

Data Science Interview Questions

1. What is the statistical test for data validation with an example, Chi-square, ANOVA test, Z statics, T statics, F statics, Hypothesis Testing?

Before discussing the different statistical tests, we need to get a clear understanding of what a null hypothesis is. A null hypothesis proposes that no significant difference exists in the set of a given observation.

Null: Two samples mean are equal.

Alternate: Two samples mean are not equal. For rejecting the null hypothesis, a test is calculated. Then the test statistic is compared with a critical value, and if found to be greater than the critical value, the hypothesis will be rejected.

Critical Value: Critical values are the point beyond which we reject the null hypothesis. Critical value tells us, what is the probability of N number of samples, belonging to the same distribution. Higher, the critical value which means lower the probability of N number of samples belonging to the same distribution.

Critical values can be used to do hypothesis testing in the following way.

1. Calculate test statistic
2. Calculate critical values based on the significance level alpha
3. Compare test statistics with critical values.

IMP - If the test statistic is lower than the critical value, accept the hypothesis or else reject the hypothesis.

Chi-Square Test:

A chi-square test is used if there is a relationship between two categorical variables.

Chi-Square test is used to determine whether there is a significant difference between the expected frequency and the observed frequency in one or more categories. Chi-square is also called the non-parametric test as it will not use any parameter

2-Anova test:

ANOVA, also called an analysis of variance, is used to compare multiples (three or more) samples with a single test. Useful when there are more than three populations. Anova compares the variance within and between the groups of the population. If the variation is much larger than the within variation, the means of different samples will not be equal. If

the between and within variations are approximately the same size, then there will be no significant difference between sample means.

Assumptions of ANOVA:

- 1-All populations involved follow a normal distribution.
- 2-All populations have the same variance (or standard deviation).
- 3-The samples are randomly selected and independent of one another.

ANOVA uses the mean of the samples or the population to reject or support the null hypothesis. Hence it is called parametric testing.

3-Z Statics:

In a z-test, the samples are assumed to be normally distributed. A z score is calculated with population parameters as “population mean” and “population standard deviation” and it is used to validate a hypothesis that the sample drawn belongs to the same population. The statistics used for this hypothesis testing is called z-statistic, the score for which is calculated as $z = (x - \mu) / (\sigma / \sqrt{n})$, where x = sample mean μ = population mean σ / \sqrt{n} = population standard deviation. If the test statistic is lower than the critical value, accept the hypothesis or else reject the hypothesis.

4- T Statics:

A t-test is used to compare the mean of the given samples. Like z-test, t-test also assumed a normal distribution of the samples. A t-test is used when the population parameters (mean and standard deviation) are unknown.

There are three versions of t-test

1. Independent samples t-test which compares means for two groups
2. Paired sample t-test which compares mean from the same group at different times
3. Sample t-test, which tests the mean of the single group against the known mean. The statistic for hypothesis testing is called t-statistic, the score for which is calculated as

$$t = (x_1 - x_2) / (\sigma / \sqrt{n_1} + \sigma / \sqrt{n_2}),$$

where

x_1 = It is mean of sample A, x_2 = mean of sample B,

n_1 = size of sample 1 n_2 = size of sample 2

5- F Statics:

The F-test is designed to test if the two population variances are equal. It compares the ratio of the two variances. Therefore, if the variances are equal, then the ratio of the variances will be 1.

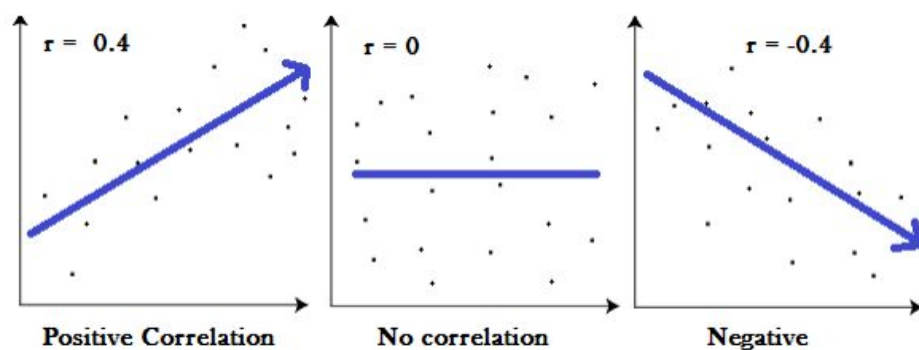
The F-distribution is the ratio of two independent chi-square variables divided by their respective degrees of freedom.

