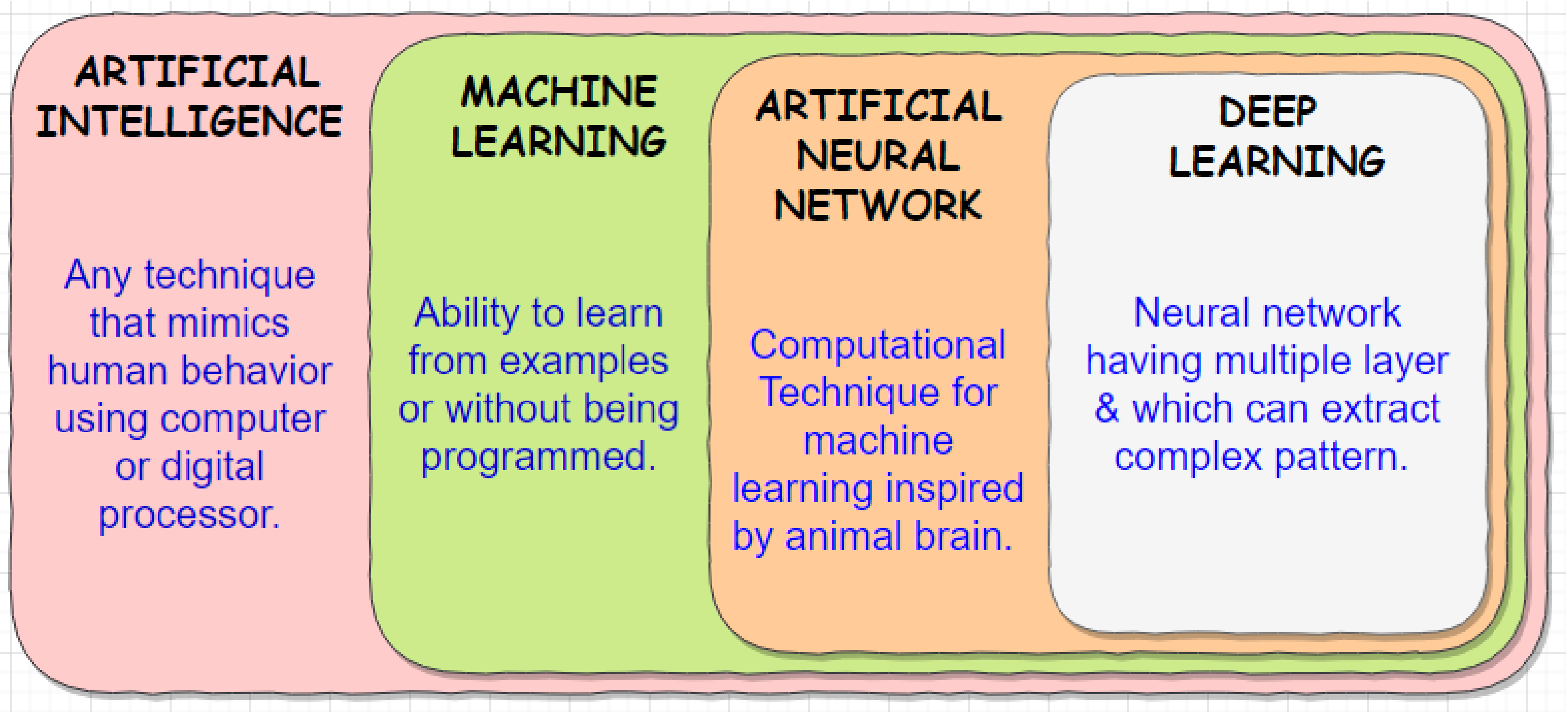
Unit-I:

Introduction of Deep learning, Neural Network, Feed Forward Neural Network, Back Forward Neural Network, the Backpropagation algorithm. Activation Function: Threshold, Sigmoid, Rectifier(ReLU), Hyperbolic Tangent (tanh), Gradient Descent, Stochastic Gradient Descent, Cost Function, Global minima and Local minima.

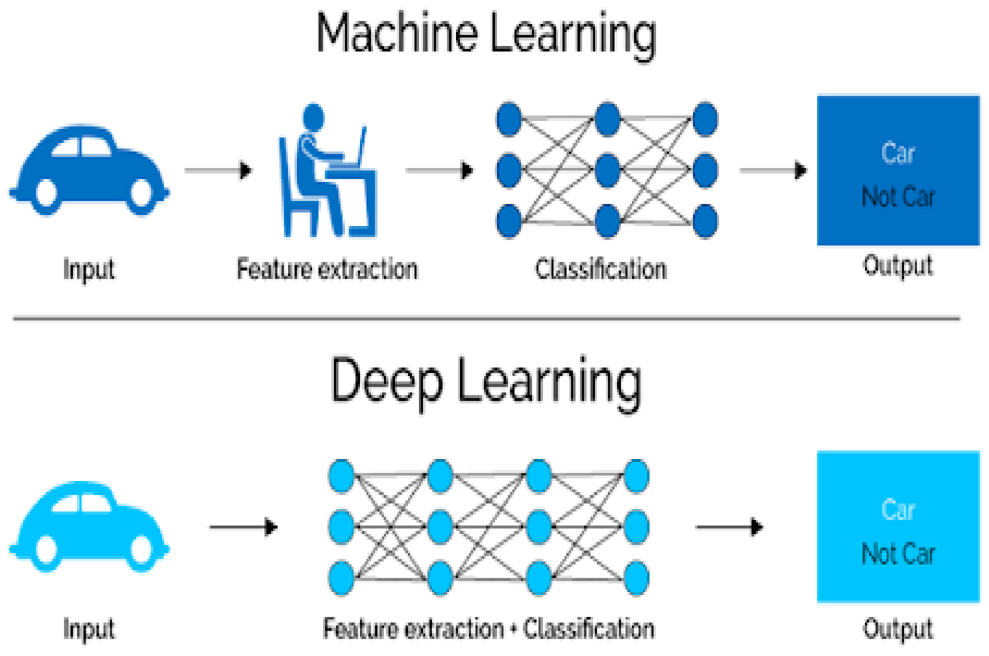

ARTIFICIAL INTELLIGENCE
Any technique that mimics human behavior using computer or digital processor.
MACHINE LEARNING
Ability to learn from examples or without being programmed.
ARTIFICIAL NEURAL NETWORK
Computational Technique for machine learning inspired by animal brain.
DEEP LEARNING
Neural network having multiple layer & which can extract complex pattern.

Introduction of Deep learning

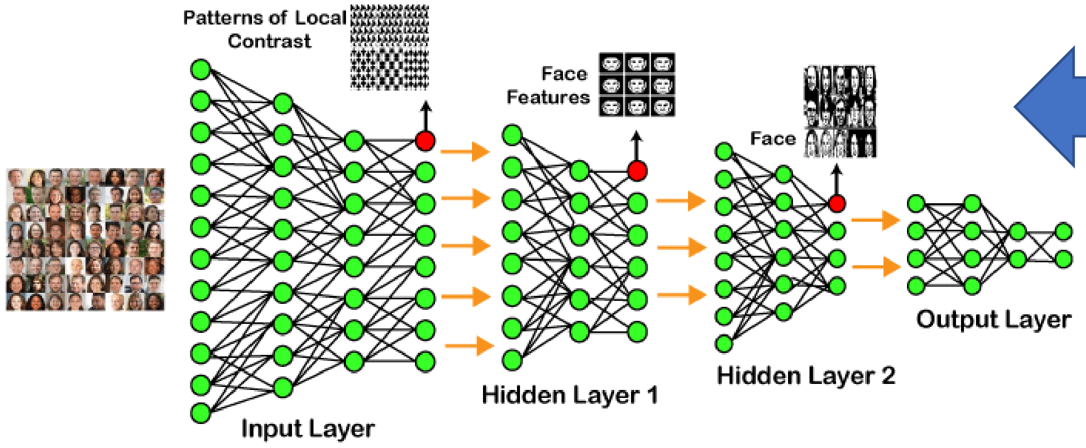
What is deep learning?

