Deep Learning Interview Questions :

1. What are Neural Networks?

Human brain is made up of connected networks of neurons. Different parts of the brain are responsible for processing different pieces of information as these parts are arranged hierarchically in layers. As information comes into brain, at each level neurons processes the information and provides the insights to make decisions.

Like this, Neural networks tries to mimic the working of human brain and replicates into machine by forming the layers consisting of neurons. A basic neural network will have three different layers i.e., Input layer, Hidden layer(for processing the inputs) and a Output layer. Information flows from one layer to another layer just like human brain. This concept is called as Neural networks as there will be a network of interconnected neurons which are connected artificially. In Neural networks, Dendrites acts as Input layer, Nucleus acts as hidden layer for processing and Axon acts as Output layer.

2. What is Deep Learning ?

Deep Leaning is the subfield of AI & ML that is inspired from the structure of the human brain. It is the subset of Machine Learning that allows us to train the model using a set of inputs and then predicting the output. Deep Learning alogrithms attempt to draw similar conclusions as humans by continually analyzing the data.

Neural networks are the underlying concept which is used in Deep Learning. If we have more than one hidden layer in our network, it is "Deep Neural Net" and this is what underpins deep learning. As these two are closely connected in that one relies on the other to function. Without neural networks, there would be no deep learning.

3. DL vs ML :

Well, both ML and DL focuses on algorithms that can learn from training data and to make predictions on test data. One of the main differences between these two is, DL algos are particularly useful for processing complex data such as images,text and speech whereas ML won't work effectively in these types of data. One more differnce is , ML algos can achieve good results with small amounts of data, whereas DL algos require large amounts of data to achieve high accuracy in predictions.Additionally, DL algos are often more computationally expensive than traditional ML algos.

- Data dependancy is high in DL
- Hardware dependancy is high in DL
- Training time is high in DL
- Interpretability is poor in DL.

4. What are the applications of DL ?

There are many day to day applications that are using Deep Learning.

- Virtual Assistants & Chatbots
- Recommendation Engines & Text Sentiment analysis
- Image caption generation & Image Colourisation.
- Adding audio to muted videos.
- Healthcare: Deep learning has been used to develop predictive models for diagnosing diseases, identifying anomalies in medical images, and predicting treatment outcomes.

5. History of DL :

History of DL is very recent times and it is quite interesting as well.You might be thinking that the Deep Learning technique has recently getting attention so it would have begun some 20-30 years ago maybe, but let me tell you that it all started since 1943.

First let's see how a neuron works ?

If we touch the surface level of a biological neuron then it consists of mainly 3 parts, Nucleus, Dendrites, and Axons. The electrical signals are received by these dendrites connected to the nucleus where some processing is done by the nucleus itself and finally it sends out a message in the form of the electrical signal to the rest of the connected neurons through axons. This is the simplest explanation of the working of a biological neuron.

In 1943, first neuron idea brought up by Alan Turing. During cold war, Americal scientists were trying to translate Russian to English and had done lots of research on intelligent machines.

In 1957, Rosenblatt invented the perceptron, a mathematical model which got inspited from biological neuron and this is the true starting of AI, which was the big thing back then.

However, in 1969 reseracher Marvin Minsky published that perceptron can not work on linear functions.He further added, how much you train perceptron, it will not learn non linear function and this was actually true. **This is known as 'First AI winter'**

In 1986, Geoffrey Hinton the father of Deep Learning published a paper about Back Propagation and this was the landmark paper in Neural Networks. He mentioned, a single perceptron can't converge non-linear functions, but a build a neural network and add hidden layers, it can actually converge any mathematical/ non-linear function. This can be achieved through back propogation.This not only made the training of Artificial Neural Network possible but also created an AI Hype where people talked about it all day.

