ISSN: 0974-5823 Vol. 7 No. 5 May, 2022

International Journal of Mechanical Engineering

A Comprehensive Survey of Techniques, Applications, and Challenges in Deep Learning: A Revolution in Machine Learning

T. Aditya Sai Srinivas[1]

Department of computer science engineering

G. Pullaiah College of Engineering and Technology

G Mahalaxmi[2]

Department of electronics and communication engineering

G. Pullaiah College of Engineering and Technology

R Varaprasad[3]

Department of computer science engineering

G. Pullaiah College of Engineering and Technology

D Raziya[4]

Department of electrical and electronics engineering

G. Pullaiah College of Engineering and Technology

Abstract

Deep learning (DL) is a hot topic in machine learning (ML). To limit the amount of time and money spent on supervised machine learning, we use DL. With a variety of methodologies and topographies, DL may be applied to address complicated problems in a variety of contexts. Features that illustrate or differentiate are learned in a layered manner. When it comes to effective security solutions, DL has made significant strides in a wide number of application domains. The best alternative for revealing high-dimensional data's complex architecture is to use the back propagation technique in this manner. DL is benefiting business, science, and government in a variety of applications such as Artificial Intelligence (AI) and ML, which can be applied to everything from cancer detection to stock market research to smart cities. As a result, the focus of this work is on the basic ideas and limitations of DL.

Keywords: Deep learning (DL), Artificial Intelligence (AI), Machine learning (ML), Deep belief network(DBN), Artificial Neural Network(ANN), Intelligent systems

1. Introduction

ML and AI began to pay more attention to neural networks in the late 1980s a lot of effective learning algorithms and network structures were made[1]. Methods such as "Backpropagation"-trained multilayer perceptrons, self-organizing maps, and radial basis function networks [2], [3]and other new things to make them more interesting. Interest in neural networks has dwindled, despite their widespread use. When Hinton and colleagues [4] first proposed "Deep Learning" (DL) in 2006, they were using artificial neural networks (ANN). The term "newgeneration neural networks" was coined once DL became a trendy topic, rejuvenating neural network research. It's because well-trained deep networks have shown excellent results in a lot of different classification and regression tasks.

DL technology has become a prominent topic ML, AI, data science (DS), and analytics as it can learn from data. As a result, it is being actively researched by a wide range of companies like Google; Microsoft; Nokia; and others. As a part of ML and AI, DL is an AI feature that mimics the human brain's ability to process information. According to Google Trends statistics, "Deep learning" has been more popular around the world in recent years. DL technology leverages many levels of data abstraction to construct computational models. DL takes longer to train than other machine learning algorithms because of the large number of parameters [5].

Using the term "deep" to describe data transformations in DL relates to how many levels of data transformation are feasible. The depth of the credit assignment path (CAP) reflects the impulsive relationship between the input and output layers in these systems[6]. It's possible to distinguish between representational and deep forms of learning. It is combinations of techniques that help the computer develop representations for detection and classification. The term "deep learning" refers to a sort of learning that is more abstract and involves multiple levels of representation. Comparing ML and Deep neural networks (DNN) is shown in Figure 1.

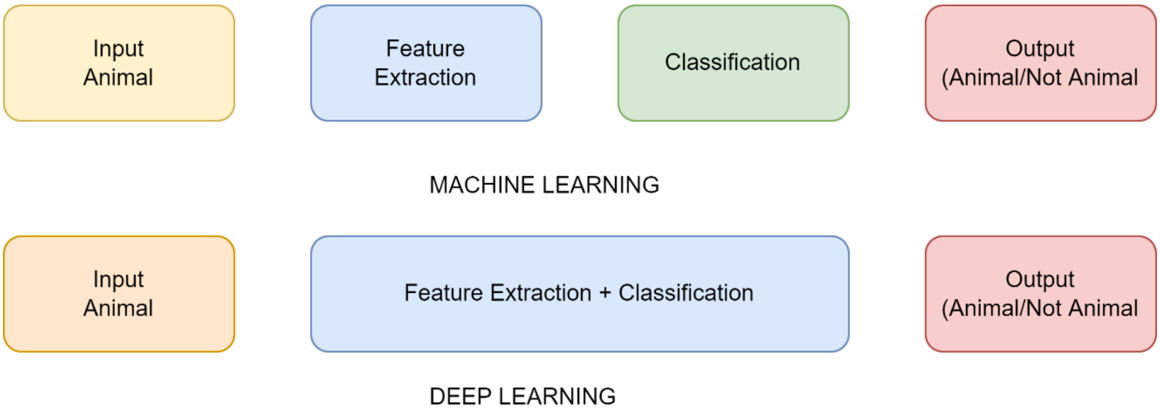

Fig. 1 Distinction between MLand DL

Deep learning employs nonlinear transformations and model abstractions when working with large databases. As an alternative, it defines how a machine accepts and modifies its internal attributes in response to abstractions and representations from previous layers.

When it comes to neural networks, DL is the best[7]. In order to construct multi-layered learning models, DL makes use of transformations and graph technologies. A number of recent DL techniques have exhibited excellent performance in areas such as audio and voice, visual data, and natural language processing (NLP) [8]–[10].

