Sinhgad Institute of Technology & Science, Narhe
Pune-411041

Department of Computer Engineering

LAB MANUAL

Laboratory Practice V

(410255)

Semester-VIII

Companion Courses:

High Performance Computing (410250)

Deep Learning (410251)

INDEX

Name of the Subject/Course: **410255 - Laboratory Practice V**

Class: **BE**

Assignment No.5

Aim: Linear regression by using Deep Neural network: Implement Boston housing price prediction problem by linear regression using Deep Neural network. Use Boston House price prediction dataset.

Objective: Students will learn

1. The concept of linear regression.
2. The concept of deep neural network.

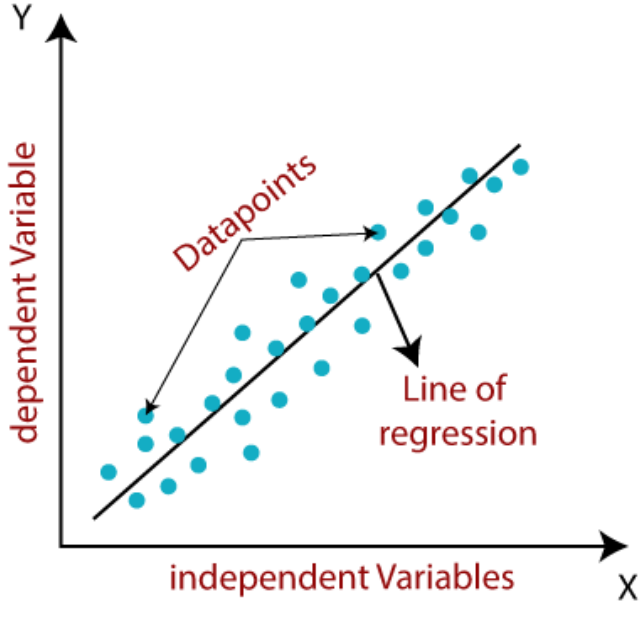
Theory:

Linear Regression:

Linear regression is one of the easiest and most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables such as sales, salary, age, product price, etc.

Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called as linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable.

The linear regression model provides a sloped straight line representing the relationship between the variables. Consider the below image:



Mathematically, we can represent a linear regression as:

$y= a_0+a_1x+ \varepsilon$

Here,

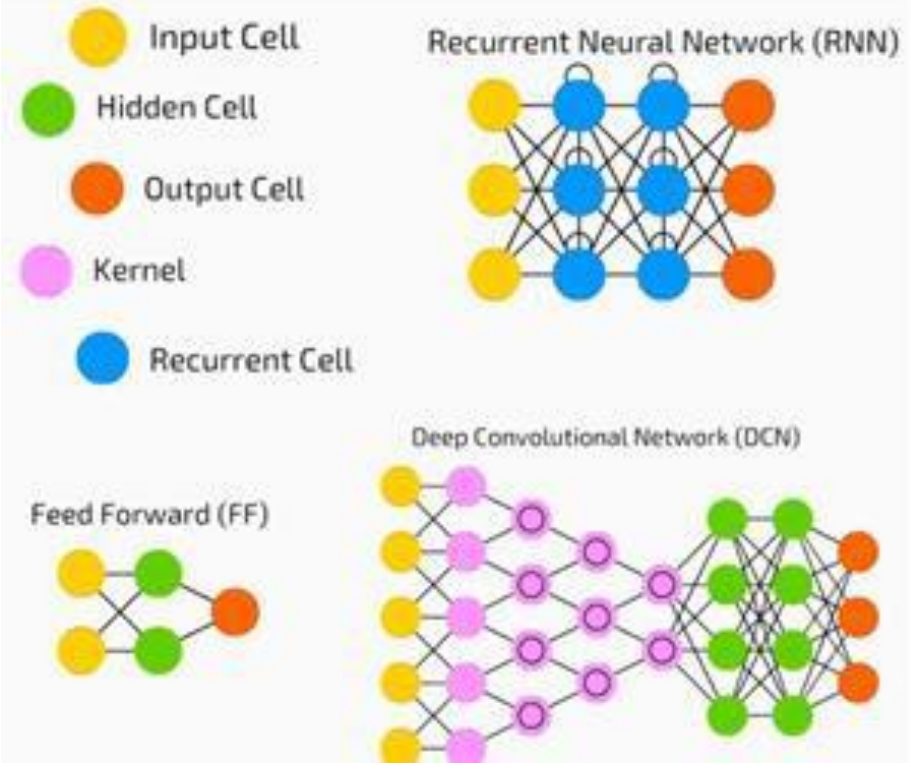
Y= Dependent Variable (Target Variable)
X= Independent Variable (predictor Variable)
a0= intercept of the line (Gives an additional degree of freedom)
a1 = Linear regression coefficient (scale factor to each input value).
ε = random error
The values for x and y variables are training datasets for Linear Regression model representation.

