

A convolutional neural network consists of an input and an output layer, as my friend mentioned earlier it consist of multiple [hidden layers](#).

The hidden layers of a CNN typically consist of convolutional layers, RELU layer i.e. activation function, pooling layers, fully connected layers and normalization layers.

Now i will shortly describe each layer type function:

convolutional layers - apply a convolution operation to the input, passing the result to the next layer. But what is convolution operation - simply speaking its just generalization.

In neural network term in the field of image recognition it determines how a matrix typically with size 3 by 3 describes one element - viz a pixel in the top-left corner of this matrix. The value is calculated using special "feature detectors". For example it can be matrixes which finds edges on image.

Why we do that?

Basically, it is meant to separate the wheat from the chaff. We want detect certain features which allow us to immediately determine at what we are looking. For example for face image we will detect nose, eyes, mouth and then we can interpret if someone is smiling or if he/her is angry. The second advantage is that the amount of parameters or weights is significantly reduced, thus lessening the computation cost.

After each convolutional can be placed **pooling layer** which also reduce dimensions of our image by combining the outputs of neuron clusters at one layer into a single neuron in the next layer. Typically are used max pooling layers or average pooling layers which computes the indicated value and set it to the scaled pixel.

ReLU (Rectified Linear Units) layer - this layer apply an element wise activation function. ReLU is used after every convolution operation. It is applied per pixel and replaces all negative pixel values in the feature map by zero. This leaves the size of the volume unchanged. ReLU is a non-linear operation.

Fully connected layer - Fully connected layers connect every neuron in one layer to every neuron in another layer. It is in principle the same as the traditional [multi-layer perceptron](#) neural network (MLP). The flattened matrix goes through a fully connected layer to classify the images.

Normalization layer defines a vector of probability values which hold information about with what probability it can be said that a given image is an object of a given class.

Our technology stack will base on Python. For operation with data we will probably use numpy library. And as a core of our application - to implement convolutional neural network we are going to use pytorch or TensorFlow library.