iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 8 **Issue:** XI **Month of publication:** November 2020

DOI: https://doi.org/10.22214/ijraset.2020.32306

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue XI Nov 2020- Available at www.ijraset.com

A Detailed Study of Deep Learning using Convolutional Neural Network Approach

Parkavi J.
Department of Computer Application, Sri Satya Sai University of Technology and Medical Sciences, Indore, India.

Abstract: ***Deep learning is an Artificial intelligence network that works like a human brain to process data for detecting objects and making decisions without human supervision. Convolutional neural network (CNN) is a subset of deep learning methodology which is used for solving complex problems composed of multiple building blocks, such as convolution layers, pooling layers, and fully connected layers, and is designed to automatically and adaptively learn spatial hierarchies of features through a back propagation algorithm [10]. This review article offers a perspective on the basic concepts of CNN and its application in the field of Image processing, and discusses its challenges and future directions in the field of AI***
Keywords: CNN, AI network, ***backpropagation algorithm, spatial hierarchy, multiple building blocks, Image processing.***

I. INTRODUCTION

Computers today can detect photos automatically, classify them and can also describe the features of the picture and write a few sentences about it with proper grammar. All this process is done with the help of Deep Learning network (CNN) which actually learns patterns that naturally occur in an image. ImageNet is an image database organized according to the WordNet hierarchy, in which each node of the hierarchy is depicted by hundreds and thousands of images [12]. Labeled images of this ImageNet is used to train the Convolutional Neural Networks using GPU-accelerated deep learning frameworks such as Caffe2, Chainer, Microsoft Cognitive Toolkit, MXNet, PaddlePaddle, Pytorch, TensorFlow, and inference optimizers such as TensorRT.

In 2009 neural network was used for speech recognition and was finally used by Google from 2012. Deep learning is compared to the structure of human brain which is also called neural network, is a subset of Machine Learning that uses its model for computing. Deep Learning models, extracts complicated information from the input images with their multi-level structures, are very helpful in extracting complicated information from input images. Convolutional neural networks are also able to drastically reduce computation time by taking advantage of GPU for computation which many networks fail to utilize[1].

A. Neuron Working Process

The connectivity pattern between neurons is inspired by the organization of the animal visual cortex ,in which the biological process was inspired by convolutional neural network. In nature, there were huge number of dendrites (inputs), a cell nucleus (processor) and an axon (output) in a neuron. The basic units of neural network are Neurons. They are interconnected with other neurons to perform various functions by receiving the input, process it and returns an output. When the neuron activates, it accumulates all its incoming inputs, and if it goes over a certain threshold it fires a signal through the axon [2]. The neuron can learn from its previous train process.

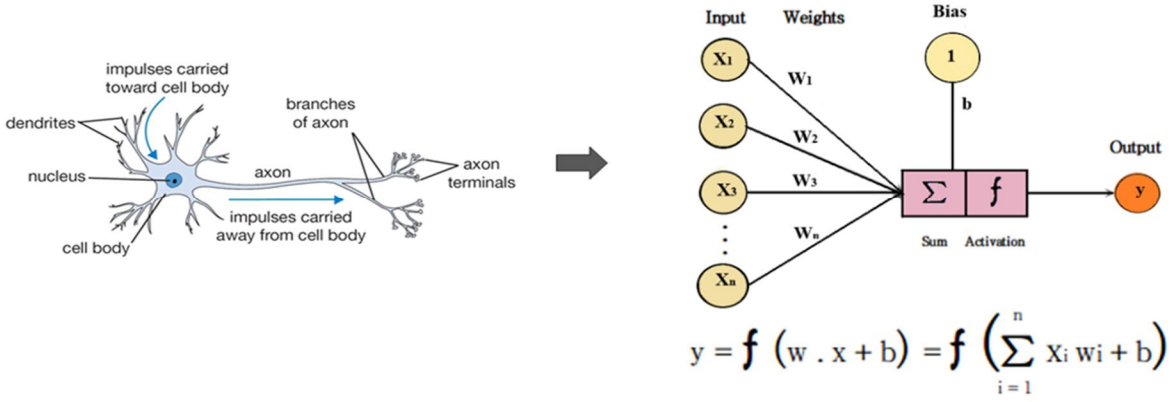


Fig. 1 Working Process of Neural Network

II. IMAGE RECOGNITION

Deep learning recognizes object in an image using convolutional Neural Network, The main component of a CNN is a convolutional layer. Convolution layer detect important features in the image pixels. It consists of an input layer, finite number of hidden layers, and an output layer. Each node in one layer is connected to other node in the right next layer.

