Notes

Module - V: Deep Learning

Learning Objectives

At the end of this module, you will be able to:

- Define the concepts of deep learning
- Analyse the neural network and its utility in modelling and solving problems
- Explain biological motivations and parallelism
- Explain recurrent neural networks
- Analyse the multilayer perceptron

Introduction

Deep learning is a specific field within the realm of machine learning that focuses on the application of neural networks with multiple layers, generally known as deep neural networks, to effectively model and solve complex problems. Neuromorphic computing is a specialised subfield within the realm of artificial intelligence that aims to emulate the structural and functional characteristics of neural networks observed in the human brain.

The discipline of deep learning has attracted significant attention and recognition due to its ability to independently gather and represent features from raw data. The aforementioned feature has greatly contributed to its outstanding performance across many domains, encompassing, but not restricted to, image and audio recognition, as well as the processing of natural language.

5.1 Concepts of Deep Learning

Deep learning is a specific field within the realm of machine learning that focuses on the application of neural networks with multiple layers, generally known as deep neural networks, to effectively model and solve complex problems. Neuromorphic computing is a specialised subfield within the realm of artificial intelligence that aims to emulate the structural and functional characteristics of neural networks observed in the human brain. The discipline of deep learning has attracted significant attention and recognition due to its ability to independently gather and represent features from raw data. The aforementioned feature has greatly contributed to its outstanding performance across many domains, encompassing, but not restricted to, image and audio recognition, as well as the processing of natural language.

5.1.1 Introduction to Deep Learning: Part I

In a startling fashion, Artificial Intelligence (AI), in the numerous areas that integrate it, has begun to faithfully imitate human behavior and cognition at the beginning of the twenty-first century. As a result, notable advancements have been made in a variety of fields, including computer-assisted medical diagnosis, DNA sequence classification, data mining, artificial vision, voice recognition, written language analysis, virtual games, robotics and any others where reasoning is a key component. Deep learning is a relatively new field of AI that was developed a few decades ago with the primary goal of enabling intelligent agents to make their own decisions, a feat that is currently only imagined in science fiction. In this way, multiple Artificial Neural Networks (ANN) methodologies are utilized in deep learning with the aim of providing an agent with a personality similar to a human. Deep Neural Networks, Convolutional Neural Networks and Deep Belief Networks are a few of these methods.

Notes

A discipline known as "Deep Learning" uses numerous hidden layers of an ANN to achieve deep abstractions in order to find patterns. The border of the region of interest in a picture is selected below figure as an example of how to achieve abstraction in a hidden layer. Depth is achieved by repeating abstraction in as many hidden layers as needed. Since there is no metric that determines how many to use, both the number of hidden layers and that of neurons range from 1 to n. Instead, this attempts to resolve the agreement to the problem, the dimensions and properties of the data set and based on the experience of who implements the ANN. Our ANN will produce the desired result, which corresponds to a classification that informs us that a "giraffe" would be represented in an input image like the one in the figure below. The representation of the input and output images for deep learning is also shown in the same graphic.

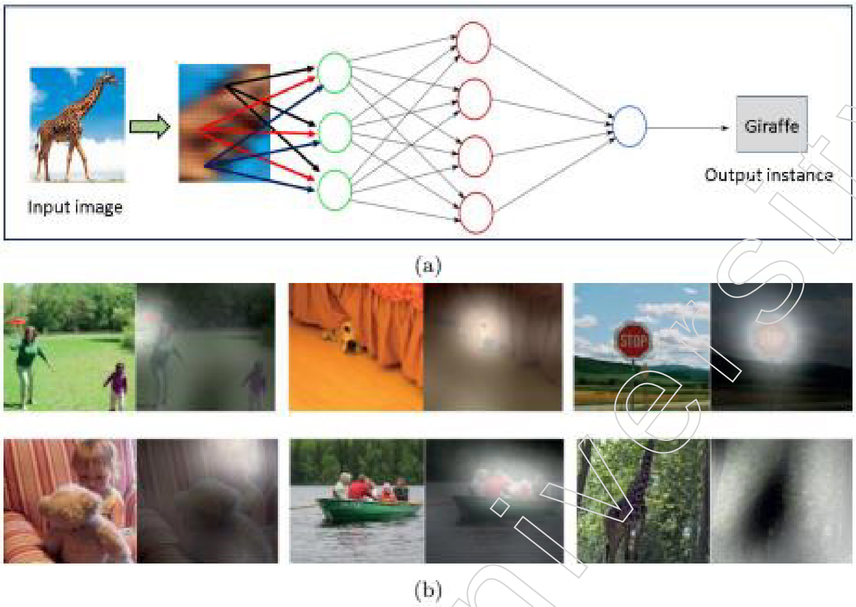


Figure: a) Abstractions using the Deep learning approach, b) Input and output images before and after processing with Deep learning.

A deep model is simply a complicated tensorial computation that may be ultimately broken down into common mathematical operations from analysis and linear algebra. Over time, the field has produced a sizable number of high-level modules with a distinct semantic and intricate models that combine these modules and have been successful in a variety of application domains. Deeper structures, or lengthy compositions of mappings, result in better performance, according to empirical data and theoretical findings.

Machine thought has long been a goal of inventors. At least since the days of ancient Greece, this yearning has existed. The entities Galatea, Talos and Pandora can be interpreted as examples of artificial life, while the mythological figures Pygmalion, Daedalus and Hephaestus can be regarded as legendary pioneers in the realm of innovation.

More than one hundred years prior to the construction of the first programmable computer, individuals pondered the possibility of machines attaining intelligence. The field of artificial intelligence (AI) is now experiencing significant growth, characterised by a wide range of practical applications and continuous research endeavours. Intelligent software is employed for the purpose of automating monotonous operations, comprehending voice or visuals, facilitating medical diagnosis and providing foundational support for scientific study. A compilation of rigorous mathematical concepts can be employed to elucidate scenarios that pose intellectual challenges for humans but are

Notes

comparatively straightforward for computers. The early stages of artificial intelligence witnessed prompt and effective resolution of such difficulties. The primary obstacle faced by artificial intelligence pertains to addressing problems that are readily and instinctively resolved by humans, such as recognising spoken words or identifying faces in images. However, these tasks are challenging for people to articulate using formalised language or explanations.

One potential solution to this challenge involves enabling computers to acquire knowledge through experiential learning and develop a comprehensive understanding of the universe by organising information in a hierarchical structure, wherein each concept is defined in relation to more fundamental notions. This approach mitigates the necessity for human operators to explicitly specify each fragment of knowledge that the computer requires, as it acquires knowledge through experiential learning. The hierarchical organisation of concepts facilitates the acquisition of sophisticated ideas by computers through the construction of complex ideas from simpler ones.

Paradoxically, tasks that are characterised by formality and abstraction, which pose significant cognitive challenges for individuals, are very straightforward for computers to accomplish. Computers have consistently outperformed even the most skilled human chess players, but only in recent times have they started to approach the level of proficiency exhibited by average individuals in tasks such as speech recognition and object detection.

In order to navigate their daily existence, individuals must possess a substantial breadth of knowledge about the world. Much of the acquired knowledge in this context exhibits characteristics of irrationality and intuition, rendering its formalisation a formidable task. For computers to exhibit intelligent behaviour, it is imperative that they possess the capability to acquire equivalent knowledge. One of the primary challenges in the field of artificial intelligence pertains to the incorporation of unstructured knowledge into machine systems.

The efficacy of these elementary machine learning algorithms is heavily influenced by the manner in which the data is represented. For example, the AI system does not conduct a physical examination of the patient when making a recommendation for a caesarean birth through the utilisation of logistic regression. In contrast, the medical practitioner furnishes the system with a range of relevant particulars, including the presence or absence of a scar on the uterus. A feature refers to any data element that is observed within the patient representation. Logistic regression can be employed to identify the associations between various patient features and distinct outcomes. Nevertheless, it is important to note that it does not have any influence on the definition of the features. In contrast to the standardised reports generated by medical professionals, it is unlikely that logistic regression can generate meaningful predictions based just on an MRI scan of the patient. The connection between specific MRI scan pixels and potential delivery difficulties is quite low.

This reliance on representations is a widespread phenomenon that may be seen in both everyday life and computer science. If the data collection is appropriately structured and indexed, computer science procedures like querying a collection of data can move forward considerably faster. Arithmetic with Roman numerals takes far longer for people to complete than with Arabic numerals. It is not surprising that the representation used has a significant impact on how well machine learning algorithms function. Take a look at Figure below for a clear visual example. By creating the ideal set of features to extract for a task and then feeding those features to a straightforward machine learning algorithm, many artificial intelligence tasks can be addressed. A measurement of the speaker's

Notes

vocal tract size, for instance, can be a useful attribute for speaker identification from sound. As a result, it strongly suggests that the speaker is either a man, woman, or little child. It might be challenging to determine which features should be retrieved for certain activities. Let's say we wanted to create a program that could identify cars in pictures. We would prefer to use the presence of a wheel as a feature since we are aware that autos have wheels. Unfortunately, it is challenging to precisely characterize a wheel's appearance in terms of pixel values. A wheel has a straightforward geometric shape, but its appearance can be complex by shadows falling on it, sunlight reflecting off its metal components, the car's fender or something in the foreground blocking part of the wheel and so on.

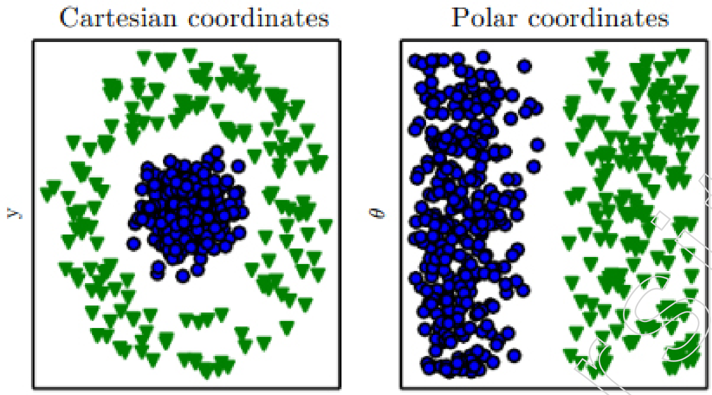


Figure: Example of different representations

Let's say we wish to use a scatterplot to divide two categories of data by drawing a line between them. It is impossible to depict part of the data in the plot on the left using Cartesian coordinates. In the figure to the right, we use polar coordinates to describe the data, which makes it easier to solve the problem using a vertical line.

Using machine learning to identify both the representation itself and the mapping from representation to output is one approach to solving this issue. Representation learning is the name given to this strategy. When compared to representations created by hand, learned representations frequently produce significantly higher performance. Additionally, they enable AI systems to quickly adapt to new jobs with a minimum of human involvement. For a simple task, a representation learning algorithm can find a decent set of features in a matter of minutes; for a difficult task, it may take hours or even months. It takes a lot of human time and effort to manually build features for a complex task; for a whole community of scholars, it can take decades.

The auto encoder is the classic illustration of a representation learning algorithm. An auto encoder combines a function called an encoder, which changes the input data's representation, with a function called a decoder, which changes the new representation back to the original format. Auto encoders are trained to make the new representation have a variety of desirable features, but they are also trained to preserve as much information as possible when an input is passed through the encoder and subsequently the decoder. Different autoencoder types strive to attain various features.

Our objective while creating features or algorithms for learning features is typically to identify the sources of variation that account for the observed data. In this context, we simply refer to individual sources of impact as "factors"; the factors are typically not multiplied together. These factors are frequently not clearly observable quantities. Instead, they might exist in the physical world as invisible forces or invisible things that have an impact on observable numbers.

Notes

They might also be mental constructs that help to explain or infer the causes of observed facts by providing helpful simplifications. They can be viewed as ideas or abstractions that aid in our understanding of the data's extensive range of variability. The age, sex, accent and words that the speaker uses are all sources of diversity when listening to a speech tape. The position of the car, its color and the sun's angle and brightness are all variables to consider while examining a photograph of a car.

The fact that many of the sources of variation have an impact on every single piece of data we are able to examine is a significant source of difficulties in many real-world applications of artificial intelligence. At night, it's possible that the individual pixels in an image of a red automobile are almost completely black. Depending on the angle of view, the silhouette of the car changes. The majority of applications call for us to separate the sources of variation and toss out the ones we don't care about. Of course, separating such high-level, abstract traits from raw data can be exceedingly challenging. Many of these sources of variance, such as a speaker's accent, cannot be found without a comprehensive understanding of the data that is almost human-level. At first look, representation learning does not appear to be helpful when getting a representation is nearly as challenging as solving the original problem.

By incorporating representations that be expressed in terms of other, simpler representations, deep learning addresses this fundamental issue in representation learning. The machine can create complicated notions from simpler ones thanks to deep learning. Figure below demonstrates how a deep learning system can combine simpler notions like corners and contours, which are in turn described in terms of edges, to represent the concept of an image of a human.

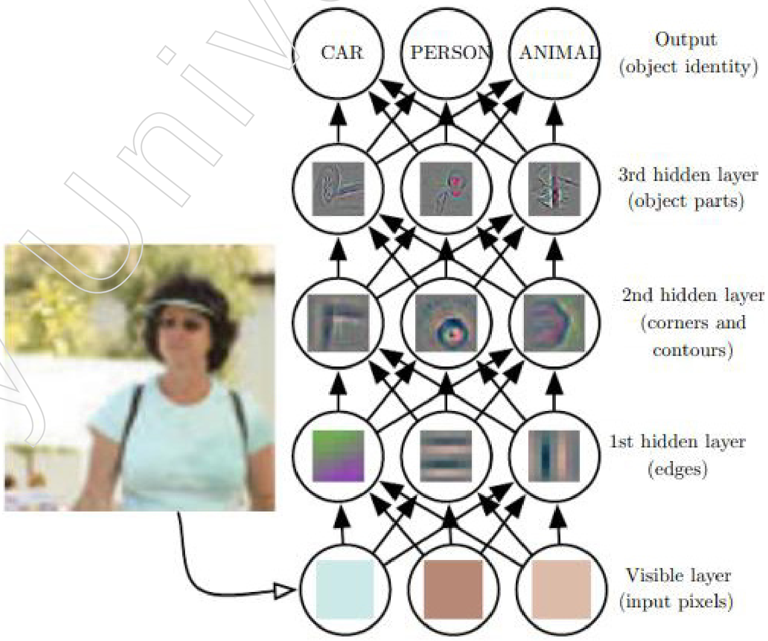


Figure: Illustration of a deep learning model

Computers face difficulties in comprehending raw sensory input data, such as a picture represented as a set of pixel values. Mapping a set of pixels to an object identity requires a substantial amount of effort. If approached directly, the task of learning or evaluating this mapping appears to be insurmountable. The solution to this difficulty is achieved through the utilisation of deep learning techniques. Deep learning effectively breaks down the intricate mapping necessary into multiple layers, each responsible for a distinct and simpler mapping inside the model.

Notes

The visible layer, referred to as such due to its inclusion of observable variables, serves as the interface for input presentation. The features of the image are subsequently retrieved through a series of hidden layers. The levels in question are commonly denoted as "hidden" due to the absence of explicit values in the observed data. Instead, the model is tasked with selecting the most suitable concepts to elucidate the relationships present in the observed data.

The visual representations in these images depict the characteristics represented by each hidden unit. The initial layer has the ability to rapidly detect edges by means of comparing the luminosity of adjacent pixels within a particular pixel set. The second latent layer may rapidly identify corners and extended contours, which are perceived as aggregations of edges, based on the edge representations provided by the first latent layer. The third latent layer possesses the capability to identify entire components of certain objects by the identification of specific clusters of corners and contours, based on the visual description provided by the second latent layer in terms of corners and contours. Finally, by utilising this depiction of the image in relation to its constituent elements, it becomes feasible to discern the entities depicted inside the visual representation.

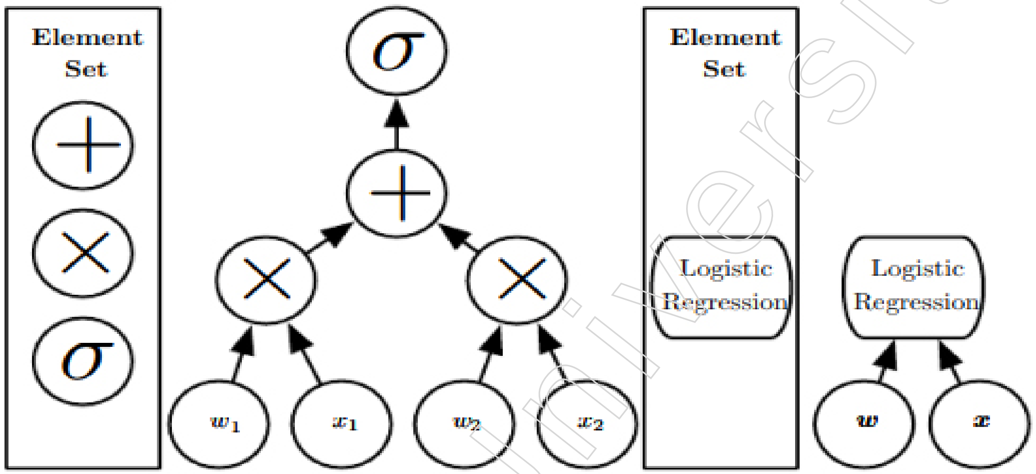


Figure:The depiction of computational networks, which map an input to an output, involves nodes that execute various operations.

Depth, which also depends on how a computational step is defined, is the distance that takes a calculation from input to output. The outcomes of a logistic regression model, where (wT x) is the input and is the logistic sigmoid function, are displayed in these graphs. If we use logistic sigmoids, addition and multiplication as the building elements of our computer language, this model has depth three. If we consider logistic regression as a component in and of itself, then this model has depth 1.

Not all of the data in an activation layer necessarily indicates reasons of variation that explain the input from a deep learning perspective. The representation also keeps track of state information so that a programme can run that can comprehend the input. This state information may be compared, like a counter or pointer in a normal computer programme. It doesn't directly affect the substance of the input, but it helps organise how the model processes data.

There are two techniques to gauge a model's depth. The first point of view centres on how many sequential operations are necessary to evaluate the architecture. A flow chart that illustrates how to calculate each output of the model given its inputs can be used to compare this to the length of the longest path. Just as two equivalent computer programmes will have various lengths based on the language the programme is written

Notes

in, the same function may be represented as a flowchart with varying depths depending on the functions we enable to be utilised as individual steps in the flow chart above. This language option can lead to two different measurements for the same architecture, as seen in Figure.