Till 1992 DL got its significance, but suddenly people realised that Neural networks will not work on all kind of data as there was not much labelled at that time and there was no sufficient computation power at that time,due to this this hype got died. The biggest drawback is machine learning algorithms like SVM, Random Forest, and GBDT evolved and became extremely popular from 1995 to 2009 and because of that US stopped funding. **This is also known as 'Second AI winter'**

Till 2006, Neural networks saw its downfall and got washed away. Again in 2006,Geoffrey Hinton published one more paper about deep neural nets. By 2006, we got enough labelled data and enough computation power. He also proved add as many as layers you want and make it a deep neural network and you can achieve anything. Because of this paper, new era got started "Deep Learning" but didn't get much recognition

2012 was the breakthrough year in Deep learning. Geoffrey Hinton developed a neural network model on top of GPU's in Imagenet competetion, the first neural network model outperformed almost every Machine Learning algorithm and that was insane.

This was the moment when the big tech giants like Google, Microsoft, Facebook, and others started seeing the potential in Deep Learning and started investing heavily in this technology.. Data + Hardware has powered Deep Learning.

6. Why DL is becoming popular now ?

- Datasets
- Hardware
- Frameworks
- Models Architecture
- Community

6.1) Deep Learning is data hungry, to train the DL model we will need lot of data. After 2012,usage of smartphones has been increased significantly and internet pricing was also reduced. On daily basis, we are generating lots of data which was not there in 1960's and 70's.

In last 10 years, companies like Google, Microsoft, Facebook spent lots of money in converting lots of unalabelled data to labelled data and made them open source which is available in public. Microsoft Coco (Image dataset),Youtube 8M (Video dataset), SQUAD wikipedia(Text dataset) are some of the public datasets. Without data there is no DL

6.2) One more biggest reason is Hardware. To process lots of data, we should also require high computation power. With CPU's we can't achieve much computation power, training time would be very high and can't process the data. To achieve this, we require GPU's which offers paraller processing technique. Instead of training in conventional CPU's, train your model in GPU and training time will gets reduced by 10 times due to parallel processing.

6.3) Third factor is development of libraries and frameworks. Training DL model is difficult work and it takes hell a lot of time to write the code from scratch.Frameworks like Tensorflow and PyTorch made this work simple.

In 2015, google released Tensorflow which was very powerful but difficult to use for normal people. Then they have developed keras library which works on top of tensorflow, it is like a intermediate library between user and tensorflow. In tensorflow 2.0, you will get keras in-built.

One more popular DL framework which was released in 2016, PyTorch. There isn't much difference between these two, industry people use tensorflow whereas research people uses Pytorch mostly.

They’ve become fairly easy to implement, especially with high-level open source libraries such as Keras, Pytorch, and TensorFlow.

6.4) Fourth factor why DL is becoming more popular is Deep Learning architecture. Depending on the problem, we can create different kinds of architectures by changing the parameters and nodes in the layers. Likethis, researchers already created some architectures on major problems which are ready to use. We can download the architecture and directly apply to our problem which is known as Transfer Learning, we will discuss about this later.Some of the few existing architectures :

- Image Classification : ResNET
- Text Classification : BERT
- Object Detection : YOLO
- Sppech generation : WaveNET

6.5) And the last reason is Community. In my opinion, this factor plays crucial role. People like researchers, DL engineers, competitors and communities like kaggle all made DL a powerful technology.

7. What is ImageNet ?

ImageNet is visual database of images contains 15 million images which are categorized into 20000 different classes. In 2010, Imagenet challenge got started "ImageNet Large Scale Visual Recognition Challenge" (ILSVRC).The competition involves training ML and DL algorithms to recognize and classify objects in the ImageNet dataset of millions of images. In 2012, this competition got widespread attention as first Deep learning model participated in this challange known as "AlexNet" that outperformed every Machine Learning at that time at an error rate of 16%. You can learn about ImageNet challenge from the below article. Please go through it.

https://medium.com/@prudhvi.gnv/imagsenet-challenge-advancement-in-deep-learning-and-computer-vision-124fd33cb948

8. Collaborative Filtering vs Content based filtering.

Collaborative filtering is a technique that looks at the past interactions between users and items to recommend new items to users. For example, if User A and User B have both watched and enjoyed the same movie in the past, then collaborative filtering might recommend a new movie to User A based on the fact that User B also enjoyed it.