Simple features such as edges and color gradients are detected by layer which are closer to the input (deeper), whereas next higher layers will merge simple features into more complex features. Finally, dense layers at the top of the network will blend very high level features and produce classification predictions [2].

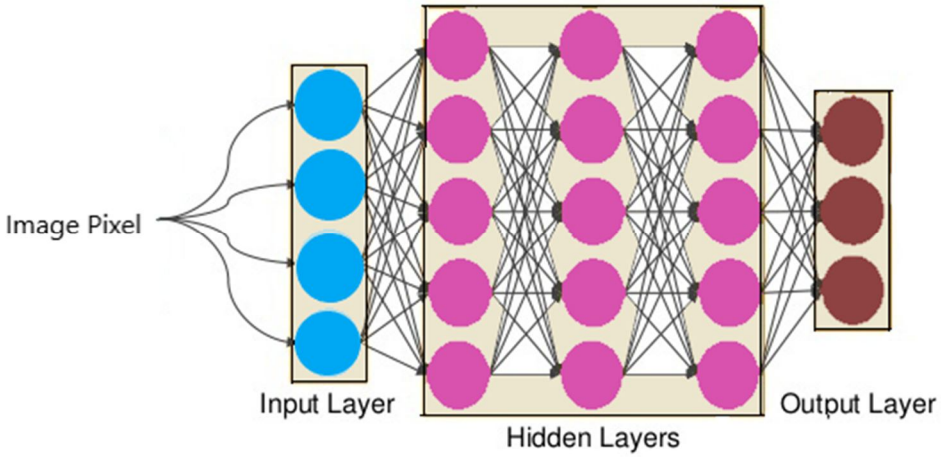


Fig. 2 Image Recognition Using Convolutional Neural Network.

1) *Input Nodes*: The Input node provide information from the outside world to the network and are together referred to as the "Input Layer". No computation is performed in any of the Input nodes – they just pass on the information to the hidden nodes [3].
2) *Hidden Nodes:* The Hidden nodes have no direct connection with the outside world (hence the name "hidden"). Computations are performed to transfer information from the input nodes to the output nodes. All the hidden nodes are grouped to forms a "Hidden Layer", while a feedforward network will only have one input layer and one output layer, it can have zero or many Hidden Layers.
3) *Output Nodes*: The Output nodes are collectively referred to as the "Output Layer" and are responsible for computations and transferring information from the network to the outside world [3].

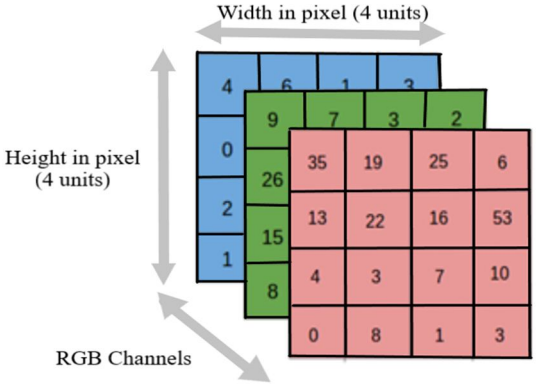


Fig. 3 RGB Channels

CNNs are usually applied to image data. All image is a matrix of pixel values. With vibrant images, particularly RGB (Red, Green, Blue)-based images, the presence of different color channels (3 in the case of RGB images) introduces an additional 'depth' field to the data, making the input 3-dimensional. Hence, for a given RGB image of size, say 255×255 (Width x Height) pixels, we'll have 3 matrices correlated with each image, one for each of the color channels. Thus, the image in its entirety, constitutes a 3-dimensional structure called the Input Volume (255x255x3).

