UDS21301J - INTRODUCTION TO DEEP LEARNING
UNIT 3

1.What are the criteria to choose DL?

The criteria for choosing DL are:

- **Understanding the model:** When considering the problem, we are solving, the best way to translate in terms of deep learning is to consider the input data in place. For ex: Is the data text or image? If it is an image, then deep learning models come into the picture, if it's a text-based problem, it is an NLP approach

- **Accuracy**: Accuracy is a factor that is, most of the time, an important criterion while selecting a deep learning model for your implementation. But one should not consider the accuracy blindly. We must note that other things also must be considered when accuracy is chosen as a metric. There are plenty of accuracy metrics available for the problem. In addition, each problem from a particular industry has its own set of metrics. Therefore, it is important to select the right metrics

- **Sufficient knowledge of your data**: Based on the fact that the amount of data is with you or the amount of data you are going to gather, finding the right deep learning model of your problem solution will differ. Building the model right from the start to finish may not be only approach. If there is a large amount of data with you, it is obvious that you will go for the model that can get trained from scratch. But the convergence of the deep learning models in a complex task. With a supportive community around you can make your life simple

- **Accuracy, speed and size**: The speed vs accuracy tradeoff is a major consideration in selecting deep learning models for your implementation. Hence there are various architectures for matching different use cases for the applications

2. What are pre trained DL models?

A pre trained deep learning model is built or developed by somebody else who is trying to solve a similar problem as you are solving. When utilizing a pre trained model, you do not have to build the model from scratch; Instead, just use the model for the model for other problem for which it was built.

Pretrained networks have different characteristics that matter when choosing a network to apply to your problem. The most important characteristics are network accuracy, speed, and size. Choosing a network is generally a tradeoff between these characteristics. Use the plot below to

compare the ImageNet validation accuracy with the time required to make a prediction using the network.

For example, when working with a self-driving car, typically, a lot of time can be spent on image recognition or you can use an already trained model from google for the same task and make your work easy. The pre-trained models may not be able to deliver the accuracy that you are expecting but it typically saves a lot of time and effort that you have to put in.

3.When to use pre trained DL model?

- Building our custom models involves many steps, from collecting the training data to performing feature extraction and creating a user interface. It also needs technical people like data engineers, machine learning engineers and others to build an efficient model. In the areas of cybersecurity, healthcare and self-driving autonomous vehicle using your custom model can be a good idea but in the other areas, it may be a lengthy process that takes away the actual focus area of work

- The pre-trained models come to the rescue. The pre-trained models usually come from the vendors and are used by the API's. The vendor's job is to train the model with the training data perform the feature extraction and others. One can use the vendors API's and a sample dataset on the pre-trained model.

- The pre-trained model does not require much effort in setting up the infrastructure and can be easily integrated into your existing applications. The pre-trained models are cheaper and can be accessed on the cloud infrastructure using API's.

4. What are the benefits of using pre -trained DL model

When the model that we build needs to be trained from the beginning, it consumes a lot of time and requires a large amount of data. Instead, a pre-trained model that takes care of the feature extraction could be used. A pre-trained model will extract all the features like edges, and circles in case of image classification. When a pre-trained model is used, it can converge very quickly as its weights are already optimized, and there is no requirement of a large amount of data for training to learn from your data.

- The pre-trained models are general and, therefore, can prove handy for a wide audience

- In the pre-trained model, the training time is non-significant on the average user's system.

- In the pre-trained model, most of the time is spent on the extraction, formatting and preprocessing.

- Using the same dataset, it was trained on makes the outcome of the pre-trained model comparable and repeatable.

5. What are transformers in Deep Learning?

- The transformer model in the deep learning arena is an encoder-decoder-based model that can learn the context of the text in the form of paragraphs and sentences, therefore tracking out the relationships in a sequential format. The Transformer models use a handful of mathematical concepts called the attention that help detect the acceptable methods of even the farther data elements in a series Influence.

- The transformer models have the ability to translate human text and speeches in near real time scenarios. This can help people with hearing difficulties when attending school gatherings or meetings. In addition, the transformer models benefit the researchers and scholars in understanding the genes in the human DNA and amino acids in the proteins, which can expedite the drug discovery process.

- The transformers find the patterns and outliers for fraud detection and prevention, thus streamlining the manufacturing processes. It can also be used in making drug recommendations in the healthcare sector. We use transformers nearly every day as we search through google or MS Bing.

6. Pre -trained DL Model in NLP?

BERT:

BERT, also known as Bidirectional Encoder Representations from Transformers, is a very popular NLP model used in recent times. In addition, a very popular pre-trained model that works on the transformer architecture is available. Bidirectional Encoder Representations from Transformers are trained on a massive amount of corpora with unlabelled text. The Pre training of the BERT model attributes to its success. When a model is trained on huge text corpora, it learns more about the context and how a natural language works.

The BERT model is a Bl-directional model, meaning it can learn from both the left and right directions of a token's context, which is particularly important when we want the model to understand the context very well. Below is an example

Code BERT:

The Code BERT is a pre-trained model made up of bidirectional neural network architecture. With the help of understanding the relationship between natural and programming language, the Code BERT can help in the tasks like code search and documentation generation. This pre trained model is evaluated on the natural language programming language tasks by fine tuning the parameters. Therefore, the model results in good performance

GPT:

The Word "GPT expands to "Generative Pre-training. This model utilizes the concept of the multi-layer Transformer Decoder as the feature extractor. When using the transformers model, we see that it does not consider performing the feature extraction in one direction, i.e. from left to the right; instead, it observes the following word while predicting the next word.