On the other hand, content-based filtering looks at the attributes of items and matches them to users with similar preferences. For example, if User A has watched and enjoyed a lot of romantic comedies in the past, then content-based filtering might recommend a new romantic comedy to User A based on the fact that it has similar attributes to other romantic comedies that User A has enjoyed.

Both techniques are used by major companies like Netflix and Amazon to provide personalized customer experiences.

9. What is Forward Propogation & Backward Propogation ?

Forward propagation refers to the process of inputting data into a neural network to generate an output by moving in a forward direction through hidden layers and applying activation functions based on weighted sums.

On the other hand, backpropagation is a method of calculating the gradient of neural network parameters and involves traversing the network in reverse order, from the output to the input layer using the chain rule. It is used for training and involves improving prediction accuracy through backwards propagation calculated derivatives. The accuracy of a node is expressed through a loss function or error rate, and backpropagation calculates the slope of the loss function of other weights in the network.

10. What are weights & bias in DL ?

Weights are the real values that are attached with each input/feature and they convey the importance of that corresponding feature in predicting the final output.

Bias is used for shifting the activation function towards left or right, you can compare this to y-intercept in the line equation.

https://towardsdatascience.com/whats-the-role-of-weights-and-bias-in-a-neural-network-4cf7e9888a0f

11. What are Acitvation functions ?

Activation functions are the mathematical functions in a neural network which are applied to output of each neuron. It computes the weighted sum of inputs and decides a particular neuron to be activated or not.

Why Activation functions ?

Our brain is fed with a lot of information simultaneously, it tries hard to understand and classify the information into “useful” and “not-so-useful” information. We need a similar mechanism for classifying incoming information as “useful” or “less-useful” in case of Neural Networks.This is where activation functions come into picture. The activation functions help the network use the important information and suppress the irrelevant data points i.e., noise.

Also activation functions converts linear inputs signals to non linear output signals which introduces non linearity in the network. If we don't use activation functions, our neural network is nothing but a linear model , which is not so powerful.

Assumptions :

- An ideal activation function should be non-linear.
- It should be differentiable and computationally inexpensive.
- It should be non-saturating. (It means we should not squeeze the input to sepcific range.)

Types of Activation functions :

- Step
- Sigmoid
- Tanh
- ReLU
- Parametric Relu
- ELU
- SoftMax

How to select Activation functions :

Well selecting activation functions is quite easy task. If you want to apply activation function in any of the hidden layers, blindly go with Relu. With Relu we can avoid vanishing gradient problem which most of the activation functions will not do. In output layer go with sigmoid function as sigmoid function maps the value between 0 and 1. Always keep in mind that ReLU function should only be used in the hidden layers. Selecting an appropriate activation function is crucial because it can significantly impact the performance of a neural network.

12. What are Loss functions ?

In ML/ DL, loss functions are the functions which calculates the error between the actual output and our predcited output.In simple terms, Loss function is a mathematical function gives the output as an integer value and it represents the error in the model.

Why Loss functions ?

If you want to improve the performance of your model, you need to measure it. Here loss functions comes into picture to evaluate how well our algorithm is modelling to the dataset. You can't improve the model unless you measure it, so loss functions play key role in model training. If the value of the

loss function is lower, then it is a good model else, we have to change the parameters of the model and minimize the loss.

Types of Loss functions :

There are many loss functions and here I have listed few of them.

- Mean Squared Error
- Mean Absolute Error
- Binary Cross Entropy/ Logloss
- Categorical Cross Entropy
- Hingeloss

How to select Loss functions:

Well selecting loss functions completely depends on the algorithm that you are working. If you are trying to predict the prices of a house and you are using Linear Regression, then go with MSE loss function. If you are working on classification problem and you want to classify the output to two classes, then go with Binary Cross Entropy. If you are working on classification problem and you want to classify the output to more than two classes, then go with Categorical Cross Entropy. If you are building a simple NN perceptron then go with Hinge loss.

