# **AI**, **ML**, **DL**

# under the hood of **BioTech**

Making the interface of **Biology** & **Technology** more intelligent

**Ayon Roy**
**ayonroy2000@gmail.com**

**AI Workshop @ ICGEB, New Delhi**
**19th September 2019**

# Hello **world** !

# I am **Ayon Roy**

Bachelor Of Technology

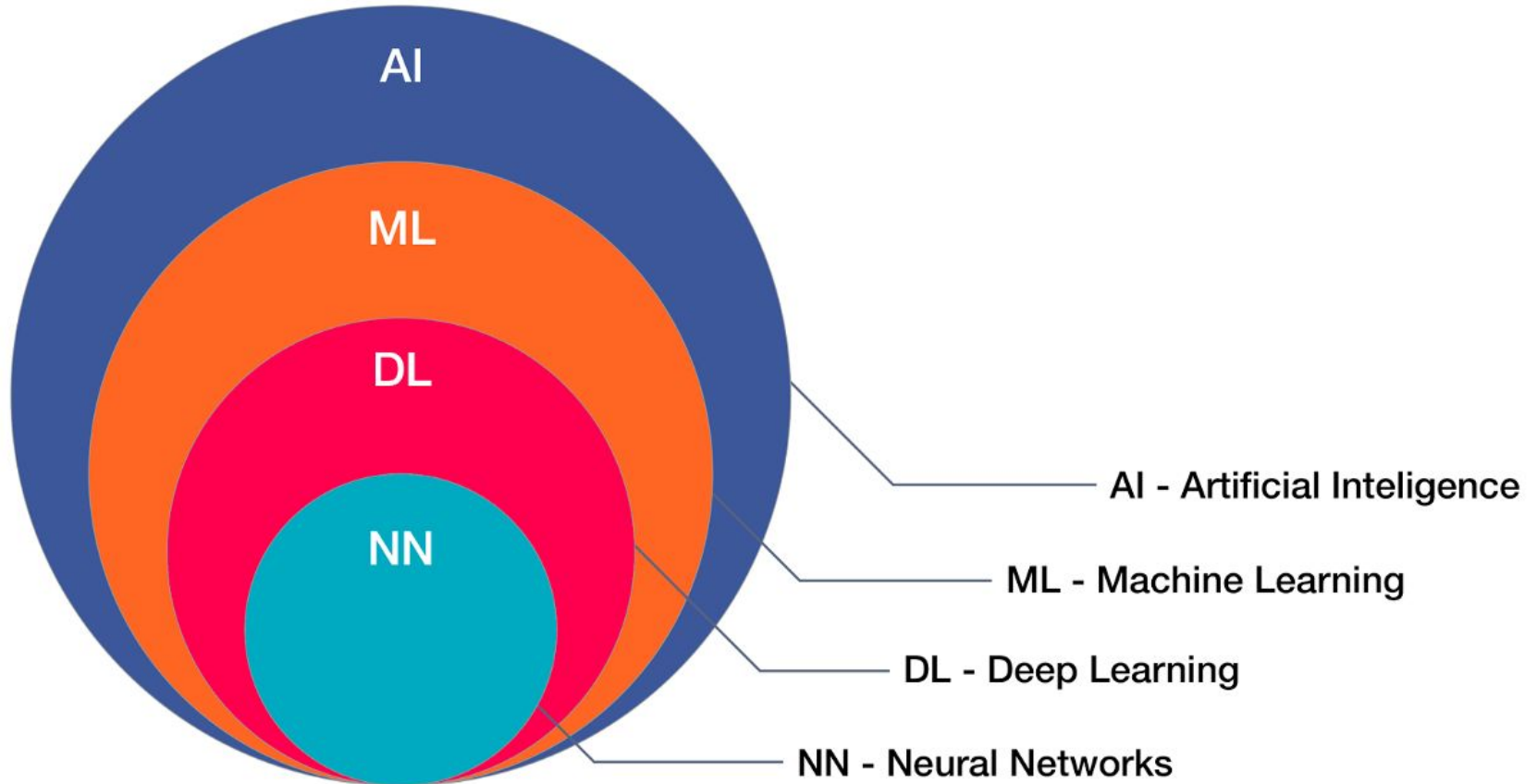Computer Science Engineering ( 2017-2021 )

**Email** : ayonroy2000@gmail.com

**Telegram / Github / LinkedIn Username** : ayonroy2000
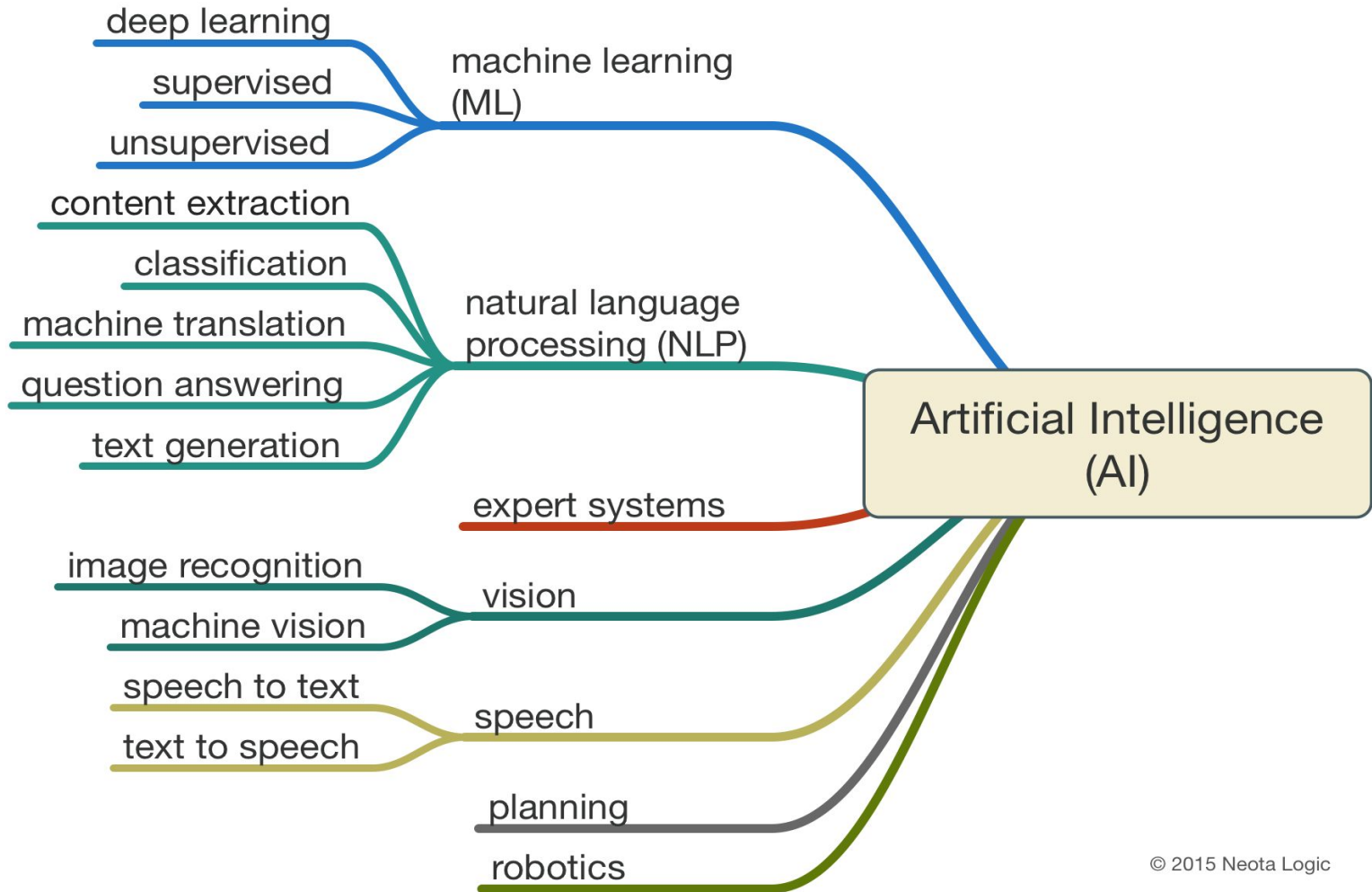
**Website :** https://AYONROY.ML/

