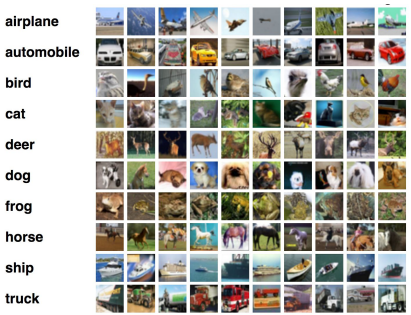
EE655: Computer Vision & Deep Learning

Lecture 07

Koteswar Rao Jerripothula, PhD
Department of Electrical Engineering
IIT Kanpur

Definition

- In image classification, we analyze different image properties to organize the image data into pre-defined categories.



- Motivation: Image Retrieval, Image Recognition, ...

Challenges

Features can be many. Which features to use for given classification problem

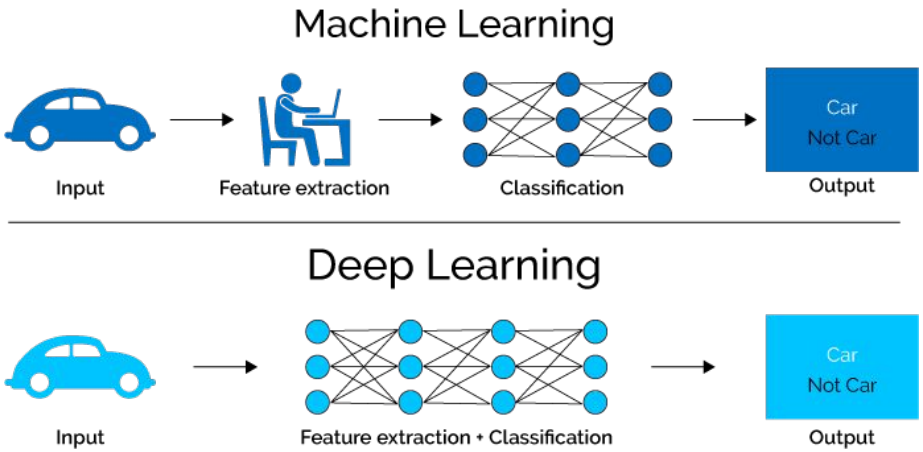
Similarly, categories can be many too for a single image

Multi-class Classification

Multi-label Classification

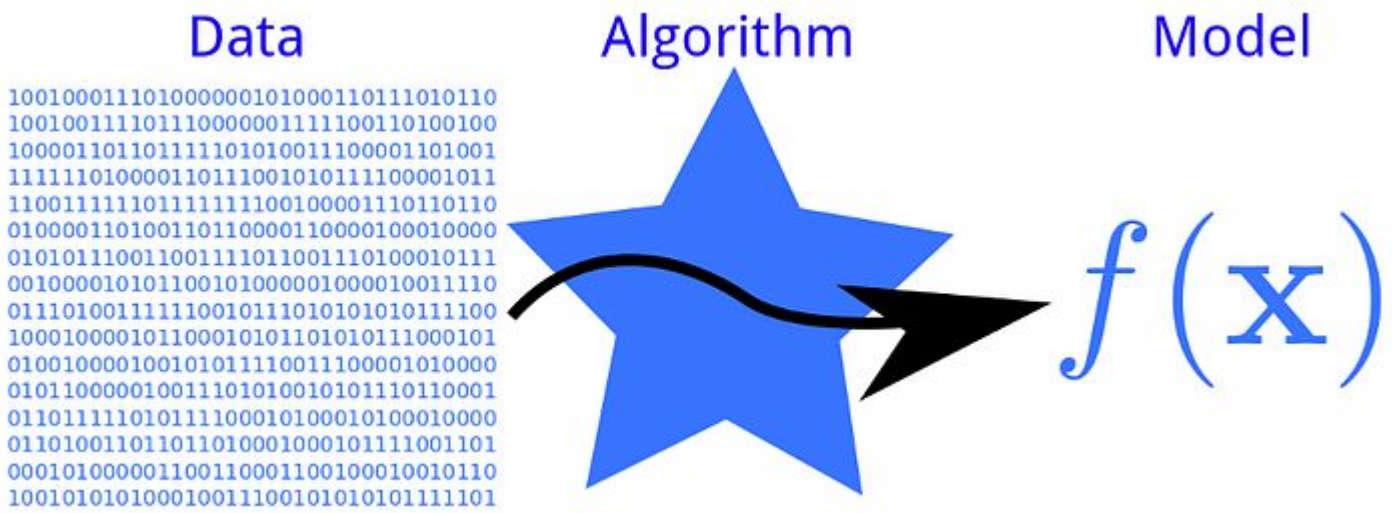
What do we need?