Deep learning is a type of [machine learning] and artificial intelligence ([AI]) that imitates the way humans gain certain types of knowledge. Deep learning models can be taught to perform classification tasks and recognize patterns in photos, text, audio and other various data. It is also used to automate tasks that would normally need human intelligence, such as describing images or transcribing audio files.

For example, in an image recognition task, the algorithm might learn to associate certain features in an image (such as the shape of an object or the color of an object) with the correct label (such as "dog" or "cat").



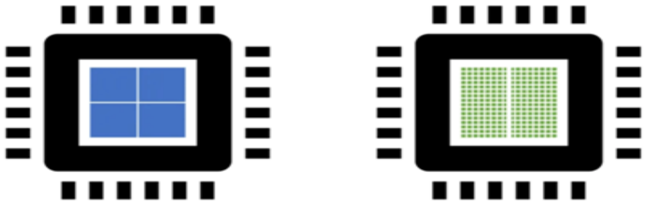
Example of Deep Learning



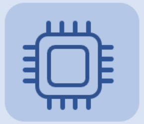
In the example, we provide the raw data of images to the first layer of the input layer. After then, these input layer will determine the patterns of local contrast that means it will differentiate on the basis of colors, luminosity, etc. Then the 1st hidden layer will determine the face feature, i.e., it will fixate on eyes, nose, and lips, etc. And then, it will fixate those face features on the correct face template. So, in the 2^{nd} hidden layer, it will actually determine the correct face here as it can be seen in the above image, after which it will be sent to the output layer. Likewise, more hidden layers can be added to solve more complex problems, for example, if you want to find out a particular kind of face having large or light complexions. So, as and when the hidden layers increase, we are able to solve complex problems.

Why deep learning is becoming so popular?

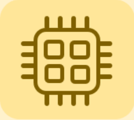
1. Data Growth
2. Hardware advancements
   GPU and TPU
3. Python & Open source Ecosystem
4. Cloud & AI Boom



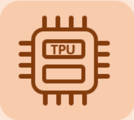
CPU	GPU
Central Processing Unit	Graphics Processing Unit
4-8 Cores	100s or 1000s of Cores
Low Latency	High Throughput
Good for Serial Processing	Good for Parallel Processing
Quickly Process Tasks That Require Interactivity	Breaks Jobs Into Separate Tasks To Process Simultaneously
Traditional Programming Are Written For CPU Sequential Execution	Requires Additional Software To Convert CPU Functions to GPU Functions for Parallel Execution



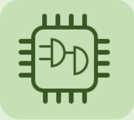
CPU
- Small models
- Small datasets
- Useful for design space exploration



GPU
- Medium-to-large models, datasets
- Image, video processing
- Application on CUDA or OpenCL



TPU
- Matrix computations
- Dense vector processing
- No custom TensorFlow operations



FPGA
- Large datasets, models
- Compute intensive applications
- High performance, high perf./cost ratio

What is neural network in simple words?

A neural network is a method in artificial intelligence that teaches computers to process data in a way that is inspired by the human brain. It is a type of machine learning process, called deep learning, that uses interconnected nodes or neurons in a layered structure that resembles the human brain.

Neural networks are composed of layers of interconnected neurons, and they learn by adjusting the weights of the connections between neurons. Simply speaking, they are a type of computer system that are designed to mimic the workings of the human brain. They are similar to other machine learning algorithms, but they are composed of a large number of interconnected processing nodes, or neurons, that can learn to recognize patterns of input data.

A **Brief History of Neural Networks**

Neural networks date back to the early 1940s when mathematicians Warren McCulloch and Walter Pitts built a simple algorithm-based system designed to emulate human brain function.

Work in the field accelerated in 1957 when Cornell University's Frank Rosenblatt conceived of the *perceptron*, the groundbreaking algorithm developed to perform complex recognition tasks.

During the four decades that followed, the lack of computing power necessary to process large amounts of data put the brakes on advances. In the 2000s, thanks to the advent of greater computing power and more sophisticated hardware, as well as to the existence of vast data sets to draw from, computer scientists finally had what they needed, and neural networks and AI took off, with no end in sight.

To understand how much the field has expanded in the new millennium, consider that ninety percent of internet data has been created since 2016. That pace will continue to accelerate, thanks to the growth of the Internet of Things (IoT).

Why Do We Use Neural Networks?

Neural networks' human-like attributes and ability to complete tasks in infinite permutations and combinations make them uniquely suited to today's big data-based applications. Because neural networks also have the unique capacity (known as *fuzzy logic*) to make sense of ambiguous, contradictory, or incomplete data, they are able to use controlled processes when no exact models are available.

According to a report published by Statista, in 2017, global data volumes reached close to 100,000 petabytes (i.e., one million gigabytes) per month; they are forecasted to reach 232,655 petabytes by 2021. With businesses, individuals, and devices generating vast amounts of information, all of that big data is valuable, and neural networks can make sense of it.

Attributes of Neural Networks

With the human-like ability to problem-solve — and apply that skill to huge datasets — neural networks possess the following powerful attributes:

- **Adaptive Learning**: Like humans, neural networks model non-linear and complex relationships and build on previous knowledge. For example, software uses adaptive learning to teach math and language arts.

- **Self-Organization**: The ability to cluster and classify vast amounts of data makes neural networks uniquely suited for organizing the complicated visual problems posed by medical image analysis.

- **Real-Time Operation**: Neural networks can (sometimes) provide real-time answers, as is the case with self-driving cars and drone navigation.

- **Prognosis**: NN’s ability to predict based on models has a wide range of applications, including for weather and traffic.

- **Fault Tolerance**: When significant parts of a network are lost or missing, neural networks can fill in the blanks. This ability is especially useful in space exploration, where the failure of electronic devices is always a possibility.

Tasks Neural Networks Perform

Neural networks are highly valuable because they can carry out tasks to make sense of data while retaining all their other attributes. Here are the critical tasks that neural networks perform:

- **Classification**: NNs organize patterns or datasets into predefined classes.
- **Prediction**: They produce the expected output from given input.
- **Clustering**: They identify a unique feature of the data and classify it without any knowledge of prior data.
- **Associating**: You can train neural networks to "remember" patterns. When you show an unfamiliar version of a pattern, the network associates it with the most comparable version in its memory and reverts to the latter.

Neural vs. Conventional Computers

One of the primary differences between conventional, or traditional, computers and neural computers is that conventional machines process data sequentially, while neural networks can do many things at once. Here are some of the other major differences between conventional and neural computers:

•**Following Instructions vs. Learning Capability**: Conventional computers learn only by performing steps or sequences set by an algorithm, while neural networks continuously adapt their programming and essentially program themselves to find solutions. Conventional computers are limited by their design, while neural networks are designed to surpass their original state.

•**Rules vs. Concepts and Imagery**: Conventional computers operate through logic functions based on a given set of rules and calculations. In contrast, artificial neural networks can run through logic functions and use abstract concepts, graphics, and photographs. Traditional computers are rules-based, while artificial neural networks perform tasks and then learn from them.

•**Complementary, Not Equal**: Conventional algorithmic computers and neural networks complement each other. Some tasks are more arithmetically based and don’t require the learning ability of neural networks. Often though, tasks require the capabilities of both systems. In these cases, the conventional computer supervises the neural network for higher speed and efficiency.

What are the current challenges and limitations of neural networks and deep learning?

