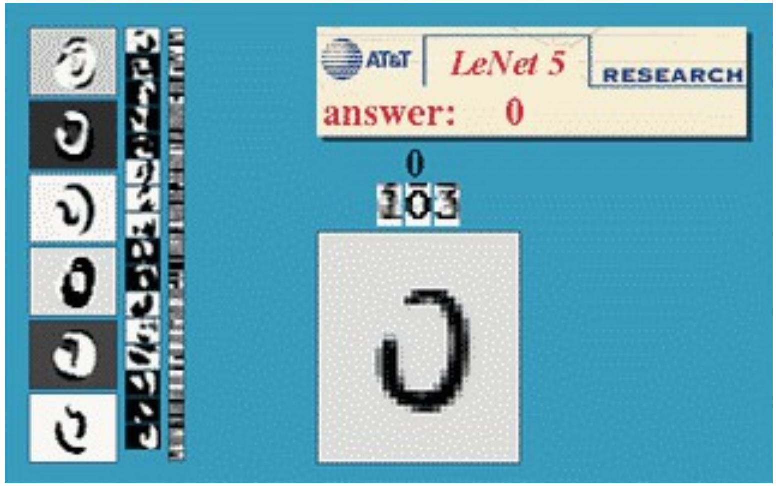
Fun

Access

Optical character recognition (OCR)

Technology to convert scanned docs to text

- If you have a scanner, it probably came with OCR software



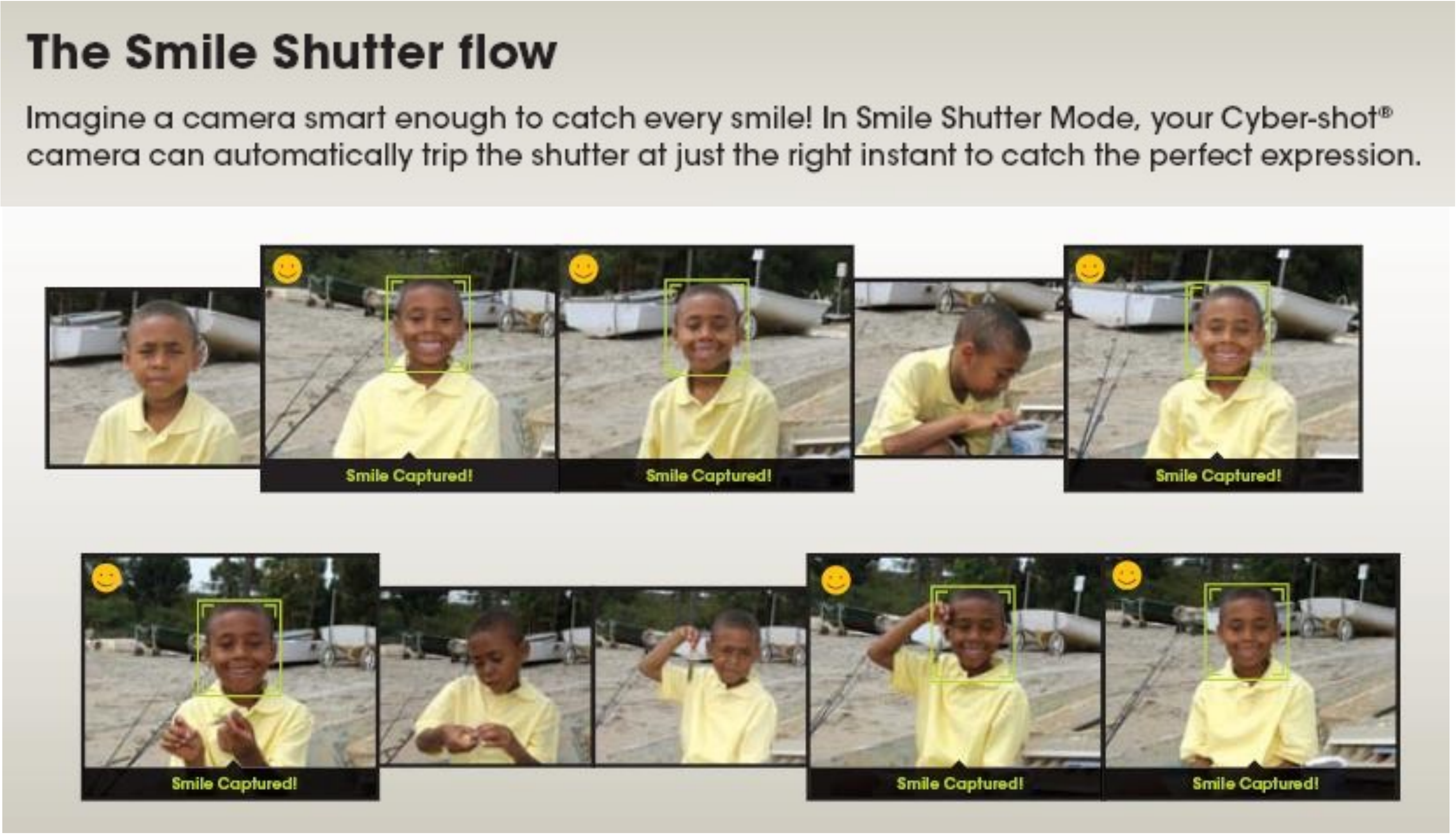
Digit recognition, AT&T labs
http://www.research.att.com/~yann/