- A Feature Extractor: A set of algorithms that can provide numeric representations of the image data.
- A Classifier: An algorithm that can analyze the features of an image and assign the pre-defined label/s



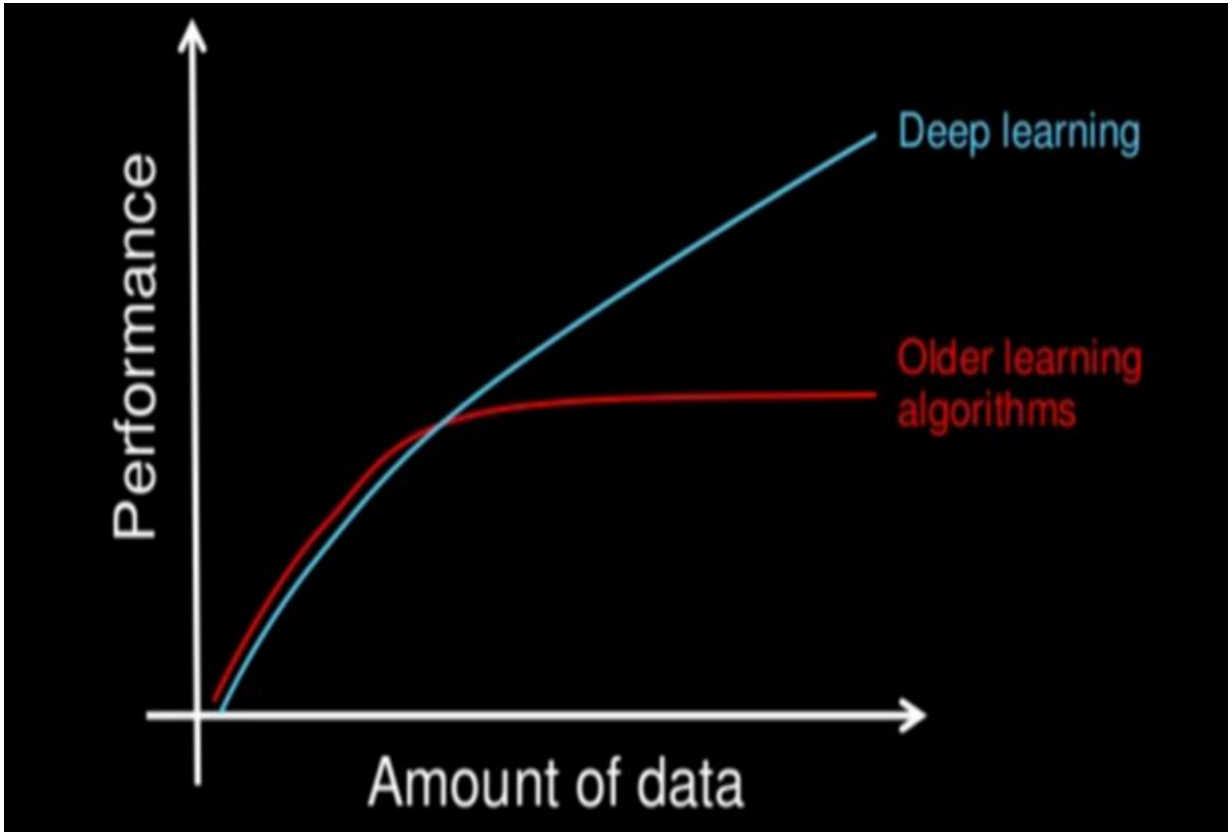
ML Algo. vs ML Model

An “*algorithm*” in machine learning is a procedure that is run on data to create a machine learning “*model*.”



MODEL PARAMETERS	HYPERPARAMETER
They are required for making predictions	They are required for estimating the model parameters
They are estimated by optimization algorithms(Gradient Descent, Adam, Adagrad)	They are estimated by hyperparameter tuning
They are not set manually	They are set manually
The final parameters found after training will decide how the model will perform on unseen data	The choice of hyperparameters decide how efficient the training is. In gradient descent the learning rate decide how efficient and accurate the optimization process is in estimating the parameters

Performance vs Labelled Data required



It comes along with a cost

- Huge data-labelling
- Huge computational requirements
- Difficulty in explaining the decisions made

What's the silver lining: Excellent Accuracy !!!

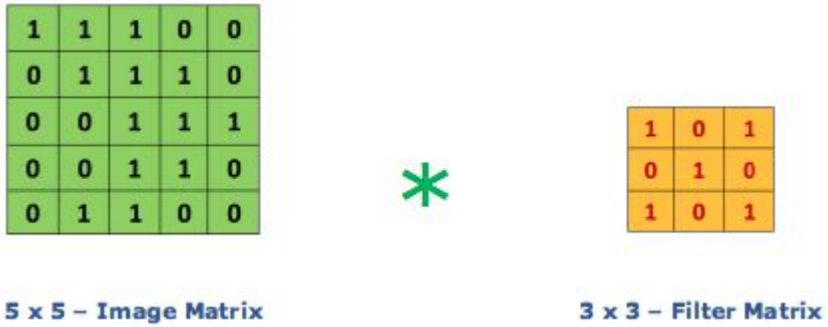
Transfer Learning!!!

