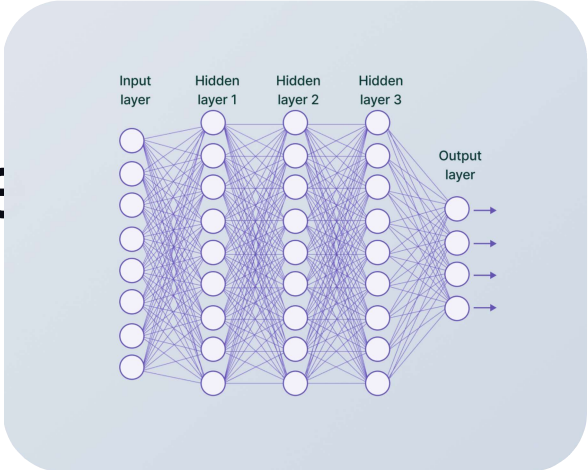
BLOG DEEP LEARNING

A Comprehensive Guide to Convolutional Neural Networks



What is a Convolutional Neural Network and how does it work? Learn about the history of CNNs and the most popular Convolutional Networks Architectures.

9 min read · June 27, 2021


Pragati Baheti
Microsoft

Artificial Intelligence aims at giving machines the capability to think and act like humans.

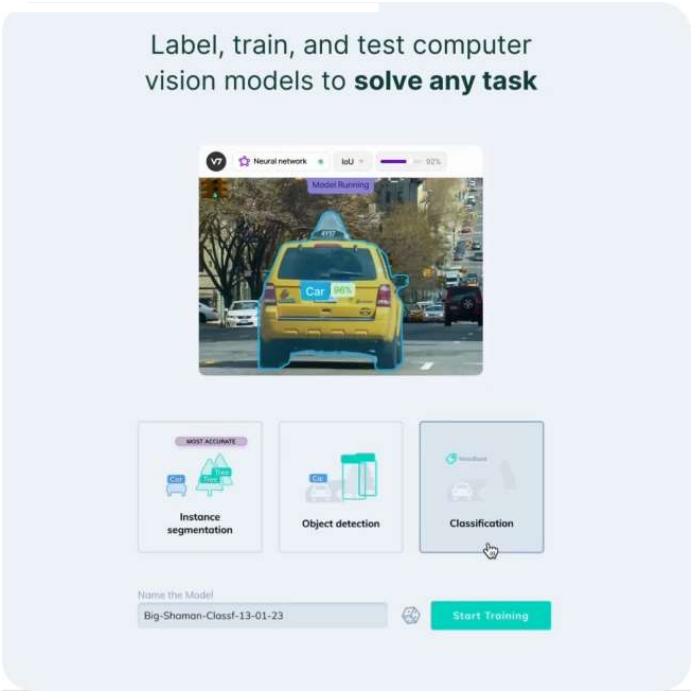
Some domains like computer vision or natural language processing require that machines exchange the naive approach of

feature extraction and learning for thinking outside the box.

Giving machines the sense of sight—the power to see and explore the world and use it in fields like image analysis, object detection, or medical image processing created the fundamentals for developing a new Deep Learning algorithm called **Convolution Neural Network**.

Here's what we'll cover:

Let's jump right into it.



Try V7 Now

Don't start empty-handed. Explore our repository of 500+ open datasets and test-drive V7's tools.

Ready to streamline AI product deployment right away? Check out:

- V7 Model Training
- V7 Workflows
- V7 Auto Annotation
- V7 Dataset Management

What is a Convolutional Neural Network?

Convolution neural network (also known as *ConvNet or CNN*) is a type of feed-forward neural network used in tasks like image analysis, natural language processing, and other complex image classification problems.

It is unique in that it can pick out and detect patterns from images and text and make sense of them. We will explore this more in-depth later in the article. However—

Before diving deeper into this topic, let's take a step back and understand the origin of the Convolutional Neural Network (CNN).

