Nile Journals

Nile Journal of Communication & Computer Science

Volume 3 , Number 1, May 2022

Journal Webpage: https://njccs.journals.ekb.eg

A Survey of Deep Learning Algorithms and its Applications

Arwa E. Abulwafa

Dept. of Computer Eng. & Systems, Faculty of Engineering, Mansoura University, Egypt.

Abstract

Deep learning has exploded in prominence in scientific computing, with its techniques being utilized by a wide range of sectors to solve complicated issues. To perform certain tasks, all deep learning algorithms employ various forms of neural networks. This article looks at how deep learning algorithms function to replicate the human brain and how important artificial neural networks are. Deep learning is a branch of machine learning that aims to get closer to artificial intelligence's core goal. The summary and induction methods of deep learning are mostly used in this study. It begins with an overview of global progress and the current state of deep learning. Second, it discusses the structural principle, characteristics, and several types of traditional deep learning models, including the stacked autoencoder, deep belief network, deep Boltzmann machine, and convolutional neural network. Third, it covers the most recent advances and applications of deep learning in a variety of disciplines, including speech recognition, computer vision, natural language processing, and medical applications. Finally, it discusses deep learning's challenges and potential research areas.

Keywords: Deep learning; Stacked auto encoder; Deep belief networks; Deep Boltzmann machine; Convolutional neural network

1. Introduction

Artificial neural networks are used in deep learning to execute complex computations on enormous volumes of data. It's a sort of machine learning that's based on the human brain's structure and function. Machines are trained using deep learning algorithms that learn from examples. Deep learning is extensively used in industries such as health care, eCommerce, entertainment, and advertising.

Deep learning is nothing more than a collection of classifiers that work together and are based on linear regression and some activation functions. Its foundation is the same as the $W^TX + b$ technique used in traditional statistical linear regression. The only difference is that in deep learning, there are many neural nodes instead of just one, which is known as linear regression in classical statistical learning. A neural network is made up of these neural nodes, and each classifier node is referred to as a neural unit of perception. Another issue worth mentioning is that there are numerous layers between the input and the output in deep learning. The number of neuronal units in a layer might range from hundreds to thousands. The hidden layers and hidden nodes are the layers that exist between the input and the output. Traditional machine learning classifiers have the disadvantage of requiring us to construct a complex hypothesis manually, however with a deep neural network, the hypothesis is created by the network itself, making it a great tool for learning nonlinear correlations.

Machine learning is classified into two stages of development: shallow learning and deep learning. Prior to the reintroduction of deep learning into the research trend in 2006, the research focus was primarily on the shallow learning framework for data processing. In comparison to deep learning, shallow learning will be confined to two non-linear feature conversion layers. Logistic Regression [1-4], Support Vector Machines [5-8], Gaussian Mixture Models [9,10], and other shallow architectures are the most frequent. So far, shallow learning has only been able to solve problems with various constraints quickly and effectively; but it cannot tackle complex problems in the actual world, such as human voices, natural images, visual scenes, and so on. Shallow learning has a restriction that prevents it from processing information in the same way that the human brain does. Hinton et al. [11] proposed a deep belief network (DBN, Deep Belief Network) that was stacked using constrained Boltzmann machines in 2006. (RBM, Restricted Boltzmann Machine). Through unsupervised learning and training, they proposed an unsupervised training algorithm with greedy layer-by-layer. The data was then used to create an initial value for supervised learning. As a result, the deep learning framework was able to solve an issue that shallow learning was unable to handle. As deep learning became more popular, a growing number of scientists and technologists began to focus on the applications of deep learning research, which aided in the advancement of human intelligence.

The study of deep learning is primarily manifested in the organization of numerous world-class artificial intelligence conferences, the formation of a world elite research group, the formation of an enterprise research team, and the ongoing applications of deep learning in artificial intelligence. Deep learning algorithms are constantly being developed, and new records are being made in a variety of data sets. For example, in a test procedure of image classification for 1000 different photos, the image classification error rate reduced to 3.5 per cent after five years of continuous improvement of the deep learning model, which is higher than the accuracy of ordinary people. In reality, employing deep learning to teach machines how to effectively identify and categories photographs was a success. The deep learning model is constantly being updated as the core technology model of artificial intelligence in the big data environment, reflecting the latest research progress of current science and technology, and the deep learning model is constantly being updated as the core technology model of artificial intelligence in the big data environment, reflecting the latest research progress of current science and technology.

2. Related Work

The first step toward neural networks was taken in 1943, when Warren McCulloch, a neurophysiologist, and Walter Pitts, a young mathematician, published a paper on how neurons may work. They proposed an electrical circuit-based neural network. Donald Hebb proposed in

1949 that brain connections became stronger with each usage [12]. In the 1950s, IBM researcher Nathanial Rochester used IBM 704 computers to mimic abstract neural networks [13]. In 1956, four scientists collaborated on the Dartmouth Summer Research Project on Artificial Intelligence, which took place during the summer. John McCarthy, Marvin L. Minsky, Nathaniel Rochester, and Claude E. Shannon were the four scientists. They made a significant contribution to AI research [14].

Following the Dartmouth study in 1957, John Von Neumann claimed that telegraph relays or vacuum tubes may be used to mimic the function of a single neuron. Frank Rosenblatt, a Cornell neurobiologist, began working on the Perceptron in 1958. He was enthralled by the activity of a fly's eye. In a fly's eye, a large part of the preparation that instructs it to flee is done. The Perceptron, which was developed as a result of this research, is the most well-known and widely used neural network today. A single layer perceptron was shown to be useful for classifying a single-valued collection of inputs into one of two categories. The perceptron calculates a weighted sum of the data sources, subtracts a limit, and outputs one of two possible qualities. Bernard Widrow and Marcian Hoff of Stanford developed the ADALINE and MADALINE 1 models in 1959. Multiple ADAptive LINear Elements were used in these models, which gave them their moniker. MADALINE was the first neural network to be used to solve a problem in the real world. It's an adaptive channel for removing echoes from telephone lines. This neuronal structure is still used in the workplace.

Surprisingly, these previous victories led people to exaggerate the capabilities of neural networks, especially given the hardware limitations at the time. The excessive excitement that emanated from the academic and technical disciplines poisoned the writing of the day. As promises were unfulfilled, disillusionment crept in. Similarly, as essayists considered the impact of "figuring machines" on a man, a sense of dread developed. Asimov's arrangement on robots revealed the implications for man's ethics and attributes when machines were capable of performing all of humanity's tasks. Interest in the field was reignited in 1982. Caltech's John Hopfield presented a paper to the National Academy of Sciences 2. His strategy was to use bidirectional wires to create more valuable devices. Previously, there was just one route for neurons to connect. A combined US-Japan Conference on Cooperative/Competitive Neural Networks was also held in 1982. Japan announced a new Fifth-Generation effort on neural networks, while US journals raised concerns that the US would be left behind in the sector (Fifth-Generation processing incorporates computerized reasoning).

The first era used switches and wires, the second era used transistors, the third era used strong state technology such as integrated circuits and higher-level programming dialects, and the fourth era used code generators.) As a result, there was increased subsidizing and, as a result, more field exploration. The American Institute of Physics began a yearly conference called Neural Networks for Computing in 1985. The first International Conference on Neural Networks, held by the Institute of Electrical and Electronics Engineers (IEEE) in 1987, gathered over 1,800 people. Schmidhuber and Hochreiter proposed the Long Short-Term Memory (LSTM) recurrent neural network structure in 1997. In the realm of deep learning, long momentary memory (LSTM) is an artificial recurrent neural network (RNN) architecture [1]. LSTM has feedback connections, unlike normal feedforward neural networks. It not only cycles single information items (such as pictures), but also the entire stream of data (for example, speech or video). Yann LeCun released Gradient-Based Learning Applied to Document Recognition in 1998, which was a significant step forward in data learning [15].