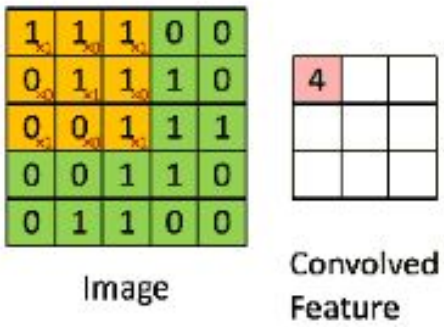
Two basic concepts we need to dive into deep learning

Convolution (for feature extraction)

Neural Networks (for classification)

Convolution

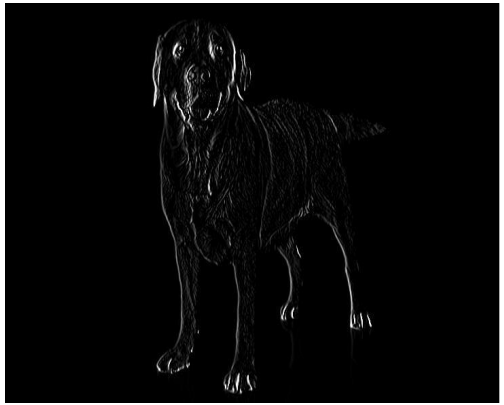




An example

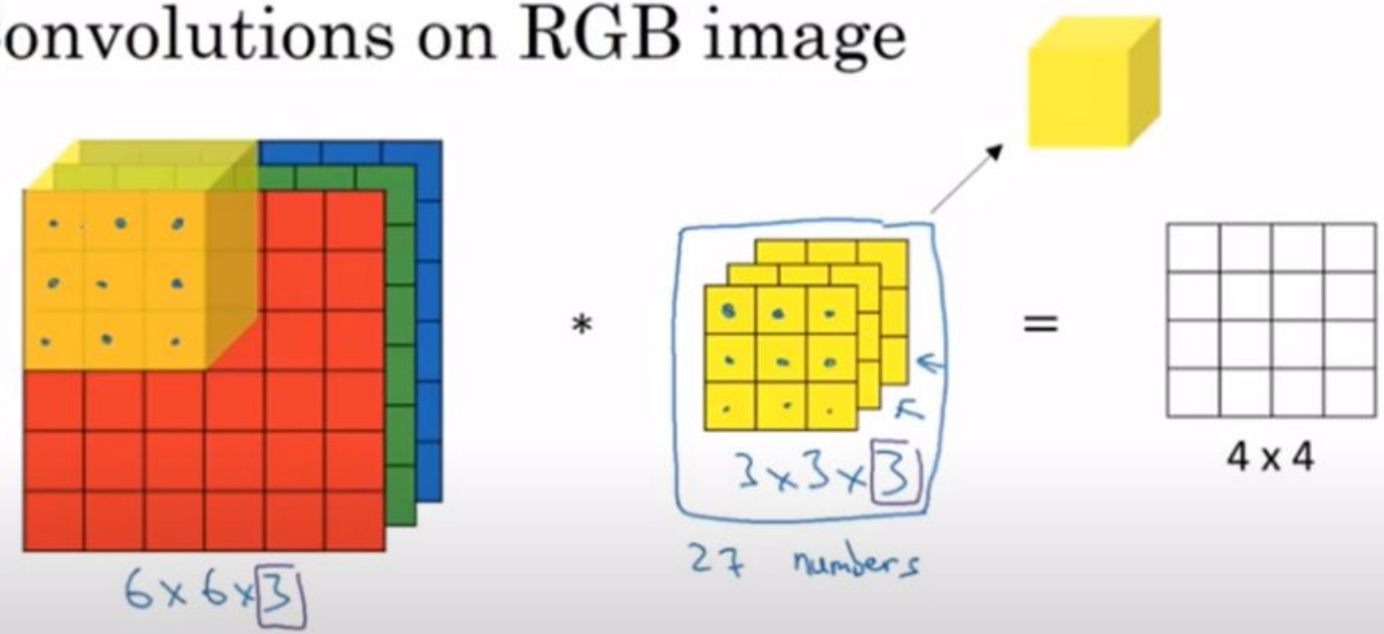
Using Prewitt Filter

$$\begin{matrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{matrix}$$



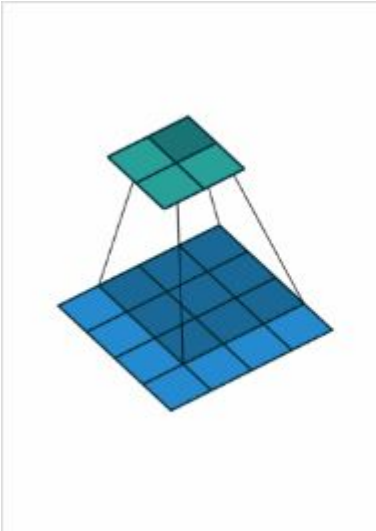
$$\begin{matrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{matrix}$$

Convolutions on RGB image



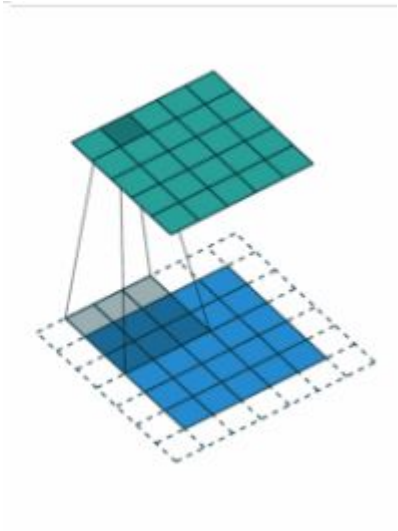
Valid (No) Padding

- Valid padding is a technique used in convolutional neural networks (CNNs) to process the input data without adding any additional rows or columns of pixels around the edges of the data.
- This means that the size of the output feature map is smaller than the size of the input data.
- Valid padding is used when it is desired to reduce the size of the output feature map in order to reduce the number of parameters in the model and improve its computational efficiency.