License plate readers
http://en.wikipedia.org/wiki/Automatic_number_plate_recognition

Face detection

- Many new digital cameras now detect faces
  - Canon, Sony, Fuji, …

Smile detection



[Sony Cyber-shot® T70 Digital Still Camera]

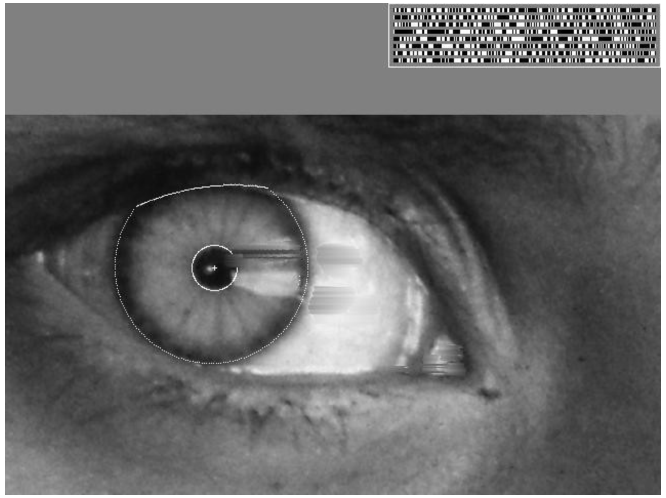
Object recognition (in supermarkets)

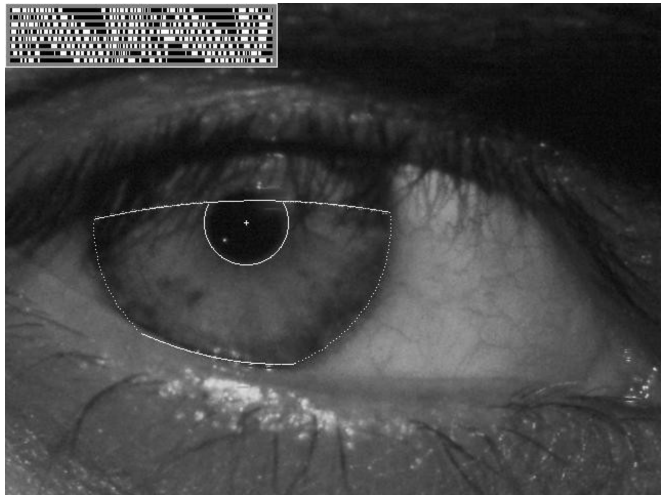
LaneHawk by EvolutionRobotics

"A smart camera is flush-mounted in the checkout lane, continuously watching for items. When an item is detected and recognized, the cashier verifies the quantity of items that were found under the basket, and continues to close the transaction. The item can remain under the basket, and with LaneHawk,you are assured to get paid for it… "

Vision-based biometrics

"*How the Afghan Girl was Identified by Her Iris Patterns*" Read the story
wikipedia





Login without a password…

Fingerprint scanners on
many new laptops,
other devices

Face recognition systems now
beginning to appear more widely
http://www.sensiblevision.com/

