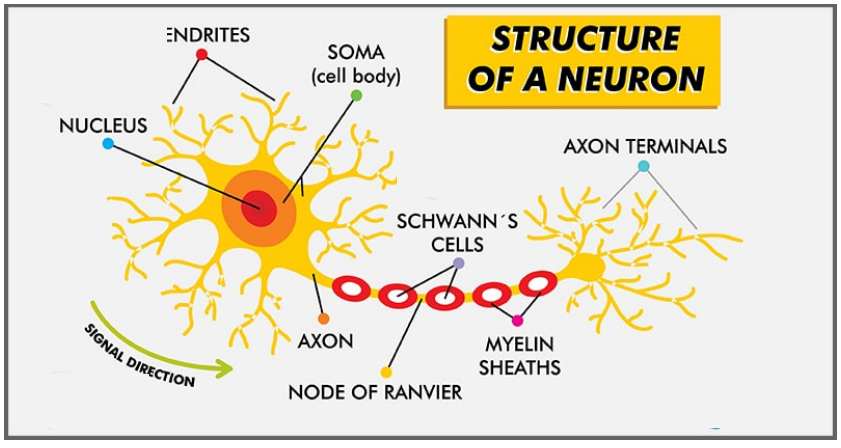
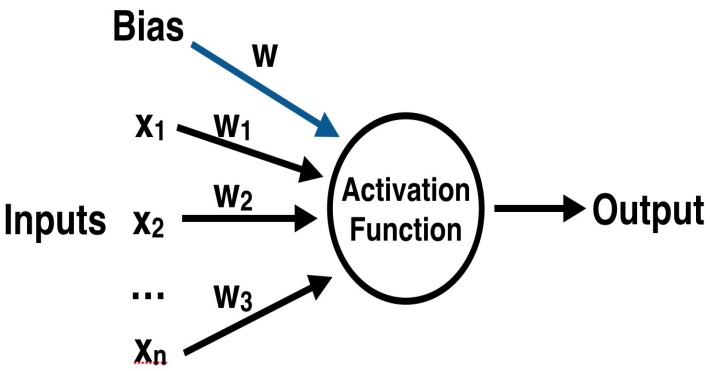
Neural Networks

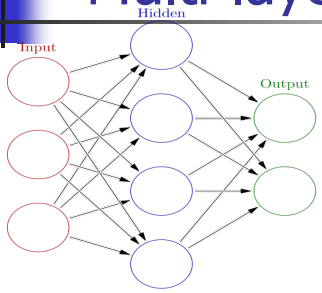
- Neural Networks are a machine learning framework that attempts to mimic the learning pattern of natural biological neural networks:

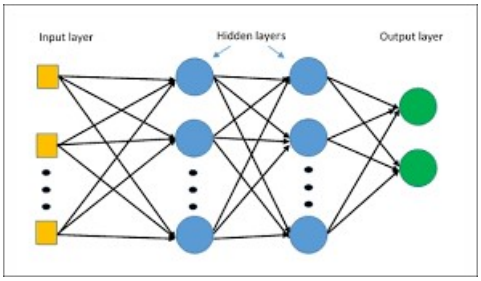


The Perceptron



Multi-layer perceptron



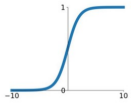


Structure of a Neural Network

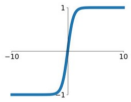
How many hidden layers should be there in Neural Network?

Activation Functions

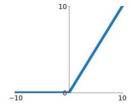
Sigmoid
$\sigma(x) = \frac{1}{1+e^{-x}}$



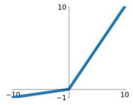
tanh
$\tanh(x)$



ReLU
$\max(0, x)$

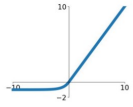


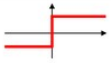
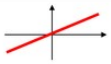
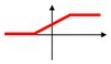
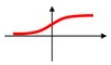
Leaky ReLU
$\max(0.1x, x)$



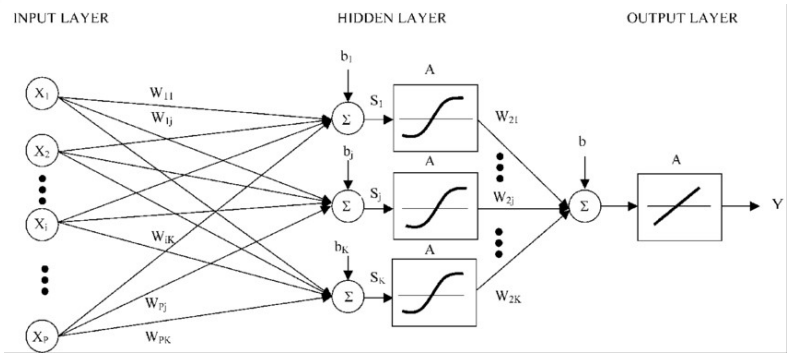
Maxout
$\max(w_1^T x + b_1, w_2^T x + b_2)$