$$F = s_1^2 / s_2^2 \text{ and where } s_1^2 > s_2^2.$$

If the null hypothesis is true, then the F test-statistic given above can be simplified. This ratio of sample variances will be tested statistic used. If the null hypothesis is false, then we will reject the null hypothesis that the ratio was equal to 1 and our assumption that they were equal.

2. Explain Correlation and Pearson Correlation Coefficient?

The correlation coefficient is a statistical measure that calculates the strength of the relationship between the relative movements of two variables. We use it to measure both the strength and direction of a linear relationship between two variables the values range between -1.0 and 1.0. A calculated number greater than 1.0 or less than -1.0 means that there was an error in the correlation measurement. A correlation of -1.0 shows a perfect negative correlation, while a correlation of 1.0 shows a perfect positive correlation.



Correlation coefficient formulas are used to find how strong a relationship is between data. The formulas return a value between -1 and 1,

where: 1 indicates a strong positive relationship. -1 indicates a strong negative relationship. A result of zero indicates no relationship at all.

Meaning

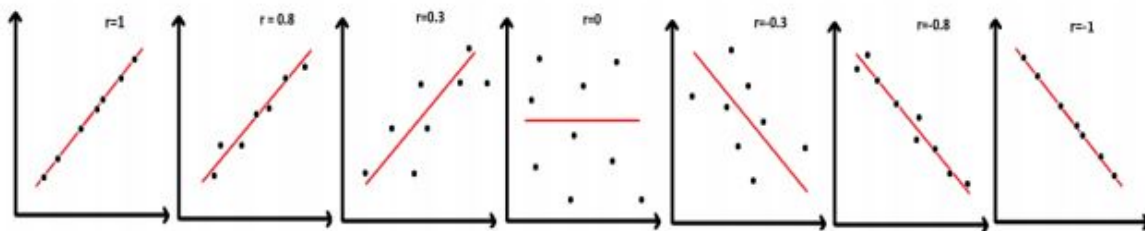
1. A correlation coefficient of 1 means that for every positive increase in one variable, there is a positive increase in a fixed proportion in the other. For example, shoe sizes go up in (almost) perfect correlation with foot length.
2. A correlation coefficient of -1 means that for every positive increase in one variable, there is a negative decrease of a fixed proportion in the other. For example, the amount of gas in a tank decreases in (almost) perfect correlation with speed.
3. Zero means that for every increase, there isn't a positive or negative increase. The two just aren't related.

What is a Negative Correlation?

Negative correlation is a relationship between two variables in which one variable increases as the other decreases, and vice versa. In statistics, a perfect negative correlation is represented by the value -1. Negative correlation or inverse correlation is a relationship between two variables whereby they move in opposite directions. If variables X and Y have a negative correlation (or are negatively correlated), as X increases in value, Y will decrease; similarly, if X decreases in value, Y will increase.

What Is Positive Correlation?

Positive correlation is a relationship between two variables in which both variables move in tandem—that is, in the same direction. A positive correlation exists when one variable decreases as the other variable decreases or one variable increases while the other increases.



We use the correlation coefficient to measure the strength and direction of the linear relationship between two numerical variables X and Y. The correlation coefficient for a sample of data is denoted by r .

Pearson Correlation Coefficient

Pearson is the most widely used correlation coefficient. Pearson correlation measures the linear association between continuous variables. In other words, this coefficient quantifies the degree to which a relationship between two variables can be described by a line. Formula developed by Karl Pearson over 120 years ago is still the most widely used today.

The formula for the correlation (r) is

Correlation Coefficient Formula

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}}$$

Where n is the number of pairs of data;

Are the sample means of all the x-values and all the y-values, respectively; and s_x and s_y are the sample standard deviations of all the x- and y-values, respectively.

