Fundamentals of Artificial Neural Networks and Convolutional Neural Networks

B.S. Kushvah

Department of Mathematics and Computing
Indian Institute of Technology (ISM), Dhanbad-826004

bskush@iitism.ac.in

MOTIVATIONS

In recent years, machine learning(ML) is extensively used in the field of science and engineering for supervised and unsupervised classification.

- The Artificial Neural Network (ANN) is an area of machine learning with the many hidden layers.
- Machine learning involves a cost function for the quantification of the expected outcomes.

MOTIVATIONS

- Deeplearning procedure based on a particular type of feedforward network is called the Convolutional Neural Networks (CNN) which was invented by LeCun (1998).
- Convolution, which is a popular array operation that is used in various forms in signal processing, digital recording, image processing, video processing, and computer vision.
- CNN automatically builds hierarchical features extractor (filter) that have enough layers to perform better than defined application specific feature extractors.

Artificial Neural Networks

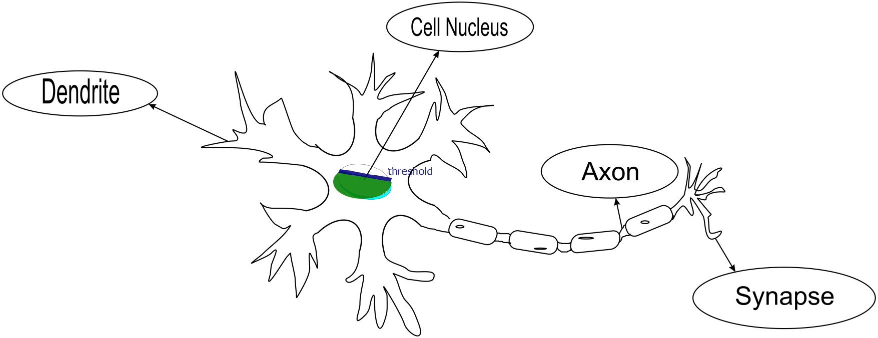


Figure : Biological Neuron

.

Artificial Neural Networks

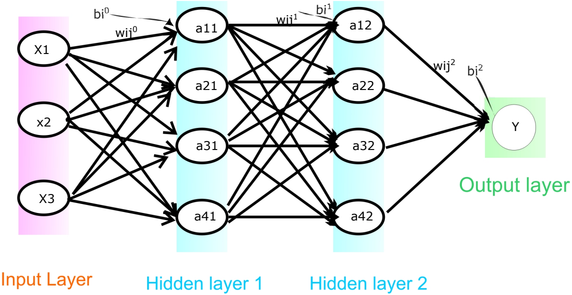


Figure : Dense Network

.

Artificial Neural Networks

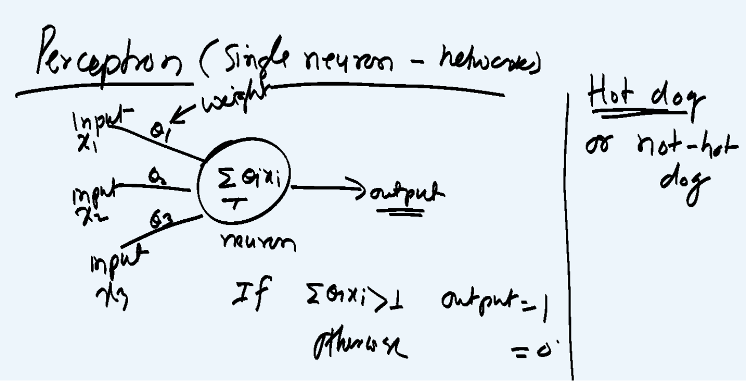
- Hebbs Rule
- McCulloch-Pitts
- Perceptrons
- Dense Neural Network
- Activation Functions
- Cost functions
- Gradient Descent/ Forward Propagation/Backward Propagation

Perceptrons

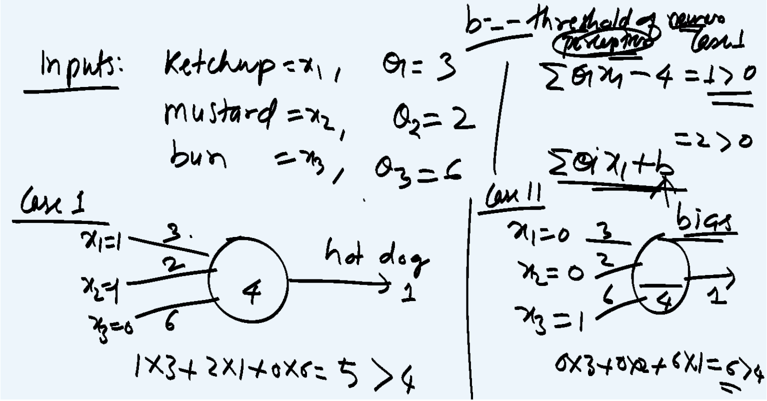
In case of the perceptron learning rule, the learning signal is the difference between the calculated output and actual (target) output of the neuron. The output Y is obtained on the basis of the net input calculated and activation function being applied over the net input $y_{in} = b + \sum_{i=1}^{n} x_i w_i$

$$y = f(y_{in}) = \begin{cases} 1 & \text{if} \quad y_{in} > \theta \\ 0 & \text{if} \quad -\theta \leq y_{in} \leq \theta \\ -1 & \text{if} \quad y_{in} < -\theta \end{cases}$$