Deep Neural Network:

A deep neural network (DNN) is an ANN with multiple hidden layers between the input and output layers. Similar to shallow ANNs, DNNs can model complex non-linear relationships.

The main purpose of a neural network is to receive a set of inputs, perform progressively complex calculations on them, and give output to solve real world problems like classification. We restrict ourselves to feed forward neural networks.

We have an input, an output, and a flow of sequential data in a deep network.



Neural networks are widely used in supervised learning and reinforcement learning problems. These networks are based on a set of layers connected to each other.

In deep learning, the number of hidden layers, mostly non-linear, can be large; say about 1000 layers.

DL models produce much better results than normal ML networks.

We mostly use the gradient descent method for optimizing the network and minimising the loss function.

We can use the **Imagenet**, a repository of millions of digital images to classify a dataset into categories like cats and dogs. DL nets are increasingly used for dynamic images apart from static ones and for time series and text analysis.

Training the data sets forms an important part of Deep Learning models. In addition, Backpropagation is the main algorithm in training DL models.

DL deals with training large neural networks with complex input output transformations.

One example of DL is the mapping of a photo to the name of the person(s) in photo as they do on social networks and describing a picture with a phrase is another recent application of DL.

Conclusion: We have studied and implemented Boston housing price prediction problem by linear regression using deep neural network.

Assignment No.6

Aim: Classification using Deep neural network: Binary classification using Deep Neural Networks Example: Classify movie reviews into positive" reviews and "negative" reviews, just based on the text content of the reviews. Use IMDB dataset.

Objective: Student will learn:

1. The classification using deep neural network.
2. The concept of binary classification.

Theory:

Binary Classification:

Binary classification is one of the most common and frequently tackled problems in the machine learning domain. In its simplest form the user tries to classify an entity into one of the two possible categories. For example, give the attributes of the fruits like weight, color, peel texture, etc. that classify the fruits as either peach or apple. Through the effective use of Neural Networks (Deep Learning Models), binary classification problems can be solved to a fairly high degree.

Binary classification refers to those classification tasks that have two class labels.

Examples include:

1. Email spam detection (spam or not).
2. Churn prediction (churn or not).
3. Conversion prediction (buy or not).

Typically, binary classification tasks involve one class that is the normal state and another class that is the abnormal state.

For example "*not spam*" is the normal state and "*spam*" is the abnormal state. Another example is "*cancer not detected*" is the normal state of a task that involves a medical test and "*cancer detected*" is the abnormal state.

The class for the normal state is assigned the class label 0 and the class with the abnormal state is assigned the class label 1.