III. CNN ARCHITECTURE

There are three main types of layers in CNN architecture:

A. *Convolution Layer*

Before the concept of convolution was presented there were lots of algorithms people used for Image classification. Features from images are created and fed those features into some classified algorithm like SVM which uses pixel level value as feature vector. It doesn't perform well because the training time is higher for large set of data's and also performance was poor when the data set comes with noise. Convolution layer uses information from adjacent pixels to down-sample the image into features by convolution and then use prediction layers to predict the target values.[5]

1) *Working Process of Convolutional Layer:* A convolution is a linear operation that involves the multiplication between an array of input data (image) and an array of weight called a filter or a kernel. The output from multiplying the filter with the input array one time is a single value [14].As the filter is applied many times to the input array, the result is a two-dimensional array of output data that represent a input filtering, As such, the two-dimensional output array from this operation is called a "feature map"

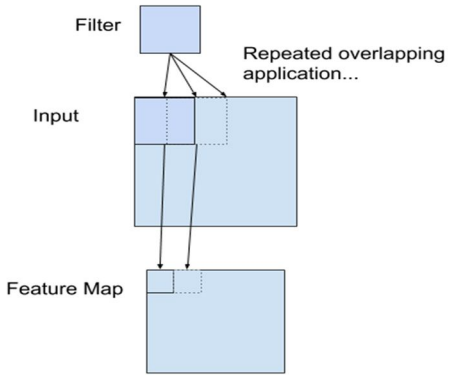


Fig. 4 Filter Applied to a Two-Dimensional Input to Create a Feature Map

Let us consider an array of input size 5 X 5 and filter of size 3 X 3. As the filter size is smaller than the input data, we use multiple filters to run over the input data to compute a Dot product. Each filter extracts different features from the input data.