Many of us get confused between Loss function and Cost function. Generally, loss function is for single training input, whereas cost function is the average loss over entire training dataset.

13. What is Gradient Descent and what are its types ?

Gradient Descent is the most common optimization algorithm in *machine learning* and *deep learning*. It is a first-order optimization algorithm. This means it only takes into account the first derivative when performing the updates on the parameters. It is a way to minimize the loss function by updating the parameters. The size of the step we take on each iteration to reach the local minimum is determined by the learning rate α. Therefore, we follow the direction of the slope downhill until we reach a local minimum.

Batch Gradient Descent:

We take the entire training set, perform forward propagation and calculate the cost function. And then we update the parameters using the rate of change of this cost function with respect to the parameters.

Here, # No.of updates = # No.of epochs

Stochastic Gradient Descent :

If you use a single observation to calculate the cost function it is known as **Stochastic Gradient Descent.** We pass a single observation at a time, calculate the cost and update the parameters.

*Here, # No.of updates = # No.of epochs * No.of data points*

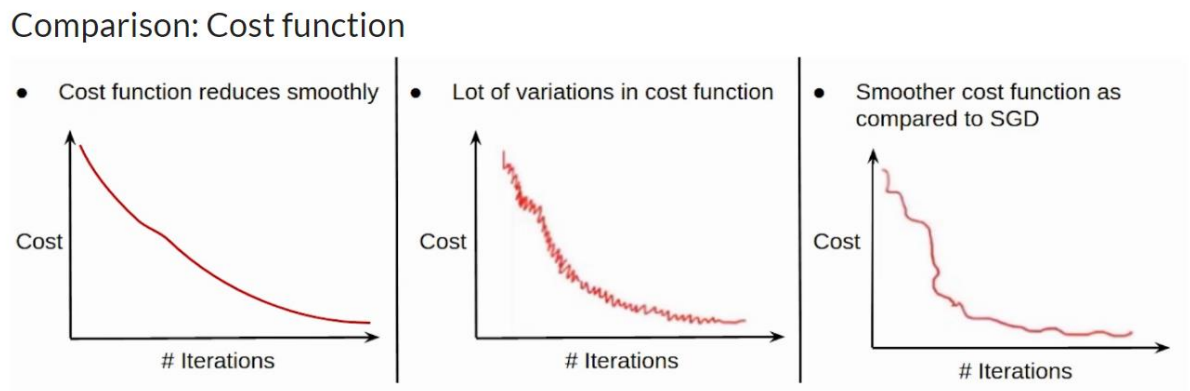
Mini-Batch Gradient Descent:

It takes a subset of the entire dataset to calculate the cost function. So if there are 'm' observations then the number of observations in each subset or mini-batches will be more than 1 and less than 'm'.

Here, # No.of updates = # No.of batches

Note : Here, batch size is always provided in multiple of 2 due to effective usage of RAM.

- Computation time : BGD > MGBD > SGD (BGD takes less time to compute and is very fast).
- Computation cost : BGD > MGBD > SGD (BGD computation cost is very high.)
- Convergence : SGD > MGBD > BGD (SGD is faster to converge as we have lots of updates).
- Noise : SGD > MGBD > BGD



In BGD, cost function reduces smoothly, whereas in SGD lot of variations are there.

14. Explain about Vanishing gradient & Exploding gradient problems.

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 layers unchanged. As a result, the gradient descent never converges to the optimum. This is known as the ***vanishing gradient*** problem.

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 exploding gradient.

How to recognize ?

- Focus on loss. If there is no change in loss,it may be a problem.
- Plot a graph between weights and epochs. If values are consistent, it could be problem.

How to handle?

- Reduce model complexity or reduce no.of layers.
- Use ReLU activation layer instead of Sigmoid and Tanh.
- Use proper weight initialization techniques.
- Use Batch Normalization and use gradient clipping.

15. How to improve the performance of a Neural network ?

We can improve the neural network performance by following ways.