Input data representation is often a critical factor in the success of a machine learning system. A good data representation is better than a bad one. As a result, feature engineering has been a major topic in ML research for quite some time now. The raw data is used to build features. In addition, it necessitates a significant amount of manual labour and is highly context-specific. HOG [11], SIFT [12], and BoW[13] are all examples of computer vision techniques that use histograms of oriented gradients. The discovery of a new feature that works well can open up a whole new field of investigation for years to come.

Since DL generates an integrated model for feature extraction and classification, it can be applied in a broad variety of contexts. Using general-purpose algorithms, a large feature set, and no human interaction are all characteristics of DL. Deep Text was developed by Facebook to sort through massive volumes of data and remove spam. In order to implement a DL strategy, the following elements are essential:

• Multiple layers or stages of nonlinear processing.

• Supervised or unsupervised learning.

It is a mechanism in which the outcomes of the previous layer are passed on to the current layer via the preceding layer. According to the importance of the data, layers are ordered according to a hierarchy. There are connections between the class target label and supervised and unsupervised learning. A system that is unsupervised is one that does not have access to supervision. A presentation [14] outlined the current state of DL models, architectures, and restrictions. Among other things, these models were studied for their ability to learn, optimise, and fine-tune. For DL, large datasets were also emphasised.

Deep Learning's Position in AI

Modern lingo for intelligent software and systems refers to them variously as AI, ML, and DL. Fig. 2 shows how DL is compared to ML and AI. According to Fig. 2, DL is a subset of ML and AI. ML automates the creation of analytical models, AI, on the other hand, is the incorporation of human behaviour and intellect into machines [15]. DL also refers to data-driven learning techniques that employ multi-layer neural networks and processing. There are several levels or stages of data processing in DL, which is why the word "Deep" was coined.

As a result, DL is akey AI technology that may be utilised to create intelligent systems and automation. Increased artificial intelligence results in better AI. DL is strongly related to "Data Science" [16] since it can learn from datasets. For sophisticated analytics and intelligent decision-making, approaches based on DL can be used. On the whole, DL contributes to technology-driven automation, intelligent systems, and the achievement of Industry 4.0 goals.

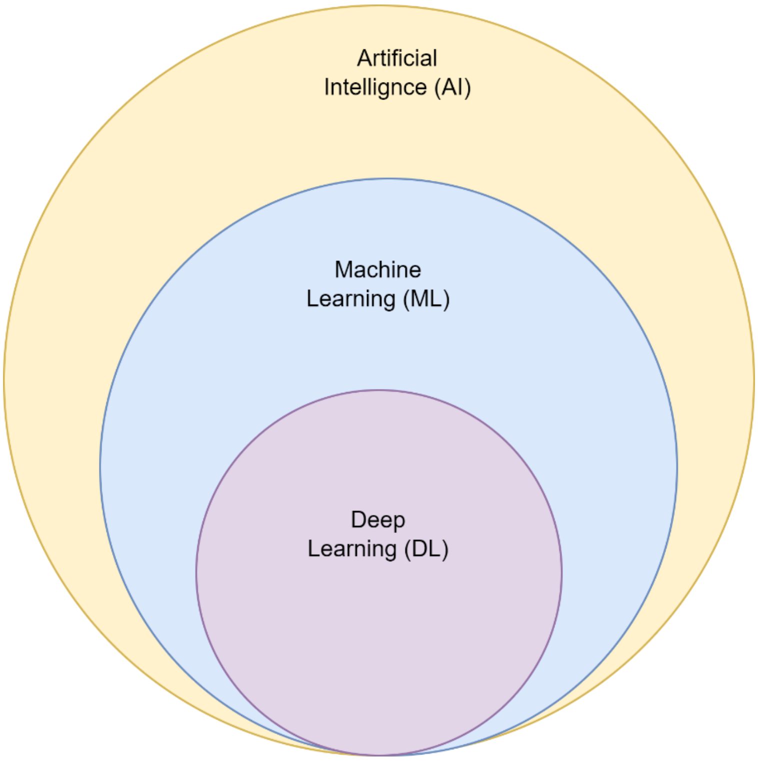

Fig.2 Position of DL in AI

What is the significance of DL in today's research and applications, and why is it important?

The Fourth Industrial Revolution is currently focusing on smart and intelligent systems, which include smart healthcare, business intelligence, smart cities, cyber security intelligence, and many more applications [16]. DL may now be used to uncover complex architectures in high-dimensional data for a wide range of applications, including security. Because of their ability to learn from prior data, DL models can be extremely useful in the development of intelligent data-driven systems. As a result, DL can change the world and the lives of its users through its automated and experience-based learning capabilities. Topics like these are common knowledge in computer science, especially in the context of today's intelligent computing. As a starting point for further research, let's take a look at how DL fits into AI.

2. Motivation to use Deep Learning

There has been a lengthy history of machine learning in computer science, and it has been utilised in a variety of scientific and engineering domains. By providing the machine with hundreds or thousands of training examples and correcting its mistakes, a human may help it learn and improve. Machine learning has grown in importance in AI, yet it is not without problems. Since humans are required to come up with the abstractions that allow computers to learn, this process takes a long time and does not accurately reflect how intelligent a machine really is.

