Basic CNN Architecture: Explaining 5 Layers of Convolutional Neural Network

by MK Gurucharan

Gurucharan M K, Undergraduate Biomedical Engineering Student | Aspiring AI engineer | Deep Learning and Machine Learning Enthusiast

DEC 7, 2020

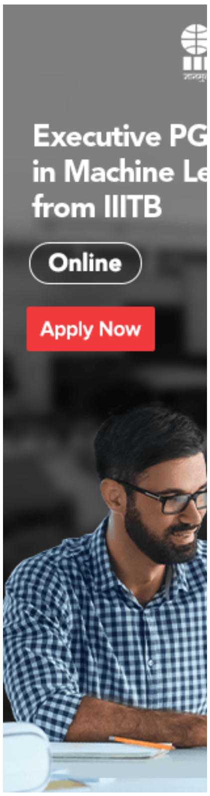
Home > Artificial Intelligence > Basic CNN Architecture: Explaining 5 Layers of Convolutional Neural Network

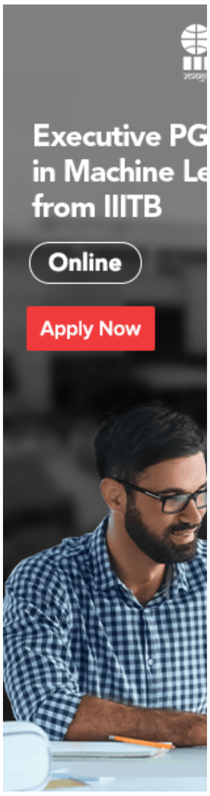
Table of Contents

Introduction
Basic Architecture
  Convolution Layers
  1. Convolutional Layer
  2. Pooling Layer
  3. Fully Connected Layer
  4. Dropout
  5. Activation Functions
LeNet-5 CNN Architecture
Conclusion
What are activation functions in CNN?
What are the basic components of the convolutional neural network architecture?
What is the benefit of standard CNN architectures?

Introduction

In the last few years of the IT industry, there has been a huge demand parti skill set known as Deep Learning. Deep Learning a subset o





These structures are called as Neural Networks. It teaches the computer to do what naturally comes to humans. Deep learning, there are several types of models such as the Artificial Neural Networks (ANN), Autoencoders, Recurrent Neural Networks (RNN) and Reinforcement Learning. But there has been one particular model that has contributed a lot in the field of computer vision and image analysis which is the Convolutional Neural Networks (CNN) or the ConvNets.

CNNs are a class of Deep Neural Networks that can recognize and classify particular features from images and are widely used for analyzing visual images. Their applications range from image and video recognition, image classification, medical image analysis, computer vision and natural language processing.

The term 'Convolution" in CNN denotes the mathematical function of convolution which is a special kind of linear operation wherein two functions are multiplied to produce a third function which expresses how the shape of one function is modified by the other. In simple terms, two images which can be represented as matrices are multiplied to give an output that is used to extract features from the image.

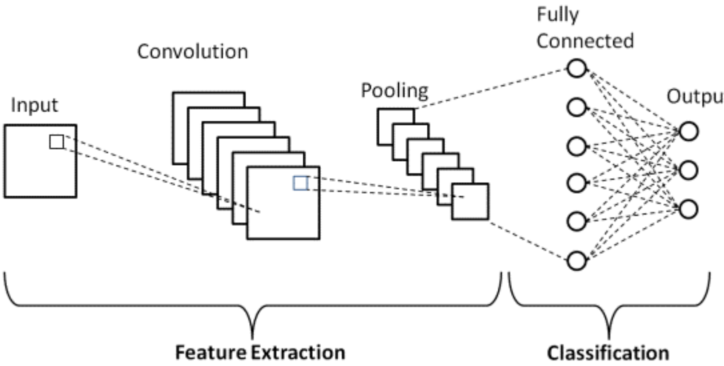
Learn Machine Learning online from the World's top Universities – Masters, Executive Post Graduate Programs, and Advanced Certificate Program in ML & AI to fast-track your career.

Learn: Introduction to Deep Learning & Neural Networks

Basic Architecture

There are two main parts to a CNN architecture

- A lution tool that separates and identifies the various features for analysis in a process called as Feature Extraction



[**Source**]

Convolution Layers

There are three types of layers that make up the CNN which are the convolutional layers, pooling layers, and fully-connected (FC) layers. When these layers are stacked, a CNN architecture will be formed. In addition to these three layers, there are two more important parameters which are the dropout layer and the activation function which are defined below.

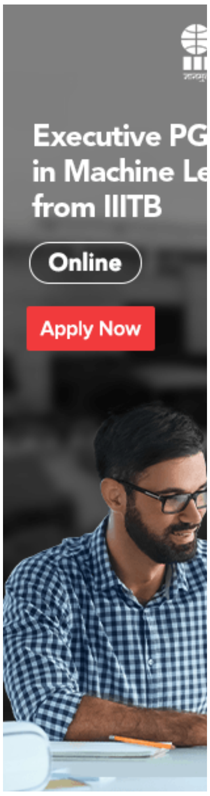
1. Convolutional Layer

This layer is the first layer that is used to extract the various features from the input images. In this layer, the mathematical operation of convolution is performed between the input image and a filter of a particular size MxM. By sliding the filter over the input image, the dot product is taken between the filter and the parts of the input image with respect to the size of the filter (MxM).

The output is termed as the Feature map which gives us information about the image such as the corners and edges. Later, this feature map is fed to other layers to learn several other features of the input image.

2. Pooling Layer

In most cases, a Convolutional Layer is followed by a Pooling Layer. The primary aim of this layer is to decrease the size of the convolved feature map to reduce the computational costs. This is performed by decreasing the connections between layers and independently operates on each feature map. Depending [illegible] used, there are several types of Pooling operations.



sum of the elements in the predefined section is computed in Sum Pooling. The Pooling Layer usually serves as a bridge between the Convolutional Layer and the FC Layer

Must Read: [Neural Network Project Ideas]

3. Fully Connected Layer

The Fully Connected (FC) layer consists of the weights and biases along with the neurons and is used to connect the neurons between two different layers. These layers are usually placed before the output layer and form the last few layers of a CNN Architecture.

In this, the input image from the previous layers are flattened and fed to the FC layer. The flattened vector then undergoes few more FC layers where the mathematical functions operations usually take place. In this stage, the classification process begins to take place.

4. Dropout

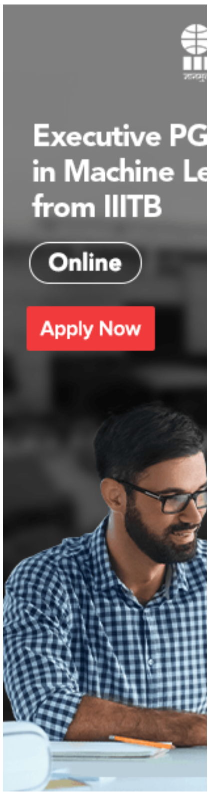
Usually, when all the features are connected to the FC layer, it can cause overfitting in the training dataset. Overfitting occurs when a particular model works so well on the training data causing a negative impact in the model's performance when used on a new data.

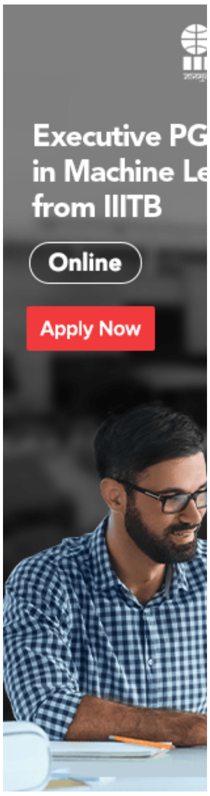
To overcome this problem, a dropout layer is utilised wherein a few neurons are dropped from the neural network during training process resulting in reduced size of the model. On passing a dropout of 0.3, 30% of the nodes are dropped out randomly from the neural network.

5. Activation Functions

Finally, one of the most important parameters of the CNN model is the activation function. They are used to learn and approximate any kind of continuous and complex relationship between variables of the network. In simple words, it decides which information of the model should fire in the forward direction and which ones should not at the end of the network.

It adds non-linearity to the network. There are several commonly usec func such as the ReLU, Softmax, tanH and the Sigmoid functions func have a specific usage. For a binary classification CNN model,





LeNet-5 CNN Architecture

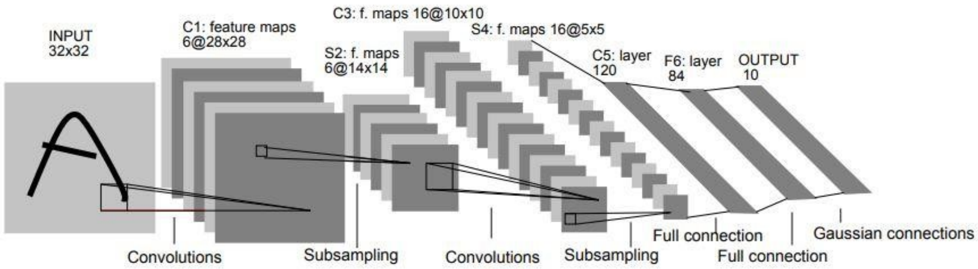
In 1998, the LeNet-5 architecture was introduced in a research paper titled "Gradient-Based Learning Applied to Document Recognition" by Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. It is one of the earliest and most basic CNN architecture.

It consists of 7 layers. The first layer consists of an input image with dimensions of 32×32. It is convolved with 6 filters of size 5×5 resulting in dimension of 28x28x6. The second layer is a Pooling operation which filter size 2×2 and stride of 2. Hence the resulting image dimension will be 14x14x6.

Similarly, the third layer also involves in a convolution operation with 16 filters of size 5×5 followed by a fourth pooling layer with similar filter size of 2×2 and stride of 2. Thus, the resulting image dimension will be reduced to 5x5x16.

Once the image dimension is reduced, the fifth layer is a fully connected convolutional layer with 120 filters each of size 5×5. In this layer, each of the 120 units in this layer will be connected to the 400 (5x5x16) units from the previous layers. The sixth layer is also a fully connected layer with 84 units.

The final seventh layer will be a softmax output layer with 'n' possible classes depending upon the number of classes in the dataset.

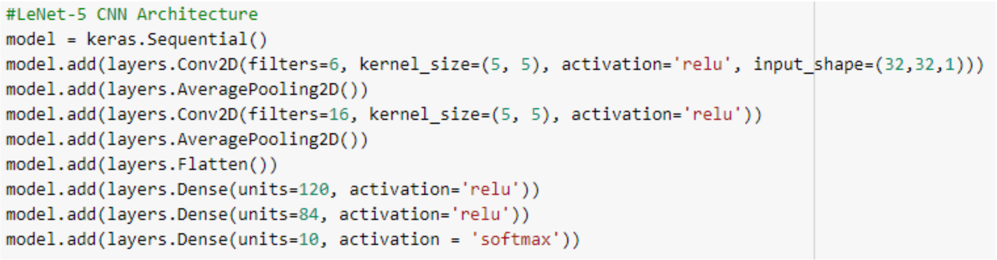


Source

The above diagram is a representation of the 7 layers of the LeNet-5 CNN Architecture.

Below are the snapshots of the Python code to build a LeNet-5 CNN a

usin s library with TensorFlow framework

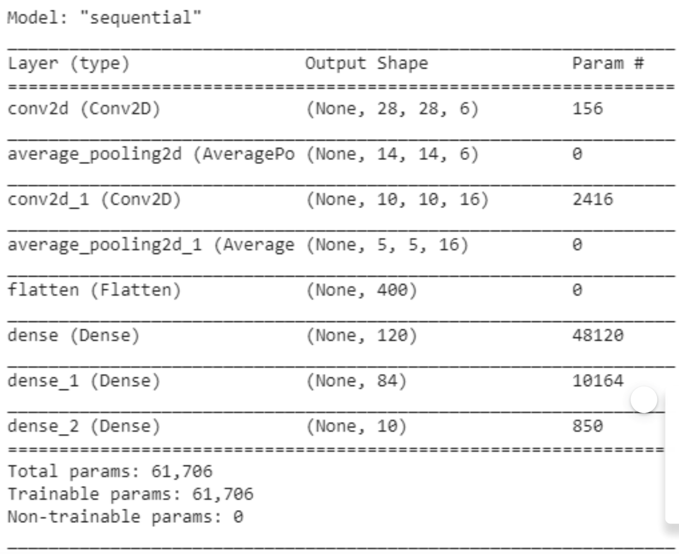


```
#LeNet-5 CNN Architecture
model = keras.Sequential()
model.add(layers.Conv2D(filters=6, kernel_size=(5, 5), activation='relu', input_shape=(32,32,1)))
model.add(layers.AveragePooling2D())
model.add(layers.Conv2D(filters=16, kernel_size=(5, 5), activation='relu'))
model.add(layers.AveragePooling2D())
model.add(layers.Flatten())
model.add(layers.Dense(units=120, activation='relu'))
model.add(layers.Dense(units=84, activation='relu'))
model.add(layers.Dense(units=10, activation = 'softmax'))
```

In Python Programming, the model type that is most commonly used is the Sequential type. It is the easiest way to build a CNN model in keras. It permits us to build a model layer by layer. The 'add()' function is used to add layers to the model. As explained above, for the LeNet-5 architecture, there are two Convolution and Pooling pairs followed by a Flatten layer which is usually used as a connection between Convolution and the Dense layers.

The Dense layers are the ones that are mostly used for the output layers. The activation used is the 'Softmax' which gives a probability for each class and they sum up totally to 1. The model will make it's prediction based on the class with highest probability.

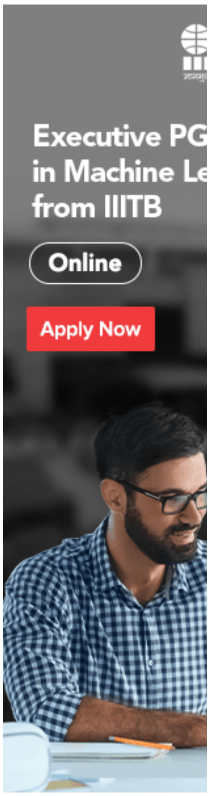
The summary of the model is displayed as below.



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 28, 28, 6)         156
_________________________________________________________________
average_pooling2d (AveragePo (None, 14, 14, 6)         0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 10, 10, 16)        2416
_________________________________________________________________
average_pooling2d_1 (Average (None, 5, 5, 16)          0
_________________________________________________________________
flatten (Flatten)            (None, 400)               0
_________________________________________________________________
dense (Dense)                (None, 120)               48120
_________________________________________________________________
dense_1 (Dense)              (None, 84)                10164
_________________________________________________________________
dense_2 (Dense)              (None, 10)                850
=================================================================
Total params: 61,706
Trainable params: 61,706
Non-trainable params: 0
_________________________________________________________________
```

Hence, in this article we have understood the basic CNN structure, it's architecture and the various layers that make up the CNN model. Also, we have seen an architectural example of a very famous and traditional LeNet-5 model with its Python program.

If you're interested to learn more about **machine learning courses,** check out IIIT-B & upGrad's [Executive PG Programme in Machine Learning & AI] which is designed for working professionals and offers 450+ hours of rigorous training, 30+ case studies & assignments, IIIT-B Alumni status, 5+ practical hands-on capstone projects & job assistance with top firms.



What are activation functions in CNN?

The activation function is one of the most vital components in the CNN model. They're utilized to learn and approximate any form of network variable-to-variable association that's both continuous and complex. In simple terms, it determines which model information should flow in the forward direction and which should not at the network's end. It gives the network non-linearity. The ReLU, Softmax, tanH, and Sigmoid functions are some of the most often utilized activation functions. All of these functions have distinct uses. For a 2-class CNN model, sigmoid and softmax functions are favored, whereas softmax is typically employed for multi-class classification.

What are the basic components of the convolutional neural network architecture?

An input layer, an output layer, and multiple hidden layers make up convolutional networks. The neurons in the layers of a convolutional network are arranged in three dimensions, unlike those in a standard neural network (width, height, and depth dimensions). This enables the CNN to convert a three-dimensional input volume into an output volume. Convolution, pooling, normalizing, and fully connected layers make up the hidden layers. Multiple conv layers are used in CNNs to filter input volumes to higher levels of abstraction.

What is the benefit of standard CNN architectures?

While traditional network architectures consisted solely of stacked convolutional layers, newer architectures look into new and novel ways of constructing convolutional layers in order to improve learning efficiency. These architectures provide general architectural recommendations for machine learning practitioners to adapt in order to handle a variety of computer vision problems. These architectures can be utilized as rich feature extractors for image classification, object identification, picture segmentation, and a variety of other advanced tasks.