1. Fine tuning the performance of Neural Network.

    - No.of hidden layers.
    - No.of neurons in each layer.
    - No.of epochs.
    - Learning rate & Batch size
    - Activation functions and optimizers.

2. By solving below problems.

    Overfitting

    - Dropout layers
    - Early stopping
    - Regularization
    - Data Augmentation

    Vanishing/ Exploding gradient

    - Use wt initialization techniques
    - Activation functions.
    - Gradient clipping.

    Normalization

    - Batch Normalization
    - Normalizing inputs
    - Normalizing activation funcs.

    Optimizers

    - SGD with momentum.
    - Adagrad
    - RMS Prop
    - Adam.

16. What is Gradient Clipping ?

Gradient clipping is a technique used to prevent the gradients from becoming too large during training, which can lead to unstable training and convergence issues. In gradient clipping, we clip the gradients to a maximum value so that they never exceed that value. This technique is particularly useful when training recurrent neural networks (RNNs) that are prone to exploding gradients.

```
# Define the optimizer with gradient clipping
sgd = SGD(lr=0.01, clipnorm=1.0, momentum=0.9)
```

17. What is Early Stopping ?

Early stopping is a technique used in neural networks to prevent overfitting or underfitting of the model during training. It allows you to specify an arbitrary number of epochs and stops training oncthe model's performance stops improving on a validation dataset.

In the below example, the EarlyStopping callback is used to monitor the validation loss and stop training if it does not improve for 10 epochs. The ModelCheckpoint callback is used to save the best model observed during training. The model is then compiled and fitted with the callbacks passed as a list.

```
# define early stopping callback
early_stop = EarlyStopping(monitor='val_loss', patience=10, verbose=1, mode='min')

# define model checkpoint callback
model_checkpoint = ModelCheckpoint('best_model.h5', monitor='val_loss', save_best_only=True, mode='min'
```

18. What is Batch Normalization ?

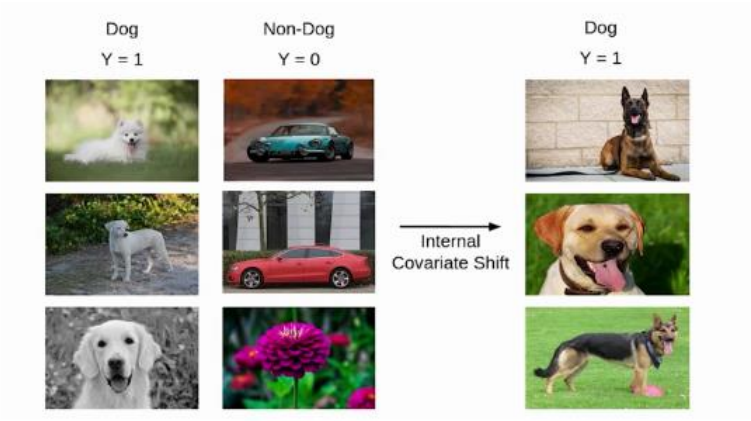
Normalization is a data pre-processing tool used to bring the numerical data to a common scale without distorting its shape. Generally, when we input the data to a machine or deep learning algorithm we tend to change the values to a balanced scale. The reason we normalize is partly to ensure that our model can generalize appropriately.

Although, our input X was normalized with time the output will no longer be on the same scale. As the data go through multiple layers of the neural network and L activation functions are applied, it leads to an internal co-variate shift in the data.

It is a two-step process. First, the input is normalized, and later rescaling and offsetting is performed.

Internal covariate shift:

Suppose we are training an image classification model, that classifies the images into Dog or Not Dog. Let's say we have the images of white dogs only, these images will have certain distribution as well. Using these images model will update its parameters. later, if we get a new set of images, consisting of non-white dogs. These new images will have a slightly different distribution from the previous images. Now the model will change its parameters according to these new images. Hence the distribution of the hidden activation will also change. This change in hidden activation is known as an internal covariate shift.



