K.S.RANGASAMY COLLEGE OF TECHNOLOGY

COMPUTER SOCIETY OF INDIA | STUDENT CHAPTER

PRESENTS

# WORKSHOP ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE


PRESENTED BY

ABISHEK HARI

# Welcome You All

# LETS PAY TRIBUTE
# TO OUR GREAT DEPARTED SOULS
# AT PULWAMA ATTACK

" JAI HIND "

# CONTENT (SESSION 1 | THEORY)

- Introduction to Machine Learning and Artificial Intelligence
- Machine Learning and Artificial Intelligence Concepts
- Uses of Machine Learning and Artificial Intelligence
- Development of Machine Learning and Artificial Intelligence
- Trends in the Machine Learning and Artificial Intelligence Domains
- Algorithms of Machine Learning and Artificial Intelligence
- Deep learning Concepts (Theory)
- Introduction to Hands-on Session

# CONTENT (SESSION 2 | DEMO)

- Software Installation
- Environmental Introduction
- Demo 1 on Object Detection Application
- Demo 2 on Sentiment Analysis
- Queries
- Feedback

# SESSION 1

# (THEORY)

# GET SET GO

# INTRODUCTION

## ARTIFICIAL INTELLIGENCE :

- "The science and engineering of making intelligent machines, especially intelligent computer programs". – John McCarthy
- Way of making computer think intelligently, in the similar manner an intelligent human thinks.
- "Can a machine think and behave like humans do?"
- Goals - To create expert systems, to implement human intelligence in machines.
- Artificial intelligence is a science and technology based on disciplines such as Computer Science, Biology, Psychology, Linguistics, Mathematics, and Engineering.
- A major thrust of AI is in the development of computer functions associated with human intelligence, such as reasoning, learning, and problem solving.
- Applications - Gaming, natural language processing, expert systems, vision systems, speech recognition, handwriting recognition, intelligent robots.

## HISTORY OF ARTIFICIAL INTELLIGENCE :

| YEAR | MILESTONE / INNOVATION |
|---|---|
| 1943 | Foundation for neural networks laid. |
| 1950 | Alan Turing introduced Turing Test for evaluation of intelligence and published Computing Machinery and Intelligence. |
| 1956 | John McCarthy coined the term *Artificial Intelligence*. |
| 1958 | John McCarthy invents LISP programming language for AI. |
| 1979 | The first computer-controlled autonomous vehicle, Stanford Cart, was built. |
| 1990 | Major advances in all areas of AI : Scheduling, Data mining, Natural Language Processing, Vision, VR, Gaming |
| 1997 | The Deep Blue Chess Program beats the then world chess champion, Garry Kasparov. |
| 2000's | Still Evolving |

## MACHINE LEARNING:

- Machine learning is a subfield of artificial intelligence (AI).
- Machine Learning is an idea to learn from examples and experience, without being explicitly programmed.
- Instead of writing code, you feed data to the generic algorithm, and it builds logic based on the data given.
- A computer program is said to learn from experience E with some class of tasks T and performance measure P. - Tom M. Mitchell
- Applications – Facial Recognition, Computer Vision, Object Detection, Email Filtering, Medical Diagnosis, Weather Prediction, Video Prediction.
- Machine Learning and Deep Learning are just ways to achieve Artificial Intelligence.

## ARTIFICIAL INTELLIGENCE CONCEPTS

- Machine Learning
- Black Box
- Neural Network
- Deep Learning
- Natural Language Processing (NLP)
- Reinforcement Learning
- Supervised Learning
- Unsupervised Learning
- Transfer Learning
- Turing Test

## MACHINE LEARNING

- Machine learning is the process by which an AI uses algorithm to perform artificial intelligence functions, It's the result of applying rules to create outcomes through an AI.

## BLACK BOX

- When the rules are applied an AI does a lot of complex math. This math, often, can't even be understood by humans, yet the system outputs useful information. When this happens it's called as black box learning.

## NEURAL NETWORK

- When we want an AI to get better at something we create a neural network. These networks are designed to be very similar to the human nervous system and brain. It uses stages of learning to give AI the ability to solve complex problems by breaking them down into levels of data

## DEEP LEARNING

- Deep Learning is what happens when a neural network gets to work. As the layers process data the AI gains a basic understanding. You might be teaching your AI to understand cats, but once it learns what paws are that AI can apply that knowledge to a different task. Deep Learning means that instead of understanding what something is, the AI begins to learn “why”.

## NATURAL LANGUAGE PROCESSING

- It takes an advanced neural network to parse human language. When an AI trained to interpret human communication it's called natural language processing. This is useful for chatbots and translation services, but it's also represented at the cutting edge by AI assistants like Alexa and Siri.

## REINFORCEMENT LEARNING

- AI is a lot more like humans than we might be comfortable believing. We learn in almost the exact same way. One method of teaching a machine, just like a person, is to use reinforcement learning. This involves giving the AI a goal that isn't defined with a specific metric, Instead of finding one specific answer the AI will run scenarios and report results, The AI takes the feedback and adjusts the next scenario to achieve better results.

## SUPERVISED LEARNING

- This is the very serious business of proving things. When you train an AI model using a supervised learning method you provide the machine with the correct answer ahead of time. Basically the AI knows the answer and it knows the question. This is the most common method of training because it yields the most data: it defines patterns between the question and answer.

## UNSUPERVISED LEARNING

- In many ways the spookiest part of AI research is realizing that the machines are really capable of learning, and they're using layers upon layers of data and processing capability to do so. With unsupervised learning we don't give the AI an answer. Rather than finding patterns that are predefined like, "why people choose one brand over another," we simply feed a machine a bunch of data so that it can find whatever patterns it is able to.

## TRANSFER LEARNING

- Another spooky way machines can learn is through transfer learning. Once an AI has successfully learned something, like how to determine if an image is a cat or not, it can continue to build on it's knowledge even if you aren't asking it to learn anything about cats. You could take an AI that can determine if an image is a cat with 90-percent accuracy, hypothetically, and after it spent a week training on identifying shoes it could then return to its work on cats with a noticeable improvement in accuracy.

## TURING TEST

- The test was originally conceived as a way of determining if a human could be fooled by a conversation, in text display only, between a human and an artificial intelligence, it has since become short hand for any AI that can fool a person into believing they're seeing or interacting with a real person.

# USES

- AI in Marketing
- AI in Banking
- AI in Finance
- AI in Agriculture
- AI in Healthcare



## AI IN MARKETING :

- Marketing is a way to sugar coat your products to attract more customers.
- We, humans, are pretty good at sugar coating, but what if an algorithm or a bot is built solely for the purpose of marketing a brand or a company? It would do a pretty awesome job!
- Searching a product in the internet in the late 2000s vs. present.
- Example - Netflix provides highly accurate predictive technology based on customer's reactions to films.
- With the growing advancement in AI, in the near future, it may be possible for consumers on the web to buy products by snapping a photo of it.

## AI IN BANKING :

- AI in banking is growing faster than you thought! AI-based systems provide customer support, detect anomalies and credit card frauds.
- Example - HDFC Bank has developed an AI-based chatbot called EVA (Electronic Virtual Assistant).
- AI solutions can be used to enhance security across a number of business sectors, including retail and finance.
- By tracing card usage and endpoint access, security specialists are more effectively preventing fraud
- Companies such as MasterCard have relied on AI and Deep Learning to detect fraudulent transaction patterns.

## AI IN FINANCE :

- Ventures have been relying on computers and data scientists to determine future patterns in the market.
- Machines can learn to observe patterns in past data and predict how these patterns might repeat in the future.
- In the age of ultra-high-frequency trading, financial organizations are turning to AI to improve their stock trading performance and boost profit.
- Example - Nomura Securities. Pursued one goal, i.e. to analyze the insights of experienced stock traders with the help of computers.
- The system stored huge data of price and trading data in its computer. This made them predict the future trading based in the historical data.

## AI IN AGRICULTURE :

- Here's an alarming fact, the world will need to produce 50 percent more food by 2050. The only way this can be possible is if we use our resources more carefully.
- AI can help farmers get more from the land while using resources more sustainably.
- Issues such as climate change, population growth and food security concerns have pushed the industry into seeking more innovative approaches to improve crop yield.
- Example - Robot called See & Spray which uses computer vision technologies like object detection.

## AI IN HEALTHCARE :

- When it comes to saving our lives, a lot of organizations and medical care centers are relying on AI.
- An organization called Cambio Health Care developed a clinical decision support system for stroke prevention that can give the physician a warning when there's a patient at risk of having a heart stroke.
- Another such example is Coala life which is a company that has a digitalized device that can find cardiac diseases.
- Similarly, Aifloo is developing a system for keeping track of how people are doing in nursing homes, home care, etc. The best thing about AI in healthcare is that you don't even need to develop a new medication. Just by using an existing medication in the right way, you can also save lives.

# DEVELOPMENT

- Builder Bot
- Basic Python - Binance Bot
- Advanced Crypto Trading Bot
- Sentiment Analysis Bot
- Trading View Signals Analysis Bot

# TRENDS

## ARTIFICIAL INTELLIGENCE TRENDS

1. The rise of AI-enabled chips.
2. Convergence of IoT and AI at the edge.
3. Interoperability among neural networks becomes key.
4. Automated machine learning will gain prominence.
5. AI will automate DevOps through AIOps.

