Review Based on Different Deep Learning Architecture and their Application

Manikandan.B[1], Diviya raju[2], Ragavi.S[3], Sathiya Priya.R[4]

[1] *Manikandan.B PG Scholar, Electronics and Communication Engineering, Kumaraguru College of Technology, Tamilnadu, India*
[2] *Diviya Raju PG Scholar, Electronics and Communication Engineering, Kumaraguru College of Technology, Tamilnadu, India*
[3] *Ragavi.S PG Scholar, Electronics and Communication Engineering, Kumaraguru College of Technology, Tamilnadu, India*
[4] *Sathiya Priya.R PG Scholar, Electronics and Communication Engineering, Kumaraguru College of Technology, Tamilnadu, India*

ABSTRACT

Deep neural networks (DNNs) are currently widely used for much artificial intelligence (AI) applications including computer vision, speech recognition, and robotics. While DNNs deliver state-of-the-art accuracy on many AI tasks, it comes at the cost of high computational complexity. Accordingly, techniques that enable efficient processing of DNNs to improve energy efficiency and throughput without sacrificing application accuracy or increasing hardware cost are critical to the wide deployment of DNNs in AI system. An up-to-date overview is provided on four deep learning architectures, namely, Artificial Neural Networks, Auto encoder, convolutional neural network, deep belief network, and restricted Boltzmann machine, Recurrent Neural Network. Different types of deep neural networks are surveyed and recent progress is summarized.

Keyword: - *Artificial Neural Networks Auto encoder, convolutional neural network, deep belief network, Restricted Boltzmann machine, Recurrent Neural Network.*

1. INTRODUCTION

Most machine learning and signal processing techniques had exploited shallow structured architectures. These architectures typically contain at most one or two layers of nonlinear feature transformations. Examples of the shallow architectures are Gaussian mixture models (GMMs), linear or nonlinear dynamical systems, conditional random fields (CRFs), maximum entropy (MaxEnt) models, support vector machines (SVMs), logistic regression, kernel regression, multi-layer perceptron's (MLPs) with a single hidden layer including extreme learning machines (ELMs). For instance, SVMs use a shallow linear pattern separation model with one or zero feature transformation layer when the kernel trick is used or otherwise. (Notable exceptions are the recent kernel methods that have been inspired by and integrated with deep learning; e.g. Cho and Saul, 2009; Deng et al., 2012; Vinyals et al., 2012; Aslan et al., 2013). Shallow architectures have been shown effective in solving many simple or well-constrained problems, but their limited modelling and representational power can cause difficulties when dealing with more complicated real-world applications involving natural signals such as human speech, natural sound and language, and natural image and visual scenes[1].Machine learning techniques have been widely applied in a variety of areas such as pattern recognition, natural language processing and computational learning. With machine learning techniques, computers are endowed with the capability of acting without being explicitly programmed, constructing algorithms that can learn from data, and making data-driven decisions or predictions. During the past decades, machine learning has brought enormous influence on our daily life with examples including efficient web search, self-driving systems, computer vision, and optical character recognition [2]. In machine learning a linear increase in the amount of features (input parameters) exponentially increases the difficulty of training. Reducing the amount of features thus exponentially makes training easier. In the past, the solution was feature extraction; selecting features and constructing new features based on the existing features in the input. This leads to less features, and therefore is less difficult to train. Feature extraction is a special form of dimensionality reduction, which is the reduction of the amount of free variables in machine learning. Feature extraction is labor-intensive as the new features are constructed by hand. Constructed features essentially construct higher [3]. Deep learning refers to a class of machine learning techniques, where many layers of information processing stages in hierarchical architectures are exploited for pattern classification and for feature or representation learning. It is in the intersections among the research areas of neural network, graphical modelling, optimization, pattern recognition, and signal processing. Three important reasons for the popularity of deep learning today are drastically increased chip processing abilities (e.g., GPU units), the significantly lowered cost of computing hardware, and recent advances in machine learning and signal/information processing research [4]. Historically, the concept of deep learning originated from artificial neural network research. (Hence, one may occasionally hear the discussion of "new-generation neural networks".) Feed forward neural networks or MLPs with many hidden layers, which are often referred to as deep neural networks (DNNs), are good examples of the models with a deep architecture. Back propagation (BP), popularized in 1980's, has been a well-known algorithm for learning the parameters of these networks. Unfortunately back-propagation alone did not work well in practice then for learning networks with more than a small number of hidden layers. The pervasive presence of local optima and other optimization challenges in the non-convex objective function of the deep networks are the main source of difficulties in the learning. Back-propagation is based on local gradient information, and starts usually at some random initial points. It often gets trapped in poor local optima when the batch-mode or even stochastic gradient descent BP algorithm is used. The severity increases 12 significantly as the depth of the networks increases. This difficulty is partially responsible for steering away most of the machine learning and signal processing research from neural networks to shallow models that have convex loss functions (e.g., SVMs, CRFs, and MaxEnt models), for which the global optimum can be efficiently obtained at the cost of reduced modeming power, although there had been continuing work on neural networks with limited scale and impact (e.g., Hochreiter and Schmidhuber, 1997; LeCun et al., 1998; Bourlard and Morgan, 1993; Deng et al., 1994s; Bridle et al., 1998; Robinson, 1994; Morgan, et al., 2005 [5]. The aim of deep learning is to create higher level representations of the data through the use of multiple layers of nonlinear operations. These higher level representations are much more useful for classification than the basic features. The complete classifier, with all its components, form the machine learning architecture. In computer vision raw pixels form a poor feature for direct classification of data; pixels may be shifted, have noise or are displayed in different color, but may still represent the same number. A deeper architecture such as a multi-layer neural network performs better in this task, as it is able to form a higher level representation of the data [3]. Recently, to deal with effective representation of information speech signals and objects deep learning is state-of-art in speech processing and localization of object detection area. It is machine learning methodology. Deep learning has turned out to be successful in tackling with many AI problems including speech information processing and object detection. As research in AI is progressing in all areas, there is a need of model, which is capable of processing the complex input data and solving different kinds of complicated tasks. Deep architectures have been proved such kind of model. It is believed that deep architectures have good learning algorithms and excellent performance. Motivation behind the study of deep architectures for speech processing and object detection is the power of deep architectures for representation of features. There are several deep architectures, but CNNs and DBNs are the two milestones in the field of speech processing and image processing as well object detection. In recent years a lot of improvements have been made to the architectures of deep feed forward neural networks, drastically increasing the classification accuracy in many classification tasks. In this paper we investigate the architectures used in deep convolutional neural networks for computer vision.