Advantages :

- Speed up the training
- Handles internal covariate shift.
- Acts as regularization.
- Reduce weight initialization impact.

19. What are Optimizers ?

Optimizers are algorithms or methods used to change the attributes of your neural network such as weights and learning rate in order to reduce the losses. Optimizers in deep learning are algorithms used to adjust the parameters of a neural network, such as weights and learning rate, to minimize an error function or maximize production efficiency. These mathematical functions help change the weights and learning rates of neural networks to reduce losses.

Drawbacks of Gradient descent :

- Weights are changed after calculating gradient on the whole dataset. So, if the dataset is too large than this may take years to converge to the minima.
- Choosing an optimum value of the learning rate. If the learning rate is too small than gradient descent may take ages to converge.
- Have a constant learning rate for all the parameters. There may be some parameters which we may not want to change at the same rate.
- May get trapped at local minima.
- Requires large memory to calculate gradient on the whole dataset.

20. What is Dense layer ?

In neural networks, the Dense layer is a type of layer that adds a fully connected layer of artificial neurons in the model. Each neuron in the Dense layer receives input from every neuron of the previous layer, and outputs to every neuron of the next layer. The number of neurons in the Dense layer and the activation function used can be adjusted to achieve the desired output. The Dense layer is commonly used in deep learning models for tasks such as classification and regression.

21. Explain the difference between a shallow network and a deep network.

A shallow network has only one hidden layer, while a deep network has multiple hidden layers. Deep networks are capable of learning more complex representations of data as they can extract multiple levels of features from the input data. This allows deep networks to achieve higher accuracy and better performance than shallow networks. However, deeper networks require more computational resources and longer training times. Additionally, deep networks can suffer from vanishing gradients or overfitting, which can affect their performance.

22. What is Perceptron ?

Perceptron is the first neural network model which was introduced in 1957. It is a type of artificial neuron that takes inputs, computes a weighted sum, and applies an activation function to produce an output.The perceptron was inspired by the biological neuron, which receives inputs from other neurons. Main difference between a neuron and a perceptron is, neuron is the biological model whereas perceptron is the mathematical model/ algorithm.

How perceptron works ?

Perceptron is a single layer neural network, means there will be one input layer, one hidden layer and one output layer. Perceptron consists of mainyl four parts : Inputs, Weights & bias & Weighted sum and Acitavation function.

Why weights and bias ?

- Weights play an important role in changing the slope of the line that separates two or more classes of data points.
- Weights tell the iportance of a feature in predicting the target value, also it tells about the relationship between a feature and a target value.
- Bias is used for shifting the activation function towards the left of right.

Working of a perceptron:

Perceptron works on the below simple steps.

- All the inputs are multiplied with their respective weights.
- Add all the muliplied weights and sum them up i.e., Weighted sum.
- Apply activation function to the weighted sum to get the value between 0 and 1.

Drawbacks of perceptron :

- Basically, there are two types of perceptron. One is Single layer neural network and other is Multi layer neural network. Single layer nn is perceptron, and multi layer neural network is called as Multi layer perceptron nothing but Neural Networks.
- Problem with this perceptron is it didn't work well on non linear data. One more problem is as perceptron is having single layer, it will not work well for complex datasets. Single layer perceptron is just like a Linear Regression machine learning model which is not able to solve non linear relationships between inputs and outputs.
- Despite its limitations, the perceptron remains an important milestone in the history of artificial intelligence and machine learning.
- The Neural Networks work the same way as the perceptron. So, if you want to know how neural network works, learn how perceptron works.

23. ANN vs CNN vs RNN

ANN:

An artificial neural network is the simplest NN, as it passes information through multiple neurons until it generates an output. It is also known as a feed-forward neural network because the inputs are processed only in a forward direction.

Challenges :

- Doesn't work well on Images.
- If we use ANN in images, we may loss important features like spatial arrangement of pixels.
- We need to do lot of unnecessary computaion for image and textual data.
- Here, we discard sequence information.

