A Study on Deep Learning as a Frontier of Machine Learning

Khushi, Preeti Negi, Simanjeet Kaur
BCA Students of Department Computer Application,Tula's Institute Dhoolkot, Dehradun, India
Under Guidance of **Mr.Rakesh Kumar** Assistant Professor in Tula's Institute Dhoolkot Dehradun.

ABSTRACT

Because of advancements and the advent of deep learning, there has been a revolution in machine learning applications in recent years. Deep learning models are superior to traditional machine learning models because they include more learning layers and a greater level of abstraction. This benefit is further supported by the fact that all components of the model directly learn from the data. Conventional machine learning models experience limits as a result of the method they use to process the data because of the growing volume of data and the increased desire to derive useful insights from the data. The increase in data volume has led to the development of more sophisticated learning algorithms that are quicker and more accurate. Every firm will undoubtedly utilize a model like this one that produces the most accurate predictions in order to stay ahead of the competition. We shall discuss some of the most widely used deep learning techniques in this paper.

Keywords:-Neural networks, machine learning, and deep learning.

1.INTRODUCTION

A set of machine learning algorithms called "deep learning," which draws its inspiration from the biological process of neural networks, is gaining ground in several fields and outperforming more established methods [1]. Only because they are capable of delivering quicker and more precise outcomes. Based on a number of methods, it aims to model high-level abstraction in data [2]. All components of the model are directly learned from the data in deep learning approaches. Lowest level features that provide an appropriate representation of the data are used as a starting point. After that, it offers higher-level abstractions for each unique issue that it is applied to. The more training data there is, the more effective deep learning becomes. With the expansion of the software and hardware infrastructure, deep learning model development has increased [3].

Deep learning models employ numerous layers made up of both linear and non-linear transformations. Conventional machine learning algorithms have demonstrated their limitations in analysis with the size of data with the increase in data size or with innovations in the field of big data [4]. In this analysis challenge, deep learning algorithms have produced improved outcomes. This method has been hailed as a technological breakthrough since it distinguishes itself from machine learning approaches that rely on outdated and conventional methods [5]. In modeling the intricate relationship between data, it is helpful. It is predicated on learning data representations rather than working on task-specific algorithms. You have the option of semi-supervised, unsupervised, or supervised learning.

The task of feature extraction is carried out by numerous layers of non-linear processing units in deep learning models. Transformation. Every layer uses the input as the output of the layer below it. It is used in supervised classification challenges as well as unsupervised pattern analysis applications. A hierarchy of notions is created by the several levels that offer the high-level abstraction. Deep generative models are deep learning models that are mostly built on artificial neural networks that are organized layer-wise. The production of observed data through the interaction of multiple components is the idea underlying this distributed representation. These stacked components enable the high-level abstraction. By adjusting the number and size of layers, a variable level of abstraction can be achieved [6].

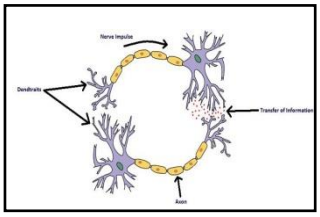
The abstraction is accomplished by using the hierarchical exploratory elements to learn from the bottom level. The deep learning methods minimize feature engineering in supervised learning applications by transforming the data into compact instantaneous representations of major components and eliminating redundancies in representation through derived layered structures. Deep learning methods can be used to solve these types of problems in unsupervised learning, where unlabeled data is more prevalent than labelled data. One deep learning model that is used to solve such unsupervised situations is deep belief networks.

Because more abstract repetitions are based on less abstraction, deep learning algorithms take advantage of the abstraction of data. These models are hence resistant to local changes in the input data. This offers a benefit in a number of pattern recognition issues. The deep learning models can better extract features from the data thanks to this invariance. These models are able to differentiate between the many causes of data fluctuation thanks to the abstraction in their representation. By manually setting the learning characteristics, deep learning models perform better than previous machine learning models. This is due to the fact that the models' design is independent of the system's training and that it relies on human domain knowledge rather than on the data that is currently accessible.

Researchers have created a variety of deep learning models that enable improved learning from the representation of vast amounts of unlabelled data. In the fields of computer vision and predictive analytics, some well-known deep learning architectures, such as convolutional neural networks (CNN), deep neural networks (DNN), deep belief networks (DBN), and recurrent neural networks (RNN), are applied as predictive models. Deep learning models are demonstrating their abilities in the work of predictive analytics to address the data analysis and learning difficulties due to the rise in data quantity and requirement to produce a quick and correct result. We now give a survey of well-liked deep learning models that are cantered on artificial neural networks in this paper. We shall address the operation and use of ANNs, CNNs, DNNs, DBNs, and RNNs.