3. Activation Functions

The activation functions that are inspired by human brain firing, i.e., it either fires or doesn't, are another crucial aspect in a neural network. In order to construct nonlinear interactions between the input and output, activation functions are used. This nonlinearity, paired with a large number of neural nodes and layers, resembles the structure of a human brain, which is why it's termed a neural network. Many activation functions exist (some of which are shown in Figure 1(B)). Different activation functions that are often employed, such as Sigmoid, Hyperbolic tangent, and Relu, are depicted in Figure 1. The activation function's job is to abstract and transform data onto a more classifiable plane.

In most cases, the data is closely clustered; the activation function's role is to transform the data onto a different plane, which aids in analyzing the effects of various dimensions in the given situation. The sigmoid activation function, which is utilized in logistic regression, is the greatest and most famous example of the activation function. In fact, the logistic regression (see Figure 1(A)) can be thought of as a single neuronal unit. The sigmoid function's job is to take any input and produce a value between 0 and 1 that can be utilized to solve classification problems. One hidden layer neural network with three hidden neural units in the hidden layer and one in the output layer is shown in Figure 1(C). The logistic regression model is comparable to this hidden unit. The distinction is that the input for the following layer comes from the one before it. We plotted a description of more than one hidden layer and more than one neuronal unit in each layer in Figure 1(D). The neural network can have several levels, and each layer can contain any number of neural units, as shown in Figure 1.

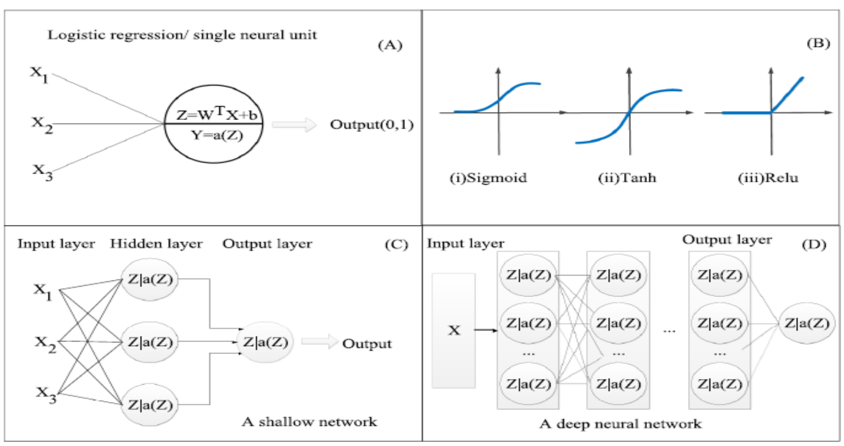

Figure 1 Types of Activation Functions

4. Parameter learning

Deep learning classifiers, like typical machine learning classifiers, need the use of mathematical methods such as gradient descent to learn parameters. When learning parameters for convex functions, the gradient descent approach comes in handy. If a function has only one absolute minimum or maximum, it is said to be convex. If the function is convex, learning the parameters is simple; otherwise, converting a nonconvex function to a convex function requires some mathematical trickery. A convex optimization problem is another name for this problem.

However, in terms of physics, neural network optimization is a non-convex problem. It has a large number of optimum (minima/maxima) positions. Learning is accomplished by minimizing the difference between the expected and actual values.

5. How Deep Learning Algorithms Work?

While deep learning algorithms use self-learning representations, they rely on artificial neural networks (ANNs) that mimic how the brain processes information. Algorithms leverage unknown elements in the input distribution to extract features, organize objects, and uncover important data patterns throughout the training phase. This happens at various levels, employing the algorithms to develop the models, much like training machines for self-learning. Several algorithms are used in deep learning models. While no network is flawless, certain algorithms are better suited to specific jobs than others. To select the best, it's necessary to have a thorough understanding of all primary algorithms.

6. Types of Deep Learning Algorithms

Deep learning algorithms can handle practically any type of data and require a lot of processing power and data to solve complex problems. Let's take a look at the top ten deep learning algorithms. The following is a list of the top ten most widely used deep learning algorithms:

1. Convolutional Neural Networks (CNNs)
2. Long Short-Term Memory Networks (LSTMs)
3. Recurrent Neural Networks (RNNs)
4. Generative Adversarial Networks (GANs)
5. Radial Basis Function Networks (RBFNs)
6. Multilayer Perceptrons (MLPs)
7. Self-Organizing Maps (SOMs)
8. Deep Belief Networks (DBNs)
9. Restricted Boltzmann Machines (RBMs)
10. Autoencoders

6.1. Convolutional Neural Networks (CNNs)

CNNs [16], also known as ConvNets, are multilayer neural networks that are primarily used for image processing and object detection. In 1988, Yann LeCun created the first CNN, which he called LeNet. It could recognize characters such as ZIP codes and numerals. CNNs are commonly used to detect abnormalities, identify satellite photos, interpret medical imaging, forecast time series, and identify anomalies. Convolutional Neural Networks (CNN) are mostly employed in image processing. It assigns weights and biases to different items in the image and distinguishes them. In comparison to other classification methods, it requires less preparation. In order to capture the spatial and temporal dependencies in a picture, CNN employs relevant filters [17, 18]. LeNet, AlexNet, VG-GNet, GoogleNet, ResNet, and ZFNet are some of the different CNN architectures. Object detection, semantic segmentation, and captioning are just a few of the applications that CNNs are utilized for.

Multiple layers process and extract features from data in CNNs: CNN features a convolution layer that consists of many filters that perform the convolution operation. CNNs have a Rectified

Linear Unit (ReLU) layer that performs operations on elements. A rectified feature map is the result. The rectified feature map is fed into a pooling layer after that. Pooling is a down sampling procedure that decreases the feature map's dimensionality. By flattening the two-dimensional arrays from the pooled feature map, the pooling layer turns them into a single, long, continuous, linear vector. When the flattened matrix from the pooling layer is given as an input, a fully connected layer arises, which classifies and labels the images. Figure 2 is an example of a CNN-processed image.

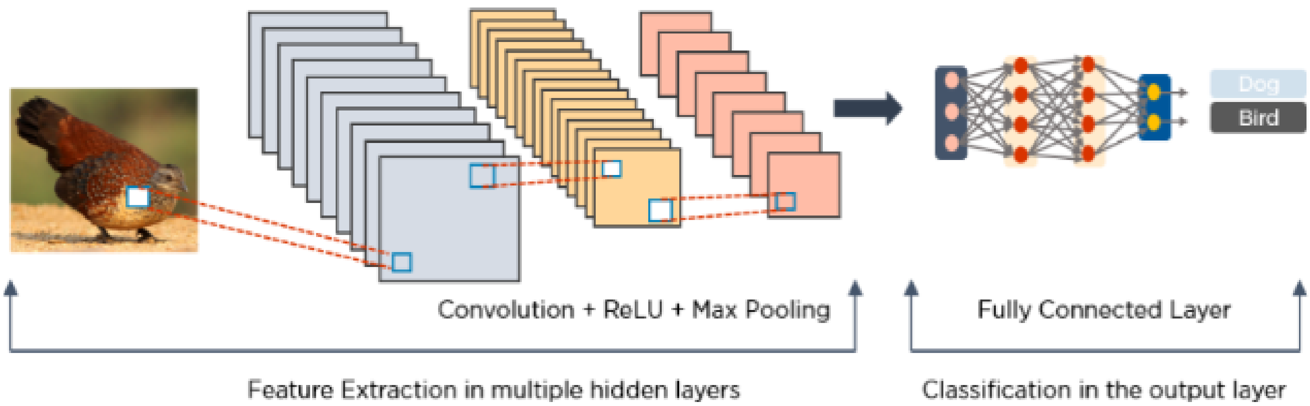

Figure 2 Example of Convolutional Neural Networks (CNNs)

