Convolution Neural Network: A Shallow Dive in to Deep Neural Net Technology

Shruti Karkra, Priti Singh, Karamjit Kaur

Abstract: ***It is always beneficial to reassess the previously done work to create interest and develop understanding about the subject in importance. In computer vision, to perform the task of feature extraction, classification or segmentation, measurement and assessment of image structures (medical images, natural images etc.) is to be done very efficiently. In the field of image processing numerous techniques are available, but it is very difficult to perform these tasks due to noise and other variable artifacts. Various Deep machine learning algorithms are used to perform complex task of recognition and computer vision. Recently Convolutional Neural Networks (CNNs-back bone of numerous deep learning algorithms) have shown state of the art performance in high level computer vision tasks, such as object detection, object recognition, classification, machine translation, semantic segmentation, speech recognition, scene labelling, medical imaging, robotics and control, , natural language processing (NLP), bio-informatics, cybersecurity, and many others. Convolution neural networks is the attempt to combine mathematics to computer science with icing of biology on it.***

CNNs work in two parts. The first part is mathematics that supports feature extraction and second part is about classification and prediction at pixel level. This review is intended for those who want to grab the complete knowledge about CNN, their development form ancient age to modern state of art system of deep learning system. This review paper is organized in three steps: in the first step introduction about the concept is given along with necessary background information. In the second step other highlights and related work proposed by various authors is explained. Third step is the complete layer wise architecture of convolution networks. The last section is followed by detailed discussion on improvements, and challenges on these deep learning techniques. Most papers consider for this review are later than 2012 from when the history of convolution neural networks and deep learning begins.

Index Terms: Convolution neural network (Covnet), Deep learning, Image Net, Neural Networks, Semantic Segmentation

I. INTRODUCTION

Since early 90's Neural networks have been quite popular in the machine learning and data science research communities but in year 2012 the graph of popularity stars increasing and still growing at very fast pace. Neural network provides an alternate methods of computing and mimic the complexity and processing of human brain. Neural network is made of several interconnected processing units (neurons), dedicated to work in parallel to solve a explicit problem (pattern recognition). They interpret sensory information or data through machine perception, labeling or clustering the raw input. The patterns they perceive are in the form of matrices, into which all certifiable information, be it pictures, text, or sound must be translated. NN help to assemble unlabeled data as per the similarities in the inputs, and data is classified in to different categories after it is trained [1]. Neural Networks have ability to learn complex, non-linear hypotheses from data without the need of modeling complex features.

Generally, in NN (Neural network) neurons are assembled in a layered structure. In most applications neural network contains minimum three types of layers – input layer, hidden layer, and output layer. The input layer receives the information from input files or from electronic sensors. The output layer sends information straightway to the outside world, to a computer process, or to any control system. Hidden layer are the middle layer that may vary from one to many. The inputs and outputs of middle layer neurons simply go to other neurons. [2]

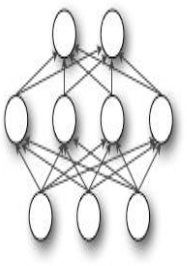


Fig1: Simple Feedforward neural network [3]

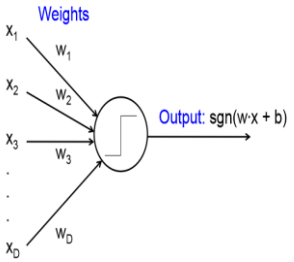


Fig2: Mathematical model for artificial neuron. [4]

The historical backdrop of neural network systems is very old and seemingly begun in the late 1800s with logical endeavors to examine the working and complexity of the human brain. Advancement here began in the mid of nineteenth century. In 1943, neurophysiologist Warren McCulloch and mathematician Walter Pitts modelled a simple neural system utilizing electrical circuits. In 1949, Donald Hebb published Hebbian Learning algorithm in honor of Donald Hebb; is one of the least complex learning rule for manmade neural networks. In 1959, Bernard Widrow and Marcian Hoff of Stanford developed models named "ADALINE" and "MADALINE (First NN applied to real world problems.) [4]. Later in 19th century single layer, multi-layer, hybrid and back propagation(most popular now a days as well) models were discovered.

Revised Manuscript Received on July 5, 2019

Shruti Karkra, Electronics and Communication Engineering,Amity University Haryana,Gurgaon,India,

Priti Singh, Electronics and Communication Engineering,Amity University Haryana,Gurgaon,India.

KaramJit Kaur, Electronics and Communication Engineering,Amity University Haryana,Gurgaon,India.

Till 2005 the development in the area of neural network is very slow but with the invention of convolution networks and other several deep layered networks the growth chart has increased exponentially.

Below table1 represents important milestones in the history of NN and deep learning architecture

Table1: Important contribution towards neural network and deep learning architectures[5].

Model/Architecture/Network	Year and contribution
MCP model: first Artificial Neural Network	1943,McCulloch & Pitts
Hebbian learning rule	1949,Hebb
Perceptron	1958, Frank Rosenblatt
Backpropagation	1974,Werbos
Boltzmann Machine	1985, Ackley,Hinton&Sejnowski
Restricted Boltzmann Machine	1986, Smolensky
RecurrentNeural Network	1986, Jordan
LSTM	1997, Hochreiter&Schmidhuber
Le net (starting age of deep learning)	1998,Lecun
Deep Belief Network	2006,Hinton
AlexNet, starting age of CNN	2012, Krizhevsky,Sutskever,&Hinton
ZFNET	2013,Matthew Zieler,RobFergus
Google	2014,Google
Resnet	2015,Kaiming He,Microsoft
Inception V3-V4	Google2016
Exception	Google2017

Neural networks, when piled or stacked with number of layers are called deep neural networks[6]. DNN(deep neural networks) also referred as deep learning networks. These are different from the traditional single-layer-hidden neural networks because of number of hidden layers or their **depth**; (i.e. the number of nodes through which information passes in a multilayered network). Former versions of neural networks for e.g. perceptron were shallow, and they have single input-output layer, and maximum one hidden layer in between. Networks with greater than three layers referred as "deep" learning.