1 Data and computation

One of the main challenges of neural networks and deep learning is the need for large amounts of data and computational resources. Neural networks learn from data by adjusting their parameters to minimize a loss function, which measures how well they fit the data. However, to achieve high accuracy and generalization, they often require millions or billions of data points, which may not be available or accessible for some tasks or domains. Moreover, training and deploying neural networks can be very expensive and time-consuming, as they involve complex mathematical operations and multiple layers of neurons. Therefore, data and computation are key factors that affect the feasibility and scalability of neural networks and deep learning.

2 Interpretability and explainability

Another challenge of neural networks and deep learning is the lack of interpretability and explainability of their outputs and decisions. Neural networks are often considered as black boxes, as it is hard to understand how they process the input data and what features they learn and use. This can pose problems for applications that require transparency, accountability, and trust, such as healthcare, finance, or law. For example, how can we trust a neural network that diagnoses a disease or recommends a treatment, if we do not know how it arrived at that conclusion? How can we debug or improve a neural network that makes a mistake or fails to perform as expected? Therefore, interpretability and explainability are essential for ensuring the reliability and ethics of neural networks and deep learning.

3 Robustness and security

A related challenge of neural networks and deep learning is the lack of robustness and security against adversarial attacks and noise. Neural networks are vulnerable to subtle perturbations or modifications of the input data, which can cause them to produce incorrect or misleading outputs. For example, adding a small amount of noise or changing a few pixels in an image can fool a neural network into misclassifying it as a different object. This can have serious consequences for applications that rely on accurate and consistent recognition, such as face recognition, autonomous driving, or biometric authentication. Therefore, robustness and security are crucial for ensuring the safety and integrity of neural networks and deep learning.

4 Generalization and transfer

A final challenge of neural networks and deep learning is the difficulty of generalizing and transferring their knowledge and skills to new or different domains or tasks. Neural networks tend to overfit the data they are trained on, which means they perform well on the training data but poorly on unseen or novel data. This can limit their ability to adapt to changing or diverse environments or scenarios. Moreover, neural networks tend to learn specific and low-level features that may not be relevant or useful for other domains or tasks. This can prevent them from leveraging their existing knowledge and skills to learn new or related ones. Therefore, generalization and transfer are important for enhancing the versatility and efficiency of neural networks and deep learning.

Neural Network

- A neural network is a very powerful machine learning mechanism which basically mimics how a human brain learns.

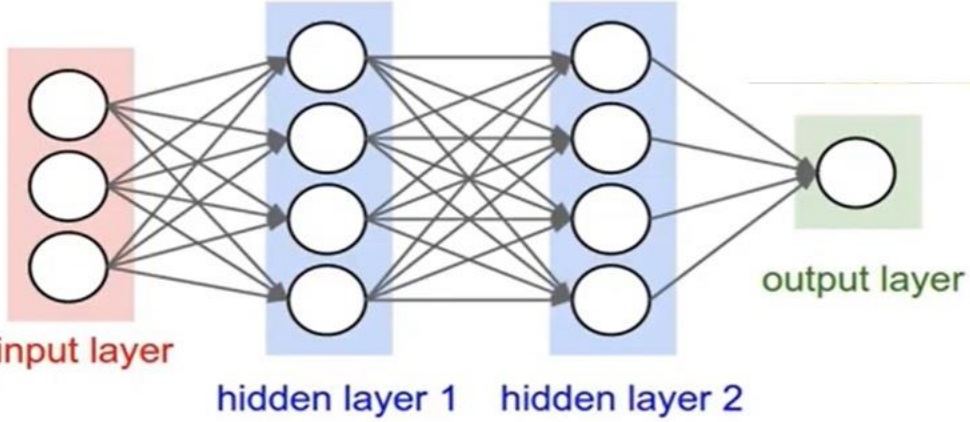
Activation Function ?

- Activation function are an extremely important feature of the artificial neuron network. They basically decide whether a neuron should be activated or not.
- It limit the output signal to a finite value.
- Logit = (Input * Weight) + Bias
- Bias is the information which can impact output without being dependent on the any feature.

- Bias is the information which can impact output without being dependent on the any feature.

- It does the non-linear transformation to the input to the input making it capable to learn and perform more complex relationship.

Neural Network



Without an activation function, the neural network is just a linear regression model.

Lets say we have five layers but no activation function. We multiple the input data by weight matrix of first, add bias and send next and again do same, and so on. We could easily combine all layer into one.

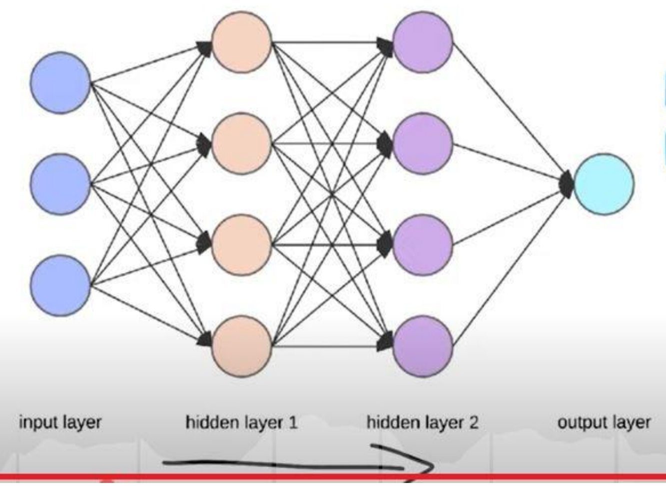
NO curve in linear regression, so can not solve complex pattern.

In second picture, we can not draw straight line slope. Hence non-linearity or curve required to solve complex problems.

- We can represent any kind of function with neural network. Hence neural network consider as UNIVERSAL FUNCTION APPROXIMATORS, means they can compute any function or any process.
- The main purpose of activation function is to introduce non-linearity in the network so it would be capable of learning more complex pattern.

Difference Between CNN,RNN and ANN

1. Artificial Neural Network (or ANN)



Feed-Forward Neural network

(forward direction)

Advantages of ANN

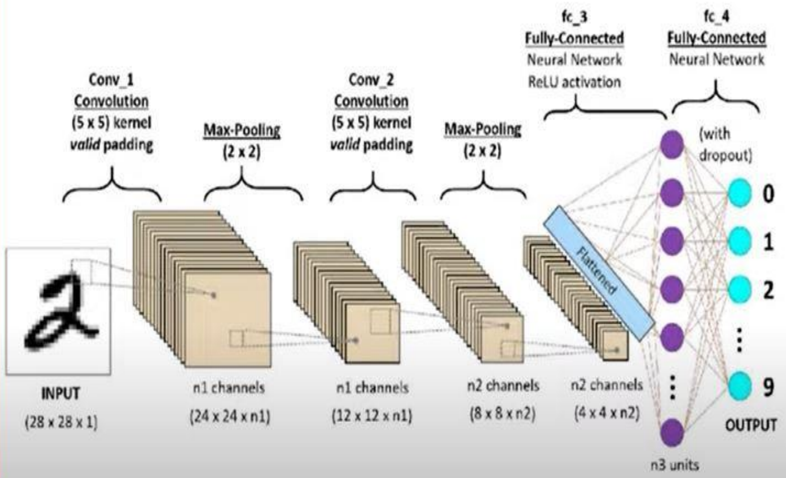
Advantages

- they store information on the entire network
- they have the ability to work with incomplete knowledge
- they offer fault tolerance and have distributed memory
- they offer us the ability to work with incomplete knowledge

Disadvantages

- they have huge hardware dependency
- they sometimes have unexplained behavior which can leave us tormeneted with results
- there is no specific rule for determining the structure of artificial neural networks and appropriate network structure is achieved through experience and trial and error

2. Convolutional Neural Network (or CNN)



entirely connected or pooled

nonlinear processing

Advantages of CNN

Advantages

- they offer very high accuracy in image recognition problems
- they are capable of automatically detecting important features without any human supervision
- weight sharing

Disadvantages

- CNNs do not encode the position and orientation of object
- they lack the ability to be spatially invariant to the input data
- a lot of training data is required in order for it to work efficiently

Applications of RNN:

- Machine Translation
- Robot Control
- Time Series Prediction
- Speech Recognition
- Speech Synthesis
- Time Series Anomaly Detection
- Rhythm Learning
- Music Composition

ANN	CNN	RNN
Tabular or Text Data	Image Data	Sequence data
No Parameter Sharing	Yes	Yes
Operate on Fixed Length input	Operate on Fixed Length input	Don't
No Recurrent Connections	No Recurrent Connections	They are Possible
No Spatial Relationships	They are Possible	No Spatial Relationships
ANN is considered to be less powerful than CNN, RNN	CNN is considered to be more powerful than others	RNN includes less feature compatibility when compared to CNN
Having fault tolerance, Ability to work with incomplete knowledge	High accuracy in image recognition problems, & weight sharing	Remembers each and every information, & offers time series prediction

What is an example of RNN in real life?