Popular algorithms that can be used for binary classification include:

1. Logistic Regression

2. k-Nearest Neighbors
3. Decision Trees
4. Support Vector Machine
5. Naive Bayes

If the model successfully predicts the patients as positive, this case is called *True Positive (TP)*. If the model successfully predicts patients as negative, this is called *True Negative (TN)*. The binary classifier may misdiagnose some patients as well. If a diseased patient is classified as healthy by a negative test result, this error is called *False Negative (FN)*. Similarly, If a healthy patient is classified as diseased by a positive test result, this error is called *False Positive(FP)*.

We can evaluate a binary classifier based on the following parameters:

i. True Positive (TP): The patient is diseased and the model predicts "diseased"
ii. False Positive (FP): The patient is healthy but the model predicts "diseased"
iii. True Negative (TN): The patient is healthy and the model predicts "healthy"
iv. False Negative (FN): The patient is diseased and the model predicts "healthy"

Conclusion: We have studied and implemented the binary classification using deep neural network.

Assignment No.7

Aim: Convolutional neural network (CNN): Use MNIST Fashion Dataset and create a classifier to classify fashion clothing into categories.

Objective: Student will learn:

1. The concept of convolutional neural network.
2. The working of convolutional neural network.

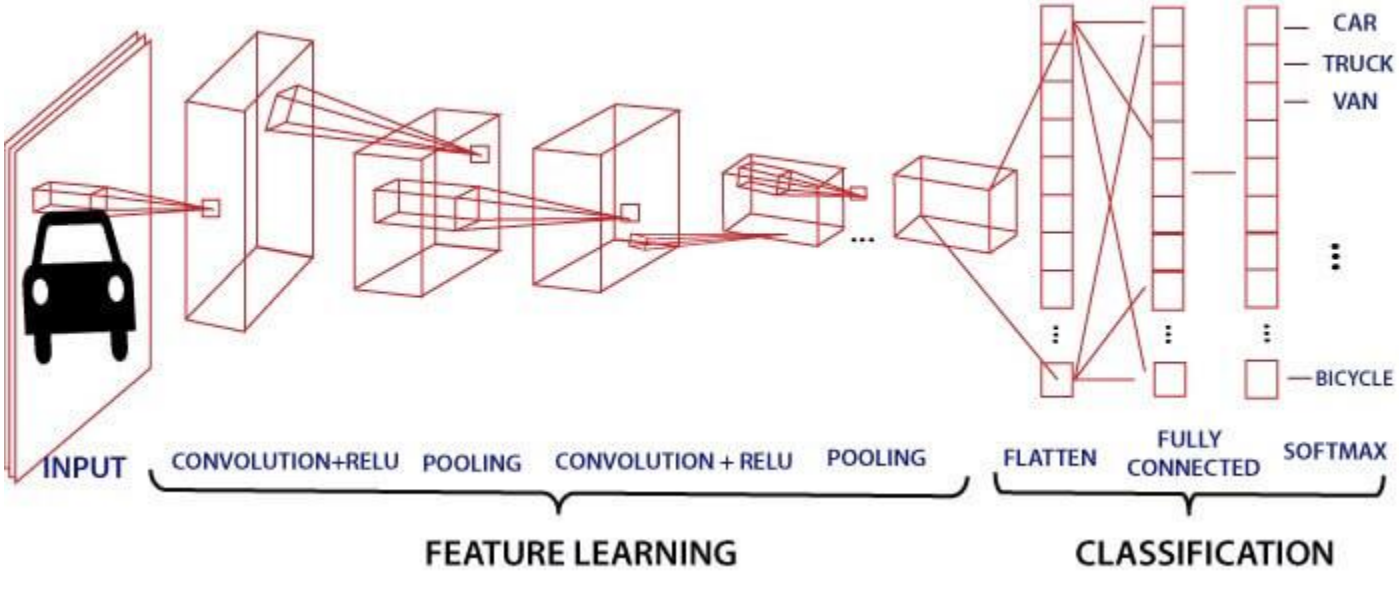
Theory:

Convolutional neural network (CNN):

Convolutional Neural Network is one of the main categories to do image classification and image recognition in neural networks. Scene labeling, objects detections, and face recognition, etc., are some of the areas where convolutional neural networks are widely used.