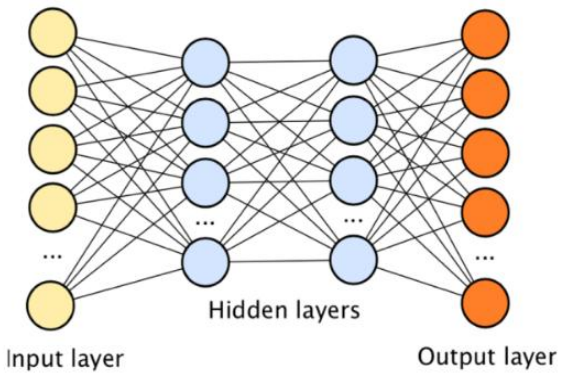


Fig3: Deep neural network with several hidden layers[7]

Based upon the previous layer output in deep-learning neural networks each layer of neurons trains on particular set of features. The more deeper you go, more complex and composite features can be recognized, because neural nets accumulate and combine features from the preceding layer. This progressive system of increasing complexity makes deep-learning networks to use sophisticated mathematical modeling to process very huge and high dimensional (structured or unstructured) data sets with millions of parameters. Data and parameters (features) are exponentially related. So, more is the number of neurons in a layer, more features it creates, and correspondingly more data it needs to train on. For example: If you have 10 features then you are required to provide at least 100 data values. There are number of deep learning architectures [8] are Such as CNN,RNN(recurrent neural network),deep belief networks, deep boltzmann machine, LSTM(long short term memory) etc. But in the field of computer vision, in large scale identification tasks, the convolution neural network model has made significant achievements since its inception [9]. CNN is modeled to adapt and pick up spatial features through backpropagation learning algorithm by using multi layered structures, such as covnet layers,relu layer , pooling layers, and fully connected layers [10]

II. RELATED WORK AND BACKGROUND AND INFORMATION

The capability (efficacy) of convolutional nets (ConvNets or CNNs) is to resolve complex image-driven recognition and classification tasks, with their precise yet simple architecture. This is one of the primary reasons, why the world is crazy about deep learning models. Convolution neural networks accredited major improvements in computer vision (CV) tasks and a few applications are; self-driving autos, drone, robots, medical diagnoses, home automation devices etc. Of late, the accessibility of larger datasets and computationally-powerful machines have driven the convolutional neural networks (CNNs) to outperform the of many conventional computer vision algorithms. [11] Number of companies have been utilizing advantages of CNN at the core of their services. For e.g. Automatics tagging feature in Facebook, Photo search feature by Google, Product recommendation and suggestions by Amazon, Google for their photo search, Amazon for their product recommendations and suggestions etc.

Convolutional neural networks ingest and process images as tensors[1] .A three-dimensional tensor is a array of numbers arranged in a cube. A 4-D tensor is the array nested one more level deeper.

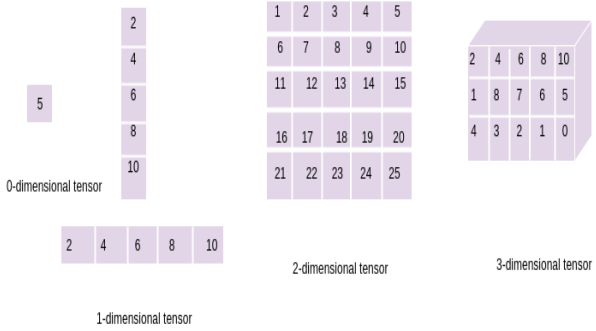


Fig 4: Structure of 1-D,2-D,3-D and 4-D Dimensions

** Tensor: The tensors are matrices of numbers with additional dimensions. Tensors maybe 1D,2D,3D,4D.High dimensional tensors are hard to visualize.*

Among various Deep learning models CNNs have been a prominent and leading technique in computer vision tasks since the surprising results (CNN classifies roughly 1.2million images in to1000 classes,) have been displayed on the object recognition competition (ILSVRC-ImageNet Large Scale Visual Recognition Competition in 2012 [6].After the success of Alexnet in the Imagenet Classification Challenge in 2012, CNNs have emerged as the leading class of methods for image classification, detection, and semantic segmentation. CNNs work by convolving the image with a variety of learned filters to extract a feature map, which represents the distribution of a variety of features in the image. While CNNs were first demonstrated for image classification tasks (assigning one class per image); were soon adopted for the tasks of object detection, recognition and pixelwise annotation (semantic segmentation). Focusing the attention of a CNN to a specific region of interest (ROI) of the image has also been explored in various ways ranging from using the ROI as additional input to the net to using different filters in different image regions. Attention-based approaches that focus the attention of the neural net on a specific region of the image, using an attention map are another major advance of recent years.

Earlier due to, the absence of large amount of information (data sets), more power consumption, and vast memory requirements the feedforward networks couldn't perform well on complex problems, like video scenes and large-scale images. Since 2006, numerous techniques have been produced to defeat the difficulties experienced in training the deep CNNs. Most remarkably, Krizhevsky et al. proposed a traditional CNN architecture and showed remarkable enhancements over past methods on the image classification task [6]. There are many variants of convolution architecture proposes by researchers as L.Jonathan S. Evan and. D. Trevor. They propose fully convolution networks for dense prediction and semantic segmentation [13]. Z. Xiang et.al. presented an integrated framework for multiscale Convolutional Networks for detection, localization and classification. Hancock et.al. demonstrated a new way of looking CovNets from the basis of Matlab and also explain that Max Pooling tends to leads to more precise classifier. Zhang et.al. explains the importance of nonlinear activation function in covnnet layer. Wang et.al. proposes the various optimization techniques used in recent years.