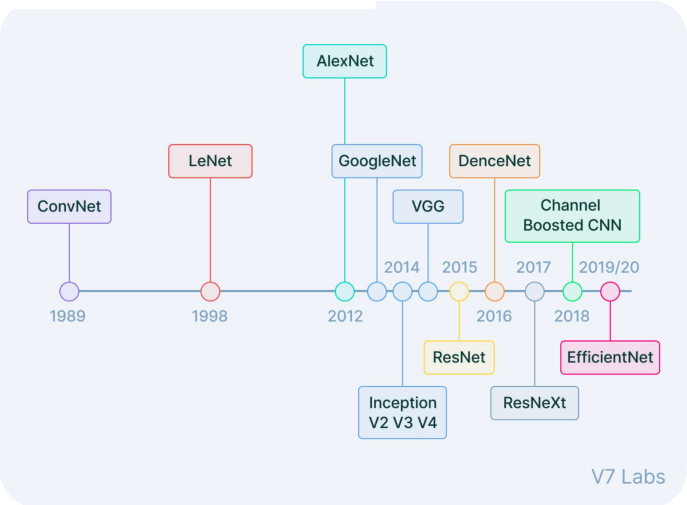
A brief history of Convolutional Neural Networks

We will start by analyzing different neural nets architectures emphasizing exactly *why* they were introduced and *what* new concepts they bring to the field that make CNN what it is today.

Let's begin with Yann LeCun's pioneering paper in 1998 in which he introduced a class of neural network architecture—LeNet that is one of the most common forms that we encounter today.

Before GPUs came into use, computers were not able to process large amounts of image data within a reasonable time, and the training was performed on images with low resolutions only.

That is why neural networks didn't spark until 2010.



Back in the day, data scientists assumed that a better algorithm would always yield better results regardless of data, but we know today that this theory is flawed.

We've come to understand that the training-validation-testing dataset should reflect the real world. This realization led to the mapping of the entire world of objects into a dataset named ImageNet.

In 2012, AlexNet architecture was introduced, consisting of five convolutional layers and three fully connected layers, plus the ReLU activation function was introduced for the first time in ConvNet.

Convolutional Neural Networks Architecture: An Overview

Now, let's explore the core building blocks for a Neural Networks Architecture.

ConvNet vs. Feed-Forward Neural Nets

You might be wondering what went wrong in feedforward networks that ConvNet later rectified.

Let's try to answer this :)

A computer sees an image as a matrix of numbers with (rows*columns*number of channels) shape. Any real-world image would be at least 200*200*3 pixels.

So, the question is: *Can we just flatten the image into one long 1D matrix?*

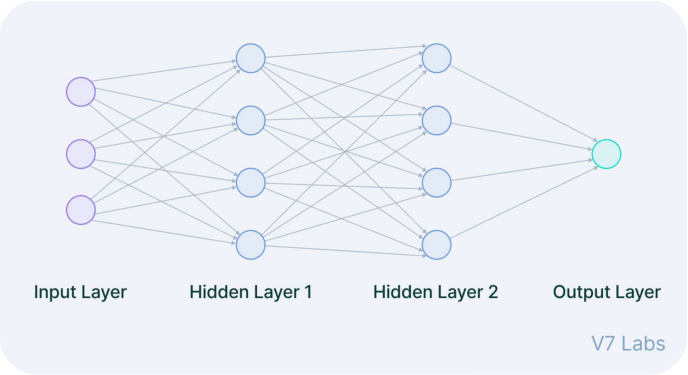
No... Not really!

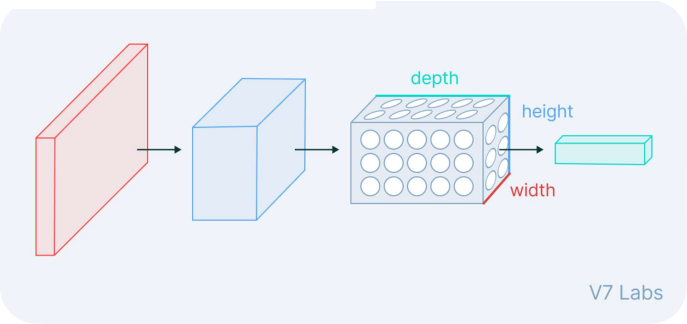
Here's why—

The neural nets that accept this long array of numbers must contain a large number of neurons. The number of weights required at the first hidden layer itself will be 20,000.