1. Find the mean of all the x-values and mean of all y-values.
2. Find the standard deviation of all the x-values (call it s_x) and the standard deviation of all the y-values (call it s_y). For example, to find s_x , you would use the following equation:
3. For each of the n pairs (x, y) in the data set, take
4. Add up the n results from Step 3.
5. Divide the sum by $s_x * s_y$.
6. Divide the result by $n - 1$, where n is the number of (x, y) pairs. (It's the same as multiplying by 1 over $n - 1$.) This gives you the correlation, r.

3. What is the difference between machine learning and deep learning?

Machine Learning | deep learning Machine Learning is a technique to learn from that data and then apply what has been learnt to make an informed decision | The main difference between deep and machine learning is, machine learning models become better progressively but the model still needs some guidance. If a machine learning model returns an inaccurate prediction then the programmer needs to fix that problem explicitly but in the case of deep learning, the model does it by himself.

>Machine Learning can perform well with small size data also | Deep Learning does not perform as good with smaller datasets.

>Machine learning can work on some low-end machines also | Deep Learning involves many matrix multiplication operations which are better suited for GPUs

>Features need to be identified and extracted as per the domain before pushing them to the algorithm | Deep learning algorithms try to learn high-level features from data.

>It is generally recommended to break the problem into smaller chunks, solve them and then combine the results | It generally focuses on solving the problem end to end

>Training time is comparatively less | Training time is comparatively more

>Results are more interpretable | Results Maybe more accurate but less interpretable

> No use of Neural networks | uses neural networks > Solves comparatively less complex problems | Solves more complex problems.

4. What is perceptron and how it is related to human neurons?

If we focus on the structure of a biological neuron, it has dendrites, which are used to receive inputs. These inputs are summed in the cell body and using the Axon it is passed on to the next biological neuron as shown below.