Unfortunately, in many real-world scenarios, there isn't enough training data to build accurate and dependable models. Even if the best learning methods are used, a new field's models may not perform well even if they have a lot of good data to work with. Data that has been labelled is expensive and difficult to come by, but unlabelled data is readily available.

DL, in contrast to machine learning, is almost always unsupervised. It's all about building massive neural networks that allow computers to learn and perform calculations on their own, without the assistance of a human.

Learning to recognise cat faces using DL is possible by training a large unsupervised neural network on its own. Large-scale recommendation systems face data scarcity, which necessitates exploring new avenues for transferring information from other data sources.

3. Taxonomy of DL techniques

DL techniques can be unsupervised, semi-supervised, or supervised, depending on the situation. Deep reinforcement learning (DRL), often known as RL, is a partially supervised (and sometimes unsupervised) learning technique that is used in a variety of situations.

Supervised learning

This method focuses on labelled data. The technique's surroundings include inputs and outputs $(X_t, Y_t) \sim p$. If the input is x_t, the clever agent guesses $y\hat{}_t = f(x_t)$ and calculates the $\iota(y\hat{}_t, y_t)$ loss. The agent then adjusts the network settings to better predict the desired results. After a good training, they'll be able to ask questions of those around them. Among the numerous types of neural networks that can be used for supervised learning are RNNs, CNNs, and DNNs. RNNs also contain GRUs and LSTMs. The ability to gather or generate data based on previously collected or generated data is a big plus of this technology. The decision boundary may be stretched if the training set contains no instances from a class. Overall, this strategy is easier to learn than others.

Semi-supervised learning

This approach uses semi-labelled datasets to learn. Generative adversarial networks (GANs) and DRLs can likewise be used in the same way. RNNs, which comprise GRUs and LSTMs, are also used in partially-supervised learning. This approach has numerous advantages besides requiring less tagged data. However, this method is flawed due to irrelevant input features in the training data. It is often used to categorise documents based on their content. Due to the difficulty in accumulating a large number of labelled documents, semi-supervised learning is the best solution.

Unsupervised Learning

This approach can be used to integrate learning without labelled data (there is no need for labelling). Using this method, the agent can learn more about the input data's structure and relationships. Other unsupervised learning methods include generative networks and clustering. Newer DL methods like Restricted Boltzmann machines (RBM), auto-encoders, and GANs have shown promise in non-linear dimensionality reduction and grouping. RNNs, which comprise GRU and LSTM algorithms, are widely used in unsupervised learning applications. Unsupervised learning's main flaws are its lack of precision in data sorting and computational cost. Clustering is a popular unsupervised learning technique [17].

Reinforcement Learning

Both Reinforcement Learning (RL) and supervised learning (SL) are based on interaction with the environment. Google Deep Mind created this method in 2013 [55]. Many improved strategies based on reinforcement learning were developed as a result. As an illustration, if the input environment collects samples of the form: $x_t \sim p$, agents predict: $y\^{}_t = f(x_t)$ and agents get costs of: $c_t \sim P(c_t| x_t, y\^{}_t)$ where P denotes the unknown probability distribution, the environment is asking a question of the agents. It returns a jumbled number as a result. Semi-supervised learning is another name for this approach. Many supervised and unsupervised methods were created based on this idea. Reinforcement learning is a significantly more complex method of learning than typical supervised techniques, because there is no simple loss function accessible. Another key distinction between supervised learning and reinforcement learning is that in the former no full access to the function is available; this means it must be accessed through interaction; in the latter, the state being dealt with is based on environment, where the input x_t is determined on previous actions [18].

The sort of reinforcement learning required for a task is determined by the problem's area or scope. For problems with multiple parameters, DRL is the optimum method. However, derivative-free reinforcement learning works effectively for situations with few parameters. **Basic Architectures of Deep Neural Network (DNN)**

DL architectures include Deep belief networks(DBN)[19], RNN. DNNs are created by adding hidden layers between the input and output layers of various ANN topologies. For example, deep neural networks can describe complex non-linear interactions and generate layered primitive object models. These networks feature no loops and data travels from the input layer to the destination. DL can be implemented in a variety of ways. Table 1 shows the year-to-year dispersion of DL architecture.

Table 1: Years of DL architecture use

Year	**The design of DL**
1990–1995	RNN
1995–2000	Long short-term memory, CNN
2000–2005	Long short-term memory, CNN
2005–2010	DBN
2010–2017	Deep stacked network, gated recurrent unit

DL architectures can be broken down into six main categories:

3.1 Auto-Encoder (AE)

Unsupervised learning is performed by an auto-encoder (AE). Initially, the network sets the output values to match the input values. The network is trying to grasp the identity function. It has three layers: input, encoding, and decoding. To recreate the network's input, the hidden layer must learn the best representations. The input is represented by a code layer. PCA(Principal Component Analysis) is comparable to auto-encoders, which are neural networks[20].

Auto-encoders are comparable to PCAs, but their flexibility is greater. While PCA only allows for linear transformations, auto-encoders allow for both linear and non-linear representations. In a DL network, auto-encoders can be stacked and layered.