# AGENDA ( 19-09-2019 )

- Introduction to Artificial Intelligence, Machine Learning, Deep Learning

- Introduction to some common Deep Learning Libraries

- Introduction to Neural Networks, Deep Neural Networks, Recurrent Neural Networks, Convolutional Neural Networks

- Definition & Applications of commonly used terminologies while developing AI Based Models

- Question & Answer Session

# Graphical Representation of what we will discuss today



AI - Artificial Inteligence

ML - Machine Learning

DL - Deep Learning

NN - Neural Networks

deep learning
supervised
unsupervised
— machine learning (ML)

content extraction
classification
machine translation
question answering
text generation
— natural language processing (NLP)

expert systems

image recognition
machine vision
— vision

speech to text
text to speech
— speech

planning

robotics

Artificial Intelligence (AI)

© 2015 Neota Logic

# But why AI, ML, DL now ?

1. The sharp decrease in costs associated with data storage and processing.

2. The advent of the Internet economy and the explosion in mobile apps.

3. The abundance of open-source tools.

4. The development of a wealth of innovative ML and DL algorithms.

5. Availability of GPUs  etc.

# So, what is AI, ML, DL ?

## Artificial Intelligence

A technique for incorporating human intelligence to machine

## Machine Learning

ML is a subset of AI that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. ML is about learn from past to predict the future.

## Deep Learning

DL is a subset of ML where artificial neural networks, algorithms inspired by the human brain, learn from large amounts of data.
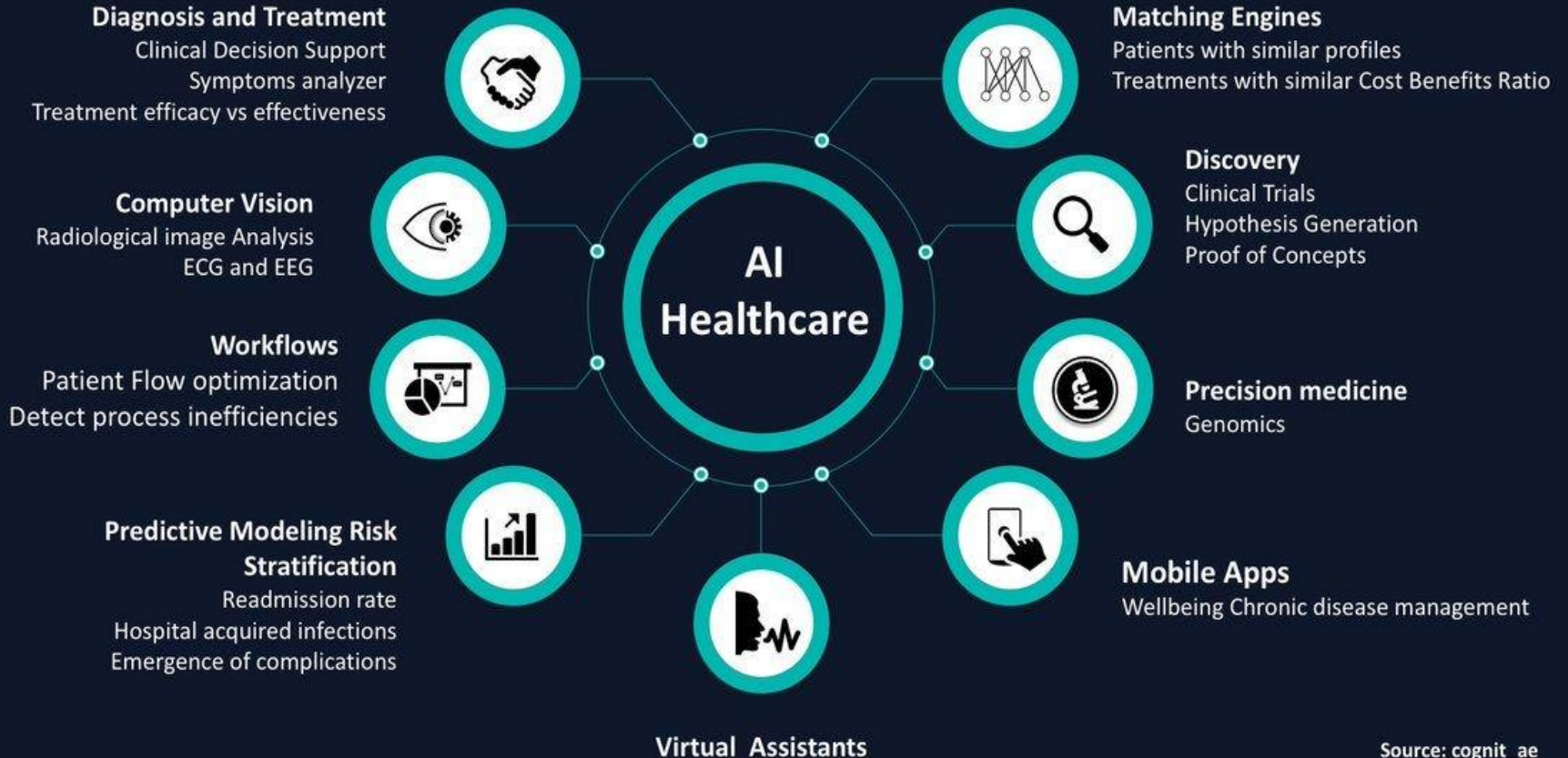
# Defining Artificial Intelligence

**AI** is the *branch of computer science* tasked with the design and construction of **intelligent agents.**

An intelligent agent is in turn any agent that is (mostly):

- **autonomous** ( able to perform tasks without *constant* guidance)
- **adaptable** (able to improve performance by learning from experience)
- **able to sense** and efficiently abstract features of its environment (perception)
- **able to act "rationally"** i.e it is moving towards maximizing a quantitative and objective utility function or performance measure.

# Applications of Artificial Intelligence



**Diagnosis and Treatment**
Clinical Decision Support
Symptoms analyzer
Treatment efficacy vs effectiveness

**Computer Vision**
Radiological image Analysis
ECG and EEG

**Workflows**
Patient Flow optimization
Detect process inefficiencies

**Predictive Modeling Risk Stratification**
Readmission rate
Hospital acquired infections
Emergence of complications

**AI Healthcare**

**Matching Engines**
Patients with similar profiles
Treatments with similar Cost Benefits Ratio

**Discovery**
Clinical Trials
Hypothesis Generation
Proof of Concepts

**Precision medicine**
Genomics

**Mobile Apps**
Wellbeing Chronic disease management

**Virtual Assistants**

Source: cognit_ae

# Defining Machine Learning

**An Approach to Achieve Artificial Intelligence**

**Subfield of AI that aims to teach computers the ability to do tasks with data, without explicit programming.**

We can get AI without using machine learning, but this would require building millions of lines of codes with complex rules and decision-trees.

So instead of hard-coding software routines with specific instructions to accomplish a particular task, machine learning is a way of "training" an algorithm so that it can learn how.

# Applications of Machine Learning

| APPLICATION | POTENTIAL ANNUAL VALUE BY 2026 | KEY DRIVERS FOR ADOPTION |
|---|---|---|
| Robot-assisted surgery | $40B | Technological advances in robotic solutions for more types of surgery |
| Virtual nursing assistants | 20 | Increasing pressure caused by medical labor shortage |
| Administrative workflow | 18 | Easier integration with existing technology infrastructure |
| Fraud detection | 17 | Need to address increasingly complex service and payment fraud attempts |
| Dosage error reduction | 16 | Prevalence of medical errors, which leads to tangible penalties |
| Connected machines | 14 | Proliferation of connected machines/devices |
| Clinical trial participation | 13 | Patent cliff; plethora of data; outcomes-driven approach |
| Preliminary diagnosis | 5 | Interoperability/data architecture to enhance accuracy |
| Automated image diagnosis | 3 | Storage capacity; greater trust in AI technology |
| Cybersecurity | 2 | Increase in breaches; pressure to protect health data |

# Defining Deep Learning

**A Technique for Implementing Machine Learning**

**Subfield of ML that uses specialized techniques involving multi-layer artificial neural networks.**

In ML, data mostly passes through linear transformational algorithms to produce output. But in DL , data goes through multiple number of non-linear transformational algorithms like matrix transformation  to obtain an output.