ELU
$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

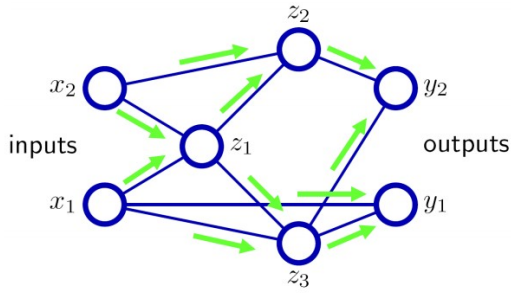


Activation function	Equation	Example	1D Graph
Unit step (Heaviside)	$\phi(z) = \begin{cases} 0, & z < 0, \\ 0.5, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Sign (Signum)	$\phi(z) = \begin{cases} -1, & z < 0, \\ 0, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Linear	$\phi(z) = z$	Adaline, linear regression	
Piece-wise linear	$\phi(z) = \begin{cases} 1, & z \geq \frac{1}{2}, \\ z + \frac{1}{2}, & -\frac{1}{2} < z < \frac{1}{2}, \\ 0, & z \leq -\frac{1}{2}, \end{cases}$	Support vector machine	
Logistic (sigmoid)	$\phi(z) = \frac{1}{1 + e^{-z}}$	Logistic regression, Multi-layer NN	
Hyperbolic tangent	$\phi(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$	Multi-layer NN	

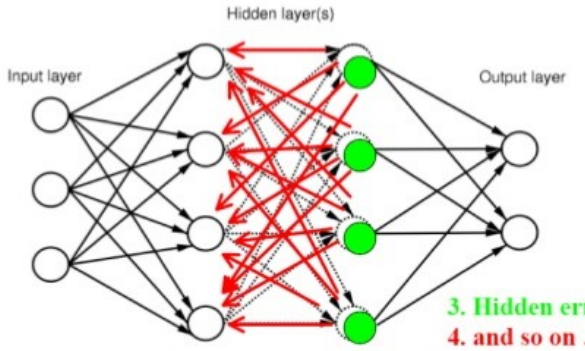
	Propagation	Back-propagation
Sigmoid	$y_s = \frac{1}{1+e^{-x_s}}$	$\left[\frac{\partial E}{\partial x}\right]_s = \left[\frac{\partial E}{\partial y}\right]_s \frac{1}{(1+e^{x_s})(1+e^{-x_s})}$
Tanh	$y_s = \tanh(x_s)$	$\left[\frac{\partial E}{\partial x}\right]_s = \left[\frac{\partial E}{\partial y}\right]_s \frac{1}{\cosh^2 x_s}$
ReLu	$y_s = \max(0, x_s)$	$\left[\frac{\partial E}{\partial x}\right]_s = \left[\frac{\partial E}{\partial y}\right]_s \mathbb{I}\{x_s > 0\}$
Ramp	$y_s = \min(-1, \max(1, x_s))$	$\left[\frac{\partial E}{\partial x}\right]_s = \left[\frac{\partial E}{\partial y}\right]_s \mathbb{I}\{-1 < x_s < 1\}$