2. Network of artificial neurons

A computing model called an artificial neural network was inspired by biological neural networks. The biological brain network, which consists of billions of neurons, receives electrochemical impulses from nearby neurons. When these signals are processed, they either store them or send them to the following nearby neurons in the network [7], [8]. It is shown in figure 1 in the paragraph below.



Biological neural network in Figure 1.

Every neuron in a living thing is connected to its neighbors and can communicate with them. The input-output signals are carried by the axons in the network. They take in environmental stimuli that result in impulses as electrochemical signals that move fast across the network. The information may be stored by a neuron or forwarded to the network. Through their dendrites, they communicate information to the nearby residents.

The operation of artificial neural networks is comparable to that of biological neural networks. A network of artificial neurons is an ANN. Every neuron in the layer is coupled to every neuron in the layer below it and the layer above it. Each time a neuron connects to another, the labels contain a weight. Each neuron receives input, which comes from the previous layer's neurons' output. They analyse this input and produce an output that is passed on to the neurons in the following layer. Each neuron in the network has an activation function that it employs to gather inputs, add inputs, and produce output [9]. Depending on the desired output, different activation function types are selected.

A straightforward artificial neural network has three layers. Three layers: the output layer, the hidden layer, and the input layer. The input layer is subjected to inputs in the form of input vectors. The amount of attributes in the input determines how many neural nodes are present in the input layer. Each input layer neuron sends its output to each hidden layer neuron, which receives it as an input. The processing layer is another name for the hidden layer. because input processing takes place mostly at this layer. The number of nodes in the hidden layer is initially determined at random and is subject to change throughout training. Each neural node's output from the hidden layer is subsequently sent to the output layer, where it serves as an input. The network's final output is subsequently generated by the output layer and collected. The kind of output determines how many nodes are present at the output layer [10]. The number of nodes in classification issues is equal to the number of classes the inputs must be divided into. Assigned. There could be just one output node in regression problems that generates an output value.

There are two different forms of feed-forward artificial neural networks based on the number of layers. The single layer feed-forward ANN and the multilayer feed-forward neural network are the two different types. There is only one layer in the network, and there are no hidden layers. This type of neural network is the simplest. Only the input layer and the output layer make up the network. For the purpose of producing the outputs, the inputs applied to the input layers are immediately passed on to the output layer.

3. Artificial neural network that is fed forward

Artificial neural networks come in many different varieties, each of which has a unique feature and can be used to solve a different kind of problem. Artificial neural networks with a feed-forward structure are widely employed in a variety of applications. The only direction that signals can move in in a feed-forward network is forward, from the input layer to the output layer [11]. We shall examine feed-back or feedback neural networks in a later section of this chapter. Every neural network uses a learning algorithm to function. Depending on the issue that the network is being applied to, different learning methods are chosen. By putting the learning algorithm into practice, the networks are trained. The back propagation learning technique is widely used and used to train neural networks in many different applications. It uses a layer's output error to modify the weight of interconnections. The earlier layers receive a backwards propagation of this fault. It is known as the back propagation algorithm for this reason [12]. For the network's supervised and unsupervised training, there are numerous other methods.

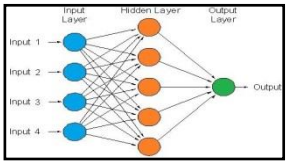


Figure 2 below illustrates the construction of a feed-forward neural network.

The architecture of a feed-forward neural network is shown in the previous figure. Each artificial neuron in this network is interconnected with the neurons in the layers above and below it. The inputs are applied to the vector of neural nodes at the input layer of the network during training. The main input is frequently combined with a bias input in learning algorithms. The training process fixes this bias value. When the initial input pattern is applied, the hidden layer receives it. The output that is gathered at the output layer is produced by the activation function utilized at neurons. The step functions are typically utilized in classification issues, and in regression problems. The logistical issues are employed. On the dataset, the network is trained across a number of epochs. Some algorithms employ gradient descent to halt the learning process when a certain error is reached.

If the inputs I1, I2,..., In are applied to the network's input layer, the net input that is subsequently received at a single hidden layer neuron is as follows:

Stepper Mechanism

The binary step function and the bipolar step function are the two different forms of step functions. If the net input is less than a predetermined threshold value, the binary step function outputs 0; otherwise, it outputs 1. It can be analytically described as given in equation 10 and graphically represented as shown in figure 3.