CNN takes an image as input, which is classified and process under a certain category such as dog, cat, lion, tiger, etc. The computer sees an image as an array of pixels and depends on the resolution of the image. Based on image resolution, it will see as **h * w * d**, where h= height w= width and d= dimension. For example, An RGB image is **6 * 6 * 3** array of the matrix, and the grayscale image is **4 * 4 * 1** array of the matrix.

In CNN, each input image will pass through a sequence of convolution layers along with pooling, fully connected layers, filters (Also known as kernels). After that, we will apply the Soft-max function to classify an object with probabilistic values 0 and 1.



Convolution Layer:

Convolution layer is the first layer to extract features from an input image. By learning image features using a small square of input data, the convolutional layer preserves the relationship between pixels. It is a mathematical operation which takes two inputs such as image matrix and a kernel or filter.

- The dimension of the image matrix is **h×w×d**.
- The dimension of the filter is **f_h×f_w×d**.
- The dimension of the output is **(h-f_h+1)×(w-f_w+1)×1**.

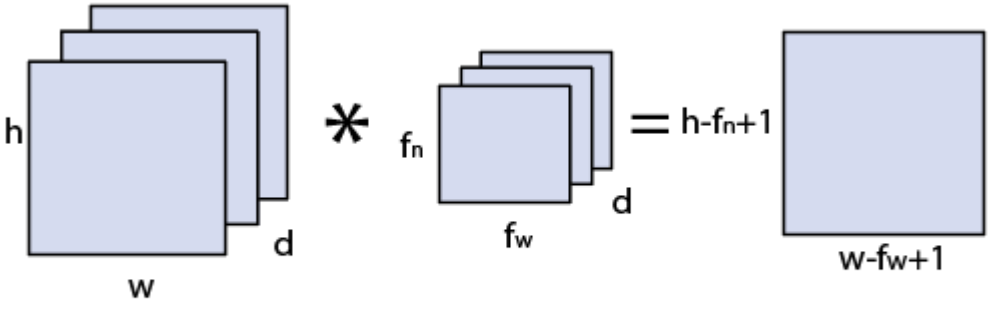


Image matrix multiplies kernl or filter matrix

Let's start with consideration a 5*5 image whose pixel values are 0, 1, and filter matrix 3*3 as:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

5 × 5 − Image Matrix 3 × 3 − Filter Matrix

The convolution of 5*5 image matrix multiplies with 3*3 filter matrix is called "**Features Map**" and show as an output.

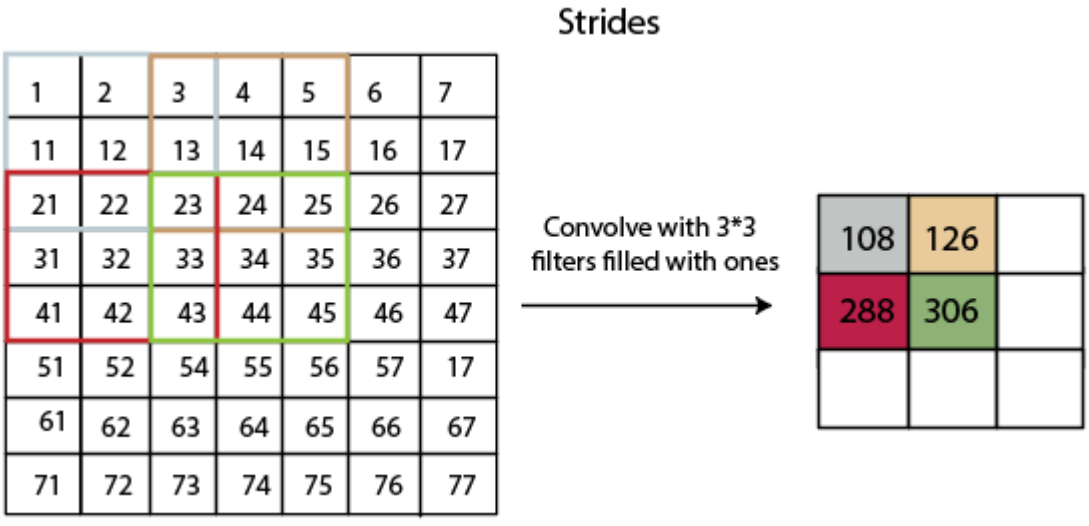
$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 3 & 4 \\ 2 & 4 & 3 \\ 2 & 3 & 4 \end{bmatrix}$$