Dendrite: Receives signals from other neurons

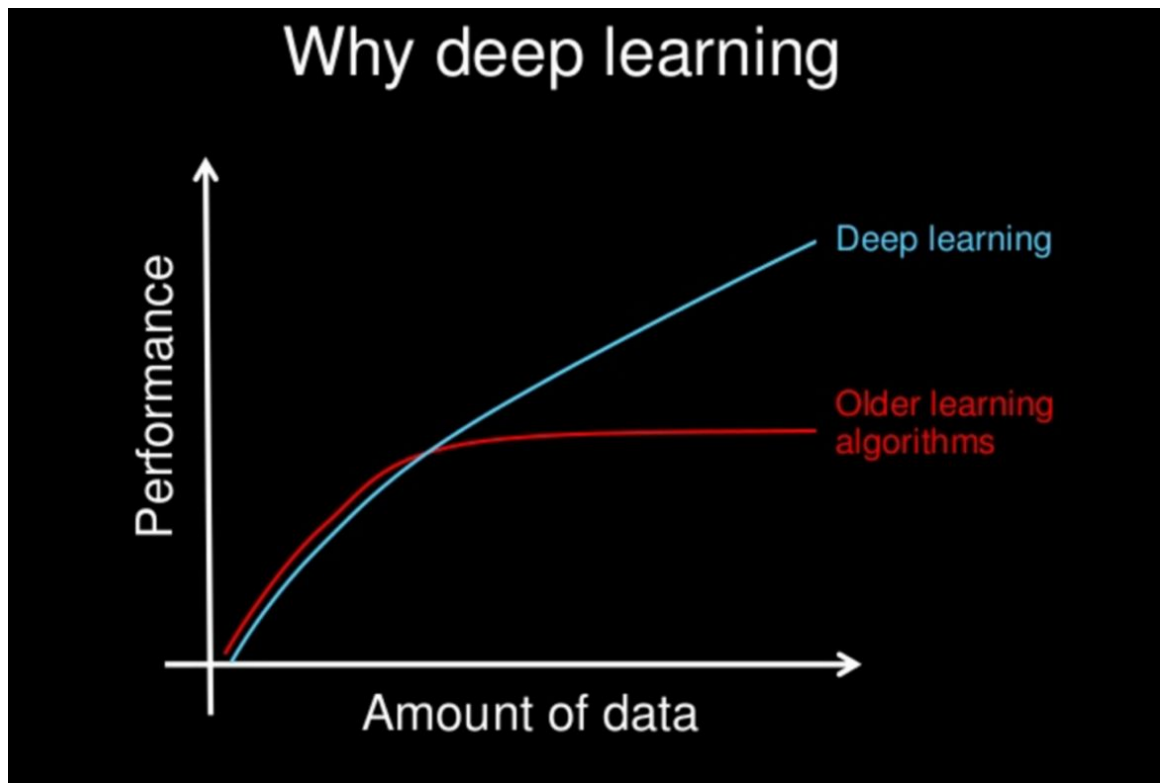
Cell Body: Sums all the inputs

Axon: It is used to transmit signals to the other cells

Similarly, a perceptron receives multiple inputs, applies various transformations and functions and provides an output. A Perceptron is a linear model used for binary classification. It models a neuron, which has a set of inputs, each of which is given a specific weight. The neuron computes some function on these weighted inputs and gives the output.

5. Why deep learning is better than machine learning?

Though traditional ML algorithms solve a lot of our cases, they are not useful while working with high dimensional data that is where we have a large number of inputs and outputs. For example, in the case of handwriting recognition, we have a large amount of input where we will have different types of inputs associated with different types of handwriting.



The second major challenge is to tell the computer what are the features it should look for that will play an important role in predicting the outcome as well as to achieve better accuracy while doing so.

6. What types of problems can be solved by using deep learning?

Deep Learning is a branch of Machine Learning, which is used to solve problems in a way that mimics the human way of solving problems.

Examples:

- Image recognition
- Object Detection
- Natural Language processing- Translation, Sentence formations, text to speech, speech to text
- understand the semantics of actions

7. What is the activation function? List down all the activation functions using mathematical Expression and example. What is the activation function?

Activation functions are very important for an Artificial Neural Network to learn and make sense of something complicated and the Non-linear complex functional mappings between

the inputs and response variable. They introduce non-linear properties to our Network. Their main purposes are to convert an input signal of a node in an A-NN to an output signal.

So why do we need Non-Linearities?

Non-linear functions are those, which have a degree more than one, and they have a curvature when we plot a Nonlinear function. Now we need a Neural Network Model to learn and represent almost anything and any arbitrary complex function, which maps inputs to outputs. Neural-Networks are considered Universal Function Approximations. It means that they can compute and learn any function at all. Most popular types of Activation functions -

- Sigmoid or Logistic
- Tanh — Hyperbolic tangent
- ReLu -Rectified linear units

Sigmoid Activation function:

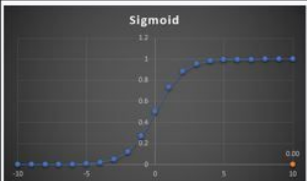
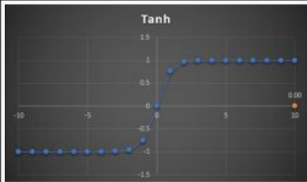
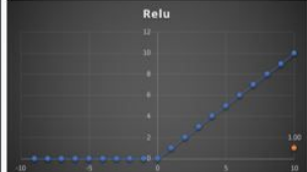
It is an activation function of form $f(x) = 1 / 1 + \exp(-x)$. Its Range is between 0 and 1. It is an S-shaped curve. It is easy to understand.

Hyperbolic Tangent function- Tanh:

Its mathematical formula is $f(x) = 1 - \exp(-2x) / 1 + \exp(-2x)$. Now the output is zero centred because its range is between -1 to 1 i.e. $-1 < \text{output} < 1$. Hence optimisation is easier in this method; Hence in practice, it is always preferred over Sigmoid function.