- Forward Propagation :
  - Sum inputs, produce activation, feed-forward

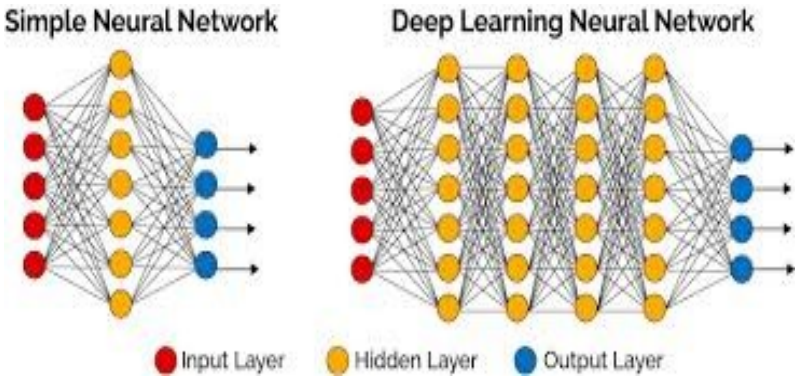


Backpropagation Learning



3. Hidden error values
4. and so on …

 89



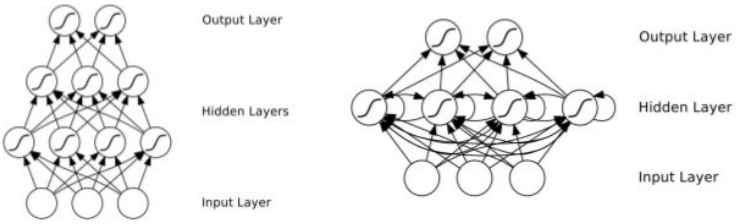
 90

Disadvantages

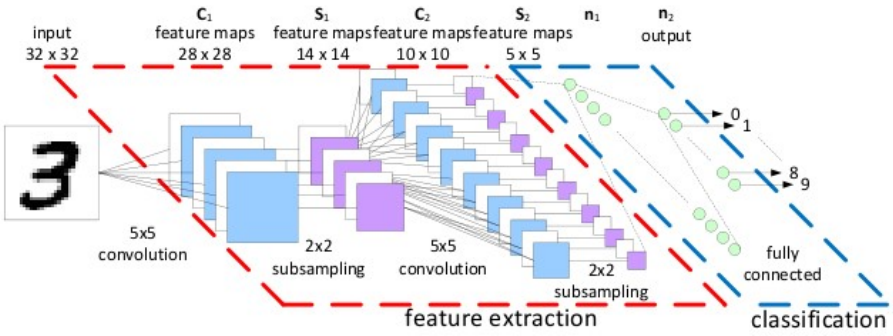
- Need large processing and storage resources
- Require a large diversity of training for real-world operation
- Black Box "Difficult to understand"