Convolved Feature

Convolution of an image with different filters can perform an operation such as blur, sharpen, and edge detection by applying filters.

Strides:

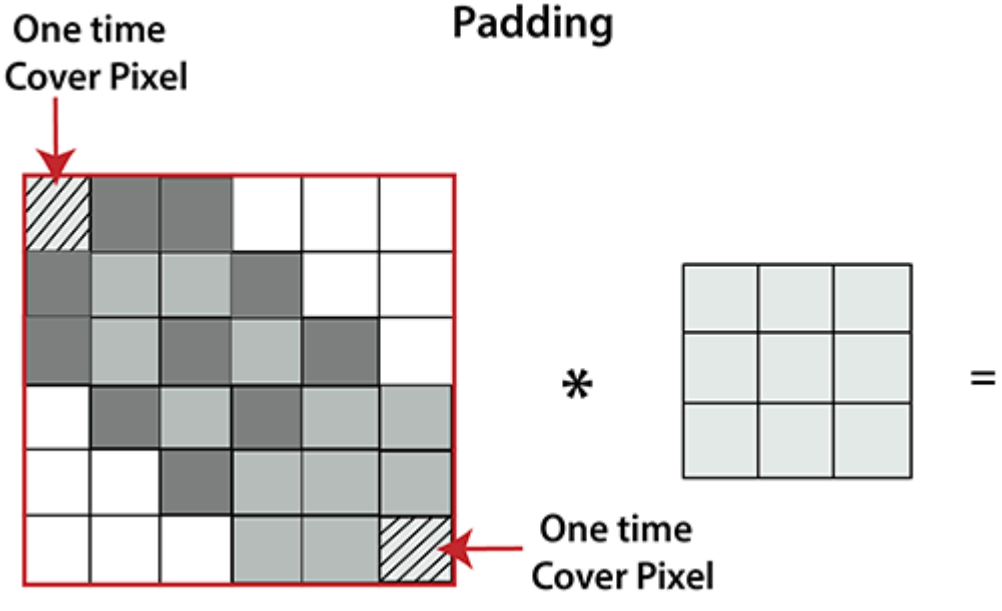
Stride is the number of pixels which are shift over the input matrix. When the stride is equaled to 1, then we move the filters to 1 pixel at a time and similarly, if the stride is equaled to 2, then we move the filters to 2 pixels at a time. The following figure shows that the convolution would work with a stride of 2.



Padding:

Padding plays a crucial role in building the convolutional neural network. If the image will get shrink and if we will take a neural network with 100's of layers on it, it will give us a small image after filtered in the end.

If we take a three by three filter on top of a grayscale image and do the convolving then what will happen?



It is clear from the above picture that the pixel in the corner will only get covers one time, but the middle pixel will get covered more than once. It means that we have more information on that middle pixel, so there are two downsides:

- Shrinking outputs
- Losing information on the corner of the image.