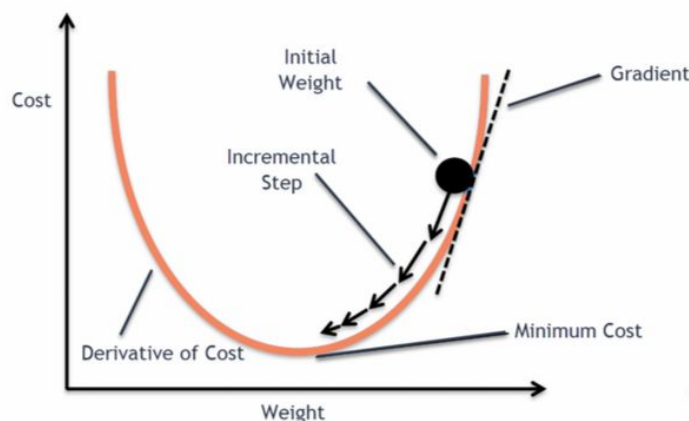
ReLu- Rectified Linear units:

It has become more popular in the past couple of years. It was recently proved that it has six times improvement in convergence from Tanh function. It's $R(x) = \max(0, x)$ i.e. if $x < 0$, $R(x) = 0$ and if $x \geq 0$, $R(x) = x$. Hence as seen in the mathematical form of this function, we can see that it is very simple and efficient.

Name	Plot	Equation	Derivative
Sigmoid		$f(x) = \sigma(x) = \frac{1}{1 + e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
Tanh		$f(x) = \tanh(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})}$	$f'(x) = 1 - f(x)^2$
Rectified Linear Unit (relu)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$

8. Detailed explanation about gradient descent using example and Mathematical expression?

Gradient descent is an optimization algorithm used to minimize some function by iteratively moving in the direction of steepest descent as defined by the negative of the gradient. In machine learning, we used gradient descent to update the parameters of our model. Parameters refer to coefficients in the Linear Regression and weights in neural networks.



The size of these steps is called the learning rate. With the high learning rate, we can cover more ground each step, but we risk overshooting the lower point since the slope of the hill is constantly changing. With a very lower learning rate, we can confidently move in the direction of the negative gradient because we are recalculating it so frequently. The Lower learning rate is more precise, but calculating the gradient is time-consuming, so it will take a very large time to get to the bottom.

Now let's run gradient descent using a new cost function. There are two parameters in cost function we can control: m (weight) and b (bias). Since we need to consider the impact each one has on the final prediction, we need to use partial derivatives. We calculate the partial derivative of the cost function concerning each parameter and store the results in a gradient.

Given the cost function:

$$f(m, b) = \frac{1}{N} \sum_{i=1}^n (y_i - (mx_i + b))^2$$

The gradient can be calculated as:

$$f'(m, b) = \begin{bmatrix} \frac{df}{dm} \\ \frac{df}{db} \end{bmatrix} = \begin{bmatrix} \frac{1}{N} \sum -2x_i(y_i - (mx_i + b)) \\ \frac{1}{N} \sum -2(y_i - (mx_i + b)) \end{bmatrix}$$

To solve for the gradient, we iterate by our data points using our new m and b values and compute the partial derivatives. This new gradient tells us about the slope of the cost function at our current position (current parameter values) and the directions we should move to update our parameters. The learning rate controls the size of our update.

9. What is gradient descent, mini-batch gradient descent, batch gradient descent, stochastic gradient descent and adam?

Gradient Descent it is an iterative machine learning optimisation algorithm to reduce the cost function, and help models to make accurate predictions. Gradient indicates the direction of increase. As we want to find the minimum points in the valley, we need to go in the opposite direction of the gradient. We update the parameters in the negative gradient direction to minimise the loss.

$$\theta = \theta - \eta \nabla J(\theta; x, y)$$

Where,

θ is the weight parameter, η is the learning rate, and $\nabla J(\theta; x, y)$ is the gradient of weight parameter θ

Types of Gradient Descent

Different types of Gradient descents are

- Batch Gradient Descent or Vanilla Gradient Descent
- Stochastic Gradient Descent
- Mini batch Gradient Descent

Batch Gradient Descent

In the batch gradient, we use the entire dataset to compute the gradient of the cost function for each iteration for gradient descent and then update the weights.

Stochastic Gradient Descent

Stochastic gradient descent, we use a single data point or example to calculate the gradient and update the weights with every iteration.

We first need to shuffle the datasets so that we get a completely randomised dataset.