Deep learning is an AI methodology, to sum up. It specifically falls under the umbrella of machine learning, a technique that enables computer systems to improve over time and with experience. Machine learning is the only feasible approach for developing AI systems that can work in difficult real-world situations. Deep learning is a particular kind of machine learning that reaches a significant level of power and flexibility by learning to represent the world as a layered hierarchy of concepts, with each concept defined in relation to simpler concepts and more abstract representations computed in terms of less abstract ones. The image below shows the relationship between these several AI disciplines. The following graphic displays a high-level schematic of each function.

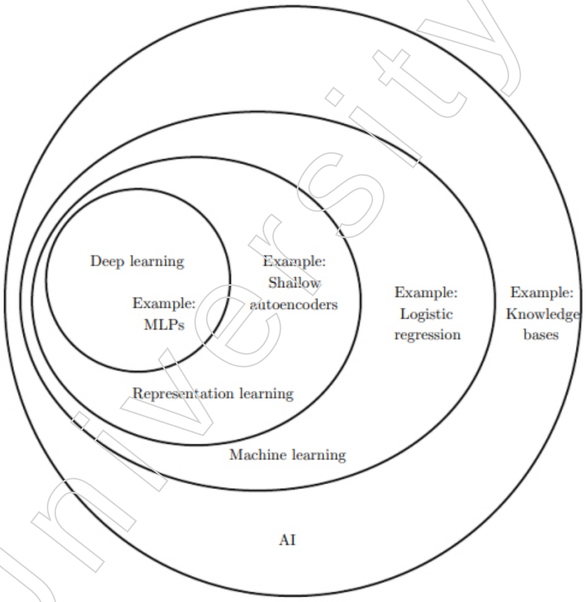


Figure: A Venn diagram illustrating how deep learning is a subset of machine learning, which is employed in many but not all methods to artificial intelligence. An illustration of an AI technology may be found in each section of the Venn diagram.

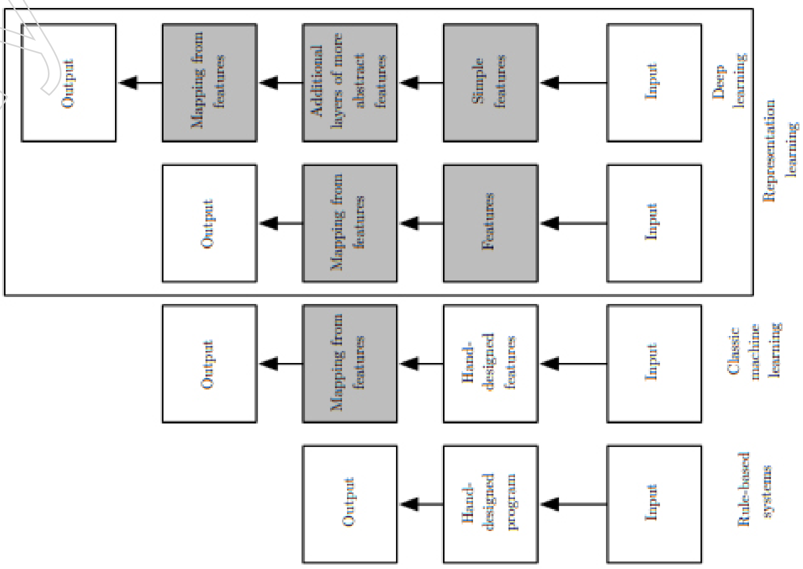


Figure: Flowcharts showing how the different parts of an AI system.

5.1.2 Introduction to Deep Learning: Part II

Historical Trends in Deep Learning: Deep learning is easy to comprehend in the perspective of history. Instead of giving a thorough history of deep learning, we highlight a few significant trends:

- Deep learning, despite its extensive and significant historical background, has been referred to by multiple titles that encompass diverse philosophical perspectives and its level of acceptance and recognition has experienced fluctuations throughout time.
- The increased availability of training data has led to the growing benefits of deep learning.
- The size of deep learning models has increased over time as computer hardware and software infrastructure has become more advanced.
- Over time, deep learning has more accurately and efficiently handled increasingly complex applications.

Deep learning, despite its rich and substantial historical foundation, has been characterised by several designations that span a range of philosophical viewpoints. Furthermore, its level of acceptance and acknowledgment has undergone variations throughout time. The proliferation of training data has resulted in the expanding advantages of deep learning.

The difficulties in feature engineering, which are essential for symbolic-based machine learning, can be overcome via deep learning. The amazing thing about deep learning is that it eliminates the need for computer programming entirely because the models can pick up the features automatically. Programmers simply need to provide the computer with a learning algorithm, expose it to terabytes of input data to train it and then let the machine figure out how to recognize the required items on its own. In other words, these computers can now learn on their own.

As a result, deep learning is a very potent technique for contemporary machine learning. On almost every metric, deep learning techniques outperform conventional symbolic-based machine learning techniques. The amount of money being invested, the number of people choosing deep learning as their field of study and the number of top technological businesses making AI the centerpiece of their long-term strategies are all indications that deep learning is on the rise. It has the ability to alter how individuals live their daily lives while revolutionizing many facets of machine perception. Some people even think that AI might one day be programmed to resemble human common sense.

Increasing Dataset Sizes:

Although the initial studies with artificial neural networks were carried out in the 1950s, one might question why deep learning has just lately come to be acknowledged as an important technology. Deep learning has been successfully applied in business since the 1990s, but until recently it was frequently thought of as more of an art than a technology and something that only a specialist could utilize. It’s true that using a deep learning algorithm effectively requires some level of competence. Fortunately, as the amount of training data increases, the level of skill needed decreases.

Though the models we train with these algorithms have undergone improvements that make it easier to train very deep architectures, the learning algorithms used to achieve human performance on hard tasks today are essentially identical to the learning algorithms that struggled to handle toy problems in the 1980s. The most significant innovation is that we can now provide these algorithms the tools they require to be successful.

Notes

Increasing Accuracy, Complexity and Real-World Impact

The initial deep models were designed to identify specific items in extremely small, closely cropped images. Since then, the size of the images that neural networks could analyze has gradually increased. Modern object recognition networks interpret detailed, high-resolution images without needing to crop the image in close proximity to the object to be identified. Similarly, while these contemporary networks routinely detect at least 1,000 different categories of items, the early networks could only distinguish two types of objects (or, in some circumstances, the absence or presence of a single kind of object).

The Image Net Large-Scale Visual Recognition Challenge (ILSVRC), which is held annually, is the biggest object recognition competition. The victory of a convolutional network in this challenge, surpassing its competitors by a substantial margin, signified a pivotal moment in the exponential growth of deep learning. The outcome of this triumph resulted in a notable decrease in the top-5 error rate, which was lowered from 26.1% to 15.3%. This signifies that the convolutional network generates a list of potential categories for each image and the accurate category was included in this list for all test samples, except for a mere 15.3%. Subsequently, deep convolutional networks have consistently emerged as the victors in these competitions. At the time of composing this text, the progress made in deep learning has resulted in a reduction of the most recent top-5 error rate in this competition to 3.6%, as depicted in the accompanying Figure.

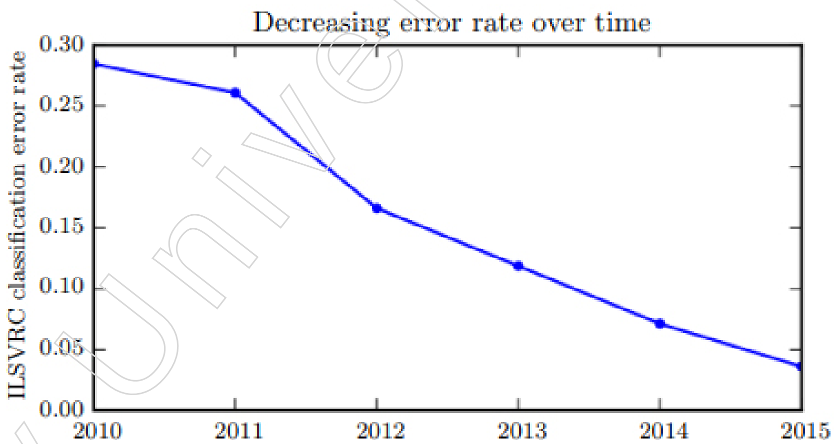


Figure:Deep networks have consistently won the competition every year and produced ever-lower mistake rates since they achieved the scale required to participate in the ImageNet Large Scale Visual Recognition Challenge.

Deep learning is a method of machine learning that has been developed over the past several decades and mainly relies on our understanding of the human brain, statistics and applied arithmetic. Its popularity and utility have greatly increased in recent years, largely as a result of more potent computers, greater datasets and methods for training deeper networks. The coming years will present many difficulties and chances for deep learning to advance and reach new heights.

5.2 Introduction to Neural Networks

Let's start with a working definition of what a "neuralnetwork" is, then move on to straightforward, straightforward explanations of some of the major elements in the definition:

A neural network refers to a network composed of interconnected processing units, also known as nodes, which exhibit a functional resemblance to the action of biological

Notes

neurons in animals. The network's processing capacity resides in the interunit connection strengths, also known as weights, which are obtained through a process of adaptation or learning from a given set of training patterns. In order to facilitate further elaboration, it is imperative to commence by briefly revisiting certain fundamental principles in the field of neuroscience. The human brain, consisting of approximately 100 billion neurons or nerve cells, is represented in a highly stylized fashion in this illustration.

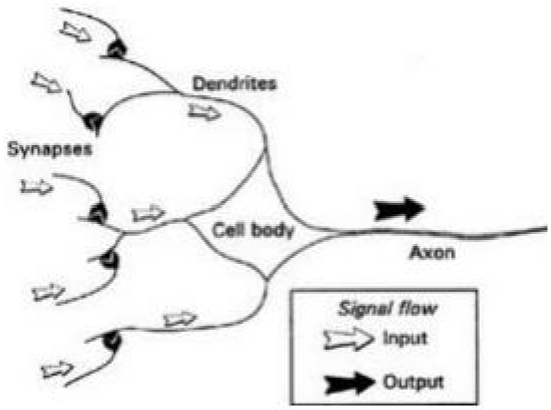


Figure: Essential components of a neuron shown in stylized form

Neurons utilise electrical signals, characterised as transient voltage fluctuations or "spikes," to facilitate intercellular communication. Synaptic electrochemical junctions, located on dendritic cell branches, facilitate interneuronal connectivity.

Typically, individual neurons possess numerous connections with other neurons, leading to a continuous stream of messages that ultimately converge at the cell body. In this context, the elements are merged or unified, resulting in the neuron generating a voltage impulse, commonly referred to as "firing," when the resultant signal surpasses a pre-established threshold. The axon, a dendritic extension, serves as a conduit for transmitting this information to adjacent neurons.

Certain incoming signals exhibit an inhibitory influence, impeding the firing process, whereas others possess an excitatory impact, promoting the generation of impulses. The synaptic connections of individual neurons, characterised by their type (excitatory or inhibitory) and intensity, are believed to play a crucial role in shaping the distinct information processing capabilities of each neuron.

The designation of connectionist systems arises from the recognition of the importance placed on interneuron connections. Consequently, the broader investigation of this approach is commonly known as connectionism. The objective is to integrate this particular design and processing technique into neural networks. The utilisation of the aforementioned terminology is commonly observed while examining neural networks within the framework of psychologically inspired models of human cognitive processes.

The nodes or units in our initial description serve as synthetic counterparts to biological neurons, with a representative illustration shown in Figure below. In the context of neuronal communication, it is customary to assign a weight to each input signal prior to its transmission to the cell body. This is due to the fact that synapses, which serve as the connections between neurons, are typically represented by a singular numerical value denoting their strength or efficacy. In this context, the activation of a node is generated through the summation of weighted signals, employing fundamental arithmetic operations. The comparison of activation to a threshold occurs in a node known as a threshold logic unit (TLU), as illustrated in the accompanying figure. If the activation

Notes

surpasses the threshold, the TLU generates a high-valued output, typically represented as "1". Otherwise, it produces a zero output. Exclusively positive weights have been employed. The diagram utilises arrow widths to symbolise signals and multiplication symbols enclosed in circles to represent weights. The values assigned to the circles are intended to be proportional to the size of the symbol. The TLU, or Threshold Logic Unit, serves as a fundamental model of a synthetic neuron.

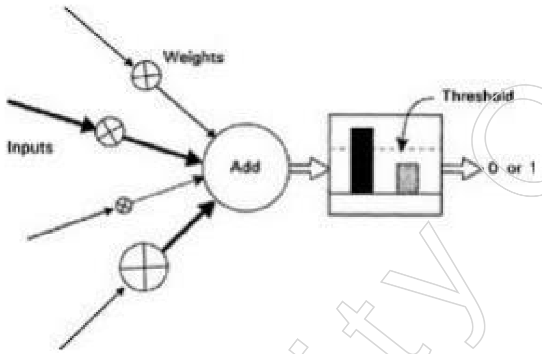


Figure : Simple artificial neuron.

Any artificial neuronal network shall be referred to as a "network" in this paper. This could be as basic as a single node or as complex as a big group of nodes, each of which is connected to every other node in the network. The figure below illustrates one kind of network.

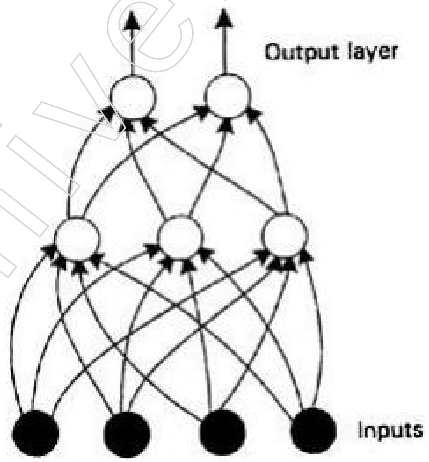


Figure: Simple example of neural network.

In the current configuration, the weights are incorporated implicitly in all connections, although the representation of each node remains limited to a circular shape. Every signal is generated from an input and undergoes two successive nodes prior to reaching an output, at which point it undergoes no further modifications. The placement of the nodes follows a tiered framework.

The feed forward structure, among several alternatives, is frequently employed to categorise an input pattern into one of multiple categories, relying on the resulting output pattern. The output layer, situated at the highest position in the picture, may consist of 26 nodes, each representing a letter of the alphabet. These nodes are utilised to determine the specific letter category to which the input character belongs. For example, if the input is a representation of the light and dark patterns in an image of handwritten letters.

To do this, one output node would be assigned to each class and it would be necessary for only one of these nodes to fire whenever a pattern from the relevant class

Notes

was provided at the input. So much for the fundamental structural components and how they work. Returning to our working definition, take note of the significance placed on experience-based learning. Real neurons' synaptic strengths may be altered in specific situations to allow each neuron to respond differently or adapt to its unique sensory input. The equivalent in artificial neurons is the alteration of the weight values. The "knowledge" the network is meant to have is stored in its weights, which develop through a process of adaptation to stimulus from a set of pattern instances. There are no computer programs involved in this information processing.

An input pattern is supplied to the net in one training paradigm known as supervised learning, which is employed in conjunction with nets of the kind shown in Figure above and its response is then compared with a goal output. For instance, if we were using our prior letter recognition example, we might enter "A" and compare the network output to the classification code for A. The weights are then adjusted based on the discrepancy between the two output patterns.

Every specific recipe for change functions as a learning principle. A new pattern is provided, the output is compared to the target and fresh alterations are applied after the necessary weight updates have been made. Iteratively repeating this series of events increases the likelihood that the network's behavior will eventually converge and each pattern's response will be near to the matching target. The entire procedure, including any pattern presentation ordering, process termination requirements, etc., makes up the training algorithm.

Why study neural networks?

Neural networks are commonly utilised in statistical analysis and data modelling as an alternative to conventional nonlinear regression or cluster analysis techniques. Consequently, these methods are commonly utilised in situations that pertain to categorization or forecasting. Illustrative instances encompass textual character recognition, image and speech recognition, as well as human sectors of expertise such as medical diagnosis, oil exploration geology and forecasting financial market indicators.

Neural networks are perceived by engineers and computer scientists as a form of parallel distributed computing, presenting a viable alternative to the conventional algorithmic approaches that have historically dominated the field of machine intelligence. This particular problem also pertains to the field of classical artificial intelligence (AI). Practitioners in this domain often prioritise the ease of implementing solutions in digital hardware or the efficacy and accuracy of particular approaches, rather than emphasising biological realism.

Nets, referred to as computer models of the animal brain, have garnered attention from neuroscientists and psychologists due to their abstraction of key properties of nerve tissue believed to be vital for information processing. There exists a degree of scepticism among neuroscientists over the final effectiveness of these simplified models, since they argue that a more comprehensive level of information is necessary to provide a thorough understanding of the mechanisms behind brain function. The artificial neurons employed in connectionist models often exhibit significant simplifications compared to their biological counterparts. The ultimate outcome remains uncertain, however, notable advancements have been made in the realm of replicating brain functions by the utilisation of knowledge pertaining to the interconnectedness of actual neurons, referred to as local "circuits."

- Neural networks were first developed in the early 1940s. In the late 1980s, it saw a rise in popularity. This came about as a result of new methods and innovations

Notes

being discovered as well as general advancements in computer hardware technology.

- Some NNs are models of biological neural networks, but historically, the field was inspired by the desire to create artificial systems capable of complex, possibly "intelligent," computations similar to those the human brain performs, thereby improving our understanding of the brain.
- Most NNs have a "training" rule of some kind. In other words, NNs "learn" from examples in the same way that kids pick out dogs from pictures of dogs and show some ability to generalize outside of the training data.

NNs vs Computers:

Digital Computers	Neural Networks
Reasoning deductively. To produce output, we use input data and established rules.	Reasoning inductively. We build the rules using training examples as input and output data.
Centralized, synchronous and serial computation all exist.	Collective, asynchronous and parallelism all describe computation.
Memory is physically stored, addressable by location and packetized.	Memory is internalized, dispersed and addressable by content.
not tolerant of faults. It stops working when one of the transistors fails.	Redundancy, fault tolerance and responsibility sharing.
Exact.	Inexact.
Static connectivity.	Dynamic connectivity.
Provided applicable and provided the rules and input data are clearly established..	Relevant when data is noisy or incomplete or when rules are ambiguous or complex.