Dealing with such a huge amount of parameters requires many neurons and it may lead to overfitting.

In contrast to feedforward neural networks, convolutional neural networks look at one patch of an image at a time and move forward in this manner to derive complete information. It involves very few neurons with fewer parameters to scan an entire image to learn essential features.





CNNs Layers

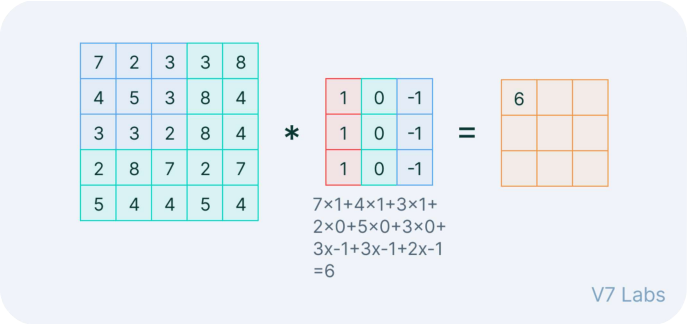
Here's an overview of layers used to build Convolutional Neural Network architectures.

Convolutional Layer

CNN works by comparing images piece by piece.

Filters are spatially small along width and height but extend through the full depth of the input image. It is designed in such a manner that it detects a specific type of feature in the input image.

In the convolution layer, we move the filter/kernel to every possible position on the input matrix. Element-wise multiplication between the filter-sized patch of the input image and filter is done, which is then summed.

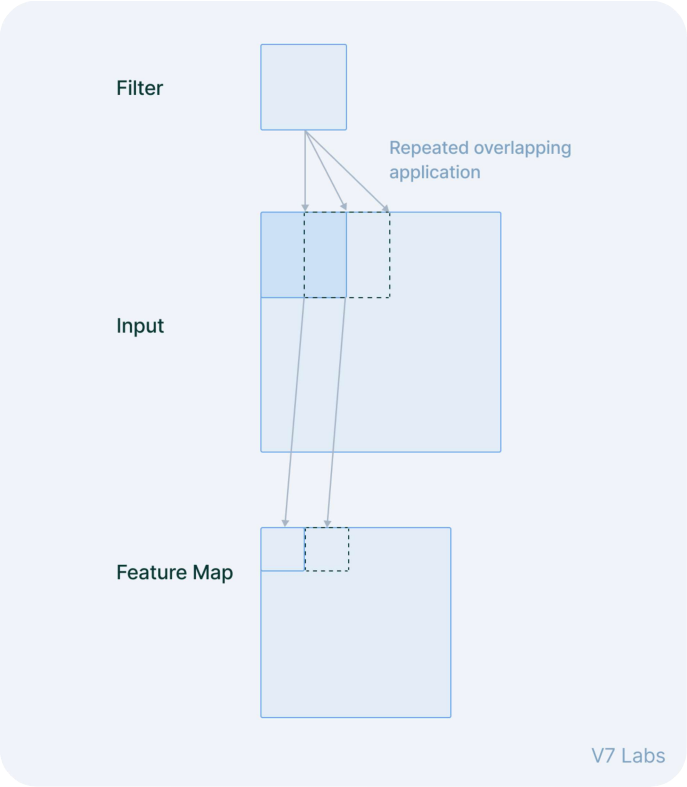


The translation of the filter to every possible position of the input matrix of the image

gives an opportunity to discover that feature is present anywhere in the image.

The generated resulting matrix is called the *feature map*.

Convolution neural networks can learn from multiple features parallelly. In the final stage, we stack all the output feature maps along with the depth and produce the output.



Now, let's go over a few important terms that you might encounter when learning about Convolutional Neural Networks.