2. DIFFERENT DEEP ARCHITECTURES

A) Artificial Neural Networks (ANN)
B) Restricted Boltzmann Machines (RBM)
C) Deep Belief Network (DBN)
D) Recurrent Neural Network (RNN)
E) Deep neural networks (DNN)
F) Auto encoder (AE)
G) Convolutional Neural Network (CNN)

A) ARTIFICAL NEURAL NETWORK

The human brain is composed of cells (neurons) that are the only cells capable of communicating with each other. This is one of the capabilities that allows humans to exhibit intelligent behaviour. ANNs are computational models based on the biological neural structure of the brain, as first proposed by McCulloch and Pitts (1943). This computational model, also known as connectionism, aims to mathematically represent and reproduce the way a human nervous system works shown in Fig.1. An ANN consists of a set of processing elements, also known as neurons or nodes, which are interconnected. It can be described as a directed graph in which each node performs a transfer function of the form

$$y_i = f_i\left(\sum_{j=1}^{n} w_{ij} x_j - \theta_i\right) \ldots\ldots (1)$$

Where y_i is the output of the node i, x_j is the jth input to the node, and w_{ij} is the connection weight between nodes i and. θ_i is the threshold (or bias) of the node. Usually, f_i is nonlinear, such as a Heaviside, sigmoid, or Gaussian function [6].

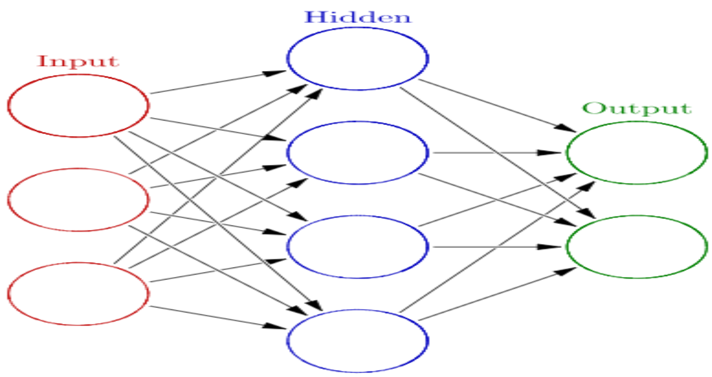


Fig -1: Artificial neural network

Artificial neural networks (ANNs) are statistical models directly inspired by, and partially modelled on biological neural networks. They are capable of modelling and processing nonlinear relationships between inputs and outputs in parallel. Artificial neural networks are characterized by containing adaptive weights along paths between neurons that can be tuned by a learning algorithm that learns from observed data in order to improve the model. In addition to the learning algorithm itself, one must choose an appropriate cost function. The cost function is what's used to learn the optimal solution to the problem being solved. This involves determining the best values for all of the tunable model parameters, with neuron path adaptive weights being the primary target, along with algorithm tuning parameters such as the learning rate. It's usually done through optimization techniques such as gradient descent or stochastic gradient descent. These optimization techniques basically try to make the ANN solution be as close as possible to the optimal solution, which when successful means that the ANN is able to solve the intended problem with high performance.
Architecturally, an artificial neural network is modelled using layers of artificial neurons, or computational units able to receive input and apply an activation function along with a threshold to determine if messages are passed along. In a simple model, the first layer is the input layer, followed by one hidden layer, and lastly by an output layer. Each layer can contain one or more neurons shown in Fig.2 .