If there are m neurons, the net input received at the hidden layer is expressed as a vector with the following notation: where w is the weight on connectivity and b is the bias value.

Yin=,[$yin\ 2$,…, $yin\ m$] (3)

Let W be the matrix of weights associated with the connections between the input layer and the hidden layer, and let I be the vector representing the inputs, then Yin will be defined as the cross product of the input vector and the weight matrix, i.e.,

Yin=I×W (4)

If this neuron's activation functions, f, is employed, then the following is the neuron's output:

$yout$ =f(yin) (5)

The output of every neuron in the hidden layer can similarly be represented as a vector as follows:

Yout=[$yout\ 1$, $yout\ 2$, … . , $yout\ m$] (7)

Now, each neuron in the output layer is given this output Yout as input. Let Zin be the net input received at the output layer, then it can be described as: If V is the matrix of weights connected to the connections between the hidden layer and the output layer, then the input received at the output layer will be the cross product of Yout and V.(8) Zin=Y out in relation to the threshold. Equation 11 and Figure 4 both provide mathematical and graphic representations of this function, respectively.

(x) = {−1, $x < \theta$ (11)

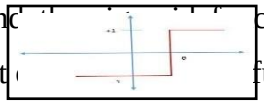
Consider Zout i to be the output produced by each of the i neurons at

If there are p nodes in the output layer, then the vector of outputs produced by each neuron can be used to represent the net output that is gathered there. It may be expressed as:

Zout=[$Zout\ 1$, $Zout\ 2$, … . , $Zout\ p$] (9)

4.FUNCTIONS OF ACTIVATION

Each neuron in the neural network produces an output known as activation of the neuron, which is the output. At first, it is thought that neurons are not engaged if they are not producing any output. The neurons begin firing their output, which is referred to as the neuron becoming activated, when the applied input values cross a particular threshold. A function that transfers the net input received to the neuron with the output is employed for this activation. The activation function is the name of this process. Depending on the issue the neural network is applied to, several kinds of activation functions are used at neurons [13]. The step function an[illegible]ction are the two most often utilized activation functions. Here, we'll give a succinct[illegible]functions.



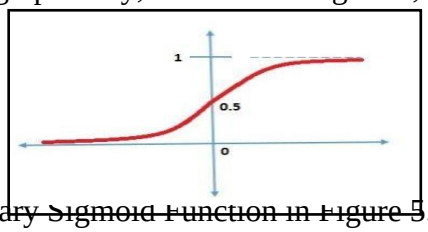
Bipolar Step Function (Figure 4)

Signature Function

The step functions cannot be differentiated since they are not continuous. Step functions cannot be employed in certain machine learning techniques because they need for continuous and differentiable activation functions. The sigmoid functions' differentiability quality can be maintained while a close approximation is made. The binary sigmoid function and the bipolar sigmoid function are the two forms of sigmoid functions employed in this kind of issue area. Each of them has a continual output.

The logistic sigmoid function is an alternative name for the binary sigmoid function. Both technically, as shown in equation 12, and graphically, as shown in figure 5, it can be represented.

f(x) = 1 1+e−αx



Binary Sigmoid Function in Figure 5.

Neural networks employ a variety of learning techniques, primarily falling into two categories: supervised learning and unsupervised learning [15].

Supervised education

The labelled data is used using the supervised learning techniques. Data that has input and output labels on it is said to be labelled data. The training pattern refers to the data used to train a neural network. In the case of supervised learning, each training pattern is made up of the associated output patterns and input patterns. The data's input and output patterns are mapped by the learning algorithms as a function. The network can produce output for an unknown input pattern once it has been trained using the learning algorithm [16]. Here, we'll give a very succinct overview of the supervised learning methods that are frequently used to train neural networks.

Hebb Law

It's one of the first learning methods that artificial neural networks have ever employed. The change in weight wi can be calculated using the Hebb rule or Hebbian learning as:

The drawback of the binary sigmoid function is that bipolar data cannot be used with it. The bipolar sigmoid function is utilized in this instance for continuous output. Both technically, as shown in equation 13, and graphically, as shown in figure 6, it can be represented.

f(x) = 1−*e*−αx (13)

1+*e*−αx

5. LECTURED BY ANNS

The architecture, the activation function, and the learning are three crucial features of an artificial neural network. Prior to network training, the weights corresponding to the network's connectivity between neurons are initialized randomly. Following the learning method, these weights are modified, and once they are set, the network is referred to as a stable fitted network. A network that has been trained can be used to solve a problem [14]. Each time the neural network is trained, its weights are changed. This process continues until a halting condition is met. The new weight at the (k+1)st iteration of the training is derived as shown in equation 14 if w(k) is the weight at the kth iteration of the training.

w(k+1) equals w(k) plus w(k) (14)

where w(k) represents the weight's change at iteration k. Different teaching techniques offer various ways to find the value of w(k).