Local connectivity refers to images represented in a matrix of pixel values. The dimension increases depending on the size of the image. If all the neurons are connected to all previous neurons as in a fully connected layer, the number of parameters increases manifold.

To resolve this, we connect each neuron to only a patch of input data. This spatial extent (also known as *the receptive field of the neuron*) determines the size of the filter.

Here's how it works in practice—

Suppose we have an input image is of size 128*128*3. If the filter size is 5*5*3 then each neuron in the convolution layer will have a total of 5*5*3 = 75 weights (and +1 bias parameter).

Spatial arrangement governs the size of the neurons in the output volume and how they are arranged.

Three hyperparameters that control the size of the output volume:

the image. The output volume has stacked activation/feature maps along with the depth, making it equal to the number of filters used.

- **Stride** - Stride refers to the number of pixels we slide while matching the filter with the input image patch. If the stride is one, we move the filters one pixel at a time. Higher the stride, smaller output volumes will be produced spatially.
- **Zero-padding—**It allows us to control the spatial size of the output volume by padding zeros around the border of the input data.

Parameter Sharing means that the same weight matrix acts on all the neurons in a particular feature map—the same filter is applied in different regions of the image.

Neural Networks: Summary

Natural images have statistical properties, one being invariant to translation.

For example, an image of a cat remains an image of a cat even if it is translated one pixel to the right—CNNs take this property into account by sharing parameters across multiple image locations. Thus, we can find a cat with the same feature matrix whether the cat appears at column i or column i+1 in the image.

ReLU Layer

In this layer, the ReLU activation function is used, and every negative value in the output volume from the convolution layer is replaced with zero. This is done to prevent the values from summing up to zero.

> 💡 **Pro tip:** Looking for a perfect source for a recap of activation functions? Check out **Types of Neural Networks Activation Functions.**

Pooling Layer

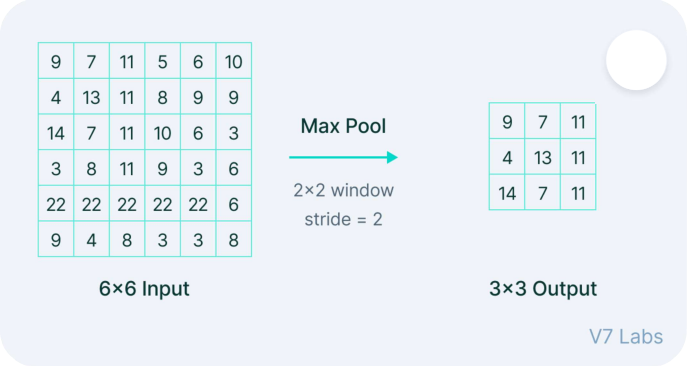
Pooling layers are added in between two convolution layers with the sole purpose of reducing the spatial size of the image representation.

The pooling layer has two hyperparameters:

- window size
- stride

From each window, we take either the maximum value or the average of the values in the window depending upon the type of pooling being performed.

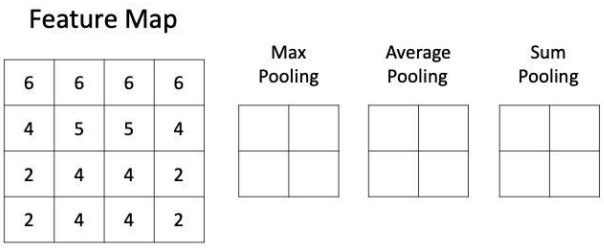
The Pooling Layer operates independently on every depth slice of the input and resizes it spatially ,and later stacks them together.



Types of Pooling

Max Pooling selects the maximum element from each of the windows of the feature map. Thus, after the max-pooling layer, the output would be a feature map containing the most dominant features of the previous feature map.

Average Pooling computes the average of the elements present in the region of the feature map covered by the filter. It simply averages the features from the feature map.