## MACHINE LEARNING TRENDS

1. Democratization of Machine Learning
2. Rise in Platform Wars
3. Data Scientist Will Become the Hottest Job
4. Robotic Process Automation
5. Impact on Cyber security

# BREAK

## ( BACK IN 15 MINUTES )

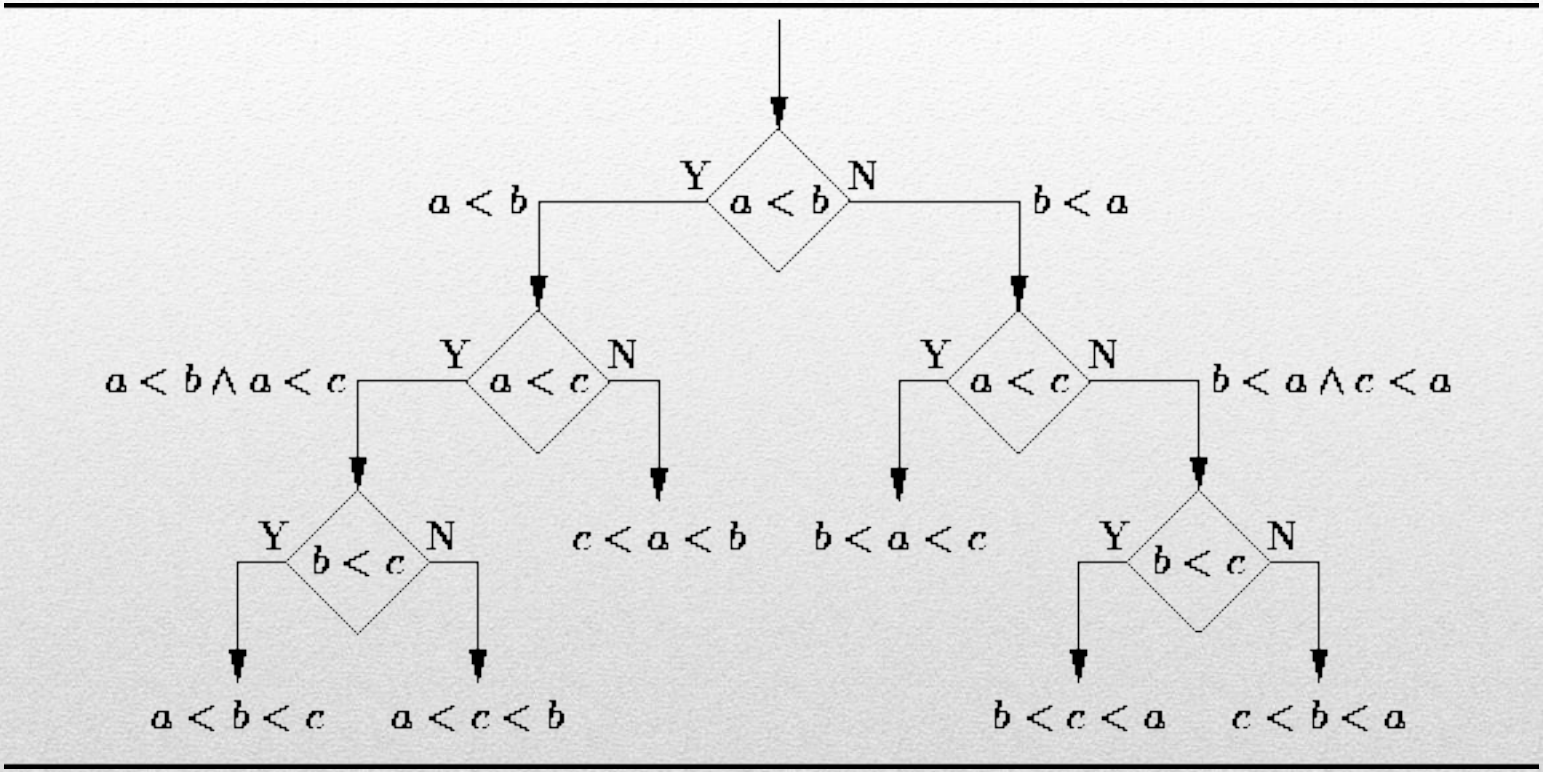
TO BE CONTINUED...

# Algorithms of Machine Learning and Artificial Intelligence

- Decision Trees
- Native Bayes Classification
- Support Vector Machines
- Clustering Algorithms
- Singular Value Decomposition
- Independent Component Analysis

## 1. DECISION TREE

- A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance-event outcomes, resource costs, and utility.
- From a business decision point of view, a decision tree is the minimum number of yes/no questions that one has to ask, to assess the probability of making a correct decision, most of the time. As a method, it allows you to approach the problem in a systematic way to arrive at a logical conclusion.


a < b
Y
a < b
N
b < a
a < b ∧ a < c
Y
a < c
N
Y
a < c
N
b < a ∧ c < a
Y
b < c
N
c < a < b
b < a < c
Y
b < c
N
a < b < c
a < c < b
b < c < a
c < b < a

## 2. NATIVE BAYES CLASSIFICATION

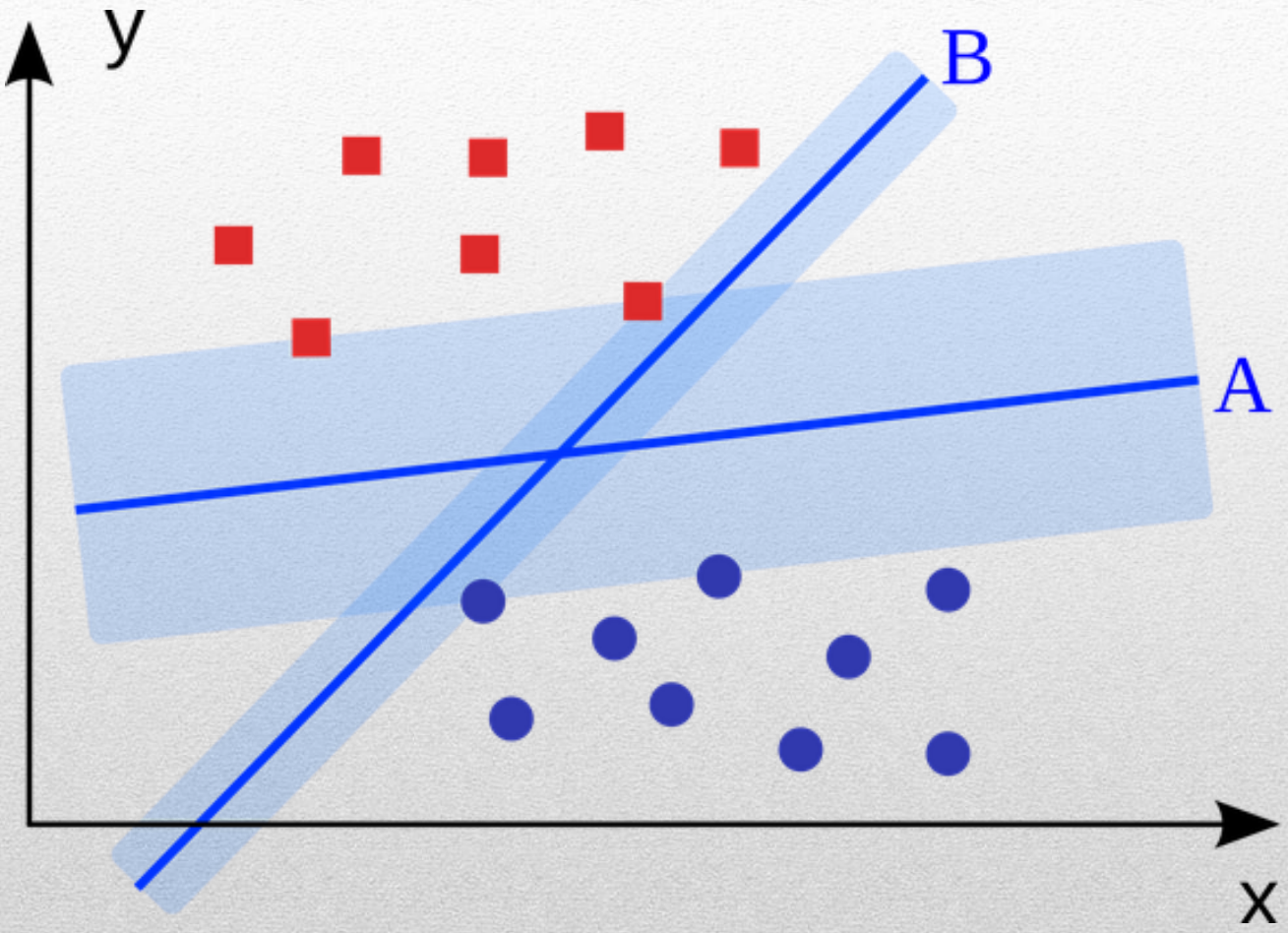
- Naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features. The featured image is the equation—with P(A|B) is posterior probability, P(B|A) is likelihood, P(A) is class prior probability, and P(B) is predictor prior probability.

**" P(A|B) = P(B|A)*P(A) / P(B) "**

- Some of real world examples are:
- To mark an email as spam or not spam
- Classify a news article about technology, politics, or sports
- Check a piece of text expressing positive emotions, or negative emotions?
- Used for face recognition software.