Other techniques to implement ML include decision tree learning, inductive logic programming, clustering, reinforcement learning, and Bayesian networks etc..

# Applications of Deep Learning

## Pharma's AlphaGo Moment: For the First Time AI Has Designed and Validated a New Drug Candidate in Days

Published on September 3, 2019

moment for Pharma. Insilico Medicine used a generative approach combined with deep reinforcement learning (the form of AI that was used in AlphaGo) to design and validate a new drug candidate end-to-end in 45 days. This is 15x faster than Pharma companies. The experimental validation confirms the ability of AI to accelerate drug discovery.

# Limitations of AI / ML / DL

> The human mind is subtle and extraordinary. You cannot tell a robot to be creative.

The main limitation is that the agents are typically only able to optimize their decisions and actions to achieve a singular goal.

Eg: A computer vision system that is able to detect melanoma from a skin photograph can be called intelligent but it is not capable of performing any other task that an intelligent person would do.

# Some Common Deep Learning Libraries

- Theano

- Tensorflow

- Keras

- PyTorch

**And Many more**

# theano

Theano is an open source library written in Python that implements many features that make it easy to write code for Neural Networks. In addition, Theano makes it very easy to **take advantage of GPU acceleration & performance.**

It **allows us to use back-propagation very easily** by calculating all the derivatives for us.

Know more about it at http://deeplearning.net/software/theano/

# TensorFlow

Tensorflow works very similarly to Theano and **all computations are represented as graphs in Tensorflow**. A Tensorflow graph is a description of the computations.

In Tensorflow, **you don't need to explicitly use of GPU**, rather it will use your GPU if you have one.

Know more about it at https://www.tensorflow.org/

Keras is a neural net Python library that can **run on top of either Theano or Tensorflow, even though it will run by default using Tensorflow.**

Keras run on either CPU or GPU. **It allows to create deep neural networks and makes it easy by using a model for the neural networks.** The main type of model is the _Sequential_ model which creates a linear stack of layers, then you can add new layers using add function.

Know more about it at https://keras.io/ .

# PyTorch

PyTorch is an **open source library based on the Torch library used for applications such as computer vision and natural language processing**. It is primarily developed by Facebook's artificial intelligence research group.

PyTorch provides two high-level features:

- Tensor computing (like NumPy) with strong acceleration via graphics processing units (GPU)
- Deep neural networks built on a tape-based autodiff system

Know more about it at https://pytorch.org/ .

# Neural Networks

# What are Neural Networks ?

Computer System inspired by biological networks of neurons that learn progressively i.e which improves performance to do tasks; by considering examples generally without task specific programming.

# Why Neural Networks ?

Neural Networks learn by example. So now computers can do things what we don't exactly know how to do.

# Applications of Neural Networks

Speech Recognition , Character Recognition, Signature Verification, Face Recognition etc.

# Single Perceptron
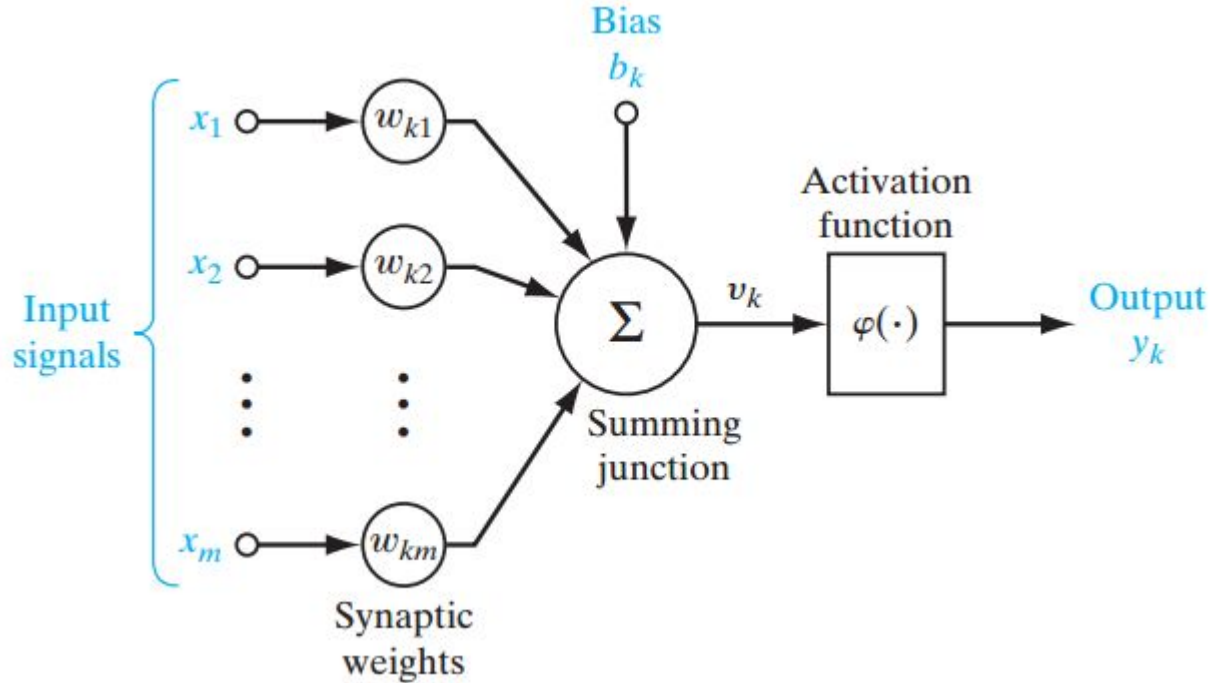


Many such Perceptrons combine together to form a Neural Network

# Working of a Neural Network
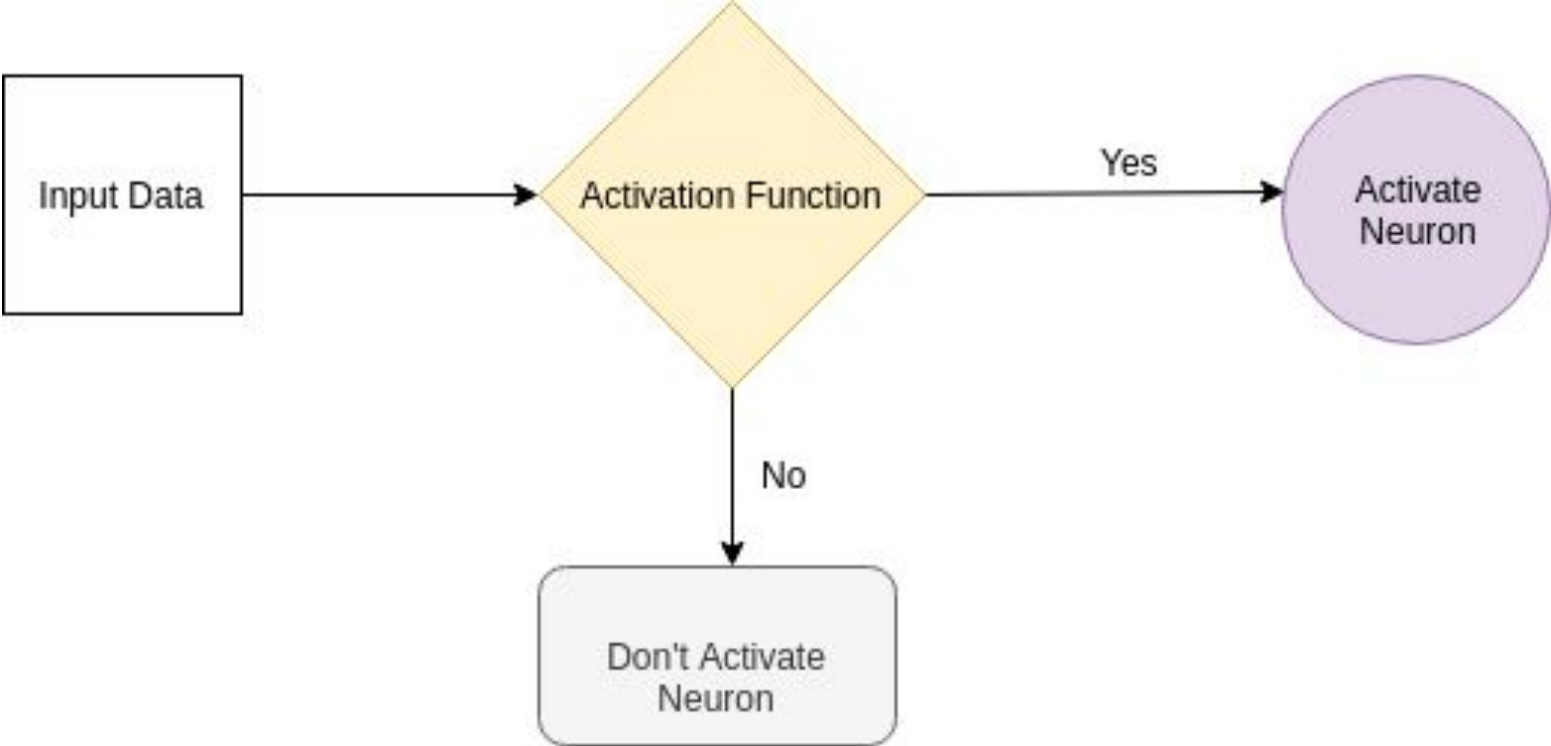
# Basic Terminologies

One **epoch** = one forward pass and one backward pass of *all* the training examples

**Batch size** = the number of training examples in one forward/backward pass. The higher the batch size, the more memory space you'll need.