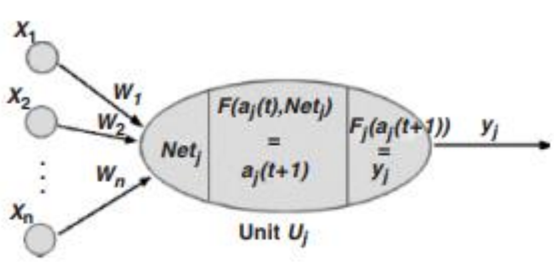


Fig -2: Artificial neuron main components

When a signal is transmitted from processing unit i to processing unit j, the signal x_i is modified by the synaptic weight (w_{ij}) associated with this communication channel. The modulated signals that arrive at unit j are added to form the net input Net_j as is shown in

$$Net_j = \sum_i x_i w_{ij} \ldots.. (2)$$

Each neuron is characterized in any instant of time t by a real value, called activation state or activation level, $a_j(t)$ Also, there is a function F, called activation function, which determines the next activation state based on the current activation state and the Net_j input of the neuron.

$$F\left(a_j(t), Net_j\right) = a_j(t+1) \ldots\ldots (3)$$

Associated with each unit there exists an output function f_j, that transforms the current activation level into an output signal y_j. This signal is sent through a unidirectional communication channel to other units in the network.

$$f_j\left(a_j(t+1)\right) = y_j \ldots. (4)$$

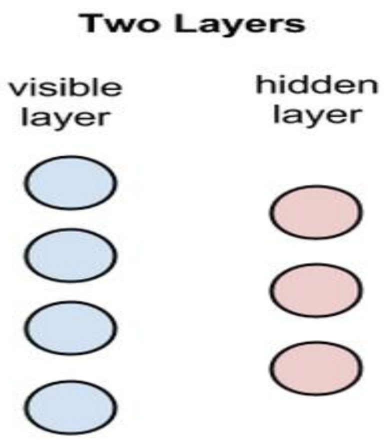
A learning algorithm is the process by which an ANN generates internal changes so that it can adapt its behaviour in response to the environment. The modifications that by the network during this process enable it to gain better performance, so it can overcome its output to the environment. When there is an external agent involved in the learning process, it receives the name of supervised learning. Back propagation is a supervised learning algorithm, used in feed-forward neural networks, which reduce the global error produced by the network over the weight space [7].

B) RESTRICTED BOLTZMANN MACHINE

A special BM consisting of a layer of visible units and a layer of hidden units with no visible-visible or hidden connections. In this part, a brief review of RBMs is given. RBMs are widely used in deep learning networks on account of their historical importance and relative simplicity [4]. The RBM was first proposed as a concept by Smolensky, and has become prominent since Hinton published his work [8] in 2006. RBMs have been used to generate stochastic models of ANNs which can learn the probability distribution with respect to their inputs. RBMs consist of a variant of Boltzmann machines (BMs). BMs can be interpreted as NNs with stochastic processing units connected bidirectional. Since it is difficult to learn aspects of an unknown probability distribution, RBMs have been proposed to simplify the topology of the network and to enhance the efficiency of the model. It is well recognized that an RBM is a special type of Markov random fields with stochastic visible units in one layer and stochastic observable units in the other layer [2]. Once the states of the units in one layer are given, all the units in the other layers will be updated. This update process will carry on until the equilibrium distribution is reached. Next, the weights within an RBM are obtained by maximizing the likelihood of this RBM. Specifically, taking the gradient of the log-probability of the training data, the weights can be updated according to

$$\frac{\partial \log p(v^0)}{\partial w_{ij}} = \langle v_i^0 h_j^0 \rangle - \langle v_i^\infty h_j^\infty \rangle \ldots..(1)$$

Where, w_{ij} represents the weight between the visible unit i and the hidden unit j. $\langle v_i^0 h_j^0 \rangle$,$\langle v_i^\infty h_j^\infty \rangle$ are the correlations when the visible and hidden units are in the lowest layer and the highest layer, respectively. RBMs are shallow, two-layer neural nets that constitute the building blocks of deep-belief networks. The first layer of the RBM is called the visible, or input, layer, and the second is the hidden layer.



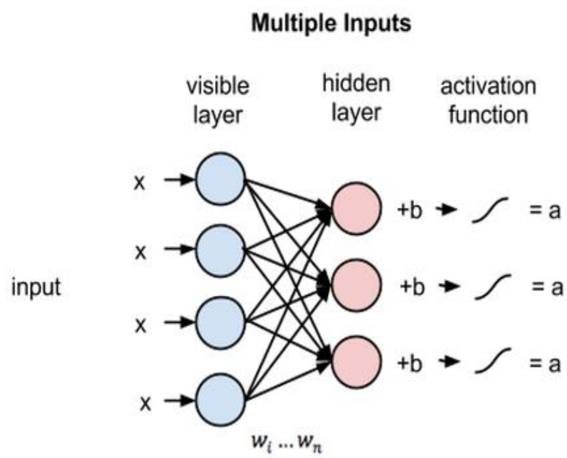


Fig -3: RBM with two layers

Fig -4: *RBM with multiple inputs*

Symmetrical means that each visible node is connected with each hidden node. Bipartite means it has two parts, or layers, and the graph is a mathematical term for a web of nodes. At each hidden node, each input x is multiplied by its respective weight w. That is, a single input x would have three weights here, making 12 weights altogether (4 input nodes x 3 hidden nodes). The weights between two layers will always form a matrix where the rows are equal to the input nodes, and the columns are equal to the output nodes. Each hidden node receives the four inputs multiplied by their respective weights. The sum of those products is again added to a bias (which forces at least some activations to happen), and the result is passed through the activation algorithm producing one output for each hidden node.