NOTE: Max Pooling performs a lot better than Average Pooling

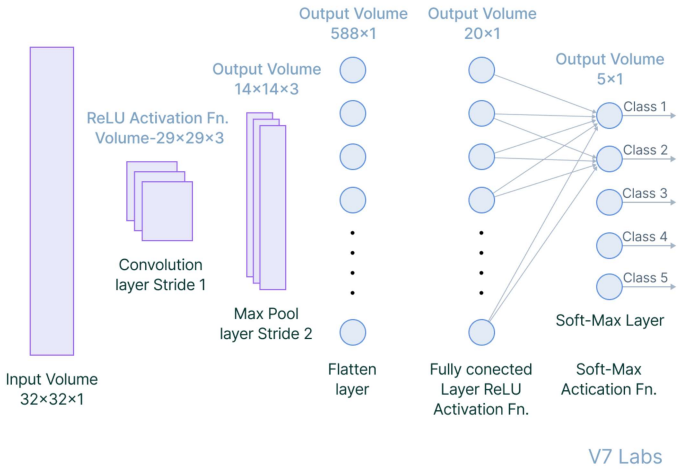
Normalization Layer

Normalization layers, as the name suggests, normalize the output of the previous layers. It is added in between the convolution and pooling layers, allowing every layer of the network to learn more independently and avoid overfitting the model.

However, normalization layers are not used in advanced architectures because they do not contribute much towards effective training.

Fully-Connected Layer

The Convolutional Layer, along with the Pooling Layer, forms a block in the Convolutional Neural Network. The number of such layers may be increased for capturing finer details depending upon the complexity of the task at the cost of more computational power.



Having been able to furnish important feature extraction, we are going to flatten the final feature representation and feed it to a

regular fully-connected neural network for image classification purposes.

How do Convolutional Neural Networks work?

Now, let's get into the nitty-gritty of how CNNs work in practice.

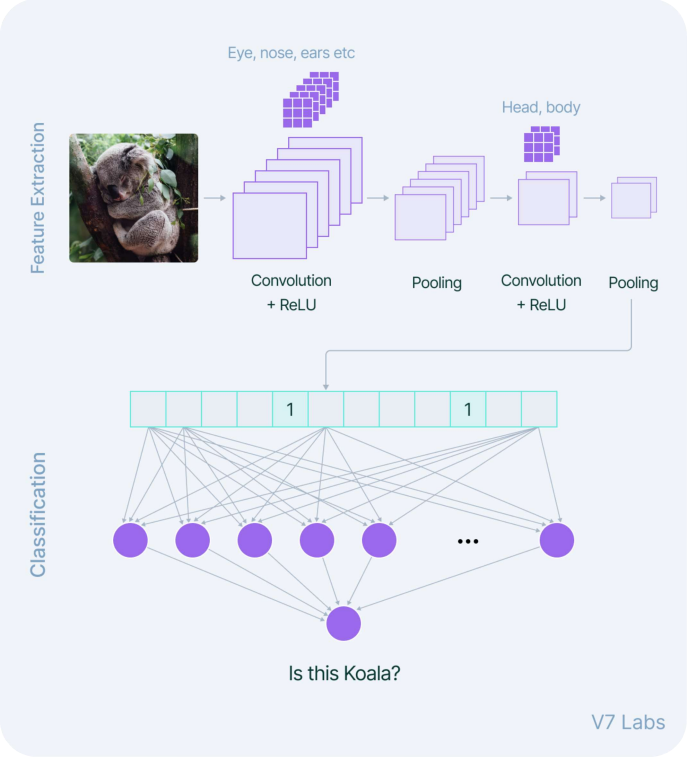
A CNN has hidden layers of convolution layers that form the base of ConvNets. Like any other layer, a convolutional layer receives input volume, performs mathematical scalar product with the feature matrix (filter), and outputs the feature maps.

Features refer to minute details in the image data like edges, borders, shapes, textures, objects, circles, etc.

At a higher level, convolutional layers detect these patterns in the image data with the help of filters. The higher-level details are taken care of by the first few convolutional layers.

The deeper the network goes, the more sophisticated the pattern searching becomes.

For example, in later layers rather than edges and simple shapes, filters may detect specific objects like eyes or ears, and eventually a cat, a dog, and whatnot.