To overcome this, we have introduced padding to an image. **"Padding is an additional layer which can add to the border of an image."**

Pooling Layer:

Pooling layer plays an important role in pre-processing of an image. Pooling layer reduces the number of parameters when the images are too large. Pooling is "**downscaling**" of the image obtained from the previous layers. It can be compared to shrinking an image to reduce its pixel density. Spatial pooling is also called

downsampling or subsampling, which reduces the dimensionality of each map but retains the important information. There are the following types of spatial pooling:

Max Pooling:
Max pooling is a **sample-based discretization process**. Its main objective is to downscale an input representation, reducing its dimensionality and allowing for the assumption to be made about features contained in the sub-region binned.
Max pooling is done by applying a max filter to non-overlapping sub-regions of the initial representation.

Average Pooling:

Down-scaling will perform through average pooling by dividing the input into rectangular pooling regions and computing the average values of each region.

Syntax

layer = averagePooling2dLayer(poolSize)

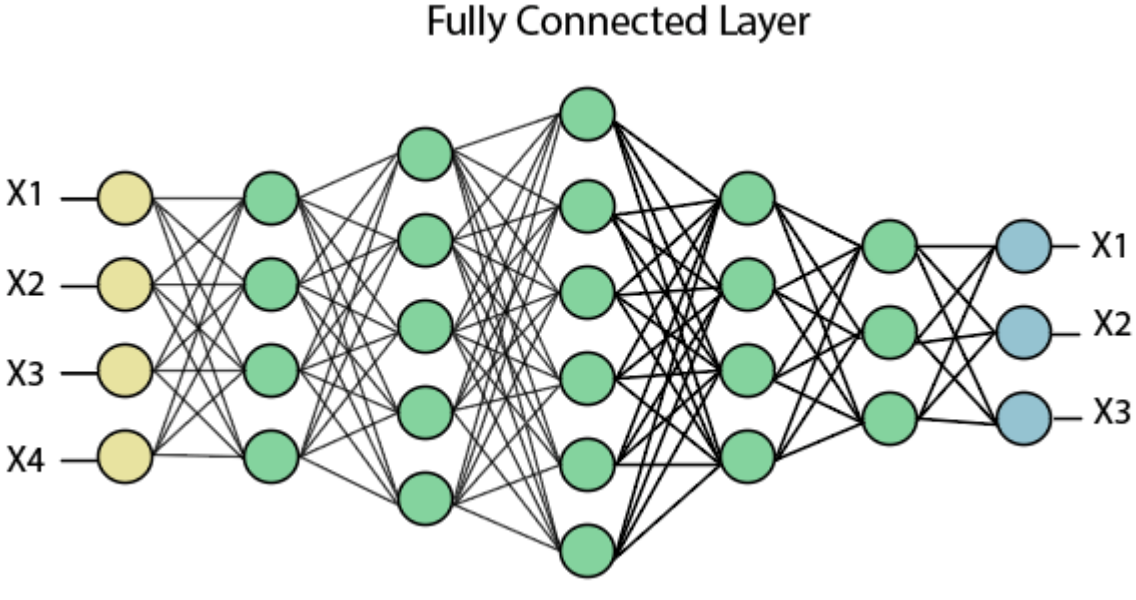
layer = averagePooling2dLayer(poolSize,Name,Value)

Sum Pooling:

The sub-region for **sum pooling** or **mean pooling** are set exactly the same as for **max-pooling** but instead of using the max function we use sum or mean.

Fully Connected Layer:

The fully connected layer is a layer in which the input from the other layers will be flattened into a vector and sent. It will transform the output into the desired number of classes by the network.



In the above diagram, the feature map matrix will be converted into the vector such as **x1, x2, x3... xn** with the help of fully connected layers. We will combine features to create a model and apply the activation function such as **softmax** or **sigmoid** to classify the outputs as a car, dog, truck, etc.

Conclusion: We have studied the concept of Convolutional Neural Network (CNN) and created a classifier to classify fashion clothing into categories.

Assignment No.8

Aim: Recurrent neural network (RNN) - Use the Google stock prices dataset and design a time series analysis and prediction system using RNN.