A RBM is consists of m visible units V = ($V1$, ….) to represent observable data and n hidden units H = ($H1$, …) to capture dependencies between observed variables. In binary RBM, the random variables (V, H) takes values (v, h) Є $\{0,1\}$ $m+n$ and the joint probability distribution under the model is given by as follow in (1):

$$P(v,h) = \frac{1}{2} e^{-E(v,h)} \text{........ (1)}$$

For all i Є {1…..,n} and j Є (1……,m), wi,j is a real valued weight associated with the edge between units Vj and Hi and bj and ci are real valued bias terms associated with jth visible and ith hidden variable.

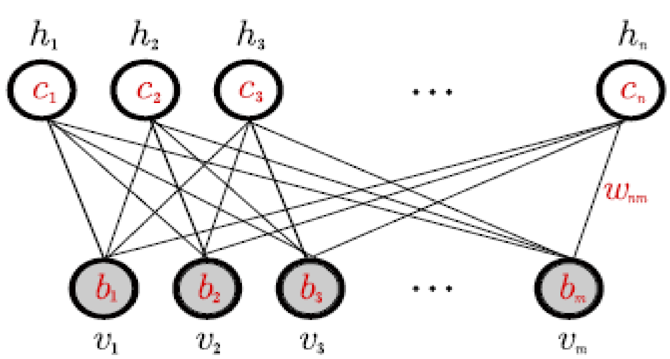


Fig -5: Restricted Boltzmann machine model

C) DEEP BELIEF NETWORKS

The main building block of a DBN is a bipartite undirected graphical model called a restricted Boltzmann machine (RBM). Due to the presence of the partition function, model selection, complexity control, and exact maximum likelihood learning in RBM's are intractable Today Deep belief network is used for recognizing and generating images. DBN is used as a nonlinear model for feature extraction and dimension reduction. DBNs are composed of multiple layers of RBMs. RBM is a Boltzmann machine where the connections between hidden visible layers are disjointed shown in Fig.6. Also the Boltzmann machine is an undirected graphical model (or Markov Random Field) [9].

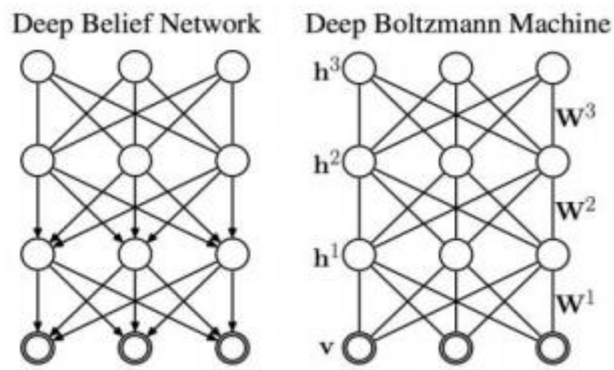


Fig -6: deep belief network

D) RECURRENT NEURAL NETWORKS

Recurrent Neural Networks are in the family of feed-forward neural networks. They are different from other feed-forward networks in their ability to send information over time-steps. Today's recurrent neural networks (RNNs) have been proving themselves as powerful predictive engines. When it comes to certain sequential machine learning tasks, such as speech recognition, RNNs are reaching levels of predictive accuracy, time and time again, that no other algorithm can match. However, the first generation of RNNs, back in the day, were not so hot. They suffered from a serious setback in their error-tweaking process that held up their progress for decades. Finally, a major breakthrough came in the late 90s that led to a new generation of far more accurate RNN. Recurrent neural networks are feed forward neural networks augmented by the inclusion of edges that span adjacent time steps, introducing a notion of time to the model. Like feed forward networks, RNNs may not have cycles among conventional edges. However, edges that connect adjacent time steps, called recurrent edges, may form cycles, including cycles of length one that are self-connections from a node to itself across time. At time t, nodes with recurrent edges receive input from the current data point ($x^{(t)}$) and also from hidden node values $h^{(t-1)}$ in the network's previous state. The output yˆ (t) at each time t is calculated given the hidden node values $h^{(t)}$ at time t. Input $x(t-1)$ at time $t-1$ can influence the output $\hat{y}^{(t)}$at time t and later by way of the recurrent connections[10].Two equations specify all calculations necessary for computation at each time step on the forward pass in a simple recurrent neural network as in Fig.7.

$$h^{(t)} = \sigma\left(W^{hx}x^{(t)} + W^{hh}h^{(t-1)} + b_h\right)\text{….. (1)}$$

