A Deep Dive into Deep Learning

Anuradha Desai[1], Disha Devani[2] and Dr. Premal Patel[3]
[1]IT Department, Silver Oak University, Ahmedabad, INDIA
[2]IT Department, Silver Oak University, Ahmedabad, INDIA
[3]CE Department, Silver Oak University, Ahmedabad, INDIA

[1]Corresponding Author: anuradha.patel02@gmail.com

ABSTRACT

Deep learning is a technique for mimicking the human brain by intimating human functionality and attempting to uncover fruitful patterns in data using a neural network. With the increase in data volume, deep learning is becoming more popular. End devices such as smartphones and IoT sensors generate data that must be appropriately analyzed using deep learning models. This paper intends to present the reader with complete understanding of the fundamentals of deep learning elements in order to make the principles more evident in the deep learning field. This study focuses on the three major types of neural networks that serve as the foundation of deep learning models. The three primary types are as follows: i) Artificial Neural Network, ii) Convolution Neural Network and iii) Recurrent Neural Network. Let's take a deep dive into each of these sorts.

*Keywords--* Deep-Learning, Neural-Network, ANN, CNN, RNN, Meaningful Insights from Data

I. INTRODUCTION TO DEEP LEARNING

AI Artificial intelligence is a set of models designed to execute tasks similar to human intellect. The primary goal of AI is to mimic and expand on how humans perceive and respond to it. The invention, training, and deployment of machine learning algorithms that mimic logical decision-making based on available data is referred to as AI modelling. ML machine learning is a broad subset of AI that uses machine learning to make systems smarter and smarter over time without human interaction. Deep learning (DL) is a sort of machine learning and artificial intelligence that mimics how humans acquire knowledge.

Deep Learning is an an extended version of ML.
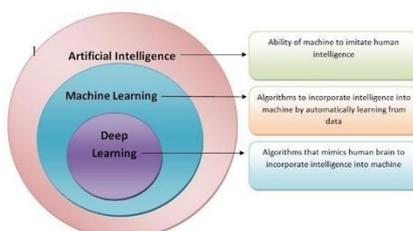
AI is a new type of digital energy that has completely revolutionized our lives. ML and Dl are two key parts of AI that convert data into information and information into knowledge. To recap, DL is a subset of ML, which in turn is a subset of AI.
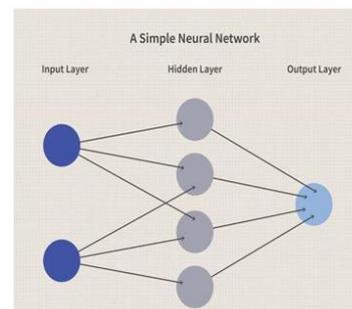
A. Neural Network—Network of DL

A Neural Network is a functional system at the heart of a deep learning algorithm that aids in the processing of raw data. Neural networks are quite similar to the human brain. Just like the brain is made up of neurons, a neural network is made up of nodes or units called artificial neurons [1][2].

Each individual neuron in the input layer performs a subtask and sends the observed pattern to the next group of neurons in the hidden layers. This process is repeated until the generated patterns reach the output layer and we obtain the desired output or result. The most significant aspect of Neural Networks is how they are trained and how they identify patterns in your features. Features in neural networks are variables or properties from your dataset that have been chosen as good predictors for your model. As a result, features are passed into the input layer.

Types of Neural Network

The three significant neural network types that serve as the foundation for the majority of pre-trained deep learning models are listed below:

- Artificial Neural Network-ANN
- Convolution Neural Network-CNN
- Recurrent Neural Network-RNN

II. ARTIFICIAL NEURAL NETWORK-ANN

ANN can be defined as weighted graphs in which artificial neurons are nodes and directed edges with weights are connections between input neurons and output neurons. The artificial neural network works by taking input and computing the weighted sum of the inputs and including a bias.[5][14].

The following is used to represent the ANN's computational function:

$$Y = \sum Xi * Wi + Bias$$

Where, X is the input signal and W is the weight associated weight corresponding to the input signal. The weight helps the neuron to decide whether it should be activated or not or up to which level it should be activated.

Initially, the function was $Xi * Wi$ and if we put weights=0 then $\sum Xi * Wi$ would be 0 and it should not be done, so to solve the problem we add $Xi * Wi$ with some bias.

We pass an activation function for the Y in equation 1. An activation function is a function added to the network to learn the complex pattern in data. A neuron's activity is determined by an activation function, which determines whether input can predict outputs using a mathematical function or not.

Types of Activation Function
1. Sigmoid Function

In situations involving binary classification, sigmoid is frequently utilized. Its value ranges from 0 to 1, and it is not zero-centered. In most operations on the output layer, the sigmoid activation function is used.

Sigmoid Function
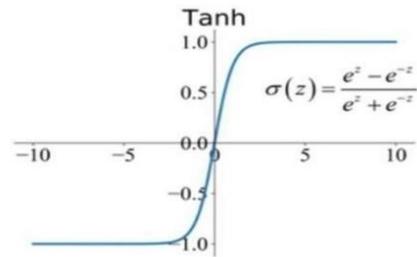
$$a = \frac{1}{1 + \exp(-z)}$$
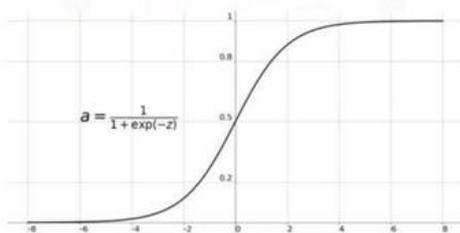
Equation: - A = 1/ (1 + e-x)

2. Tanh Function

Tanh function's value, which ranges from -1 to 1, is most frequently employed in hidden layers. As a result, the tanh function aids in centering the data by bringing the mean near to 0, which greatly facilitates learning for the subsequent layer.

$$\tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Tanh

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

3. ReLU Function

Scientists discovered the activation function known as RELU to solve the vanishing gradient issues that the Sigma and Tanh functions have, which slow down learning. When unsure of the activation function to employ for hidden layers, just choose RELU because it is very efficient computationally.
Equation:- max(0,x)

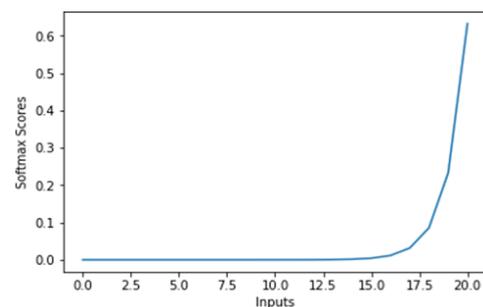**ReLU**

$$R(z) = max(0, \ z)$$
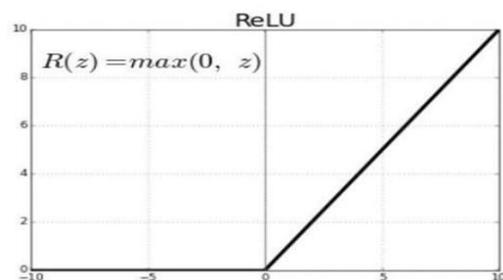
4. Leaky ReLU Function

ReLU has also limitations for the gradient descent problems but there is another flavor of RELU called Leaky ReLU. It tries to reduce the value close to zero.
Equation: - MAX (0.01X, X)

5. Softmax Function

Softmax is the function used for output layer when we are dealing with multiclass classification.

$$softmax(z_i) = \frac{exp(z_i)}{\sum_j exp(z_j)}$$

Forward propagation describes the whole

procedure up to the output layer. Additionally, training is used to adjust the weights of neural nodes based on the mistake rate recorded during the previous iteration. We use back-propagation if the loss function is high.

Error rate or Loss function can be defined as difference between real and predicted output.

We employ Optimizers to adjust weights while performing backpropagation in order to decrease the loss function.

Types of Optimizers
1. Gradient Descent

One of the most used iterative optimization techniques in machine learning, gradient descent is used to train both deep learning and machine learning models. It aids in determining a function's local minimum.

Gradient Descent's primary objective is to repeatedly shrink the loss function.
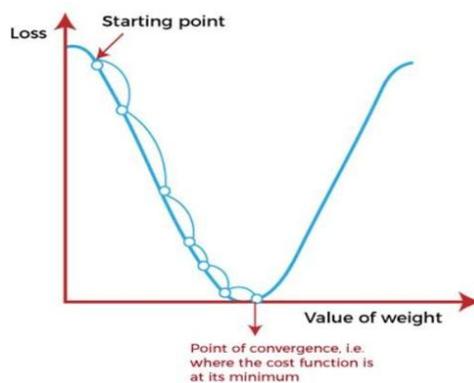
From this starting point, we will derive the first derivative, or slope, and then compute its steepness using a tangent line. When fresh parameters are generated, the slope gradually becomes less steep, and at the lowest point, it approaches the lowest point, which is known as a point of convergence. The slope is steeper at the starting point or arbitrary point [10].

The number of steps needed to get to the lowest point is referred to as steepness or learning rate.

2. Stochastic Gradient Descent

For each iteration of SGD, a small number of samples are chosen at random rather than the entire dataset. If you use a standard Gradient Descent optimization technique and your dataset contains a million samples, you will need to use all one million samples to complete one iteration of the Gradient Descent. This process must be repeated until the minima are reached. As a result, performing it becomes exceedingly expensive computationally.

Utilizing stochastic gradient descent, this issue is resolved.

3. Mini-Batch Gradient Descent

The procedure known as mini-batch gradient descent divides training data into smaller batches that are then used to compute error and update model coefficients as necessary. It is the deep learning application of gradient descent that is utilized the most

frequently.

4. Momentum Based Gradient Descent

The gradient descent optimization algorithm is extended by momentum. Momentum is intended to speed up the optimization procedure, i.e., reduce the number of iterations required to obtain the ideal minima or improve the algorithm's potential to provide an extremely superior outcome. The past values, or taking the average of the previous gradients, are what determine the new weight update.

5. Adagrad

Adaptive gradient optimizer is referred to by the term Adagrad. Adagrad's methodology involves using various learning rates for each parameter based on iteration. It's an optimizer with parameter-specific learning rates that adjust in accordance with how frequently a parameter is changed during training.

6. RMSProp

RMSProp, often known as the root mean square prop, is an adaptive learning technique that seeks to enhance Adagrad. It uses the "exponential moving average" rather than the cumulative sum of squared gradients, as in Adagrad.

7. Adam

Adam optimization algorithm is now more widely used in computer vision and natural language processing deep learning applications. A combination of Momentum-based GD and RMS prop is used in the Adam optimizer.

III. CONVOLUTIONAL NEURAL NETWORK

CNN is a subset of deep learning that is applied to a variety of tasks, particularly those using picture data. They are therefore excellent candidates for Computer Vision (CV) and image-processing-based tasks like face recognition and autonomous vehicles.

A simple explanation of convolutional neural network.

A picture appears to the computer as a grid of numbers. Since a deep neural network has so many hidden layers and a larger image has three RGB channels, millions of weights must be calculated just between the input layer and hidden layer, which can result in calculations of billions of weights. This is too much computation for your small computer to handle.

Thus, drawbacks of ANN for image categorization include:
1. Too much computation.
2. Treats local pixels the same as pixels far apart. (If some pixels are moved around, it should be still able to detect the object but with ANN it's hard.)
3. Sensitive to the location of an object in an image.

A convolutional layer and a pooling layer makeup a CNN.

Convolutional layer - The convolutional layer is where most computations take place. It has filters

that are necessary for feature recognition. Filters identify spatial patterns in images, such as vertical and horizontal edges. So, using the original image as a starting point, we run a filter operation. It operates by taking a grid from the original image, such as a 3X3 grid, and multiplying each number with a filter. You create a feature map with that specific feature discovered when you apply this filter or convolutional procedure. Simply put, filters are feature detectors [3].

Pooling Layer

It eliminates excessive calculation and needless parameters. Instead of the feature map (positioned feature) produced by the convolution layer, additional operations will now be carried out on the reduced feature Map (summarized features).

Types of Pooling Layers

1. Max Pooling

The max pooling operation chooses the greatest number from the feature map's area. As a result, significant features from the preceding feature map will be present in the output following the max-pooling procedure.

2. Average Pooling

The average of the components in the area of the feature map is calculated using average pooling. An average of the elements (features) in a patch is produced by average pooling.

Benefits of Pooling

- Reduces dimension and computation.
- Reduces overfitting as there are fewer parameters (features).
- Model is tolerant towards variations.
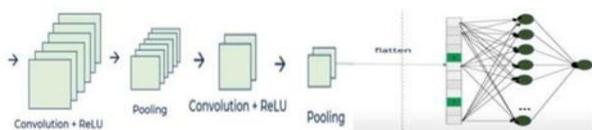
Applying convolution, pooling, and an activation function to your input image will help it become more nonlinear between many of the convolutional layers. Employ RELU) and then apply pooling up to n times if you are unsure of the activation function to use. After conducting the convolutional operation between the convolutional layers, "flattening" is the following step. The feature maps' resultant 2-D arrays are all flattened into a single continuous linear array or matrix. In order to classify an image, the flattened vector is then provided as input to a completely dense layer.

IV. RECURRENT NEURAL NETWORK-RNN

RNNs are a particular kind of deep learning output from the present state that are provided as input for the following state. In RNN, inputs and outputs are interdependent. RNNs feature a memory that keeps track of all information related to calculations [14].

Uses of RNN

- Autocomplete
- Translator
- Name entity recognition
- Sentimental analysis
- Chat box
- Natural Language Processing

Every piece of knowledge is retained by an RNN over time. Only the ability to remember past inputs make it helpful for time series prediction. Long Short-Term Memory is the term for this.

V. CONCLUSION

Therefore, the purpose of this survey study is to provide essential knowledge on deep learning concepts in simple, understandable terms. Therefore, this study will mostly benefit those who are just beginning their careers in data science.

REFERENCES

[1] Zewen Li , Wenjie Yang , Shouheng Peng & Fan Liu. (2020). A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE*.

[2] N. Aloysius & M. Geetha. (2017). A review on deep convolutional neural networks. *Proc. Int. Conf. Commun. Signal Process. (ICCSP)*, pp. 588-592.

[3] A. Dhillon & G. K. Verma. (2020). Convolutional neural network: A review of models methodologies and applications to object detection. *Prog. Artif. Intell., 9*(2), 85-112.

[4] Khan, et al. (2020). A survey of the recent architectures of deep convolutional neural networks. *Artif. Intell. Rev., 53*, 5455-5516.

[5] A survey of artificial neural network machine learning algorithm. *IJSRSET, 7*(3).

[6] H. Ajmal, S. Rehman, U. Farooq, Q. U. Ain, F. Riaz & A. Hassan. (2018). Convolutional neural network based image segmentation: A review. *Proc. SPIE, 10649*.

[7] W. Rawat & Z. Wang. (2017). Deep convolutional neural networks for image classification: A comprehensive review. *Neural Comput., 29*(9), 2352-2449.

[8] Z. Zhang et al. *Differentiable learning-to-group channels via groupable convolutional.*