Same Padding

- In same padding, we add additional rows and columns of pixels around the edges of the input data so that the size of the output feature map is the same as the size of the input data.



Compute the output of following image when convolved with the following filter. Assume valid padding.

23	145	1
34	132	10
76	145	32

Input's Channel-1

11	221	12
13	190	65
45	196	56

Input's Channel-2

34	154	75
85	190	89
56	178	90

Input's Channel-3

-1	1

Filter's Channel-1

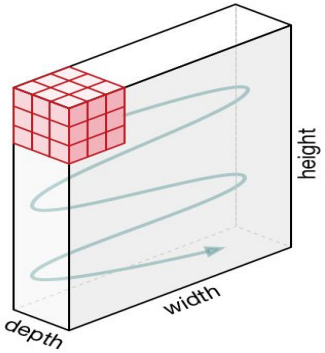
-1	0

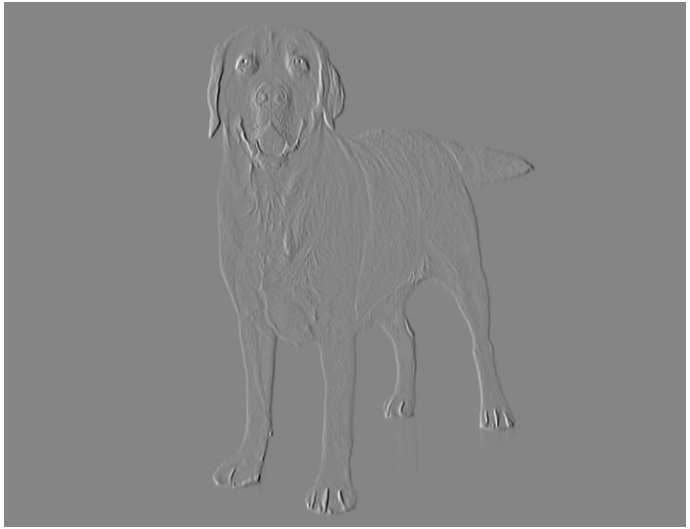
Filter's Channel-2

0	1

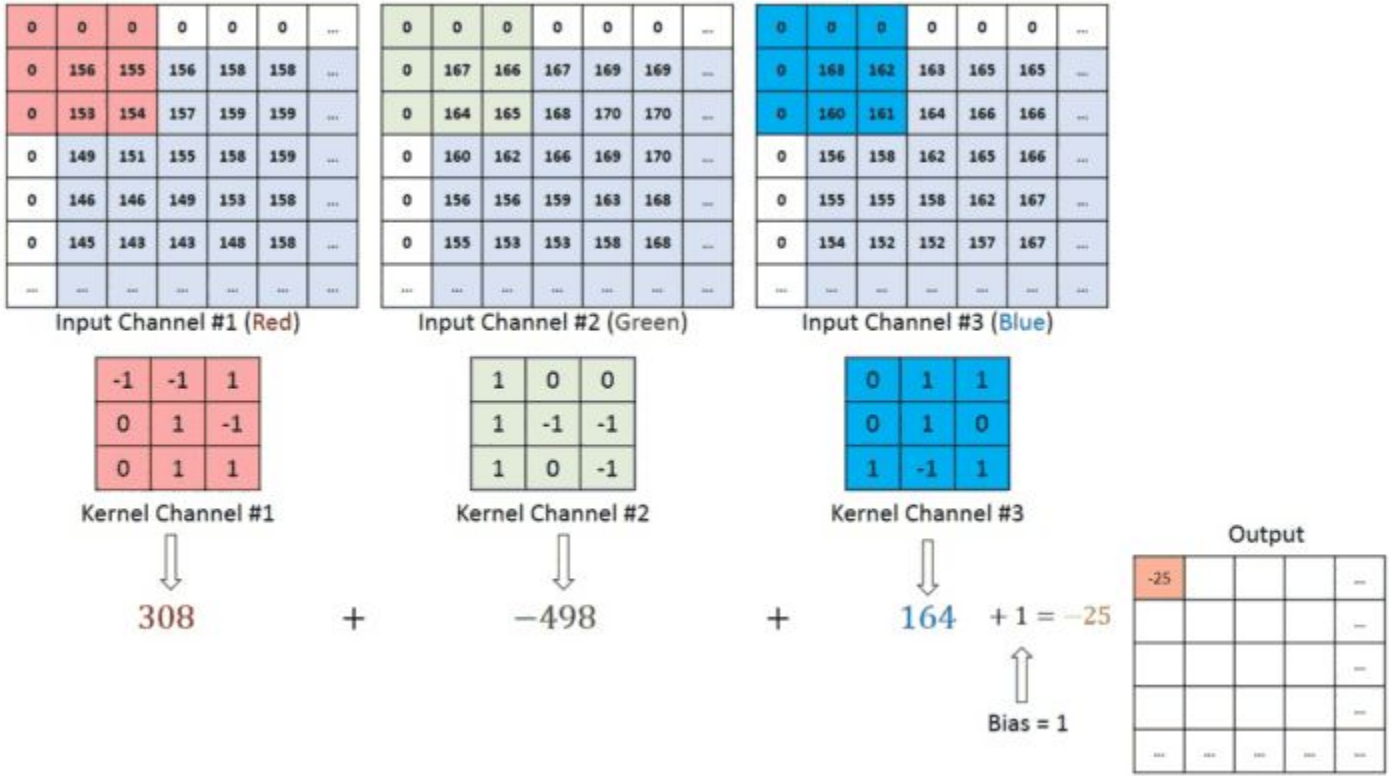
Filter's Channel-3

Using 3D Prewitt Filter





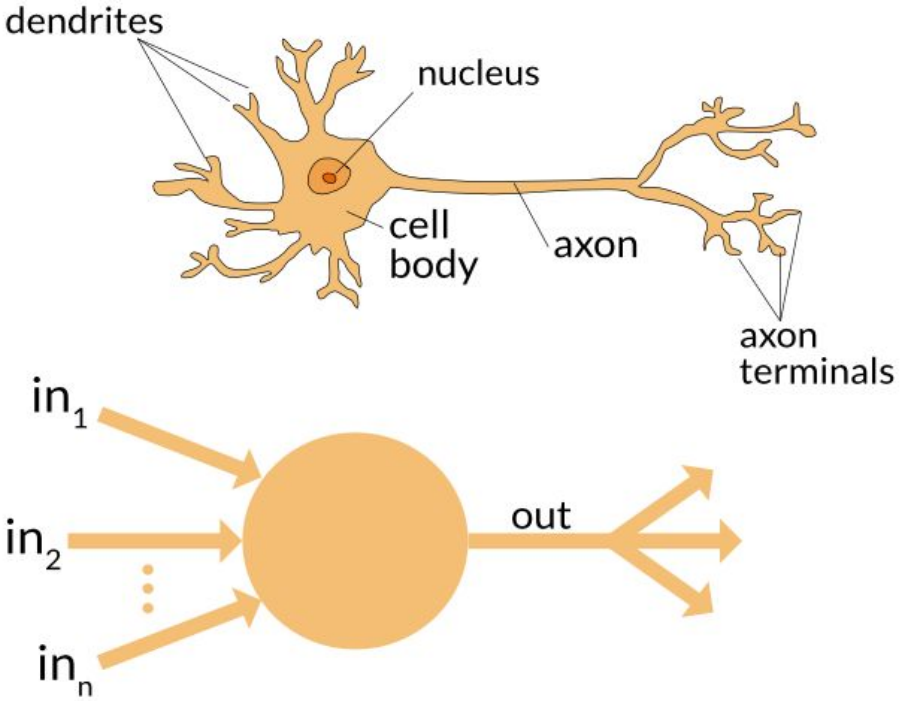
Convolution on RGB Volume



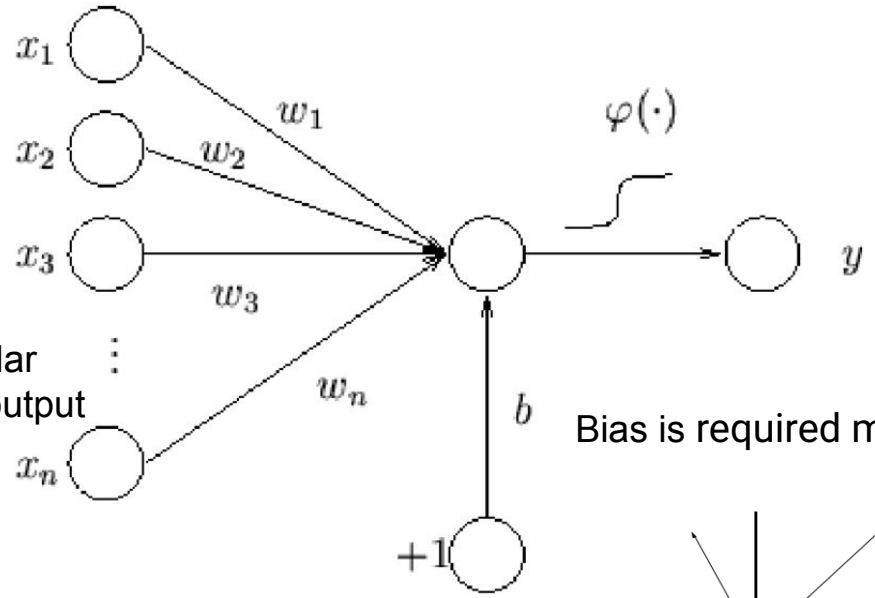
Neural Networks

Mimicking a neuron

dendrites receive neuron signals, and **axons** transmit them

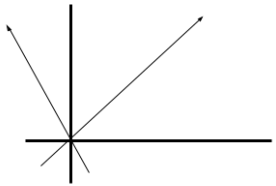


Perceptron model receives input(s), processes it, and then produces an output.