As the datasets are random and weights, are updated for every single example, an update of the weights and the cost functions will be noisy jumping all over the place

Mini Batch Gradient Descent

Mini-batch gradients is a variation of stochastic gradient descent where instead of a single training example, a mini-batch of samples are used.

Mini -batch gradient descent is widely used and converges faster and is more stable. The batch size can vary depending upon the dataset.

As we take batches with different samples, it reduces the noise which is a variance of the weights updates, and that helps to have a more stable converge faster.

10. How to initialise biases in deep learning?

It is possible and common to initialise the biases to be zero since the random numbers in the weights provide the asymmetry braking. For ReLU non-linearities, some people like to

use small constant values such as 0.01 for all biases because this ensures that all ReLU units fire in the beginning, therefore obtain, and propagate some gradient. However, it is unclear if this provides a consistent improvement (in fact some results seem to indicate that this performs worse) and it is more commonly used to use 0 bias initialisation.

11. What is learning Rate?

Learning Rate The learning rate controls how much we should adjust the weights concerning the loss gradient. Learning rates are randomly initialised. Lower the values of the learning rate slower will be the convergence to global minima. Higher values for the learning rate will not allow the gradient descent to converge. Since our goal is to minimise the function cost to find the optimised value for weights, we run multiple iteration with different weights and calculate the cost to arrive at a minimum cost.

12. What is backward propagation?

Back-propagation is the essence of the neural net training and this method of fine-tuning the weights of a neural net based on the error rate obtained in the previous epoch. Proper tuning of the weights allows us to reduce error rates and to make the model reliable by increasing its generalisation. Backpropagation is a short form of "backward propagation of errors." This is the standard method of training artificial neural networks. This helps to calculate the gradient of a loss function with respect to all the weights in the network.

Most prominent advantages of Backpropagation are:

- Backpropagation is fast, simple and easy to program.
- It has no parameters to tune apart from the numbers of input.
- It is the flexible method as it does not require prior knowledge about the network
- It is the standard method that generally works well.
- It does not need any special mentions of the features of the function to be learned.

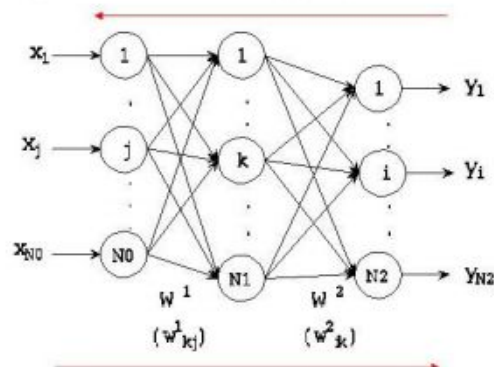
The BackPropagation Algorithm

Main idea:

For each example in the training set:

- compute the output signal
- compute the error corresponding to the output level
- propagate the error back into the network and store the corresponding delta values for each layer
- adjust each weight by using the error signal and input signal for each layer

Computation of the error signal (BACKWARD)



Computation of the output signal (FORWARD)

13. How do we assign weights in deep learning?

We already know that in a neural network, weights are usually initialised randomly and that kind of initialisation takes a fair/significant amount of repetitions to converge to the least loss and reach the ideal weight matrix. The problem is, that kind of initialisation is prone to vanishing or exploding gradient problems.

General ways to make it initialise better weights:

A) ReLU activation function in the deep nets.

1. Generate a random sample of weights from a Gaussian distribution having mean 0 and a standard deviation of 1.
2. Multiply the sample with the square root of $(2/n_i)$. Where n_i is the number of input units for that layer.

B) Likewise, if you're using Tanh activation function :

1. Generate a random sample of weights from a Gaussian distribution having mean 0 and a standard deviation of 1.
2. Multiply the sample with the square root of $(1/n_i)$ where n_i is several input units for that layer.

Q14. What is optimiser in deep learning, and which one is the best?

Deep learning is an iterative process. With so many hyperparameters to tune or methods to try, it is important to be able to train models fast, to quickly complete the iterative cycle. This is the key to increase the speed and efficiency of a machine learning team.