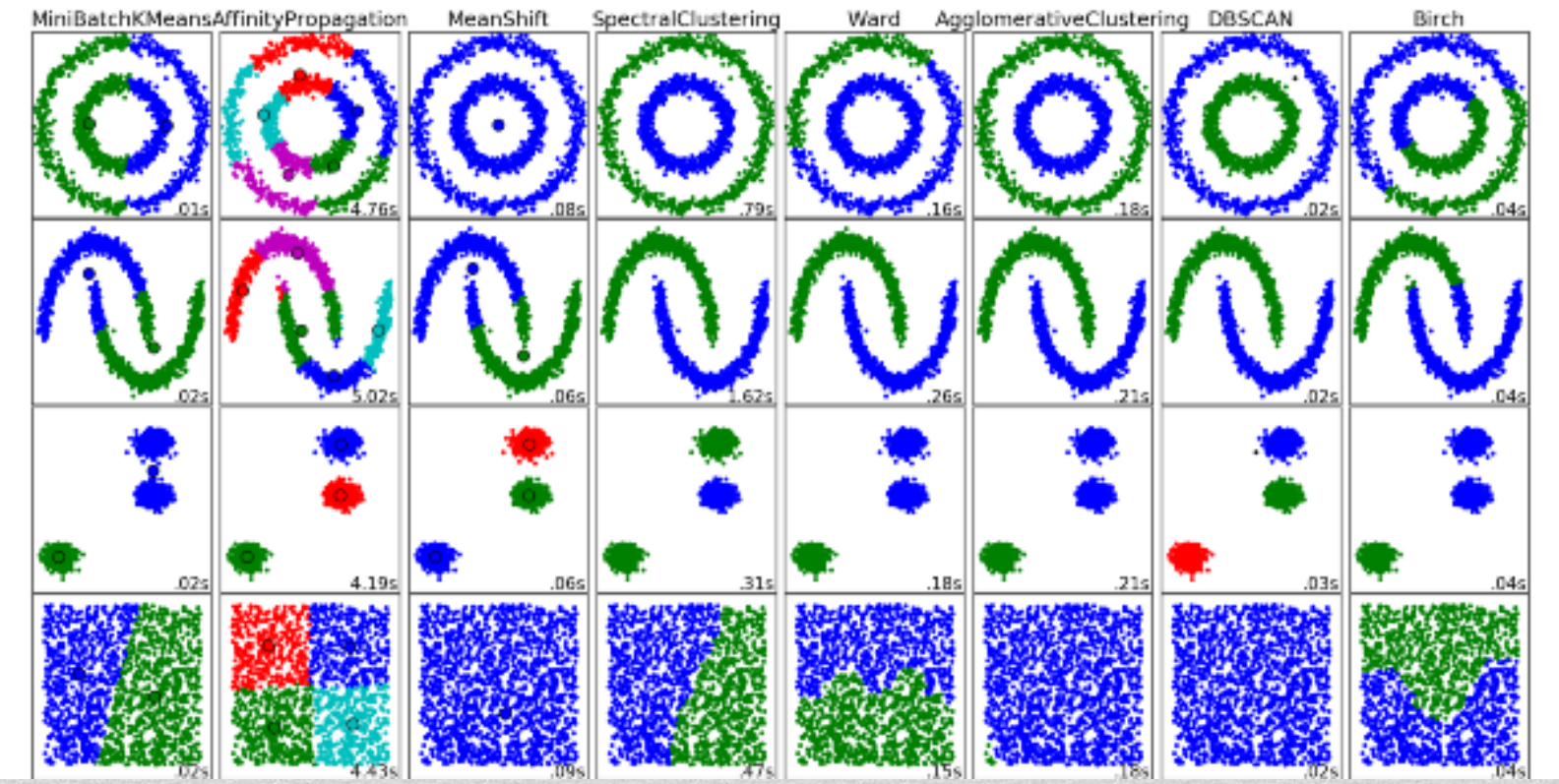
## 3. SUPPORT VECTOR MACHINES

- SVM is binary classification algorithm. Given a set of points of 2 types in N dimensional place, SVM generates a (N—1) dimensional hyperplane to separate those points into 2 groups.
- Say you have some points of 2 types in a paper which are linearly separable. SVM will find a straight line which separates those points into 2 types and situated as far as possible from all those points.
- In terms of scale, some of the biggest problems that have been solved using SVMs (with suitably modified implementations) are display advertising, human splice site recognition, image-based gender detection, large-scale image classification.


y
B
A
x

## 4. CLUSTERING ALGORITHM

- Clustering is the task of grouping a set of objects such that objects in the same group (*cluster*) are more similar to each other than to those in other groups.
- Every clustering algorithm is different, and here are a couple of them:
- Centroid-based algorithms
- Connectivity-based algorithms
- Density-based algorithms
- Probabilistic
- Dimensionality Reduction
- Neural networks / Deep Learning


MiniBatchKMeans
AffinityPropagation
MeanShift
SpectralClustering
Ward
AgglomerativeClustering
DBSCAN
Birch
.01s
4.76s
.08s
.79s
.16s
.18s
.02s
.04s
.02s
5.02s
.06s
1.62s
.26s
.21s
.02s
.04s
.02s
4.19s
.06s
.31s
.18s
.21s
.03s
.04s
.02s
4.43s
.09s
.47s
.15s
.18s
.02s
.04s

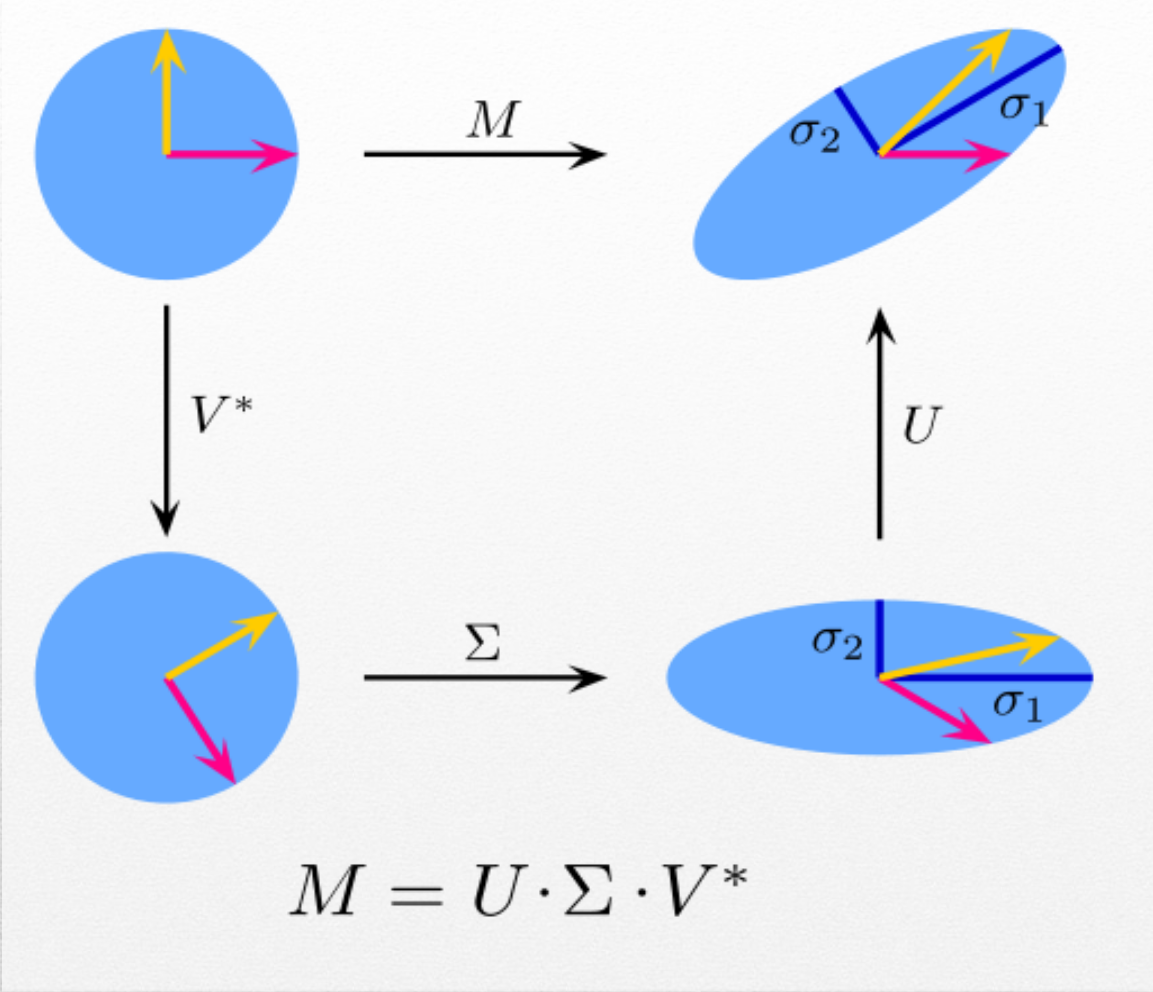
## 5. SINGULAR VALUE DECOMPOSITION

- In linear algebra, SVD is a factorization of a real complex matrix. For a given *m* * *n* matrix M, there exists a decomposition such that

**"M = UΣV"**

where U and V are unitary matrices and Σ is a diagonal matrix.