The GPT model utilizes the mask multi-head attention concept to overcome the LSTM challenges; hence, the Transformer only considers one part of the input text called the one-way Transformer. GPT-2 is a successor of the GPT model that is trained on more than 1.8 billion parameters and many web pages. The objective of GPT-2 is quite simple, predicting the next word with the presence of all the other prior words in the context. The heterogeneous capabilities of the datasets achieve this objective of containing demonstrations of various jobs across different verticals. The GPT-2 model is 10 times more powerful than its successor GPT-2 as it is pre-trained 10 times more than the GPT Model.

XLNET:

The conventional auto-regressive language models are more applied for the natural language generation workloads, which cannot constitute the two-way context. However, on the other hand, the self-encoding models function like the BERT model, which comprises the independence assumptions and does not predict the correlation. Moreover, in this model, the MLM pre-training target setting does not enable the pre-training process to be smooth, and the model's performance degrades for the text generation activities.

XLNET takes advantage of both the autoregressive and the self-encoding models. The model. rearranges the input sequence order and predicts the result. However, the predicted outcome is still in the same order as the original order, and the context is taken in the shuffled order.

6. Artificial neural network

Neuron:

- Neurons are the building block of Artificial Neural Networks like the biological neuron of the human brain cells.

Weight:

• The weights are responsible for making the bonding between the network nodes strong. The weight is the deciding factor for the network's input, influencing the output. The consequences are randomly initialized in the network.

Activation Function:

- The artificial neural networks learn the complex patterns from the data with the Activation function.

- Compared to a biological neuron, it is the activation function that decides to fire the neuron to the next layer.

- The working of the Artificial Neural Network is the same way, it takes the output of the previous neuron and converts it into a form that can be taken to the next neuron, The activation function's job is to induce the process of non-linearity in the network so the web understands the complex pattern in the dataset.

Types of Activation Function:
Sigmoid activation Function:

- It is one of the most widely used non-linear activation functions, Sigmoid transforms the values between the range 0 and 1.

- Sigmold function is a non-linear activation function meaning that the neurons with the sigmoid function result in the output being non-linear.

Linear Function:

- The linear activation function's equation is the same as that of the straight-line equation. Irrespective of the no of layers and with linear nature, the output layer is a linear function of the first layer.

Tanh Activation Function:

The tanh activation is a modified version of the sigmoid function. However, it works similar to it and derived from each other.

RELU Activation Function:

RELU is a most commonly used activation function that gives a value of 0 for negative input, and for positive values, it returns that value.

Forward Propagation:

Forward Propagation is a technique in the artificial neural network that means movement of the process forward, and Propagation implies spreading. In the forward Propagation, we move in one direction, l.e. from the input to the output in the forward direction

Backward Propagation:

- Backpropagation is the cream of training an artificial neural network.

- In Backpropagation, the fine-tuning of the weights of a neural net is based on the error rate captured (ie. iteration). Tuning the weights will reduce error rates, increasing the model reliability and generalization.

Cost Function:

- The cost function can be described as a measure that gives an error between the predicted value and the actual of the model.

- The cost function is an output predicted by the network for a pint x with parameters 0.

Gradient Descent

- In the backpropagation phase, it is necessary to resample the gradient of the network's parameters in the reverse direction, updating until the network is optimized. This whole process is called Gradient Descent.

- The Gradient Descent minimizes the loss function or the cost function.

- The activation function lets us know the change required by considering the partial derivative of the process.

Stochastic Gradient Descent

- In the backpropagation phase, it is necessary to resample the gradient of the network's parameters in the reverse direction, updating until the network is optimized. This whole process is called Gradient Descent.

- The Stochastic Gradient Descent is an optimization technique where random data samples are considered instead of the whole set for each iteration. The random samples are taken in batches to calculate the gradient descent. The entire dataset is viewed in a typical gradient algorithm, and the minima are calculated in a less noisy environment. When taking a large dataset, it could be problematic. In Stochastic Gradient Descent, the samples are taken in batches.

Learning Rate

The model parameters are updated continuously like weights during the model training process, known as "Step Size" or "Learning Rate".

The learning rate is a configurable hyperparameter used in the model training process. having a value between 0 and 1.

• A model's learning rate should not be significant; if the case, then it can cause the model to converge quickly, resulting in a suboptimal solution. On the other hand, It cannot be small, making the process stuck.

It is advisable to use small learning rates while fine-tuning the ConvNet model. This is because we expect that the model is well- trained and that it may not be good to distort the weights too fast and too much as that could result in a very poor model.

Batches, Epochs and Iteration

Batches: The number of data samples processed before the model is updated is called Batch Size.

Epochs: Epochs can be defined as one complete cycle of a neural network or the trained model. An epoch consists of one or more batches.

Iterations: The Number of Batches to complete an epoch is called an Iteration, say, for example, we have 5000 data samples. The samples can be divided into five batches of 1000 samples each of 1 epoch