No. of **iterations** = number of passes, each pass using [batch size] number of examples. To be clear, one pass = one forward pass + one backward pass (we do not count the forward pass and backward pass as two different passes).

**Example**

**if you have 1000 training examples, and your batch size is 500, then it will take 2 iterations to complete 1 epoch.**

**Loss Function**

Method of evaluating how well your algorithm models your dataset. If the predictions are totally off, it will output a higher number. If they're pretty good, it'll output a lower number. As you change pieces of your algorithm to try and improve your model, your loss function will tell you if you're getting anywhere.

**Optimizers**

During the training process, we change the parameters (weights) of our model to minimize the loss function, and make our predictions better. But how to do that? Optimizers tie together the loss function and model parameters to shape and mold the model into its most accurate possible form.

**The loss function is the guide to the terrain, telling the optimizer when it's moving in the right or wrong direction.**

**Learning Rate :** The learning rate is a parameter that determines how much an updating step influences the current value of the weights. When training a neural network, if the learning rate is :

- **Too small** a learning rate and your neural network may not learn at all
- **Too large** a learning rate and you may overshoot areas of low loss (or even overfit from the start of training)

**Momentum :** It is a value between 0 and 1 that increases the size of the steps taken towards the minimum by trying to jump from a local minima.

- If the momentum term is large then the learning rate should be kept smaller.
- A large value of momentum also means that the convergence will happen fast.
- But if both the momentum and learning rate are kept at large values, then you might skip the minimum with a huge step.
- A small value of momentum cannot reliably avoid local minima, and can also slow down the training of the system.

A right value of momentum can be either learned by hit and trial or through cross-validation.

Gradient Descent converging

Gradient Descent diverging (stepsize too large)

w0

w1
w2

w3
w1

w0 w2
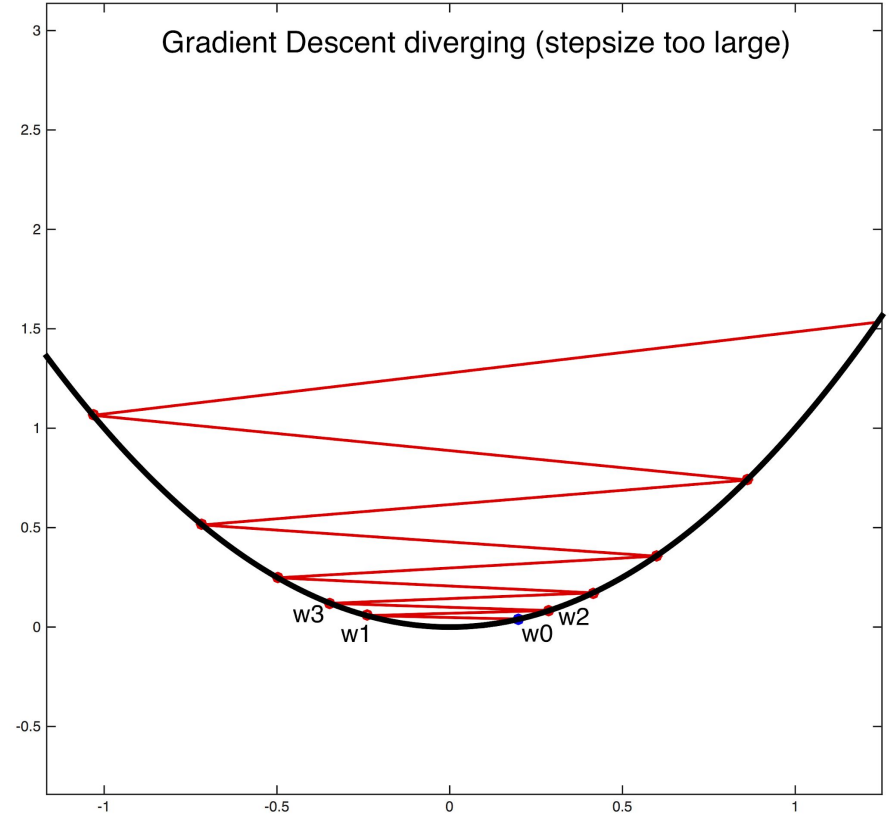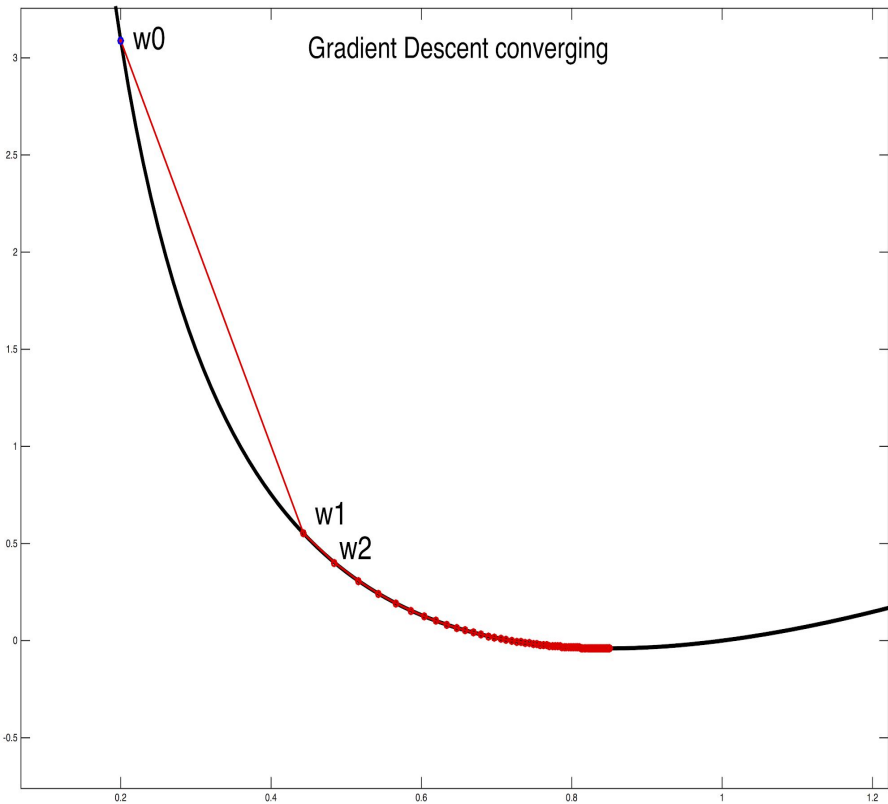
Gradient Descent is a way to find the minimum of a function.

**Flatten** : The process of converting all the resultant 2 dimensional arrays into a single long continuous linear vector. This means you can combine all the found local features of the previous convolutional layers. Each feature map channel in the output of a CNN layer is a "flattened" 2D array created by adding the results of multiple 2D kernels (one for each channel in the input layer).

**Batch Normalization :** A technique for improving the performance and stability of artificial neural networks. It is a technique to provide any layer in a neural network with inputs that are zero mean/unit variance. Debates are there that it does not reduce internal covariate shift, but rather smooths the objective function to improve the performance and it also helps in length-direction decoupling, and thereby accelerates neural networks.