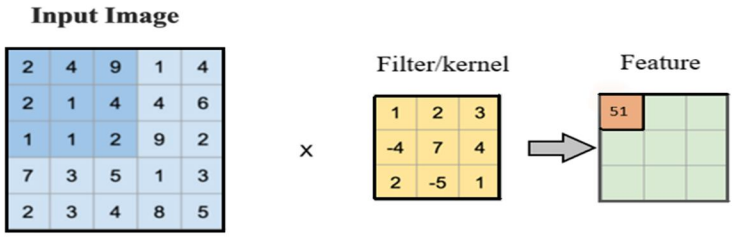


Fig. 5 Convolution Process Step 1

2*1 + 4*2 + 9*3 + 2*(-4) + 1*7 + 4*4 + 1*2 + 1*(-5) + 2*1 = 51

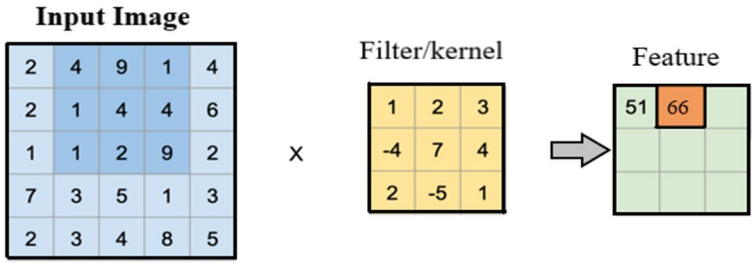


Fig.6 Convolution Process Step 2

4*1 + 9*2 + 1*3 + 1*(-4) + 4*7 + 4*4 + 1*2 + 2*(-5) + 9*1 = 66 and so on …

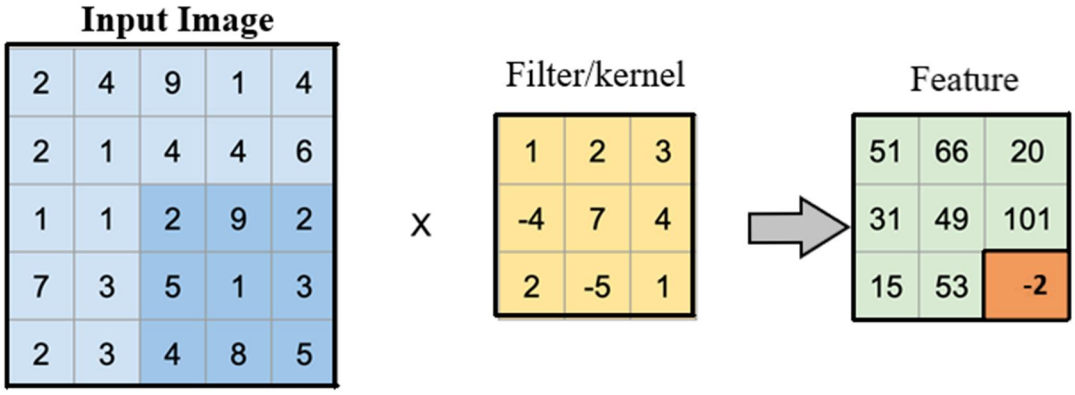


Fig .7 Convolution Operation Step — Final

2*1 + 9*2 + 2*3 + 5*(-4) + 1*7 + 3*4 + 4*2 + 8*(-5) + 5*1 = -2

The filter is applied systematically to each overlapping parts of the input date from left to right, top to bottom. Systematic application of same filter across the input data helps in discovering the specific type of features anywhere from the input.

2) *Strides:* Stride is a parameter which describe the number of shift the convolution filter moves at each step over the input matrix. If the stride is set to 1, the filter will move one pixel or unit at a time. We can increase the stride values to move the filter over the input with different intervals to extract different kinds of features.

In Fig 6. We set the stride value as 1 so we are moving the filter by one pixel

The equation to calculate the feature size is

Feature size= ((Input size – Kernel size)/stride) + 1

We can put the values and check for the above example

Feature size = ((5 –3) /1) + 1 = 3

If we set the stride value as 2 with filter size 3X3 on an image (input) of size 5X5 would be able to get a 2X2 feature matrix.

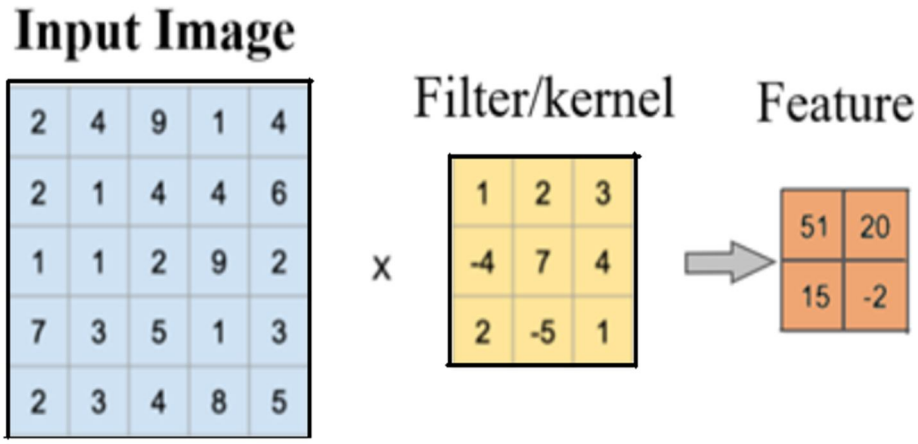


Fig. 8 Convolution Operation with Filter Size 3 and Stride 2

3) *Problem with Simple Convolution Layers:* For a gray scale (m x m) input image and (f x f) filter/kernel, the size of the image resulting from a convolution operation is

(m– f + 1) x (m – f + 1).

For example, for an (6 x 6) image and (2 x 2) filter, the outcome resulting after convolution operation would be of size (5 x 5). Thus, the image reduce every time a convolution operation is performed. This places an upper limit to the number of times such an operation could be performed before the image reduces to nothing thereby prevent us from building deeper networks.

Also, the pixels on the corners and the edges are used lot less than those in the middle.