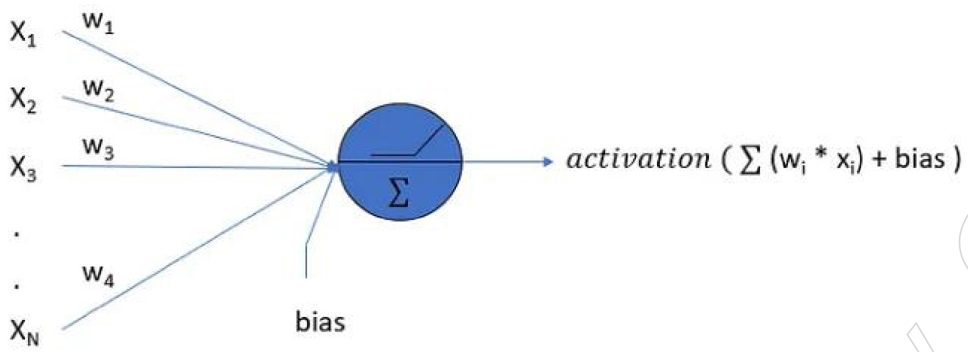
Neural networks are designed to mimic the information processing mechanisms of the human brain, enabling them to replicate fundamental cognitive capabilities. Due to its efficient processing and rapid reaction capabilities, it is employed for a diverse range of real-time tasks.The construction of an artificial neural network encompasses various components that are derived from the biological nervous system. An artificial neural network is composed of numerous interconnected processing components, sometimes referred to as Nodes. These nodes establish a connection link with other nodes in order to facilitate their interconnection.

The connection link is characterised by the presence of weights, which serve to encapsulate information pertaining to the incoming signal. The weights undergo updates with each iteration and input. The weights of the neural network and its architectural configuration are commonly referred to as the "trained neural network" once all training data instances have been processed. This method is commonly referred to as the training of neural networks. In order to address the specific concerns delineated in the problem statement, the employed neural network model is utilised. Artificial neural networks have the capability to address a wide range of tasks, including but not limited to classification problems, pattern recognition and data clustering.

Artificial neural networks are used because they can learn quickly and adapt. They are able to use the training data they receive to learn "how" to tackle a particular problem. After learning, you can utilize it to address that particular problem fast, effectively and

Notes

accurately. Air Traffic Control, Optical Character Recognition, which is utilized by various scanning apps like Google Lens, Voice Recognition, etc. are some examples of real-world neural network applications.



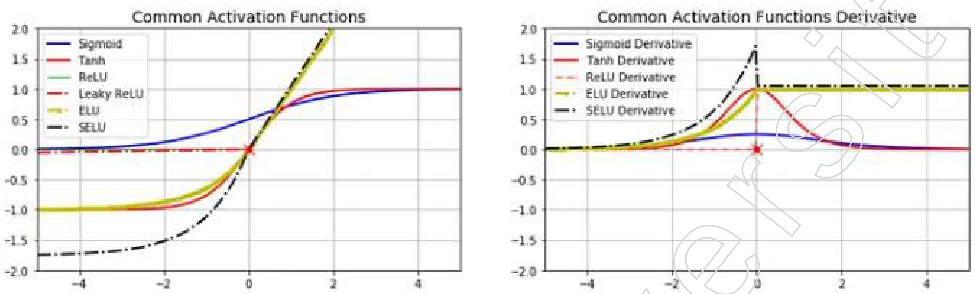


Figure: Common activation functions in neural networks

The activation function is a crucial hyper-parameter during the training of neural networks and we must decide which activation function to employ in both the hidden layers and the output layer. The activation function determines whether or not to fire a neuron by calculating the bias term and the weighted total of the input. The input signal is subjected to an activation function. which may be a linear or non-linear transformation and the output is given as input to the next layer of neurons.

Basic building blocks of neural network training:

It is vital to explain the fundamental building blocks of neural network training before going on to other activation functions and their variants. These blocks can be further broken down into the forward pass, the network output error measurement and the backward pass. The training cases are first given to the neural network in forward pass. A forward cascade of calculations is then performed utilizing the current set of weights across the layers to get the network's output prediction. Second, the discrepancy between the desired output and the actual projected output—the output error of the network—is measured. Thirdly, we back-propagate through each layer to determine the contribution of each connection to the error before adjusting the connection weights to lower the error.

Humans Versus Computers: Stretching the Limits of Artificial Intelligence:

One additional advantage is that neural networks provide a rapid means of adjusting the complexity of a model by incorporating or removing neurons from the architecture based on the quantity of training data or computational resources at hand. The increasing availability of data and the enhanced computational powers of modern computers have surpassed the limitations of traditional machine learning methods, rendering them

Notes

incapable of fully harnessing the current possibilities. This phenomenon elucidates a substantial proportion of the recent achievements observed in neural networks. The depicted figure illustrates the given condition.

	Artificial Intelligence	Human Intelligence
Processing	Based on algorithms and mathematical models	Based on cognitive processes and biological structures
Learning	Based on data and feedback loops	Based on experience, intuition, and creativity
Speed	Can process data and perform tasks much faster than humans	Slower than AI in processing large amounts of data, but can make complex decisions quickly
Adaptability	Can quickly adapt to new data and situations	Can adapt to new situations, learn from experience, and make decisions based on context
Emotions	Lacks emotions and empathy	Capable of feeling emotions and empathy
Creativity	Limited ability to be creative or think outside of the box	Capable of creativity, imagination, and innovation
Ethics	Does not have a moral code or conscience	Has a moral code and conscience that guides decision-making
Physical Limitations	Does not have physical limitations, can operate 24/7	Limited by physical capabilities and requires rest and maintenance

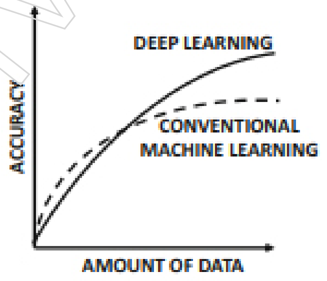


Figure:An example of comparing the precision of a big neural network and a standard machine learning algorithm. When there is enough data and computer power, deep learning becomes more appealing than traditional methods. There has been a "Cambrian explosion" in the use of deep learning technology in recent years as a result of an increase in computer power and data accessibility.

5.2.1 The Neuron Model:

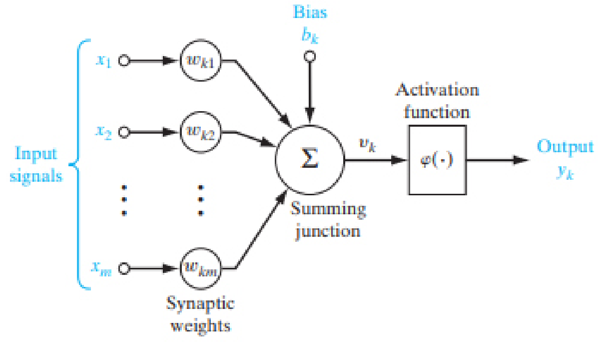


Figure : Nonlinear model of a neuron, labeled k.

Notes

A neuron is an information-processing cell that is essential to a neural network's functionality. The model of a neuron, which serves as the inspiration for creating a large family of neural, is shown in the image below. Here, we list the neural model's fundamental three components:

A compilation of synapses, also referred to as connecting points, wherein each synapse possesses a unique weight or strength. At the input of synapse j, which is linked to neuron k, the signal xj is multiplied by the synaptic weight wkj. It is advisable to pay attention to the notation used for the subscripts of the synaptic weight wkj.

The terminal region of the synapse, which is associated with the weight, is commonly known as the. The synaptic weight of an artificial neuron exhibits a range of values that can fluctuate between negative and positive, in contrast to the synaptic weight of a biological synapse in the brain.

- A linear combiner is a device that adds input signals and weights them according to the strength of the synapses in each neuron.
- a neuron's output amplitude is limited by an activation function.The output signal's permitted amplitude range is squashed (limited) to a specific value by the activation function, which is also known as a squashing function.
- Typically, the closed unit interval [0,1] or, alternatively, [-1,1] is used to represent the normalized amplitude range of a neuron's output.
- The externally applied bias, represented by bk, is likewise included in the neural model of the previous Figure. Depending on whether it is positive or negative, the bias bk has the effect of either raising or reducing the net input of the activation function.
- We may write the following two equations to represent the neuron k shown in the previous Figure in mathematical terms:

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \qquad y_k = \varphi(u_k + b_k)$$

The input signals are denoted as x1, x2,... andxm, while the corresponding synaptic weights of neuron k are represented as wk1, wk2,... andwkm. The output of the linear combiner resulting from the input signals is denoted as uk and bk represents the bias. The activation function is denoted as (•) andyk represents the output signal of the neuron.In the depicted model illustrated in the aforementioned Figure, the utilisation of bias bk results in the application of an affine transformation on the output uk of the linear combiner, as exemplified by:

$$v_k = u_k + b_k$$

In particular, the relationship between the induced local field, or activation potential, vk of neuron k and the linear combiner output uk is altered in the way depicted in Figure below and these two terms will now be used interchangeably. This depends on whether the bias bk is positive or negative. Note that the graph of vk versus uk no longer crosses through the origin as a result of this affine change.

Notes

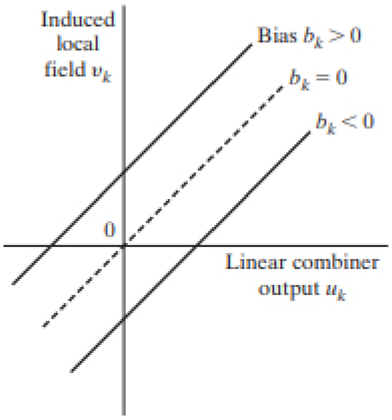


Figure : Affine transformation produced by the presence of a bias; note that vk = bk at uk = 0.

An external parameter of neuron k is the bias (bk). We can explain its existence using the equation above. The combination of the two equations can be expressed equivalently as follows:

$$v_k = \sum_{j=0}^{m} w_{kj} x_j \quad \text{and} \quad y_k = \varphi(v_k)$$

As a result, we can reformulate the neuron k model, as seen in Figure below. The influence of the bias is taken into consideration in this figure in two ways:

- introducing a fresh input signal fixed at +1 and
- a fresh synaptic weight corresponding to the bias bk. Despite the visual differences between the two Figutre models, they are technically equal.

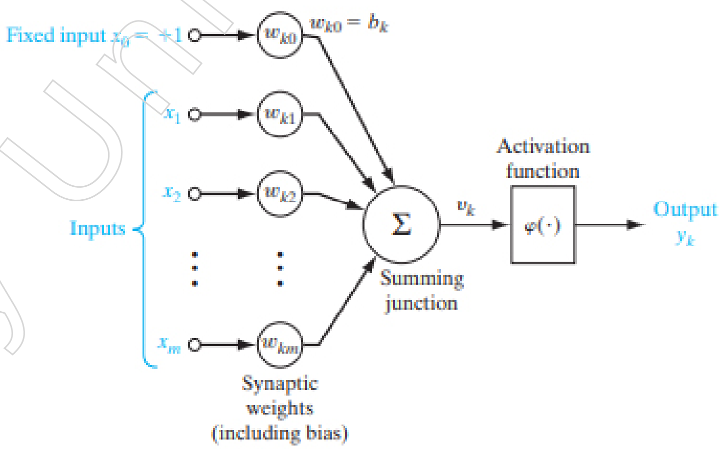


Figure: Another nonlinear model of a neuron; accounts for the bias bk w .

Types of Activation Function: The output of a neuron is defined by the activation function, indicated as (v), in terms of the induced local field v. Following, we define two fundamental categories of activation functions:

Threshold Function: Figure following describes this type of activation function.,

$$\varphi(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{if } v < 0 \end{cases}$$

we have:

Notes

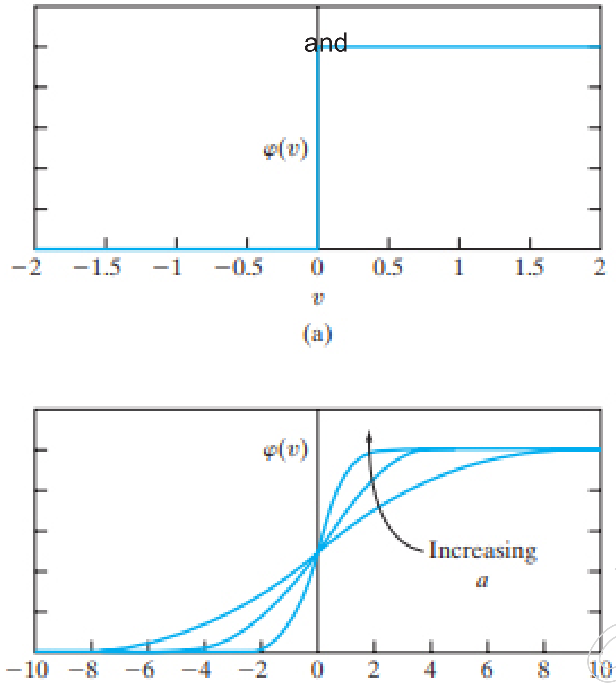


Figure : (a) Threshold function. (b) Sigmoid function for varying slope parameter a.

Such a neuron is known as the McCulloch-Pitts model in neural computation in honor of the groundbreaking work carried out by McCulloch and Pitts. In this paradigm, a neuron's output has a value of 1 if its induced local field is nonnegative and a value of 0 otherwise.The McCulloch-Pitts model's all-or-none attribute is described by this claim.

Sigmoid Function: The most typical type of activation function utilized in the creation of neural networks is the sigmoid function, whose graph is "S"-shaped. It is described as a strictly rising function with a delicately balanced linear and nonlinear behavior. The logistic function is an illustration of a sigmoid function and is defined as follows:

$$\varphi(v) = \frac{1}{1 + \exp(-av)}$$

where an is the sigmoid function's slope parameter. We can get sigmoid functions with various slopes by changing the parameter a, as shown in Fig.below. In actuality, a/4 is the slope at the origin. The sigmoid function reduces to a threshold function in the limit as the slope parameter approaches infinity.In contrast to threshold functions, which only consider values between 0 and 1, sigmoid functions consider a continuous range of values between 0 and 1.

In the sense that its input-output behavior is exactly defined for all inputs, the neural model illustrated in the aforementioned Fig. is deterministic. It is preferable to use a stochastic neural model as the foundation for the analysis in some neural network applications.

The activation function of the McCulloch-Pitts model is given a probabilistic interpretation using an analytical trace table approach. A neuron is only allowed to exist in one of two states, let's say +1 or -1. A neuron's decision to fire—that is, to change from a "off" state to a "on" state—is probabilistic. Let x represent the neuron's current state and P(v) represent the likelihood that it will fire, with v standing for the neuron's induced local field. We may then type:

Notes

$$x = \begin{cases} +1 & \text{with probability } P(v) \\ -1 & \text{with probability } 1 - P(v) \end{cases}$$

The sigmoid function is a common option for P(v):

where T is a pseudotemperature that is utilized to regulate the noise level and, consequently, the firing uncertainty. However, it is crucial to understand that T is not the actual temperature of a neural network, whether it be a biological or synthetic neural network. As was already mentioned, we should instead consider T to be merely a variable that regulates the thermal fluctuations that represent the effects of synaptic noise. The stochastic neuron given by the two equations above simplifies to the McCulloch-Pitts model, which is a noiseless (i.e., deterministic) form, when.

5.2.2 The Neural Network and Its Utility in Modelling and Solving Problems

The Neural Network:The understanding that the human brain computes entirely differently from the traditional digital computer has driven research on artificial neural networks, sometimes known as "neural networks," from their beginnings. The brain is a very sophisticated, parallel, nonlinear computer (information processing system). It has the ability to organize the neurons that make up its structural components such that it can carry out some computations (including pattern recognition, perception and motor control) much more quickly than the current fastest digital computer. Take human vision as an example, which is an example of an information-processing activity.

The visual system's job is to give us a picture of our surroundings and, more importantly, to give us the knowledge we need to interact with that environment. To be more precise, activities of considerably lower complexity take a very long time on a powerful computer, whereas the brain routinely completes perceptual identification tests (e.g., recognizing a familiar face embedded in an unfamiliar scene) in about 100-200 ms.

Another illustration would be a bat's sonar. An active echolocation system is sonar. Bat sonar not only communicates information about the distance to a target (such as a flying bug), but also about the relative velocity, size and size of numerous elements on the target, as well as the azimuth and elevation of the target. A plum-sized brain performs the intricate neural computations required to retrieve all of this information from the target echo. An engineer working on radar or sonar would be jealous of how easily and successfully an echolocating bat can pursue and catch its prey.

So how does a bat's brain or a human's brain accomplish this? A brain has significant structure from birth and the capacity to develop its own set of behavioral guidelines through what we typically refer to as "experience." In fact, experience is acquired over time. The human brain is hardwired in large part during the first two years after birth, but development continues even after that point.

A massively parallel distributed processor made up of simple processing units, known as a neural network, has a built-in inclination for storing and making use of experiencing information.

It resembles the brain in two respects:

- Through a process of learning, the network picks up knowledge from its surroundings.
- The learned information is stored in synaptic weights, which are the strengths of interneurons.

A learning algorithm refers to the computational procedure employed to facilitate the execution of the learning process. The intention of this process is to systematically modify the synaptic weights of the network in order to attain a predetermined design objective. The prevailing methodology employed in the creation of neural networks involves the manipulation of synaptic weights. This approach closely aligns with the principles of linear adaptive filter theory, a well recognised and effectively employed framework across various academic domains. The potential for neural networks to modify their own design arises from the capacity of neurons in the human brain to undergo cell death and the formation of new synaptic connections.

Benefits of Neural Networks:

It is clear that a neural network gets its processing power from two sources: first, its highly parallel distributed topology and second, its capacity for learning and generalization. Generalization is the neural network's ability to generate plausible results for inputs that weren't present during training (learning). With the help of these two information-processing abilities, neural networks are able to identify reliable approximations to difficult complex (large-scale) problems. However, in real-world applications, neural networks cannot deliver the answer on their own. In place of that, a consistent system engineering methodology needs to incorporate them. In particular, an interesting difficult problem is broken down into a number of relatively easy tasks and neural networks are given a subset of those tasks that are compatible with their natural skills. But it's vital to understand that before we can create a computer architecture that resembles the human brain, we still have a long way to go (if ever).

Neural networks offer the following useful properties and capabilities:

Nonlinearity: Linear and nonlinear artificial neurons can also be implemented. A neural network possesses inherent nonlinearity due to its composition of interconnected nonlinear neurons. Moreover, the characteristic of nonlinearity in this context is distinct in that it is distributed across the entire network. The feature of nonlinearity holds significant importance, particularly when the input signal (e.g., a speech signal) is inherently nonlinear due to the physical mechanism creating it.