6.2. Long Short-Term Memory Networks (LSTMs)

Long-term dependencies can be learned and remembered using LSTMs [19], which are a form of Recurrent Neural Network (RNN). The default behavior is to recall past information over long periods of time. LSTMs keep track of data throughout time. Because they remember past inputs, they are valuable in time-series prediction. Four interacting layers communicate in a unique way in LSTMs, which have a chain-like structure. LSTMs are commonly employed for voice recognition, music creation, and pharmaceutical research, in addition to time-series predictions. First, they forget about the portions of the previous state that aren't significant. They then update the cell-state values selectively. Finally, the state of some portions of the cell's output. Figure 3 is a diagram illustrating how LSTMs work.

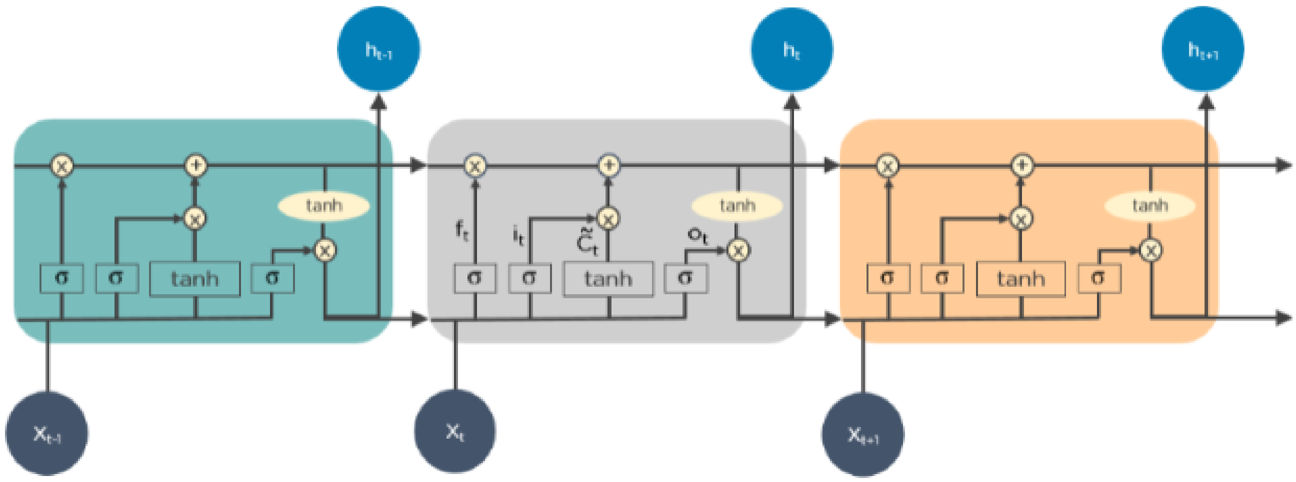

Figure 3 Long Short-Term Memory Networks (LSTMs)

6.3. Recurrent Neural Networks (RNNs)

The outputs from previous states are given as input to the present state in recurrent neural networks (RNN) [20]. RNN's hidden layers have the ability to remember information. The output created in the previous state is used to update the concealed state. RNN may be used to predict time series since it has Long Short-Term Memory [19], which allows it to remember prior inputs. The outputs from the LSTM can be given as inputs to the current phase since RNNs contain connections that create directed cycles. The LSTM's output becomes an input to the current phase, and its internal memory allows it to remember prior inputs. Image captioning, time-series analysis, natural-language processing, handwriting identification, and machine translation are all common uses for RNNs. Figure 4 shows how an RNN looks like after it's fully unfolded.

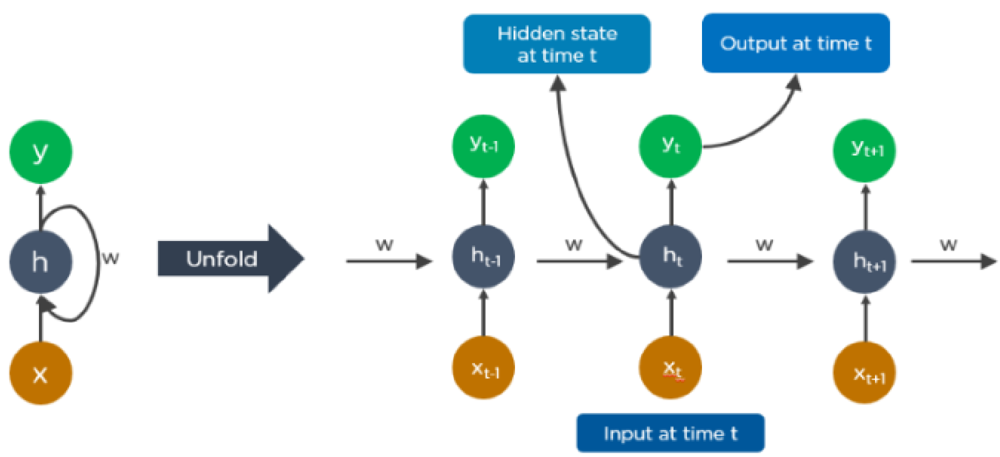

Figure 4 Recurrent Neural Networks (RNNs)

At time t-1, the output feeds into the input at time t. The output at time t feeds into the input at time t+1 in the same way. RNNs can handle any length of the input. The computation takes into consideration historical data, and the model size does not grow in proportion to the input size. An example of how Google's autocompleting feature works is illustrated in Figure 5.

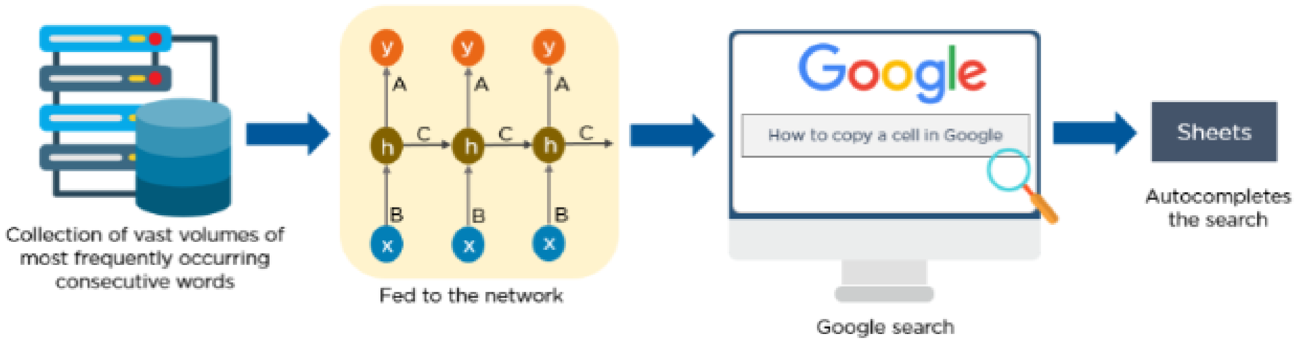

Figure 5 Recurrent Neural Networks (RNNs) for Google

6.4. Generative Adversarial Networks (GANs)

Ian Goodfellow spoke on Generative Adversarial Networks (GAN). It is made up of two networks: a Generator network and a Discriminator network. The generator creates the content, while the discriminator checks it for accuracy. The generator makes natural-looking images, and the discriminator determines whether or not they are natural. The GAN algorithm is a two-player minimax algorithm. Convolutional and feed-forward Neural Nets are used in GANs [21].

