Basic Overview of Convolutional Neural Network (CNN)

<https://medium.com/dataseries/basic-overview-of-convolutional-neural-network-cnn-4fcc7dbb4f17>

Convolutional Neural Network is a class of deep neural network that is used for Computer Vision or analyzing visual imagery.

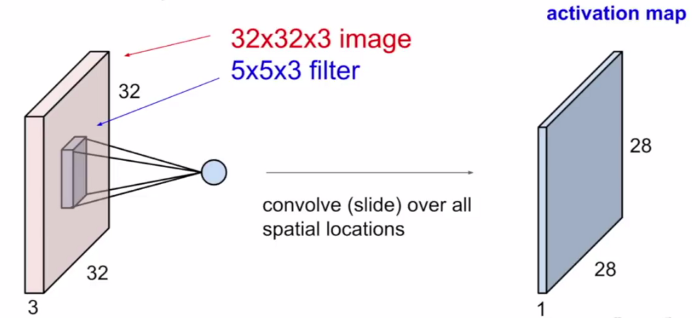
## CONVOLUTIONAL LAYER

Computers read images as pixels and it is expressed as a matrix (NxNx3) — (height by width by depth). Images make use of three channels (RGB), so that is why we have a depth of 3.

The Convolutional Layer makes use of a set of learnable filters. A filter is used to detect the presence of specific features or patterns present in the original image (input). It is usually expressed as a matrix (MxMx3), with a smaller dimension but the same depth as the input file.

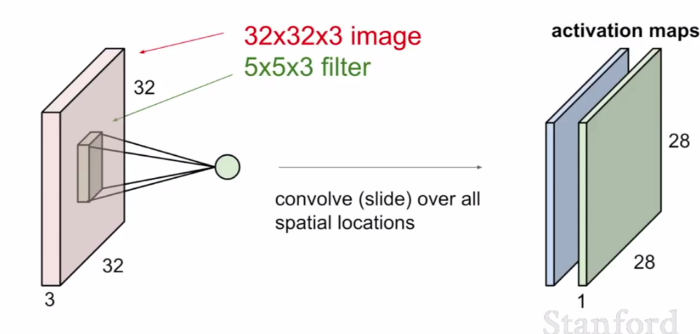
This filter is convolved (slided) across the width and height of the input file, and a dot product is computed to give an activation map.

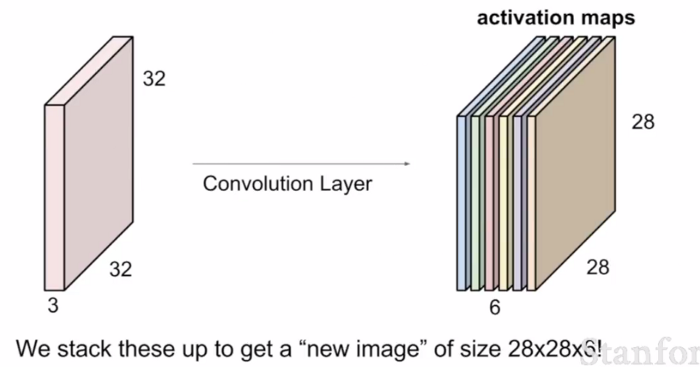
Different filters which detect different features are convolved on the input file and a set of activation maps is outputted which is passed to the next layer in the CNN.



[The use of filters to get Activation Maps](http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture6.pdf)

Source: <http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture6.pdf>





[Set of Activation maps](http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture6.pdf)

Dobili smo 6 aktivacijskih mapa jer smo jednu sliku provukli kroz 6 filtera

FORMULA ZA DOBIVANJE DIMENZIJA AKTIVACIJSKE MAPE!

There is a formula which is used in determining the dimension of the activation maps:

(N + 2P — F)/ S + 1;

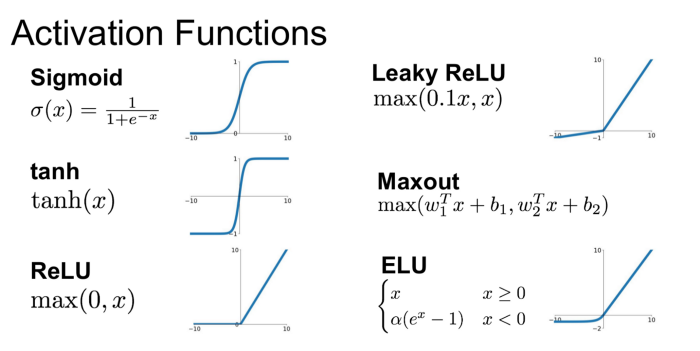
where N = Dimension of image (input) file

* P = Padding
* F = Dimension of filter
* S = Stride

## ACTIVATION FUNCTION

The activation function is a node that is put at the end of or in between Neural Networks. They help to decide if the neuron would fire or not.