III. COVNET ARCHITECTURE

This multilayered neural network (CNN) is biologically inspired by the animal visual cortex. Very useful for all image processing applications CNN's initial/early layers identify features like lines, edges etc., and final/later layers assemble these features to higher-level attributes of the input. CNNs consist of multiple convolutional layers followed by a small number of fully-connected layers [14]. Architecture of convolutional neural net is divided in to four steps. First the image is feed into a convolutional layer (convolution operation performed using kernels of size 3x3 or 5x5) to extract features. Then Non-linearity (Relu Activation function) is added to each feature or activation map (output after convolution operation). In the third/next step pooling is done to reduces the dimensions of the extracted features (using down-sampling) while conserving the required spatial information. Now this output is fed to a fully connected multilayer perceptron; which assigns class labels or computes probabilities of a given class being present in the input. At the end the network is trained by back-propagation.

Figure below shows a CNN architecture for an object classification image task. Every layer of a CNN transforms the input volume to an output feature map, and finally driving to the fully connected layers, that results in mapping of the input data to a 1-Dimensional feature vector.

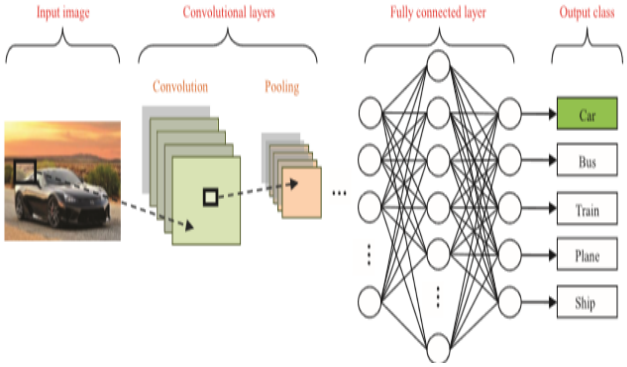


Fig5: CNN architecture[15]

Complete CNN architecture is summarized as below

1.3D volume of neuron
2.Convolution layer
3.Nonlinear activation — Feature extraction
4. Pooling layer
5.Fullyconnectedlayer — classification/recognition etc.

A. 3D Neuron

In CNN the convolution layer is the essential part that performs feature extraction, and perform linear & nonlinear functions, i.e. convolution operation (linear operation) and activation function (nonlinear operation). In convolution a small array of numbers, known as kernel (filter or neuron that have weight or parameters), is convolved along with the input (tensor or receptive field). An element-wise product is performed between input feature vector (input tensor) and each pixel of kernel and then added to get the output value. The output obtained after this convolution operation is called Feature map or activation map. (Fig.6.1). This technique is continued,i.e. apply multiple kernels (different feature vector) to build up an arbitrary number of feature maps(Fig.6.2). These feature maps represent various features of the input vectors. So different filters can be used as different feature extractors (Fig. 6.3). In convolution operation the size and the number of filters are the two main hyper parameters. Generally, the filter is of size 3x3, but sometimes filter size 7x7 or 5x5 is also used. The number of filters is random and controls the depth of output feature vector [10].

B. Convolution Layers

In CNN the convolution layer is the essential part that performs feature extraction, and perform linear & nonlinear functions, i.e. convolution operation (linear operation) and activation function (nonlinear operation).

In convolution a small array of numbers, known as kernel (filter or neuron that have weight or parameters), is convolved along with the input (tensor or receptive field). An element-wise product is performed between input feature vector (input tensor) and each pixel of kernel and then added to get the output value. The output obtained after this convolution operation is called Feature map or activation map. (Fig.6.1). This technique is continued,i.e. apply multiple kernels (different feature vector) to build up an arbitrary number of feature maps(Fig.6.2). These feature maps represent various features of the input vectors. So different filters can be used as different feature extractors (Fig. 6.3). In convolution operation the size and the number of filters are the two main hyper parameters. Generally, the filter is of size 3x3, but sometimes filter size 7x7 or 5x5 is also used. The number of filters is random and controls the depth of output feature vector [10].

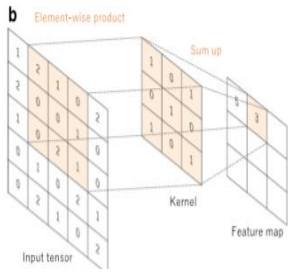


Fig:6.1StepwiseConvolution operation[10]

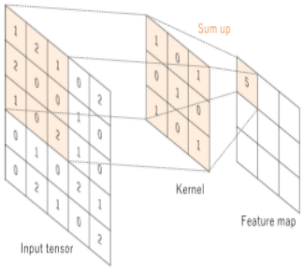


Fig:6.2 Stepwise Convolution operation [10]

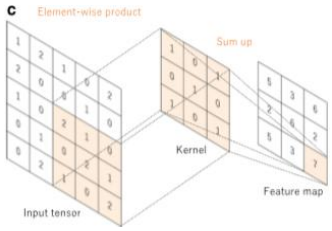


Fig:6.3 Stepwise Convolution operation [10]

very early layer, you could visualize the kernels (filters) that applied across the input vector are horizontal, vertical or diagonals line filters (low level features), that help to generate map of sharp boundaries in the image. Convolutional networks use these filters and slices of the image's feature and map them one by one. As the layers progress in the covnet structure the high level features revealed. By learning distinctive portions of a feature space, convolutional nets permit effortlessly versatile feature engineering.At high level, these filters work as feature identifiers (features like edges, vertical & horizontal lines, curves or the other simplest characteristics that images may have in common).

Element wise convolution operation can be summarized as:

$$Y_k = f.(W_k * X)$$

Where X is the input image, W_k is the convolutional filter with kth feature map. Y_k is the kth output feature map. * the multiplication sign here refers to the 2dimensional convolutional operator, that perform the element by element multiplication i.e. product of the convolution filter at each pixel position of the input tensor; and f(·) is the nonlinear activation function [15].Nonlinear activation functions allow for the extraction of nonlinear features.