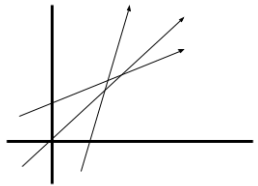


Weights show the effectiveness of a particular input in determining the output

Bias is required make the model flexible.

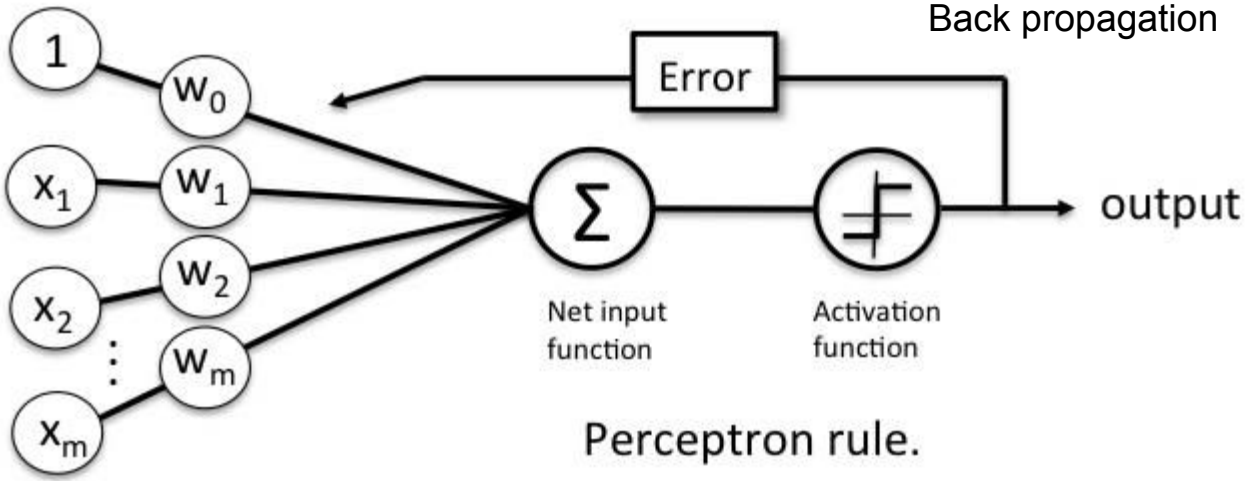


Y = mx

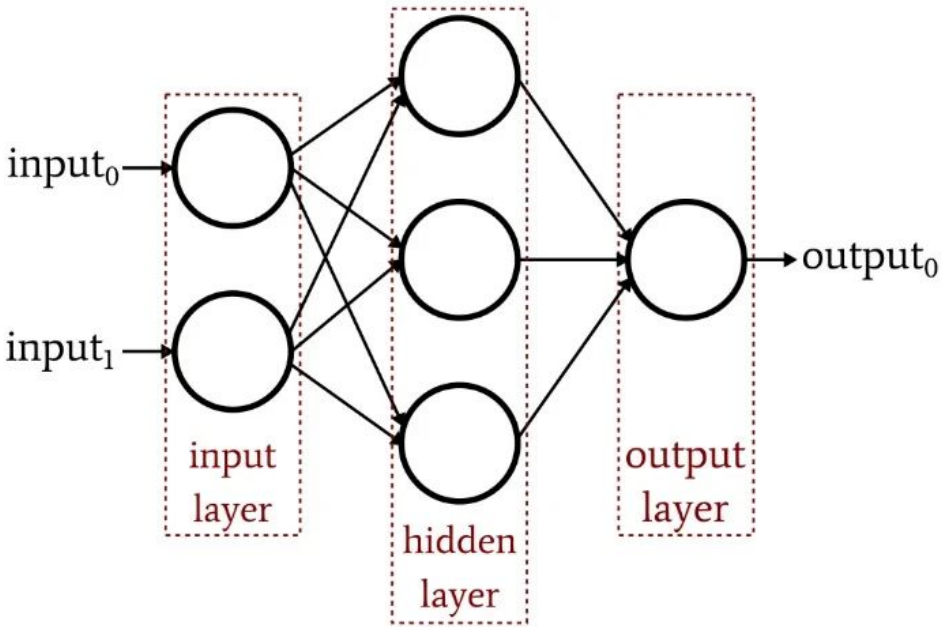


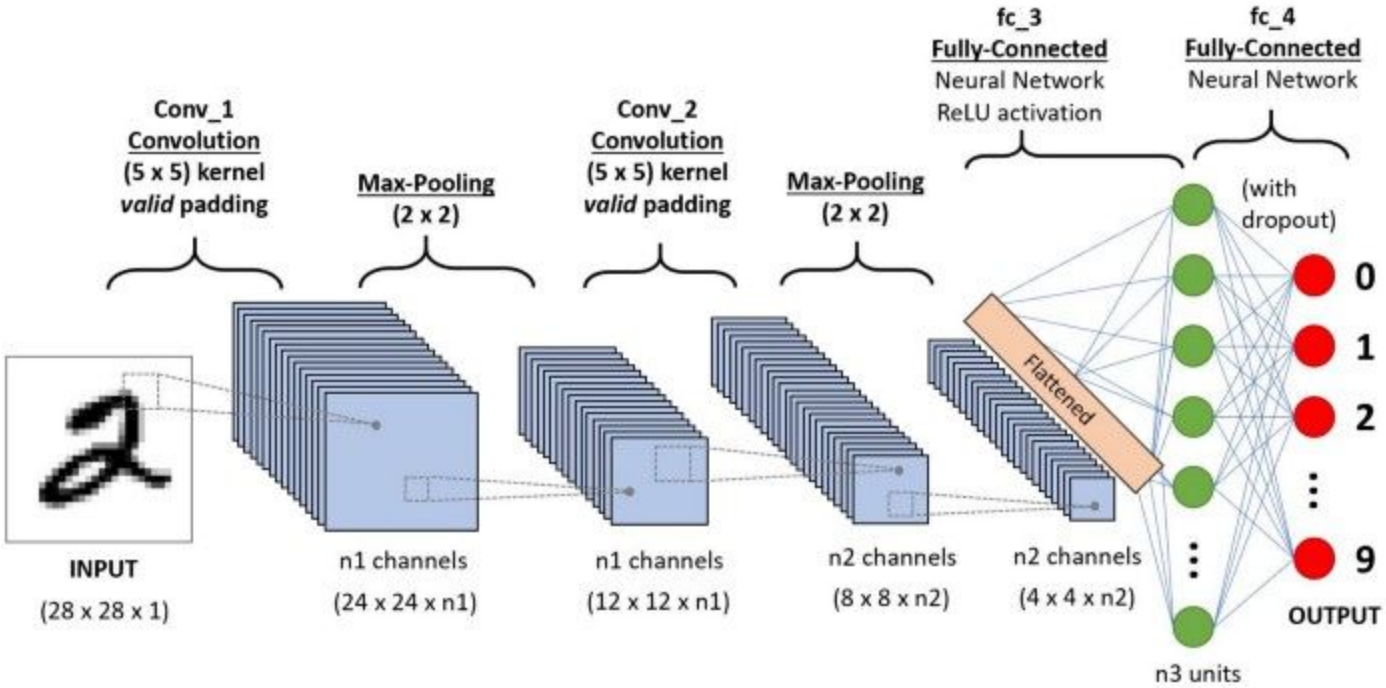
Y = mx + c

Perceptron has two functions

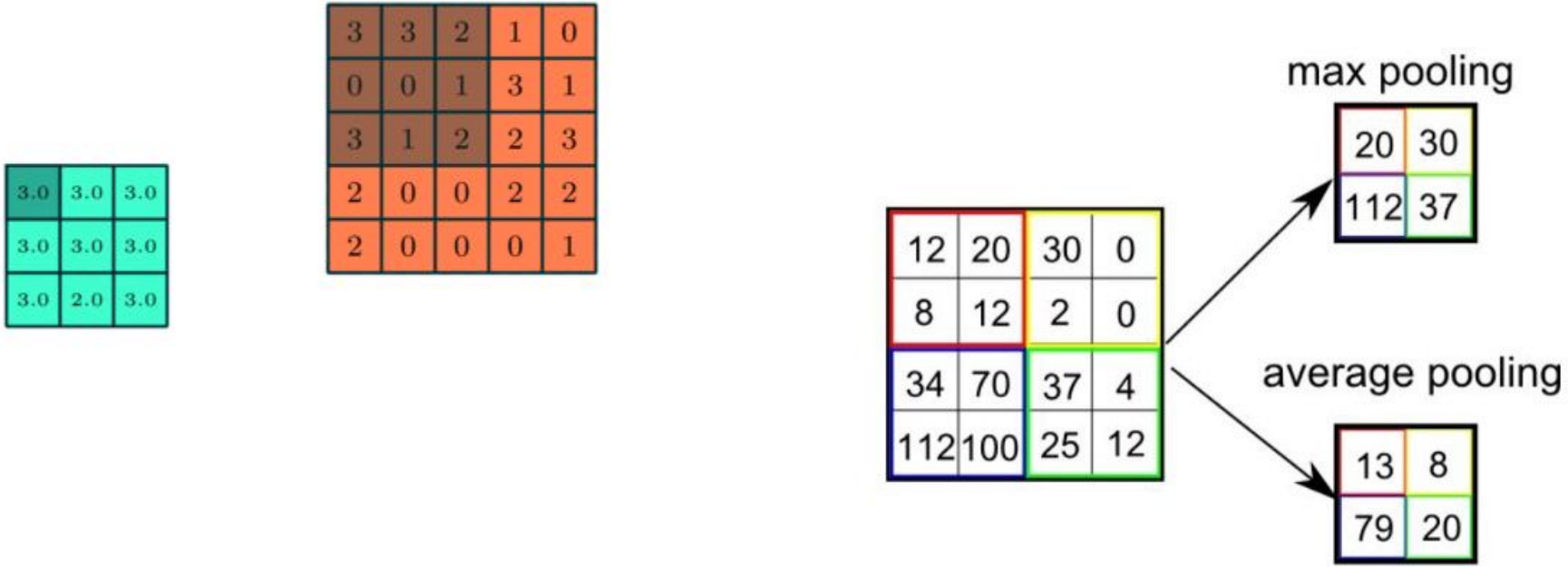