GANs are deep learning generative algorithms that generate new data instances that are similar to the training data. GAN is made up of two parts: a generator that learns to generate fake data and a discriminator that learns from that data. GANs have become increasingly popular over time. They can be used to improve astronomy photographs as well as to imitate gravitational lensing for dark matter investigations. GANs are used by video game producers to upscale low-resolution, 2D graphics in older games by using image training to recreate them in 4K or greater resolutions. GANs aid in the creation of realistic images and cartoon characters, as well as the creation of photographs of human faces and the rendering of 3D objects.

The discriminator learns to tell the difference between the bogus data generated by the generator and the genuine sample data. The generator generates fraudulent data during early training, and the discriminator quickly learns to recognize it as such. To update the model, the GAN delivers the results to the generator and discriminator. Figure 6 is a diagram illustrating how GANs work.

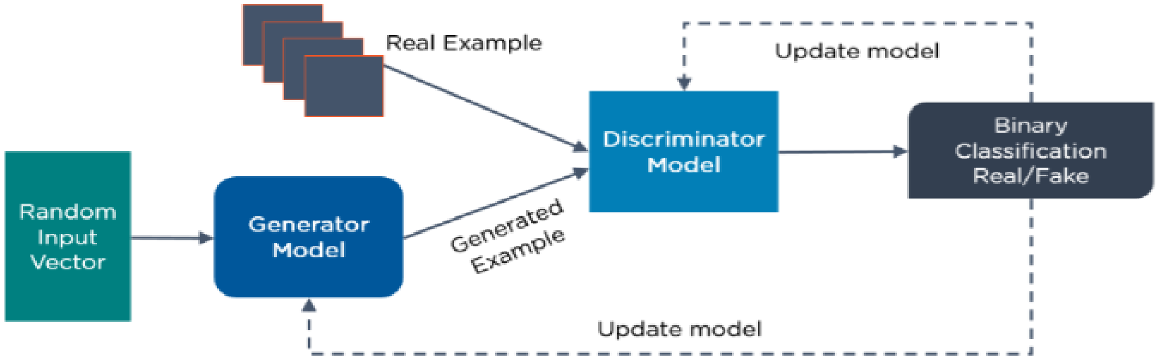

Figure 6 Generative Adversarial Networks (GANs)

6.5. Radial Basis Function Networks (RBFNs)

Radial basis functions are used as activation functions in RBFNs [22], which are a sort of feedforward neural network. They are used for classification, regression, and time-series prediction and have an input layer, a hidden layer, and an output layer. The similarity of the input to examples from the training set is used by RBFNs to do classification. The input layer of RBFNs is fed via an input vector. They have an RBF neuron layer. The output layer has one node per category or class of data, and the function finds the weighted total of the inputs. The Gaussian transfer functions, which have outputs that are inversely proportional to the distance from the neuron's center, are found in the neurons in the hidden layer. The output of the network is a linear combination of the radial-basis functions of the input and the parameters of the neuron. Consider the RBFN shown in Figure 7.

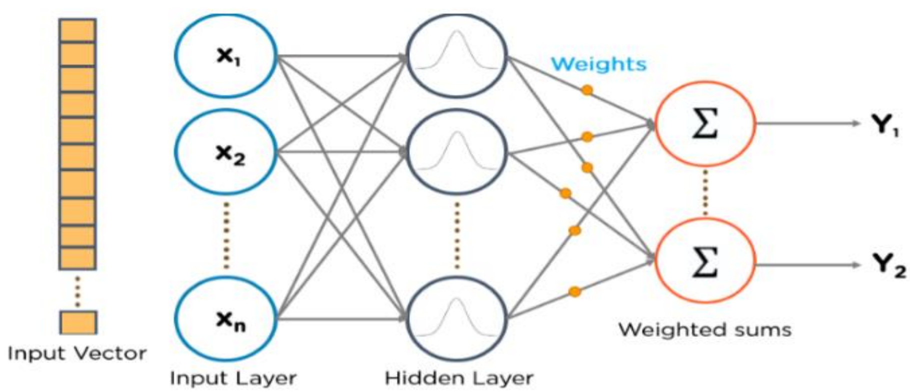

Figure 7 Radial Basis Function Networks (RBFNs)

6.6. Multilayer Perceptrons (MLPs)

MLPs [23] are a great starting point to learn more about deep learning. MLPs are a type of feedforward neural network that includes multiple layers of perceptron with activation functions. MLPs are made up of two fully connected layers: an input layer and an output layer. They have the same set of input and output layers, but they can have several hidden layers, and they can be used to create speech recognition, image recognition, and machine translation software.

The data is fed into the network's input layer using MLPs. The signal flows in one way because the layers of neurons are connected in a graph. MLPs use the weights that exist between the input layer and the hidden layers to compute the input. To decide which nodes to fire, MLPs use activation functions. ReLUs, sigmoid functions, and tanh are all activation functions. From a training data set, MLPs train the model to grasp the correlation and learn the dependencies between the independent and target variables. An MLP is shown in Figure 8 as an example. To classify photos of cats and dogs, the diagram computes weights and bias and applies appropriate activation functions.

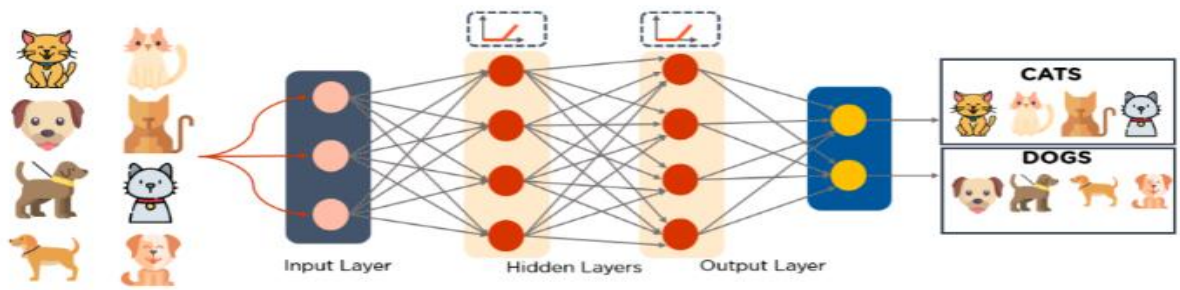

Figure 8 Multilayer Perceptrons (MLPs)

6.7. Self-Organizing Maps (SOMs)

Professor Teuvo Kohonen created SOMs [24], which enable data visualization by using self-organizing artificial neural networks to reduce the dimensions of data. The problem of humans being unable to visualize high-dimensional data is addressed through data visualization. SOMs are designed to assist people in comprehending this multi-dimensional data. SOMs use a vector at random from the training data to initialize weights for each node. SOMs look at each node to see which weights are most likely to be the input vector. The Best Matching Unit is the winning node (BMU).

The BMU's neighborhood is discovered through SOMs, and the number of neighbors decreases with time. The sample vector is given a winning weight using SOMs. The weight of a node changes as it gets closer to a BMU. The farther away a neighbor is from the BMU, the less it learns from it. For N iterations, SOMs repeat step two. A diagram of an input vector with various colors is shown in Figure 9. This information is fed into a SOM, which converts it to 2D RGB values. Finally, it categorizes and divides the various colors.