Convolution neural networks retain much less connections and parameters so they are less demanding and quite simple to instruct and train, while hypothetically their best performance is expected to be only slightly poor[12].

Three hyperparameters regulate and controls the size of output feature vector (volume): The Filter size, Depth, Padding and Stride.

1. **Kernel Size**: The kernel size **or filter Size** defines the field of view of the convolution. A common choice is 3x3 and 5x5 pixels. 7x7 are also used depending on the application.

2. Depth (Filter count): Filter count: this is the most variable parameter. Using more filters results in a more powerful model, but we risk overfitting due to increased parameter count. Usually we start with a small number of filters at the initial layers, and progressively increase the count as we go deeper into the network.The depth of the output feature vector (output activation Map) is the total number (count) of filter applied to the input tensor. For e.g.if input image has volume 240*240*3 filter may have size 5*5*3).i.e. the outputs obtained from convolution between input and each kernel filter is stacked together to make the depth dimension of the output activation map. For e.g. let an input of size 32*32*3 is convolved with filters of size 5*5*3 and filter count is 10 with padding. Then the output dimensions will be 28*28*10.[18]

3.Stride: The stride defines the step size of the kernel i.e. during convolution operation how many pixel will translated horizontally and then vertically. Striding helps in reduction of size of output volume. In some Advanced CNN models, max pooling is also used instead of striding to reduce the For stride value 1, one pixel from the filter is shifted toward right horizontally. For stride value 2 (or uncommonly 3 or more) the pixels slide by 2 steps horizontally at a time. This procedure produce smaller output feature maps spatially(also called down sampling). Bigger strides (i.e. 2 or more) lead to fewer steps, means a large stride will produce a lesser output volume (activation map). This is imperative, to the fact that in convolution network architectures, the size of the matrices at each layer is in linear relationship with computational cost and training time.A larger stride means small output volume and less time to compute. For less and minimal overlap between the receptive fields bigger strides are suggested. Bigger stride skip over potential locations and make smaller resultant map.The following figure demonstrates a stride of size 2. Note that the feature map got smaller[21]

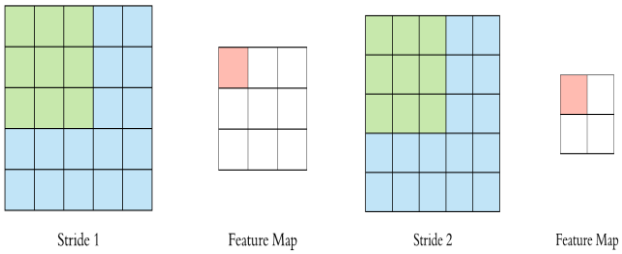


Fig:7 Output feature map with stride rate S= 1 (a) and stride rate S=2 (b)[21]

4. Padding:Padding is Stuffing of zero values around the edges or border of image.. It defines how the border of a sample is handled. It is used to control the output volume. With many succeeding convolutional layers, the output activation map very becomes smaller because some area is lost in every convolution layer. Very small output volume may loss some important spatial information. which creates another problem (not good prediction or classification). In networks having many layers, this approach (padding) is required so as to prevent the outputs from becoming too reduced. In case of higher stride value more than one layer of zeros are stuffed around the image. So zero padding enables us to regulate and dominate the spatial size of the output volumes [20]. Refer figure 8, the gray area around the input tensor is padding. Corners are either pad with zeros or with the values on the edge.

The size of the output volume is related to size of input volume (*W*), the receptive field size of the Conv Layer neurons (*F*), the stride value (*S*), and the value of padding (P) on the border. Formula for size of output volume is:
Size of output Volume = $(W-F+2P)/S+1$
For e.g. With input of dimensions 7x7, filter of size 3x3 ,zero padding and stride 1,the output volume we get is 5x5.Similarly with stride 2 the output is 3x3.[20].

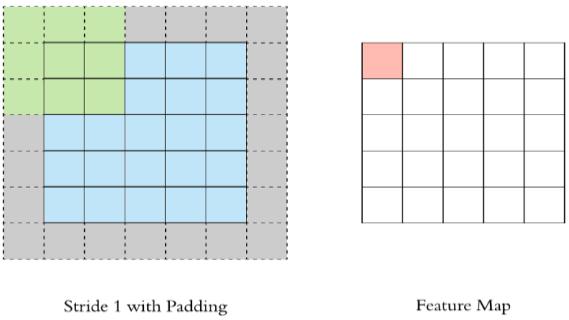


Fig:8 Zero padding[21]

In general, considering padding P=0 and stride S=1 ensure same spatial size of input and output volume.
Real-world example. The Krizhevsky et al. (2012) proposed Imagenet winner architecture have input images of dimensions[227x227x3]. At very first CovNet Layer, the neurons with receptive field (kernels) size F=11 , zero padding P=0and stride S=4 . So output volume with these values is given as

Output size =(227 - 11)/4 + 1 = 55

Depth of the covnet layer is given as K=96 .So output volume has size of [55x55x96].
All the neurons of dimensions [55*55*96] associated to the region of dimension [11x11x3] in the input volume. 96 is the depth or number of filters associated to dimension [11x11x3] of input, but with different feature weights.[20]

C. Rectifier Linear Unit

For a neural network to be influential, it must contain non-linearity. Which is done by passing the weighted sum of its inputs through an activation function, and CNN is no different in this case. So in CNN result of the convolution operation is passed through *relu* activation function[24] Earlier the nonlinear functions for e.g. sigmoidal function or Tanh were used, but researchers claimed that without making any noticeable difference to accuracy of the network the ReLu work far better as NN system is able to train faster [12].At lower layers NN trains very slowly because the gradient decreases exponentially, ReLu layer helps to improve the this vanishing gradient problem.

ReLU is an element wise operation (pixel by pixel).When ReLu operation is applied on the feature map, all the negative pixel values are replaced with zero. ReLu is to introduce non-linearity in Neural Network System, as most of the real-world data is not linear(Convolution is a linear operation, and non-linearity is accounted by adding a non-linear function like ReLU). All convolution network use ReLu operation, without this the network won't reach out its true potential.

The ReLU layer works as the function f(x) = max(0, x) for all values of input volume. Fundamentally Inception of ReLu in CNN just replace all negative pixel values to 0. It also overstate the nonlinear properties of the complete model without disturbing the receptive fields (output volume) of the conv layer.[22].

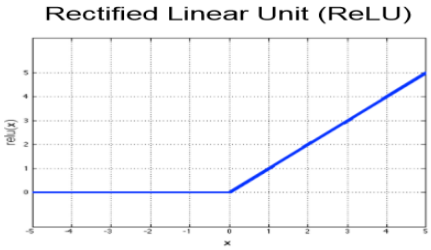
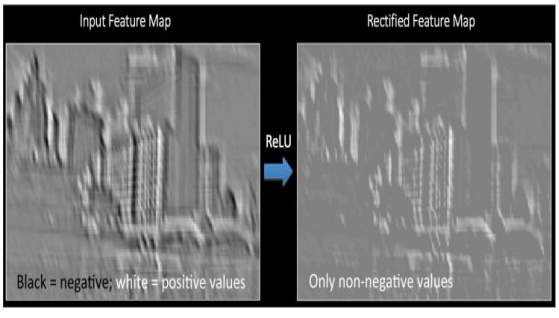


Figure 9 : Activation function ReLu[23]



(a) Figure 10: ReLU operation [10][23]

D. Pooling Layer

After a convolution operation we usually perform *pooling. Pooling* reduce the dimensionality (size of receptive field). Thus input volume for next conv layer decreases. Pooling helps to decrease the number of parameters, which both decreases the training time of NN and reduces overfitting. In pooling layers each feature map is down sampled, height and width is reduced, but the depth and important information is kept intact. Downsampling reduce the simultaneous information but it is good as subsampling reduce the computational overhead for next layer. Pooling also helps in achieving positional and rotational invariance. The most common type of pooling is *max pooling* and average pooling, where the maximum value in the pooling window or average of the pixel in window is considered. Like convolution operation we need to mention window size and stride size but this layer does not have any parameters. It just slides a window over its input& pulls out the max value or average value in the window. Other types/options for pooling layers are L2-norm pooling, stochastic pooling , spatial pyramid pooling, and def-pooling [5] The natural thinking behind this layer is that once explicit feature of the original input volume is known , its original location is not as significant as its relative location with respect to the other features. Pooling layer reduces the spatial dimensions of the input volume (as only length & width changes, depth remain same) very drastically. This layers serves two fundamental needs. First ,the count parameters or weights is decreased by 75%,hence the memory requirements thus decreasing the computation cost. The second is helps to reduce **overfitting**. Here is the result of max pooling using a 2x2 window and stride 2.

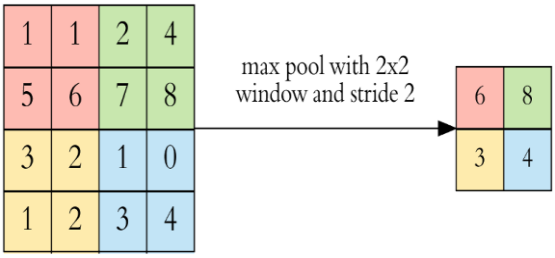


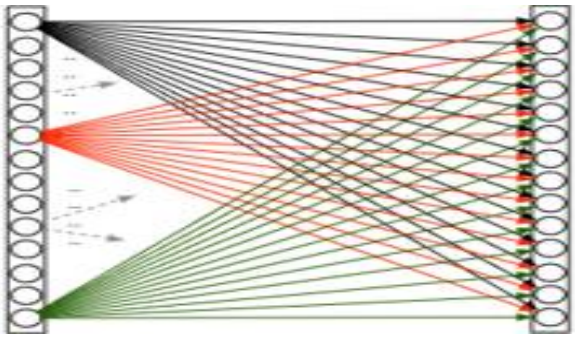
Fig11: Max Pooling[18]

IV. FULLY CONNECTED LAYER

Following number of CovNet layers (convolution, ReLu and pooling layers), the fully connected layers perform high level reasoning.At Fully connected layers the higher-order features transformed to class scores or probabilities. These predictions are used as output of the network with size volume [1 × 1 × N]; N is the number of expected output classes . For e.g. In CIFAR dataset the output volume dimensions are [1x1x10] here *N* is 10 which means total 10 classes of objects are present in dataset.[25]

As the name implies FC layers are completely coupled or connected to all the neurons in the former layer (Fig12). Some experts explain fully connected layer and convolution layer separately but some assumed FC layers as special case of convolution layer. At fully connected layer the multidimensional convolution operation converted to 1D convolution operations [26] .

A few CNN architectures use several FC layers at the end of the network. For example, AlexNet , has two FC layers which are followed by a softmax layer (convert output to estimated posterior probability distribution of various target classes) at the end. Deciding number of fully connected layers and depth is very ostensible as it has been observed that removing any internal layer led to drop in performance.[12].CNN can be applied to 2D as well 3D data and eventually F C layers transforms /convert the 2D or 3 D feature maps into a 1D feature vector).e.g. graph data ,NLP applications. MRI data.[27]



ull Connected Layer (FC1) Full Connected Layer (FC2)

Figure 12: FC layers example with two layers[26]

Deploying fully connected layers in CNN makes it expensive in terms of training or learning parameters. However these days, there are several new techniques (e.g.global average pooling, stitching and sifting etc.) that can be utilized as an option of fully connected networks. The predictive score of a class is determined at top most layer (last classification layer) using a soft-max layer. Depending upon the highest score, the classifier assign output to the corresponding class.[28].

Soft-max is the activation function which is applied to the last FC layer or classification layer and is usually different from other activation functions. It convert the output to estimated posterior probability distribution of the various target classes and normalizes output real values ranges between 0 and 1[10].

A. Training

Training of CNN is similar to the training other neural network but it is bit more mathematically complex due to convolution operations. Training a network on training dataset means to lessen the differences between estimated target predictions and actual ground truth label**.** CNNs, and ANNs in general use learning algorithms to adjust their free parameters (i.e., weights and baises) in order to attain the required network output [30]. The popular algorithm used for training is backpropagation . Backpropagation calculates the gradient (cost or loss or performance) of an objective function to calculate that how to minimize error (that affect network performance) by adjusting network parameters.[15].

Convolution network learns from training data. During the training process, the error is backpropagated and reduced until it is constant or can not be reduced further [29].

Training of CNN using backpropagation on some training database helps network to give label to each category of image. For example Pascal Voc database has 21 million natural images and CNN can classify this huge database to 10,000 categories.

1.Back Propagation (BP): In neural network backpropagation is a procedure that propagates the errors in backward direction to adjust the weights. Training of neural network with BP algorithm works in four progressive steps; the forward pass, error calculation (loss estimation), backward pass and weight update. Back propagation method is used to achieve gradient descent in neural networks. [31].

During the forward pass, the training image (array of numbers for e.g.32*32*3) passed through the entire network. As, at first all weights or filter values are initialized randomly, So particularly at output , no preference is given to any number or label. With the current weights network does not make any sensible conclusion about the classification scores. This leads to the **loss or error function**. The cost function (loss or error calculation) measures the variance between output predictions and given ground truth marks. For multiclass classification normally used loss functions are; cross entropy, mean squared error and many more. For first few iterations, the loss is very high and reduces gradually as the new weights updated ad no. of iterations increases. Now, when the output predictions are same as the ground truth labels (i.e. network got right prediction) the training of network stops. To accomplish this level the loss needs to be minimized.

During backward pass, the error is propagated back toward initial feed forward layers of network, where using chain rule the derivative is calculated. After this the weights (which contribute most towards loss) based upon loss function ,learning rate and activation gradient are updated.[24].

Here main objective is to minimize the loss function (similar to optimization problem in calculus). Gradient descent is the most popular optimization algorithm/function that iteratively learns and revise the learnable parameters to optimize or minimize the loss).In feed forward network Gradient or derivative represents the relationship between the loss calculated and the weight ie estimates the rate of change of error with respect to change in weight. This relationship is given by chain rule:

$$\frac{dError}{dweight} = \frac{dError}{dactivation} * \frac{dactivation}{dweight}$$

So learning in deep learning network is just adjusting the weights with respect to errors until it can not be reduced further.

2. Learning Rate: It is one of the significant hyper parameter and chosen by programmer. Learning rate is set before the training starts. If the learning rate is high that means bigger steps are taken to adjust the weights and model will take less time to converge on optimum set of weights. But if the learning rate is very high, then steps will not be precise and too big to reach optimum point (bigger jumps will skip the right weight updation and loss cannot be minimized.).

One cycle of complete forward pass, loss function, backward pass, & weight is concluded as one iteration. For each set of training images whole network is repeatedly trained on fixed number of iterations (known as Batch). Once weights are update and no error can be decreased further the training steps and network assumed to be well trained.[24].

B. Testing:

Data Set is divided in to three categories as training, validation and test data set. The network first go through the training data set (forward pass), and loss or error function is calculated. Then the error is propagated back to the previous layers and accordingly weights are updated. During validation data set, networks performance is over looked and other hyper parameter are fine tuned. After training and validation the final model's performance is evaluated and analysed using test data .Test data set plays important role in the best possible execution of machine learning models.[10]. For e.g. Cityscapes data set contains 5000 annotated images, from which 2975 images are for training purpose ,500 for validation data set, and rest 1525 are for testing purpose. Total classes that have been categorized on this data set are 19 [37].

C. Overfitting:

Overfitting is a phenomenon whereby a network works well on the training data set, but performs poorly on the test set. It is a fundamental problem/difficulty in machine learning because overfitted network model can not be generalized to new and unseen data. An Overfitted network learns some statistical features which are specific to particular training data set, means the network will learn some unnecessary details hence not able to perform well on new data set. For well fit network model the loss and accuracy on training dataset and validation data set should be monitored regularly. The network is said to be under fit if it does not perform well on training as well as on validation sets. On the off chance if the network fits excessively well on the training datset then it may lose its ability to generalize. There have been a few strategies proposed to limit overfitting. The best approach is to acquire large set of more training data. Other technique that can be use are: dropout, batch normalization, and data augmentation,[10]

1.Data Augmentation: This technique is especially helpful for small datasets .The human vision system is excellent at adapting image translations, rotations and other forms of distortions. Covnets are not very good at handling such distortions, they could fail terribly even due to minor translations.

1. Data Augmentation: It is a popular procedure to artificially enlarge a data set. It use label -preserving transformations to increase data set.[12].In particular, they created more images by applying translations & horizontal reflections, jittering, cropping to the training images, changing RGB channel intensities in training images, vertical flipping, rotation, whitening, shifting and other distortions.

Using this technique the network is able to handle distortions , hence, can work well on real time data.

2.Drop Out: An overfit network performs poorly on the test set. This is often due to excessive dependence on the presence of specific features in the training set. Dropout is the technique to combat over-fitting and reduce test errors. It works by randomly setting some activations to 0, or essentially killing them. By doing this, the network is forced to explore more ways of classifying the images instead of over-depending on some features. Drop out makes the network less sensitive and improve the generalization ability of the model[31]. The dropped out neuron do not contribute towards forward pass and back ward pass. So when input is presented to network it adapts it as a different model but each one of these models share same load. This strategy diminishes complex adjustments of neurons, and neuron do not depend on the nearness of other different neurons. It consequently, compelled to adapt more features and Making CNN a robust network.[12]

3. Batch Normalization: This technique is used to normalize the output of previous layers. Normalization of output of former layer is done by subtracting the batch mean and dividing it by the batch standard deviation. It eliminates the need of drop out layer and also helps to reduce overfitting[32].Likewise dropout, Batch normalization adds some noise to each hidden layer's activations. Therefore, if we use batch normalization, we will use less dropout, which is a good thing because we are not going to lose a lot of information. However, we should not depend only on batch normalization for regularization; we should better use it together with dropout.