**Decay** : An additional term in the weight update rule that causes the weights to exponentially decay to zero, if no other update is scheduled.

**Dense Layer :** The most basic neural network architecture in deep learning containing the densely connected or fully-connected layers. In this layer, all the inputs and outputs are connected to all the neurons in each layer.

**Dropout** refers to not considering neurons during particular forward or backward pass during the training phase of certain set of neurons which is chosen at random. It *is done "to prevent over-fitting"*. A fully connected layer occupies most of the parameters, and hence, neurons develop co-dependency amongst each other during training which curbs the individual power of each neuron leading to over-fitting of training data.

# Hyperparameter Tuning

Tuning the hyperparameters effectively can lead to a massive improvement in the overall performance.

Following are a few common hyperparameters we frequently work with in a deep neural network:

- Learning rate – α
- Momentum – β
- Adam's hyperparameter – β1, β2, ε
- Number of hidden layers
- Number of hidden units for different layers
- Learning rate decay
- Mini-batch size

Learning rate usually proves to be the most important among the above. This is followed by the number of hidden units, momentum, mini-batch size, the number of hidden layers, and then the learning rate decay.

**Hyperparameter tuning relies more on experimental results than theory, and thus the best method to determine the optimal settings is to try many different combinations evaluate the performance of each model.**

# Activation Functions

**Sigmoid**

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

**tanh**

$\tanh(x)$

**ReLU**

$\max(0, x)$
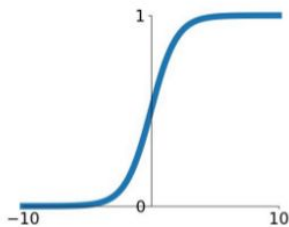
**Leaky ReLU**
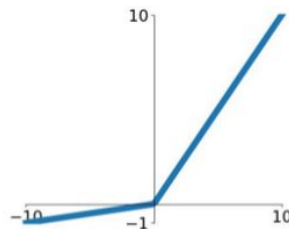
$\max(0.1x, x)$
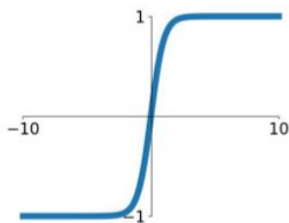
**Maxout**

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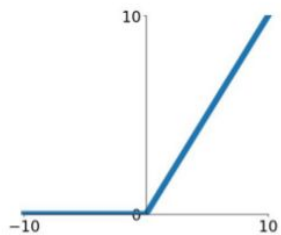
**ELU**

$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$
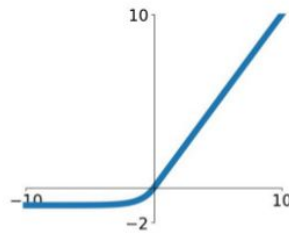
providing lift for classification and forecasting models



*Deep*
Neural
Networks
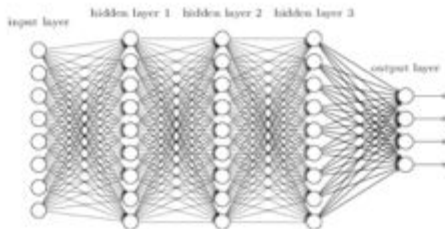
feature extraction and classification of images



*Convolutional*
Neural
Networks
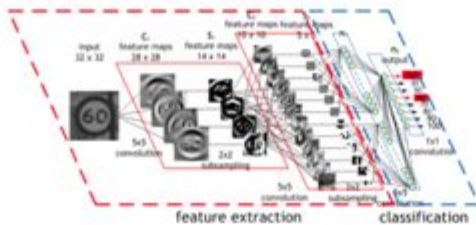
for sequence of events, language models, time series, etc.



*Recurrent*
Neural
Networks
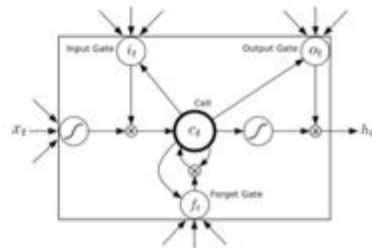
# Deep Neural Network



"Non-deep" feedforward neural network

input layer

hidden layer

output layer

Deep neural network

input layer

hidden layer 1   hidden layer 2   hidden layer 3

output layer

**Effect of Depth**

Number of hidden layers vs Test accuracy (%)

# Convolutional Neural Network



| 1st Convolution + ReLU | 1st Pooling | 2nd Convolution + ReLU | 2nd Pooling | Fully Connected | Fully Connected | Output Predictions |