CNN:

Convolution Neural Network takes an image as input and learns the various features of the image through filters, also known as kernels. This allows the model to learn and catch up with the different objects present in the picture. For example, CNN will learn the features of apple that differ from orange so that when we provide apple input, it can differentiate between the two. They are good for processing grid like topology or images.

Challenges :

- Computationally bit expensive.
- Another challenge that CNNs face is vanishing gradients.

RNN:

A recurrent neural network (RNN) is a type of artificial neural network which uses sequential data or time series data. These deep learning algorithms are commonly used for ordinal or temporal problems, such as language translation, natural language processing (nlp), speech recognition.

26. What is Data Augmentation ?

A common problem with machine learning and deep learning models is overfitting. This means that the model accuracy is very high on the training dataset but fails to do so on the unknown data. To address the challenge of overfitting, we can increase the size of the dataset, i.e., exposing the model to new data for a better generalization. Additional data is not always available, can be challenging to gather, and is time-consuming and expensive. In such a situation, data scientists use a process called Data Augmentation. Using data augmentation, we can expand the size of the real data by several augmentation techniques. This process is beneficial in projects with smaller datasets and models that experience overfitting. Data augmentation helps in improving model accuracy.

Machine learning and deep learning models can use data augmentation. It is possible to augment image, text, audio, and video type of data. Several deep learning frameworks – Keras, Tensorflow, Pytorch, and more have built-in functionality for augmentation, while many open-source python libraries are developed especially for augmentation.

Augmentation techniques for different data types:

- **Image**: The augmentation techniques for images allow scaling, flipping, rotating, cropping, change in brightness/contrast/sharpness/blur, color filtering, and many more.
- **Text**: The augmentation techniques for text support NLP tasks by word/sentence shuffling, word replacement with synonyms, paraphrasing, and so on.
- **Audio** and **Video** techniques can manipulate the data by introducing noise, changing speed, etc.

27. What are CNN pre-trained models ?

- 1980 : Neocognitron
- 1989 : Lenet-5
- 2012 : Alexnet
- 2013 : ZFnet
- 2014 : VGGnet
- 2015 : Googlenet
- 2016 : RESnet

28. What is Transfer learning ?

Transfer learning is a technique in machine learning where a pre-trained model is trained on one task and is re-purposed on a second related task. Instead of starting the learning process from scratch, the pre-trained model is used as a starting point and adapted to the new task. This approach can save time and computational resources, and can be particularly useful when the new task has limited labeled data. Transfer learning can be applied to a variety of machine learning tasks, including image classification, natural language processing, and speech recognition.

Ways of doing tranfer learning :

- Feature Extraction
- Fine tuning

<u>Advantages :</u>

- **Better initial model**: In other methods of learning, you must create a model from scratch. Transfer learning is a better starting point because it allows us to perform tasks at a higher level without having to know the details of the starting model.
- **Higher learning rate:** Because the problem has already been taught for a similar task, transfer learning allows for a faster learning rate during training.
- **Higher accuracy after training:** Transfer learning allows a deep learning model to converge at a higher performance level, resulting in more accurate output, thanks to a better starting point and higher learning rate.

29. Explain CNN architectures.

Lenet5:

LeNet5 is a 5-level convolutional network by LeCun in 1998 that classifies digits and used by several banks to recognise the hand-written numbers on cheques digitised in 32x32 pixel greyscale input images.

*It has 5 layers. 3 conv(6 5*5 filters, 16 5*5 filters, 120 5*5 filters) + 2 FC layer(84 neurons & 10 classes)*

Layer	# filters / neurons	Filter size	Stride	Size of feature map	Activation function
Input	-	-	-	32 X 32 X 1	
Conv 1	6	5 * 5	1	28 X 28 X 6	tanh
Avg. pooling 1		2 * 2	2	14 X 14 X 6	
Conv 2	16	5 * 5	1	10 X 10 X 16	tanh
Avg. pooling 2		2 * 2	2	5 X 5 X 16	
Conv 3	120	5 * 5	1	120	tanh
Fully Connected 1	-	-	-	84	tanh
Fully Connected 2	-	-	-	10	Softmax

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. **They trained their network on 1.2 million high-resolution images into 1000 different classes with 60 million parameters and 650,000 neurons.** The training was done on two GPUs with split layer concept because GPUs were a little bit slow at that time.