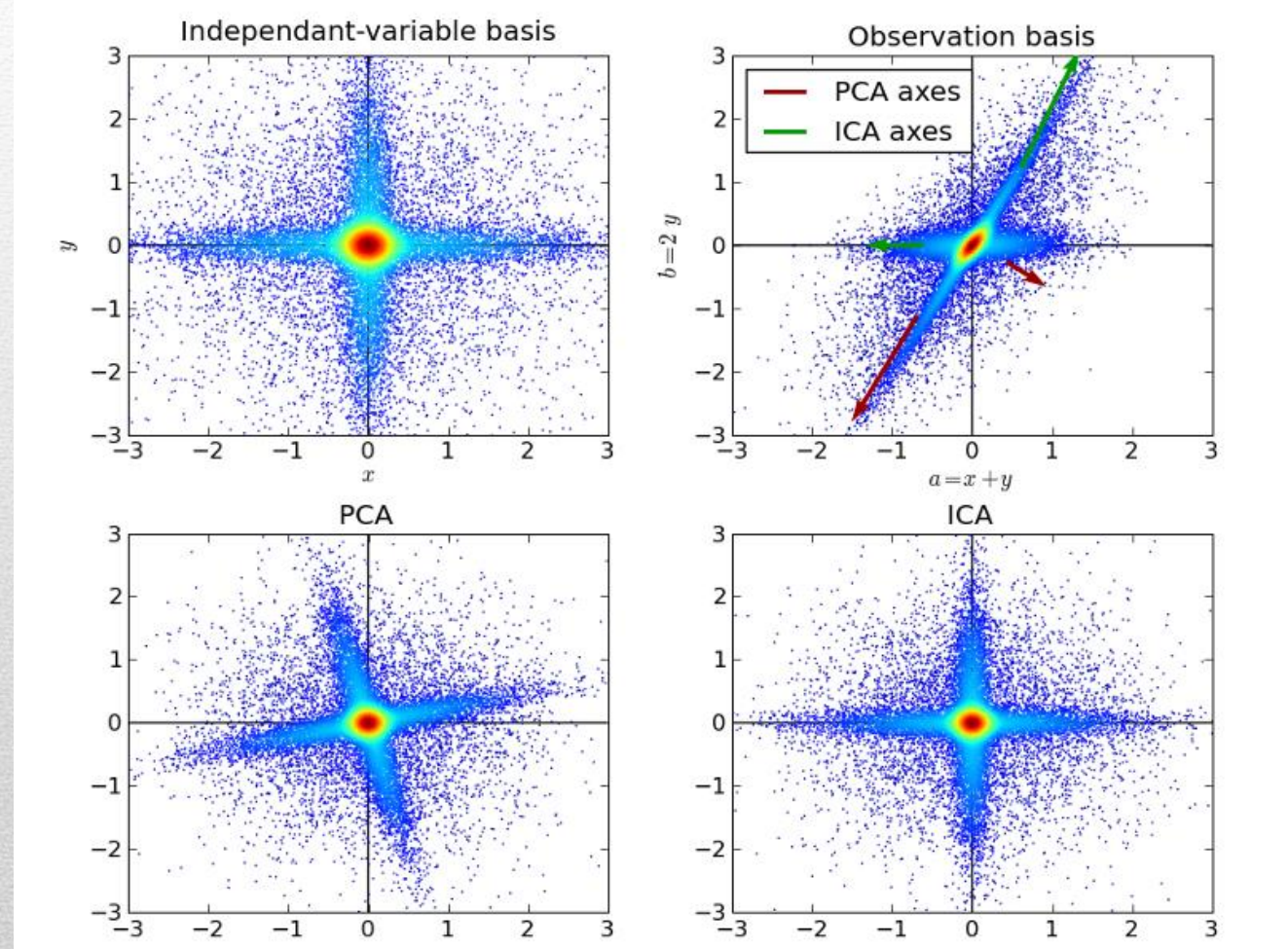
- PCA is actually a simple application of SVD.
- In computer vision, the 1st face recognition algorithms used PCA and SVD in order to represent faces as a linear combination of "eigenfaces", do dimensionality reduction, and then match faces to identities via simple methods; although modern methods are much more sophisticated, many still depend on similar techniques.


M
σ2
σ1
V*
U
Σ
σ2
σ1
M = U · Σ · V*

## 6. INDEPENDENT COMPONENT ANALYSIS

- ICA is a statistical technique for revealing hidden factors that underlie sets of random variables, measurements, or signals.
- ICA defines a generative model for the observed multivariate data, which is typically given as a large database of samples.
- In the model, the data variables are assumed to be linear mixtures of some unknown latent variables, and the mixing system is also unknown.
- The latent variables are assumed non-gaussian and mutually independent, and they are called independent components of the observed data.
- ICA is related to PCA, but it is a much more powerful technique that is capable of finding the underlying factors of sources when these classic methods fail completely.

- Its applications include digital images, document databases, economic indicators and psychometric measurements



# DEEP LEARNING CONCEPTS (THEORY)

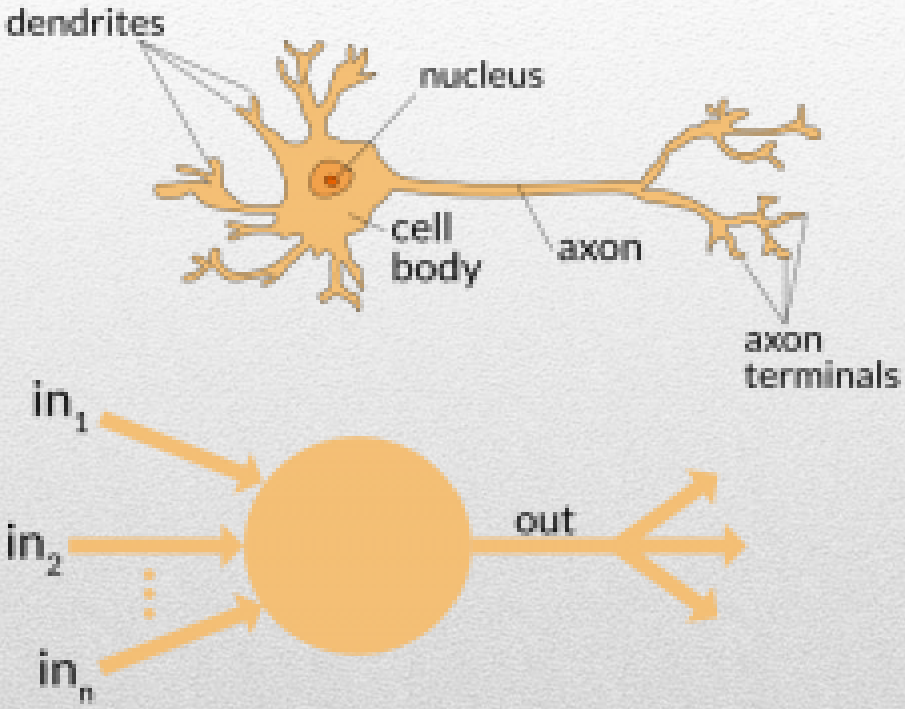
- Basics of Neural Networks
- Common Activation Functions
- Convolutional Neural Networks
- Recurrent Neural Networks

# BASICS OF NEURAL NETWORKS

## 1. NEURON

- Just like a neuron forms the basic element of our brain, a neuron forms the basic structure of a neural network.
- Just think of what we do when we get new information. When we get the information, we process it and then we generate an output.
- Similarly, in case of a neural network, a neuron receives an input, processes it and generates an output which is either sent to other neurons for further processing or it is the final output.

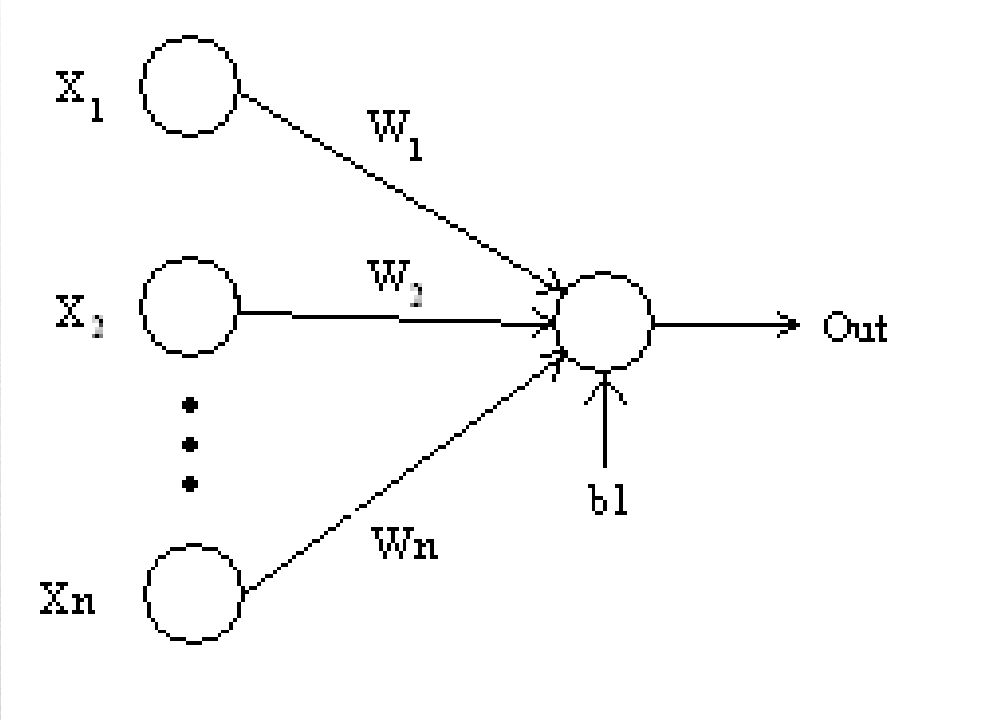
## NEURON


dendrites
nucleus
cell body
axon
axon terminals
$in_1$
$in_2$
$in_n$
out

## 2. WEIGHTS

- When input enters the neuron, it is multiplied by a weight.
- For example, if a neuron has two inputs, then each input will have has an associated weight assigned to it.
- We initialize the weights randomly and these weights are updated during the model training process.
- The neural network after training assigns a higher weight to the input it considers more important as compared to the ones which are considered less important. A weight of zero denotes that the particular feature is insignificant.
- Let's assume the input to be a, and the weight associated to be W1. Then after passing through the node the input becomes a*W1