V CONCLUSION

This review presents that CNN are arguably the most popular deep learning architecture and best known for their ability to recognize patterns/images.. To the best of our knowledge, in this paper we try to include and cover the literature which focuses on most robust convolution deep learning architecture.A brief introduction of classical neural net model have been included to give the necessary background knowledge about the subject in interest. Layer wise structure of convolution neural network have been included to give the reader a detailed information about CNN. Its enormous advantages like automatic detection, feature extraction, parameter sharing, exponential computing power etc. empowers CNN models to work on any hardware, and making them all around alluring. .The general discoveries show that CNN establishes a promising procedure with high grade performances in terms of accuracy ,precision, and classification. .However, the accomplishment of each Convolutional neural network model is profoundly reliant on the nature of the informational collection used. CNN is especially subject to the size and nature of the information. Given an organized dataset, CNNs are even fit for outperforming people at image recognition tasks.

Though CNN have shown state of the art performance in high level computer vision tasks, and enjoys a massive hype at the moment**.** But before concluding any study it is important to bear in mind that adoption of such technologies presents many challenges in addition to all buzz and excitement. we must consider/highlight the drawbacks that go along with.The biggest disadvantages are their „black box" naturethat leads to lack of transparency.CNN are computationally very expensive and time consuming to train with traditional CPUs.Neural networks suffers with the problem of over-fitting and generalization.

Convolution neyral network has lot of potential, but needs to overcome a few challenges before becoming a more versatile tool. The interest and enthusiasm for the field is, however, growing, and already today we see incredible real-world applications of this technology.

REFERENCES

1. Nicholson, C., & Gibson, A. (2014, November 27). A Beginner's Guide to Neural Networks and Deep Learning. Retrieved from https://skymind.ai/wiki/neural-network
2. Anderson,D., & McNeill, G.,(1992,August 20).ARTIFICIAL NEURAL NETWORKS TECHNOLOGY.A DACS state-of-art report Griffiss AFB, NY
3. Maind S., & Wankar ,(2014,January). Research Paper on Basic of Artificial Neural Network , International Journal on Recent and Innovation Trends in Computing and Communication,2(1),96-100. ISSN: 2321-8169
4. Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 65(6), 386-408. doi:10.1037/h0042519
5. Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018,February 1). Deep Learning for Computer Vision: A Brief Review. Computational Intelligence and Neuroscience, Hindawi 2018, 1-13. doi:10.1155/2018/7068349
6. Simonyan ,K., & Zisserman A.(2015 ,April 10).VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION, Visual Geometry Group, Published as a conference paper at ICLR 2015, University of Oxford.
7. Dickson, B. (2018, March 01). The limits and challenges of deep learning. Retrieved from https://bdtechtalks.com/2018/02/27/limits-challenges-deep-learning-gary-marcus.
8. Nguyen, H. D., Le, A. D., & Nakagawa, M. (2015,Nov). Deep neural networks for recognizing online handwritten mathematical symbols. 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR). doi:10.1109/acpr.2015.7486478
9. Al-Saffar, A. A., Tao, H., & Talab, M. A. (2017,Oct). Review of deep convolution neural network in image classification. 2017 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET). doi:10.1109/icramet.2017.8253139
10. Yamashita, R., Nishio, M., Do, R. K., & Togashi, K. (2018 June). Convolutional neural networks: An overview and application in radiology. Insights into Imaging, 9(4), 611-629. doi:10.1007/s13244-018-0639-9
11. Paszke,A.,Chaurasia,A.,Kim,S.,Culurciello,E.,(2016 June 16). ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation, Computer Vision and Pattern Recognition (cs.CV), arXiv:1606.0016 June 162147v1
12. Sharma, A. (2017, December 05). Convolutional Neural Networks in Python.Retrievedfrom https://www.datacamp.com/community/tutorials/convolutional-neural-networks-python
13. Long, J., Shelhamer, E., & Darrell, T. (2016,May). Fully convolutional networks for semantic segmentation. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2015.7298965 UC Berkeley, arXiv:1605.06211v1
14. .Ardakani, A., Condo, C., Ahmadi, M., & Gross, W. J. (2017 Oct 17). An Architecture to Accelerate Convolution in Deep Neural Networks. IEEE Transactions on Circuits and Systems I: Regular Papers, 65(4), 1349-1362. doi:10.1109/tcsi.2017.2757036.

15. Rawat, W., & Wang, Z. (2017 Oct). Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. Neural Computation, 29(9), 2352-2449. doi:10.1162/neco_a_00990.
16. Zahangir, M., Taha, M., T., Christopher, Westberg, Stefan, . . . K., V. (2018, September 12). The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches. Computer Vision and Pattern Recognition, arXiv:1408.3264
17. Shea, K., & Nash , R.(2015 Dec 2).An Introduction to Convolutional Neural Networks, Neural and Evolutionary Computing, arXiv:1511.08458v2.
18. Gupta, D., & Dishashree. (2018, July 13). Architecture of Convolutional Neural Networks (CNNs) demystified. Retrieved from https://www.analyticsvidhya.com/blog/2017/06/architecture-of-convolutional-neural-networks-simplified-demystified/
19. Murphy,J.(2016).An Overview of Convolutional Neural Network Architectures for Deep Learning,IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Microway, Inc.Fall 2016.
20. Karpathy05, A. (n.d.). CS231n Convolutional Neural Networks for Visual Recognition. Retrieved from http://cs231n.github.io/ ,Stanford university lecture notes.
21. Zhang, Q. (2018). Convolutional Neural Networks. 3rd International Conference on Electromechanical Control Technology and Transportation, 434-439. doi:10.5220/0006972204340439 © 2018 by SCITEPRESS – Science and Technology Publications.
22. Nair, V., & E. Hinton, G. (2010). Rectified Linear Units Improve Restricted Boltzmann Machines. Proceedings of the 27th International Conference on Machine Learning, Haifa, Israel, 2010.
23. Ujjwalkarn, U. (2017, May 29). An Intuitive Explanation of Convolutional Neural Networks. Retrieved from https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/
24. Deshpande, A. (2016,july). A Beginners Guide To Understanding Convolutional Neural Networks Part 2. Retrieved from https://adeshpande3.github.io/adeshpande3.github.io/A-Beginners-Guide-To-Understanding-Convolutional-Neural-Networks-Part-2/
25. Dev, D. (2017). Deep Learning with Hadoop. Birmingham: Packt Publishing.
26. Wei Ma,W.,Lu, J.(2017 Dec 4).An Equivalence of Fully Connected Layer and Convolutional Layer, Machine Learning (cs.LG) Department of of Computer Science EPFL, Lausanne, arXiv:1712.01252v1 .
27. Gibson, A., & Patterson, J. (2017, August). Deep Learning. Retrieved from https://www.oreilly.com/library/view/deep-learning/9781491924570/ch04.html. Publisher: O'Reilly Media, Inc., ISBN: 9781491924570
28. Hinton, G. E., Osindero, S., & Teh, Y. (2006). A Fast Learning Algorithm for Deep Belief Nets. Neural Computation, 18(7), 1527-1554. doi:10.1162/neco.2006.18.7.1527
29. Liu, T., Fang, S., Zhao, Y., Wang, P., & Zhang, J. (n.d.). Implementation of Training Convolutional Neural Networks. Retrieved from https://arxiv.org/pdf/1506.01195
30. Kang, M., & Hong, K. (2018). Automatic Bird-Species Recognition using the Deep Learning and Web Data Mining. 2018 International Conference on Information and Communication Technology Convergence (ICTC). doi:10.1109/ictc.2018.8539463
31. Joost van Doorn, Van J..(2014 June 23).Analysis of Deep Convolutional Neural Network Architectures, 21th Twente Student Conference on IT Enschede, The Netherlands. Copyright 2014,
32. Loffe, S., & Szegedy, C. (2015,March 2). Batch Normalization: Accelerating Deep Network Training by ... Retrieved from https://arxiv.org/pdf/1502.03167v3.pdf,
33. Garcia, A. G., Escolano, S. O., Opera, S., Martinez, V. V., & Rodriguez, J. G. (2017, April 22). 1 A Review on Deep Learning Techniques Applied to Semantic Segmentation, Submitted to TPAMI, Computer Vision and pattern recognition,, arXiv:1704.06857v1 [cs.CV] 22 Apr 2017
34. Dar, P., & Analytics Vidhya. (2018, April 04). 25 Open Datasets for Deep Learning Every Data Scientist Must Work With. Retrieved from https://www.analyticsvidhya.com/blog/2018/03/comprehensive-collection-deep-learning-datasets/?cv=1
35. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998, November). Gradient-based learning applied to document recognition Proc. IEEE 86(11): 2278–2324, 1998.).
36. Zeiler, M. D., & Fergus, R. (2014). Visualizing and Understanding Convolutional Networks. Computer Vision – ECCV 2014 Lecture Notes in Computer Science, 818-833. doi:10.1007/978-3-319-10590-1_53 Copyright Springer International Publishing Switzerland.
37. Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2017 May 12). DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(4), 834-848. doi:10.1109/tpami.2017.2699184
38. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., & Wei, Y. (2017). Deformable Convolutional Networks. 2017 IEEE International Conference on Computer Vision (ICCV). doi:10.1109/iccv.2017.89.
39. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., . . . Rabinovich, A. (2015 Sep). Going deeper with convolutions. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.201
40. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2016.308
41. Sergey, C., Vanhoucke, L., Vincent, V., & Alex, A. (2016, August 23). Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. Computer Vision and Pattern Recognition (cs.CV) Retrieved from https://arxiv.org/abs/1602.07261
42. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2016.90
43. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Identity Mappings in Deep Residual Networks. Computer Vision – ECCV 2016 Lecture Notes in Computer Science, 630-645. doi:10.1007/978-3-319-46493-0_38.
44. Chollet, F. (2017A pril 4). Xception: Deep Learning with Depthwise Separable Convolutions. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2017.195

AUTHORS PROFILE

Shruti Karkra is faculty in Electronics Engineering department at Amity University Haryana. She holds her B.Tech and M.Tech in Electronics and communication Engineering. She is pursuing PhD from Amity university in Image Processing .She has published 23+ papers in various national international journals (including scopus indexed) and presented various papers in national and international conferences.

Priti Singh Dr Priti Singh is M.Tech and Ph. D from IIT Delhi in Instrument Technology and having more than 19 years of experience in teaching and research. Presently, working as the Deputy Dean, Student welfare and Head of Electronics and Electrical department at Amity University Gurgaon. She is a merit and a national scholarships holder at the school and University level. She received Gold medal in Masters.Her research areas are Digital Signal Processing, Image Processing, Optical Instrumentation and Laser Applications in Optical Metrology. She has published 50+ International / National Journal papers, 03 books as an editor in these areas.

Karamjit Kaur Sekhon is faculty in Electronics Engineering department at Amity University Haryana. She has done PhD from Thapar University Patiala in Optical Fiber Communication. She is also the member of various technical bodies including IEEE, IET, IACSIT, IAENG, IAENG Society of Electrical Engineering, IAENG Society of Imaging Engineering and SIE. She has 40+ publications in national/international conferences/journals, attended/organized 30 faculty development workshops held time to time. Also, she has worked as member of reviewer committee for national and international journals, member editorial boards, Programme committee member/ reviewer for international conferences organized by IEEE, Springer and IACSIT.