Auto-encoders can be classified as the following:

1. **De-noising Auto-encoder:** It's a more improved auto-encoder. These encoders address the identity functions by corrupting and rebuilding the input. This method is sometimes called stochastic auto-encoding[20][21].
2. **Sparse Auto-encoder:** These auto-encoders learn to extract features from unlabelled data. input is received, and not too often [22].

3. **Variational Auto-Encoder (VA**E): The variational autoencoder has three components: encoder, decoder, and loss function. Massive datasets can be modelled with numerous variables using them. It is also called as HD network[23].
4. **Contractive Auto-encoder (CAE):** Despite the fact that these are resilient networks, they differ from other auto-encoders in that contractive auto-encoders use encoder function to generate robustness, whereas de-noising auto-encoders use reconstruction process to create robustness[21].

A collection of data is represented using dimensionality reduction using auto-encoders to work with high-dimensional data. The Auto-encoder makes use of two basic structures: Denoising Auto-encoder (DAE) and Sparse Auto-encoder (SAS) (AE). Sparse auto-encoding uses hidden unit activation states to experience network weight, while De-noising auto-encoding uses noise data. An auto-encoder assesses the input before transferring it to an intrinsic transformation.

4.2 Convolutional Neural Network (CNN)

A CNN is an animal visual brain multi-layered neural network. LeCun et al. established the first CNN. In addition to images, CNN can identify handwritten characters such as postal codes. As indicated in the architectural diagram above, early layers recognise edges, whereas subsequent layers recombinate features to generate high-level input attributes and then classify it. Pooling reduces the dimensionality of the collected features[24].Convolution and pooling are the following steps, followed by a multilayer perception with properly coupled layers. To identify picture features in the final "output layer," back propagation algorithms are used. The system's accuracy and performance are greatly boosted by CNN's unique characteristics including local connectivity and shared weights. All other DL strategies fail. It is the most extensively used architecture. Figure 3 shows the data flow from inputs, convolutional, pooling, hidden, and output layers of a CNN.

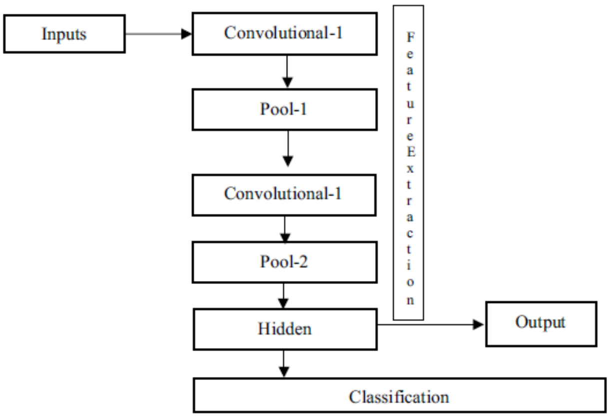

Fig. 3 CNN Architecture

4.3 Restricted Boltzmann Machines and Deep Belief Network

A Restricted Boltzmann machines (RBM)[25][19] depicts the hidden layer, the visible layer, and their symmetric relationship in undirected graphical fashion. Inputs cannot communicate with the hidden layer while utilising RBM. To construct multilayer network architecture for the deep belief network, a novel training approach with hidden layers was applied. RBMs are also known by other names. The input layer represents fundamental sensory data, whereas the hidden layer represents an abstract description of it. The output layer is solely responsible for network categorization[26].Pre-training and fine-tuning are unsupervised stages of the training process. Unsupervised training of RBM's initial hidden layer allows it to recreate its inputs. Like the original RBM, this one needs the first hidden layer's inputs and outputs to operate. That's why every layer is prepared and trained. Pre-training is now complete, and fine-tuning can begin. After all output nodes have been labelled with the values or labels they represent, the network is trained using gradient descent learning or back-propagation.

4.4 Deep Stacking Networks

DSNs are also called as deep convex networks. DSN is unlike other DL frameworks. It is named deep because it has many deep individual networks, each with its own set of hidden layers. The DSN views training as a series of unique challenges. The DSN is composed of modules that are integral to the network's architecture and network. The DSN is composed of three modules. Each module in the model is comprised of three components: an input, a hidden zone, and an output. Outputs of the layer that came before it as well as the valid input vector. DSN trains each module independently to ensure that it is productive, competent, and capable of cooperating. Supervised training use back propagation for each module, rather than the entire network. Due to the fact that DSNs outperform DBNs, they are the recommended network architecture.

4.5 LSTM/GRU Network

Hochreiter and Schimdhuber worked together to create the long Short-Term Memory (LSTM), which is now found in a wide range of technologies. IBM chose LSTMs primarily for use in speech recognition. Cells are the basic building blocks of LSTMs, and they're used to store information that can be used as an input to the model throughout time. This aids the device in remembering the most recently computed value. The flow of information into and out of a memory unit or cell is controlled by three ports known as gates, which are located on the unit's surface.

• New information is loaded into the memory by way of the input port, which is also known as a "gate."

• It is used to aid the cell in memorising new data when an existing piece of information is forgotten. The second gate is referred to as the forgets port controls gate.

• The output gate's function is to regulate the information contained in the cell and used as the cell's output.