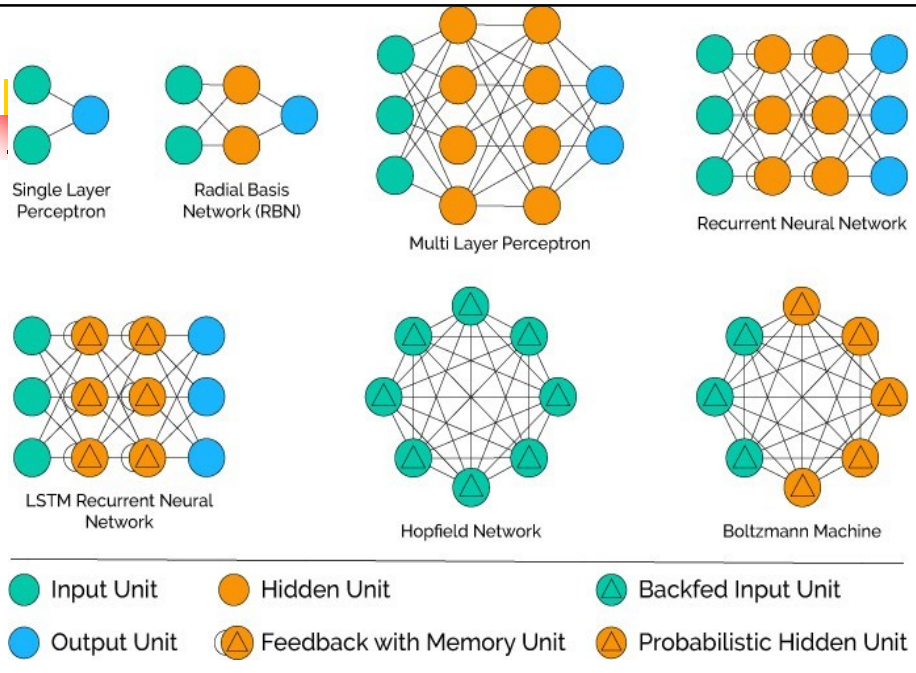
CNN vs RNN

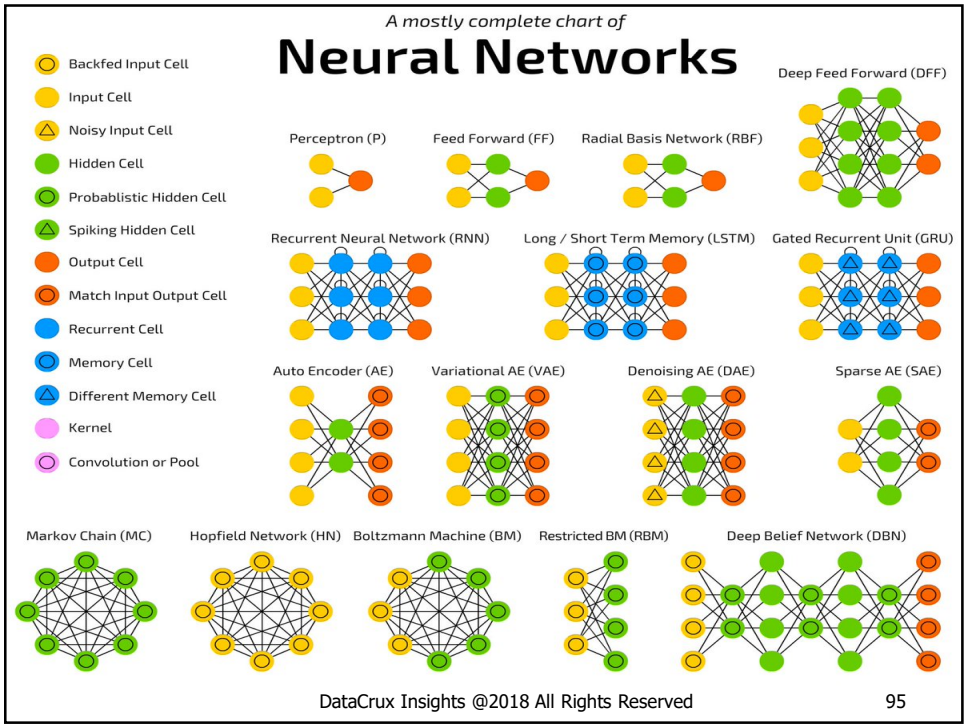
Feedforward NN vs. Recurrent NN



Recurrent neural networks (RNNs) allow cyclical connections.







Convolutional Neural Network

- The process of building a Convolutional Neural Network always involves four major steps.
- **Step - 1 : Convolution**
- **Step - 2 : Pooling**
- **Step - 3 : Flattening**
- **Step - 4 : Full connection**


```
classifier.add(Conv2D(32, (3, 3),
 input_shape = (64, 64, 3), activation =
  'relu'))
```

32=>Number of filters

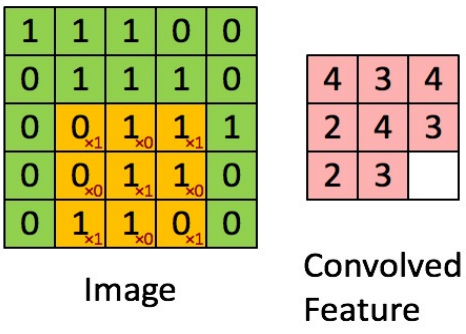
Conv2D ->images , Conv3D ->videos

input_shape=>image shape

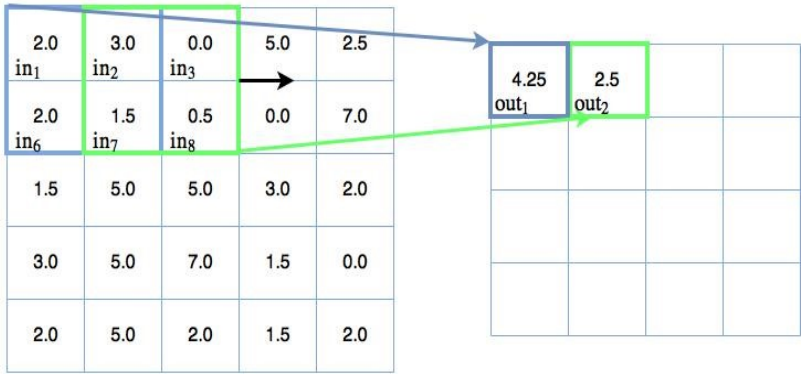
3=>RGB colours

CNN –Filter

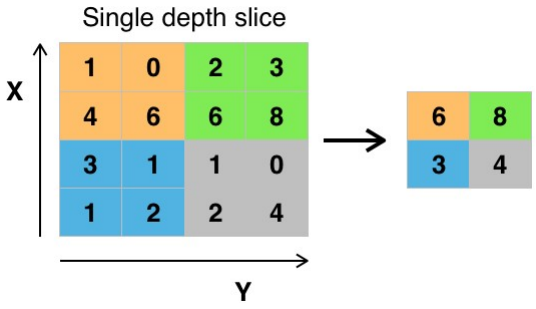
- every network layer acts as a detection filter for the presence of specific features or patterns present in the original data