Input–Output Mapping: A common learning paradigm known as supervised learning or learning with a teacher includes changing the synaptic weights of a neural network using a set of labeled training examples, or task examples. A distinct input signal and a corresponding desired (target) response make up each sample.In order to minimize the difference between the desired response and the actual response of the network produced by the input signal in accordance with an appropriate statistical criterion, the network's synaptic weights (free parameters) are modified. This is done by presenting the network with an example chosen at random from the set. Until the network reaches a stable state where there are no more noticeable changes in the synaptic weights, the network training is repeated for numerous examples in the set.

Training examples that have already been used may be used again, but in a new sequence.As a result, the network builds an input-output mapping specific to the given challenge in order to learn from examples. A similar strategy is similar to the study of nonparametric statistical inference, a field of statistics that focuses on model-free estimates, or, from a biological perspective, Tabula rasa education In this context, the term "nonparametric" refers to the absence of any prior assumptions regarding a statistical model or the input data. Take a look at a pattern classification task as an illustration. In this challenge, you must assign a physical object or event represented by an input signal to one of several predetermined categories (classes). To "estimate"

Notes

arbitrary decision boundaries without using a probabilistic distribution model in the input signal space for the pattern-classification task is the aim of a nonparametric approach to this problem. A similar perspective is implicitly adopted by the supervised learning paradigm, which draws parallels between the input-output mapping performed by a neural network and nonparametric statistical inference.

Adaptivity: In neural networks, synaptic weights can automatically change in response to environmental changes. For example, a neural network can be swiftly retrained to adapt to minor changes in the environmental conditions that it operates in after being trained to function in a specific context. When working in a nonstationary environment (one whose statistics change over time), a neural network may be taught to modify its synaptic weights in real time.

A neural network is a useful tool for adaptive pattern classification, adaptive signal processing and adaptive control due to its adjustable capabilities and natural architecture for pattern classification, signal processing and control applications.

As a general rule, it can be asserted that a system's performance is expected to be more dependable when the system is required to operate in a nonstationary environment the more adaptable the system is built to be while still retaining stability. But it's important to remember that adaptivity doesn't always lead to robustness; in fact, it might even have the opposite effect.

For example, a quick-changing adaptive system may be more prone to respond to fictional disruptions, which could cause a sudden fall in system performance. To fully use adaptivity, the major temporal constants of the system must be long enough for it to ignore spurious disturbances and short enough for it to respond to significant environmental changes.

Evidential Response: A neural network can be created to provide information regarding the confidence in the judgment made as well as which specific pattern to choose when it comes to pattern classification. If ambiguous patterns do appear, this later knowledge may be used to eliminate them and boost the network's classification performance.

Uniformity of Analysis and Design: In essence, neural networks are versatile data processors. We say this in the sense that neural networks are applied in all domains using the same nomenclature. This characteristic can appear in various ways:

- All neural networks contain neurons in one way or another as a basic component.
- This similarity enables the sharing of theories and learning algorithms across many neural network applications.
- It is possible to create modular networks by seamlessly integrating components.

VLSI Implementability: A neural network has the potential to be quick for the calculation of some tasks due to its massively parallel nature. A neural network is highly suited for use with very-large-scale-integrated (VLSI) technology because of the same characteristic. The ability to capture extremely complicated activity in a highly hierarchical manner is one of the advantages of VLSI.

Contextual Information: A neural network's physical composition and level of activation serve as a representation of knowledge. The overall activity of every neuron in the network has the capacity to influence every individual neuron. As a result, a neural network handles contextual information organically.

Fault Tolerance: In terms of resilient computation, a hardware-implemented neural network has the potential to be inherently fault tolerant, meaning that performance declines smoothly under challenging operating conditions. For instance, recall of a stored pattern is of worse quality if a neuron or one of its connecting pathways is destroyed. However, because the network's data is spread, significant harm must be done before the network's general responsiveness is substantially compromised. Therefore, a neural network in theory displays a gentle performance decline as opposed to catastrophic collapse. Robust computation has some empirical support, but this support is typically unregulated. It could be required to make corrections when creating the algorithm to train the network in order to ensure that the neural network is, in fact, fault resistant.

Neurobiological Analogy: The brain, which serves as living proof that fault-tolerant parallel processing is not only physically possible but also quick and effective, serves as the inspiration for the creation of a neural network. (Artificial) neural networks are used as a research tool by neurobiologists to interpret neurobiological events. Engineers, on the other hand, look to neurobiology for novel solutions to challenges that are more complex than those based on traditional hardwired design methods. The following two examples, which are each representative of one of these two points of view:

- In comparison to neural network models based on recurrent networks, linear system models of the vestibulo-ocular reflex (VOR) are more straightforward. The oculomotor system includes the vestibulo-ocularbreflex. By rotating the eyes counterclockwise to the head, VOR works to keep the retinal image (i.e., visual) stable.Premotor neurons in the vestibular nucleus, which receive and interpret head rotation information from vestibular sensory neurons, mediate the VOR by sending the results to the motor neurons in the eye muscles.Because both the VOR's input (head rotation) and output (eye rotation) can be precisely set, it is highly suited for modelling. It is also a very straightforward reflex and the constituent neurons' neurophysiological characteristics have been extensively discussed. The vestibular nuclei's premotor neurons are the most intricate and fascinating of the three different neuronal kinds.The VOR has previously been modeled using control theory and lumped, linear system characteristics. Although these models helped to explain some of the VOR's general characteristics, they provided little information about the characteristics of the individual neurons that make up the VOR. Neural network modelling has significantly improved this scenario. Many of the static, dynamic, nonlinear and dispersed elements of signal processing by the neurons that mediate the VOR, especially the vestibular nuclei neurons, can be reproduced and explained using recurrent network models of VOR.
- More than any other area of the brain, the retina is where we start to make connections between the initial cerebral images and the outside world, which is represented by a visual sense, as well as between that sense's actual physical image projected onto a variety of receptors. The retina is a delicate layer of neural tissue that covers the back of the eye. It is the job of the retina to transform an optical image into a neural image that may then be sent down the optic nerve to a variety of locations for additional analysis.As seen by the synaptic structure of the retina, this is a difficult undertaking. The conversion of an optical image into a neural image occurs in the retinas of all vertebrates in three steps: phototransduction by a layer of receptor neurons; transmission of the resulting signals (produced in response to light) by chemical synapses to a layer of bipolar cells; and transmission of these signals, again by chemical synapses, to output neurons known as ganglion cells.

In synaptic stages, there exist specialised neurons known as horizontal cells and amacrine cells, which are laterally coupled. The primary role of these neurons is to modulate the transmission of synaptic signals between different layers.Inter-plexiform

Notes

cells serve as centrifugal elements responsible for conveying impulses from the inner synaptic layer to the outer synaptic layer. Scientists have successfully developed electrical chips that have a striking resemblance to the structural characteristics of the human retina. The term “neuromorphic integrated circuits” refers to electronic chips.

The neuromorphic image sensor has an array of photoreceptors interconnected with analogue circuitry, whereby each pixel is represented. The artificial system exhibits characteristics similar to the human retina, since it possesses the ability to perceive edges, respond to small variations in luminosity and identify patterns of motion. One notable advantage of the neurobiological analogy, as exemplified by neuromorphic integrated circuits, is its capacity to instill optimism and conviction, while also offering some degree of substantiation, regarding the potential beneficial effects of applying knowledge about neurobiological structures to the advancement of electronics and VLSI technology for neural network development.

5.2.3 Connect to the Biological Motivations and Parallelism

Evolution and Parallel Processing: Evolution has evolved creatures in the biological world to maximize their chances of survival and procreation. In biology, parallelism is demonstrated by the simultaneous occurrence of several processes within an organism. For instance, the body’s cells do a variety of tasks simultaneously, including digestion, breathing and circulation. This simultaneous processing improves flexibility and efficiency.

Neural Networks and Brain Parallelism: One of the most intricate biological structures, the human brain, demonstrates astounding parallelism. Neurons process information simultaneously, enabling quick perception, thought and action. In the field of machine learning, artificial neural networks were created as a result of this parallelism notion. These networks enable computers to carry out tasks like image recognition and natural language processing by simulating the connectivity of neurons.

Genetic Parallelism and Diversity: The parallel evolution of various species or populations in response to various environmental conditions gives rise to genetic diversity. This parallelism is essential for the survival of all life on Earth because it enables species to adapt to shifting environmental conditions. Similar to this, parallelism is used in computers to handle difficult operations by breaking them down into smaller, more manageable components that may be performed concurrently, improving overall efficiency.

Ecosystems and Distributed Computing: The effectiveness of parallelism in nature is shown by ecosystems. In parallel, several species interact and contribute to the efficiency of the ecosystem. Breaking down workloads into smaller subtasks and executing them on different machines is known as distributed computing, a parallel processing strategy in technology. This is similar to how several species work together to maintain the balance of an ecosystem.

Biological Hierarchies and Parallel Architectures: From cells to tissues to organs to entire animals, biological systems frequently display hierarchical order. Parallelism is made easier by this hierarchical structure because different levels can carry out separate tasks at the same time. Similar to this, parallel architectures in computing are created with numerous layers, each of which increases the system’s overall processing capacity and effectiveness.

Survival Strategies and Task Parallelism: Different parallelism-based survival strategies have been created by biological creatures. Animals, for example, may find food, fend off predators and reproduce all at once. Task parallelism in computing refers to the division of a task into smaller, concurrently executable tasks. This method boosts efficiency and speed in a manner similar to how nature multitasks.

5.2.4 Popular CNN (Convolutional Neural Network) Architectures:

Convolutional Neural Network:

Artificial intelligence is rapidly narrowing the disparity in skills between robots and humans. A multitude of scholars and non-professionals are engaged in diverse facets of the field of artificial intelligence with the aim of developing remarkable innovations. The discipline of computer vision is a remarkable area of study. The primary goal of computer vision is to emulate the perceptual abilities of humans in artificial systems. Examples of well-known computer vision applications include image detection, image tagging, image recognition, image classification, image analysis, video analysis and natural language processing.

The emergence and advancement of deep learning techniques in the field of computer vision has garnered significant interest and engagement from numerous researchers over the course of several years. CNN is the predominant choice for constructing the majority of computer vision algorithms. Convolutional neural networks employ deep learning techniques to assign learnable biases and weights to various objects within an input image, enabling their differentiation.

CNN requires less preprocessing than other techniques. The best learning algorithm for understanding visual content is therefore a CNN. Additionally, it has proven to be exceptionally good at segmenting, retrieving and classifying images. People outside of academia are now interested in CNN because of its success. Microsoft, Google, AT&T, NEC and Facebook are just a few of the businesses working to expand CNN architecture.

Additionally, they possess research teams that are actively engaged in investigating state-of-the-art convolutional neural network (CNN) architectures. Currently, the majority of leading participants in image processing and computer vision competitions predominantly utilise models based on deep Convolutional Neural Networks (CNNs). There exist numerous modifications of the fundamental Convolutional Neural Network (CNN) design. An examination of the introduction to CNN, the evolution of CNN over time, various design aspects employed by CNN and an architectural analysis of each form of CNN, including an assessment of their respective strengths and weaknesses.

This is the first review that almost covers every aspect of CNN for computer vision, including its background, CNN architecture designs, advantages and disadvantages, applications and the work that needs to be done in the future.

- This review helps readers decide wisely on their future research in the field of CNN for computer vision;
- The limits of the current CNN architectures are included in this manuscript, which inspires the construction of a new architecture. Additionally, it outlines the benefits and drawbacks of practically all common CNN variations;
- This survey paper's fascinating section classifies the CNN architecture into eight types based on its implementation criteria;
- In addition, other CNN applications are described so that readers might use CNN in areas other than computer vision;
- It offers a detailed overview of upcoming research directions in the field of CNN for computer vision.

Notes

This research paper is organized with the first portion providing a thorough understanding of CNN. The parallels between CNN and the visual brain of the ape are then explained.

CNN components: The following figure displays different CNN elements. It's crucial to comprehend the many CNN components and their uses in order to learn about the developments in CNN architecture:

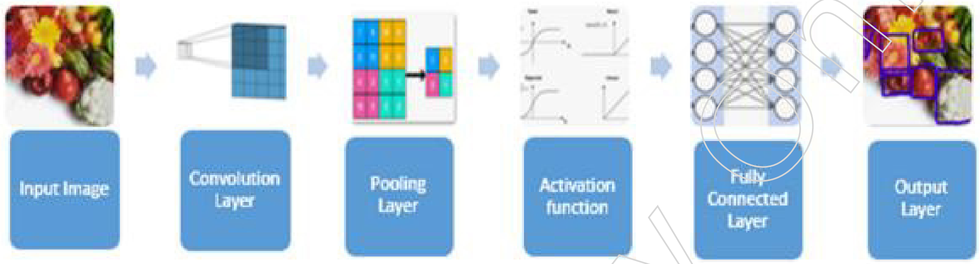


Figure: various CNN components.

Input Image: The building blocks of a computer image are called pixels. They serve as the binary representation of the visual data. The digital image is composed of a series of pixels with a range of 0-255 that are arranged in a matrix-like pattern. The brightness and hue of each pixel are specified by its pixel value. In the first second after seeing an image, human brains digest a huge amount of data. In order to span the whole visual field, each neuron in the human brain has its own receptive field and is connected to other neurons. Each neuron in the biological vision system responds to stimuli in the receptive field, which makes up a very small percentage of the visual field. Similar to this, each neuron in CNN only processes data inside its specific receptive area. Before moving on to more complicated patterns, like faces and objects, the CNN layers are designed to recognize simpler patterns first, such lines and curves. Therefore, it is conceivable to assert that utilizing a CNN might provide machines eyesight.

Convolution Layer: In the CNN design, the convolution layer is a crucial layer. As illustrated in Figure below, it utilizes a 3 × 3 or 5 × 5 filter and accepts images as input.

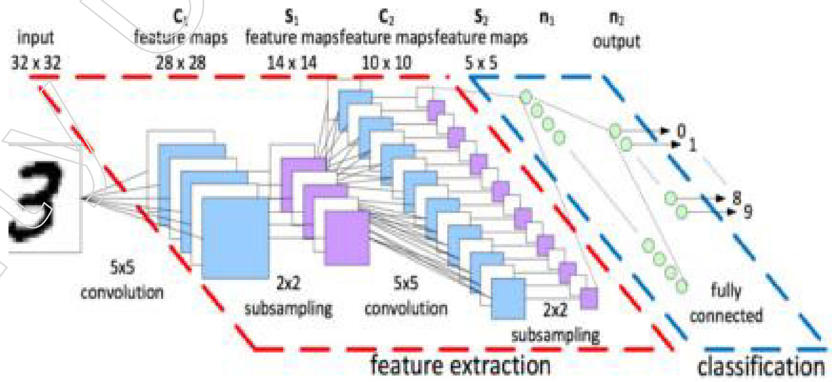


Figure: Convolution layer.

The input image, which is shown in blue in the figure below, is covered by the green filter one pixel at a time, beginning at the top left. The filter multiplies its values by the values that overlap with the image as it passes over it, then adds all of the values to produce a single value output for each overlap until the entire image has been visited.

Notes

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

1	0	1
0	1	0
1	0	1

Figure: Input and filter image

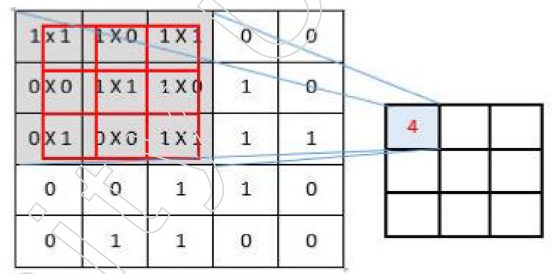
In instances where an image possesses many channels, such as the RGB (red, green and blue) model, it is observed that the kernel's depth aligns with the depth of the input image. The process of matrix multiplication is performed on the stacks Kn and In ([K1, I1], [K2, I2], [K3, I3]) as depicted in the accompanying diagram. The outcomes of this operation are subsequently merged with the bias term to produce a dense channel with a single depth.

Each every neuron inside the output matrix possesses an underlying receptive field. The initial ConvLayer is responsible for capturing low-level information, like gradient direction, edges, colour and similar characteristics. Through the incorporation of multiple layers, the architectural structure adapts to the overarching characteristics, resulting in a neural network that possesses a comprehensive comprehension of the images inside the dataset. The figures depict the sequential stages of the convolution process.

1x1	1X0	1X1	0	0
0X0	1X1	1X0	1	0
0X1	0X0	1X1	1	1
0	0	1	1	0
0	1	1	0	0

4		

Figure: Calculation of filter slides over input image.



$1 \times 1 + 1 \times 0 + 1 \times 1 + 0 \times 0 + 1 \times 1 + 1 \times 0 + 0 \times 1 + 0 \times 0 + 1 \times 1$

Figure: First step of convolution

Feature Extraction: CNN is renowned for its capacity to automatically extract attributes. The RGB picture matrix calculation is shown in the figure below. CNN typically uses padding to prevent the size of the feature maps from decreasing at each layer, which is not what is desired. The process results in two different kinds of results:

- a kind where the input's dimensions are reduced relative to the twisted feature;
- a type where the dimensionality is either kept or improved, rather than diminished. To accomplish this purpose, padding is used.

Notes

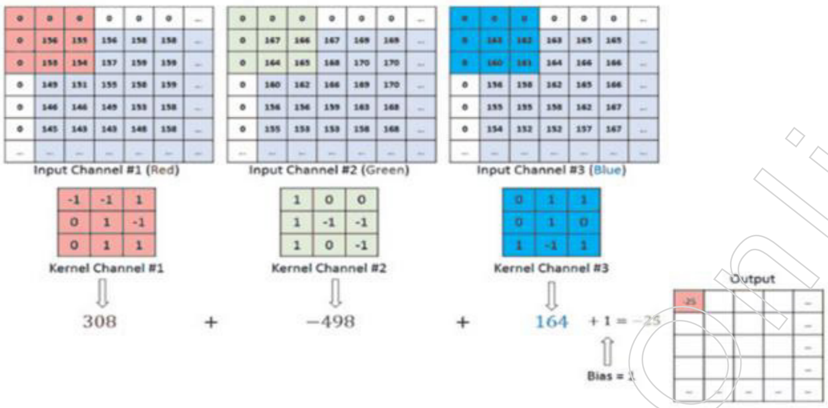


Figure :Matrix calculation.