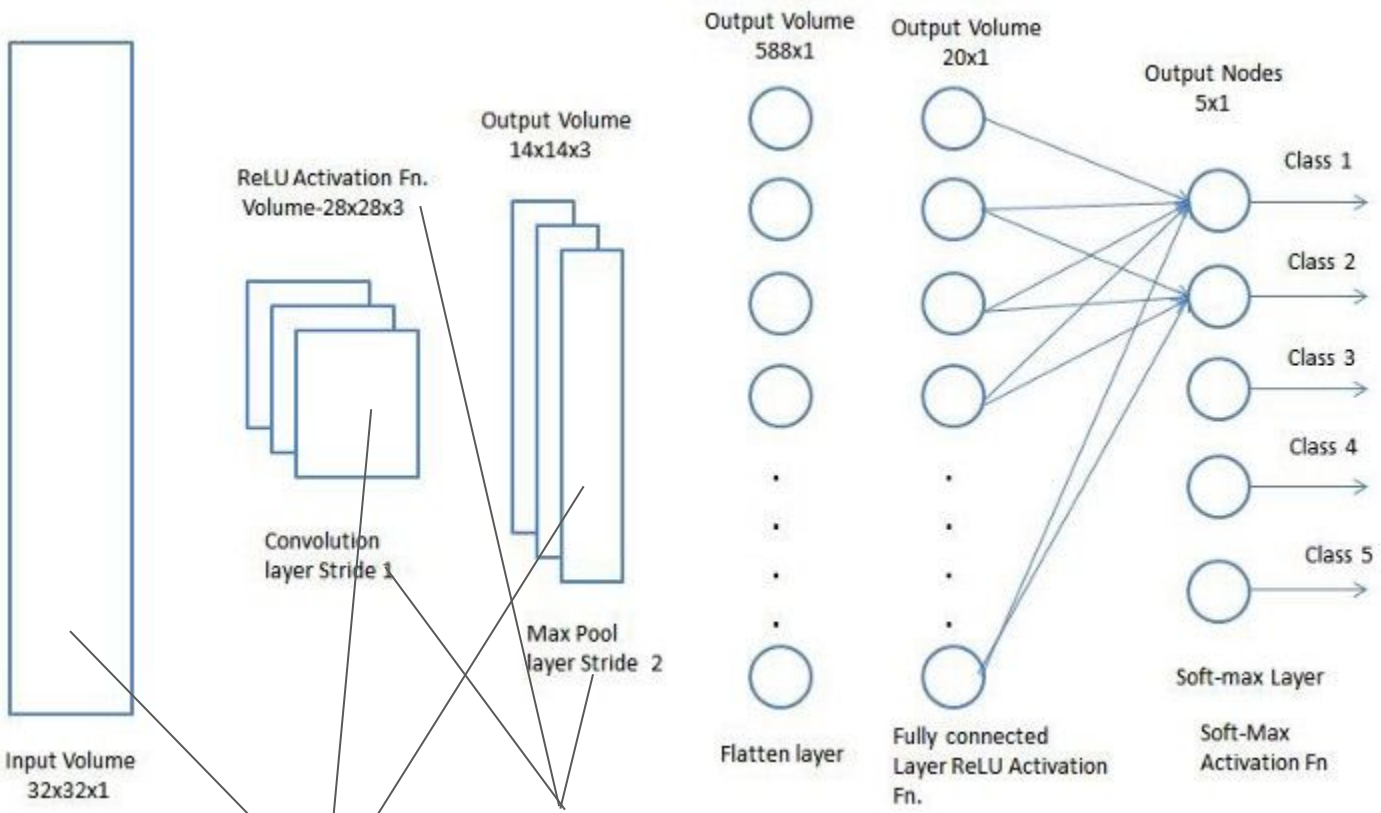
Multi-layer Perceptron (Neural Network)




Conv_1
Convolution
(5 x 5) kernel
valid padding
Max-Pooling
(2 x 2)
Conv_2
Convolution
(5 x 5) kernel
valid padding
Max-Pooling
(2 x 2)
fc_3
Fully-Connected
Neural Network
ReLU activation
fc_4
Fully-Connected
Neural Network
(with
dropout)
Flattened
INPUT
(28 x 28 x 1)
n1 channels
(24 x 24 x n1)
n1 channels
(12 x 12 x n1)
n2 channels
(8 x 8 x n2)
n2 channels
(4 x 4 x n2)
n3 units
0
1
2
9
OUTPUT

Pooling Layer (To reduce the spatial size)





Layers (carrying out operations

Feature Volumes

Target Vector

Multi-class Classification

One-hot Encoded Target Vector with a **single 1**

[0 1 0 0]

Multi-label Classification

One-hot Encoded Target Vector with multiple 1s

[0 1 1 0]