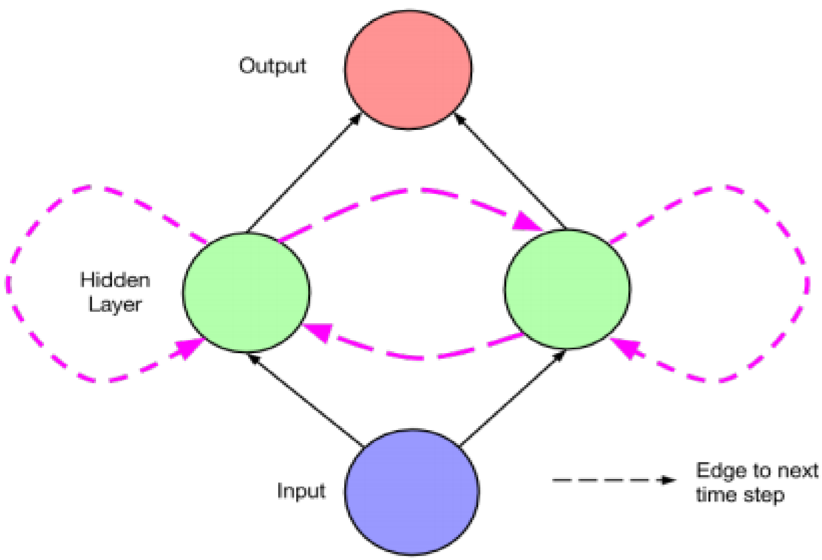


Fig -7: A simple recurrent network

At each time step t, activation is passed along solid edges as in a feed forward network. Dashed edges connect a source node at each time t to a target node at each following time $t + 1$.

$$\hat{y}^{(t)} = softmax(W^{yh}h^{(t)} + b_y) \ldots\ldots (2)$$

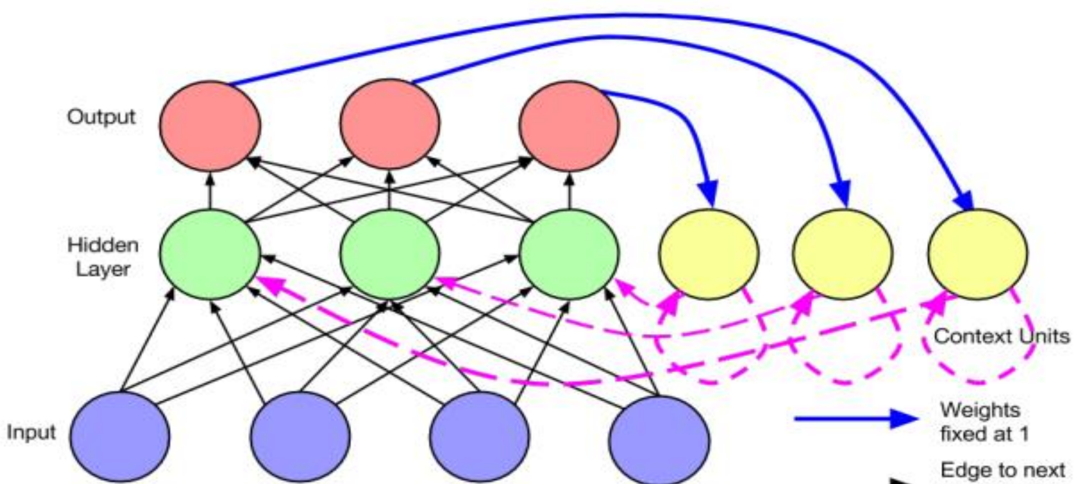


Fig -8: A recurrent neural network as proposed by Jordan

Output units are connected to special units that at the next time step feed into themselves and into hidden units. The time domain by concatenating sequences of three tokens. For each three token segment, e.g. “011”, the first two tokens (“01”) are chosen randomly and the third (“1”) is set by performing xor on the first two. Random guessing should achieve accuracy of 50%. A perfect system should perform the same as random for the first two tokens, but guess the third token perfectly, achieving accuracy of 66.7%.

E) DEEP NEURAL NETWORKS

DNN is inspired by human visual system; deep neural networks are kind of neural network with one input, one output and multiple fully-connected hidden layers in between. Each layer is represented as a series of neurons and progressively extracts higher and higher-level features of the input until the final layer essentially makes a decision about what the input shows [9] **.**

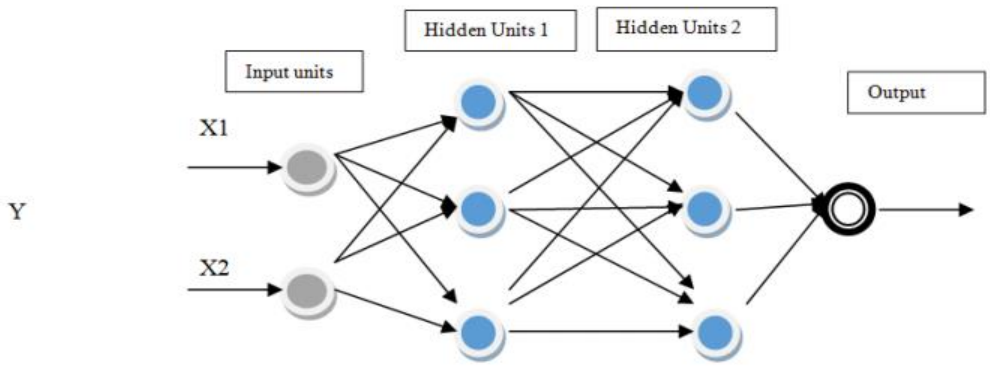


Fig -9: Simple deep neural network

Training process of deep neural networks

The Process of improving the accuracy is called Training, Training just like a Machine learning method. To train a network, the output from forward prop is compared to the output that is known to be correct and the cost is different of two. The point of the training is to make cost as possible, across millions of training examples. To do this network tweaks the weights and biases step by step until the prediction closely match the correct output shows in Fig.9. Once trained well, a neural network has the potential to make accurate prediction each time. Learns by generating an error signal that measures the difference between the predictions of the network and the desired values and then using this error signal to change the weights (or parameters) so that predictions get more accurate.