Identification of Hot Dog



Identification of Hot Dog



Logic AND function implementation

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

Logic AND function

Let us consider $\theta = 0, \alpha = 1,\ y = b + w_1 x_1 + w_2 x_2$

x_1	x_2	t	Nety_{in}	y	$\triangle w_1$	$\triangle w_2$	$\triangle b$	w_1	w_2	b
								0	0	0
1	1	1	0	0	1	1	1	1	1	1
1	-1	-1	1	1	-1	1	-1	0	2	0
-1	1	-1	2	1	1	-1	-1	1	1	-1
-1	-1	-1	-3	-1	0	0	0	1	1	-1

Table : EPOCH-1

Logic AND function

x_1	x_2	t	Nety_{in}	y	$\triangle w_1$	$\triangle w_2$	$\triangle b$	w_1	w_2	b
								1	1	-1
1	1	1	1	1	0	0	0	1	1	-1
1	-1	-1	-1	-1	0	0	0	1	1	-1
-1	1	-1	-1	-1	0	0	0	1	1	-1
-1	-1	-1	-1	-1	0	0	0	1	1	-1

Table : EPOCH-2

Final weights are: $w_1 = 1, w_2 = 1$ and bias $b = -1$ then we have $y = -1 + x_1 + x_2$

Multi-layer Perceptrons:Logic XOR function

The XOR function is given by $y = x_1\bar{x_2} + \bar{x_1}x_2$ i.e. $y = z_1 + z_2$ where $z_1 = x_1\bar{x_2}$ and $z_2 = \bar{x_1}x_2$. Let $w_{11} = w_{21} = 1$, threshold=1 and learning rage $\alpha = 1.5$. We update weights as

$$w_{ij}(new) = w_{ij}(old) + \alpha(t - output)x_i.$$

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

Tables for z_1 and z_2 are given as:

x_1	x_2	z_1
0	0	0
0	1	0
1	0	1
1	1	0

,

x_1	x_2	z_2
0	0	0
0	1	1
1	0	0
1	1	0

Logic XOR function

$$z_1 = w_{11}x_1 + w_{21}x_2$$

x_1	x_2	t	Net z_{1in}	z_1	w_{11}	w_{21}
					1	1
0	0	0	0	1	1	1
0	1	0	1	1	1	-0.5
1	0	1	1	1	1	-0.5
1	1	0	0.5	0	1	-0.5

Table : EPOCH-1

Logic XOR function

$z_1 = w_{11}x_1 + w_{21}x_2$

x_1	x_2	t	Net z_{1in}	z_1	w_{11}	w_{21}
					1	-0.5
0	0	0	0	0	1	-0.5
0	1	1	0	0	1	-0.5
1	0	0	1	1	1	-0.5
1	1	0	0	0	1	-0.5

Table : EPOCH-2

Weights are: $w_{11} = 1, w_{21} = -0.5$ then we have $z_1 = x_1 - 0.5x_2$

Logic XOR function

$$z_2 = w_{12}x_1 + w_{22}x_2$$

x_1	x_2	t	Net z_{1in}	z_1	w_{12}	w_{22}
					1	1
0	0	0	0	0	1	1
0	1	1	1	1	1	1
1	0	0	1	1	-0.5	1
1	1	0	0.5	0	-0.5	1

Table : EPOCH-1

Logic XOR function

$z_2 = w_{12}x_1 + w_{22}x_2$

x_1	x_2	t	Net z_{1in}	z_2	w_{12}	w_{22}
					-0.5	1
0	0	0	0	0	-0.5	1
0	1	1	1	1	-0.5	1
1	0	0	0.5	0	-0.5	1
1	1	0	0.5	0	-0.5	1