For instance, the complex matrix is found to have dimensions of 5 x 5 x 1 when the 5 x 5 x 1 picture is reinforced into a 7 x 7 x 1 image and then applied to the 3 x 3 x 1 kernel over it, as shown in Figure below. It signifies that the input and output images share the same dimensions and padding. If the same process is carried out without padding, the output could include an image with smaller dimensions. Consequently, a 5 x 5 x 1 image will change into a 3 x 3 x 1 image.

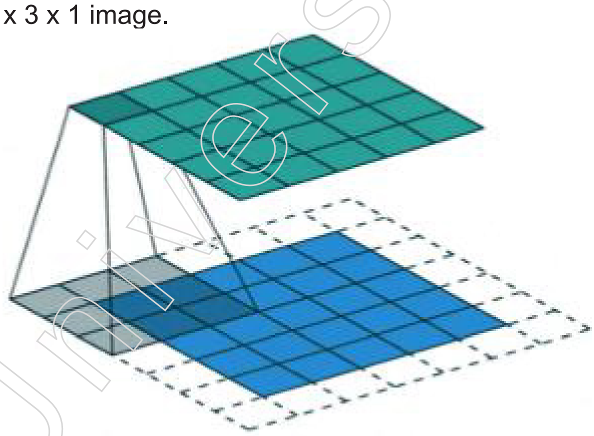


Figure: Padding.

During the forwarding pass, the kernel traverses the picture's width and height. It produces a graphic illustration of the relevant receptive region. A two-dimensional representation of the image that displays the kernel's response at each spatial position of the image is created as a result: an activation map. The kernel's size when it slips is measured in strides. Assume the input picture has the dimensions W × W × D. The output volume can be determined using the following formula if the quantity of kernels with spatial dimensions of F, stride S and padding P is unknown:

$$Wout = \frac{W - F + 2P}{S} + 1$$

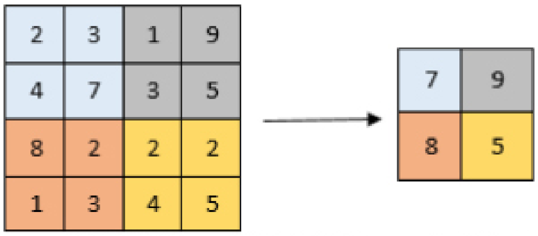
This will result in a result of size Wout× Wout × Dout.

Pooling Layer: Following the acquisition of the feature maps, a pooling (sub-sampling) layer must be added to CNN with a convolution layer. The pooling layer's function is to reduce the spatial size of the convolved feature. The amount of computing resources needed to process the data is decreased as a result of the dimensionality reduction. This helps to maintain the model's practical training and the extraction of leading characteristics that are rotational and positional invariant. Pooling shortens the

Notes

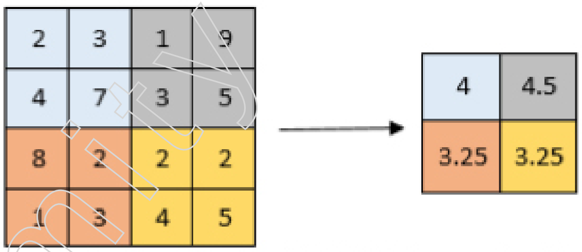
training period and avoids over-fitting. Maximum pooling and average pooling are the two types of pooling.

- Maximum Pooling: The input of the pooling layer is a tensor. In the context of maximum pooling, as depicted in the accompanying Figure, a kernel with dimensions n × n (specifically 2 × 2 in the aforementioned instance) is traversed across the matrix. Subsequently, the maximum value inside the kernel is identified and placed in the corresponding position within the output matrix.



Max Filter with 2 X 2 filter and stride 2

- **Average Pooling:** In the process of average pooling, a kernel with dimensions n × n is systematically moved across the input matrix. At each position, the average of all the values within the kernel is computed and this average is then placed in the corresponding position of the output matrix. The aforementioned process is iterated for each individual channel within the input tensor. The output tensor has been obtained as a consequence. It is imperative to bear in mind that throughout the process of pooling, the dimensions of the image are reduced in terms of height and width, while the number of channels, representing the depth, remains unchanged. The pooling layer is responsible for computing a summary statistic of the neighbouring outputs, which is then used to occasionally replace the network output.
- As a result, it helps to decrease the spatial dimension of the representation, which lowers the amount of computation and weights needed. Each slice of the representation undergoes the pooling process separately. According to Figure below, the rectangle neighborhood average, the rectangle neighborhood L2 norm and a weighted average based on the distance from the central pixel are all examples of pooling functions. Maximum pooling, which presents the neighborhood's most noteworthy output, is the most popular technique.



Average Pool with 2 X 2 filter and stride 2

Figure: Average pooling

Nonlinearity Layer (Activation Function): In CNN layers, the activation function is crucial. An additional mathematical function known as an activation function receives the filter's output. Rectified linear unit, or ReLu, is the most often used activation function in CNN feature extraction. Utilizing the activation function is mostly done to interpret the output of neural networks, such as yes or no. Depending on the activation function, the output values are mapped between -1 and 1 or 0 and 1, etc. There are two different categories for the activation functions:

Notes

- **Linear Activation Function:** This use the function F(x) = CY as the linear activation function. The output signal is proportional to the input after being multiplied by constant c (the weight of each neuron). Given that it simply provides a yes or no response and not a range of options, the linear function may be preferable to the step function.
- **Non-linear Activation:** Functions of Non-linear Activation Today's neural networks employ non-linear activation functions. In order to learn and model complex data, such as photos, videos, sounds and non-linear or high-dimensional data sets, the model must be able to construct intricate mappings between the network's inputs and outputs.

Fully Connected Layer: As seen in the figure below, a completely linked layer is nothing more than a feed-forward neural network. The very bottom layers of the network are where you'll find fully connected layers. The output layer of the last pooling or convolutional layer is flattened before being supplied as input to a fully connected layer. When the output is flattened, all of the values that were obtained after the last pooling or convolutional layer are unrolled into a vector (3D matrix).

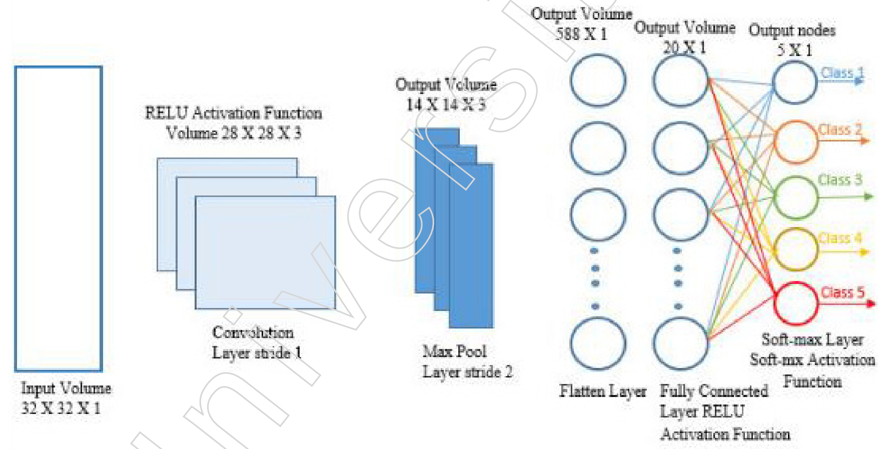


Figure: Fully connected Layer

A quick method for learning nonlinear combinations of high-level features represented by the output of the convolutional layer is to add an FC layer. The FC layer is learning a potentially nonlinear function in that area.

Architectural Evolution of CNNs: The many architectural categories of CNN variants are described in the following figure. These categories are all explained in detail in this section.

Spatial Exploitation-Based CNNs: Biases, weights, the number of layers, neurons, activation function, stride, filter size, learning rate and other factors are among the many variables in a CNN model. Convolutional procedures take into account the vicinity (locality) of input pixels, allowing for the investigation of various correlation levels using various filter sizes. Different filter sizes correspond to various granularity levels; typically, small filters collect fine-grained information, while large filters retrieve coarse-grained information. In order to improve performance, researchers started using spatial filters in the early 2000s. It was discovered that a spatial filter and network learning are related. During this time, numerous research showed that CNN might perform better on coarse and fine-grained details by modifying filters.

CNN Based on Depth: The primary concept underlying deep CNN design is that the network may successfully approximation the goal function with the aid of more

Notes

mappings (nonlinear) and more sophisticated feature hierarchies. The depth of the network has been a crucial factor in supervised training. Deep networks are more adept at representing particular function classes than shallow systems. A theorem known as “universal approximation” was first forth in 2001. It described how any function might be approximated by a single hidden layer. However, this results in an unrealistically large number of neurons and an exponentially high cost of computation. In 2011, Bengio and Delalleau proposed that deeper networks could preserve the network’s theatrical impact for less money. Bengio showed this empirically in 2013 and concluded that deep networks are computationally more efficient for complex activities. In the ILSVRC-2014 competition, VGG and Inception fared the best, supporting the idea that depth is a crucial factor in controlling a network’s capacity for learning.

CNNs with Multiple Paths: Deep CNNs are frequently effective at difficult tasks. Performance concerns, explosion problems, or gradient fading might occasionally affect them and are caused by increasing the depth rather than overfitting. An rise in test error and training error is a result of the vanishing gradient problem. For deep learning networks, the theory of cross-layer connectivity or multi-path was put out. By avoiding some intermediate levels, shortcut connections or multiple paths can connect one layer to another analytically, enabling a tailored information flow between the layers. Cross-layer connection is used to divide the network into the various components. The vanishing gradient issue is resolved by these paths by extending the gradient to lower layers.

Feature-Map Exploitation Based CNNs: Due to its ability to perform automatic feature extraction and hierarchical learning, CNN has been a popular choice for MV tasks. The choice of features has a significant impact on how well classification, segmentation and detection modules work. CNN uses a kernel, often referred to as a mask and associated weights to dynamically choose features. Additionally, numerous stages of feature extraction are carried out, allowing for a variety of features (referred to as feature maps or channels in CNN). Some of the feature maps, however, don’t perform well or at all in object discrimination. Over-fitting of the network may result from excessive feature sets producing a noise effect. This suggests that selection of feature maps can be quite important in enhancing network generalization in addition to network engineering.

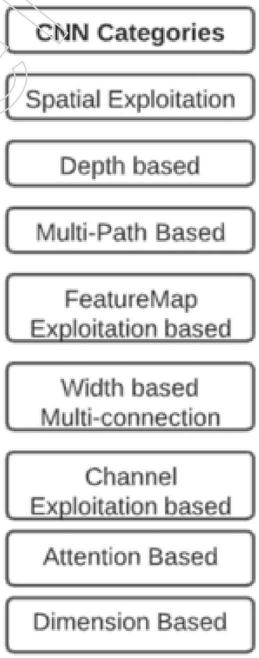


Figure: CNN variants categories.

Multi-Connection Depending on the Width: The primary focus of CNN developments from 2012 to 2015 was on maximizing the depth and effectiveness of connections for network regularization. In 2019, Kawaguchi revealed that the network’s width is also quite important.This suggests that width, in addition to depth, is crucial when creating learning philosophies. It is demonstrated that in order to maintain a universal approximation property while also gaining in depth, neural networks with ReLU activation functions must be wide enough. The failure of several layers to learn useful features is a serious problem with deep neural network topologies. Even though adding more layers (increasing depth) may enable the learning of different feature representations, this does not always improve the NN’s capacity for learning. Furthermore, if a deep network’s maximum width does not exceed its input dimension, it cannot arbitrarily approximate a class of continuous functions on

Notes

a small set. To overcome this problem, the research's emphasis shifted from deep and narrow designs to wide and thin architectures.

Exploitation-Based Feature-Map (ChannelFMap) CNNs: As a result of its ability to perform automatic feature extraction and hierarchical learning, CNN has drawn a lot of interest from people working on computer vision issues. The choice of features has a significant impact on how well classification, segmentation and detection modules work. CNN chooses features in a dynamic manner by varying the weights associated with a kernel, sometimes referred to as a mask. Additionally, CNN uses numerous feature extraction processes to mine different kinds of characteristics. Some feature maps, however, are either insignificant or have no impact on object discrimination. Massive feature sets could produce a noise effect that overfits the network. This suggests that, in addition to network engineering, the choice of feature mappings can be crucial in enhancing network generalization. In the literature, feature maps and channel terminology are sometimes used interchangeably.

CNNs That Are Based on Attention: Different levels of abstraction are crucial in defining the discrimination power of the NN. Learning involves using several abstraction hierarchies that are centered on properties important for picture localisation and recognition. The attention effect is what the visual system of the human is called. Any scene can be seen by a human by integrating quick glances at it and concentrating on context-relevant elements. This method enhances the visual structure capture process by concentrating on specific areas and taking into account several interpretations of objects at a given location. RNN and LSTM use an interpretation that is more or less similar. Attention modules are used as a progressive element in RNN and LSTM networks and new tasters are weighted according to how frequently they appeared in preceding rounds. Many researchers employ the convolutional neural network's attention principle to enhance representation and get around computational constraints. This idea of attention also helps CNN develop the ability to detect objects even against cluttered backgrounds and challenging circumstances.

Dimension-Based CNN: The traditional convolutions layer simultaneously encodes channel-wise and spatial information, but it is computationally costly. The development of separable (or depth-wise separable) convolutions, which independently encode spatial and channel-wise information using point-wise and depth-wise convolutions, increased the efficiency of conventional convolutions. The point-wise convolutions become a computational bottleneck since this factorization, despite being much more effective, exerts a heavy computational weight on them.

5.2.5 RNNs (Recurrent Neural Networks):

Recurrent Neural Networks (RNNs) are a particular sort of neural network design that are mostly employed for pattern recognition in a stream of input. Such information can include handwriting, genomes, language, or numerical time series that are frequently created in professional settings (such as stock markets or sensors). However, if those are respectively divided into a number of patches and handled as a sequence, they are likewise relevant to photographs. RNNs are used at a higher level in speech recognition, language modelling and text generation, image description and video tagging. The way information moves through the network distinguishes Recurrent Neural Networks from Feedforward Neural Networks, sometimes referred to as Multi-Layer Perceptrons (MLPs). While RNNs have cycles and broadcast data back into itself, feed forward networks just transmit data via the network. As a result, they are able to expand the capability of feed forward networks such that they take into account both the current input Xt and past inputs

Notes

X0:t1. The figure below provides a high-level visualization of this disparity. Keep in mind that one Hidden Layer block H is used to aggregate the option of having several hidden levels here. There are certainly other hidden layers that may be added to this block.

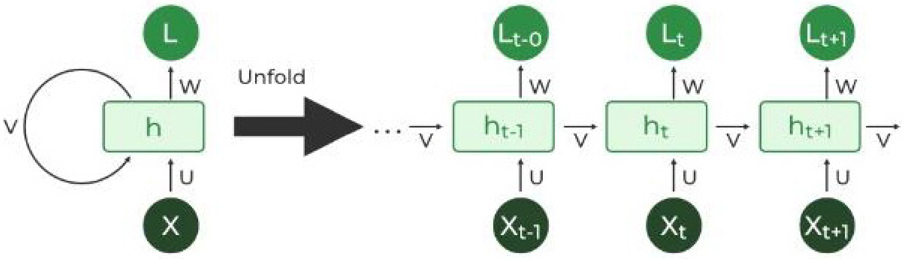


Figure: Recurrent neural network

The mathematical notation proposed in the aforementioned study can be employed to visually represent the process of transferring data from the preceding iteration to the concealed layer. In this context, the notation Ht∈R n×h and Xt∈R n×d is employed, where n represents the quantity of samples, d denotes the number of inputs per sample and h signifies the number of hidden units. In addition, we utilise a bias parameter bh∈ R 1×h, a hidden-state-to-hidden-state matrix Whh∈ R h×h and a weight matrix Wxh∈ R d×h. To facilitate the utilisation of gradients in the backpropagation process, the aforementioned data is subsequently fed into an activation function, commonly characterised by a logistic sigmoid or hyperbolic tangent (tanh) function. The combination of these notations yields the concealed variable, as described in the first equation and the resultant variable, as indicated in the second equation.

$$\mathbf{H}_t = \phi_h\left(\mathbf{X}_t\mathbf{W}_{xh} + \mathbf{H}_{t-1}\mathbf{W}_{hh} + \mathbf{b}_h\right)$$

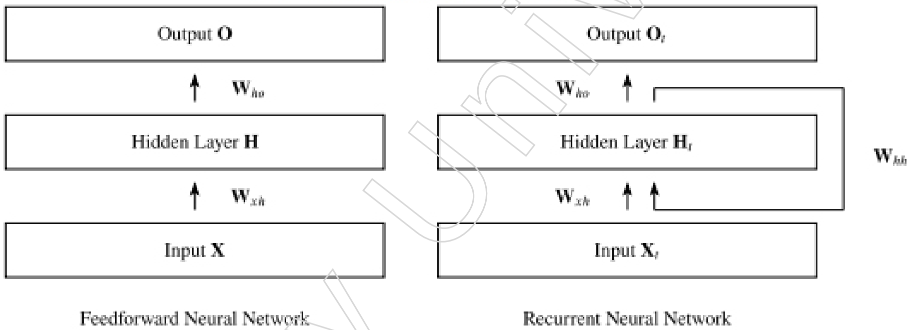


Figure:Visualisation of differences between Feedfoward NNs und Recurrent NNs

$$\mathbf{O}_t = \phi_o\left(\mathbf{H}_t\mathbf{W}_{ho} + \mathbf{b}_o\right)$$

Since Ht includes Ht−1 recursively and this process happens for each time step, the RNN contains traces of all hidden states that came before Ht−1 in addition to Ht−1. We can readily observe the distinction we previously outlined if we contrast that notation for RNNs with a syntax comparable to that for feedforward neural networks. The computation for the hidden variable is shown in the equation below and the output variable is shown in the equation that follows.

$$\mathbf{H} = \phi_h\left(\mathbf{X}\mathbf{W}_{xh} + \mathbf{b}_h\right)$$

$$\mathbf{O} = \phi_o\left(\mathbf{H}\mathbf{W}_{ho} + \mathbf{b}_o\right)$$

Architecture Of Recurrent Neural Network: The input and output architecture of RNNs is identical to that of other deep neural network architectures. But there are variations in how information moves from input to output. In RNN, the weight across

Notes

the network is constant, in contrast to Deep Neural Networks where we have separate weight matrices for each Dense network. For each input Xi, the state hidden state Hi is calculated. By use the formulas below:

$$h = \sigma(UX + Wh_{-1} + B)$$

$$Y = O(Vh + C) \text{ Hence}$$

$$Y = f\,(X, h\,, W, U, V, B, C)$$

In this case, S is the state matrix and element si represents the network's state at timestep i. W, U, V, c and b are network parameters that are shared throughout timesteps. How RNN Function:The recurrent neural network consists of a single fixed activation function unit at each time step. The hidden state of each unit is denoted as its internal state. The concealed state at a specific time step represents the network's existing knowledge of the past. At each time step, the hidden state is modified to incorporate the network's evolving comprehension of the preceding events. The provided recurrence relation is utilised for updating the hidden state.