The first hidden layer in the network dealing with images is usually a convolutional layer.

When adding a convolutional layer to a network, we need to specify the number of

filters we want the layer to have.

A filter can be thought of as a relatively small matrix for which we decide the number of rows and columns this matrix has. The value of this feature matrix is initialized with random numbers. When this convolutional layer receives pixel values of input data, the filter will convolve over each patch of the input matrix.

The output of the convolutional layer is usually passed through the ReLU activation function to bring non-linearity to the model. It takes the feature map and replaces all the negative values with zero.

But—

We haven't addressed the issue of too much computation that was a setback of using feedforward neural networks, did we?

It's because there's no significant improvement.

The pooling layer is added in succession to the convolutional layer to reduce the dimensions.

We take a window of say 2x2 and select either the maximum pixel value or the average of all pixels in the window and continue sliding the window. So, we take the feature map, perform a pooling operation, and generate a new feature map reduced in size.

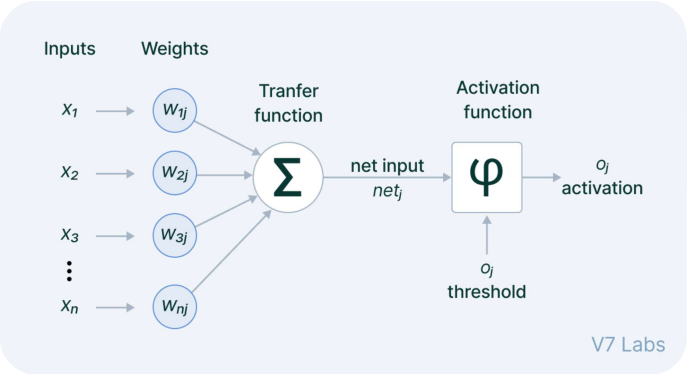
Pooling is a very important step in the ConvNet as reduces the computation and makes the model tolerant towards distortions and variations.

The convolutional layer was responsible for the feature extraction. But—

What about the final prediction?

A fully connected dense neural network would use a flattened feature matrix and predict according to the use case.

> 💡 **Pro tip:** Looking for image data to work on different computer vision problems? Check out 65+ free datasets for machine learning.



Popular Convolutional Networks Architectures

Now, let's go over a few of the most common convolutional neural network architectures.

LeNet

This was the first introduced convolutional neural network. LeNet was trained on 2D images, grayscale images with a size of 32*32*1. The goal was to identify hand-

written digits in bank cheques. It had two convolutional-pooling layer blocks followed by two fully connected layers for classification.

AlexNet

AlexNet was trained on the Imagenet dataset with 15 million high-resolution images with 256*256*3.

ReLU activation function was used between convolution layers and pooling layers for the first time as well as the overlapping pooling with stride < window size. It had five convolutional-pooling layer blocks followed by three fully connected dense layers for classification.

VGGNet

VGGNet came with a solution to improve performance rather than keep adding more dense layers in the model.

The key innovation came down to grouping layers into blocks that were repetitively used in the architecture because more layers of narrow convolutions were deemed more powerful than a smaller number of wider convolutions.

A VGG-block had a bunch of 3x3 convolutions padded by 1 to keep the size of output the same as that of input, followed by max pooling to half the resolution. The architecture had *n* number of VGG blocks followed by three fully connected dense layers.

GoogLeNet

This architecture has Inception blocks that comprise 1x1, 3x3, 5x5 convolution layers followed by 3x3 max pooling with padding (to make the output of the same shape as the input) on the previous layer and concatenates their output.

It has 22 layers, none of which are fully connected layers. It requires a total of 4 million parameters which is still 12 times fewer parameters than previous architectures like AlexNet.

ResNet

It was observed that with the network depth increasing, the accuracy gets saturated and eventually degrades. Therefore, data scientists proposed a solution of skip connections.

These connections provide an alternate pathway for data and gradients to flow, make training fast, and enable skipping one or more layers. The idea of residual blocks was proposed which was based on the fact that deeper models should not produce higher training error than their shallow counterparts.