F) AUTOENCODER

An auto encoder (AE), which is another type of ANNs, is also called an autoassociator. It is an unsupervised learning algorithm used to efficiently code the dataset for the purpose of dimensionality reduction[[11]**,**[12]**,**[13]**]**. During the past few decades, the AEs have been at the cutting edge among researches on the ANN. In 1988, Bourlard and Kamp [14] found that a multilayer perceptron (MLP) in auto-association mode could achieve data compression and dimensionality reduction in the areas like information processing. Recently, the AEs have been employed to learn generative models of data shown in Fig.10. The input data is first converted into an abstract representation which is then converted back into the original format by the encoder function.

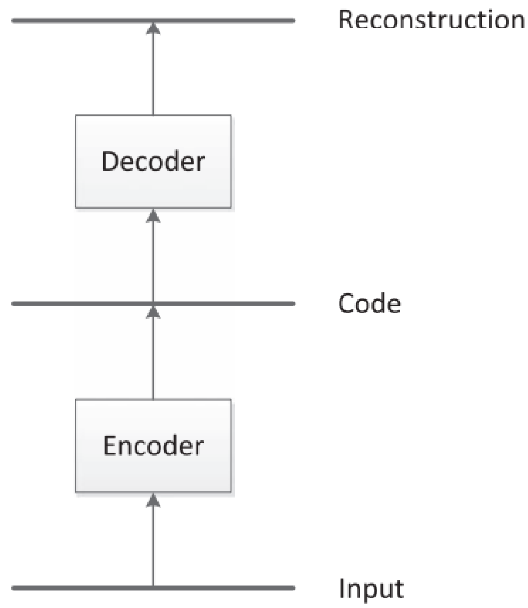


Fig -10: Schematic diagram of AE

More specifically, it is trained to encode the input into some representation so that the input can be reconstructed from that representation. Essentially, the AE tries to approximate the identity function in this process. One key advantage of the AE is that this model can extract useful features continuously during the propagation and filter the useless information. Similar to that for the DBNs, the training process for an AE can also be divided into two stages: the first stage is to learn features using unsupervised learning and the second is to fine-tune the network using supervised learning. To be specific, in the first stage, feed-forward propagation is first performed for each input to obtain the output value $\hat{x}$ then squared errors are used to measure the deviation of $\hat{x}$ from the input value. Finally, the error will be back propagated through the network to update the weights. In the fine-tuning stage, with the network having suitable features at each layer, we can adopt the standard supervised learning method and the gradient descent algorithm to adjust the parameters at each layer [2].

G) CONVOLUTIONAL NEURAL NETWORK

The goal of a CNN is to learn higher-order features in the data via convolutions. They are well suited to object recognition with images and consistently top image classification competitions. They can identify faces, individuals, street signs, platypuses, and many other aspects of visual data. CNNs overlap with text analysis via optical character recognition, but they are also useful when analyzing words as discrete textual units. They're also good at analyzing sound. The efficacy of CNNs in image recognition is one of the main reasons why the world recognizes the power of deep learning. The concept of CNNs is inspired by time-delay neural networks (TDNN). In a TDNN, the weights are shared in a temporal dimension, which leads to reduction in computation. In CNNs, the convolution has replaced the general matrix multiplication in standard NNs. In this way, the number of weights is decreased, thereby reducing the complexity of the network. Furthermore, the images, as raw inputs, can be directly imported to the network, thus avoiding the feature extraction procedure in the standard learning algorithms. It should be noted that CNNs are the first truly successful deep learning architecture due to the successful training of the hierarchical layers. The CNN topology leverages spatial relationships so as to reduce the number of parameters in the network, and the performance is therefore improved using the standard back propagation algorithms. Another advantage of the CNN model is that it requires minimal pre-processing [2]. For CNN, the input data need to be organized as a number of feature maps which will feed into CNN. In case of speech processing, how to organize the speech feature vectors is important for processing of CNN. The input can be thought as spectrogram. In CNN one's the input feature maps are formed, the convolution and the pooling layer apply their respective operation to generate activations of the unit in that layer. Layer can also refer as ply in CNN. After several convolutions and pooling layers, finally fully connected layers are formed which results in high-level reasoning of neural network.