4) *Padding:* The size of the featured map is always smaller than the input image, if we want to maintain something to prevent our feature map from shrinking, we use padding. In padding we add zeros to all sides of the image

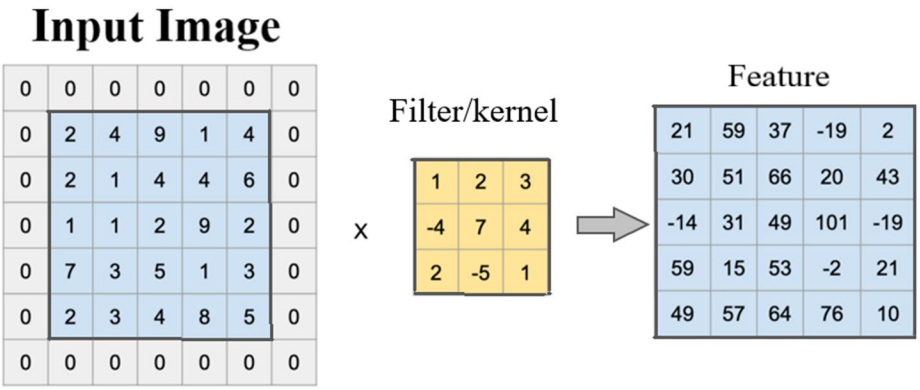


Fig. 9 Convolution Operation Kernel Size 3, Stride 1 and Padding 1

In the above figure we can see one more layer added to the 5 X 5 image considering padding =1, now it is converted to 6 X 6 image so more frame covers the edge pixel of the image, since more information is provided, we get more accurate features.
The equation to calculate the feature size considering a padded image size =1 is
Feature size = ((Image size + 2 * padding size – filter size)/stride) + 1
So by applying the value, Feature size = ((5 + 2 * 1 − 3) / 1) + 1= 5

B. Pooling Layer

Pooling Layer helps to reduce the size of features map by reducing the parameters and the number of computations performed in the network. The size of the pooling filter is smaller than the size of the features map.

1) *Max Pooling:* Max pooling is a pooling operation that chooses the topmost element from the region of the feature map covered by the filter. Thus, the outcome after max-pooling layer would be a feature map containing the most prominent features of the previous feature map.

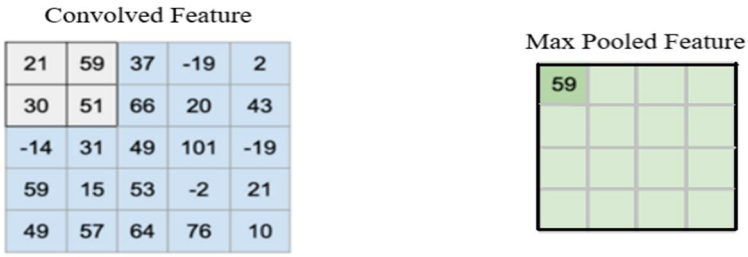


Fig. 10 Max Pooling Step 1

The above example shows Max pooling operation with a filter having size of 2 and stride of 1.

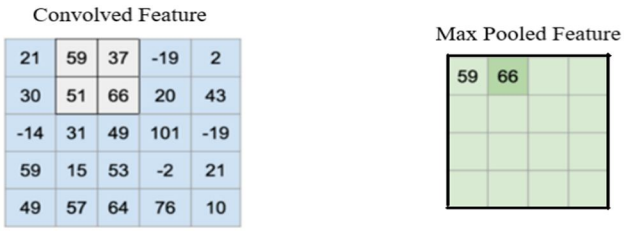


Fig. 11 Max Pooling Step 2 and So On....

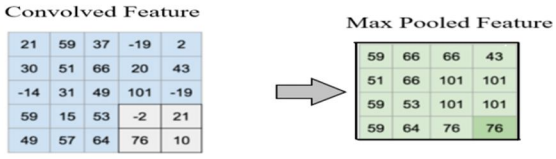


Fig. 12 Final Step

There is one more pooling called Average Pooling which takes the average value from the region of the feature map covered by the filter. Max pooling gives the most prominent feature by discarding unwanted noise and hence is better than average pooling.
The rectified linear activation function or ReLU for short is a piecewise linear function that will return the input directly if it is positive, otherwise, it will return zero. It has become the default activation function for many types of neural networks because a model that uses it is easier to train and often achieves better performance.[9]

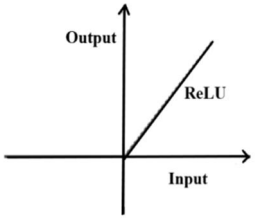


Fig 12. The Rectified Linear Unit (Relu) Function

C. Fully Connected Layer

Fully connected layer is also called Feed Forward Neural Network, the output from the final pooling is feed as an input to the Fully connected Layer (i.e.) the final pooling matrix values are flatten by unrolling all the values into its vector.