Apple's Siri and Google's voice search both use Recurrent Neural Networks (RNNs), which are the state-of-the-art method for sequential data. It's the first algorithm with an internal memory that remembers its input, making it perfect for problems involving sequential data in machine learning.

What is a real life example of ANN?

Google Smart home technology, Siri, Alexa, Cortana, etc, are just some of the real-life examples of ANN at work. Further, ANN often divides its function into various requirements when it comes to languages.

Activation function is a function which decide the output of a particular node in any neural network.

Why Are Activation Functions Essential?

Without activation functions, neural networks would just consist of linear operations like matrix multiplication. All layers would perform linear transformations of the input, and no non-linearities would be introduced.

Most real-world data is non-linear. For example, relationships between house prices and size, income, and purchases, etc., are non-linear. If neural networks had no activation functions, they would fail to learn the complex non-linear patterns that exist in real-world data.

Activation functions enable neural networks to learn these non-linear relationships by introducing non-linear behaviors through activation functions. This greatly increases the flexibility and power of neural networks to model complex and nuanced data.

Types of Activation Functions

Neural networks leverage various types of activation functions to introduce non-linearities and enable learning complex patterns. Each activation function has its own unique properties and is suitable for certain use cases.

For example, the sigmoid function is ideal for binary classification problems, softmax is useful for multi-class prediction, and ReLU helps overcome the vanishing gradient problem.

Using the right activation function for the task leads to faster training and better performance.

Choosing the Right Activation Function

The choice of activation function depends on the type of problem you are trying to solve. Here are some guidelines:

For binary classification:

Use the sigmoid activation function in the output layer. It will squash outputs between 0 and 1, representing probabilities for the two classes.

For multi-class classification:

Use the softmax activation function in the output layer. It will output probability distributions over all classes.

If unsure:

Use the ReLU activation function in the hidden layers. ReLU is the most common default activation function and usually a good choice.

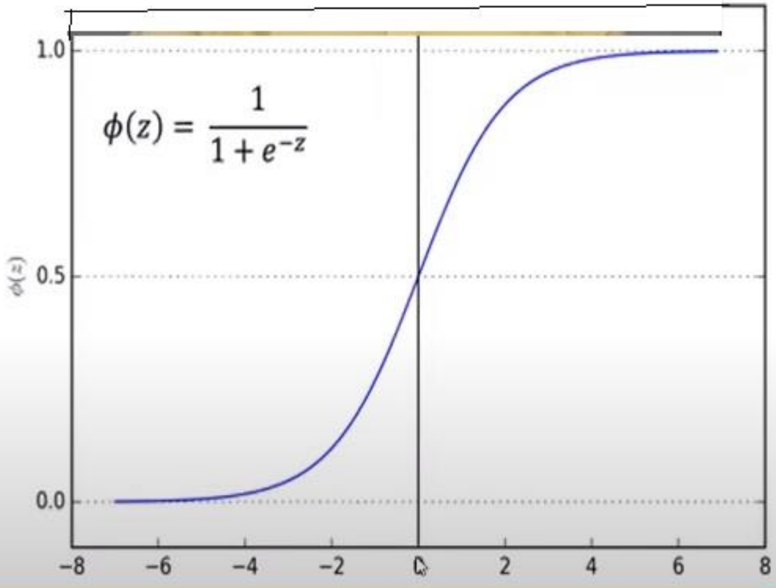
Some important Activation Functions.

- Sigmoid (Logistic Activation Function)
- TanH (Hyperbolic tangent Activation Function)
- ReLu (Restricted Linear Unit)
- SoftMax

Sigmoid

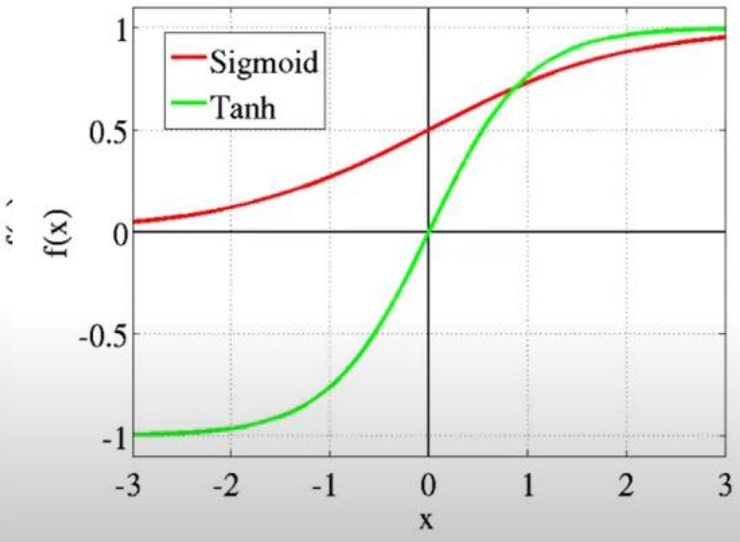
- The main reason why we use sigmoid function is because it exists between **(0 to 1).**

- Therefore, it is especially used for models where we have to **predict the probability** as output. Since probability of anything exists only between the range of **0 and 1,** sigmoid is the right choice



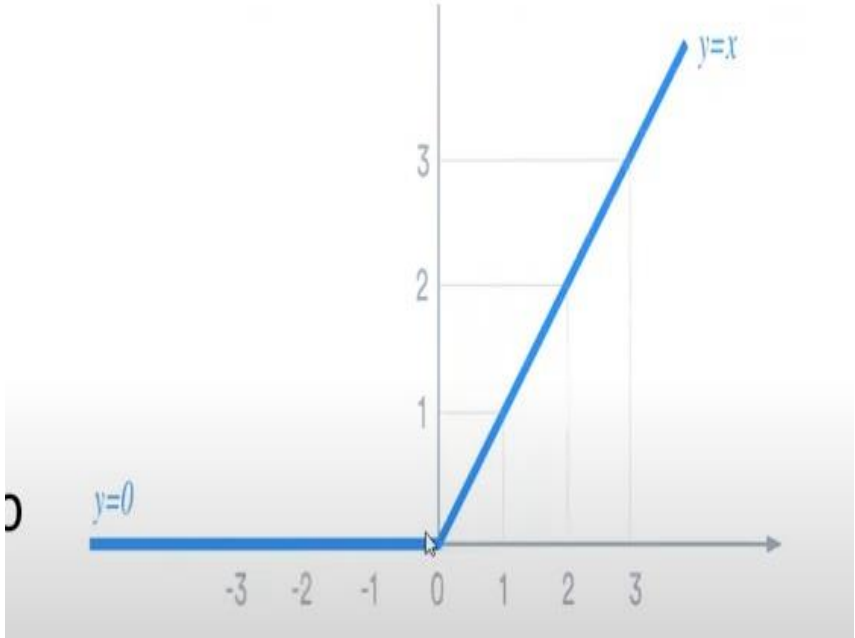
TanH

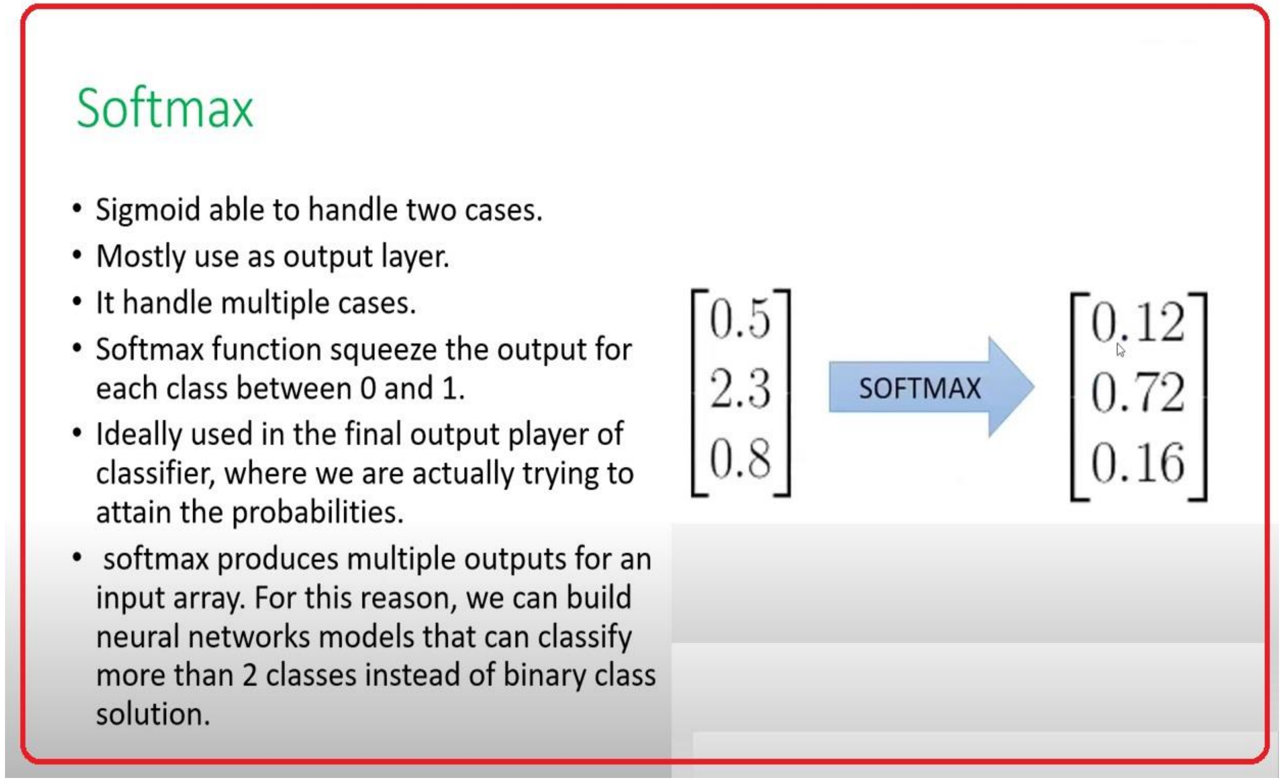
- tanh is also like logistic sigmoid but in better way. The range of the tanh function is from (-1 to 1). tanh is also sigmoidal (s - shaped).
- TanH is often preferred over the sigmoid neuron because it is zero centered.
- The advantage is that the negative inputs will be mapped strongly negative and the zero inputs will be mapped near zero in the tanh graph.



ReLu (Restricted Linear Unit)

- The ReLU is the most used activation function in the world right now.Since, it is used in almost all the convolutional neural networks or deep learning.
- As you can see, the ReLU is half rectified (from bottom). f(z) is zero when z is less than zero and f(z) is equal to z when z is above or equal to zero.
- **Range:** [0 to infinity)





The Softmax and Sigmoid functions are both activation functions used in neural networks. The primary difference between them is that the Softmax function works for multiclass classification problems and the Sigmoid function is a better option for binary-class problems.

Linear Activation Function (Identity)

In deep learning, data scientists use linear activation functions, also known as identity functions, when they want the output to be the same as the input signal**. Identity is differentiable, and like a train passing through a station without stopping, this activation function doesn't change the signal in any way, so it's not used within internal layers of a DL network.**

Although, in most cases, this might not sound very useful, it is when you want the outputs of our neural network to be continuous rather than modified or discrete.

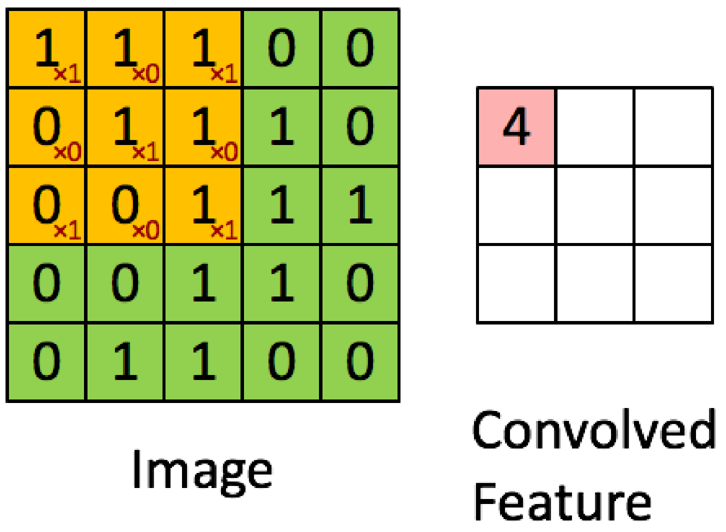
There is no convergence of data, and nothing decreases either. If you use this activation function for every layer, then it would collapse the layers in a neural network into one. So, not very useful unless that's exactly what you need or there are different activation functions in the subsequent hidden layers.

Convolution, Padding, Stride, and Pooling in CNN

Convolution operation

The convolution is a mathematical operation used to extract features from an image. The convolution is defined by an image kernel. The image kernel is nothing more than a small matrix. Most of the time, a 3x3 kernel matrix is very common.

In the below fig, the green matrix is the original image and the yellow moving matrix is called kernel, which is used to learn the different features of the original image. The kernel first moves horizontally, then shift down and again moves horizontally. The sum of the dot product of the image pixel value and kernel pixel value gives the output matrix. Initially, the kernel value initializes randomly, and its a learning parameter.



There are some standard filters like **Sobel filter,** contains the value 1, 2, 1, 0, 0, 0, -1, -2, -1, the advantage of this is it puts a little bit more weight to the central row, the central pixel, and this makes it maybe a little bit more robust. Another filter used by computer vision researcher is instead of a 1, 2, 1, it is 3, 10, 3 and then -3, -10, -3, called a **Scharr filter**. And this has yet other slightly different properties and this can be used for vertical edge detection. If it is flipped by 90 degrees, the same will act like horizontal edge detection.