One way to manage the cell is to use its weight. The training strategy known as Back propagation through time (BPTT) is needed in order to increase weight gain. The approach relies on network output errors to optimise the algorithm. The update and the reset are the two gates that make up the Gated Recurrent Unit (GRU). An update gate's job is to notify the prior cell's contents that they need to be replaced. Using a reset gate, the previous cell's contents can be transferred to the current input. Normal RNNs can be simulated with the GRU by setting their reset and update gates to 1. To put it another way: The GRU model is easier to use than the LSTM model. In comparison to other methods, it may be taught fast and is viewed as more efficient.

4.6 Recurrent Neural Network

RNN is the fundamental network architecture since it includes a variety of network structures. Unlike complete feed-forward connections, recurrent networks have a connection that can be utilised to feed back into preceding levels. For real-time simulation, it employs past recollection of the input. Back-propagation is a strategy for improving, specialising, and expanding networks over time (BPTT). Table 2 summarises the numerous deep neural network architecture's application domains.

Table 2: DNN architectures and applications

Architecture	**Major application areas**
Auto-encoder	Natural language processing, understanding compact representation of data
Convolutional neural networks	Document analysis, face recognition, image recognition, natural language processing, video analysis
Deep belief networks	Failure prediction, information retrieval, image recognition, natural language understanding
Deep stacking networks	Continuous speech recognition, Information retrieval
LSTM/GRU networks	Gesture recognition, handwriting recognition, image captioning, natural language text compression, speech recognition
Recurrent neural networks	Handwriting and speech recognition
Restricted Boltzmann machine	Collaborative filtering, classification, dimensionality reduction, feature learning, regression, and topic modeling

5. Advanced Architectures of Deep Neural Network

Deep neural networks can be modelled in a variety of ways due to the network's many capabilities. These models are termed deep models, and they include the following:

AlexNet: The net is named after the scientists. Created by Alex Krizhevsky, Geoffrey Hinton, and their team of DL researchers. The convolutional and pooling layers are layered on top of each other, with layers totally interwoven. The system's superiority and scalability are based on GPU use. AlexNet uses a GPU to process and train quickly.

Visual Graphic Group Net: Network for Visual Graphics Experts from Oxford's Visual Graphics Group constructed this pyramid-shaped net. The model's bottom layers are wide and its top layers deep. Smaller layers are made using VGG convolutional and pooling layers.

GoogleNet: GoogleNet The Net was formed when Google researchers named the architecture. Unlike VGG's 19 layers, this one has 22. The inception module is a new Google Net technique. A single layer has many feature extractors, all of which help the network operate better. Several of these modules are layered to make the final conceptualization module. The model converges faster due to joint and parallel training. Pre-trained GoogleNet is smaller than VGG, allowing for faster training.

ResNet: The Residual Network is the basic building block of ResNet. A full node to node network is formed by stacking the remaining modules. The ability to train ResNet with a high number of residual layers is its main benefit.

ResNeXt: ResNeXt Based on ResNet's principles, it features an updated and novel architecture for better performance.

RCNN (Regions with Convolutional Neural Network): It relies on creating a bounding box around the image's objects and identifying the object in the image.

YoLo(You only look once): This architecture delivers a solution to the challenge of image recognition. To determine the object's class, the image is segmented into bounding boxes, and a single recognition method is run on each of those sections. After classifying the items, the boxes are carefully blended to form a tight possible bounding box around them. To solve day-to-day issues, it is employed in real time.

SqueezeNet: The SqueezeNet architecture is the best choice for a low bandwidth environment. An inception procedure for this network architecture will need approximately 100 MB of storage space. When things take a turn for the worst, a fire module comes to the rescue.

SegNet: SegNet is widely recognised as the most effective image segmentation model. SegNet is a DNN that segments images. This is done by arranging processing layers of encoders and decoders. Keeping fine details in the split image is SegNet's main strength. Pooling indices link encoder and decoder networks. In addition, the flow of information is straightforward.

GAN: Generative Adversarial Networks (GAN) are an innovative network architecture that generates wholly new and diverse pictures, as well as those that are not included in the training dataset

Characteristics of Deep Learning

DL has a wide range of applications due to the following features: These include decision fusion, mobile devices, transfer learning, class imbalance, and human activity detection.

Thus, the following are the Characteristics of DL:

1. Extremely useful in a variety of fields.
2. With the inclusion of more than two layers, it is referred to as "deep" because it is solely dependent on neural networks.
3. Be able to pick things up quickly and easily.
4. Has the ability to better utilise datasets Take advantage of the data to learn how to extract features.
5. Outperform human performance on computationally intensive tasks.
6. DL involves very little manual engineering.
7. Optimized results.
8. DL networks are structure-dependent, activation function-dependent, and data representation-dependent.
9. Describe highly variable characteristics with only a few factors.
10. Predictive accuracy can be considerably improved.
11. Solve computationally intensive problems.
12. High-dimensional sensory inputs can be analysed to derive features.
13. Secure and robust generalisation capacity that requires less training data.
14. Voice activity detection can be improved by combining the advantages of different characteristics.
15. In terms of feature representation, this approach outperforms the ML model.
16. Improvements can be made to the estimation of covariance for prediction applications.
17. DL networks are not pre-trained and do not require prior data or knowledge.
18. When dealing with vast amounts of unlabelled data, DNN's representation is particularly useful because of its new approach to deciphering the representations.
19. These networks are capable of extracting intricate features due to their high-level abstraction.
20. It's becoming easier to recognise patterns in large datasets.