The formula for calculating the current state:

$$h_t = f(h_{t-1}, x_t)$$

where: ht -> current state ,ht-1 -> previous state ,xt -> input state

Formula for applying Activation function(tanh):

$$h_t = \tanh\,(W_{hh}h_{t-1} + W_{xh}x_t)$$

where: whh -> weight at recurrent neuron ,wxh -> weight at input neuron.

The formula for calculating output:

$$y_t = W_{hy}h_t$$

where :Yt -> output ,Why -> weight at output layer

Backpropagation Through Time (BPTT): Because the neural network in RNN is ordered, each variable is computed one at a time in a specific order, such as first h1, then h2, then h3 and so on. Therefore, we will sequentially perform backpropagation to each of these concealed temporal stages.Figure below: L(θ)(loss function) is dependent on h3, which is dependent on h2 and W, which is dependent on h1 and W, which is dependent on h0 and W, where h0 represents a constant beginning condition.

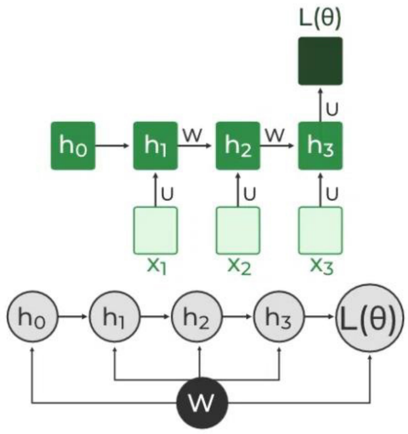


Figure:Backpropagation Through Time (BPTT) In RNN.

Notes

Types Of RNN: Based on the quantity of inputs and outputs in the network, there are four different types of RNNs.

One to One: This kind of RNN, commonly referred to as a "vanilla neural network," functions similarly like any other straightforward neural network. There is only one input and one output in this neural network.

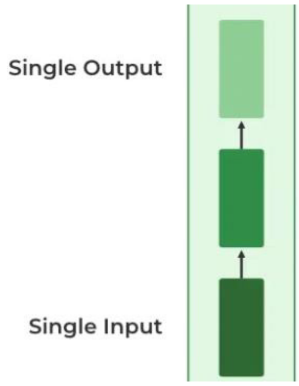
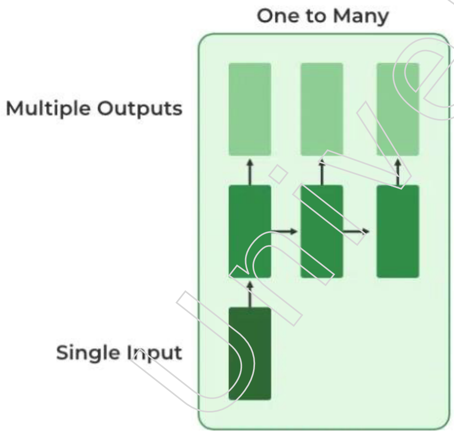


Figure:One to One RNN.

One To Many: This kind of RNN has a single input and numerous associated outputs. One of the most popular applications of this network is picture captioning, in which we anticipate a sentence with many words given an image.



Many to One:This kind of network only produces one output after receiving several inputs at various network states. Use of this kind of network is made for issues like sentimental analysis. When many words are provided as input, we merely forecast the sentiment of the sentence to be the output.

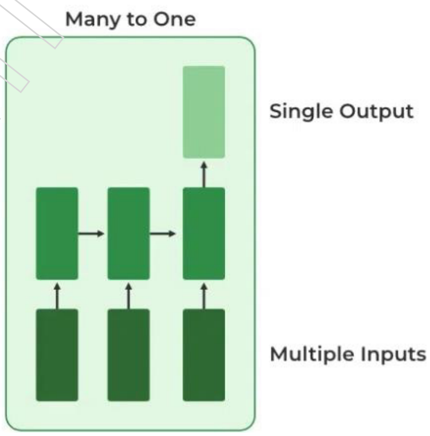


Figure:Many to One RNN

Notes

Variation Of Recurrent Neural Network (RNN):Several new, more sophisticated RNNs have been developed to address issues like disappearing gradient and explosive gradient descent. Some of these include;

- **Bidirectional Neural Network (BiNN):** An adaptation of a recurrent neural network known as a bidirectional neural network (BiNN) combines the outputs of both directions to produce the input. In cases like NLP tasks and Time-series analysis issues, where the context of the input is more crucial, BiNN is helpful.
- **Long Short-Term Memory (LSTM):** Short-Term Long-Term The read-write-forget (RWF) concept describes how memory operates. Given an input of information, the network reads and writes the data that will be most valuable in forecasting the output before forgetting the data that won't be. Three new gates are added to the RNN to accomplish this. Only the chosen information is transmitted via the network in this way.

5.3 Neural Networks: Its Types and Applications

Artificial neural networks (ANNs) and simulation neural networks (SNNs) are alternative terms used to refer to neural networks, which are a subset of machine learning and serve as the foundational component of deep learning methodologies. The structure and nomenclature of its organisation are derived from the human brain, emulating the mechanisms by which actual neurons engage in communication. Computers can utilise this approach to develop an adaptive system that facilitates continuous progress by instructing them based on errors. Artificial neural networks are employed in order to tackle challenging tasks such as document summarization or facial recognition.

Neural networks provide the capability to effectively categorise and cluster data, so augmenting the existing data management and storage infrastructure with an additional layer of grouping and categorization. These algorithms aid in the process of data categorization by organising unclassified data into groups or clusters, using similarities observed among sample inputs, with the use of a labelled dataset for training purposes.

Major Categories of Neural Networks:Some of the main types of neural networks include the following:

Classification: Neural networks frequently perform well in classification tasks, which necessitate labeled datasets for supervised learning. For instance, while recognizing visual patterns in hundreds of photos, neural networks can assign labels fast and consistently. Through practice, they develop the ability to solve challenging, confusing challenges. The brain network develops an innate ability to recognize the most important factors. As a result, the data scientist is not obliged to offer attributes that would allow cats and dogs to be distinguished from one another.

Sequence learning: Data sequences are used as input or output in the machine learning category known as "sequence learning." Sequential learning can be done with text streams, audio files, video clips, measurements and more.

Function approximation: Function approximation is a strategy that uses previous or current observations from the domain to approximate an unknown underlying function. Artificial neural networks can learn to approximate a function.

5.3.1 Perceptron:

The term "perceptron" is extensively employed in the fields of machine learning and artificial intelligence, enjoying widespread usage among practitioners. Acquiring knowledge in the fields of machine learning and deep learning, which encompass a

Notes

collection of weights, input values or scores and a threshold, constitutes the initial stage of this undertaking. The perceptron is a fundamental element within an artificial neural network. The Perceptron was originally developed by Frank Rosenblatt during the mid-20th century with the purpose of doing certain computations to discern the capabilities of input data or commercial intelligence.

A linear machine learning algorithm called the perceptron is utilized for supervised learning for different binary classifiers. With the help of this algorithm, neurons can learn new elements and process them one at a time while preparing. In-depth knowledge of the perceptron will be covered in this tutorial's section on "Perceptron in Machine Learning," along with a brief overview of its fundamental operations. Let's begin with a brief overview of the perceptron.

What is the Perceptron model in Machine Learning: For the supervised learning of various binary classification tasks, the perceptron machine learning algorithm is used. Moreover, a perceptron can be classified as an artificial neuron or unit within a neural network, which contributes to the field of business intelligence by facilitating the identification and analysis of particular input data calculations. The perceptron model is widely regarded as one of the most effective and straightforward types of artificial neural networks. Nevertheless, the system employs binary classifiers within a supervised learning framework. The structure of this system can be conceptualised as a single-layer neural network, consisting of four primary components: input values, weights and bias, net sum and an activation function.

Basic Components of Perceptron: The perceptron model, which is a binary classifier consisting of three key components, was developed by Mr. Frank Rosenblatt. The following items are enumerated below:

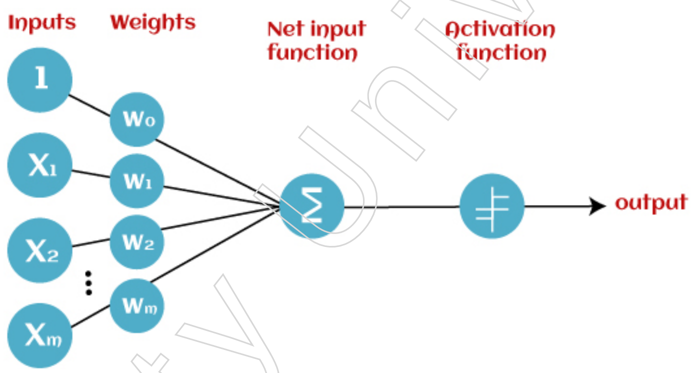


Figure: Basic Components of Perceptron

- **Input Nodes or Input Layer:** The primary component of the Perceptron system responsible for receiving the initial data for further processing is the aforementioned component. Each input node has a numerical value that is authentic.
- **Wight and Bias:** The weight parameter quantifies the degree of connectivity between the units. This particular attribute holds significant importance for components of the Perceptron. The influence of the input neuron on the output is directly proportional to its weight. Moreover, the point of intersection in a linear equation might be conceptualised as a form of inherent prejudice.
- **Activation Function:** The aforementioned aspects are of paramount importance and serve as determining factors in the firing or non-firing of a neuron. The activation function can be conceptualised primarily as a step function.Various activation mechanisms:
- Sign function, Step function andSigmoid function

Notes

The data scientist creates the desired outputs by using the activation function to make a judgment call based on multiple issue statements. By examining whether the learning process is slow or has disappearing or exploding gradients, it is possible to distinguish between different activation functions (such as Sign, Step and Sigmoid) in perceptron models.

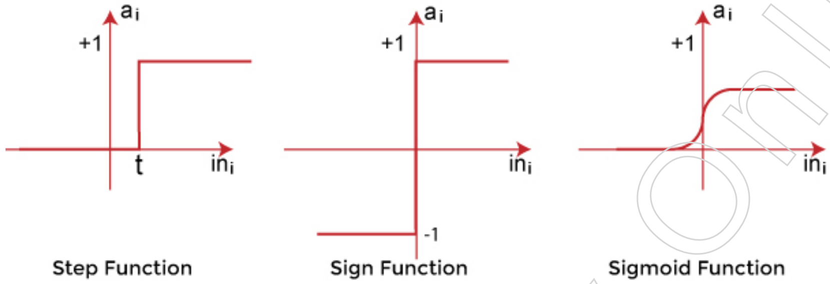


Figure : Activation Function structure

How does Perceptron work:

Perceptrons are single-layer neural networks that are used in machine learning. The composition of these entities consists of four primary components: input values (Input nodes), weights and bias, net sum and an activation function. To compute the weighted sum, the perceptron model performs a multiplication operation between each input value and its corresponding weight. The weighted sum is further processed using the activation function 'f' to obtain the desired output. The activation function, commonly known as the step function, is denoted by the sign 'f'.

The activation function, sometimes referred to as the step function, plays a critical role in facilitating the transfer of output values within the specified range of (0,1) or (-1,1). It is important to note that the strength of a node can be assessed based on the magnitude of its input weight. In a similar vein, the inclusion of a bias value in an input facilitates the manipulation of the curve of the activation function.

Perceptron model works in two important steps as follows:

Step-1: To obtain the weighted total, multiply all input values by the relevant weight values in the first step, then add the results. The weighted sum can be determined mathematically in the manner shown below:

∑wi*xi = x1*w1 + x2*w2 +…wn*xn This weighted sum should be supplemented with a unique term called bias 'b' to enhance the model's performance.

$$\sum wi*xi + b$$

Step-2: The weighted sum discussed above is combined with an activation function in the second phase to provide output that can either be binary or continuous.

$$Y = f(\sum wi*xi + b)$$

Types of Perceptron Models: Perceptron models are classified into two categories based on the layers. These are listed below:

- Single-layer Perceptron Model
- Multi-layer Perceptron model

Single Layer Perceptron Model: One of the simplest types of artificial neural networks (ANN) is this one. A threshold transfer function and a feed-forward network are both included in a single-layered perceptron model. The single-layer perceptron model's primary goal is to examine things that can be linearly separated into binary categories.

- The weight parameters of a single layer perceptron model are initially allocated random values as the model's processes do not rely on recorded data. Moreover, it calculates the sum of all the inputs in terms of weight. The model exhibits activity by generating an output of +1 when the cumulative sum of all inputs over a specified threshold.
- The model's performance is considered satisfactory if the output matches the predetermined or threshold value, while the weight requirement remains constant. Nevertheless, this model exhibits certain discrepancies that manifest when varying weight input values are employed. To ensure accurate output and minimise errors, it is necessary to make several modifications to the input weights.
- "Single-layerperceptrons can only learn linearly separable patterns."

Multi-Layered Perceptron Model: The multi-layer perceptron model possesses a greater number of hidden layers in comparison to the single-layer perceptron model, while maintaining a same model structure.The backpropagation algorithm, also referred to as the multi-layer perceptron model, operates in a two-step process as outlined below:

- **Forward Stage:** During the forward propagation stage, the activation functions are initiated at the input layer and propagate through the subsequent layers until reaching the output layer.
- **Backward Stage:** During the backward step, the weight and bias parameters are modified in order to align with the requirements of the model. At this juncture, the disparity between the desired output and the observed output originated at the output layer and culminated at the input layer.

Consequently, a comparative analysis has been conducted between a multi-layered perceptron model and other artificial neural networks featuring different layers, whereby the activation function is non-linear. This comparison is made in relation to a single layer perceptron model.

In the context of deployment, alternative activation functions, such as sigmoid, TanH andReLU, can be employed instead of linear activation functions. The multi-layer perceptron model possesses the ability to effectively process both linear and non-linear patterns, hence exhibiting enhanced processing capability. Additionally, it may use logic gates like AND, OR, XOR, NAND, NOT, XNOR and NOR.

Applications:

- **Data Compression:** Data compression is the process of encoding, rearranging, or otherwise changing data to make it smaller. It entails re-encoding data with fewer bits than the original representation in its most basic form.
- **Streaming Encoding:** Faster training is achieved by using an encoding technique that whitens the real-valued input data sent to the first hidden units of a fully connected neural network.

Perceptron Function: The output of a perceptron function, f(x), is obtained by multiplying the input, x, by the learned weight coefficient, w. We can formulate it mathematically as follows: If w.x+b>0, f(x)=1; otherwise, f(x)=0; where 'w' denotes a vector of real-valued weights, 'b' denotes the bias and 'x' is a vector of input x values.

Characteristics of Perceptron: The following traits apply to the perceptron model:

- A machine learning algorithm called perceptron is used to learn binary classifiers under supervision. The weight coefficient is automatically learned in a perceptron.
- Weights are first multiplied by input features to determine whether to fire the neuron or not.

Notes

- To determine if the weight function is larger than zero, the activation function applies a step rule.
- In order to distinguish between the two linearly separable classes, +1 and -1, the linear decision boundary is drawn.
- It must have an output signal if the total of all input values is greater than the threshold value; otherwise, no output will be displayed.

Limitations of Perceptron Model: A perceptron model has the following drawbacks:

- The hard limit transfer function of a perceptron limits the output to a binary number (0 or 1) alone.
- Only sets of input vectors that can be linearly separated can be classified using perceptrons. Non-linear input vectors are difficult to correctly categorize.

Future of Perceptron: The Perceptron model has a very promising and important future since it aids in the interpretation of data by creating intuitive patterns and using them afterwards. The future of perceptron technology will continue to support and facilitate analytical behavior in machines that will, in turn, increase the efficiency of computers. Machine learning is a rapidly expanding field of Artificial Intelligence that is constantly evolving and in the developing stage. With the aid of artificial neurons, the perceptron model is constantly improving and solving complicated issues effectively.

5.3.2 Feed Forward Neural Network:

This topic pertains to feedforward neural networks, which are sometimes known as deep feedforward networks or multi-layer perceptrons. For example, these networks form the basis for convolutional and recurrent neural networks, which are extensively employed in computer vision applications. We will endeavour to render the significant principles easily comprehensible and remember, while avoiding excessive mathematical intricacies. Deep learning technologies are widely employed in several domains such as mobile applications, machine translation and search engines. The mechanism of action involves stimulating the brain to discern and establish patterns from diverse sets of input. This remarkable technology significantly relies on the utilisation of feedforward neural networks due to their efficacy in assisting programmers with tasks such as non-linear regression, function approximation, as well as pattern recognition and classification.

A feedforward neural network is a class of artificial neural networks characterised by the absence of cyclic connections among the nodes. The reverse of a feed forward neural network is a recurrent neural network, wherein specific routes are cyclically traversed. The feed forward model is considered the most elementary form of neural network due to its unidirectional processing of input. While the transmission of data may occur through multiple subterranean nodes, it consistently progresses in a unidirectional manner, without any retrograde movement.

How does a Feed Forward Neural Network work:

The single layer perceptron is a prevalent illustration of a feed-forward neural network in its fundamental form. Multiple inputs are introduced into the layer inside this model and subsequently multiplied by the corresponding weights. The summation of the weighted input values yields a cumulative total. The resulting number is commonly 1 andin the event that the cumulative aggregate of the values falls below the specified threshold, the output value is assigned as -1. The threshold is commonly established at a value of zero. The single layer perceptron plays a vital role as a feed forward neural network model in classification problems. Single-layerperceptrons may incorporate certain elements of artificial intelligence.

Notes

The neural network has the capability to evaluate the outputs of its nodes in relation to the desired values through a mechanism referred to as the delta rule. This property facilitates the network in adjusting its weights in order to generate output values that exhibit enhanced accuracy. The process of learning and training leads to the phenomenon known as gradient decline. The process of updating weights in multi-layered perceptrons, commonly referred to as back-propagation, exhibits a high degree of similarity. Under these conditions, the hidden layers of the network are modified based on the output values produced by the top layer.