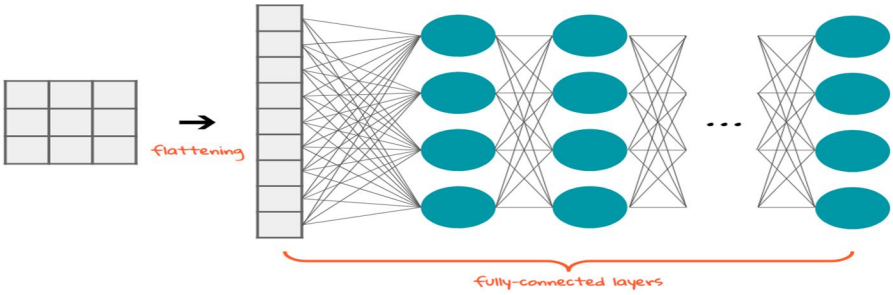


Fig. 13 Flattened vectors

These vectors are then connected to the fully connected network. We apply one or more fully connected layers at the last of a CNN. Adding a fully-connected layer help out non-linear combinations of the high-level feature outcomes by the convolutional layers.

1) *Soft Max:* After passing through the fully connected layer , Softmax activation function is used in the final layer to get the probabilities of input being in any particular classification.

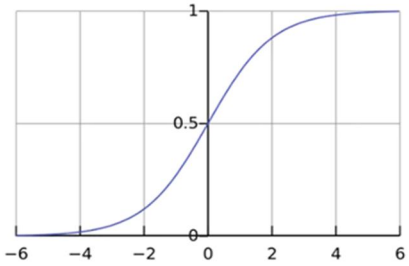


Fig. 16 Softmax Function

It is mainly used to normalize the CNN output to fit between zero and one. It is used to act for the certainty “probability” in the network output.

The normalization is formulated by dividing the exp value of the examined output by the summation of the exp value of each possible output.