6. Recognizing and Using a Variety of Data Sources

An in-depth understanding of and representation of data is required in order to build an intelligent system that is data driven in a certain application area. DL models can be represented in a variety of ways based on the data that exists in the actual world.[27].Sequential Data: It's a series of steps. The model must take into consideration the sequential manner of the incoming data. Video, audio, and time series are all examples of sequential information. Tabular Data: The rows and columns of a tabular dataset are clearly defined. A tabular dataset is a collection of data organised into rows and columns in a database table. Data of the specified type must be included, and it must be named. If you're looking for an efficient method of grouping data into rows and columns, this is it. DL algorithms may be used to create intuitive data-driven systems.

The properties and dependencies of the DL

Unlike ML modelling, DL modelling automates the extraction of features. ML methods such as support vector machines (SVM), decision trees, random forests, naive Bayes, and k-means clustering are widely used in a variety of industries [97]. DL models, on the other hand, use convolution neural networks, which must be taken into account when developing DL models for real-world applications. Black-box Perception and Interpretability :Black-box Accurately interpreted and perceived. Interpretability is crucial when evaluating DL vs ML. A "black-box" approach is used to describe findings from DL. In contrast, rule-based ML systems [28] provide clear logic rules (IF-THEN) for making judgments that can be understood by humans. The extracted rules are easier to comprehend, alter or remove depending on the target applications, as we have described in previous work [29].

Data Dependencies: Data dependent DL requires a lot of data to construct a data-driven model for a certain problem domain. DL algorithms often perform badly when data volumes are small [24]. Using the provided criteria will increase the performance of the standard machine-learning algorithms [30].

Hardware Dependencies: A computer's hardware is required to function. With large datasets, DL approaches require massive calculations to train a model. GPUs are used to speed up processes since their advantage over a CPU grows as the number of calculations increases. As a result, DL training requires GPU hardware. DL, on the other hand, necessitates more powerful processors with GPUs than standard ML.

Model Training and Execution time: The model training procedure takes longer using the DL approach because of the large number of parameters. As an example, training DL models can take weeks, whereas training ML algorithms can be done in seconds to hours. To put it another way, DL algorithms run far faster than traditional ML methods.

Feature Engineering Process: Extract features (characteristics, traits, and attributes) from raw data with the use of domain expertise. DL is distinct from other ML techniques since it aims to extract high-level attributes directly from data [31]. As a result, each feature extractor may be built faster and with less effort thanks to DL. With DL, data growth can be handled more efficiently than with standard ML. Comparing the performance of data-driven and traditional ML methods. Parallelized matrix and tensor operations, as well as gradient computation and optimization, are used to generate and train DL models[32] in this algorithm. With these and many other features required for implementation and the building of DL models, PYTORCH [33] (with a high-level API called Lightning) and Tensor Flow [34] (also offering Keras) are the two most popular DL frameworks available today.

7. Applications of Deep Learning

Speech, NLP, and computer vision are all examples of this. Using hierarchical inputs, these are non-linear classification issues It wasn't until 2011 that the Google Brain team created a deep neural network capable of understanding high-level concepts like "cat" only by watching YouTube videos. DL is being used by Facebook to improve the recognition of faces and other objects in photographs and videos that are posted on a daily basis.

So, here is a list of DL Applications:

1. Self-Driving Cars

Autonomous driving relies heavily on DL. A model is built, the computers are trained, and the conclusions are tested in a safe environment using millions of data points. Aside from developing self-driving cars, the Uber Artificial Intelligence Labs in Pittsburgh are also working to integrate smart functions such as food delivery. The primary goal of innovators is to solve problems in unconventional ways. Continuous testing and implementation means that drivers are exposed to an ever-increasing number of scenarios, ensuring the safety of everyone on the road. Sophisticated traffic models are being built using sensor and camera data, which are helping to identify routes, signage for pedestrians, and real-time traffic flow and obstacles. As a matter of fact, 3-D mapping is now only used in high-traffic areas and has not proven to be as effective in averting accidents as conventional mapping. Because it is impossible to achieve the same level of precision and reliability with no maps, Teddy Ort, a graduate student at CSAIL, claims that this strategy has never been tried before. Using this method, self-driving cars can navigate roadways that have not yet been mapped by tech companies.

2. Natural Language Processing (NLP)

One of the most challenging obstacles people confront understands the complexities of language, such as grammar, semantics, and nuances in tonality, emotions, and even sarcasm. It is only through continual exposure and training as an infant that humans learn how to respond to every situation in a unique way. Using DL techniques, Natural Language Processing (NLP) may teach robots how to correctly respond to a variety of questions. Legal document summaries are widely utilised and tested, making paralegals obsolete. The application of DL in natural language processing is increasing. Convolutional and Recursive neural networks (RNN), Reinforcement learning(RL) techniques, and memory augmentation techniques are all aiding in the development of natural language processing techniques (NLPs). Linear semantic links can be established using a combination of word embedding and distributed representations.