Problem Type	Last-layer Output Nodes	Hidden-layer activation	Last-layer activation	Loss function
Binary classification	1	RELU (first choice), Tanh (for RNNs)	Sigmoid	Binary Crossentropy
Multi-class, single-label classification	Number of classes		Softmax	Categorical Crossentropy
Multi-class, multi-label classification	Number of classes		Sigmoid (one for each class)	Binary Crossentropy
Regression to arbitrary values	1		None	MSE
Regression to values between 0 and 1	1		Sigmoid	MSE/Binary Crossentropy

Function	Range	0-centered	Saturation	Vanishing Gradient	Computation
Sigmoid	0,1	No	For negative and positive values	Yes	Compute-intensive
Tanh	-1,1	Yes	For negative and positive values	Yes	Compute-intensive
ReLu	0, +∞	No	For negative values	Yes (Better than sigmoid and tanh)	Easy to compute
Leaky ReLu	-∞, +∞	Close	No	No	Easy to compute

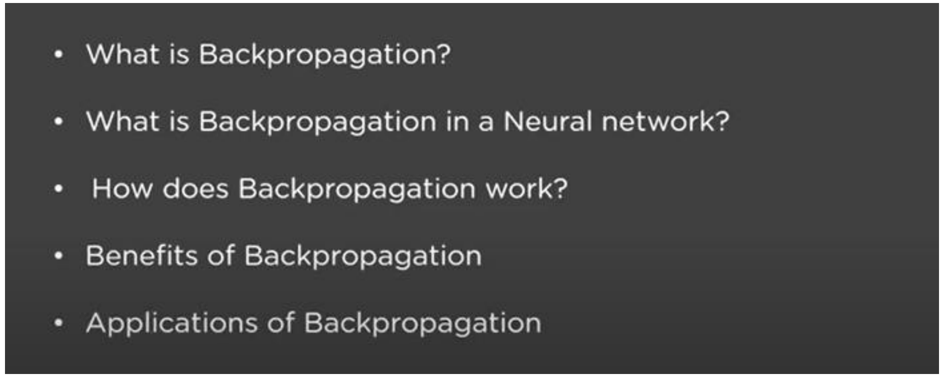
Vanishing

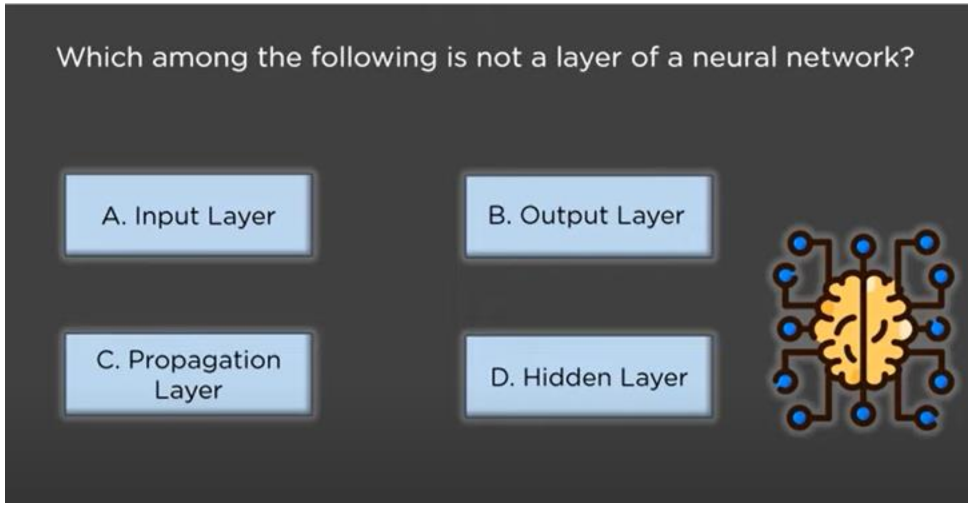
As the backpropagation algorithm advances downwards(or backward) from the output layer towards the input layer, the gradients often get smaller and smaller and approach zero which eventually leaves the weights of the initial or lower layers nearly unchanged. As a result, the gradient descent never converges to the optimum. This is known as the ***vanishing gradients*** problem.

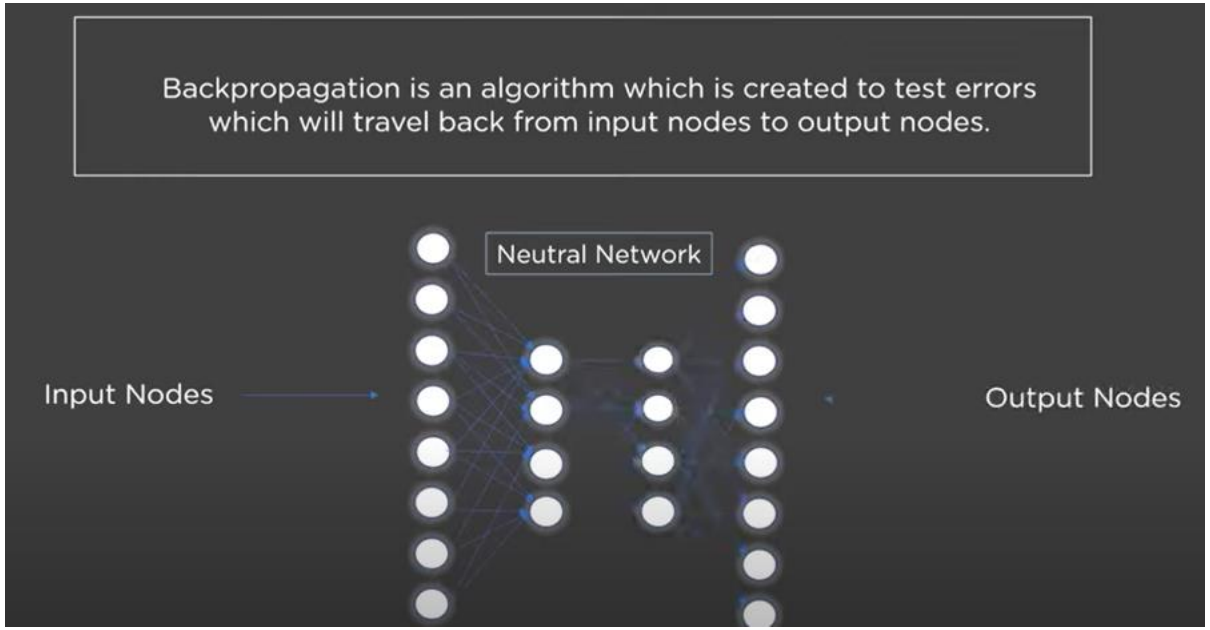
Exploding

On the contrary, in some cases, the gradients keep on getting larger and larger as the backpropagation algorithm progresses. This, in turn, causes very large weight updates and causes the gradient descent to diverge. This is known as the ***exploding gradients*** problem.

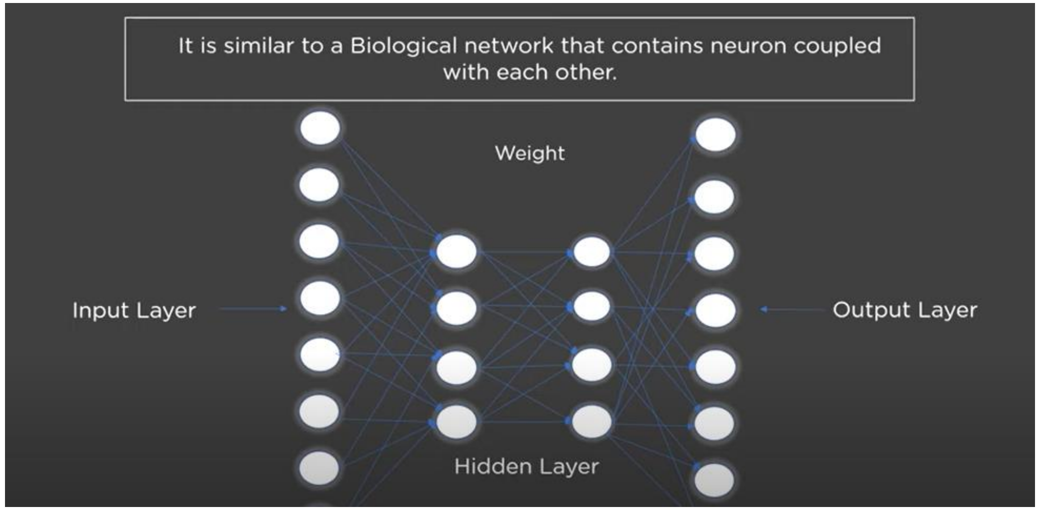
Backpropagation in Neural Networks








What is Backpropagation
in neural network?