dog (0.01)
cat (0.04)
boat (0.94)
bird (0.02)
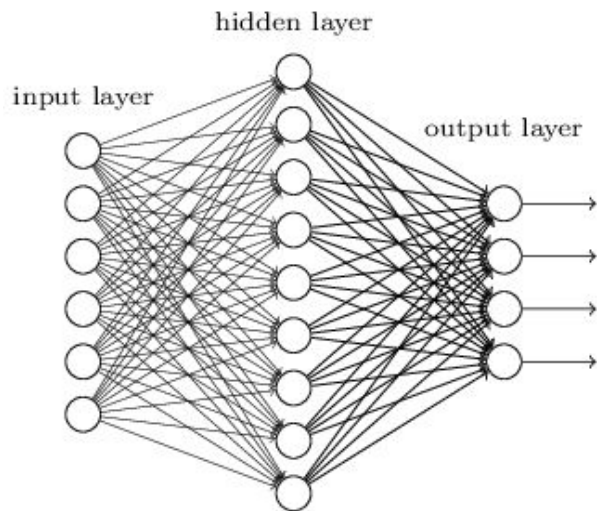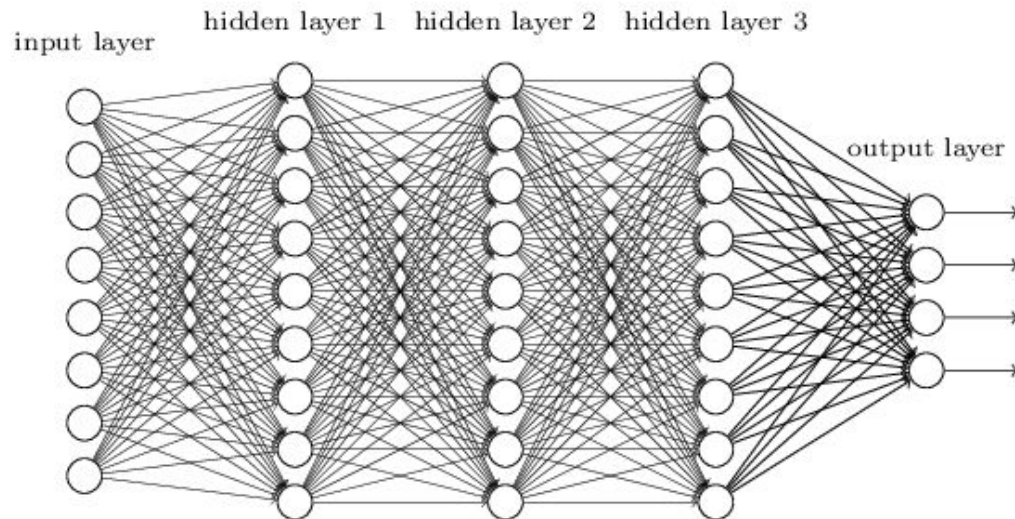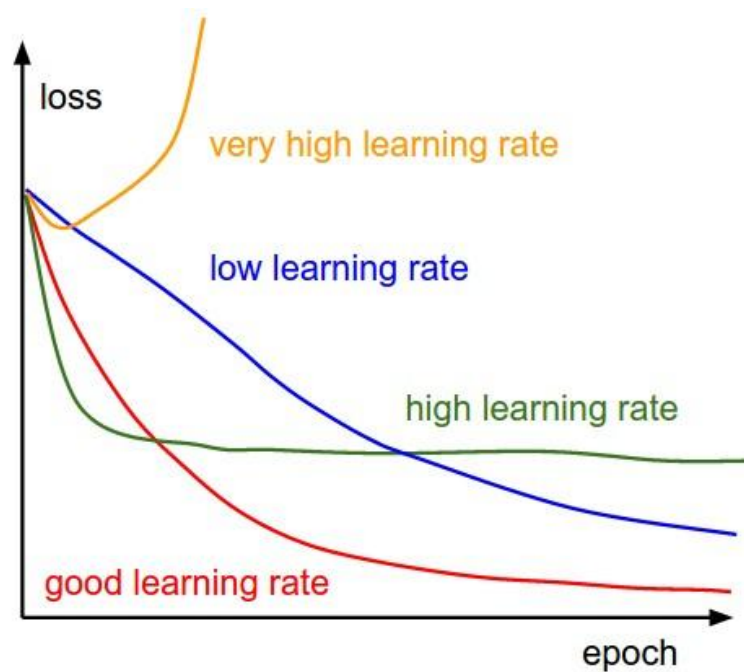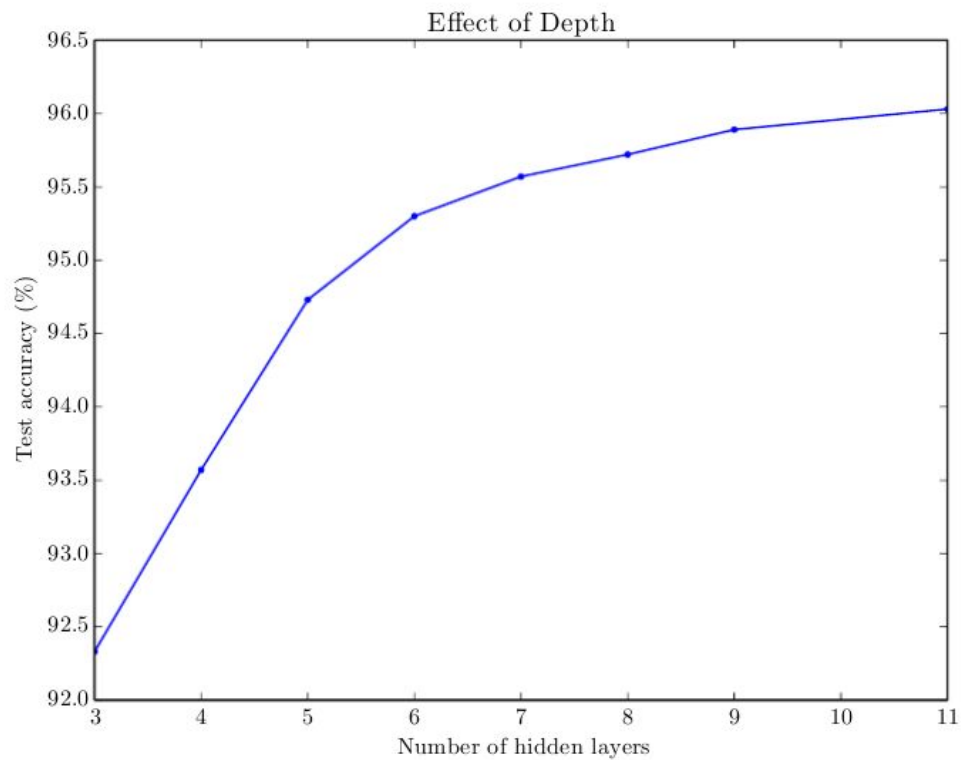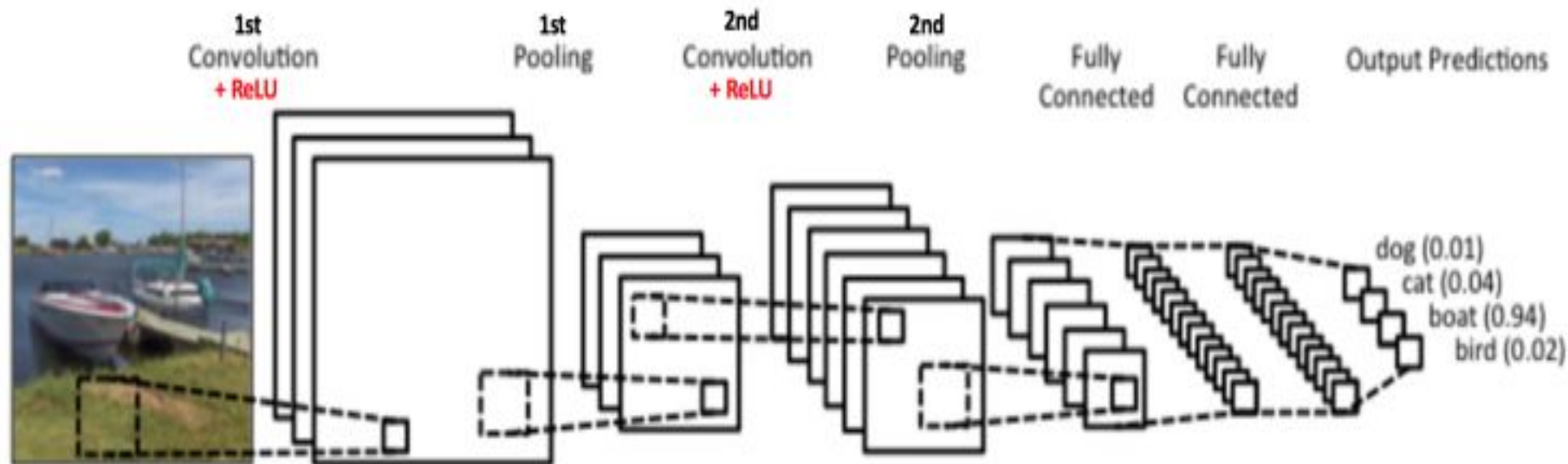
Input -> Conv -> ReLU -> Conv -> ReLU -> Pool -> ReLU -> Conv -> ReLU -> Pool ->Fully Connected