“The activation function is the non linear transformation that we do over the input signal. This transformed output is then sent to the next layer of neurons as input.” — [Analytics Vidhya](https://www.analyticsvidhya.com/blog/2017/10/fundamentals-deep-learning-activation-functions-when-to-use-them/)



Activation Functions

We have different types of activation functions just as the figure above, but for this post, my focus will be on **Rectified Linear Unit** (ReLU).

**ReLU** function is the most widely used activation function in neural networks today. One of the greatest advantage ReLU has over other activation functions is that it does not activate all neurons at the same time. From the image for ReLU function above, we’ll notice that it converts all negative inputs to zero and the neuron does not get activated. This makes it very computational efficient as few neurons are activated per time. It does not saturate at the positive region. In practice, ReLU converges six times faster than tanh and sigmoid activation functions.

Some disadvantage ReLU presents is that it is saturated at the negative region, meaning that the gradient at that region is zero. With the gradient equal to zero, during backpropagation all the weights will not be updated, to fix this, we use**Leaky ReLU**. Also, ReLU functions are not zero-centered. This means that for it to get to its optimal point, it will have to use a zig-zag path which may be longer.

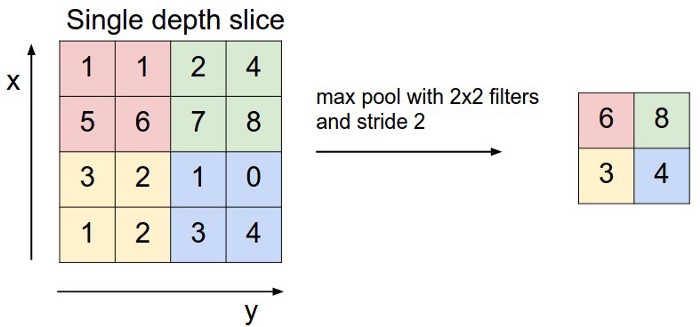
## POOLING LAYER

The Pooling layer can be seen between Convolution layers in a CNN architecture. This layer basically reduces the number of parameters and computation in the network, controlling overfitting by progressively reducing the spatial size of the network.

There are two operations in this layer; Average pooling and Maximum pooling. Only Max-pooling will be discussed in this post.

**Max-pooling**, like the name states; will take out only the maximum from a pool. This is actually done with the use of filters sliding through the input; and at every stride, the maximum parameter is taken out and the rest is dropped. This actually down-samples the network.

Unlike the convolution layer, the pooling layer does not alter the depth of the network, the depth dimension remains unchanged.



Max-pooling

FORMULA ZA OUTPUT OD MAX-POOLINGA:

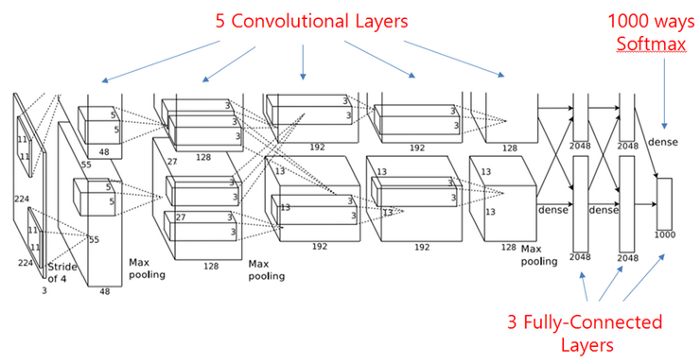
Formula for the output after Max-pooling:

* (N — F)/ S + 1;
* where N = Dimension of input to pooling layer
* F = Dimension of filter
* S = Stride

## FULLY-CONNECTED LAYER

In this layer, the neurons have a complete connection to all the activations from the previous layers. Their activations can hence be computed with a matrix multiplication followed by a bias offset. This is the last phase for a CNN network.

The Convolutional Neural Network is actually made up of **hidden layers** and **fully-connected layer(s)**.



Složena konvolucijska mreža

[S](https://world4jason.gitbooks.io/research-log/content/deepLearning/CNN/Model%20&%20ImgNet/alexnet/alexnet.html)ource: <https://world4jason.gitbooks.io/researchlog/content/deepLearning/CNN/Model%20&%20ImgNet/alexnet/alexnet.html>