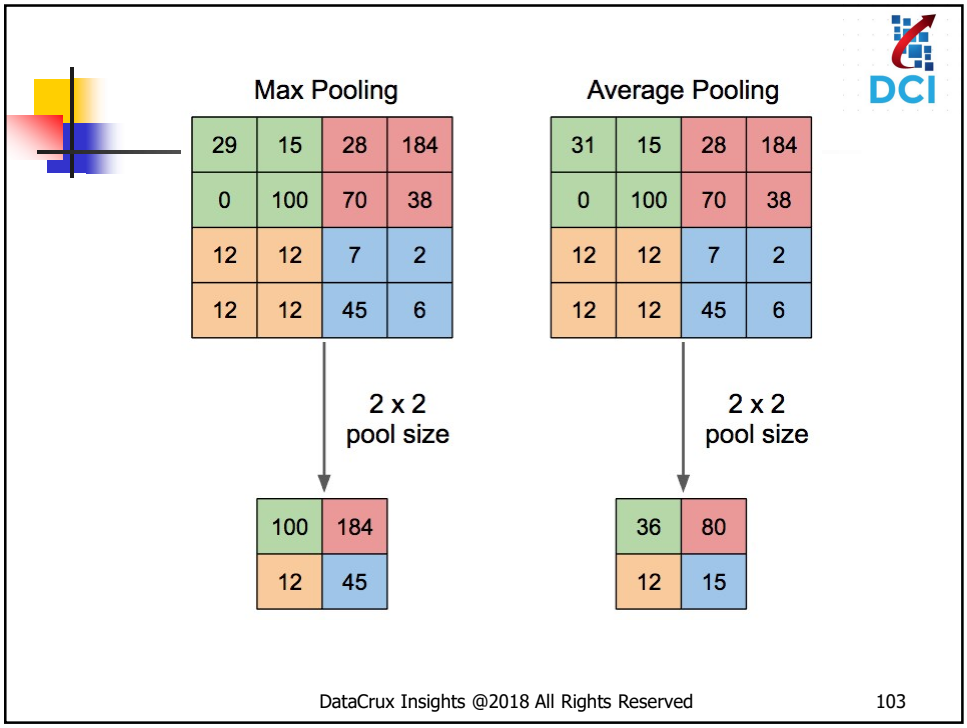


CNN

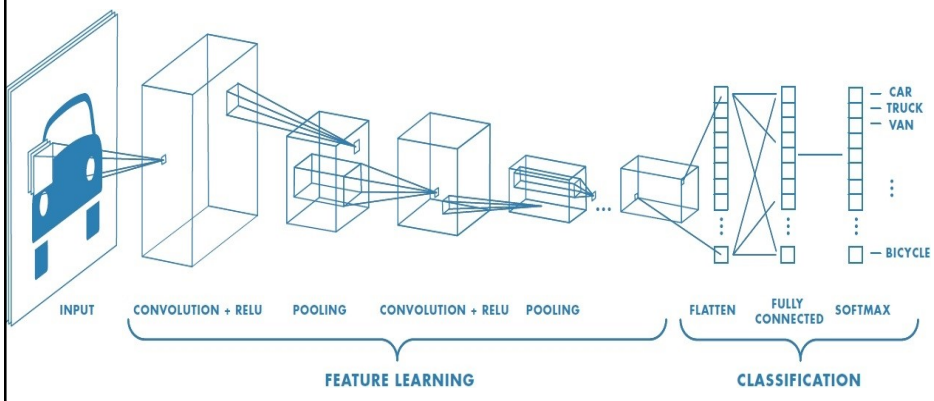


Max pooling

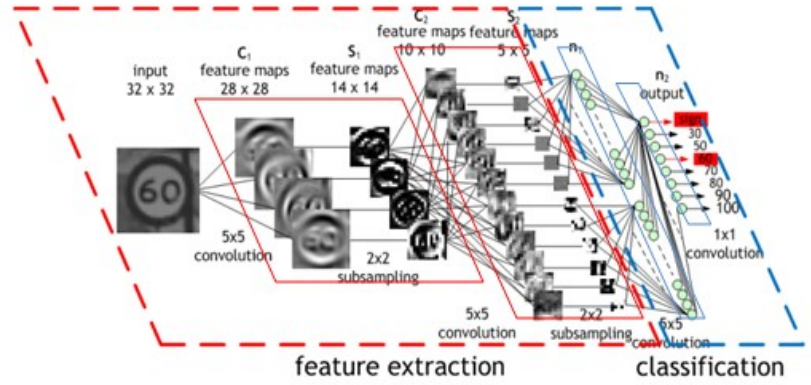




 103



 104



 105