## WEIGHTS



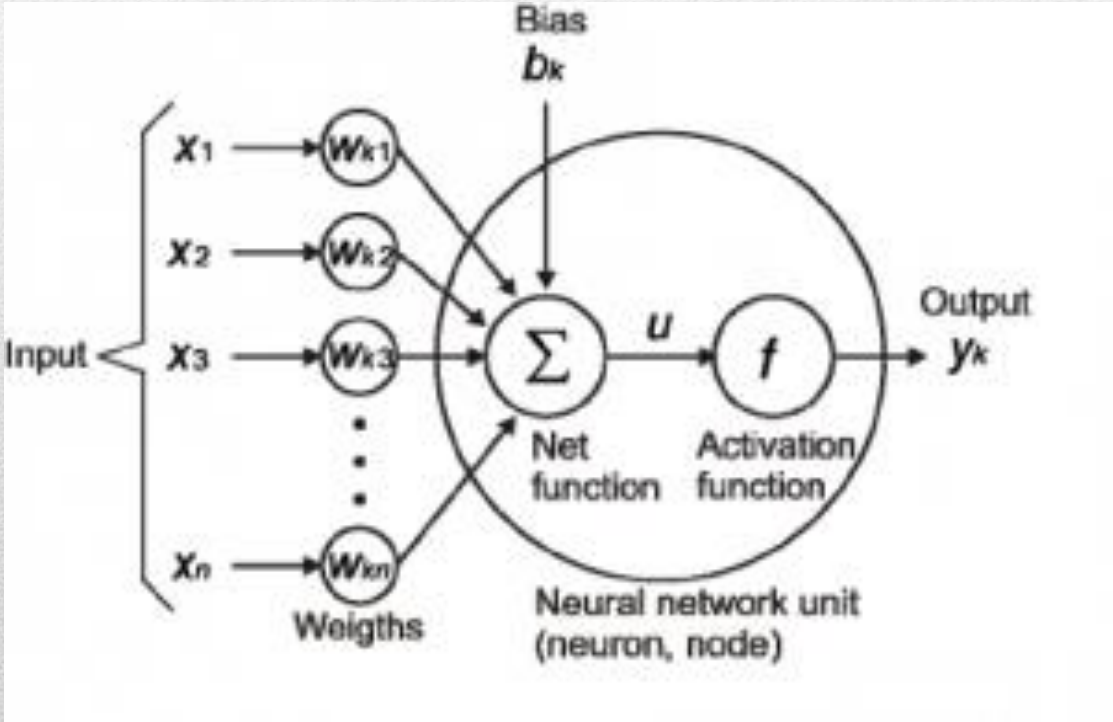
## 3. BIAS

- In addition to the weights, another linear component is applied to the input, called as the bias.
- It is added to the result of weight multiplication to the input.
- The bias is basically added to change the range of the weight multiplied input.
- After adding the bias, the result would look like a*W1+bias.
- This is the final linear component of the input transformation.

## 4. ACTIVATION FUNCTION

- Once the linear component is applied to the input, a non-linear function is applied to it.
- This is done by applying the activation function to the linear combination.
- The activation function translates the input signals to output signals.
- The output after application of the activation function would look something like **f(a*W1+b)** where f() is the activation function.
- In the below diagram we have “n” inputs given as X1 to Xn and corresponding weights Wk1 to Wkn.
- We have a bias given as bk.

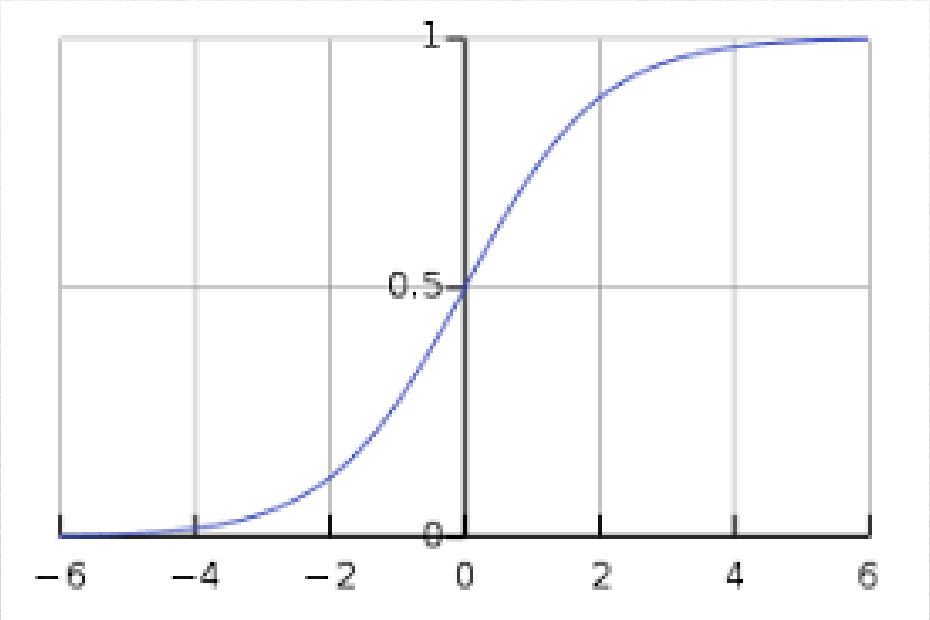
- We have a bias given as bk. The weights are first multiplied to its corresponding input and are then added together along with the bias. Let this be called as u. " **u=∑w*x+b** "
- The activation function is applied to u i.e. f(u) and we receive the final output from the neuron as **yk = f(u)**



# COMMONLY APPLIED ACTIVATION FUNCTIONS

## 5. SIGMOID

- One of the most common activation functions used is Sigmoid. It is defined as: **"sigmoid(x) = $1/(1+e^{-x})$"**

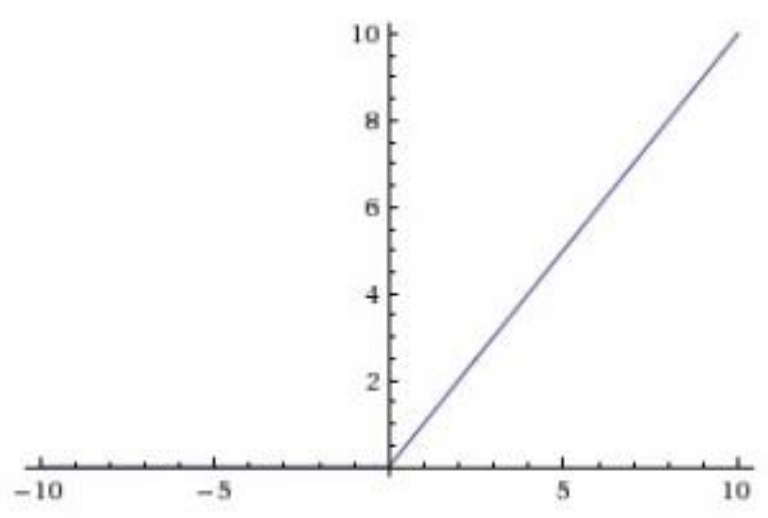


## 6. RELU (RECTIFIED LINEAR UNITS)

- Instead of sigmoids, the recent networks prefer using ReLu activation functions for the hidden layers. The function is defined as:

**"f(x) = max(x,0)"**

- The output of the function is X when X>0 and 0 for X<=0.




- The major benefit of using ReLU is that it has a constant derivative value for all inputs greater than 0. The constant derivative value helps the network to train faster.

## 7. SOFTMAX

- Softmax activation functions are normally used in the output layer for classification problems.
- It is similar to the sigmoid function, with the only difference being that the outputs are normalized to sum up to 1.
- The sigmoid function would work in case we have a binary output, however in case we have a multiclass classification problem, softmax makes it really easy to assign values to each class which can be easily interpreted as probabilities.
- It's very easy to see it this way – Suppose you're trying to identify a 6 which might also look a bit like 8.

- The function would assign values to each number as below. We can easily see that the highest probability is assigned to 6, with the next highest assigned to 8 and so on…

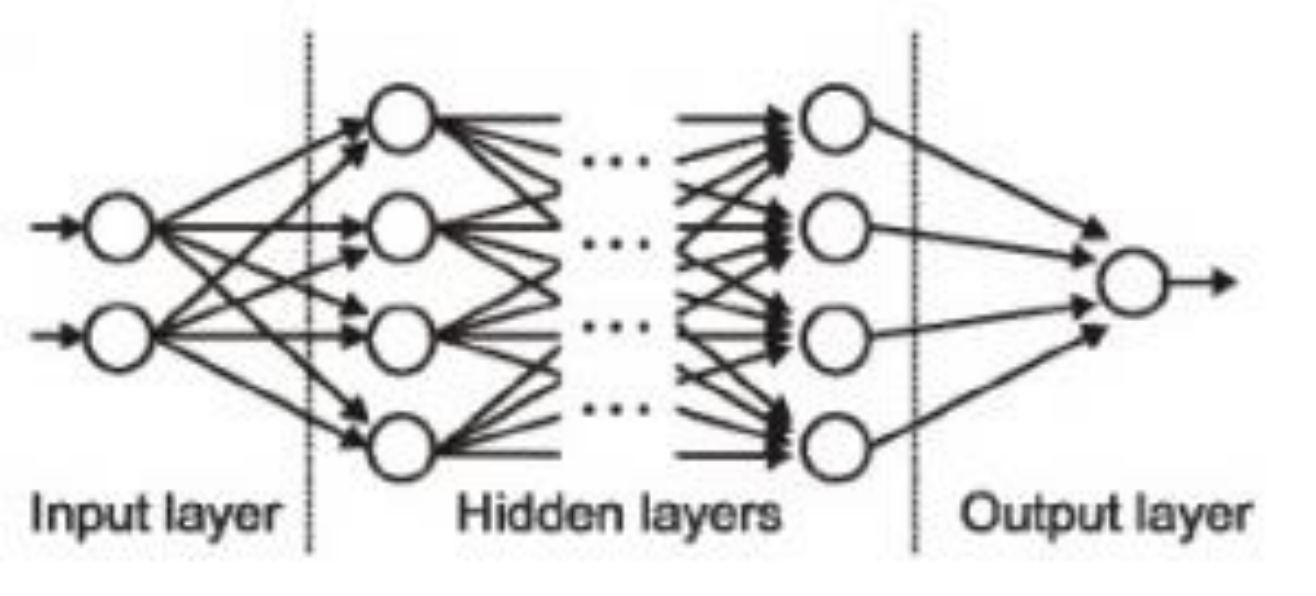
## 8. NEURAL NETWORKS

- Neural Networks form the backbone of deep learning. The goal of a neural network is to find an approximation of an unknown function.
- It is formed by interconnected neurons.
- These neurons have weights, and bias which is updated during the network training depending upon the error.
- The activation function puts a nonlinear transformation to the linear combination which then generates the output.
- The combinations of the activated neurons give the output.

## 9. INPUT / OUTPUT / HIDDEN LAYERS

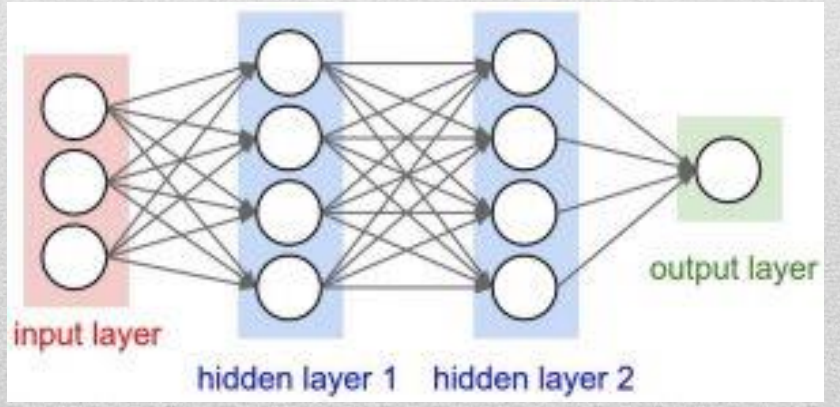
- Simply as the name suggests the input layer is the one which receives the input and is essentially the first layer of the network.
- The output layer is the one which generates the output or is the final layer of the network.
- The processing layers are the hidden layers within the network.
- These hidden layers are the ones which perform specific tasks on the incoming data and pass on the output generated by them to the next layer.
- The input and output layers are the ones visible to us, while are the intermediate layers are hidden.

- The input and output layers are the ones visible to us, while are the intermediate layers are hidden.



## 10. MLP (MULTI LAYER PERCEPTION)

- A single neuron would not be able to perform highly complex tasks.
- Therefore, we use stacks of neurons to generate the desired outputs.
- In the simplest network we would have an input layer, a hidden layer and an output layer.
- Each layer has multiple neurons and all the neurons in each layer are connected to all the neurons in the next layer.
- These networks can also be called as fully connected networks.



## 11. FORWARD PROPAGATION

- Forward Propagation refers to the movement of the input through the hidden layers to the output layers.
- In forward propagation, the information travels in a single direction FORWARD.
- The input layer supplies the input to the hidden layers and then the output is generated.
- There is no backward movement.

## 12. COST FUNCTION

- When we build a network, the network tries to predict the output as close as possible to the actual value.
- We measure this accuracy of the network using the cost/loss function.
- The cost or loss function tries to penalize the network when it makes errors.
- Our objective while running the network is to increase our prediction accuracy and to reduce the error, hence minimizing the cost function.

- The most optimized output is the one with least value of the cost or loss function.
- If I define the cost function to be the mean squared error, it can be written as

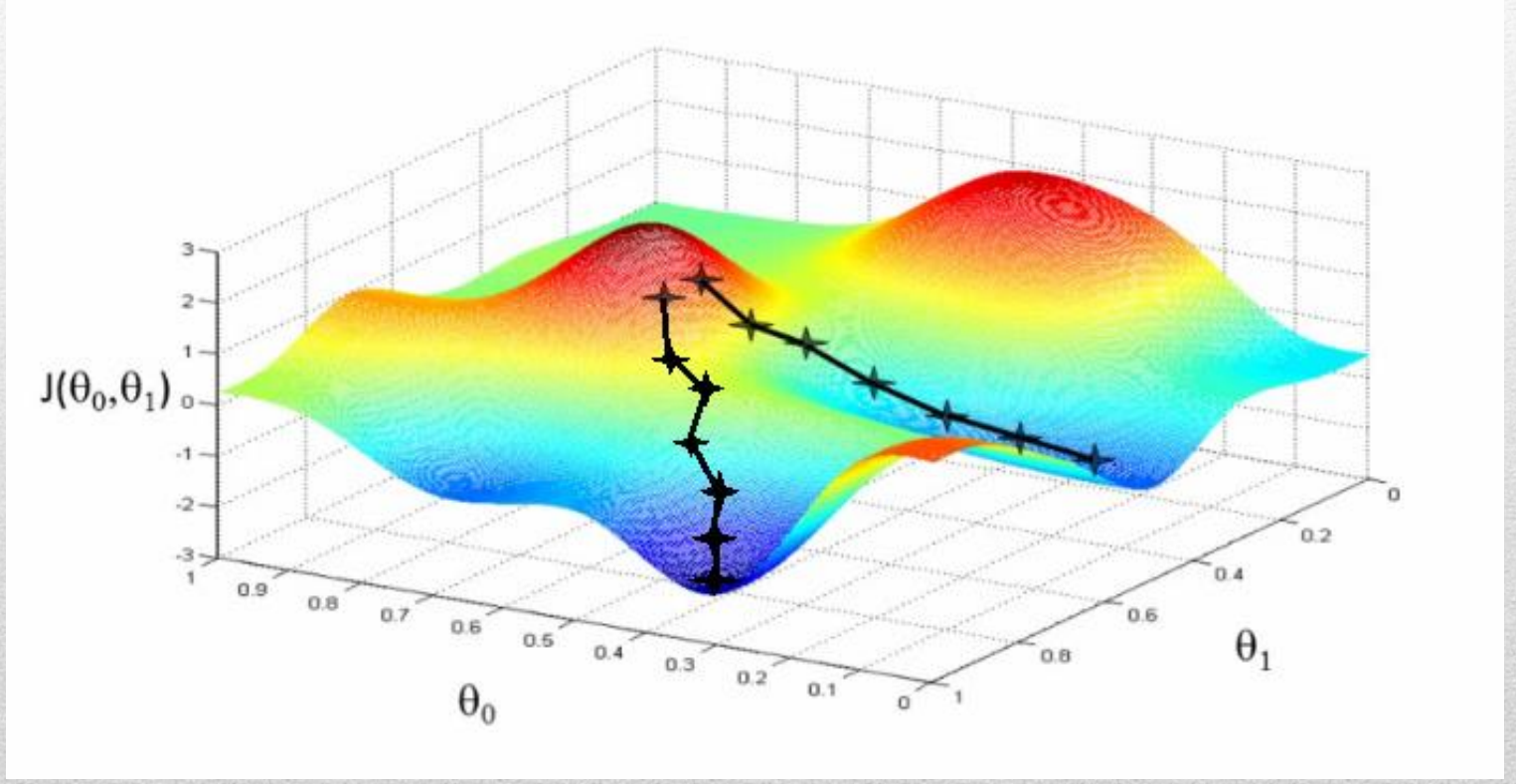
$$\text{“ } C = 1/m \sum (y - a)^2 \text{ “}$$

- where m is the number of training inputs, a is the predicted value and y is the actual value of that particular example.
- The learning process revolves around minimizing the cost.

## 13. GRADIENT DESCENT

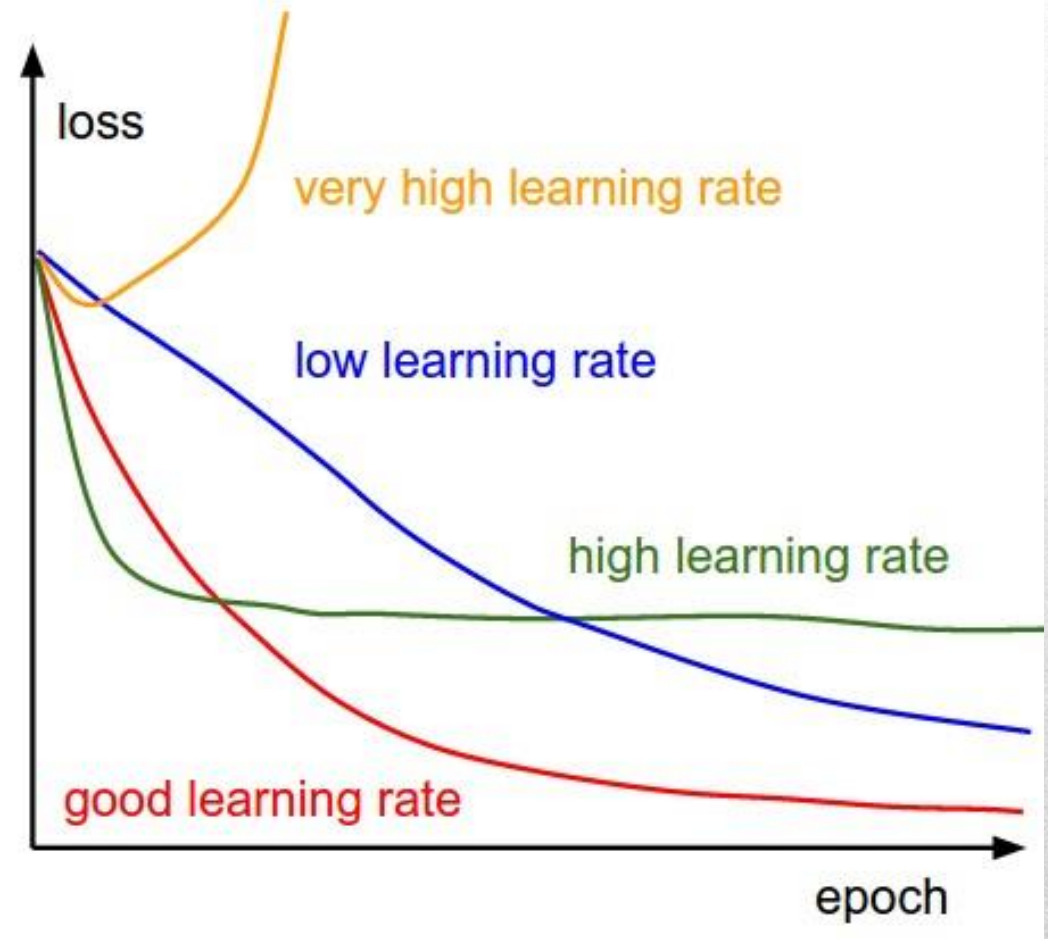
- Gradient descent is an optimization algorithm for minimizing the cost.
- To think of it intuitively, while climbing down a hill you should take small steps and walk down instead of just jumping down at once.
- Therefore, what we do is, if we start from a point x, we move down a little i.e. delta h, and update our position to x-delta h and we keep doing the same till we reach the bottom.
- Consider bottom to be the minimum cost point.