Hence the importance of optimisation algorithms such as stochastic gradient descent, mini-batch gradient descent, gradient descent with momentum and the Adam optimiser.

Adam optimiser is the best one.

Given an algorithm $f(x)$, it helps in either minimisation or maximisation of the value of $f(x)$. In this context of deep learning, we use optimisation algorithms to train the neural network by optimising the cost function J . The cost function is defined as:

$$J(W, b) = \sum_{i=1}^m L(y'^i, y^i)$$

The value of the cost function J is the mean of the loss L between the predicted value y' and actual value y .

The value y'' is obtained during the forward propagation step and makes use of the Weights W and biases b of the network. With the help of optimisation algorithms, we minimise the value of Cost Function J by updating the values of trainable parameters W and b .

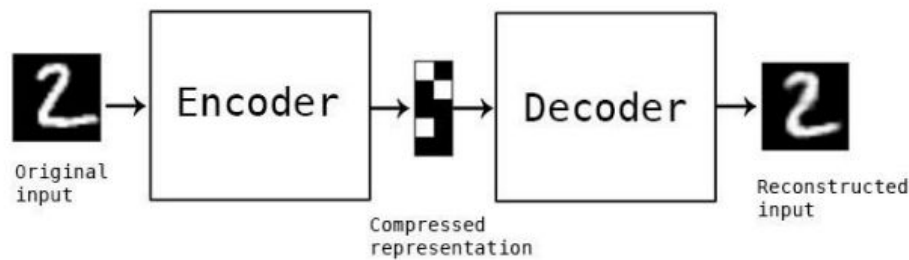
15. What are autoencoders?

An autoencoder, neural networks that have three layers: An input layer, a hidden layer which is also known as an encoding layer, and a decoding layer.

This network is trained to reconstruct its inputs, which forces the hidden layer to try to learn good representations of the inputs.

An autoencoder neural network is an unsupervised Machine-learning algorithm that applies backpropagation, setting the target values to be equal to the inputs. An autoencoder is trained to attempt to copy its input to its output. Internally, it has a hidden layer which

describes a code used to represent the input.



Autoencoder Components:

Autoencoders consists of 4 main parts:

- 1- Encoder: In this, the model learns how to reduce the input dimensions and compress the input data into an encoded representation.
- 2- Bottleneck: In this, the layer that contains the compressed representation of the input data. This is the lowest possible dimension of the input data.
- 3- Decoder: In this, the model learns how to reconstruct the data from the encode represented to be as close to the original inputs as possible.
- 4- Reconstruction Loss: In this method that measures how well the decoder is performing and how closed the output is related to the original input.

Types of Autoencoders:

1. Denoising auto encoder
2. Sparse autoencoder
3. Variational autoencoder (VAE)
4. Contractive autoencoder (CAE)

16. What is CNN?

This is the simple application of a filter to an input that results in inactivation. Repeated application of the same filter to input results in a map of activations called a feature map, indicating the locations and strength of a detected feature in input, such as an image.

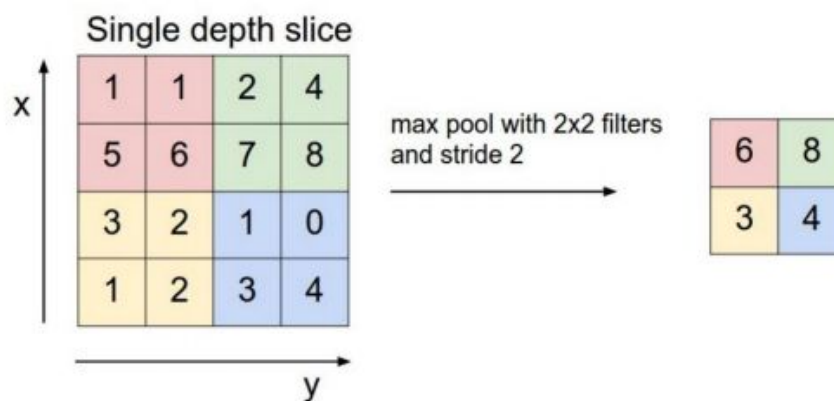
Convolutional layers are the major building blocks which are used in convolutional neural networks.