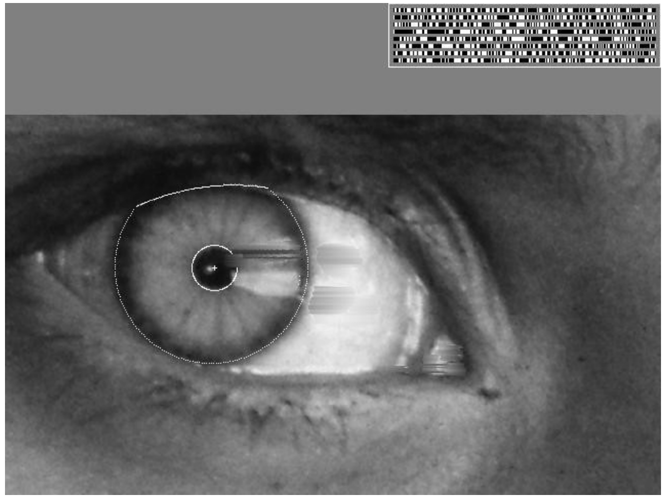
Login without a password…

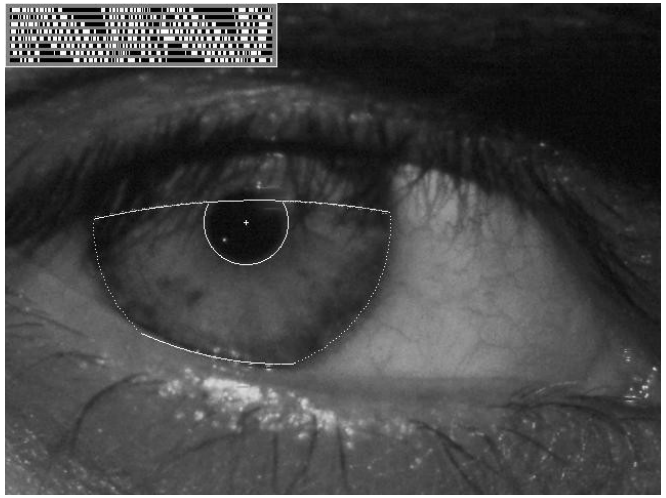
Fingerprint scanners on
many new laptops,
other devices

Face recognition systems now
beginning to appear more widely
http://www.sensiblevision.com/

Vision-base

"*How the Afghan Girl was Identified by Her Iris Patterns*" Read the story
wikipedia





Special effects: shape capture

The Matrix movies, ESC Entertainment, XYZRGB, NRC

Sports

Sportvision first down line
Nice explanation on www.howstuffworks.com

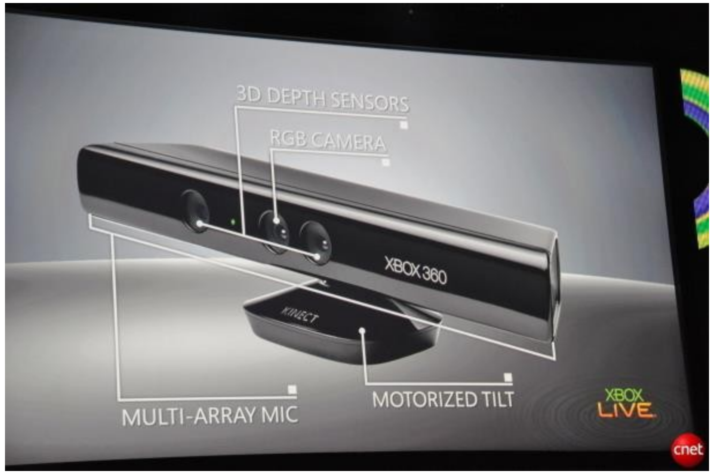
http://www.sportvision.com/video.html

Google cars

http://www.nytimes.com/2010/10/10/science/10google.html?ref=artificialintelligence

Interactive Games: Kinect

- Object Recognition: http://www.youtube.com/watch?feature=iv&v=fQ59dXOo63o
- Mario: http://www.youtube.com/watch?v=8CTJL5lUjHg
- 3D: http://www.youtube.com/watch?v=7QrnwoO1-8A
- Robot: http://www.youtube.com/watch?v=w8BmgtMKFbY



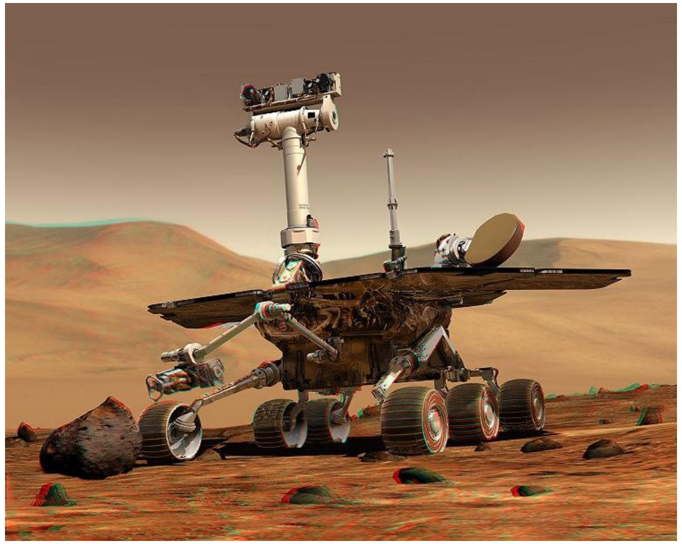
Vision in space

NASA'S Mars Exploration Rover Spirit captured this westward view from atop a low plateau where Spirit spent the closing months of 2007.

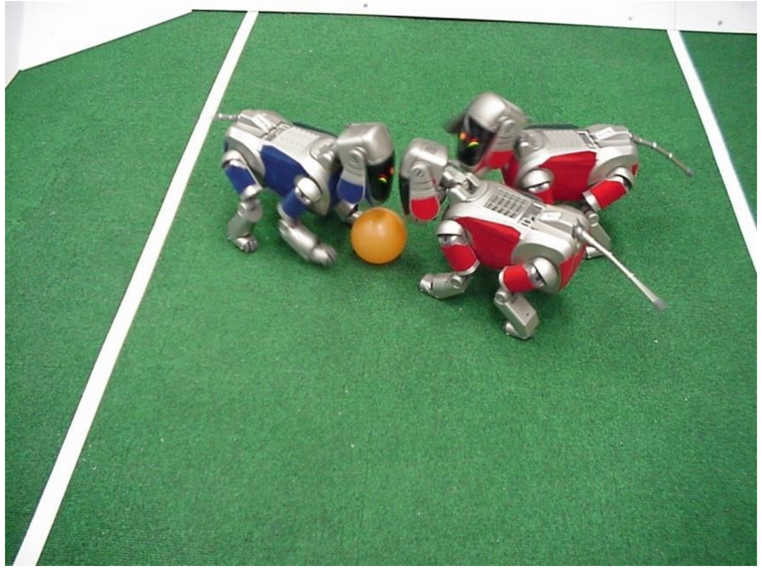
Vision systems (JPL) used for several tasks

- Panorama stitching
- 3D terrain modeling
- Obstacle detection, position tracking
- For more, read "Computer Vision on Mars" by Matthies et al.

Mobile robots



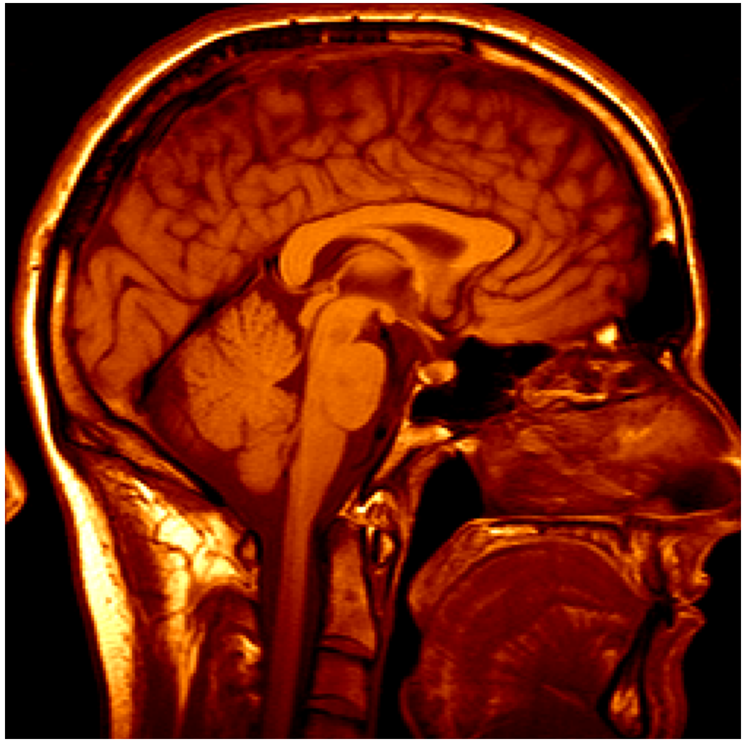
NASA's Mars Spirit Rover http://en.wikipedia.org/wiki/Spirit_rover



http://www.robocup.org/

Saxena et al. 2008
STAIR at Stanford

Medical imaging



3D imaging
MRI, CT

Image guided surgery
Grimson et al., MIT

Computer Vision (Cntd.)

✓ <u>Application:</u>

- Laptop: Biometrics auto-login (face recognition, 3D), OCR
- Smartphones: QR codes, computational photography (Android Lens Blur, iPhone Portrait Mode), panorama construction (Google Photo Spheres), face detection, expression detection (smile), Snapchat filters (face tracking), Google Tango (3D reconstruction), Night Sight (Pixel)
- Web: Image search, Google photos (face recognition, object recognition, scene recognition, geolocalization from vision), Facebook (image captioning), Google maps aerial imaging (image stitching), YouTube (content categorization)
- VR/AR: Outside-in tracking (HTC VIVE), inside out tracking (simultaneous localization and mapping, HoloLens), object occlusion (dense depth estimation)
- Motion: Kinect, full body tracking of skeleton, gesture recognition, virtual try-on
- Medical imaging: CT / MRI reconstruction, assisted diagnosis, automatic pathology, connectomics, endoscopic surgery
- Industry: Vision-based robotics (marker-based), machine-assisted router (jig), automated post, ANPR (number plates), surveillance, drones, shopping
- Transportation: Assisted driving (everything), face tracking/iris dilation for drunkeness, drowsiness, automated distribution (all modes)
- Media: Visual effects for film, TV (reconstruction), virtual sports replay (reconstruction), semantics-based auto edits (reconstruction, recognition)

Understanding and Summarization of **Machine Learning (ML)** and **Deep Learning (DL)** based applications

Machine Learning (ML) Applications

Machine Learning enables systems to learn from data and make predictions or decisions without being explicitly programmed

Common ML Applications

Domain	Application	Description
🎯 **Classification**	Email Spam Detection	Classifies emails as spam or not based on content
💳 **Finance**	Credit Risk Assessment	Predict loan default using past customer data
🏥 **Healthcare**	Disease Prediction	Predict conditions like diabetes, cancer from medical records
🚗 **Automotive**	Driver Behavior Analysis	Classify aggressive vs. safe drivers using sensor data
🛒 **Retail**	Customer Segmentation	Group customers based on behavior for targeted marketing
🌐 **Web**	Recommendation Systems	Suggest movies/products using collaborative filtering
🏢 **HR**	Resume Screening	Filter candidates based on job description and resume keywords

Deep Learning (DL) Applications

Deep Learning is a subset of ML based on **neural networks** with multiple layers. It excels in tasks involving **large, unstructured data** like images, audio, and text.

Common DL Applications

Domain	Application	Description
🖼️ **Computer Vision**	Image Classification (e.g., cats vs. dogs)	Classify images using CNNs
🧠 **Healthcare**	Tumor Detection from MRI/CT	Segment and detect tumors using CNN/UNet
🎙️ **Speech**	Speech-to-Text	Convert voice to text using RNNs/ Transformers
📄 **NLP**	Text Summarization, Chatbots	GPT, BERT models generate and understand human text
🚗 **Autonomous Cars**	Object Detection in Real-time	Detect pedestrians, lanes using YOLO/RCNN
🎞️ **Video Analysis**	Action Recognition in Videos	Classify human actions using 3D CNNs/LSTMs
🎮 **Gaming/Robotics**	Reinforcement Learning for Agents	Train agents to learn from environments (e.g., AlphaGo)

Comparison: ML vs. DL Applications

Criteria	Machine Learning	Deep Learning
Data Requirement	Works with smaller datasets	Needs large amounts of data
Feature Engineering	Manual (you extract features)	Automatic (learns features from raw data)
Speed and Training	Faster to train	Slower and more resource-intensive
Interpretability	Easier to interpret	Often a black box
Common Models	SVM, Decision Tree, Random Forest	CNN, RNN, Transformer, GAN

AI Tools and Frameworks

TensorFlow

TensorFlow is an open-source **machine learning** and **deep learning** framework developed by **Google**.

It's used to build, train, and deploy **neural networks** and **ML models** for tasks like:

- ❖ Image classification
- ❖ Text analysis
- ❖ Speech recognition
- ❖ Time-series forecasting
- ❖ And much more

Key Features of TensorFlow

- Build models using **simple Python code**
- High-level API via **Keras**
- Run on **CPU**, **GPU**, or **TPU**
- Easily deploy on **web**, **mobile**, or **cloud**
- Visualize training with **TensorBoard**

Why we use tensor flow ?

- We use **TensorFlow** because it is a **powerful, flexible, and scalable** open-source framework designed to build and deploy **machine learning (ML)** and **deep learning (DL)** models.

Here’s why TensorFlow is widely used:

1. Ease of Model Building

- Provides **high-level APIs** like **Keras** to easily define, train, and evaluate models.
- Lets you focus on the model, not the math.

2. Deep Learning Capabilities

- Supports neural networks, CNNs, RNNs, transformers, etc.
- Suitable for computer vision, NLP, audio recognition, and more.

3. Scalability

- Can train models on **CPUs, GPUs, TPUs**, and even across **clusters of machines**.
- Easily scalable for large datasets.

4. Production Ready

- TensorFlow Serving, TensorFlow Lite, and TensorFlow.js let you deploy models:
  - On servers (cloud)
  - On mobile devices
  - In browsers

5. Visualization Tools

- TensorBoard helps track metrics like accuracy and loss during training.
- Visualizes computation graphs and profiling.

6. Community and Ecosystem

- Backed by **Google**, with a huge community.
- Tons of pre-trained models, tutorials, and support

Applications:

- Image recognition (e.g., cat vs. dog)
- Text classification (e.g., spam filter)
- Speech recognition
- Time series forecasting
- Recommendation systems

How to Install TensorFlow

- Just open your terminal (Command Prompt or Anaconda Prompt) and run:

```
pip install tensorflow
```

- If you're using a Jupyter Notebook, use:

```
!pip install tensorflow
```

- https://www.tensorflow.org/

Pandas: Python Data Analysis Library

- **Pandas** is a fast, powerful, and flexible **Python library** used for **data manipulation and analysis**. It is especially useful when working with **structured data** (e.g., tables, spreadsheets, CSV files).
- **Pandas** is a **Python library** used for **data analysis**, **data manipulation**, and **data cleaning**. It is built on top of **NumPy** and designed to work with **structured data** (tabular, time-series, matrix data, etc.).
- **Name Origin**: Derived from "**Panel Data**," a term used in econometrics.
- **Main data structures**:
    - Series: 1D labeled array
    - DataFrame: 2D labeled table (rows and columns)

Why Use Pandas?

- Easy handling of **tabular data** (rows & columns)
- Built-in tools for **cleaning**, **transforming**, and **analyzing** data
- Works seamlessly with **NumPy**, **Matplotlib**, **Scikit-learn**, etc.

- How to Install Pandas

```
pip install pandas
```

https://pandas.pydata.org/Pandas_Cheat_Sheet.pdf

NLTK (Natural Language Toolkit)

- **NLTK** stands for **Natural Language Toolkit**. It is a powerful **open-source Python library** designed for working with **human language data (text)**.
- **NLTK is a Python library for Natural Language Processing (NLP)**. It provides easy-to-use interfaces to over 50 corpora and lexical resources, along with text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning.

Key Features of NLTK:

Feature	Description
Tokenization	Breaking text into words or sentences
Stopword Removal	Filtering out common words (like "the", "is")
Stemming & Lemmatization	Reducing words to their root form
POS Tagging	Labeling words with their grammatical part of speech
Named Entity Recognition (NER)	Identifying names of people, places, dates, etc.
Text Classification	Classifying text (e.g., spam detection)
Corpora & Lexical Resources	Access to large text datasets and tools like WordNet
Parsing	Analyzing grammatical structure of sentences

Use Cases of NLTK:

- Sentiment Analysis
- Chatbots
- Machine Translation
- Spam Detection
- Information Retrieval
- Topic Modeling
- Named Entity Recognition