**Input Image**

$m \times n \times 3$ image

Input

Convolution, Pooling, and ReLU

max

Fully Connected

Softmax Output

# Convolutional Layer



Original Image

Feature Matrix
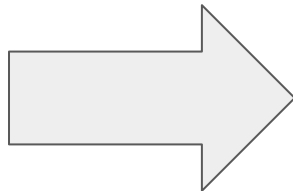
Image

Convolved Feature

# ReLU Layer



convolution

Biased ON/OFF ReLU

next convolution

# Pooling Layer



Max(1, 1, 5, 6) = 6

max pool with 2x2 filters and stride 2

Rectified Feature Map
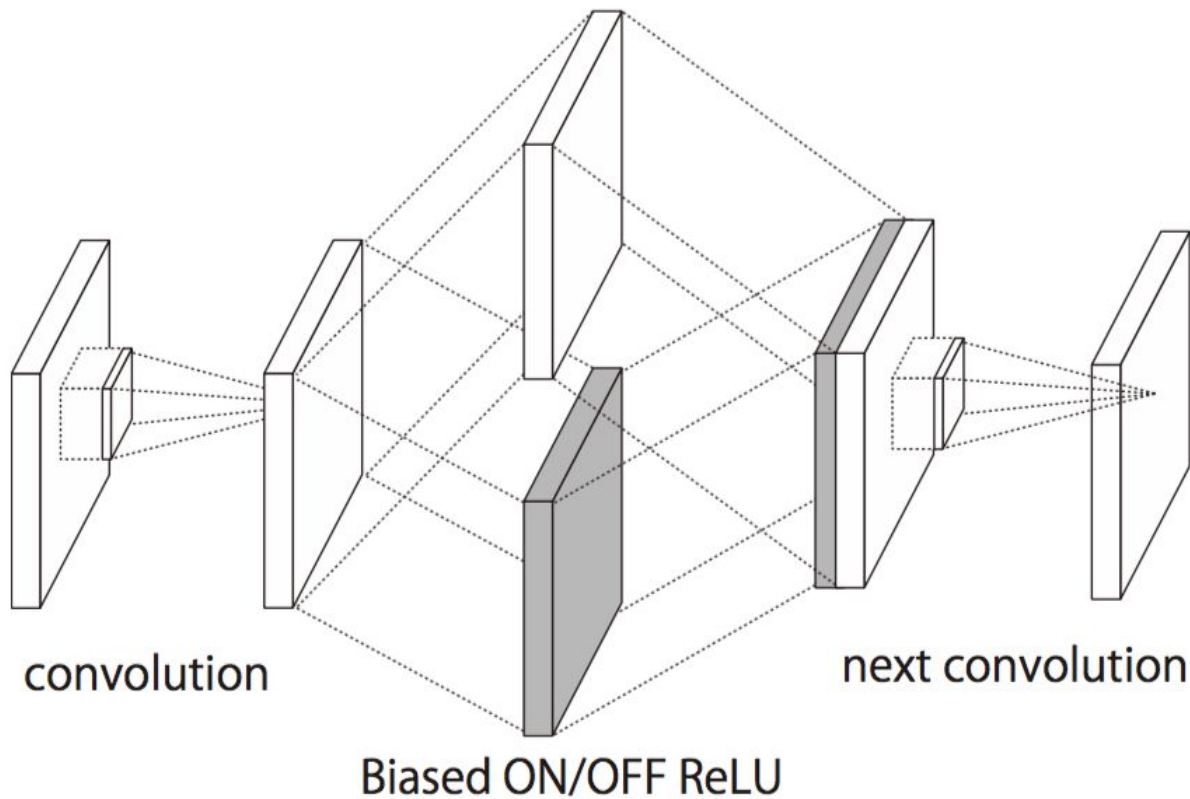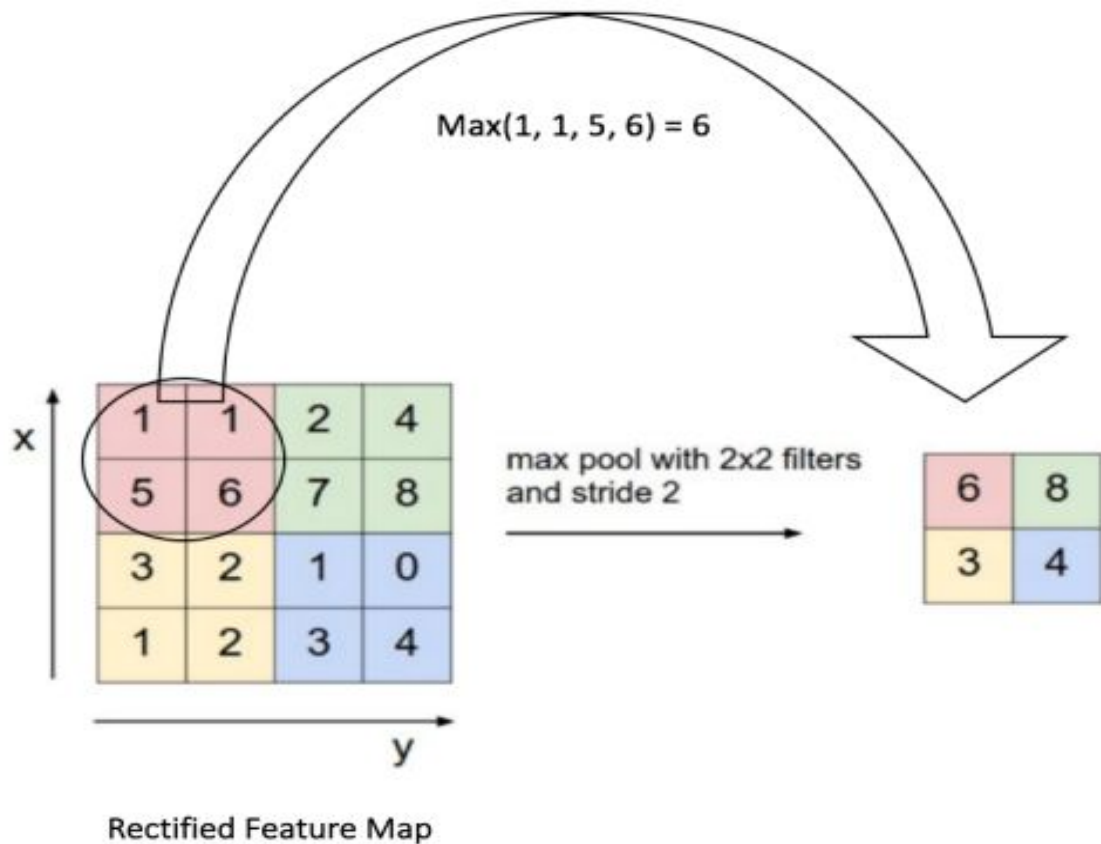
# Final Connected Layer
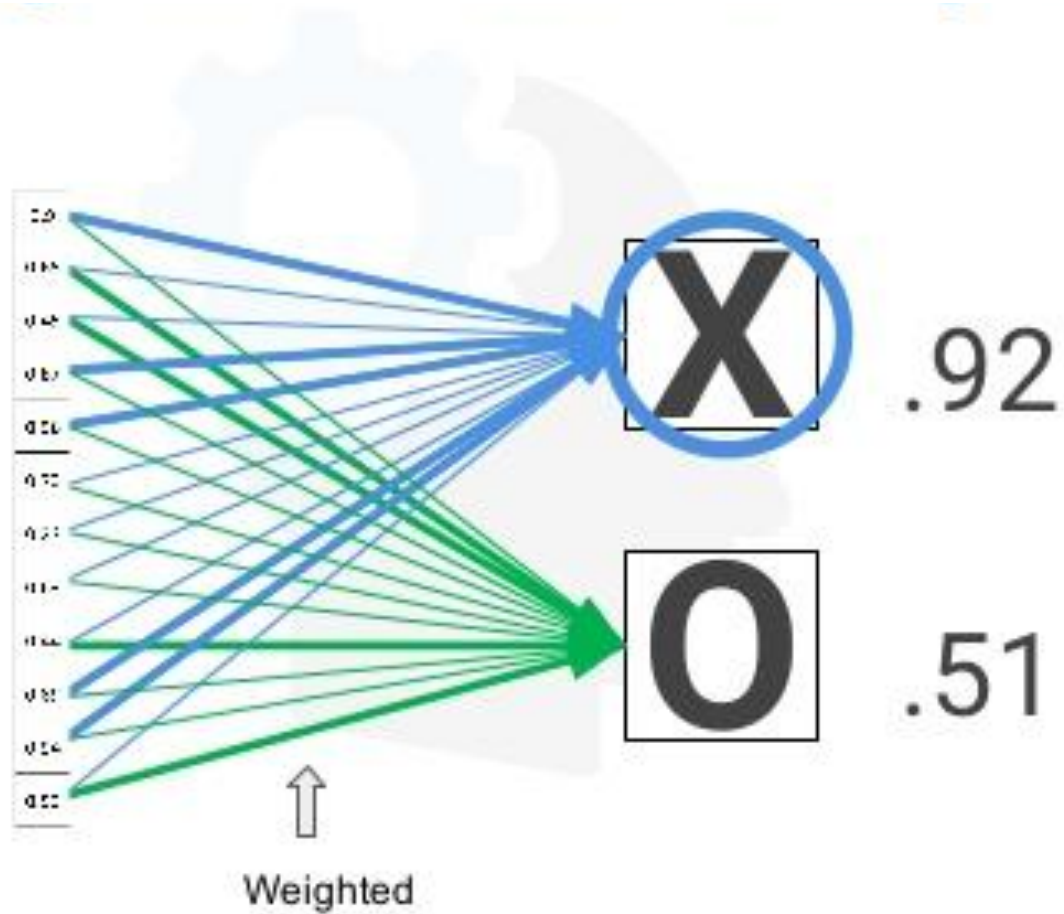
# SoftMax Layer & Sigmoid Activation Function

| Soft-Max | Sigmoid |
|---|---|
| Used for multi-classification in logistic regression model. | Used for binary classification in logistic regression model. |
| The probabilities sum will be 1 | The probabilities sum need not be 1 |
| Used in the different layers of neural networks. | Used as activation function while building neural networks |
| The high value will have the higher probability than other values. | The high value will have the high probability but not the higher |

