Convolutional Neural Network: A Study

Dr. Rohan
Senior Software Engineer,
Ratna Global Technologies Pvt. Ltd.,

Abstract- *With the emergence of the Artificial Neural Network, the discipline of machine learning has undergone a drastic change recently (ANN). In popular machine learning tasks, these computational models with biological inspiration perform noticeably better than earlier iterations of artificial intelligence. One of the most striking ANN architecture designs is Convolutional Neural Network's is (CNN). CNNs are typically employed to tackle challenging image-driven pattern recognition problems and, thanks to their accurate yet straightforward architecture, provide a streamlined way to get started with ANNs.In this article, CNNs are briefly introduced, along with recently released papers and innovative methods for creating these amazingly outstanding image recognition models. This introduction presupposes that you are knowledgeable about the principles of ANNs*

Keywords- Artificial Neural Network, Machine Learning

I. INTRODUCTION

The operation of organic nervous systems, such as the human brain, is a major source of inspiration for artificial neural networks (ANNs), which are computational processing systems. The primary building block of ANNs is a large number of interconnected computational nodes (also known as neurons), which collaborate in a distributed manner to learn from the input in order to optimise its final result.

A model of an ANN's fundamental composition may be found in Figure 1. The input would be loaded into the input layer, which would then distribute it to the hidden layers. The input is typically in the form of a multidimensional vector. Following judgements from the preceding layer, the hidden layers will consider whether a stochastic shift inside themselves is detrimental or beneficial to the output.

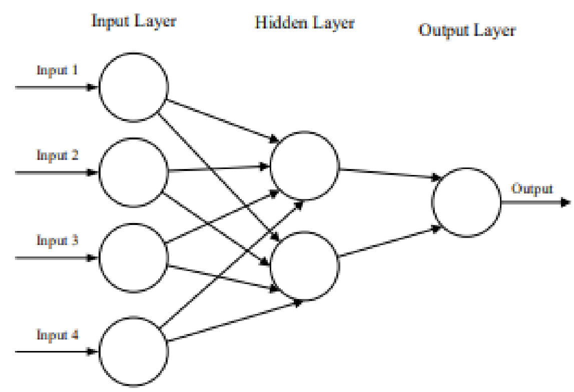

Fig 1: Simple three layered Neural Network

In tasks involving image processing, supervised and unsupervised learning are the two main learning paradigms. Learning with pre-labeled inputs that serve as targets is known as supervised learning. There will be a collection of training examples for each example input values (vectors) and one or more designated output values that are related to them.

Through accurate calculation of the output value of training example by training, this type of training aims to lower the models total classification error.

Unsupervised learning differs from supervised learning in that there are no labels in the training set. The ability of the network to decrease or enhance an associated cost function typically serves as a metric for success. The majority of image-focused pattern-recognition tasks, however, typically rely on classification employing supervised learning.

Similar to conventional ANNs, convolutional neural networks (CNNs) are made up of neurons that learn to optimise themselves. Each neuron will continue to take in information and carry out an operation (like a scalar).

The foundation of innumerable ANNs is a product followed by a non-linear function. The entire network will still express a single perceptual scoring function from the input raw picture vectors to the class score at the end (the weight).

The last layer will include loss functions related to the classes, and all of the standard techniques created for conventional ANNs are still applicable.

The fact that CNNs are largely employed in the field of pattern detection within images is the sole significant distinction between CNNs and conventional ANNs. This enables us to incorporate image-specific elements into the design, improving the network's suitability for image-focused tasks and lowering the number of setup parameters.

Traditional types of ANN have many drawbacks, one of which is that they frequently struggle when attempting to compute image data due to the processing complexity necessary.

Due to its comparatively low picture dimensionality of just 28x28, common machine learning benchmarking datasets like the MNIST database of handwritten digits are suitable for most types of ANN. Using this dataset, the first hidden layer of a single neuron will have 784 weights (28x28x1)

II. CNN ARCHITECTURE

CNNs place a lot of emphasis on the idea that the input will be made up of images. This concentrates the architecture's setup to best meet the requirements for handling the particular type of data. One of the main variations is that the layers of the CNN are made up of neurons arranged into three dimensions, the spatial dimensionality of the input (height and breadth) and the depth. The depth describes the third dimension of an activation volume rather than the total number of layers within the ANN. In contrast to normal ANNS, each layer's neurons only connect to a small portion of the layer before it.