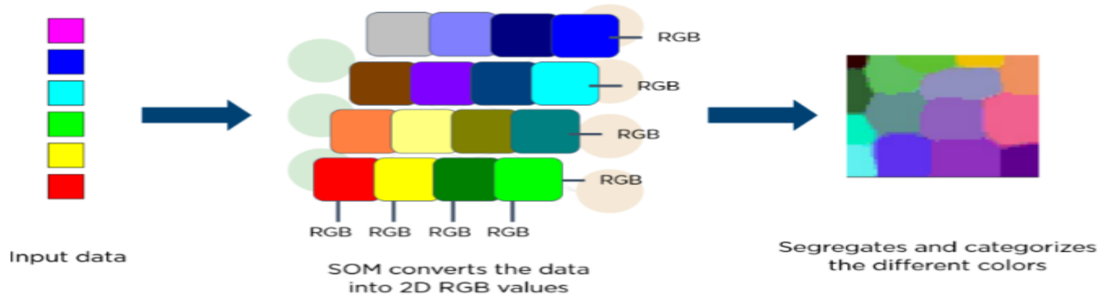

Figure 9 Self-Organizing Maps (SOMs)

6.8. Deep Belief Networks (DBNs)

The first step for training the deep belief network is to learn features using the first layer. Then use the activation of trained features in the next layer. Continue this until the final layer. Restricted Boltzmann Machines (RBM) is used to train layers of the Deep Belief Networks (DBNs), and the feed-forward network is used for fine-tuning. DBN learns hidden pattern globally, unlike other deep nets where each layer learns complex patterns progressively [25].

DBNs are generative models that consist of multiple layers of stochastic, latent variables. The latent variables have binary values and are often called hidden units. DBNs are a stack of Boltzmann Machines with connections between the layers, and each RBM layer communicates with both the previous and subsequent layers. Deep Belief Networks (DBNs) are used for image-recognition, video-recognition, and motion-capture data. Greedy learning algorithms train DBNs. For learning the top-down, generative weights, the greedy learning method employs a layer-by-layer approach. On the top two buried layers, DBNs do Gibbs sampling steps. The RBM defined by the top two hidden layers is sampled in this stage. DBNs use a single pass of ancestral sampling through the rest of the model to generate a sample from the visible units. DBNs learn that a single bottom-up pass can infer the values of the latent variables in each layer. An example of DBN architecture is shown in Figure10:

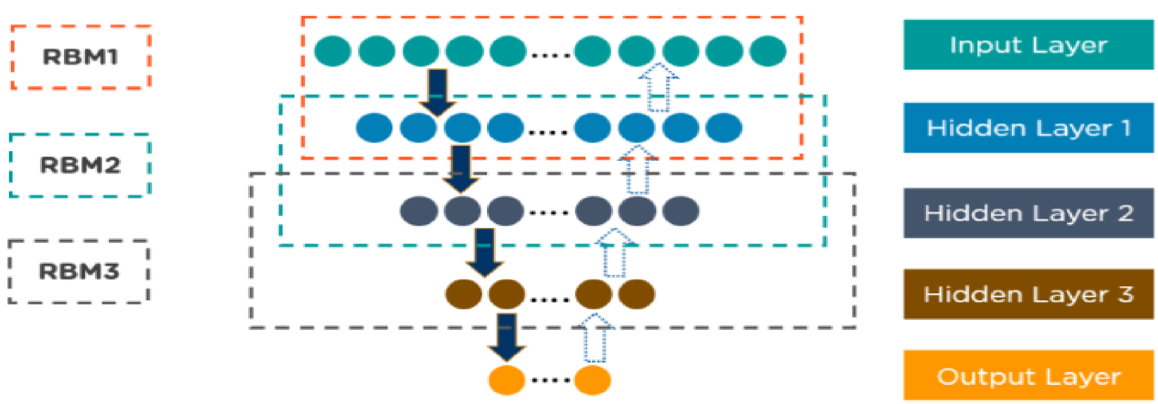

Figure 10 Example of Deep Belief Networks (DBNs)

6.9. Restricted Boltzmann Machines (RBMs)

RBMs [26] are randomized neural networks developed by Geoffrey Hinton that can learn from a probability distribution across a collection of inputs. For dimensionality reduction, classification, regression, collaborative filtering, feature learning, and topic modelling, this deep learning algorithm is utilized. RBMs are the fundamental components of DBNs. RBMs are divided into two layers: visible and hidden units. Every visible unit is linked to every hidden unit. RBMs have no output nodes and have a bias unit that is coupled to all of the visible and hidden units.

RBMs have two phases: forward pass and backward pass. RBMs accept the inputs and translate them into a set of numbers that encodes the inputs in the forward pass. RBMs combine every input with individual weight and one overall bias. The algorithm passes the output to the hidden layer. In the backward pass, RBMs take that set of numbers and translate them to form the reconstructed inputs. RBMs combine each activation with individual weight and overall bias and pass the output to the visible layer for reconstruction. At the visible layer, the RBM compares the reconstruction with the original input to analyze the quality of the result. Figure 11 illustrates how RBMs function:

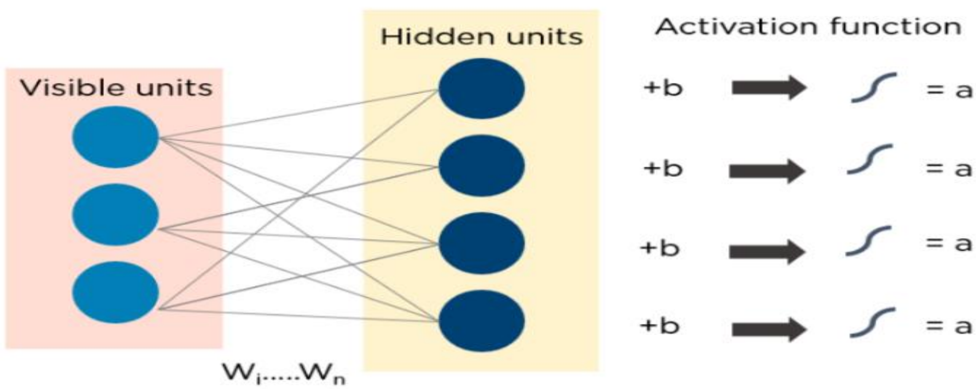

Figure 11 Restricted Boltzmann Machines (RBMs)

6.10. Autoencoders

Autoencoders [27] are a kind of feedforward neural network where the input and output are both the same. In the 1980s, Geoffrey Hinton invented autoencoders to overcome unsupervised learning difficulties. They're neural networks that have been trained to repeat data from the input layer to the output layer. Autoencoders are utilized in a variety of applications, including drug discovery, popularity prediction, and image processing. The encoder, the code, and the decoder are the three essential components of an autoencoder. Autoencoders are designed to take in information and turn it into a different form. Then they try to recreate the original input as closely as possible. When a digit's image isn't clear, it's sent into an autoencoder neural network. Autoencoders encode the image first, then compress the data into a smaller form. Finally, the image is decoded by the autoencoder, which produces the reconstructed image. Figure 12 shows how autoencoders work:

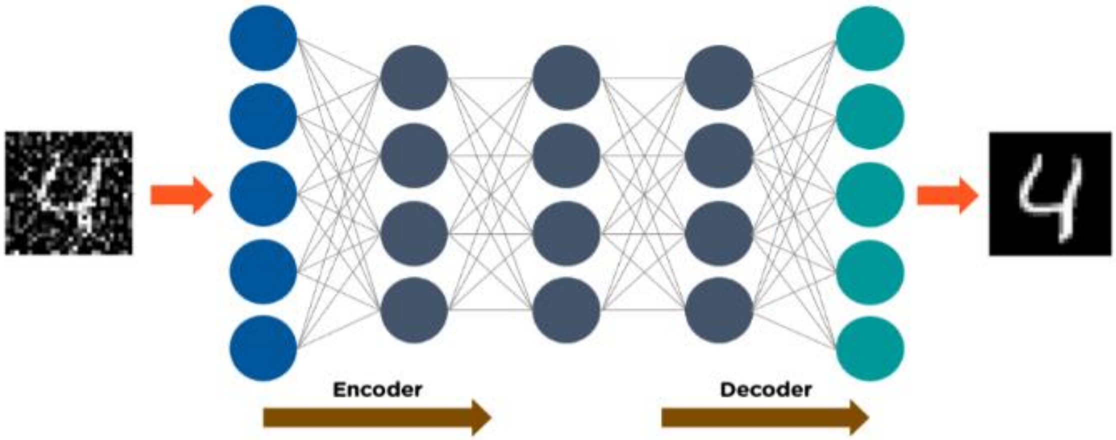

Figure 12 Autoencoders

Autoencoders are used to reduce the dimension of data, as well as to solve problems like novelty detection and anomaly detection. The first layer in an autoencoder is produced as an encoding layer and then transposed as a decoder. Then, using the unsupervised method, teach it to duplicate the input. Fix the weights of that layer after training. Then go to the next layer until all of the deep net's layers have been pre-trained. Then go back to the original issue (Classification/Regression) that we want to solve with deep learning and optimize it using stochastic gradient descent, starting with the weights learned during pre-training.

Autoencoder network consists of two parts [28]. The input is translated to a latent space representation by the encoder, which can be denoted in (1):

$$h = f(x) \qquad (1)$$

The input is reconstructed from the latent space representation by the decoder, which can be denoted in (2):

$$r = g(h) \qquad (2)$$

In essence, autoencoders can be described in (3). r is the decoded output which will be similar to input x:

$$g\big(f(x)\big) = r \qquad (3)$$

7. Applications of deep learning

In this section applications of deep learning in various areas will be covered. Following are the various applications of Deep learning.

7.1. Natural language processing

Deep learning is used in many domains in natural language, including voice translation, machine translation, computer semantic comprehension, and so on. In truth, deep learning has only been successful in two fields: image processing and natural language processing. In 2012, Schwenk et al. [29] suggested a Deep Neural Network-based phrase-based statistical machine translation system (DNN). It learned meaningful translation probabilities for unseen sentences that were not included in the training set. Dong et al. [30] introduced a new AdaMC (Adaptive Multi-Compositionality) layer in the recursive neural network in 2014. This model included many composition functions, each of which was adaptively chosen based on the input parameters.

Tang et al. [31] presented a DNN for sentiment analysis on Twitter data in 2014. Google introduced its deep learning-based Word Lens identification engine in 2015, which used word lenses in real-time call translation and video translation. This technology could not only read the words in real-time, but it could also translate them into the target language. Furthermore, the translation job might be done over the phone without the need for networking. More than a visual translation of 20 languages might be done with today's technology. In addition, Google offered a Gmail automatic mail reply feature that used a deep learning model to extract email content and analyze it semantically. Finally, a response is generated depending on the semantic analysis. This method differs significantly from standard e-mail auto-responder capabilities.

7.2. Speech recognition

The researchers put in a lot of effort to achieve Human-Computer Interaction. Davis and others at the Bell Institute succeeded in developing the world's first experimental system that can recognize 10 English digital pronunciations in 1952. Speech recognition research has a few decades of history, and voice recognition was the dictator in some fields, as it was named one of the top 10 events in computer development by the US press. Speech recognition technology has progressed considerably during the last two decades. A huge number of voice recognition devices or apps have begun to transfer from the lab to the market as the deep learning model improves.

Baidu released Deep Speech in 2014, a voice recognition system that uses deep learning technology and can attain an accuracy of 8% in noisy conditions. The phrase recognition error rate of Baidu's Deep Speech 2 was decreased to 3.7 per cent in February 2016. You et al. [32] introduced a node pruning strategy for reconstructing the DNN in 2015, which resulted in a novel bottleneck characteristic. In addition, Maas et al. [33] investigated alternative DNN architectures and settings for training very big voice data in 2017. They discovered that simple architecture and simple optimization strategies outperformed the other, more sophisticated models.

7.3. Medical applications

Deep learning's forecasting function, as well as its automatic feature detection, making it a preferred tool for disease diagnosis. Deep learning applications in medicine, whether in the use of frequency or in the use of species, are always improving. Li et al. [34] proposed the use of customized CNN to categorize lung image patches in 2014. To avoid overfitting, this model uses the dropout method and a single-volume structure. Li et al. [35] introduced a DNN-based framework for distinguishing the identity phases of Alzheimer's Disease (AD) using MRI and PET scan data in 2015. Srinukunwattana et al.

[36] introduced a spatially constrained convolutional neural network (SC-CNN) in 2016 to assess histopathology images and identify malignant cells' nuclei. Their SC-CNN method outperformed the traditional feature classification method in terms of accuracy. Google created a visual technology for detecting early-stage ocular disorders in 2016. They collaborated with the Moorfields Eye Hospital to give early preventative measures for diseases like diabetic retinopathy and age-related macular degeneration. A month later, Google applied deep learning techniques to create a head and neck cancer radiotherapy approach that could effectively regulate the patient's radiotherapy time while also minimizing the radiotherapy of the damage. Deep learning in the realm of precision medical care will become more important with the further development of deep learning technologies.

7.4. Computer vision

Artificial intelligence's most important application is computer vision [37]. It's an interdisciplinary field that studies how computers can understand digital images or videos to a high degree. For target object detection, tracking, measuring, and other visual difficulties, it can employ computers and cameras to replace the human eye. After that, take care of the graphics so that the computer can perform image processing beyond the human eye's capabilities. Baidu said in 2015 that it would improve ImageNet picture classification recognition performance. For the first time in computer performance, the image identification error rate was less than 5% in the test, which was beyond the human level mistake. Computer vision is a broad phrase that encompasses a wide range of academic topics. Followings are some well-known directions which comes under umbrella of computer vision.

1. Image segmentation
2. Face recognition
3. Object detection
4. Image semantic segmentation
5. Video object segmentation
6. Background/foreground separation

7.5. Deep learning on graphs

Researchers have been working on novel strategies for learning patterns from graph-structured data in recent years. Deep learning on graphs has been used to solve a diverse range of challenges. In 2018, for example, Qiu et al. [38] introduced an end-to-end deep learning framework for influential user prediction that used the user's local graph structure as input. Researchers have been working on novel strategies for learning patterns from graph-structured data in recent years. Deep learning on graphs has been used to solve a diverse range of challenges. In 2018, for example, Qiu et al. [38] introduced an end-to-end deep learning framework for influential user prediction that used the user's local graph structure as input.