A covnets is the sequence of layers, and every layer transforms one volume to another through differentiable functions.

Different types of layers in CNN:

Let's take an example by running a covnets on of image of dimensions 32 x 32 x 3.

1. Input Layer: It holds the raw input of image with width 32, height 32 and depth 3.
2. Convolution Layer: It computes the output volume by computing dot products between all filters and image patches. Suppose we use a total of 12 filters for this layer we'll get output volume of dimension 32 x 32 x 12.
3. Activation Function Layer: This layer will apply the element-wise activation function to the output of the convolution layer. Some activation functions are RELU: $\max(0, x)$, Sigmoid: $1/(1+e^{-x})$, Tanh, Leaky RELU, etc. So, the volume remains unchanged. Hence output volume will have dimensions 32 x 32 x 12.
4. Pool Layer: This layer is periodically inserted within the convnets, and its main function is to reduce the size of volume which makes the computation fast, reduces memory and also prevents overfitting. Two common types of pooling layers are max pooling and average pooling. If we use a max pool with 2 x 2 filters and stride 2, the resultant volume will be of dimension 16x16x12.

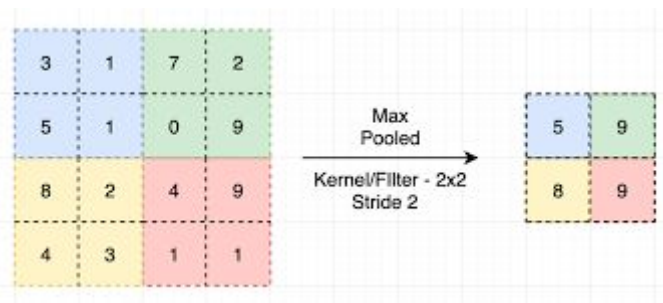


5. Fully-Connected Layer: This layer is a regular neural network layer that takes input from the previous layer and computes the class scores and outputs the 1-D array of size equal to the number of classes.

17. What is pooling, padding, filtering operations on CNN?

Pooling Layer

It is commonly used to periodically insert a Pooling layer in-between successive Conv layers in a ConvNet architecture. Its function is to progressively reduce the spatial size of the representation to reduce the number of parameters and computation in the network, and hence to also control overfitting. The Pooling Layer operates independently on every depth slice of the input and resizes it spatially, using the MAX operation.



The most common form is a pooling layer with filters of size 2x2 applied with a stride of 2 down samples every depth sliced in the input by two along both width and height, discarding 75% of the activations. Every MAX operation would, in this case, be taking a max over four numbers (little 2x2 region in some depth slice). The depth dimension remains unchanged.

18. What is the Evolution technique of CNN?

It all started with LeNet in 1998 and eventually, after nearly 15 years, led to ground-breaking models winning the ImageNet Large Scale Visual Recognition Challenge which includes AlexNet in 2012 to Google Net in 2014 to ResNet in 2015 to an ensemble of previous models in 2016. In the last two years, no significant progress has been made, and the new models are an ensemble of previous groundbreaking models.

LeNet in 1998

LeNet is a 7-level convolutional network by LeCun in 1998 that classifies digits and is used by several banks to recognise the hand-written numbers on cheques digitised in 32x32 pixel grayscale input images.

AlexNet in 2012

AlexNet: It is considered to be the first paper/ model, which rose the interest in CNNs when it won the ImageNet challenge in the year 2012. It is a deep CNN trained on ImageNet and outperformed all the entries that year.

VGG in 2014

VGG was submitted in the year 2013, and it became a runner up in the ImageNet contest in 2014. It is widely used as a simple architecture compared to AlexNet.

GoogleNet in 2014

In 2014, several great models were developed like VGG, but the winner of the ImageNet contest was GoogleNet. GoogLeNet proposed a module called the inception module that includes skipping connections in the network, forming a mini-module, and this module is repeated throughout the network.

ResNet in 2015

There are 152 layers in the Microsoft ResNet. The authors showed empirically that if you keep on adding layers, the error rate should keep on decreasing in contrast to “plain nets” we're adding a few layers resulted in higher training and test errors.