2.1 Different Layers of CNN Architecture.

1. Input Layer
2. Convolution Layer
3. Pooling Layer
4. Fully Connected Layer
5. Activation functions

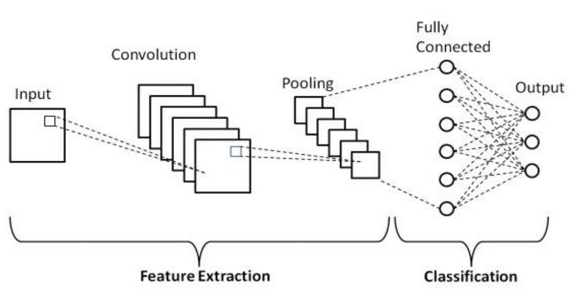

Fig.2 CNN Architecture

Input Layer: The input layer will store the image's pixel values, as with other ANN variants.

Convolution Layer: The foundation of a CNN is a convolutional layer. It has a number of filters (or kernels), whose settings must be learned over the course of training. Typically, the filters' size is smaller than the original image. Each filter produces an activation map after it convolves with the image. For convolution, the filter is slid over the image's height and breadth, and at each spatial location, the dot product between each filter element and the input is determined.

Utilizing the spatial local correlation of the input, the convolutional layer's local connection enables the network to train filters that respond maximally to a local region of the input (for an input image, a pixel is more correlated to the nearby pixels than to the distant pixels). In addition, because the filter and input are convolutioned to create the activation map, the filter parameters are shared across all local positions. The number of criteria for effective communication, effective learning, and effective generalisation are decreased through weight sharing.

These kernels often have a low spatial dimension yet cover the entire depth of the input. Each filter is convolved across the spatial dimensions of the input by the convolutional layer as the data reaches it, creating a 2D activation map.

The scalar product is calculated for each value in that kernel as we move through the input. In Figure 3. This will teach the network kernels that "fire" when they detect a particular feature at a specific spatial place in the input. These are frequently referred to as activations.

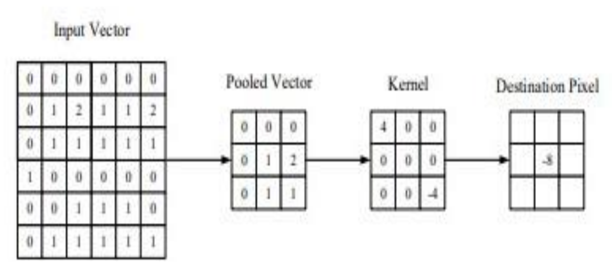

Fig.3. The input vector is placed over the kernel's central element, which is then calculated and replaced with a weighted sum of any surrounding pixels and itself.

Pooling Layer: Typically, a pooling layer is added in between two convolutional layers that follow one another. By downsampling the representation, the pooling layer lowers the computation's complexity and number of parameters. The average or maximum pooling function is available. Max pooling is frequently employed because it performs better.

There are only two commonly noted max-pooling strategies since the pooling layer is harmful. The pooling layers' stride and filters are often both set to 2 x2, which enables the layer to stretch throughout the input's full range of spatial dimensions. Additionally, overlapping pooling may be used with a stride of 2 and a kernel size of 3. A kernel size greater than 3 will typically cause the model's performance to suffer significantly because of the destructive nature of pooling.

Fully connected Layer: Neurons in the fully connected layer have direct connections to the neurons in the two adjacent layers; they are not connected to any neurons in those layers. This is comparable to how neurons are placed in conventional ANN models.

The input image from the layers below is flattened and supplied to the FC layer in this. The flattened vector is then put through a few additional FC layers, where the standard operations on mathematical functions happen. The classification procedure starts to take place at this point. Because two fully connected layers will function better than one connected one, two layers are connected. These CNN layers lessen the need for human oversight.