Monti et al. [39] have introduced a geometric deep learning framework based on a convolutional neural network and a recurrent neural network in 2017. By forecasting accurate ratings in the recommendation system, our model assisted with the matrix completion problem. In 2015, Duvenaud et al. [40] introduced a deep learning model for producing chemical characteristics based on convolutional neural networks, which solved the deep learning and graphs problem in chemistry. Gilmer et al. [41] created a deep learning framework for chemical property prediction based on a message-passing neural network in 2017. Kearnes et al. [42] built a molecular graph convolutional neural network for undirected molecular graphs in 2016. In 2018, You et al. [43] proposed a goal-directed graph generation model based on reinforcement learning called the Graph Convolutional Policy Network (GCPN). The approach has been widely used in chemistry and drug development, where novel molecules must be discovered within certain chemical parameters such as drug-likeness and synthetic accessibility.

Cao and Kipf [44] introduced the Generative Adversarial Network (GAN) in 2018, which is based on a likelihood-free generative model. This model could also generate compounds with specific molecular characteristics. Coley et al. [45] used a graph convolutional network on an undirected molecular graph to address the molecular graph representation problem in 2017. They took into account atom and bond attributes, atom neighbor, radii, and other parameters in addition to the molecular graph structural attribute. Xie et al. [46] developed the Crystal Graph Convolutional Neural Network framework in 2018, which was capable of learning material attributes from the crystal atomic link structure, which might be extremely useful in new material design. Ktena et al. [47] applied graph convolutional neural networks to predict graph similarity in identity brain diseases in 2017. It was usual practice to treat complex diseases by administering a large number of medications at once that targeted complex diseased proteins.

However, when another medicine is present, the effect of changing one drug is often not noticed in clinical trials. In 2018, Zitnik et al. [48] presented Decagon, a graph convolutional network-based framework, to overcome this challenge. Decagon was able to forecast what side effects two medications could have on a patient. Parisot et al. [49,50] employed graph convolutional networks to predict brain illness in 2017 and 2018. Assouel et al. [51] also suggested a conditional graph generative model in 2018.

7.6. Intelligent transportation system

Smart cities are the research emphasis of the twenty-first century [52, 53], and intelligent transportation systems (ITS) are at the heart of them. Throughout history, transportation systems have served as the backbone of every country. According to a report published in 2011 by Zhang et al. [53], 40% of the world's population spends at least one hour on the road every day. Vehicles are becoming more difficult to control without the assistance of technology as the world's population grows. Citizens of the United States used 181,541 public transportation vehicles in 2019, taking 9.9 billion trips totaling 55.8 billion kilometers. It appears that smart transportation is in high demand throughout the world's major cities.

Letters and digits to sound photos and movies are all examples of transportation data. For example, image recognition and video surveillance are required for an autonomous passenger counter that predicts revenue collection. We need to examine which route people took the most and at what time, in addition to the automatic passenger counter. It requires GPS and road map data. Non-human created data, such as 'weather,' is occasionally required. These disparate data originate from a variety of sensors located in various areas, such as traffic lights, autos, and so on.

Destination prediction, traffic signal control, demand prediction, traffic flow prediction, transportation mode, and combinatorial optimization are the primary problems that ITS works on. Veras et al. [54] published work in 2019 that shows how deep learning has been used to solve the following difficulties.

1. Destination prediction
2. Demand Prediction
3. Traffic Flow Prediction
4. Travel Time Estimation
5. Predicting Traffic Accident Severity
6. Predicting the Mode of Transportation
7. Trajectory Clustering
8. Navigation
9. Demand Serving
10. Traffic Signal Control
11. Combinatorial Optimization

8. Conclusion

Deep learning technology is used in a variety of disciplines and research areas, including speech recognition, image processing, graphs, medicine, and computer vision. It is one of the most rapidly evolving and adaptable technologies in history. The issues arise from the existence of large amounts of complex data, which makes it difficult to use deep learning to address the problem successfully. Building an adequate deep learning model in the context of an application is becoming increasingly difficult. Although deep learning is still in its infancy and there are still issues to be resolved, it has demonstrated a great learning ability. In the realm of future artificial intelligence, it is still a hot study topic. This paper has gone over some of the more well-known advances in deep learning and their applications in a variety of fields. Finally, deep learning applications are discussed in more detail. Because there are so many scientific problems that are being solved every day, deep learning can occasionally obtain surprising and better results in fields like image processing and diabetic retinopathy diagnosis, which is exceedingly difficult to diagnose by human experts. Diabetic retinopathy diagnosis is, in truth, nothing more than an application of image processing. As a result, a breakthrough solution in one discipline may be a game-changer in another. Deep learning is gaining a lot of traction, and new applications and technologies are being developed every day. Following are a few active study fields that, based on our little understanding, will continue to receive attention in the near future. (1) Generative models based on deep neural networks, such as Generative adversarial networks, (2) Deep learning for non-Euclidean data, such as Deep learning for graphs, Geometric deep learning, and Hyperbolic neural networks, (3) Deep Learning for spatiotemporal data mining, and (4) How to improve the structures and algorithms of a deep neural network model, among other topics.

9. References

[1] D.A. Freedman, "Statistical Models: Theory and Practice", Cambridge University Press, 2009.

[2] C. Mood, "Logistic regression: Why we cannot do what we think we can do, and what we can do about it", European Sociological Review, vol. 26, no. 1, pp. 67-82, 2010.

[3] D.G. Kleinbaum and M. Klein, "Analysis of matched data using logistic regression", Logistic Regression: A Self-Learning Text, Springer, pp. 227-265, 2002.

[4] D.W. Hosmer Jr, S. Lemeshow and R.X. Sturdivant, "Applied Logistic Regression", John Wiley & Sons, vol. 398, 2013.

[5] R. Soentpiet, "Advances in Kernel Methods: Support Vector Learning", MIT press, 1999.

[6] M.A. Hearst, S.T. Dumais, E. Osuna, J. Platt and B. Scholkopf, “Support Vector Machines”, IEEE Intelligent Systems and their Applications, vol. 13, no. 4, pp. 18-28, 1998.

[7] I. Steinwart and A. Christmann, “Support Vector Machines”, Springer Science & Business Media, 2008.

[8] N.N. Schraudolph, “Fast curvature matrix-vector products for second-order gradient descent”, Neural computation, vol. 14, no. 7, pp. 1723-1738, 2002.

[9] S.Z. Li, “Encyclopedia of Biometrics: I-Z”, Springer Science & Business Media, vol. 2, 2009.

[10] J.J. Verbeek, N. Vlassis and B. Kröse, “Efficient greedy learning of Gaussian mixture models”, Neural Computation, vol. 15, no. 2, pp. 469-485, 2003.

[11] G.E. Hinton, S. Osindero and Y.-W. Teh, “A fast learning algorithm for deep belief nets, Neural Computation, vol. 18, no. 7, pp. 1527-1554, 2006.

[12] D.O. Hebb, “The organization of behavior; a neuropsychological theory”, A Wiley Book in Clinical Psychology, vol. 62, pp. 78, 1949.

[13] D. Crevier, “AI: The Tumultuous History of the Search for Artificial Intelligence”, Basic Books, Inc., 1993.

[14] J. McCarthy, M.L. Minsky, N. Rochester and C.E. Shannon, “A proposal for the dartmouth summer research project on artificial intelligence”, 1955, AI magazine, vol. 27, no. 4, pp. 12-12, 2006.

[15] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, “Gradient-based learning applied to document recognition”, Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, 1998.

[16] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai and T. Chen, “Recent advances in convolutional neural networks, Pattern Recognition, vol. 77, pp. 354-377, 2018.

[17] Q. V. Le, “A tutorial on deep learning part 2: Autoencoders, convolutional neural networks and recurrent neural networks”, Google Brain, vol. 20, pp. 1-20, 2015.