Objective: Student will learn:

1. The concept of Recurrent Neural Network (RNN)

Theory:

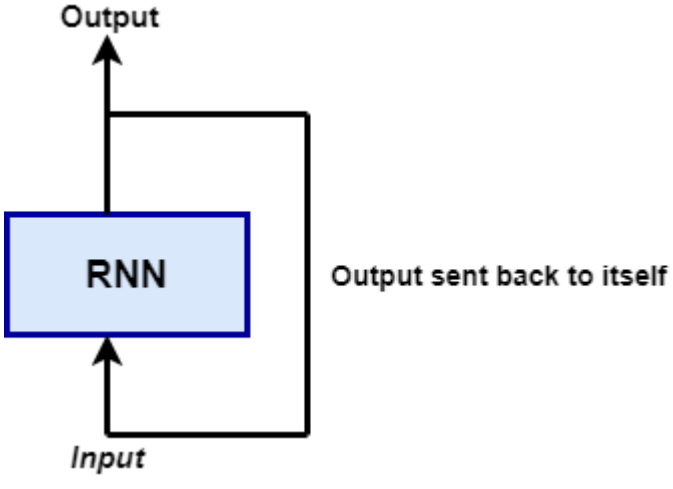
Recurrent neural network (RNN):

A recurrent neural network (**RNN**) is a kind of artificial neural network mainly used in **speech recognition** and **natural language processing** (NLP). RNN is used in deep learning and in the development of models that imitate the activity of neurons in the human **brain**.

Recurrent Networks are designed to **recognize patterns** in sequences of data, such as **text, genomes, handwriting, the spoken word,** and **numerical** time series data emanating from sensors, stock markets, and government agencies.

A recurrent neural network looks similar to a traditional neural network except that a memory-state is added to the neurons. The computation is to include a simple memory.

The recurrent neural network is a type of deep learning-oriented algorithm, which follows a sequential approach. In neural networks, we always assume that each input and output is dependent on all other layers. These types of neural networks are called recurrent because they sequentially perform mathematical computations.



Application of RNN:

RNN has multiple uses when it comes to predicting the future. In the financial industry, RNN can help predict stock prices or the sign of the stock market direction (i.e., **positive** or **negative**).

RNN is used for an autonomous car as it can avoid a car accident by anticipating the route of the vehicle.

RNN is widely used in **image captioning, text analysis, machine translation,** and **sentiment analysis**. **For example,** one should use a movie review to understanding the feeling the spectator perceived after **watching**

the movie. Automating this task is very useful when the movie company can not have more time to review, consolidate, label, and analyze the reviews. The machine can do the job with a higher level of accuracy.

Limitations of RNN:

RNN is supposed to carry the information in time. However, it is quite challenging to propagate all this information when the time step is too long. When a network has too many deep layers, it becomes untrainable. This problem is called: vanishing gradient problem.

If we remember, the neural network updates the weight use of the gradient descent algorithm. The gradient grows smaller when the network progress down to lower layers.

The gradient stays constant, meaning there is no space for improvement. The model learns from a change in its gradient; this change affects the network's output. If the difference in the gradient is too small (i.e., the weight change a little), the system can't learn anything and so the output. Therefore, a system facing a vanishing gradient problem cannot converge towards the right solution.

Training through RNN:

- The network takes a single time-step of the input.
- We can calculate the current state through the current input and the previous state.
- Now, the current state through h_{t-1} for the next state.
- There is n number of steps, and in the end, all the information can be joined.
- After completion of all the steps, the final step is for calculating the output.
- At last, we compute the error by calculating the difference between actual output and the predicted output.
- The error is backpropagated to the network to adjust the weights and produce a better outcome.

Conclusion: We have studied the concept of Recurrent Neural Network (RNN) and designed a time series analysis and prediction system.

Assignment No.9

Aim: Mini-project on High Performance Computing.

Assignment No.10

Aim: Mini-project on Deep Learning.