Feed forward Neural Network's Layers: The following are the components of a feedforward neural network:

- The layer responsible for receiving information consists of neurons that are situated within it. The subsequent tier obtains the information subsequent to that. The quantity of variables inside the dataset is equivalent to the overall count of neurons present in the input layer.
- The neural network architecture includes a hidden layer, which functions as an intermediary layer positioned between the input and output layers. A multitude of neurons undergo modifications to the inputs inside this particular layer. Subsequently, communication is established with the output layer.
- The ultimate layer, referred to as the output layer, is determined by the architectural design of the model. Furthermore, given your understanding of the intended outcome, the output layer corresponds to the anticipated characteristic.
- The application of weights on neurons serves to quantify the strength of the connections between them. A weight possesses a numerical value that is within the range of 0 to 1.

Applications of Feed Forward Neural Networks:

Despite their simplicity, Feed Forward Neural Networks offer advantages in certain machine learning applications because of their streamlined architecture. Utilising a conservative mediator for moderation, an approach could involve deploying multiple feed-forward neural networks in isolation from each other. Analogous to the human brain, this method employs a multitude of individual neurons to effectively process and understand tasks of greater complexity. The integration of the individual findings from each network can be performed in order to get a consolidated and cohesive output.

Neural networks find extensive use across diverse domains. The area units for several of them are indicated as follows:

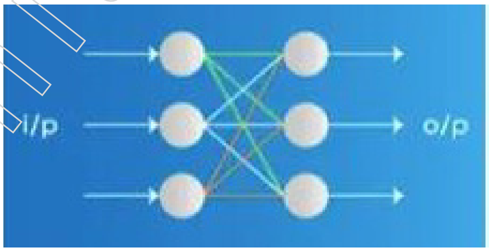


Figure: Feed Forward Neural Network

- Pattern Recognition: Pattern recognition refers to the utilisation of a machine learning algorithm for the purpose of identifying and discerning patterns. Pattern recognition refers to the process of categorising data by utilising either preexisting knowledge or statistical information obtained from patterns and their representations.

Notes

- Computer Vision: Artificial intelligence (AI) encompasses a specialised area known as computer vision, which empowers computers and systems to extract relevant information from digital photos, videos and other visual inputs. This extracted data serves as a basis for subsequent actions or recommendations made by the AI system.
- Physiological feedforward system: The concept of feedforward management is shown by the typical preemptive regulation of heart rate by the central autonomic system before engaging in physical activity.
- Gene regulation and feedforward:A recurrent motif may be observed in these widely recognised networks and empirical evidence has demonstrated that this motif serves as a feedforward mechanism for detecting enduring alterations in the atmosphere.
- Automating and managing machines
- Parallel feedforward compensation with derivative: This is a more recent method of transforming the minimum part of an open-loop transfer system from the non-minimum part.

Advantages:

- With a small middleman to ensure moderation, a number of Feedforward networks can function independently.
- The streamlined architecture of feed forward neural networks can improve machine learning.
- With a moderated intermediary, multiple networks in feed-forward networks operate independently.
- Several neurons are required in the network for complex tasks.
- In contrast to perceptrons and sigmoid neurons, which are otherwise complex, neural networks can handle and process nonlinear data with ease.
- Decision boundaries are a challenging problem that a neural network handles.
- The neural network architecture can change based on the data. For instance, recurrent neural networks (RNNs) excel at processing text and voice, while convolutional neural networks (CNNs) excel at processing images.
- GPUs are necessary for neural networks to handle large datasets for high computational and hardware performance.

Disadvantages:

- Insufficient for deep learning.
- to optimize more variables.
- losing knowledge of the neighborhood.
- Translation invariance isn't the issue.

5.3.3 Multilayer Perceptron:

A multilayer perceptron consists of an input layer, an output layer and one or more hidden layers, with each layer comprising several interconnected neurons. In contrast to the Perceptron, which requires neurons to have an activation function that enforces a threshold, such as ReLU or sigmoid, the neurons in a Multilayer Perceptron have the flexibility to apply any arbitrary activation function.

The multilayer perceptron is considered the most prevalent and extensively utilised type of neural network. In the vast majority of instances, the transmission of signals occurs from the input to the output within the network. The absence of a loop and the lack of influence of a neuron's output on itself are notable characteristics. The architectural design in question is commonly referred to as "feedforward." The phrase "hidden" pertains to strata that are not inherently connected to their immediate environment. There is ongoing discussion in the literature over the classification of the input layer as an independent layer inside the network, as its primary function is to transmit input signals to the higher layers without engaging in any input processing. The inputs are organised into clusters inside the input layer. However, for the sake of our analysis, we will focus just on the layers composed of individual neurons. Moreover, there are feedback networks that possess the ability to send impulses bidirectionally due to the presence of reaction links inside the network.

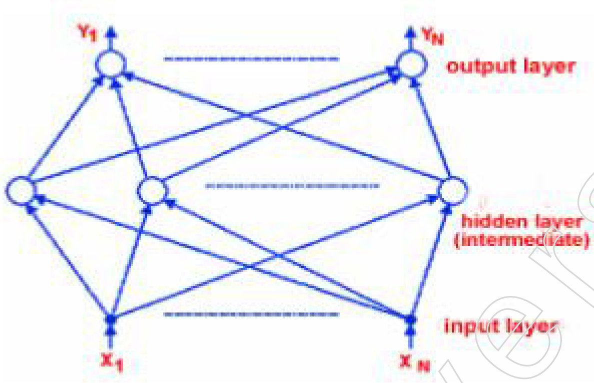


Figure: The multilayer perceptron.

These networks possess significant influence and exhibit a high degree of complexity. The network exhibits dynamic behaviour, continuously adapting until it finds a state of equilibrium. Furthermore, with every alteration in input, the network actively seeks a new state of equilibrium. The necessity to enhance the intricacy of choice regions prompted the incorporation of many layers. A perceptron consisting of only one layer and one input generates decision regions that are semi-planar.

The network's output has the ability to estimate convex decision regions, which are formed by the intersection of semi-planes generated by the neurons. This is achieved by introducing an additional layer where each neuron acts as a standard perceptron for the outputs of the neurons in the preceding layer. An alternate option that can be considered is a three-layer perceptron, as depicted in the accompanying illustration.

It was shown that, if the activation functions of neurons are linear, multilayer networks do not offer an improvement in processing power when compared to networks with a single layer since a linear function of a linear function is likewise a linear function. Non-linear activation functions are precisely where the multilayer perceptron's power lies. Except for polynomial functions, almost every non-linear function can be utilized for this. Currently, the single-pole (or logistic) sigmoid, as shown in Figure below, is the function that is most frequently utilized.

$$f(s) = \frac{1}{1+e^{-s}}.$$

Notes

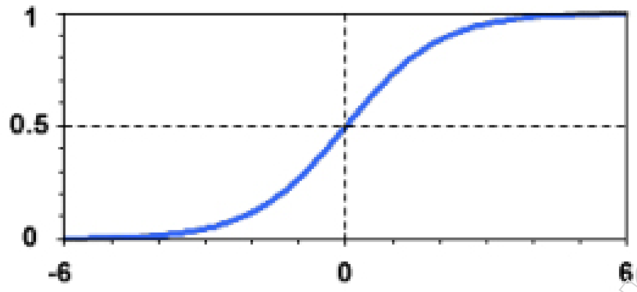


Figure : Sigmoid single-pole activation function.

The bipolar sigmoid function (the hyperbolic tangent), for a=2, is depicted in Figure below:

$$f(s) = \frac{1 - e^{-a \cdot s}}{1 + e^{-a \cdot s}}.$$

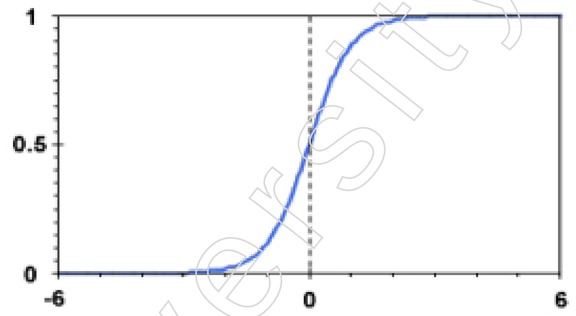


Figure : Sigmoid single-pole activation function.

This supports the multilayer perceptron's ability to serve as a universal approximator. Additionally, it was shown that neural networks can calculate specific polynomial expressions by applying the Stone-Weierstrass theorem to them: if there are two networks that calculate precisely two functions, f1 and f2, then there is a larger network that calculates precisely a polynomial expression of f1 and f2.

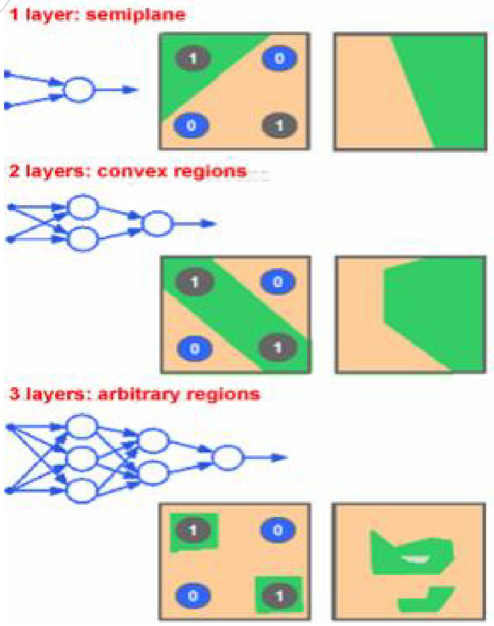


Figure: Decision regions of multilayer perceptrons.

Notes

The most popular and well-known type of neural network is a multi-perceptron, which is made up of trained units like those in the figure below. Each of these units creates a weighted input total to which a constant is added. Following that, this amount is processed via a nonlinear function that is frequently referred to as the activation function. The majority of units are interconnected in a “feed forward” way, i.e. interconnections that form an aloop, as seen in the following figure.

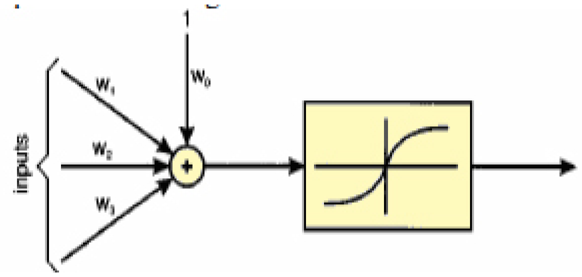


Figure: Example network “feed forward”. Each circle represents a unit of the type shown in Figure . Each connection between units is a share. Each unit also has an entry in the diagonal are not shown.

Working of MultiLayer Perceptron Neural Network:

- The characteristic of the dataset is represented by the input node.
- Each input node sends the hidden layer’s hidden layer the vector input value.
- Each edge in the hidden layer has a weight that is multiplied by the input variable. The sum of all the production values from the concealed nodes is calculated. to produce the result.
- In the buried layer, the active nodes are recognized using the activation function.
- The output layer receives the output.
- Determine the discrepancy between output as planned and output as achieved at the output layer.
- Following the computation of the anticipated output, the model employs backpropagation.

Advantages of MultiLayer Perceptron Neural Network:

- Non-linear issues can be solved with ease by MultiLayer Perceptron Neural Networks.
- It is capable of handling big datasets and difficult issues.
- This paradigm is used by developers to address the fitness issue with neural networks.
- It uses backpropagation to increase accuracy and lower prediction error.
- The Multilayer Perceptron Neural Network accurately predicts the outcome after model training.

Disadvantages of MultiLayer Perceptron Neural Network:

- This neural network uses a lot of computation, which occasionally raises the model’s overall cost.
- Only when the model has received perfect training will it function well.
- Due of the close connections in this model, there are more parameters and nodes are redundant.

Notes

5.3.4 Convolutional Neural Network:

Convnets, also known as convolutional neural networks (CNNs), are a particular type of feedforward neural network. In that they are composed of neurons with trainable weights and biases, they are quite similar to the neural networks discussed above. The fundamental distinction is that we can encode certain features in the CNN design since it implicitly assumes that the input is image-like. Convolutions in particular capture translation invariance (i.e., filters are position independent). As a result, the forward function is more effective, there are much less parameters, the network is easier to improve and the dependence on the amount of the data is reduced.

Unlike conventional neural networks, CNNs' layers feature neurons arranged in a few different dimensions, including channels, width, height and number of filters in the most basic 2D example. Similar to an MLP, a convolution neural network is made up of a series of layers, each of which modifies the activations or outputs of the layer before it using a different differentiable function. The convolution layer, pooling layer and fully connected layers are the most typical building blocks you will find in most CNN architectures. There are other layers used in CNNs as well and they will be covered in following sections. These layers essentially function as dimensionality reduction, feature extractors and classification layers, respectively. The whole convolutional layer of a CNN is created by stacking these CNN layers.

General Model of Convolution Neural Network:

General Model:The conventional artificial neural network (ANN) model typically consists of multiple hidden layers, in addition to a single input layer and a single output layer. A designated neuron receives an input vector X and produces an output vector Y by the application of a function F, as denoted by the general equation presented herein.

$$F(X,W) = Y$$

where W stands for the weight vector, which symbolizes how strongly neurons in two adjacent layers are connected. The weight vector that was created can now be utilized to classify images. The classification of images based on pixels has been extensively studied in the literature. Contextual information, such as the image's shape, gives better results or outperforms, nonetheless. CNN is a model that is attracting interest due to its ability to classify data based on context. The graphic below describes the CNN model in its entirety. Convolution layer (a), pooling layer (b), activation function (c) and fully connected layer (d) make up a standard CNN model. Below is an illustration of each component's functionality:

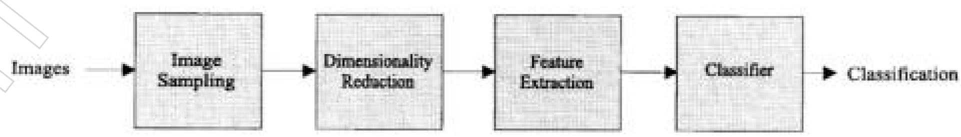


Figure: Elementary constituents of CNN

Convolution Layer: The input layer of the system receives an image that requires classification and the predicted class label is generated by utilising features extracted from the image. The receptive field refers to the specific connection established between an individual neuron in the subsequent layer and a subset of neurons in the preceding layer. The retrieval of local details from the input image is accomplished by utilising the concept of receptive field.

The formation of a weight vector is a result of the connection between a neuron's receptive field and a specific place in the preceding layer, which in turn relates to the neurons in the subsequent layer. Given that the neurons on a two-dimensional plane possess identical weights, it becomes feasible to discern similar characteristics that manifest across multiple locations within the input data. The representation of this concept is depicted in the figure provided below.

The generation of the feature map is achieved by systematically moving the weight vector, also known as the filter or kernel, along the input vector. The convolution operation refers to the procedure of horizontally and vertically moving the filter. The aforementioned methodology involves the generation of N filters and N feature maps through the extraction of N distinct features from the input image.

These features are subsequently positioned on a singular layer, thereby representing each unique feature. The local receptive field phenomenon results in a significant reduction in the number of trainable parameters. The output value aij in the subsequent layer for a given point (i,j) is computed using the convolution process, as described by the formula provided below:

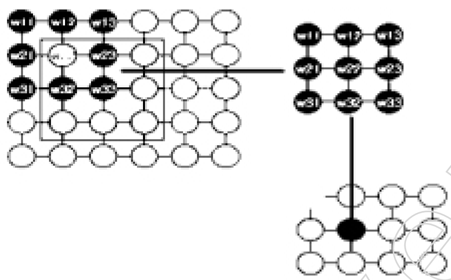


Figure: Receptive field of particular neuron in the next layer.

$$a_{ij} = \sigma\left((W * X)_{ij} + b\right)$$

In this context, X is the input received by the layer, W denotes a filter or kernel that is applied to the input, b represents the bias term, * signifies the convolution operation andsymbolises the nonlinearity incorporated into the network.

Pooling Layer: Once a feature has been found, the exact positioning of this feature becomes less significant. As a result, the pooling or sub-sampling layer is typically positioned following the convolution layer. The utilisation of the pooling technique offers notable benefits, such as the introduction of translation invariance and a substantial reduction in the number of trainable parameters. As seen in the visual representation provided, a certain window is selected for the purpose of conducting the pooling procedure. Subsequently, the input elements encompassed within said window are subsequently sent through the pooling function.

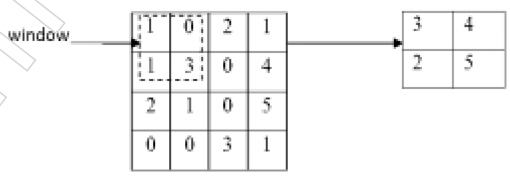


Figure: Pooling operation performed by choosing a 2 x 2 window.

The pooling function generates an additional output vector. There exist a limited number of pooling techniques, including average pooling and max-pooling. Among them, max-pooling is the most commonly employed approach, known for its significant reduction in map size. The error does not propagate to the winning unit during computation mistakes, as it does not partake in the forward flow.

Notes

Fully Connected Layer: The fully connected network in traditional models is analogous to the fully connected layer. The fully connected layer receives the output of the first phase, which comprises repetitive convolution and pooling and computes the dot product of the weight vector and the input vector to produce the final output. Gradient descent, sometimes referred to as batch mode learning or the offline approach, lowers the cost function by estimating the cost throughout the full training dataset. It changes the parameters only once every epoch, or complete traversal of the training dataset. Although it produces global minima, the size of the training dataset has a significant impact on how long it takes to train the network. Stochastic gradient descent was used to replace this method of decreasing the cost function.

Activation Function: The utilisation of the sigmoid activation function in conventional machine learning techniques is extensively documented in the academic literature. The utilisation of Rectified Linear Unit (ReLU) has demonstrated its superiority over its predecessor in terms of creating non-linearity, primarily due to two key factors. To begin with, the computation of the partial derivative of the Rectified Linear Unit (ReLU) function is straightforward. Furthermore, it has been observed that saturating non-linearities, such as the sigmoid function represented by ‡−□−, exhibit poorer computational speed compared to non-saturating non-linearities, such as the Rectified Linear Unit (ReLU), even when considering the training time.

Third, ReLU prevents gradients from going away. However, a big gradient that is flowing through the network reduces ReLU efficiency and an update in weight prevents activation of the neuron, which results in the Dying ReLU problem, a significant difficulty that is frequently experienced. Leaky ReLU, where is a tiny constant, can be used to solve this problem. If x>0, the function activates as f(x)=x and if x<0, the function activates as x.