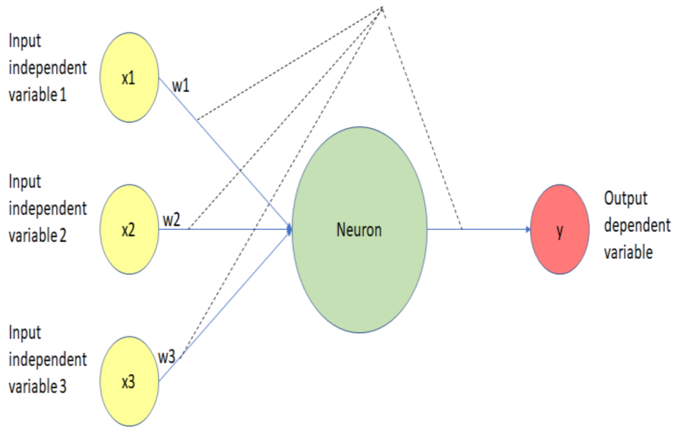

It is similar to a Biological network that contains neuron coupled with each other.
Weight
Input Layer
Output Layer
Hidden Layer

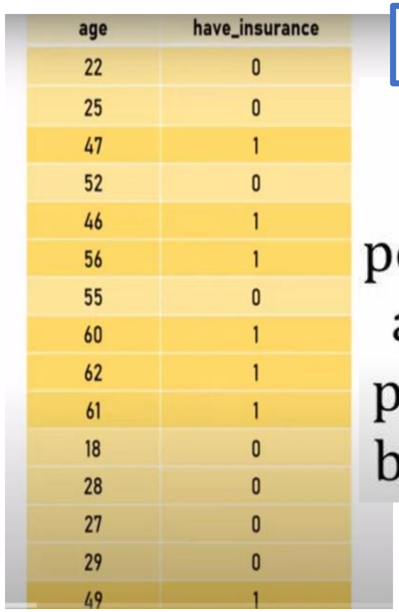

How does
Backpropagation work?

What is Neurons

Neurons are the building blocks of the nervous system. They receive and transmit signals to different parts of the body.

Neurons in deep learning models are nodes through which data and computations flow. Neurons work like this: They receive one or more input signals. These input signals can come from either the raw data set or from neurons positioned at a previous layer of the neural net.

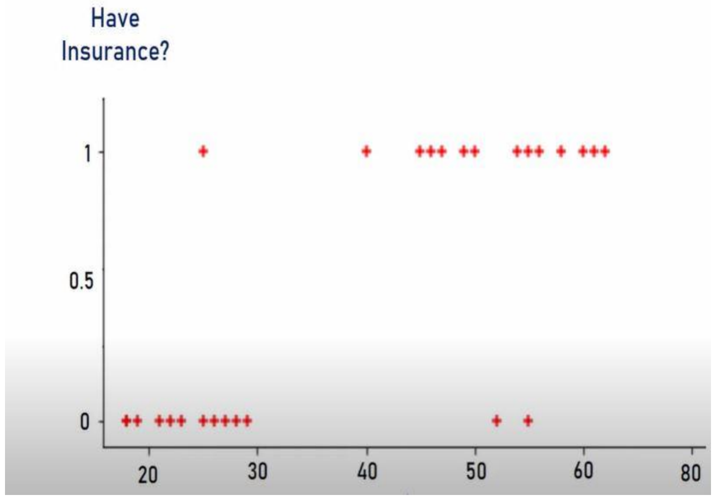


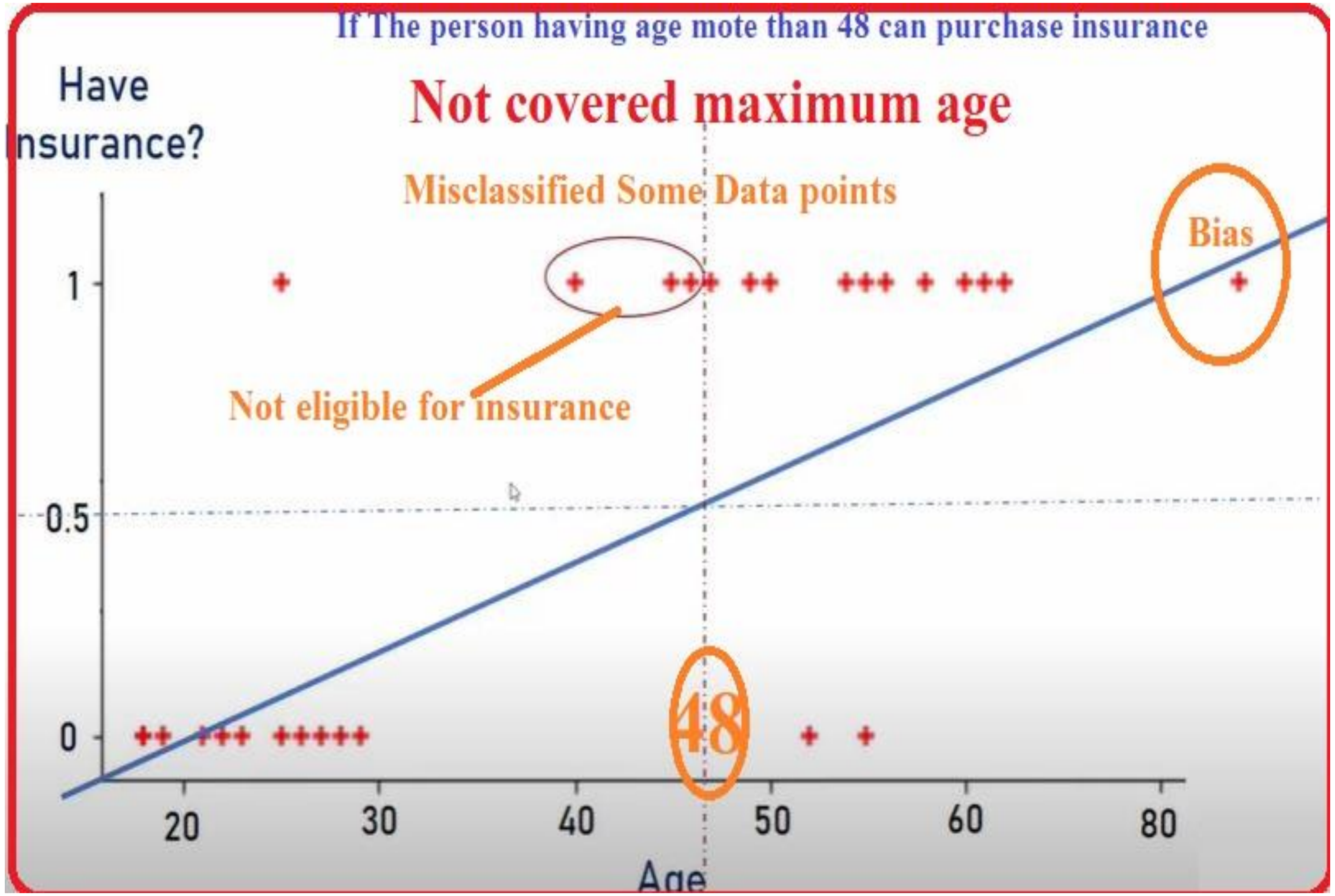


age	have_insurance
22	0
25	0
47	1
52	0
46	1
56	1
55	0
60	1
62	1
61	1
18	0
28	0
27	0
29	0
49	1