3. Visual Recognition

Assume you're looking through an album of old photos. After sorting them, you decide to put a few of them in frames. This had to be done manually due to the lack of metadata. The only way to arrange them was by date, and even then, some downloaded images were missing their associated metadata. Photos may now be sorted by location, faces, a group of people, events, dates, and other factors thanks to DL. Visual recognition systems with many levels from basic to complex are needed to find a specific photo in a collection (such as Google's picture library). A large-sized picture in this area of digital media management, Convolutional neural networks (CNN), Tensorflow, and Python are employed extensively to improve visual recognition.

4. Healthcare

According to NVIDIA, GPU computing is the driving force behind the healthcare industry's transformation. Physicians, researchers, and clinicians are all benefiting from GPU-accelerated applications and systems. Medical DL projects include detecting life-threatening diseases early, accurately, and quickly; augmenting clinicians to deal with a shortage of doctors and healthcare providers; standardisation of pathology results and treatment protocols; and understanding genetics to predict future disease risk and bad health episodes In the healthcare industry, readmissions cost tens of millions of dollars each year. It is possible to reduce readmissions while cutting healthcare costs through the use of neural networks and DL techniques.

Automatic Machine Translation: CNNs are capable of recognising photographs that have visible text. A picture can be used to re-create them after they've been translated and converted into text. Instant translation via visuals Words, phrases, and sentences can be translated automatically with this programme. Machine translation using DL beats previous methods in two distinct ways: 1)Automated translation of written material 2)Software for Transforming Pictures. There is often no sequence planning for translations. Algorithms need to learn the relationships between words in order to learn a new language.

Image – Language Translations: Image - Language translations are an intriguing DL application. With the Google Translate app, you can now translate photographic images with text into any language you wish. Simply position the camera on top of the device, and your phone will interpret the image, OCR it (convert it to text), and convert it to text, then translate it into the selected language.

Pixel Restoration: When it came to video zooming before DL, the idea was a non-starter. An algorithm developed by Google Brain in 2017 can identify faces in low-resolution images. This method was utilised in the Pixel Recursive Super Resolution. Photos are sharpened and significant characteristics are highlighted, making it easier to recognise people's personalities.

5. Demographic and Election Predictions

In order to test the power of DL, Gebru et al. used 50 million Google Street View pictures. The results were excellent as always. Cars were identified by the computer when it learned how to identify them. Cars were categorised by make, model, body type, and year of manufacture. The success of this DL capabilities prompted more research.

6. Deep Dreaming

The use of DL Networks to improve computer images was first identified by Google researchers in 2015. Deep Dreaming is an important part of one of today's DL applications. Using this technique, the computer may create a new dream by hallucinating over an old shot. The type of hallucination you experience is dependent on your particular brain network and how much of it is exposed to the hallucinogen.Researchers at Sussex University built a virtual reality hallucination machine using deep dreaming.

7. Colorization of Black and White Images

Using grayscale photos as an input, the process of image colorization creates colourized versions of the originals (as output). Traditionally, this was done by hand due to the difficulty of the task. DL Technology currently uses context and objects in the shot to colour the image in a manner similar to a human operator.

8. Challenges

DL approaches have shown to be the most effective because of their many layers and high level of abstraction. We can now rely on artificial intelligence to do nearly all of the same functions as humans. In order to feel the exhilaration of triumph, today's technology must overcome numerous challenges. As a result, DL faces the following difficulties:

•DL algorithms have to keep track of the input data all the time.

• Algorithms must make sure that the conclusion is clear.

• Technology that needs a lot of resources, like high-performance GPUs and storage space.

• New ways to analyse big data. They are called "black boxes."

• Hyper parameters and a complicated design are present.

In order to do a lot of math, you need a lot of computing power. As a result, you may have problems in your area.

• Impossible to solve on a computer.

• You need a lot of data to do this job.

Because of the complicated problems and calculations, it costs a lot.

• There isn't a strong theoretical base.

• It's hard to figure out the topology and training parameters for DL. DL gives computers new data-processing tools and infrastructures for processing data and lets them learn about things and how they look.

9. Conclusion

DL is a fast-growing area of ML, and it's only going to get better. Many areas are now using DL algorithms, proving their effectiveness and adaptability. With DL, we can plainly see its importance, progress, and possibility for further research. It's also important to think about the learning hierarchy and monitoring while designing a successful DL programme. The supervisor prioritises database maintenance, which necessitates a hierarchy for effective data classification.

DL is based on the optimization of existing ML applications and the hierarchical layer processing of DL. Image processing and speech recognition are two examples of how DL can be put to use. Using DL in conjunction with face and speech recognition can be a useful security tool both today and in the future. There are numerous uses for digital image processing as a study field. Artificial intelligence researchers are focusing on DL since it appears to be a true optimization. According to the success wave, DL is gaining adoption in many applications as more data and processing power are available. When it comes to applications like natural language processing, remote sensing and healthcare, it is expected that DL will continue to advance rapidly in the future years.

References

[1] J. Karhunen, T. Raiko, and K. Cho, “Unsupervised deep learning: A short review,” *Adv. Indep. Compon. Anal. Learn. Mach.*, pp. 125–142, 2015.

[2] K.-L. Du and M. N. S. Swamy, *Neural networks and statistical learning*. Springer Science \& Business Media, 2013.

[3] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011.