CNN Architecture Overview

CNNs transform the input data from the input layer through all connected layers into a set of class scores given by the output layer. There are many variations of the CNN architecture, but they are based on the pattern of layers, as demonstrated in fig.11. CNN is comprised of one or more convolution layers followed by one or more fully connected layers, similar to neural network as shown in Fig.11. It consists of alternating convolution and pooling. In ordinary neural network each neurons of one layer connected with the neurons of another layer with a Learnable weights and biases, where each neuron after receiving the inputs perform a dot products optionally followed by non-linearity. But CNN is a back propagation neural network having [15] weight kernels of two dimensions operate on images. It is having three layers, first one is Convolution layer, the second one is pooling layer and the third one is Fully-Connected layer. The Network contains [16] set of layers, where each layer contains one or more planes. Basically it is explicitly assume that the inputs of CNN are images, hence the dimension of the input to the CNN layer is $m \times m \times r$ image which contains the raw pixel values of the image, where the height and width of the image is m and r is the number of channels. For example for an RGB image $r = 3$. The convolution layer is having k filters of size $n \times n \times p$, where n should be smaller than the image dimension and the size of p can either be same as r or can be smaller than r. This layer computes the output of neurons that are connected to local regions in the input. Then the element wise activation function is applied, such as max (o, x) thresholding at zero. At pooling layer down sampling operations along with spatial dimensions (width, height,) is performed, resulting the volume with less no of Dimensions. And finally fully connected layer computes the class scores. This is the way how convolution transforms the original image layer to the final class scores. CNN exploit the knowledge, that the [17] inputs arise from a spatial structure, it is not an independent element [11].

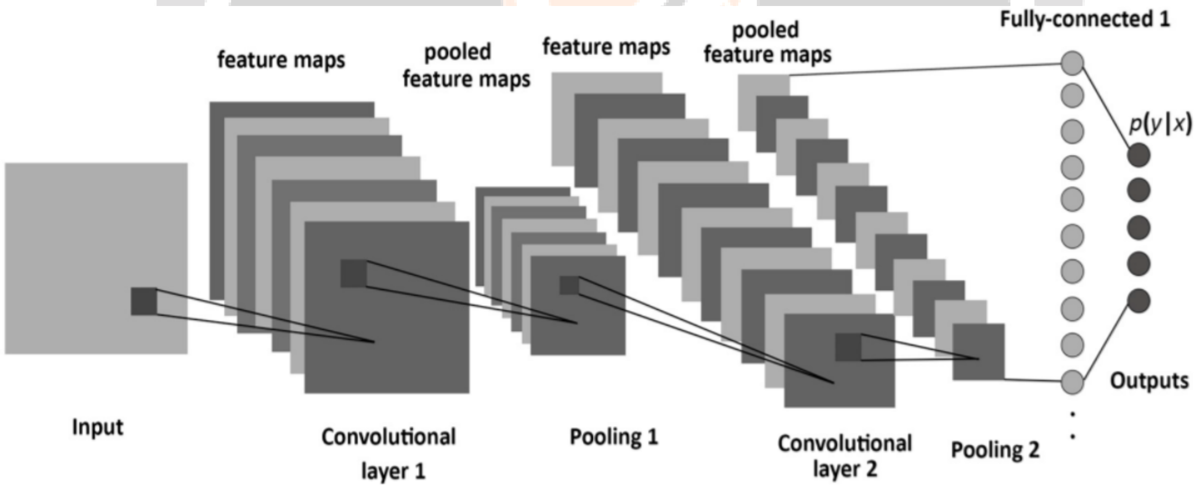


Fig -11: Convolutional neural network architecture

CNN is easier to train as it is having very less number of parameters with fewer connections, but despite of its relative efficiency of their local architecture, it shows expensive to apply in large scale for high resolution images. Deep CNN is capable of solving this problem by using a highly optimized Graphics Processing Unit execution of 2D convolution and all other operations natural in training CNN.

3. APPLICATIONS OF DEEP LEARNING

Deep learning techniques have provided powerful tools to deal with big data analysis. In recent years, massive amounts of data have been collected in various fields including cyber security, medical informatics [18], and social media. Deep learning algorithms are used to extract high-level features from these data in order to obtain hierarchical representations. Recently, deep learning has attracted the attention of many high-tech enterprises such as Google, Facebook and Microsoft. The architecture of deep networks has been widely applied in speech recognition and acoustic modelling for audio classification. Besides, deep learning approaches also play an important role in the area of image processing such as handwritten classification [19], high-resolution remote sensing scene classification, single image super resolution (SR), multi-category rapid serial visual presentation Brain Computer Interfaces (BCI), and domain adaptation for large-scale sentiment classification [20]. Moreover, deep architectures have also been employed in multi-task learning for NLP with an enhanced inference robustness [[21], [22]]. In the following, we will make a general review on several selected applications of the deep networks: speech recognition, computer vision, and pattern recognition. Detection is one of the most widely known sub-domains in computer vision. It seeks to precisely locate and classify the target objects in an image. In the detection tasks, the image is scanned to find out certain special issues. For example, we can use image detection to find out the possible abnormal tissues or cells in medical images. The deformable part-based model (DPM) proposed by Felzenszwalb is one of the most popular methods [23]. As demonstrated in, due to their strong abilities to capture the geometric information such as object locations, DNNs have been widely used for detection and have shown outstanding performance.