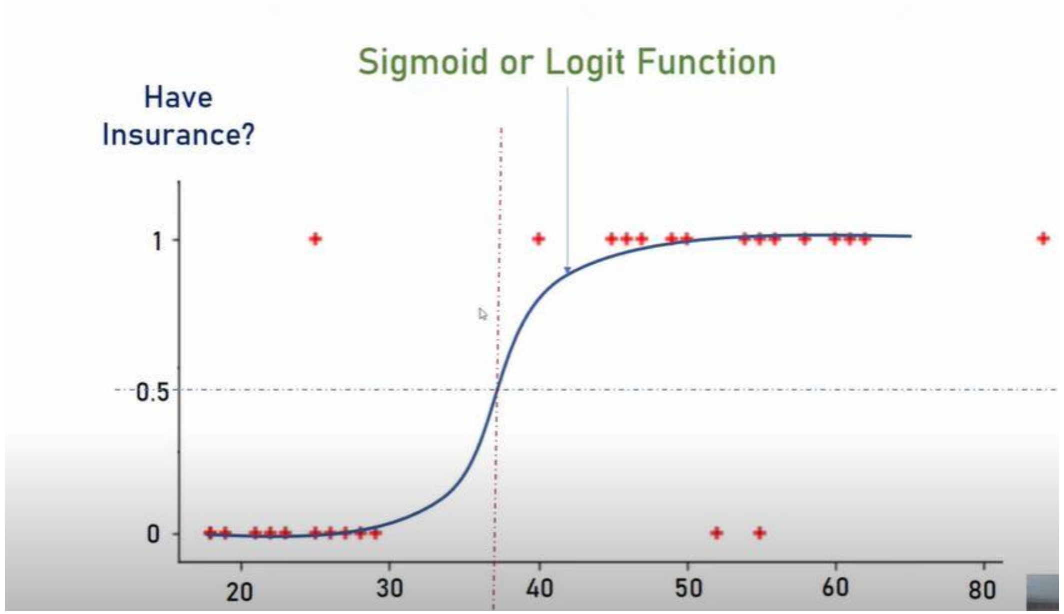
Binary Classification

Given an age of a person, come up with a **function** that can predict if person will buy insurance or not




If The person having age mote than 48 can purchase insurance
Have Insurance?
Not covered maximum age
Misclassified Some Data points
Bias
Not eligible for insurance
1
0.5
0
48
20
30
40
50
60
80
Age

IF we Want to cover maximum age group Then We will use Sigmoid Function



$$sigmoid(z) = \frac{1}{1+e^{-z}}$$

e = Euler's number ~ 2.71828

$sigmoid(200) = \frac{1}{1+2.71^{-200}}$ = almost close to 1

$sigmoid(-200) = \frac{1}{1+2.71^{200}}$ = almost close to 0

Sigmoid function converts input into range 0 to 1

Step 1

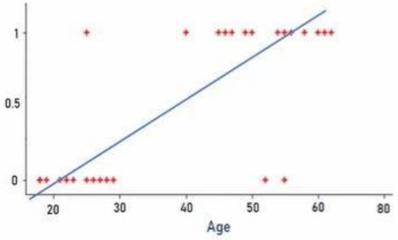
$$y = m * x + b$$

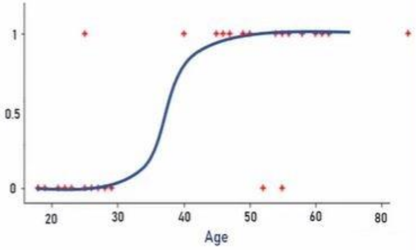
Age

Step 2

$$z = \frac{1}{1 + e^{-y}}$$

If person will buy insurance





$$y = 0.042 * x - 1.53$$

Age

NEURONS FOR SINGLE VARIABLES

value < 0.5 = person will not buy insurance

value >= 0.5 = person **will** buy insurance

Age = 35 → $y = 0.042 * x - 1.53$ | $z = \frac{1}{1 + e^{-y}}$ → 0.48

$$y = 0.042 * \underset{\text{Age}}{x} - 1.53$$

$$y = 0.042 * \underset{\text{Age}}{x1} + 0.008 * \underset{\text{Income}}{x2} + 0.2 * \underset{\text{Education}}{x3} - 1.53$$

$$y = w1 * x1 + w2 * x2 + w3 * x3 + b$$

$$y = \sum_{i=0}^{n} w^i x^i + b$$

Agriculture

1. Optimize yield production by using data from sensors and satellites taking into account temperature, humidity, etc.

Aerospace & Defence

2. Identify objects from images acquired via satellites
3. Use surveillance cameras to detect suspicious events or gather intelligence

Automotive

4. Develop autonomous things including vehicles. There are numerous deep learning models used in such devices including those for detecting traffic signs & lights, other vehicles, pedestrians, etc.

Financial services

5. Trading: Estimate future stock market prices
6. Fraud detection: Detect fraudulent activities with higher accuracy and fewer false positives
7. Evaluate a client's creditworthiness by analyzing information from multiple sources and responding to loan applications faster

Healthcare

11. Diagnose diseases leveraging medical imaging solutions, for example recognition of potential cancerous lesions on radiology images
12. Personalize medical treatments
13. Determine patients most at risk in the healthcare system

Insurance

14. Automate claims and damage analysis from reports or images
15. Image-based risk prediction for home insurance
16. Pricing risk

Manufacturing

Manufacturing companies including discrete manufacturing like automotive or other industrial companies (e.g. oil&gas) rely on deep learning algorithms:

17. Provide advanced analytics tools for processing big data about manufacturing

18. Generate automated alerts about the issues of production lines (e.g. on quality assurance or safety) using sensor data to notify relevant teams on time

19. Support predictive maintenance systems by analyzing images and other sensor data

20. Empower industrial robots with sensors and computer vision skills

Manufacturing

Manufacturing companies including discrete manufacturing like automotive or other industrial companies (e.g. oil&gas) rely on deep learning algorithms:

17.Provide advanced analytics tools for processing big data about manufacturing

18.Generate automated alerts about the issues of production lines (e.g. on quality assurance or safety) using sensor data to notify relevant teams on time

19.Support predictive maintenance systems by analyzing images and other sensor data

20.Empower industrial robots with sensors and computer vision skills

17.Monitor working environment around heavy machineries automatically to ensure people and items are at a safe distance

Pharmaceuticals & Medical Products

22. Drug discovery: Prediction of drug effects, monitoring the use of drug and identifying its side effects
23. Enable precision medicine which includes remedies based on genetic, environmental or lifestyle factors (also called personalised medicine)

Public sector

24. Make predictions about population health risks
25. Facial recognition for security checks

Retail & E-commerce

26. Offer new shopping experiences such as “Just Walk Out” stores, and checkout-less shopping. For more, feel free to read our article on cashierless stores.
27. Other shopping experiences powered by deep learning include voice-enabled shopping and in-store robots.
28. Image search: Scanning the image of the product to find the product on the store or suggest similar alternatives
29. Forecasting product demand more accurately according to buying habits analysis and future trend predictions
30. Deliver effective inventory management to prevent of out-of-stock and oversupply

Analytics

33.Most deep learning applications empower analytics solutions. Therefore analytics departments rely on deep learning in numerous cases

Customer success

34.Chatbots offering immediate and personalized customer service
35.Monitor customers’ responses, reviews and social media activity to identify what they say about the brand

36.Churn prevention: Examine data in customer feedback forms/texts, identify potential churners and communicate with the customer without losing time

Cybersecurity

35.Intrusion detection/prevention systems (IDS / IPS): Investigate user activities and network traffic to prevent malicious activities and reduce false alerts

Operations

36.Automatically extract data from documents using deep learning models

Sales & Marketing

•Create personalised advertisements according to browsing data
•Identify potential clients that are most likely to buy the solution
•Logo and counterfeit item detection in social media for brand protection