$$\text{Softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$$

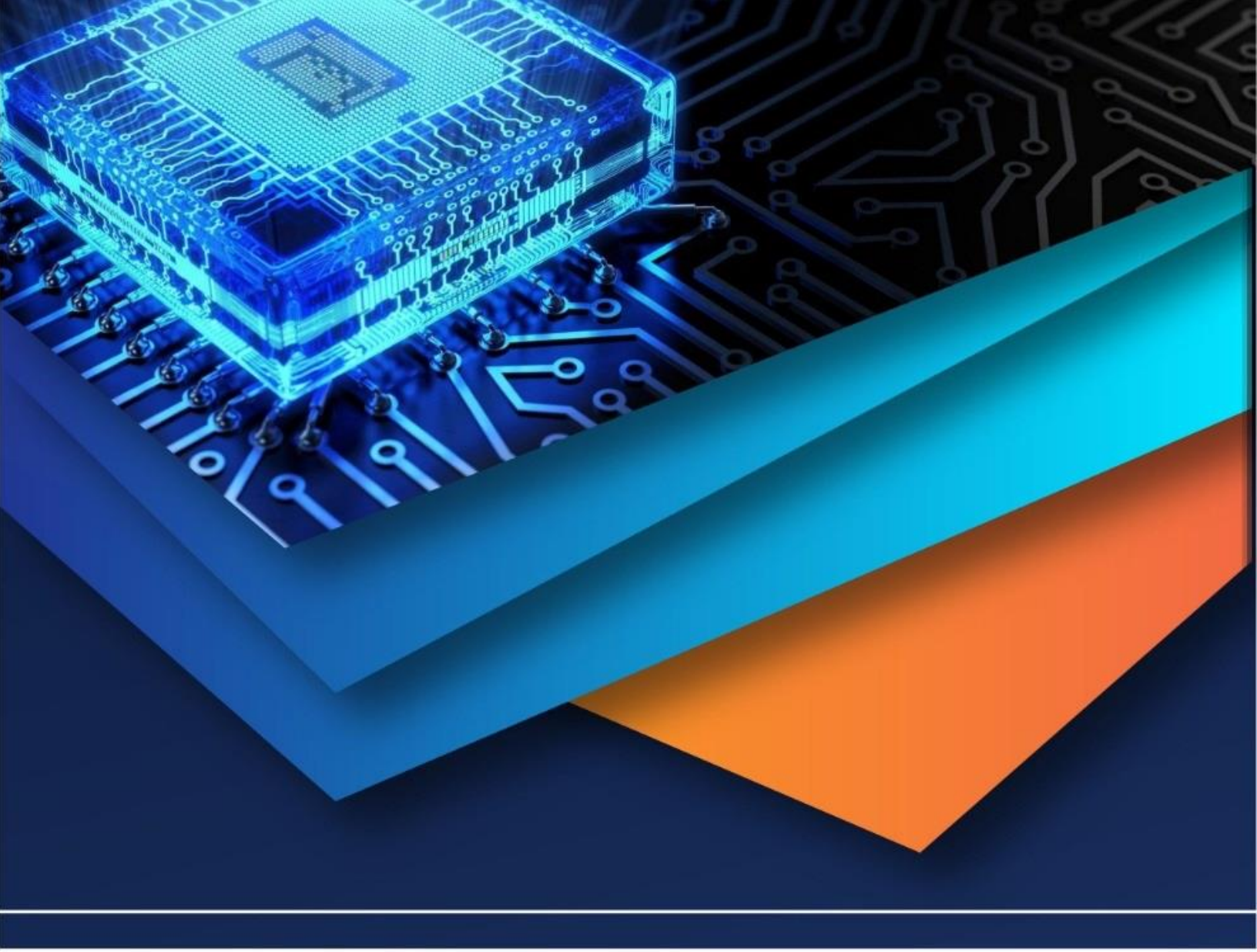
The Softmax Function is a generalization of the Logistic Function, and it makes sure that our forecast add up to 1.
Most of the time the Softmax Function is linked to the Cross Entropy Function. After the application of the Softmax Function in CNN, its duty is to check the dependability of the model using as Loss Function the Cross Entropy Function, in order to increase the performance of our neural network. There are various advantages to using the Cross Entropy Function. One of the best is that if for instance at the beginning of back propagation the output value is much smaller than the actual value, the gradient descent will be very slow. Because Cross Entropy uses the logarithm, it guides the network to assess even bigger errors.

IV. CONCLUSION

Major advantage of convolutional neural network in Deep Learning is that it can detect relevant features independently in high dimensional data's compared to other network. This study has given an overview of the basic understanding of CNN architecture which helps in identifying the correct input with the help of various undergoing process. The weakness of this model is very difficult to understand how much data/ layers required to complete a performance. CNN model works mainly based on the size and quality of the trained input. CNN performs extraordinary with well-prepared Dataset and are capable of surpassing human at visual recognition task. However, they are still not strong enough to visual artifacts such as glare and noise, which humans can able to perform. The concept of CNN is still under developing stage and researchers are working with the properties of CNN such as active attention and online learning allowing the network to evaluate new set of values which are different from what they are trained on.

REFERENCES

[1] www.infrrd.ai/blog/image-processing-with-deep-learning-a-quick-start-guide
[2] medium.com/@raycad.seedotech/convolutional-neural-network-cnn-8d1908c010ab
[3] ujjwalkarn.me/2016/08/09/quick-intro-neural-networks/
[4] towardsdatascience.com/convolution-neural-networks-a-beginners-guide-implementing-a-mnist-hand-written-digit-8aa60330d022
[5] towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2
[6] deepai.org/machine-learning-glossary-and-terms/stride
[7] machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/
[8] www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/
[9] machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/
[10] pubmed.ncbi.nlm.nih.gov/29934920/
[11] havasnewyork/IBM-Watson-Developer-Certification-Study-Guide
[12] www.metaculus.com/questions/5492/most-popular-img-classification-benchmark-22/
[13] maryambafandkar.me/a-quick-introduction-to-neural-networks/
[14] chatbots.marketing/a-gentle-introduction-to-convolutional-neural-networks/
[15] www.andreaperlato.com/aipost/cnn-and-softmax/



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY