Convolution neural networks

Artificial Intelligence (AI) has grown tremendously in recent years, helping to close the technological divide between people and robots. Researchers as well as amateurs work on a variety of different facets of the field in order to bring about fantastic results. The field of Computer Vision is just one example of many.
If we can give machines the same vision as humans, they will be able to perform a wide range of functions, such as image and video recognition, image analysis and classification, media re-creation (including emulation of human speech), recommendation systems (including translation), and natural language processing (including transcription). Since the invention of the Convolutional Neural Network, advances in Computer Vision using Deep Learning have been built and developed over time.

Introduction

With a Convolutional Neural Network (ConvNet/CNN), an image is fed into a machine learning algorithm that uses weights and biases to determine the relevance of different characteristics and objects in the image. When compared to other classification methods, the amount of pre-processing required by a ConvNet is significantly reduced. While crude approaches rely on hand-engineered filters, ConvNets are capable of picking up on these traits with enough training.

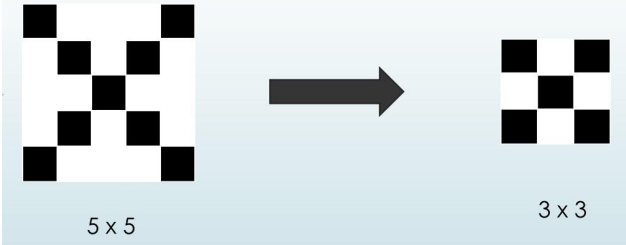
As in the human brain's connectivity pattern, a ConvNet's architecture is influenced by the Visual Cortex's arrangement. The Receptive Field is a small portion of the visual field where individual neurons respond to stimuli. A large number of these fields can be stacked on top of one another to fill the full visual field.

Convolution

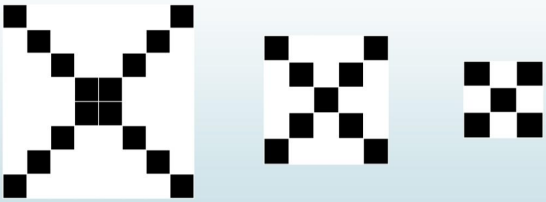
Functional analysis uses convolution to show how one function's shape changes when another one is added to it. In mathematics, this is known as a convolution on two functions (f and g). Convolution is used to describe both the final result and the computation process. The essential idea is that there is statistical stationarity in all of the images. The mean, variance, and autocorrelation structure do not change over time in a stationary process. To be mathematically exact, stationarity is a series that has no trend, a constant variance over time, a consistent autocorrelation structure across time, and no periodic fluctuations.

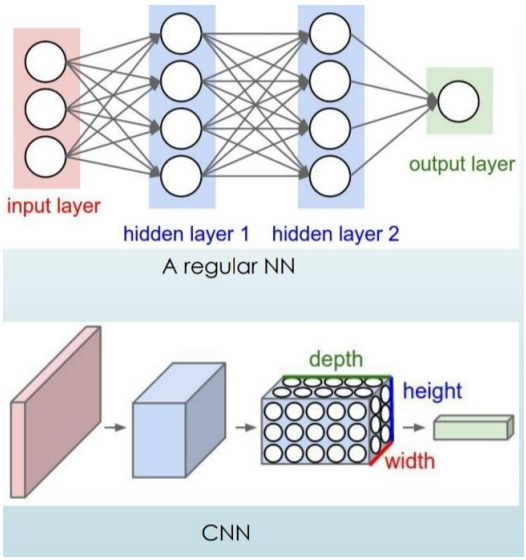
Let's take a look back at something...

- Neural Networks: The Standard Method: An input (a single vector) is passed through a number of hidden layers by Neural Networks, as we have seen. All of the neurons in a layer are totally connected to all the neurons in the layer below them, and the neurons in a single layer operate completely independently of one another and share zero connections.
- The "output layer," the final and most linked layer, represents the class scores in classification settings.
- Regular neural networks have the drawback of not scaling well to images.
- The great concept is that we don't require a big number of categorization factors.



Calculate the total number of parameters



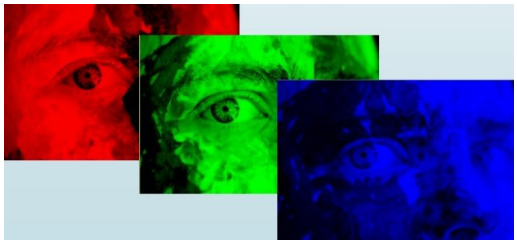


Let's consider a step by step example

Look at the picture below

Specifications of this picture

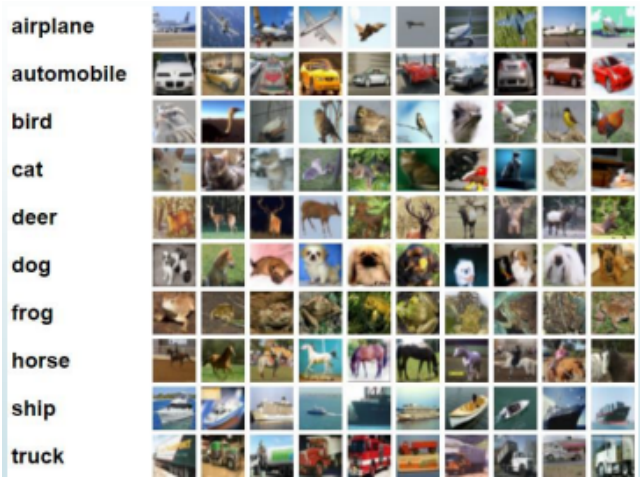
Colorful.jpeg
Dimensions: 525 x 350 (Width : 525 pixels | Height : 350 pixels)
Resolution : 75 dpi
24 bit color channel
34,203 byte

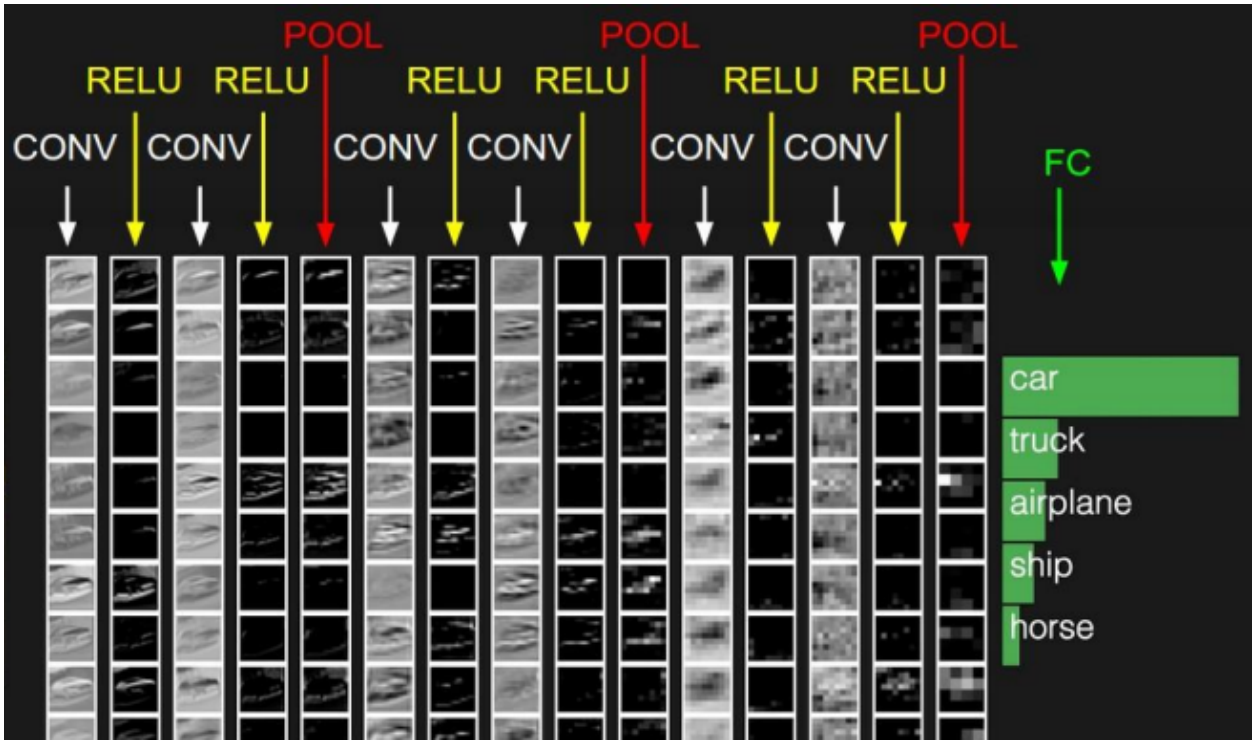


Layers utilised in the construction of convolution networks

Layers used to construct ConvNet architecture include the Convolutional, Pooling, and Fully-Connected types of layers.

Let's take example of Cifar-10 dataset as a sample case study.The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images





- To keep the image's raw pixel values, use INPUT [32x32x3]. In this case, the image has three colour channels: R,G,B.
- Using the CONV layer, the output of neurons connected to small input volume regions will be computed, with each neuron computing a dot product between its weights and the small input volume region to which it is attached. If we chose to employ 12 filters, the resulting volume may be [32x32x12].

- Elements will be activated one at a time by the RELU layer using a function such as the max(0,x) thresholding. As a result, the volume size remains the same ([32x32x12])

- POOL layer will perform a downsampling operation along the spatial dimensions (width, height), resulting in volume such as [16x16x12]

- Fully-connected FC layer will compute class scores, resulting in a volume of size [1x1x10].

Convolution

A Convolutional Network's fundamental building piece, the Conv layer does the bulk of the computation.

Each filter is slid over the width and height of the input volume (more accurately, convolved) and dot products are calculated between the filter entries and the input at any place during forward pass. (5 x 5 x 3)

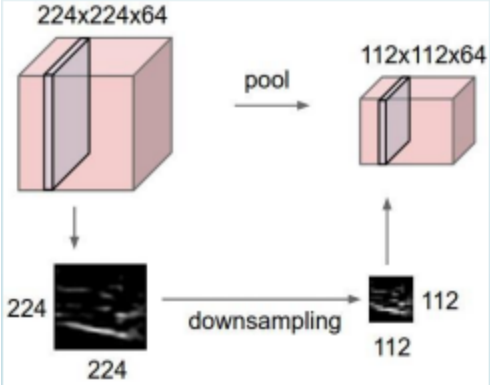
A two-dimensional activation map will be generated when we move the filter over the input volume, showing the filter's reactions at various spatial locations.

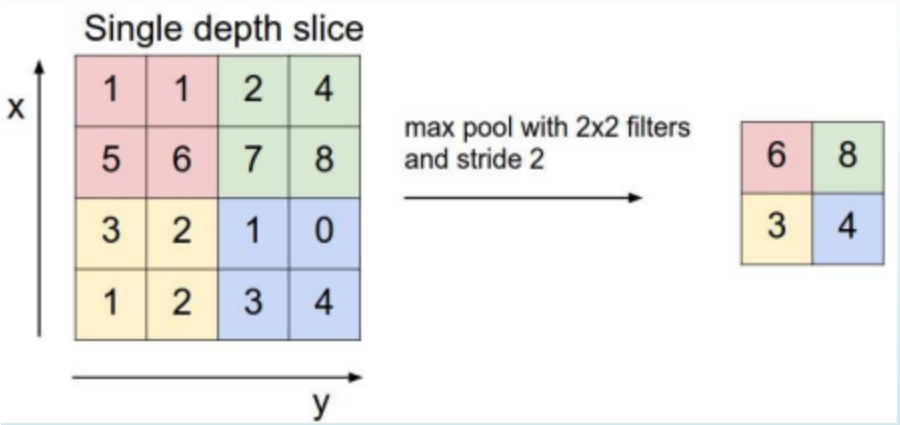
When a visual element such as an edge or a splotch appears on the first layer, the network intuitively learns filters that activate. As the network layers rise, whole honeycomb or wheel-like patterns may appear.

Pooling

In a ConvNet architecture, it is usual practise to insert a Pooling layer between subsequent Conv layers. As the representation's spatial size shrinks, it reduces the number of parameters and computations in the network, which helps keep overfitting in check.

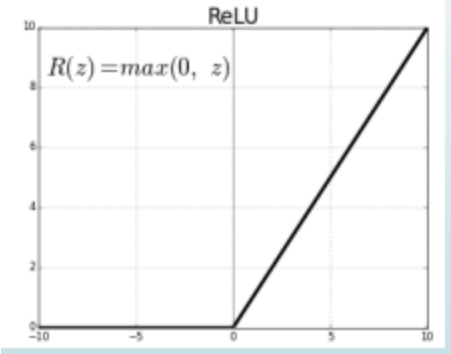
The Pooling Layer uses the MAX operation to enlarge the input spatially based on each depth slice independently.





ReLu Layer

To Output real values and stop them till zero.This does introduce a non-linearity in the process.



Applications of CNN

- Face recognition
- Character recognition
- Object Recognition
- Object Segmentation
- Pose Detection
- GeoSpatial Terrain observation
- Forest Fire Spreading
- Flood area segmentation
- Damage assessment
- Almost most of the data has Statistical Stationarity

Instead of using Feed-Forward Neural Networks, why not use ConvNets?

Isn't an image nothing more than a pixel value matrix? As a result, why not just use a Multi-Level Perceptron to classify the image instead of flattening it? Not really, I'm afraid. When dealing with simple binary images, the approach may display an average precision score when predicting classes, but when dealing with complicated images with pixel dependencies all over, it will have little to no accuracy.

Because of its ability to apply relevant filters, a ConvNet may successfully capture an image's spatial and temporal dependencies. Because of the reduced number of parameters and reusability of weights, the architecture better fits the picture dataset. To put it another way, the network can be taught to better understand the complexities of an image.