[18] R. Yamashita, M. Nishio, R. Do, and K. Togashi, “Convolutional neural networks: an overview and application in radiology”, Insights into imaging, vol. 9, no. 4, pp. 611-629, 2018. doi: 10.1007/s13244-018-0639-9.

[19] B. Lindemann, T. Müller, H. Vietz, N. Jazdi and M. Weyrich, “A survey on long short-term memory networks for time series prediction”, Proceedings of CIRP, vol. 99, pp.650-655, 2021.

[20] F. M. Bianchi, E. Maiorino, M. C. Kampmeyer, A. Rizzi, and R. Jenssen, “An overview and comparative analysis of recurrent neural networks for short term load forecasting”, arXiv preprint arXiv:1705.04378, 2017.

[21] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets”, Advances in neural information processing systems, vol. 27, pp. 2672-2680, 2014.

[22] A. Tavakkoli, “Foreground-background segmentation in video sequences using neural networks”, Intelligent Systems: Neural Networks and Applications, 2005.

[23] H. Alla, L. Moumoun and Y. Balouki, “A Multilayer Perceptron Neural Network with Selective-Data Training for Flight Arrival Delay Prediction”, Scientific Programming, 2021.

[24] D. Miljković, “Brief review of self-organizing maps”. In 2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pp. 1061-1066, 2017.

[25] R. Salakhutdinov and G. Hinton, “Semantic hashing”, International Journal of Approximate Reasoning, vol. 50, no. 7, pp. 969-978, 2009. doi: 10.1016/j.ijar.2008.11. 006.

[26] U. Fiore, F. Palmieri, A. Castiglione and A. De Santis, “Network anomaly detection with the restricted Boltzmann machine”, Neurocomputing vol. 122, pp. 13-23, 2013.

[27] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol and L. Bottou, “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion”, Journal of machine learning research, vol. 11, no. 12, 2010.

[28] N. Hubens, “Deep inside: Autoencoders - towards data science”, vol. 25, 2018.

[29] H. Schwenk, “Continuous space translation models for phrase-based statistical machine translation”, Proceedings of COLING 2012: Posters, 2012, pp. 1071-1080.

[30] L. Dong, F. Wei, M. Zhou and K. Xu, “Adaptive multi-compositionality for recursive neural models with applications to sentiment analysis”, Proceedings of the National Conference on Artificial Intelligence, vol. 2, pp. 1537-1543, 2014.

[31] D. Tang, F. Wei, B. Qin, T. Liu and M. Zhou, “Coooolll: A deep learning system for twitter sentiment classification”, Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), 2014, pp. 208-212.

[32] Y. You, Y. Qian, T. He and K. Yu, “An investigation on DNN-derived bottleneck features for GMM-HMM based robust speech recognition”, Proceedings of 2015 IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP), IEEE, 2015, pp. 30-34.

[33] A.L. Maas, P. Qi, Z. Xie, A.Y. Hannun, C.T. Lengerich, D. Jurafsky and A.Y. Ng, “Building DNN acoustic models for large vocabulary speech recognition”, Computer Speech & Language, vol. 41, pp. 195-213, 2017.

[34] Q. Li, W. Cai, X. Wang, Y. Zhou, D.D. Feng and M. Chen, “Medical image classification with convolutional neural network”, Proceedings of 2014 13th International Conference on Control Automation Robotics & Vision (ICARCV), IEEE, 2014, pp. 844-848.

[35] F. Li, L. Tran, K.-H. Thung, S. Ji, D. Shen and J. Li, “A robust deep model for improved classification of AD/MCI patients”, IEEE journal of biomedical and health informatics, vol. 19, no. 5, pp. 1610-1616, 2015.

[36] K. Sirinukunwattana, S.E.A. Raza, Y.-W. Tsang, D.R. Snead, I.A. Cree and N.M. Rajpoot, “Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images”, IEEE transactions on medical imaging, vol. 35, no. 5, pp. 1196-1206, 2016.

[37] I. Brilakis, and C. T. M. Haas, “Infrastructure computer vision”, Butterworth-Heinemann, 2019.

[38] J. Qiu, J. Tang, H. Ma, Y. Dong, K. Wang and J. Tang, “Deepinf: Social influence prediction with deep learning”, Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 2110-2119.

[39] F. Monti, M. Bronstein and X. Bresson, “Geometric matrix completion with recurrent multi-graph neural networks”, Advances in Neural Information Processing Systems, pp. 3697-3707, 2017.

[40] D.K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik and R.P. Adams, “Convolutional networks on graphs for learning molecular fingerprints”, Advances in Neural Information Processing Systems, pp. 2224-2232, 2015.

[41] J. Gilmer, S.S. Schoenholz, P.F. Riley, O. Vinyals and G.E. Dahl, “Neural message passing for quantum chemistry”, arXiv preprint arXiv:1704.01212, 2017.

[42] S. Kearnes, K. McCloskey, M. Berndl, V. Pande and P. Riley, “Molecular graph convolutions: moving beyond fingerprints”, Journal of computer-aided molecular design, vol. 30, no. 8, pp. 595-608, 2016.

[43] J. You, B. Liu, Z. Ying, V. Pande and J. Leskovec, “Graph convolutional policy network for goal-directed molecular graph generation”, Advances in Neural Information Processing Systems, pp. 6410-6421, 2018.

[44] N. De Cao and T. Kipf, “MolGAN: An implicit generative model for small molecular graphs”, arXiv preprint arXiv:1805.11973, 2018.

[45] C.W. Coley, R. Barzilay, W.H. Green, T.S. Jaakkola and K.F. Jensen, “Convolutional embedding of attributed molecular graphs for physical property prediction”, Journal of chemical information and modeling, vol. 57 no. 8, pp. 1757-1772, 2017.

[46] T. Xie and J.C. Grossman, “Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties”, Physical review letters, vol. 120, no. 14, pp. 145301, 2018.

[47] S.I. Ktena, S. Parisot, E. Ferrante, M. Rajchl, M. Lee, B. Glocker and D. Rueckert, “Distance metric learning using graph convolutional networks: Application to functional brain networks”, Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, pp. 469-477, 2017.

[48] M. Zitnik, M. Agrawal and J. Leskovec, “Modeling polypharmacy side effects with graph convolutional networks”, Bioinformatics vol. 34, no. 13, pp. i457-i466, 2018.

[49] S. Parisot, S.I. Ktena, E. Ferrante, M. Lee, R.G. Moreno, B. Glocker and D. Rueckert, “Spectral graph convolutions for population-based disease prediction”, Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, pp. 177-185, 2017.

[50] S. Parisot, S.I. Ktena, E. Ferrante, M. Lee, R. Guerrero, B. Glocker and D. Rueckert, “Disease prediction using graph convolutional networks: Application to autism spectrum disorder and Alzheimer’s disease”, Medical image analysis, vol. 48, pp. 117-130, 2018.

[51] R. Assouel, M. Ahmed, M.H. Segler, A. Saffari and Y. Bengio, “Defactor: Differentiable edge factorization-based probabilistic graph generation”, arXiv preprint arXiv:1811.09766, 2018.

[52] Y. Yuan, Z. Xiong and Q. Wang, "ACM: Adaptive cross-modal graph convolutional neural networks for rgb-d scene recognition", Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 9176-9184, 2019.

[53] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu and C. Chen, "Data-driven intelligent transportation systems: A survey", IEEE Transactions on Intelligent Transportation Systems, vol. 12, no. 4, pp. 1624-1639, 2011.

[54] M. Veres and M. Moussa, "Deep learning for intelligent transportation systems: a survey of emerging trends", IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 8, pp. 3152-3168, 2019.