AlexNet contains eight layers with weights, first five are convolutional, and the remaining three are fully connected. The output of last fully-connected layer is fed to a 1000-way softmax which produces a distribution over the 1000 class labels. Relu is applied after the very convolutional and the fully connected layer. Dropout is applied before the first and second fully connected year. The network has the 62.3 million parameters and needs 1.1 billion computation units in a forward pass. We can also see convolution layers, which accounts for 6% of all the parameters, consumes 95% of the computation.

*It has 8 layers. 5 conv(96 11*11 filters, 256 5*5 filters, 384 3*3 filters,384 3*3 filters, 256 3*3 filters) + 3 FC layer(4096 neurons, 4096 neurons & 1000 classes)*

Layer	# filters / neurons	Filter size	Stride	Padding	Size of feature map	Activation function
Input	-	-	-	-	227 x 227 x 3	-
Conv 1	96	11 x 11	4	-	55 x 55 x 96	ReLU
Max Pool 1	-	3 x 3	2	-	27 x 27 x 96	-
Conv 2	256	5 x 5	1	2	27 x 27 x 256	ReLU
Max Pool 2	-	3 x 3	2	-	13 x 13 x 256	-
Conv 3	384	3 x 3	1	1	13 x 13 x 384	ReLU
Conv 4	384	3 x 3	1	1	13 x 13 x 384	ReLU
Conv 5	256	3 x 3	1	1	13 x 13 x 256	ReLU
Max Pool 3	-	3 x 3	2	-	6 x 6 x 256	-
Dropout 1	rate = 0.5	-	-	-	6 x 6 x 256	-

Layer	# filters / neurons	Filter size	Stride	Padding	Size of feature map	Activation function
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
Dropout 1	rate = 0.5	-	-	-	6 x 6 x 256	-
Fully Connected 1	-	-	-	-	4096	ReLU
Dropout 2	rate = 0.5	-	-	-	4096	-
Fully Connected 2	-	-	-	-	4096	ReLU
Fully Connected 3	-	-	-	-	1000	Softmax

VGGnet:

- VGG16 has a total of 16 layers that has some weights.
- Only Convolution and pooling layers are used.
- Always uses a 3 x 3 Kernel for convolution. 20
- 2×2 size of the max pool.
- 138 million parameters.
- Trained on ImageNet data.
- It has an accuracy of 92.7%.
- Another version that is VGG 19, has a total of 19 layers with weights.
- It is a very good Deep learning architecture for benchmarking on any particular task.
- The pre-trained networks for VGG is made open-source, so it can be commonly used out of the box for various types of applications.

ResNet:

At the ILSVRC 2015, so-called Residual Neural Network (ResNet) by the Kaiming He et al introduced the anovel architecture with “skip connections” and features heavy batch normalisation. Such skip connections are also known as the gated units or gated recurrent units and have the strong similarity to recent successful elements applied in RNNs. Thanks to this technique as they were able to train the NN with 152 layers while still having lower complexity than the VGGNet. It achieves the top-5 error rate of 3.57%, which beats human-level performance on this dataset.

Year	CNN	Developed by	Place	Top-5 error rate	No. of parameters
1998	LeNet(8)	Yann LeCun et al			60 thousand
2012	AlexNet(7)	Alex Krizhevsky, Geoffrey Hinton, Ilya Sutskever	1st	15.3%	60 million
2013	ZFNet()	Matthew Zeiler and Rob Fergus	1st	14.8%	
2014	GoogLeNet(19)	Google	1st	6.67%	4 million
2014	VGG Net(16)	Simonyan, Zisserman	2nd	7.3%	138 million
2015	ResNet(152)	Kaiming He	1st	3.6%	