[4] G. E. Hinton, “Deep belief networks,” *Scholarpedia*, vol. 4, no. 5, p. 5947, 2009.

[5] Y. Xin *et al.*, “Machine learning and deep learning methods for cybersecurity,” *Ieee access*, vol. 6, pp. 35365–35381, 2018.

[6] L. Alzubaidi *et al.*, “Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions,” *J. big Data*, vol. 8, no. 1, pp. 1–74, 2021.

[7] S. Dargan, M. Kumar, M. R. Ayyagari, and G. Kumar, “A survey of deep learning and its applications: a new paradigm to machine learning,” *Arch. Comput. Methods Eng.*, vol. 27, no. 4, pp. 1071–1092, 2020.

[8] A. Adeel, M. Gogate, and A. Hussain, “Contextual deep learning-based audio-visual switching for speech enhancement in real-world environments,” *Inf. Fusion*, vol. 59, pp. 163–170, 2020.

[9] H. Tian, S.-C. Chen, and M.-L. Shyu, “Evolutionary programming based deep learning feature selection and network construction for visual data classification,” *Inf. Syst. Front.*, vol. 22, no. 5, pp. 1053–1066, 2020.

[10] T. Young, D. Hazarika, S. Poria, and E. Cambria, “Recent trends in deep learning based natural language processing,” *ieee Comput. Intell. Mag.*, vol. 13, no. 3, pp. 55–75, 2018.

[11] A. Zheng and A. Casari, *Feature engineering for machine learning: principles and techniques for data scientists*. “ O’Reilly Media, Inc.,” 2018.

[12] D. G. Lowe, “Object recognition from local scale-invariant features,” in *Proceedings of the seventh IEEE international conference on computer vision*, 1999, vol. 2, pp. 1150–1157.

[13] L. Wu, S. C. H. Hoi, and N. Yu, “Semantics-preserving bag-of-words models and applications,” *IEEE Trans. Image Process.*, vol. 19, no. 7, pp. 1908–1920, 2010.

[14] S. Paul, L. Singh, and others, “A review on advances in deep learning,” in *2015 IEEE Workshop on Computational Intelligence: Theories, Applications and Future Directions (WCI)*, 2015, pp. 1–6.

[15] I. H. Sarker, M. H. Furhad, and R. Nowrozy, “Ai-driven cybersecurity: an overview, security intelligence modeling and research directions,” *SN Comput. Sci.*, vol. 2, no. 3, pp. 1–18, 2021.

[16] I. H. Sarker, “Data Science and Analytics: An Overview from Data-Driven Smart Computing, Decision-Making and Applications Perspective,” 2021.

[17] M. M. Saeed, Z. Al Aghbari, and M. Alsharidah, “Big data clustering techniques based on Spark: a literature review,” *PeerJ Comput. Sci.*, vol. 6, p. e321, 2020.

[18] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, “Deep reinforcement learning: A brief survey,” *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 26–38, 2017.

[19] T. Aditya Sai Srinivas and S. S. Manivannan, “Prevention of Hello Flood Attack in IoT using combination of Deep Learning with Improved Rider Optimization Algorithm,” *Comput. Commun.*, vol. 163, pp. 162–175, 2020.

[20] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[21] 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.,” *J. Mach. Learn. Res.*, vol. 11, no. 12, 2010.

[22] A. Makhzani and B. Frey, “K-sparse autoencoders,” *arXiv Prepr. arXiv1312.5663*, 2013.

[23] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *arXiv Prepr. arXiv1312.6114*, 2013.

[24] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[25] B. Marlin, K. Swersky, B. Chen, and N. Freitas, “Inductive principles for restricted Boltzmann machine learning,” in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 509–516.

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

[27] I. H. Sarker, “Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions,” *SN Comput. Sci.*, vol. 2, no. 6, pp. 1–20, 2021.

[28] I. H. Sarker, “Machine learning: Algorithms, real-world applications and research directions,” *SN Comput. Sci.*, vol. 2, no. 3, pp. 1–21, 2021.

[29] I. H. Sarker, A. Colman, and J. Han, “Recencyminer: mining recency-based personalized behavior from contextual smartphone data,” *J. Big Data*, vol. 6, no. 1, pp. 1–21, 2019.

[30] I. H. Sarker, A. S. M. Kayes, and P. Watters, “Effectiveness analysis of machine learning classification models for predicting personalized context-aware smartphone usage,” *J. Big Data*, vol. 6, no. 1, pp. 1–28, 2019.

[31] L. Deng and D. Yu, “Deep learning: methods and applications,” *Found. trends signal Process.*, vol. 7, no. 3--4, pp. 197–387, 2014.

[32] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O’Reilly Media, 2019.

[33] A. Paszke *et al.*, “Pytorch: An imperative style, high-performance deep learning library,” *Adv. Neural Inf. Process. Syst.*, vol. 32, pp. 8026–8037, 2019.

[34] M. Abadi *et al.*, “Tensorflow: A system for large-scale machine learning,” in *12th $\{$USENIX$\}$ symposium on operating systems design and implementation ($\{$OSDI$\}$ 16)*, 2016, pp. 265–283.

[35] Yogesh Hole et al 2019 J. Phys.: Conf. Ser. 1362 012121