Architectures Of Convolution Neural Network: In CNN, numerous architectures have been created and put into use. Below are succinct descriptions of those architectures:

LeNet Architecture: The ability of multi-layer networks to learn from extremely complicated and high-dimensional input makes them suited for picture recognition tasks. The following paragraph provides a summary of the LeNet architecture, which was proposed in 1998 and leverages datasets. The figure below shows the LeNet5 architecture. It comprises eight layers, five of which are convolutional and three of which are fully linked. A plane has 25 inputs per unit. Units in the first hidden layer receive input from the 5 × 5 area, a small portion of the input image that is transferred to the first hidden layer. Receptive field of the unit refers to this particular section of the input image. In a plane, every unit has the same weight vector. The unit's output is kept in the same spot on the feature map.

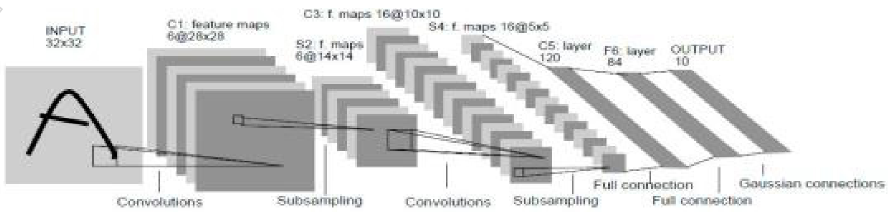


Figure: Architecture of LeNet5, a CNN where each box represents a different feature map.

The adjacent units in the feature map are the outcome of the adjacent units in the preceding layer. Contiguous receptive fields overlap as a result of this. Convolution layer is the first layer and it is made up of neurons that produce sigmoid activation when applied to the weighted sum. As seen in figure, if a 5 × 5 area is selected as an input and a horizontal shift on the area is carried out, it will result in the overlap of four rows and

Notes

five columns while computing contiguous units in the feature map. Different feature maps are produced as a result of applying various weight vectors to the same input image.

The produced feature maps can be used to extract various features. The fundamental characteristic of CNN is that a small change in the input has no impact on the feature map. Since it is not important to have a precise position for a feature in an image, subsampling is done to lower the precision value. Sub-sampling has been shown in the second layer of the previous graphic, as seen. The amount of feature maps obtained through subsampling is equal to that of feature maps obtained through convolution. Here, the average of the four inputs has been computed for the sub-sampling layer 2 × 2 area, multiplied by the trainable coefficient, added the trainable bias and then sent to the sigmoid function. As the spatial resolution is reduced layer by layer, an increase in the number of feature maps may be seen. The back propagation approach is used to carry out the learning.

AlexNet Architecture:This piece provides a succinct description of the AlexNet architecture, a modified version of LeNet that has been proposed. For the purpose of classifying 1.2 million high resolution photos into 1000 different classes, it is composed of three completely connected layers and five convolution layers. To train the network more quickly, non-saturating neurons and effective GPU implementation are used. As a result, a huge network is needed to enable the network to classify objects from millions of photos, which may ultimately result in a high demand for training a very large number of weights and the overfitting issue.

This issue was solved by using the dropout method. In this method, the neurons that have a probability of 0.5 are damped and do not participate in forward or backward propagation. Overfitting is significantly reduced since the neurons that depend on these damped neurons are forced to learn the most robust features entirely on their own. The dropout method doubles the amount of iterations needed to converge. It takes five to six days to train the network using two GTX 580 3GB GPUs. The main characteristics of this design included the addition of ReLU non-linearity in CNN, which caused the convergence rate to rise quickly.

GoogleNet Architecture: oogleNet is a popular concept that was proposed. After winning the ILSVRC14 competition, it became enlightened. The main objective was to create a model with a smaller budget that could consume less memory, power and trainable parameters. The number of trainable parameters employed in the network was dramatically decreased by the model. The following is a general description of the architecture. In essence, it employs 12 million fewer parameters than the model put forward by. The architecture attempted to create a network that could more accurately identify the items in an image.

This might be accomplished by expanding the network's size, which would increase the number of layers, but a key negative of this idea was that doing so would increase the number of parameters that would need to be trained, which would create the issue of overfitting. Another significant drawback is that as the number of filters increases, so does the computation, which raises overhead. Implementing a sparse matrix was the suggested approach. In order to create the best network topology, the highly correlated units join to form a cluster in the previous layer and send input to the next layer. Even though the computations are sped up by 100 times, the overhead of cache misses still exists when using the non-uniform sparse matrix. Using highly optimized numerical libraries to achieve faster computations is also ineffective. As a result, the state of the art relies on uniform sparse matrices.

Notes

Advantages of Convolutional Neural Network (CNN):

- One of the primary characteristics of Convolutional Neural Networks (CNNs) is their ability to efficiently process images. This phenomenon can be attributed to the utilisation of a technique known as convolution, wherein a filter is applied to an image to extract relevant features for the specific task at hand. Convolutional neural networks (CNNs) are capable of processing information in a more efficient and expedient manner compared to other methods. This is achieved by the reduction of data volume that necessitates processing.
- One advantage of Convolutional Neural Networks (CNNs) is their ability to achieve high accuracy rates. This phenomenon can be attributed to the practise of analysing extensive datasets, which enables individuals to develop the ability to identify complex patterns inside photographs. This characteristic renders them very suitable for tasks such as facial recognition or object identification, as they may undergo training to achieve precise recognition of certain items or properties.
- Moreover, Convolutional Neural Networks (CNNs) exhibit robustness to noise, hence facilitating their ability to identify patterns in images that have been compromised or distorted. This phenomenon can be attributed to the process of extracting features from images through the utilisation of several layers of filters, hence endowing them with enhanced resistance to noise compared to alternative techniques.
- Transfer learning is a notable advantage of convolutional neural networks (CNNs). This capability allows CNNs to be initially trained for a specific task and subsequently utilised for another task without requiring much additional guidance or instruction. This phenomenon can be attributed to the fact that convolutional neural networks (CNNs) possess the ability to extract attributes that are often sufficiently generalizable to be applied across a wide range of tasks, hence rendering them a versatile tool for several applications.
- Automated feature extraction – The feature extraction process is automated by CNNs, allowing them to learn to recognize patterns in images without the need for manual feature engineering. Because the CNN can be trained to recognize the relevant features, they are perfect for jobs where the features that are crucial to the task are not known in advance.

Disadvantages of Convolutional Neural Network (CNN):

- One of the primary limitations of Convolutional Neural Networks (CNNs) is their significant computational requirements. The reason for this phenomenon is that Convolutional Neural Networks (CNNs) may have numerous layers and parameters, resulting in a substantial requirement for computational resources and memory during both training and operation. Due to this factor, its suitability for use in certain applications with limited resources may be questionable.
- One challenge that arises when working with tiny datasets is that Convolutional Neural Networks (CNNs) require large datasets in order to achieve high levels of accuracy. This phenomenon can be attributed to the extensive examination of various instances of patterns in images prior to acquiring the ability to identify them. The phenomenon of overfitting can occur in the CNN model when the size of the dataset is insufficient, leading to an excessive specialisation of the model to the training data and subsequently poor performance when presented with new, unseen data.
- In order to achieve high accuracy rates, Convolutional Neural Networks (CNNs)

Notes

require the use of large datasets. This phenomenon can be attributed to the extensive analysis of various patterns in images that individuals undertake prior to acquiring the ability to identify them. The phenomenon of overfitting can occur in the CNN model when the size of the dataset is insufficient, leading to the model being excessively specialised to the training data and therefore exhibiting poor performance when presented with new, unseen data. One further limitation of Convolutional Neural Networks (CNNs) is their lack of interpretability. Therefore, understanding the decision-making process employed by CNN is a significant challenge. This issue might provide a challenge in applications where it is crucial to comprehend the underlying reasoning behind a specific decision.

- **Vulnerability to adversarial attacks –** Adversarial assaults, which entail purposefully modifying the input data to trick the CNN into making false judgments, are another danger to CNNs. In applications like driverless vehicles, where safety is a top priority, this can be a major issue.
- **Limited ability to generalize –** Finally, CNNs can only a limited extent generalize to novel circumstances. As a result, they might not perform well when presented with images that are substantially dissimilar from those in the training dataset. This may present a challenge in applications where the CNN must handle a wide range of image types.

Applications of Convolutional Neural Networks: The evident contenders encompass conventional convolutional neural network (CNN) implementations observed in everyday scenarios, such as software for speech recognition, image categorization and facial identification. These beliefs are very prevalent among individuals without specialised knowledge, such as ourselves and significantly influence our everyday experiences, particularly inside image-centric social media sites such as Instagram. Presented here is a compilation of several prominent applications associated with CNN (Convolutional Neural Networks) as developed and utilised by CNN (Cable News Network).

Understanding Gray Areas: The goal of adding gray areas to CNNs is to present a far more accurate representation of reality. CNNs currently observe a true and false value for each inquiry, operating largely just like a machine. However, we recognize as humans that there are countless shades of gray in the real world. It will be easier for the computer to comprehend and absorb fuzzier logic if it can see the gray region that exists in human thought and that we try to avoid. This will enable CNN to have a more complete picture of what people see.

Advertisin: With the introduction of programmatic buying and data-driven tailored advertising, CNNs have already made a significant effect in the advertising industry.

Other Interesting Fields: With the development of autonomous automobiles, mimicking human behavior robots, aids for human genome mapping projects, earthquake and natural catastrophe forecasting and perhaps even self-diagnosis of medical issues, CNNs are positioned to be the technology of the future. Therefore, you wouldn't even need to make a trip to the clinic or make an appointment with a doctor to be sure your severe cold symptoms are just the flu and not the signs of a rare disease. The diagnosis of brain cancer is one issue on which researchers are focusing using CNNs. More lives that are affected by brain cancer may be saved if the disease is detected sooner.

Summary

- Deep learning is a subfield of machine learning that focuses on using artificial neural networks to model and solve complex problems. It has gained significant attention

Notes

due to its ability to automatically learn and extract features from data, enabling the creation of highly accurate predictive models. Deep learning has revolutionized various fields, including image and speech recognition, natural language processing, game playing and more. Deep learning's ability to automatically learn from data has led to significant breakthroughs and has reshaped the landscape of artificial intelligence. It continues to advance with research into more effective architectures, training algorithms and applications.

- Neural networks are computational models inspired by the human brain's structure and functionality. They are used in various machine learning tasks, including pattern recognition, classification, regression and more. Neural networks consist of interconnected layers of nodes (neurons) that process and transform data to make predictions or decisions. At the core of a neural network is the neuron, a computational unit that processes input data and produces an output. The neuron model, also known as the perceptron model, consists of several components. The neuron model is a basic building block that forms the foundation of artificial neural networks, a neural network is composed of multiple layers of interconnected neurons.
- Neural networks have the ability to model highly complex relationships in data, making them suitable for tasks like image recognition, language processing and more. Their utility lies in their capacity to automatically learn and extract features from the data, eliminating the need for manual feature engineering. Through a process called backpropagation, neural networks adjust their weights and biases during training to minimize the difference between predicted and actual outputs. Neural networks have achieved remarkable successes in various fields, including image classification, speech recognition, natural language processing and playing games. Their versatility and ability to capture intricate patterns make them a powerful tool for solving a wide range of problems.
- The design of artificial neural networks is inspired by the biological neurons and neural networks in the human brain. While artificial neural networks are not direct replicas of biological systems, there are some connections and motivations. While the analogy between artificial and biological neural networks is not perfect, these connections provide a conceptual framework for understanding the design and behavior of deep learning models. Recurrent Neural Networks (RNNs) are designed for sequence data, where the order of elements matters. RNNs are used in tasks like natural language processing, speech recognition and time series analysis, where the order of data is crucial for understanding and making predictions.These architectures showcase the diversity and specialization of neural networks for various types of data and tasks. Understanding their strengths and design principles is essential for effective application in solving real-world problems.
- The perceptron is the fundamental building block of neural networks and is the simplest form of an artificial neuron. It was introduced by Frank Rosenblatt in the 1950s. The perceptron takes a set of inputs, applies weights to them, sums the weighted inputs, adds a bias term and then passes the result through an activation function. A feed-forward neural network, also known as a single-layer neural network, consists of an input layer, one or more hidden layers and an output layer. Each neuron in a layer is connected to all neurons in the subsequent layer, but there are no connections within the same layer. Convolutional Neural Networks (CNNs) are specialized for processing grid-like data, particularly images. They excel at detecting patterns and features within images due to their unique architecture. CNNs are widely used in image classification, object detection, image generation and various computer vision tasks.

Notes

Glossary

- **ANNs:** Artificial neural networks
- **DNNs:** Deep neural networks
- **CNNs:** Convolutional neural networks
- **RNNs:** Recurrent neural networks
- **DBNs:** Deep belief networks
- **GPUs:** Graphics Processing Units

Check Your Understanding

1. What is the primary advantage of deep learning models over traditional machine learning models?
   a) Simplicity and ease of implementation
   b) Faster training times
   c) Ability to automatically learn features from data
   d) Reduced need for labeled training data
2. What is the role of the "hidden layers" in a deep neural network?
   a) They automatically learn features from the data.
   b) They process input data and produce final predictions.
   c) They connect the input layer to the output layer.
   d) They prevent overfitting by adding noise to the data.
3. What does the term "backpropagation" refer to in the context of deep learning?
   a) A training technique for reinforcement learning models.
   b) A method for adjusting neural network parameters using gradients.
   c) A type of activation function used in convolutional layers.
   d) An approach to ensemble learning using multiple models.
4. What is a key challenge addressed by regularization techniques in deep learning?
   a) Increasing the number of layers in neural networks.
   b) Reducing the depth of neural networks.
   c) Minimizing the number of training epochs.
   d) Avoiding overfitting and improving generalization.
5. What is the primary purpose of neural networks in machine learning?
   a) To capture and learn patterns from data
   b) To perform mathematical operations on data
   c) To automate repetitive tasks in data preprocessing
   d) To generate random numbers for statistical analysis
6. In the context of a neuron model, what is the purpose of the activation function?
   a) To determine the initial weight of the neuron
   b) To add a constant bias to the neuron's output
   c) To introduce non-linearity into the neuron's output
   d) To connect the neuron to other neurons in the network

Notes

7. What is a key advantage of using neural networks for modelling and solving complex problems?
   a) They require minimal computational resources for training.
   b) They eliminate the need for labeled training data.
   c) They can only be applied to problems with a small number of variables.
   d) They can automatically learn and extract features from data.
8. What is a key similarity between artificial neural networks and biological neural networks?
   a) Artificial neural networks consist of neurons that communicate through physical connections.
   b) Both types of networks have hierarchical structures with hidden layers.
   c) Biological neural networks process information sequentially.
   d) Artificial neural networks can perform autonomic functions.
9. Which popular CNN architecture introduced the concept of residual blocks?
   a) LeNet
   b) VGGNet
   c) AlexNet
   d) ResNet
10. What is the primary advantage of Recurrent Neural Networks (RNNs) in handling sequence data?
    a) RNNs have the ability to capture temporal dependencies and context.
    b) RNNs require less training data compared to other types of networks.
    c) RNNs can process sequences in parallel.
    d) RNNs eliminate the need for activation functions.
11. Which type of neural network is best suited for processing grid-like data, such as images?
    a) Multilayer Perceptron
    b) Recurrent Neural Network
    c) Convolutional Neural Network
    d) Radial Basis Function Network
12. In the context of a perceptron, what role does the activation function play?
    a) It determines the number of hidden layers in the network.
    b) It adjusts the weights of the inputs during training.
    c) It introduces non-linearity to the perceptron's output.
    d) It controls the flow of data in feedback loops.
13. What is the primary characteristic of a feed-forward neural network?
    a) It has connections that loop back from later layers to earlier layers.
    b) It processes data in a sequential manner.
    c) It only has an input layer and an output layer.
    d) It lacks feedback connections between layers.

14. What distinguishes a multilayer perceptron from a single-layer perceptron?
    a) Multilayer perceptrons have multiple input layers.
    b) Multilayer perceptrons can have one or more hidden layers.
    c) Multilayer perceptrons are used exclusively for regression tasks.
    d) Multilayer perceptrons don't use activation functions.
15. Which neural network architecture is specifically designed for processing grid-like data, such as images?
    a) Recurrent Neural Network (RNN)
    b) Multilayer Perceptron (MLP)
    c) Convolutional Neural Network (CNN)
    d) Radial Basis Function Network (RBFN)
16. What are the disadvantages of CNN?
    a) High computational requirements
    b) Difficulty with small datasets
    c) Vulnerability to adversarial attacks
    d) All of the above
17. What is he working of MultiLayer Perceptron Neural Network?
    a) The characteristic of the dataset is represented by the input node.
    b) Each input node sends the hidden layer's hidden layer the vector input value.
    c) Each edge in the hidden layer has a weight that is multiplied by the input variable. The sum of all the production values from the concealed nodes is calculated. to produce the result.
    d) All of the above
18. Which neural network is a type of artificial neural network in which there is no cycle in the connections between the nodes?
    a) Feed forward
    b) Recurrent
    c) Convolutional
    d) Artificial
19. Which among the following areas Deep Learning has made substantial progress?
    a) Convolutional neural networks (CNNs)
    b) Recurrent neural networks (RNNs)
    c) Deep belief networks (DBNs)
    d) All of the above
20. What is he full form of ILSVRC?
    a) Imagine Net Large-Scale Visual Recognition Challenge
    b) Image Net Large-Scale Visual Recognition Challenge
    c) Image Net Large-Scale Video Record Challenge
    d) Image Net Larger-Scale Visual Recorder Challenge

Notes

Exercise

1. Define deep learning
2. The Neuron Model
3. What do you meant by the neural network and its utility in modelling and solving problems?
4. Define RNNs.
5. Define perceptron and multilayer perceptron.
6. Explain feed forward neural network and convolutional neural network.

Learning Activities

1. What are some challenges and considerations when training deep neural networks on small datasets?
2. Discuss the role of gradient descent in optimizing the neural network's weights and its impact on the training process

Check Your Understanding-Answers

1. c	2. a	3. b	4. d
5. a	6. c	7. d	8. b
9. d	10. a	11. c	12. c
13. d	14. b	15. c	16. d
17. d	18. a	19. d	20. b

Further Readings and Bibliography

1. "Deep Learning" by Ian Goodfellow, YoshuaBengio and Aaron Courville.
2. "Neural Networks and Deep Learning: A Textbook" by Charu C. Aggarwal.
3. "Deep Learning for Computer Vision" by RajalingappaaShanmugamani.
4. "Deep Learning with PyTorch" by Eli Stevens, Luca Antiga and Thomas Viehmann.