Table : EPOCH-2

Weights are: $w_{12} = -0.5, w_{21} = 1$ then we have $z_2 = -0.5x_1 + x_2$

Logic XOR function

$$y = v_1 z_1 + v_2 z_2,$$

x_1	x_2	z_1	z_2	y
0	0	0	0	0
0	1	0	1	1
1	0	1	0	1
1	1	0	0	0

Logic XOR function

$z_2 = w_{12}x_1 + w_{22}x_2$

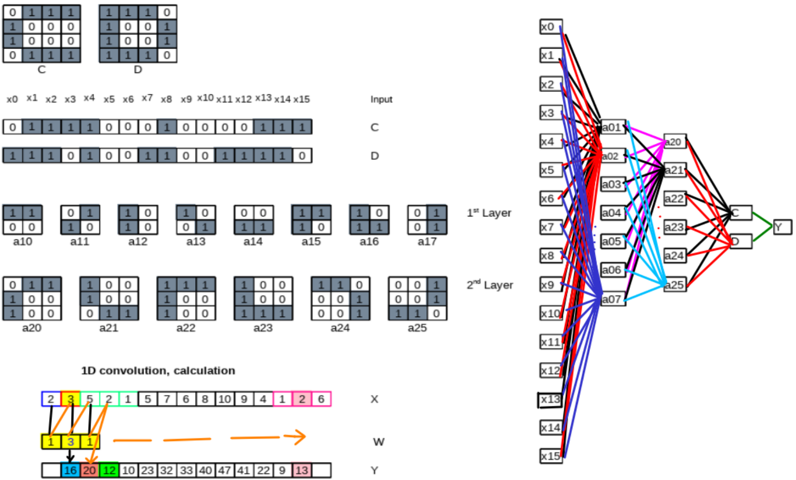
z_1	z_2	t	Net y_{1in}	y	v_1	v_2
					1	1
0	0	0	0	0	1	1
0	1	1	1	1	1	1
1	0	1	1	1	1	1
0	0	0	0	0	1	1

Table : EPOCH-1

Final weights are: $v_1 = 1, v_2 = 1$ then we have $y = z_1 + z_2$ Hence, we have $w_11 = 1, w_{21} = -0.5, w_{12} = -0.5, w_{22} = 1$ and $v_1 = 1, v_2 = 1$.

Example of Dense ANN:

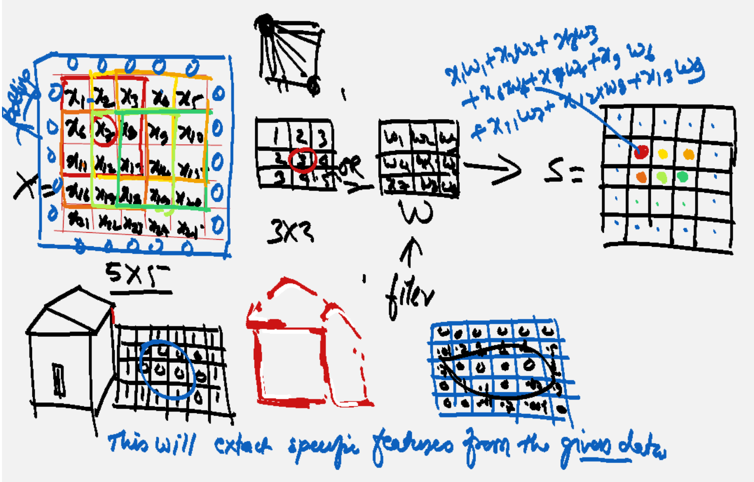
Classification of C or D using Dense ANN



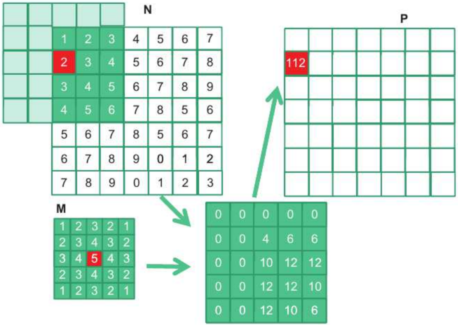
Convolutional Neural Networks

- Convolution is an array operation where each output data element is a weighted sum of a collection of neighboring input elements.
- The weights used in the weighted sum calculation are defined by an input mask array, commonly referred to as the convolution kernel.

2D Convolution

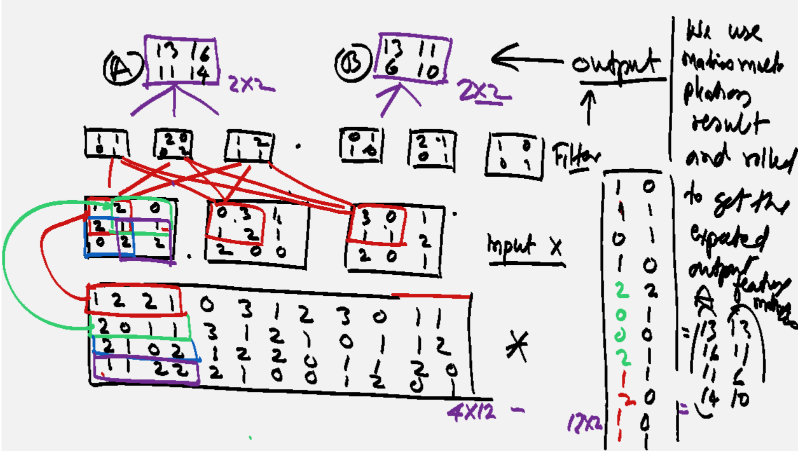


2D Convolution



Source: Kirk,D.B.,Hwu,W.W.(2012).Programming Massively Parallel Processors: A Hands-on Approach.Netherlands:Elsevier Science.

2D Convolution filter banks



Thank You!