# Recurrent Neural Network
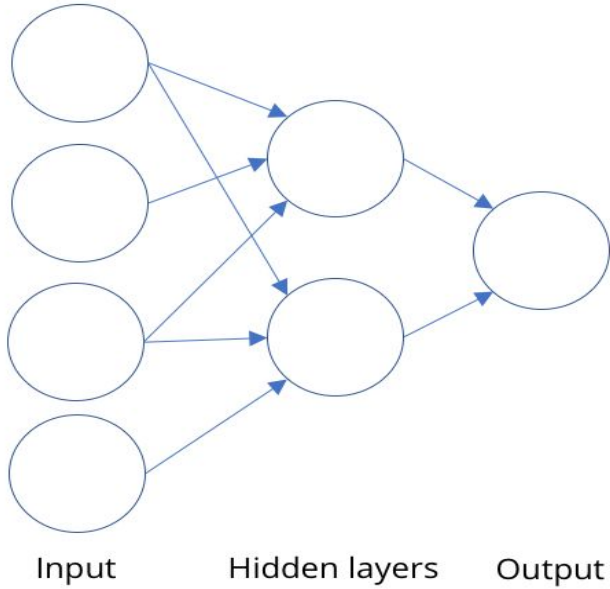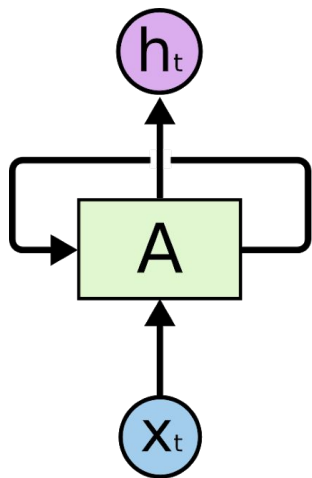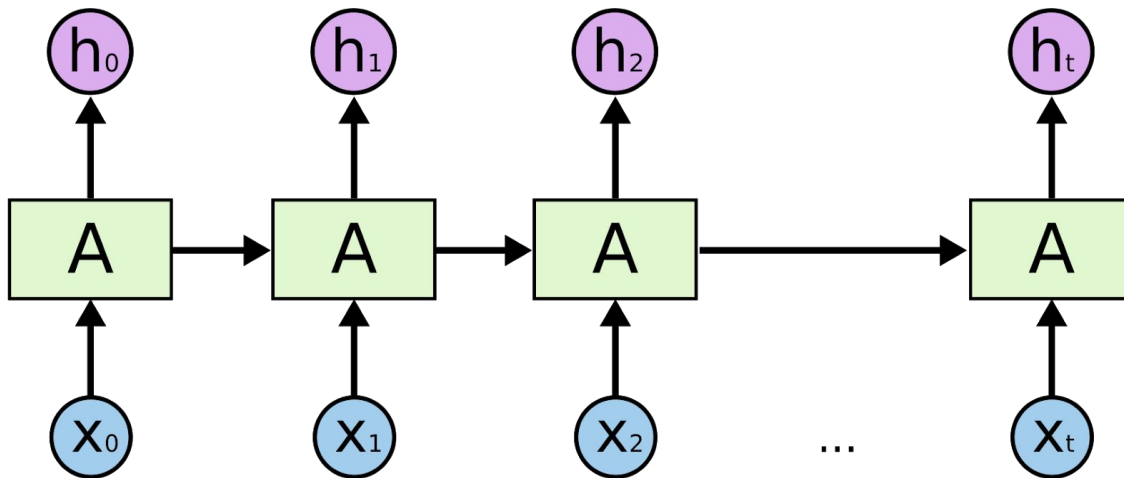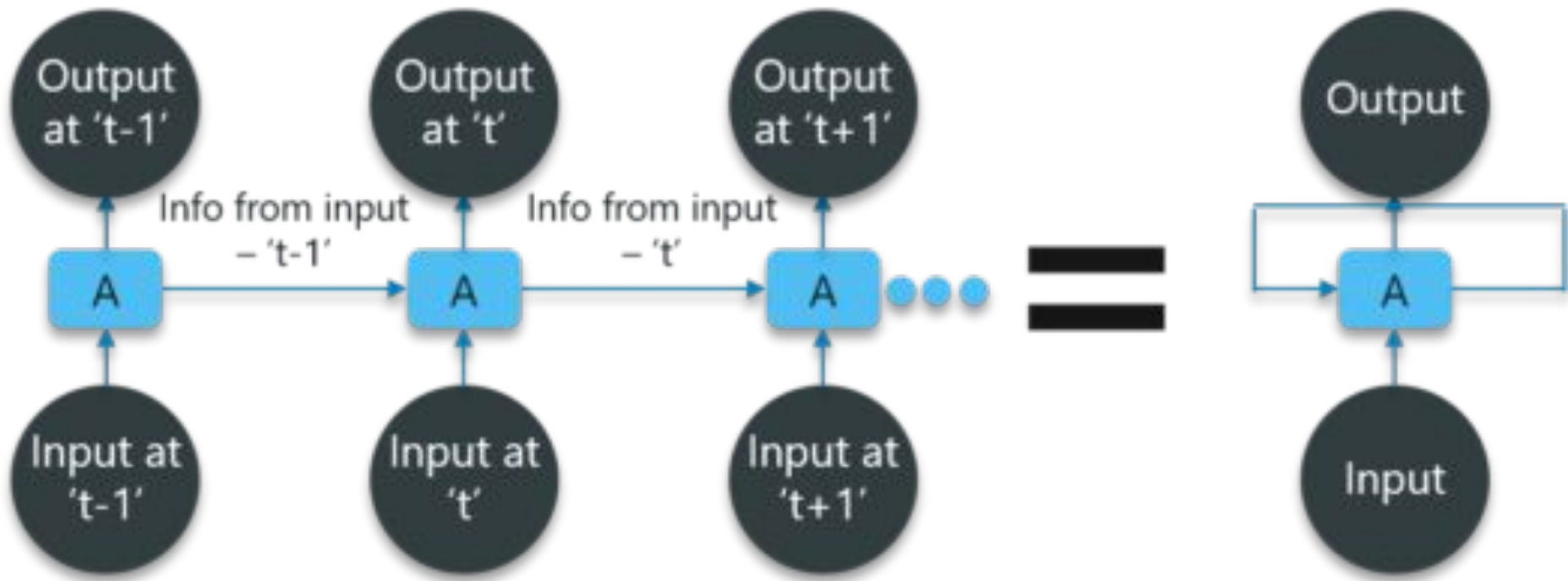


Typical neural network
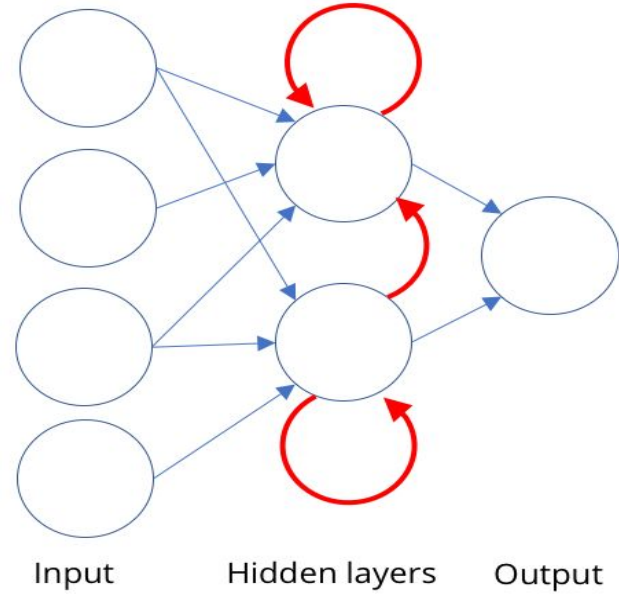
Recurrent neural network
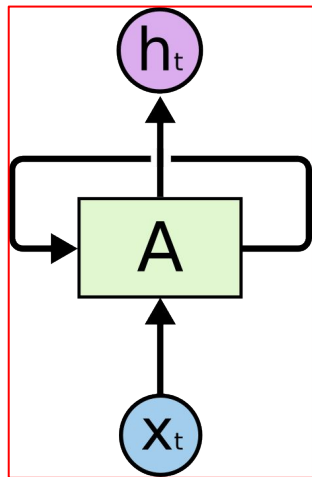
**Structure of a RNN**

But RNNs uses backpropagation algorithm for every timestamp in order to train the Neural Network. Back propagation has 2 major issues - Vanishing Gradient , Exploding Gradient .

There are concepts like LSTMs etc. to overcome these issues.

# Danke Scheon

Questions ? Any Feedbacks ? Did you like the talk?
Tell me about it.

If you think I can help you,
connect with me via

**Email** : ayonroy2000@gmail.com

**LinkedIn / Github / Telegram Username** : ayonroy2000

**Website** : https://AYONROY.ML/