As a matter of fact, a deeper network was made from shallow networks by setting other layers in the deeper network to be identity mapping.

DenseNet

A limitation that was seen in ResNet was that of vanishing gradients. The key solution was

to create short paths from early layers to later layers to train deep networks. All layers were connected directly to each other.

ZFNet

It is a modification of AlexNet. The major difference is that the architecture of ZFNet uses 7x7 filters, whereas AlexNet uses 11x11 filters based on the thought that bigger filters might lead to loss of information. These changes proved effective and improved efficiency.

> 💡 **Pro tip:** Explore one of ConvNets major areas of application - Computer Vision.

Convolutional Neural Networks: Summary

Finally, let's summarize everything we have learned today.

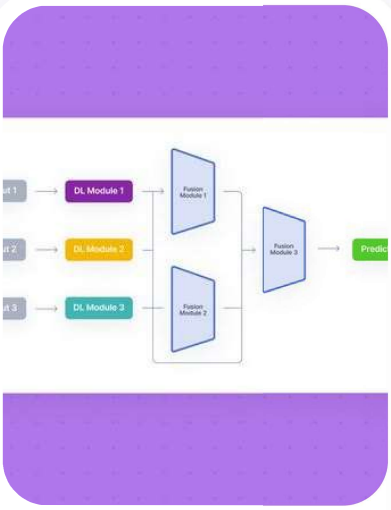
- The Convolutional Neural Network is a type of artificial neural network commonly applied in image processing problems.
- A fully connected neural network involves far more computations than a ConvNet ,which does not work well in images.
- There are a couple of layers that make CNN unique—the convolutional layer and the Pooling layer.

- The convolutional layer works by placing a filter over an array of image pixels and creates a convolved feature map. It is simply looking at an image through a window that allows you to see the presence of specific features.
- Pooling layer down samples and reduces the size of the feature map.
- There are two types of pooling- Max Pooling that takes the maximum value from a particular convolved input or Average Pooling that simply takes the average of all the values.
- A fully connected layer is needed, which takes input from the flattened feature map to perform classification.
- A Convolutional Neural Network is a feedforward network that filters spatial data.

Pragati Baheti
Microsoft

Pragati is a software developer at Microsoft, and a deep learning enthusiast. She writes about the fundamental mathematics behind deep neural networks.

Related articles

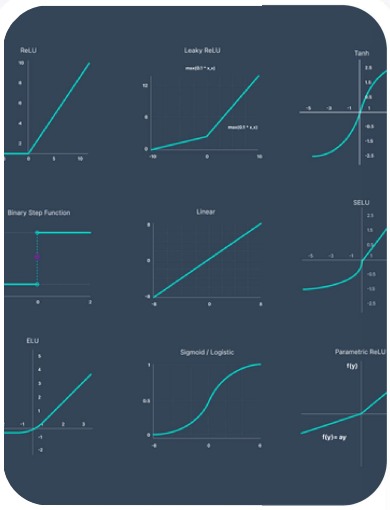


DEEP LEARNING

Multimodal Deep Learning: Definition, Examples, Applications

Konstantinos Poulinakis

min read

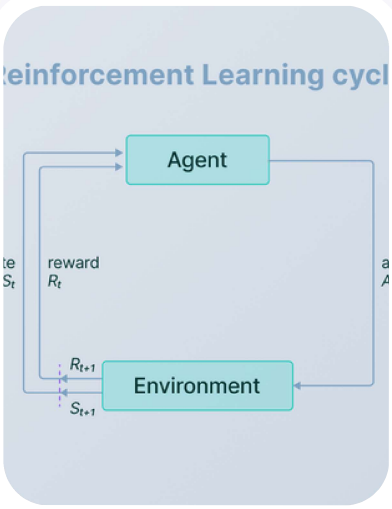


DEEP LEARNING

Activation Functions in Neural Networks [12 Types & Use Cases]

Pragati Baheti

14 min read



DEEP LEARNING

The Beginner's Guide to Deep Reinforcement Learning [2023]

Pragati Baheti

10 min read