Activation Function: The activation function is one of the most crucial elements of the CNN model. They are employed to discover and approximation any type of continuous and complex link between network variables. In layman's terms, it determines which model information should shoot ahead and which should not at the network's end. The network gains nonlinearity as a result. The ReLU, Softmax, tanH, and Sigmoid functions are a few examples of regularly used activation functions. Each of these operations has a particular use. Sigmoid and softmax functions are preferred for a CNN model for binary classification, and softmax is typically employed for multi-class classification. operations in mathematics.

III. IMAGE AUGMENTATION IN CNN

By modifying the existing data, image augmentation creates new data that can be used for model training. In other words, it is the process of enhancing the dataset that is made available for deep learning model training.

```
#Importing the reuired libraries
from keras.preprocessing.image import ImageDataGenerator
from keras.preprocessing.image import load_img
from keras.preprocessing.image import img_to_array
from matplotlib import pyplot
import matplotlib.pyplot as plt
from numpy import expand_dims

# Loading the image
image1 = load_img("G:\\2.jpg")
image1
image1.size

data = img_to_array(image1)
samples = expand_dims(data, 0)
datagen = ImageDataGenerator(rotation_range= 90,
                             width_shift_range= 0.2,
                             height_shift_range= 0.2,
                             zoom_range= 0.5,
                             horizontal_flip=True,
                             brightness_range=[0.5, 1.5],
                            fill_mode='nearest')

it = datagen.flow(samples, batch_size=1)

fig, ax = plt.subplots(nrows=1, ncols=3, figsize=(15,15))
for i in range(6):
    pyplot.subplot(330 + 1 + i)
    batch = it.next()
    image = batch[0].astype('uint8')
    pyplot.imshow(image)
pyplot.show()
```

Fig.4 Code for Image augmentation (rotation).

Fig. 5 Original Image.

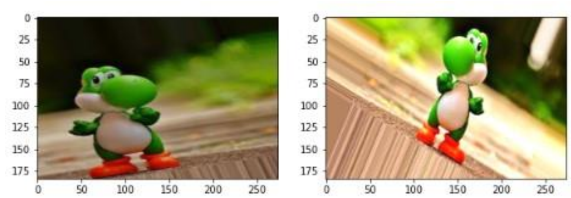

Fig. 6 Augmented Image

IV. CONCLUSION

Convolutional neural networks are different from other types of artificial neural networks in that they focus on a particular sort of input rather than the full issue area. This then makes it possible to put up a much simpler network design.

The fundamental ideas of convolutional neural networks have been explained in this paper, along with an explanation of the layers needed to construct one and recommendations for how to arrange the network for the majority of image processing tasks.

Recent years have seen a slight slowdown in the study of neural networks for image processing. This is in part because people have the wrong idea about how complicated and knowledgeable it is to start modelling these incredibly potent machine learning algorithms.

REFERENCES

[1] Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Computer Vision–ECCV 2014, pp. 818–833. Springer (2014)
[2] Szegedy, C., Toshev, A., Erhan, D.: Deep neural networks for object detection. In: Advances in Neural Information Processing Systems. pp. 2553–2561 (2013)
[3] Cires¸an, D.C., Giusti, A., Gambardella, L.M., Schmidhuber, J.: Mitosis detection in breast cancer histology images with deep neural networks. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2013, pp. 411–418. Springer (2013)
[4] Zeiler, M.D., Fergus, R.: Stochastic pooling for regularization of deep convolutional neural networks. arXiv preprint arXiv:1301.3557 (2013) .
[5] Tivive, F.H.C., Bouzerdoum, A.: A new class of convolutional neural networks (siconnets) and their application of face detection. In: Neural Networks, 2003. Proceedings of the International Joint Conference on. vol. 3, pp. 2157–2162. IEEE (2003)
[6] Egmont-Petersen, M., de Ridder, D., Handels, H.: Image processing with neural networksa review. Pattern recognition 35(10), 2279–2301 (2002)
[7] LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Backpropagation applied to handwritten zip code recognition. Neural computation 1(4), 541–551 (1989)