- Mathematically, to find the local minimum of a function one takes steps proportional to the negative of the gradient of the function.



## 14. LEARNING RATE

- The learning rate is defined as the amount of minimization in the cost function in each iteration.
- In simple terms, the rate at which we descend towards the minima of the cost function is the learning rate.
- We should choose the learning rate very carefully since it should neither be very large that the optimal solution is missed and nor should be very low that it takes forever for the network to converge.


loss
very high learning rate
low learning rate
high learning rate
good learning rate
epoch

## 15. BACKPROPAGATION

- When we define a neural network, we assign random weights and bias values to our nodes.
- Once we have received the output for a single iteration, we can calculate the error of the network.
- This error is then fed back to the network along with the gradient of the cost function to update the weights of the network.
- These weights are then updated so that the errors in the subsequent iterations is reduced.
- This updating of weights using the gradient of the cost function is known as back-propagation.

## 16. BATCHES

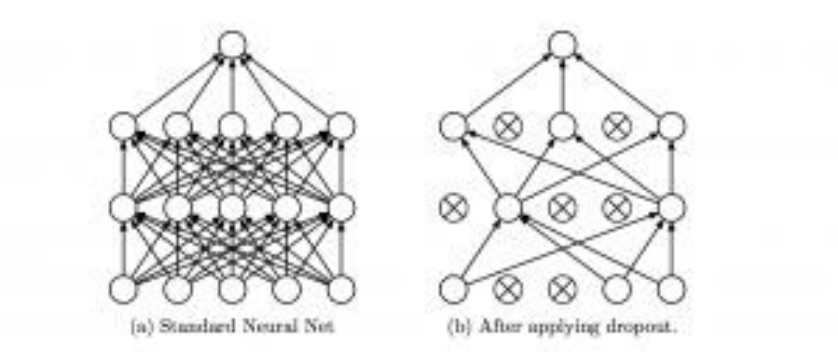
- While training a neural network, instead of sending the entire input in one go, we divide in input into several chunks of equal size randomly.
- Training the data on batches makes the model more generalized as compared to the model built when the entire data set is fed to the network in one go.

## 17. EPOCHS

- An epoch is defined as a single training iteration of all batches in both forward and back propagation.
- This means 1 epoch is a single forward and backward pass of the entire input data.

## 18. DROPOUTS

- Dropout is a regularization technique which prevents over-fitting of the network.
- As the name suggests, during training a certain number of neurons in the hidden layer is randomly dropped.
- This means that the training happens on several architectures of the neural network on different combinations of the neurons.
- You can think of drop out as an ensemble technique, where the output of multiple networks is then used to produce the final output.



## 19. BATCH NORMALIZATION

- As a concept, batch normalization can be considered as a dam we have set as specific checkpoints in a river.
- This is done to ensure that distribution of data is the same as the next layer hoped to get.
- When we are training the neural network, the weights are changed after each step of gradient descent.
- This changes the how the shape of data is sent to the next layer.
- But the next layer was expecting the distribution similar to what it had previously seen.
- So we explicitly normalize the data before sending it to the next layer.


"Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," Ioffe and Szegedy 2015

# CONVOLUTIONAL NEURAL NETWORKS

## 20. FILTERS

- A filter in a CNN is like a weight matrix with which we multiply a part of the input image to generate a convoluted output.
- Let's assume we have an image of size 28*28.
- We randomly assign a filter of size 3*3, which is then multiplied with different 3*3 sections of the image to form what is known as a convoluted output.
- The filter size is generally smaller than the original image size.
- The filter values are updated like weight values during backpropagation for cost minimization.

INPUT

| | | | | |
|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |

FILTER

| | | |
|---|---|---|
| 1 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |

CONVOLVED FEATURE

| | | |
|---|---|---|
| 4 | | |
| | | |
| | | |

| | | | | |
|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |

| | | |
|---|---|---|
| 1 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |

| | | |
|---|---|---|
| 4 | | |
| 2 | | |
| | | |

| | | | | |
|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |

| | | |
|---|---|---|
| 1 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |

| | | |
|---|---|---|
| 4 | 3 | |
| 2 | | |
| | | |

| | | | | |
|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |

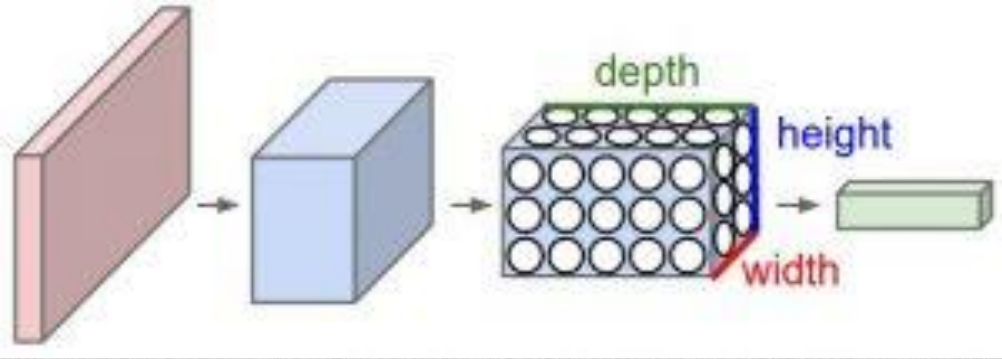
| | | |
|---|---|---|
| 1 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |

| | | |
|---|---|---|
| 4 | 3 | 4 |
| 2 | 4 | 3 |
| 2 | 3 | 4 |

## 21. CNN (CONVOLUTIONAL NEURAL NETWORKS)

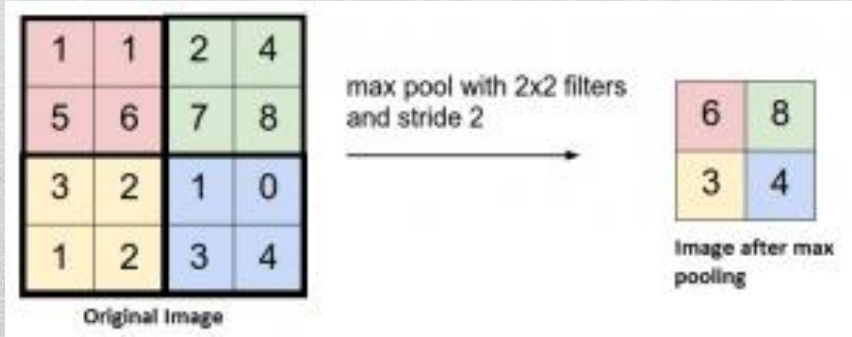
- Convolutional neural networks are basically applied on image data.
- Suppose we have an input of size (28*28*3), If we use a normal neural network, there would be 2352(28*28*3) parameters.
- And as the size of the image increases the number of parameters becomes very large.
- We “convolve” the images to reduce the number of parameters (as shown above in filter definition).

- As we slide the filter over the width and height of the input volume we will produce a **2-dimensional activation map** that gives the output of that filter at every position.
- We will stack these activation maps along the depth dimension and produce the output volume.



## 22. POOLING

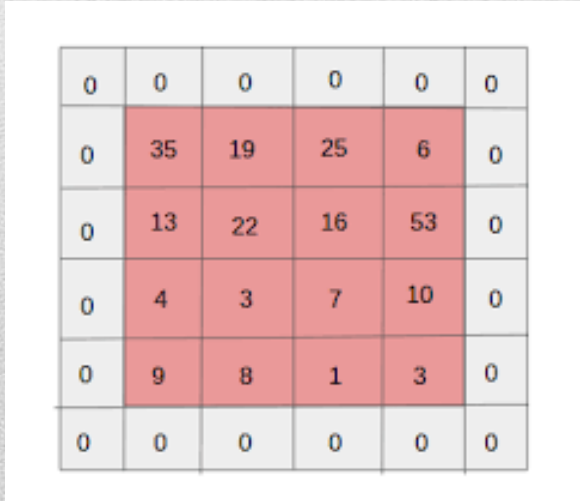
- It is common to periodically introduce pooling layers in between the convolution layers.
- This is basically done to reduce a number of parameters and prevent over-fitting.
- The most common type of pooling is a pooling layer of filter size(2,2) using the MAX operation.
- What it would do is, it would take the maximum of each 4*4 matrix of the original image.



## 23. PADDING

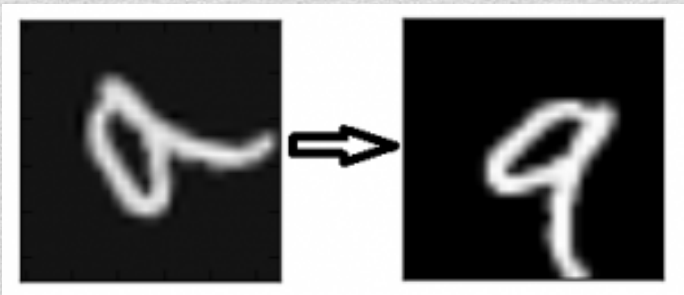
- Padding refers to adding extra layer of zeros across the images so that the output image has the same size as the input. This is known as same padding.
- Valid padding refers to keeping the image as such an having all the pixels of the image which are actual or “valid”.

.



## 24. DATA AUGMENTATION

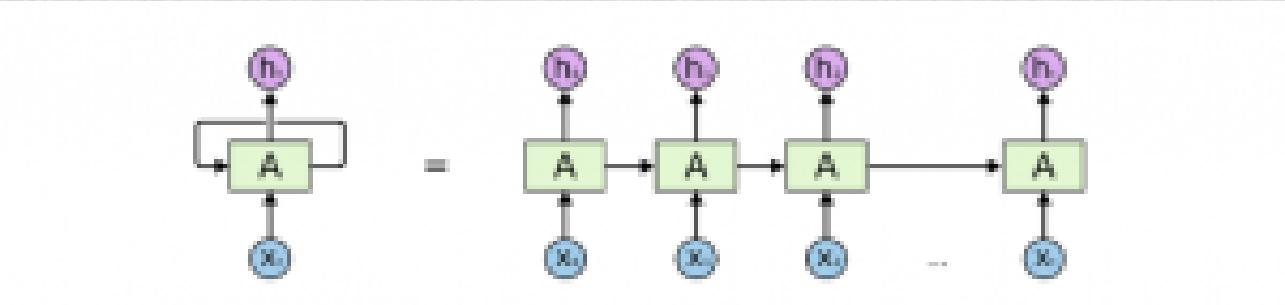
- Data Augmentation refers to the addition of new data derived from the given data, which might prove to be beneficial for prediction.
- For example, it might be easier to view the cat in a dark image if you brighten it, or for instance, a 9 in the digit recognition might be slightly tilted or rotated.
- In this case, rotation would solve the problem and increase the accuracy of our model.
- By rotating or brightening we're improving the quality of our data. This is known as Data augmentation.



# RECURRENT NEURAL NETWORK

## 25. RECURRENT NEURONS

- A recurrent neuron is one in which the output of the neuron is sent back to it for t time stamps.
- If you look at the diagram the output is sent back as input t times.
- The unrolled neuron looks like t different neurons connected together.
- The basic advantage of this neuron is that it gives a more generalized output.



## 26. RNN (RECURRENT NEURAL NETWORKS)

- Recurrent neural networks are used especially for sequential data where the previous output is used to predict the next one.
- In this case the networks have loops within them. The loops within the hidden neuron gives them the capability to store information about the previous words for some time to be able to predict the output.
- The output of the hidden layer is sent again to the hidden layer for t time stamps.

- The output of the recurrent neuron goes to the next layer only after completing all the time stamps.
- The output sent is more generalized and the previous information is retained for a longer period.
- The error is then back propagated according to the unfolded network to update the weights. This is known as **backpropagation through time(BPTT).**

## 27. VANISHING GRADIENT PROBLEM

- Vanishing gradient problem arises in cases where the gradient of the activation function is very small.
- During back propagation when the weights are multiplied with these low gradients, they tend to become very small and “vanish” as they go further deep in the network.
- This makes the neural network to forget the long range dependency.
- This generally becomes a problem in cases of recurrent neural networks where long term dependencies are very important for the network to remember.
- This can be solved by using activation functions like ReLu which do not have small gradients.

## 28. EXPLODING GRADIENT PROBLEM

- This is the exact opposite of the vanishing gradient problem, where the gradient of the activation function is too large.
- During back propagation, it makes the weight of a particular node very high with respect to the others rendering them insignificant.
- This can be easily solved by clipping the gradient so that it doesn't exceed a certain value.

# INTRODUCTION TO DEMO SESSION

- **DEMO SESSION 1**

Real Time Object Detection - Object Detection is the process of finding real-world object instances like car, bike, TV, flowers, and humans in still images or Videos.

- **DEMO SESSION 2**

Sentiment Analysis - The Sentiment Analysis is the name of the problem that with a sentence or text the machine gets capable to analyze and predict with the **maximum** precision possible the sentiment that will be obtained by a person when reads it or the contextual opinion related to something

- **ENVIRONMENT** – Tensorflow using the Jupyter Notebook and Anaconda

# SESSION 2

## ( DEMO )

## GET SET GO

# SOFTWARE INSTALLATION

- Step 1 – Install the Anaconda “anaconda.exe”
- Step 2 – Run the Anaconda prompt and locate anaconda using command.

  **“ where anaconda ”**
- Step 3 – Set working directory using command

  **“ cd C:\Users\Admin\Anaconda3 ”**
- Step 4 - Create the yml file to install dependencies using command

  **“ echo.>tensorflowproj.yml ”**

- Step 5 – Edit the yml file using command

  **“ notepad tensorflowproj.yml ”**

- Step 6 – Insert the following in the yml file and save it

  **name: tensorflowproj**

  **dependencies:**

  **- python=3.6**

  **- jupyter**

  **- ipython**

  **- pandas**

  **- seaborn**

- Step 7 – Compile the yml file using command

  **“conda env create –f tensorflowproj.yml”**

- Step 8 – Ensure that the environment is created using command

  **“ conda env list ”**

- Step 9 – Activate the environment using command

  **“ activate tensorflowproj ”**

- Step 10 – Check that all the dependencies are installed using command.

  **“ which python ”**

  **“ which jupyter ”**

  **“ which ipython ”**

- Step 11 – Install Tensorflow using command

  **“ conda install tensorflow ”**

- Step 12 – Activate the TensorFlow environment again using command

  **“ activate tensorflowproj ”**

- Step 13 – Open the Jupyter Notebook IDE using command

  **“ Jupyter Notebook ”**

- Step 14 – To close the Jupyter Notebook workspace use key combination

  **“ CTRL + C “**

# ENVIRONMENTAL INTRODUCTION

- Anaconda
- Python 3.4 or higher
- Jupyter
- IPython
- NumPy
- Pandas
- Keras
- OpenCV
- Matplotlib
- Tensorflow

## ANACONDA

- **Anaconda** is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. Package versions are managed by the package management system *conda*.

## PYTHON

- **Python language** is one of the most flexible languages and can be used for various purposes. Python has gained huge popularity base of this. Python does contain special libraries for machine learning namely SciPy and NumPy which great for linear algebra and getting to know kernel methods of machine learning. The language is great to use when working with machine learning algorithms and has easy syntax relatively. For beginners, this is the best language to use and to start with.

## JUPYTER / IPYTHON

- **Jupyter** or **IPython Notebook** is a web application that allows you to run live code, embed visualizations and explanatory text all in one place.

## NUMPY / PANDAS

- **NumPy and Pandas** working together, Pandas depends upon and interoperates with NumPy, the Python library for fast numeric array computations. For example, you can use the DataFrame attribute. values to represent a DataFrame df as a NumPy array. You can also pass pandas data structures to NumPy methods.

## KERAS

- **Keras** is a high-level neural networks API, written in Python and capable on running on top of Tensorflow, CNTK or Theano. It was developed with a focus on enabling fast experimentation. Allows for easy and fast prototyping, Supports both CNN and RCNN**,** Runs seamlessly on CPU and GPU.

## MATPLOTLIB

- **Matplotlib** is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+. There is also a procedural “pylab” interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged. SciPy makes use of Matplotlib.

## OPENCV

- **OpenCV (Open Source Computer Vision)** is released under BSD license, OpenCV was designed for computational efficiency and with a strong focus on real-time applications, Written in optimized C/C++, the library can take advantage of multi-core processing. Enabled with OpenCL, it can take advantage of hardware acceleration of the underlying heterogeneous compute platform.

## TENSORFLOW

- Tensorflow is Google's Open Source Machine Learning Framework for dataflow programming across a range of tasks.
- Nodes in the graph represent mathematical operations, while the graph edges represent the multi-dimensional data arrays (**tensors**) communicated between them.
- Tensors are just multidimensional arrays, an extension of 2-dimensional tables to data with a higher dimension. There are many features of Tensorflow which makes it appropriate for Deep Learning.
- So, without wasting any time, let's see how we can implement Object Detection using Tensorflow.

# DEMO 1 ON OBJECT DETECTION APPLICATION

# QUERIES ???

( THEORY | DEMO )

# FEEDBACK...

## CONTACT DETAILS


| | | |
|---|---|---|
| ABISHEK HARI | ABISHEKHARI007@GMAIL.COM | 9600984034 |
| PRADEEISH PRABHAKARAN | PRADEEISHPRABHA@GMAIL.COM | 7402076993 |


FEEL FREE TO CONTACT US @ ANYTIME

# THANK YOU ALL...

BEST WISHES FROM

CSI STUDENT CHAPTER | KSRCT