Δwi = Ii × t (15)

where t is the target value and Ii is the appropriate input value. The Hebb rule's restriction prevents it from learning if the target is 0. This is due to the fact that when we set t=0, the change in weight (wi) will become 0. Therefore, it is used whenever both the input and output have a bipolar structure [17].

5.2.2.Rule of Perceptron Learning

According to perceptron learning, only the weights associated with the interconnections should be changed if the neuron's output is not equal to the goal output, else they shouldn't be [18]. This rule describes how to calculate the weight change wi:

Wi = (t-yout) Ii (16), where is a constant and is referred to as the learning rate.

Alpha Rule

The Widrow-Hoff rule and the Least Mean Square (LMS) rule are other names for the Delta rule. It is a popular technique for learning that is used to train neural networks. By lowering the mean squared error between the activation and the goal value, it generates the output in binary form [19]. The Delta rule states that the weight change is calculated as:

Wi = T-Yin Ii (17), when the symbols are employed in their conventional contexts.

Algorithm for Backpropagation

The most common learning method used for supervised learning to train an artificial neural network is backpropagation. According to the training patterns, the neural network in this technique continuously modifies the interconnection weights based on mistake and divergence from the intended output. This method's error is estimated at the network's output layer and is transmitted back through the network layers [20]. When talking about the training of the hybrid deep neural network in chapter 6, we will go into more detail about this strategy.

5.3. Unsupervised Education

Unsupervised learning is the process of learning from data when the input-output labels are absent from the training data used to train the neural network. In this instance, the algorithms discover structure in the data. Unsupervised learning is used to solve several machine learning issues, including clustering and anomaly detection [21]. The input vectors that are to be applied to the neural network in clustering problems are joined during neural network training to produce clusters. As soon as a fresh input vector is applied to a trained or stable neural network, the output response indicates the cluster to which the input vector belongs. Winner-Takes-All is a prominent unsupervised learning approach that is used for grouping using neural networks. The neuron with the highest overall input is chosen as the winner according to a competitive learning rule [22].

6. MODELS OF DEEP LEARNING

Different deep learning models have been developed by researchers and are used in diverse problem domains. The numerous learning layers are the shared feature of all the models. We will give a brief overview of some of the most often used deep learning models in this section.

6.1. Neural Deep Network

A multilayer feed-forward artificial neural network version is the deep neural network. Between the input layer and the output layer, there are multiple hidden layers [23]. Each hidden layer has a similar amount of neurons. The number of neurons is initially fixed at random and is manually modified during network training. Greater complexity and, hence, a decline in training effectiveness may be caused by more nodes in the hidden layer. The number of nodes at this tier is therefore carefully chosen. The compositional model used by this architecture refers to the item as a layered composition of primitives. In the training data, it has the ability to simulate intricate non-linear relationships. Utilizing more hidden layers in the network has the advantage of allowing for the compilation of characteristics from lower layers. These characteristics may allow for a more efficient modeling of complex data [24].

The deep neural networks are also accompanied by two problems. First, the overfitting problem, which is prevalent in many neural network models, and then the computation time problem. Due to the introduction of additional layers, the issue of overfitting has a higher likelihood of occurring in deep neural networks. This problem causes it to represent the uncommon dependencies in the training set. On training data, the network performs better, but on validation data, its accuracy suffers. Regularization techniques like weight decay or sparsity can be employed during training to prevent the problem of overfitting in deep neural networks by preventing the modeling of uncommon relationships. Overfitting can be beaten with an increase in shorter training sessions. The learning model's calculation time is influenced by a number of factors, including the layer size, learning rate, and initial weights [25]. The system becomes more complex and takes longer to compute as a result of the additional nodes in the hidden layers. While choosing each of these characteristics, it should be carefully taken into account.

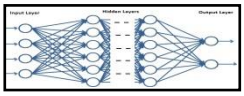


Figure 7 illustrates a typical deep neural network architecture.

Deep neural network (Figure 7)

The multilayer feed-forward artificial neural networks and the deep neural network share a lot of similarities in their processing. The backpropagation learning technique is frequently used to determine the correspondence between the desired output and the actual output when training a network. In this procedure, the weight change is determined as follows:

w (t+1) equals w (t) plus C plus (t) (18).

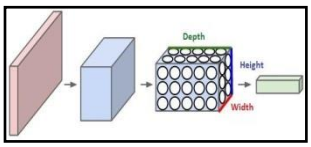
∂wij

where is the stochastic term, C is the cost function, and is the learning rate. The weight of the connection between the ith node of one layer and the jth node of the following layer is called wij.

There are numerous uses for deep neural networks. They are used in a variety of domains, including mobile advertising, drug discovery, customer relationship management, image recognition, natural language processing, and automatic speech recognition.

6.2. Neural Convolutional Network

A multilayer perceptron is a version of the convolutional neural network. The biological process of visualizing serves as a source of inspiration. This model is made up of neurons, bias values, and learnable weights. There are several hidden layers between the input layer and the output layer in addition to an input layer and an output layer. The convolution layers, pooling layers, fully connected layers, and normalizing layers make up the network's hidden layers. They are created with the least amount of pre-processing necessary in mind [26, 27]. Figure 8 in the text below shows how a convolutional neural network is built.



Convolutional neural network, Figure 8.

Convolutional neural network architecture consists of neurons in three dimensions: depth, height, and breadth, which are displayed in one of the layers. Every layer of the network converts the 3D input volume into a 3D output volume of activated neurons. Convolutional neural networks often have the following architecture:

(i).Convolutional Layer: The network's convolutional layer is referred to as the fundamental component and consists of a group of programmable filters. According to reports, these filters are surrounds the layer convolved. The input is given a convolution process before being sent on to the following layer as a result of this operation. When filters are engaged after identifying a particular sort of feature at a specific spatial input position, the network learns from the filters as they operate.

(ii).Pooling Layer: A major problem in artificial neural networks, overfitting is prevented by the convolutional neural network by using the pooling layer. A type of non-linear down-sampling called pooling combines the outputs of one layer's neurons into a single neuron in the following layer. The max-pooling provides the maximum output for each set after dividing the input data into a set of non-overlapping slices.

(iii).Local Connectivity: In convolutional neural networks, only the nearby neurons in adjoining layers are connected between neurons in a given layer. This feature prevents the connectivity issue when handling a huge volume of input, which lowers the network's complexity.

(iv).Convolutional neural networks have a feature called parameter sharing that aids in managing the free parameters. The network's neurons share weight vectors and bias values, which results in less parameter optimization and quicker convergence during training.

Convolutional neural networks are used to solve text analytics and phrase classification issues in the field of natural language processing [28]. It is also employed in time-series analysis, which aids in forecasting weather, market prices, and tidal heights in the ocean [29]. Convolutional neural network design has been applied to the prediction of DNA sequence binding [30]. By foreseeing the interactions between biological proteins and chemicals, these designs are also utilized in the drug discovery process [31].

Convolutional neural networks have proved quite helpful in many sectors and have produced superior outcomes, however this model also has significant drawbacks. It needs a large data collection, which means that training will take a while. Due to the fact that this architecture is GPU-based, performance and scalability issues are also involved [32].

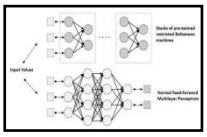
6.3.Networks of Deep Belief

A subtype of a deep neural network is a deep belief network. It is a graphical model made up of numerous hidden unit levels. The latent variables are the hidden components. Layers of the network are connected, but there is no communication between individual network units. This graphical model picks out the training data's deep hierarchical structure as it learns [33]. The edges of the graphical model can be either directed or undirected. The network is trained in two stages back-to-back: unsupervised training first, followed by supervised training. When educated on a set of examples during unsupervised training, the network learns to probabilistically rebuild its inputs. These training steps turn the layers into feature detectors. Following this, the network is trained under supervision to carry out the classification task [34].

By dividing the deep belief network's design into two components—the belief network and the Restricted Boltzmann Machine—the network may be explained. The stochastic variables are included in the belief network, which is a directed acyclic graphical model. These variables can only exist in one of two states, 0 or 1, and the likelihood that they will become 1 is determined by a bias and weighted inputs from other units. The belief network addresses two different the learning problem and the problem of inference. The learning problem modifies the relationship between learning variables whereas the inference problem infers the state of the unobserved variables. This aids the network in producing the data that is observed.

The artificial neural network's generative models, known as Restricted Boltzmann Machines, learn from the probability distribution of a collection of inputs [35].

Figure 9 illustrates a typical deep belief network architecture.



Deep Belief Network in Figure 9.

A feedforward multilayer perceptron and a restricted Boltzmann machine (RBM) make up the deep belief network. The multilayer perceptron is employed during the fine tuning phase, while the RBM is used during the pre-training phase. The neurons that make up the network's hidden units can be deduced from the other observable variables even if they cannot be seen directly. The distribution between the observed input vector X and the nth hidden layer hn in deep belief networks is described as:

P(X, h1, h2,..., hn) = (n2 P(hi| hi+1))P (hn1, hn) (19)

Where X=h0, P(hi1), and P(hn1), respectively, represent the conditional and joint distributions of the observable units. The network learns a layer of features from the visible units during the initial round of training. Then, in the following step, it learns features in a second hidden layer while treating the activation of previously learnt features as a visible unit. When the learning for the final hidden layer is accomplished after proceeding with successive phases in this manner, the entire network is considered to have been trained.

In order to strengthen the financial industries, Deep Belief Networks have been applied in financial business predictions. Additionally, these networks have been used to anticipate time series, which has led to the prediction of financial markets, signal processing, and weather data. This model has also successfully predicted draught. It is also employed to forecast the level of interior noise in vehicles [36].

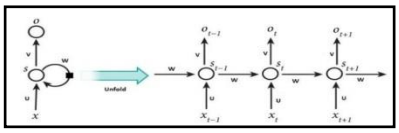
Although Deep Belief Networks have a wide range of applications, this model also has certain drawbacks. Boltzmann Machines, which are used to create deep belief networks, have the drawback of exponentially increasing model training times as machine size increases.

6.4.Continuous Neural Networks

The category of artificial neural networks includes recurrent neural networks. These networks have directed cyclic connections running along a sequence between their internal nodes. It displays a time sequence's dynamic temporal dynamics. These networks process in the internal memory states of input pattern [37]. In traditional artificial neural networks, each input value in an input vector is processed independently because they are not related to one another. The output of many activities, however, depends on earlier calculations made in a sequential process. Recurrent neural networks are used for problems like these where the inputs are processed sequentially. Because it completes the same task for each element in the sequence, this network is referred to as recurrent. The network's memory is where it

keeps the data from earlier calculations. In reality, these networks can only retain a small number of past calculations [38].

The design shown in figure 10 below can be used to describe how the recurrent neural network functions.



Recurrent neural network, Figure 10.

In the illustration above, a recurrent neural network is shown unfolding into a complete network to handle a series of inputs. Each input value in the sequence is handled by one layer in this case. The weights and bias are unchanged if all the layers are folded or combined into a single hidden layer because there is only one hidden layer used in the network. Let st be the hidden state or memory at timestamp t and xt be an input to the network at timestamp t. Based on the input at the current timestamp and the concealed state at the previous timestamp, this st is determined as follows:

f(U. xt + W. st-1) = st (20)

tanh or ReLU are two examples of f, which is a nonlinear function. First timestamp initializes st-1 to zero. The output at timestamp t is represented by the expression ot

(V. st) = ot = f (21)

In the aforementioned equation 4.21, the logistic function f can either be a normalized exponential function or a softmax function.

Recurrent neural networks share these characteristics across all timestamps, in contrast to other deep neural networks that use distinct parameters like weights and bias at various hidden layers. This facilitates learning by lowering the number of parameters. In some applications, an input is needed at every timestamp, and an output is generated at every timestamp. However, it is not required in all recurrent neural network applications. This is due to the usage of hidden state, which records data regarding specific sequences.

The recurrent neural network has several extensions. Brief summaries of each are provided below.

• Bidirectional Recurrent Neural Networks: This network is founded on the idea that the output at a given timestamp depends not only on the components that came before it in the sequence but also on the elements that will come after it. Two recurrent neural networks are stacked on top of one another in its architecture. Based on the hidden state of both networks, it calculates its output [39].

In addition to having numerous layers per timestamp, deep bidirectional recurrent neural networks are similar to bidirectional recurrent neural networks in other ways. It has the advantage of greater learning ability, although a substantial amount of training data is required [40].

• Long Short-Term Memory (LSTM) Networks: This recurrent neural network variation is used to get over the back propagation learning vanishing gradient issue. The memory units in these networks are referred to as cells, and they are highly effective in capturing long-term dependencies. The cells decide internally which information will be saved and which information will be destroyed when given the current input xt and the previous state st-1 [41].

There are several uses for recurrent neural networks in predictive analytics. It has long been a popular tool for making stock market predictions [42]. Its use in time series prediction has produced performance that is more universal than that of other models [43]. These networks have been combined with dynamic weights and used to forecast the software's dependability [44]. The recurrent neural network is utilized to forecast wind speed by including spatial correlation variables [45].

RNNs have some restrictions in addition to the aforementioned significant uses. These networks take a long time to train. Before training, the number of hidden neurons in RNNs must be fixed. The size of the context must be minimal while processing a vocabulary [46].

6.5 Summary and Future Directions

We have covered the various deep learning application strategies in this paper. In the field of machine learning, each of these models has a stellar track record. These models provide room for new features to be added, allowing for more effective use across a wide range of disciplines. To take advantage of the model's potential for prediction, the new methodologies may be incorporated. These models' efficiency can also be increased through parameter adjustment. These models' efficiency can also be increased through parameter adjustment. Therefore, it can be claimed that this model has a very broad opportunity and open scope.

REFERENCES

[1]. I Goodfellow, Y Bengio, A Courville, 2016, “Deep Learning”, MIT Press, Online.

[2]. L Deng, D Yu, 2014, “Deep Learning: Methods and Applications”, Fundamentals and Trends in Signal Processing, Vol-7, Issue-3, Pages- 197-387.

[3]. R Nisbet, G Miner, K Yale, 2018, “Chapter-19, Deep Learning”, Handbook of Statistical Analysis and Data Mining Applications, 2nd Edition, Academic Press.

[4]. X-W Chen, X Lin, 2014, “Big Data Deep Learning: Challenges and Perspectives”, IEEE Access, Vol-2.

[5]. R D Hoff, Accessed 2018, “Deep Learning: A Breakthrough Technology”, MIT Technology

Review, Online.

[6]. M M Najafabadi, F Villanustre, T M Khoshgoftar, N Seliya, R Wald, E Muharemagic, 2015, "Deep learning applications and challenges in big data analytics", Journal of Big Data, Vol-2, Issue-1, Pages- 1-21.

7]. B. Yegnarayana, "Artificial Neural Networks", Prentice- Hall of India, Latest Edition.

[8]. M V Garven, S Bohte, Accessed 2018, "Artificial Neural Networks as Models of Neural Information Processing", Frontiers Research Topics, Online.

[9]. J J Hopfield, 1988, "Artificial neural networks", IEEE Circuits and Device Magazine, Vol-4, Issue-5, Pages- 3- 10.

[10]. A Abraham, 2005 "Artificial Neural Networks", Handbook of Measuring System Design, Willey Online Library.

[11]. G Bebis, M Georgiopoulos, 1994, "Feed-forward neural networks", IEEE Potentials, Vol-13, Issue-4, Pages- 27- 31.

[12]. M Buscema, 1998, "Back Propagation Neural Networks", Substance Use & Misuse, Vol-33, Issue-2, Pages- 233-270.

[13]. S Roy, U Chakraborty, 2013 "Introduction to Soft Computing: Neuro-Fuzzy and Genetic Algorithms", Pearson Education India, 1st Edition.

[14]. S Haykin, 1998, "Neural Networks: A Comprehensive Foundation" Prentice-Hall, 2nd Edition.

[15]. A K Jain, J Mao, K K Mohiuddin, 1996, "Artificial neural networks: a tutorial", Computer, Vol-29, Issue-3, Pages- 31-44.

[16]. R Reed, R J Marks, 1999, "Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks", A Bradford Book (MIT Press).

[17]. R Kempter, W Gerstner, J L Hemmen, 1999, "Hebbian learning and spiking neurons", Physical Review E, Vol- 59, Issue-4, Pages- 4498-4514.

[18]. H T Ng, W B Goh, K L Low, 1997, "Feature selection, perceptron learning, and a usability case study for text categorization", Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval, Philadelphia, Pennsylvania, USA Pages- 67-73.

[19]. P Auer, H Burgsteiner, W Maass, 2018, "A learning rule for very simple universal approximators

consisting of a single layer of perceptrons", Neural Networks, Vol-21, Issue-5, Pages- 786-795.

[20]. H Adeli, S-L Hung, 1994, "Machine learning: neural networks, genetic algorithms, and fuzzy systems", John Willey & Sons.

[21]. T Hastie, R Tibshirani, J Friedman, 2008, "Unsupervised Learning", The Elements of Statistical Learning, Springer Series in Statistics, Page- 485-585.

[22]. S Kaski, T Kohonen, 1994, "Winner-take-all networks for physiological models of competitive learning", Neural Networks, Vol-7, Issue-6, Pages- 973-984.

[23]. Y Bengio, 2009, "Learning Deep Architectures for AI", Foundations and Trends in Machine Learning, Vol-2,Issue-1, Pages- 1-127.

[24]. J Ngiam, A Khosla, M Kim, J Nam, H Lee, A Ng, 2011, "Multimodal Deep Learning", Proceedings of 28th International Conference on Machine Learning, WA, USA.

[25]. C Szegedy, A Toshev, D Erhan, 2013, "Deep Neural Networks for Object Detection", Proceedings of the Advances in Neural Information Processing Systems 26.

[26]. H H Aghdam, E J Heravi, 2017, "Guide to convolutional neural networks: a practical approach to traffic sign detection and classification", Springer Publication.

[27]. A Krizhevsky, I Sutskever, G E Hinton, 2012, "ImageNet Classification with Deep Convolutional Neural Networks", Proceedings of Advances in Neural Information Processing Systems 25.

[28]. N Kalchbrenner, E Grefenstette, P Blunsom, 2014, "A Convolutional Neural Network for Modelling Sentences", Proceedings of the 52nd Annual Meeting of the Association of Computational Linguistics, Baltimore,Maryland, Pages- 655-665.

[29]. Y LeCun, Y Bengio, 1998, "Convolutional networks for images, speech, and time series", The handbook of brain theory and networks, Pages- 255-258.

[30]. H Zeng, M D Edwards, G Liu, D K Gifford, 2016, "Convolutional neural network architectures for predicting DNA-protein binding", Bioinformatics, Vol- 32, Issue-12, Pages- 121-127.

[31]. D Strigl, K Kofler, S Podlipnig, 2010, "Performance and Scalability of GPU-based Convolutional Neural Networks", Proceedings 18th Euromicro International Conference on Parallel, Distributed and Network-Based Processing, Pisa, Italy.

[32]. G Hinton, 2009, "Deep Belief Networks", Scholarpedia, Vol-4, Issue-5.

[33]. G E Hinton, S Osindero, Y W Teh, 2006, "A Fast Learning Algorithm for Deep Belief Nets", Neural Computation, Vol-18, Issue-7, Pages- 1527-1554.

[34]. R Salakhutdinov, A Mnih, G Hinton, 2007, "Restricted Boltzmann machines for collaborative filtering", Proceeding of 24th International Conference on Machine Learning, Oregon, USA, Pages- 791-798.

[35]. H Larochelle, Y Bengio, 2008, "Classification using discriminative restricted Boltzmann machines", Proceeding of 25th International Conference on Machine Learning, Finland, Pages- 536-543.

[36]. L R Medsker, L C Jain, 2001, "Recurrent Neural Networks: Design and Applications", CRC Press LLC.

[37]. X Li, X Wu, 2015, "Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition", Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, QLD, Australia.

[38]. M Schuster, K K Paliwal, 1997, "Bidirectional recurrent neural networks", IEEE Transactions on Signal Processing, VOl-45, Issue-11, Pages- 2673-2681.

[39]. O Irsoy, C Cardie, 2014, "Opinion Mining with Deep Recurrent Neural Networks", Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 720–728.

[40]. S Hochreiter, J Schmidhuber, 1997, "Long Short-Term Memory", Neural Computation, Vol-9, Issue-8, Pages- 1735-1780.

[41]. E W Saad, D V Prokhorov, D C Wunsch, 1998, "Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks", IEEE Transactions on Neural Networks, Vol-9, Issue-6, Pages-1456-1470.

[42]. J T Connor, R D Martin, L E Atlas, 1994, "Recurrent neural networks and robust time series prediction", IEEE Transactions on Neural Networks, Vol-5, Issue-2, Pages-240-254.

[43]. Q P Hu, M Xie, S H Ng, G Levitin 2007, "Robust recurrent neural network modelling for software fault detection and correction prediction", Vol-92, Issue-3, Pages- 332-340.[44]. T G Barbounis, J B Theochairs, M C Alexiadis, P S Dokopoulos, 2006, "Long-term wind speed and power forecasting using local recurrent neural network models", IEEE Transactions on Energy Conversion, Vol-21, Issue-1, Pages-273-284.

[45]. E Levin, 1990, "A recurrent neural network: Limitations and training", Neural Networks, Vol-3, Issue-6, Pages- 641-650.

[46]. M Sundermeyer, I Oparin, J-L Gauvain, B Freiberg, R Schluter, H Ney, 2013, "Comparison of feedforward and recurrent neural network language models", Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, BC, Canada.