4. CONCLUSION

The use of deep neural networks (DNNs) has seen explosive growth in the past few years. They are currently widely used for many artificial intelligence (AI) applications including computer vision, speech recognition and robotics and are often delivering better than human accuracy. Consequently, techniques that enable efficient processing of deep neural network to improve energy-efficiency and throughput without sacrificing accuracy with cost-effective hardware are critical to expanding the deployment of DNNs in both existing and new domains. s. Some widely-used deep learning architectures are investigated and selected applications to computer vision, pattern recognition and speech recognition are highlighted. Based on these deep learning approaches, we can now use unsupervised learning algorithms to process the unlabeled data. Moreover, the trade-off between accuracy and computational complexity can be adjusted with flexibility in most deep learning algorithms. With the rapid development of hardware resources and computation technologies, we are confident that deep neural networks will receive wider attention and find broader applications in the future

5. REFERENCES

[1]. li deng, "A tutorial survey of architectures, algorithms, and applications for deep learning," Sip, vol. 3, no. 2, pp. 1–29, 2014.
[2]. W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," Neuro computing, vol. 234, no. November 2016, pp. 11–26, 2017.
[3] J. Van Doorn, "Analysis of Deep Convolutional Neural Network Architectures," arXiv Prepr. arXiv1009.5473, 2014
[4]. L. Deng, "Three Classes of Deep Learning Architectures and Their Applications: A Tutorial Survey," Research.Microsoft.Com, 2013
[5]. L. Deng and D. Yu, D Eep L Earning : 2014.
[6]. X. Yao, "Evolving artificial neural networks," Proc. IEEE, vol. 87, no. 9, pp. 1423–1447, 1999.
[7]. J. E. Villaverde, J. E. Villaverde, D. Godoy, D. Godoy, A. Amandi, and A. Amandi, "Learning styles' recognition in e-learning environments with feed-forward neural networks," J. Comput. Assist. Learn, vol. 22, no. 3, pp. 197–206, 2006.
[8]. G. E. Hinton, S. Osindero, and Y.-W. Teh, "Communicated by Yann Le Cun A Fast Learning Algorithm for Deep Belief Nets 500 units 500 units," Neural Comput., vol. 18, pp. 1527–1554, 2006.
[9]. P. Chandra, M. T. Student, and B. Dwarahat, "A Survey on Deep Learning its Architecture and Various Applications," vol. 1, no. 2, pp. 7–14, 2017.
[10] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A Critical Review of Recurrent Neural Networks for Sequence Learning," pp. 1–38, 2015
[11] A. V. N. Reddy and C. H. P. Krishna, "A Survey on Applications and Performance of Deep Convolution Neural Network Architecture for Image Segmentation," vol. 10, no. 9, pp. 979–989, 2017.
[12] a J. Holden et al., "Reducing the Dimensionality of," vol. 313, no. July, pp. 504–508, 2006.
[13] J. Schmidhuber, "Deep Learning in neural networks: An overview," Neural Networks, vol. 61, pp. 85–117, 2015.
[14] H. Bourlard and Y. Kamp, "Auto-Association by Multilayer Perceptrons and Singular Value Decomposition," Biol. Cybern, vol. 59, pp. 291–294, 1988.

[15] B. Sahiner et al., “Classification of mass and normal breast tissue: A convolution neural network classifier with spatial domain and texture images,” IEEE Trans. Med. Imaging, vol. 15, no. 5, pp. 598–610, 1996.
[16] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, “Face recognition: A convolutional neural-network approach,” IEEE Trans. Neural Networks, vol. 8, no. 1, pp. 98–113, 1997.
[17]. P. Y. Simard, D. Steinkraus, and J. C. Platt, “Best practices for convolutional neural networks applied to visual document analysis,” Seventh Int. Conf. Doc. Anal. Recognition, 2003. Proceedings. vol. 1, no. Icdar, pp. 958–963, 2003.
[18] N. Zeng, Z. Wang, H. Zhang, W. Liu, and F. E. Alsaadi, “Deep Belief Networks for Quantitative Analysis of a Gold Immunochromatographic Strip,” Cognit. Comput. vol. 8, no. 4, pp. 684–692, 2016.
[19]. I.-J. Kim and X. Xie, “Handwritten Hangul recognition using deep convolutional neural networks,” Int. J. Doc. Anal. Recognit. vol. 18, no. 1, pp. 1–13, 2015.
[20]. X. Glorot, A. Bordes, and Y. Bengio, “Domain Adaptation for Large-Scale Sentiment Classification: A Deep Learning Approach,” Proc. 28th Int. Conf. Mach. Learn., no. 1, pp. 513–520, 2011.
[21]. R. Collobert and J. Weston, “A unified architecture for natural language processing: Deep neural networks with multitask learning,” Proc. 25th Int. Conf. Mach. Learn., pp. 160–167, 2008.
[22]. N. D. Lane and P. Georgiev, “Can Deep Learning Revolutionize Mobile Sensing?,” Proc. 16th Int. Work. Mob. Comput. Syst. Appl. - HotMobile ’15, pp. 117–122, 2015. [23]. Reference 4
[23]. P. F. Felzenszwalb, R. B. Girshick, D. Mcallester, and D. Ramanan, “Object Detection with Discriminatively Trained Part Based Models,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 9, pp. 1–20, 2009