**ALEXNET**

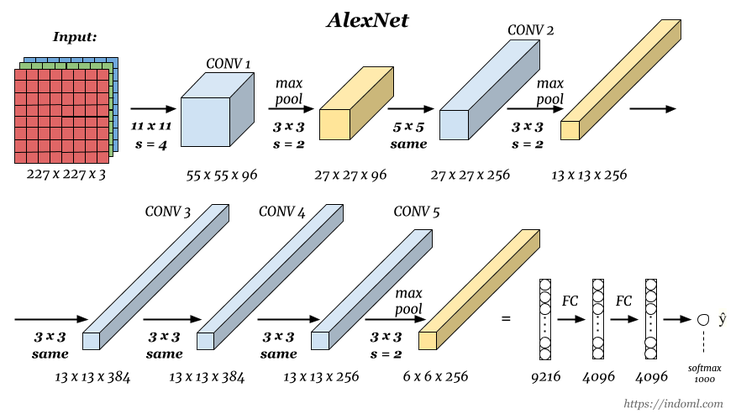
This Architecture was developed based on ImageNet Dataset.

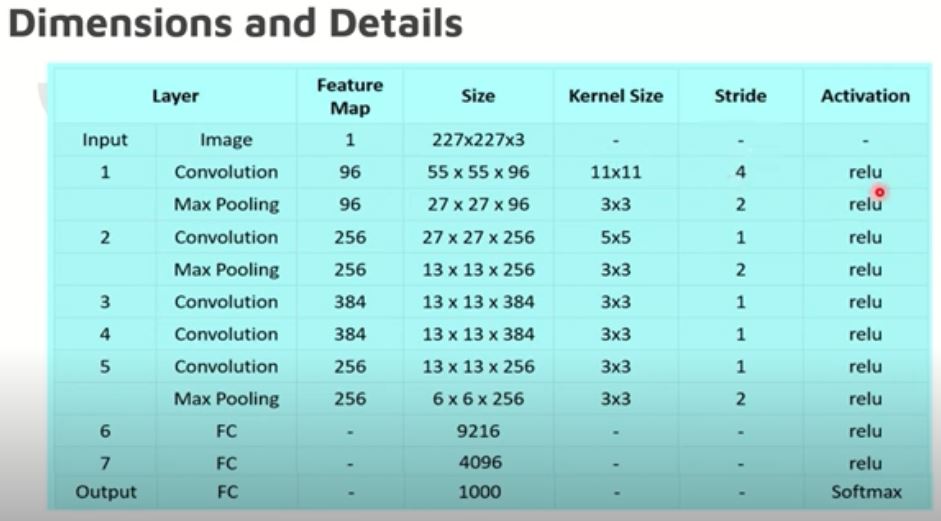
This Architecture is kind of an experiment where the researchers got the best results.

The Alexnet led to the creation of the other architectures such as vgg16, resnet, google net etc.

Because Alexnet has some disadvantages which led to the invention of various architectures.

The input size of AlexNet is fixed, so they downsampled the images to 256x256 by first rescaling images such that the shorter side was of length 256, and then cropping out the central 256×256 patch from the resulting images.





The input image size is 227x227x3 as given in the above table.

The relu activation function is used in all the layers except output layer.

We will pass the input to the convolution layer where the size of filter is 11\*11 and we are using 96 filters with stride of 4. The output of the convolution layer is 55\*55\*96.

55\*55 output came by using the formula. 96 channels because we are using 96 filters.

Formula= ((n-f+2p)/s)+1

The output of convolution layer is given as input to the maxpooling layer.

Here we use 3\*3 filter with stride =2

Output of maxpooling is 27\*27\*96.

In maxpooling the number of channels doesn’t change.

27\*27 output came by using the formula.

In the same way all the remaining layers have been created.

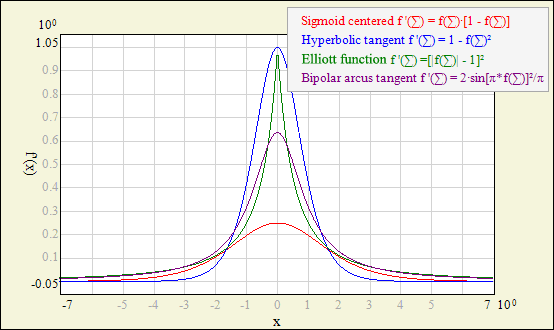
**Why does AlexNet achieve better results?**

***ReLU Non-Linearity:***

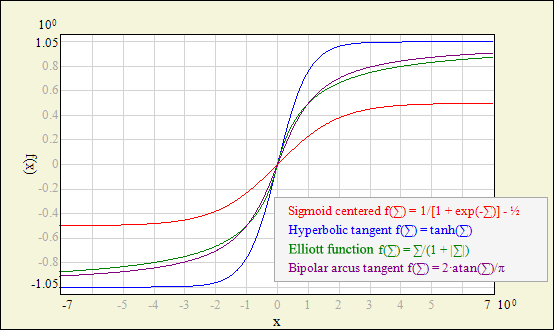
Relu activation function is used.

Relu function: f (x) = max (0, x)

We can see from the below picture that if the backward input is larger, the gradient of tanh or sigmoid function will be very small(gradient vanishing), which leads to slow down the training speed.

http://www.ire.pw.edu.pl/~rsulej/NetMaker/index.php?pg=n01

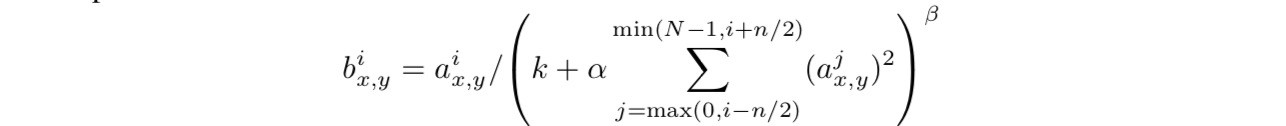
The result showed that using ReLU function to training was 8 times faster than tanh to reach the same error rate.Besides, if using tanh or sigmoid, we have to normalize inputs in -1~1 to avoid them entering the saturation region.



<http://www.ire.pw.edu.pl/~rsulej/NetMaker/index.php?pg=n01>

***Local Response Normalization:***

Althought ReLUs have the desirable property that they do not require input normalization to prevent them from saturating, in AlexNet, they still applied normalization after applying the ReLU nonlinearity in certain layers.



They found that response normalization reduces the top-1 and top-5 error rates by 1.4% and 1.2%, respectively.

***Overlapping Pooling***

In AlexNet, they used overlapping pooling to reduce dimensions. They found that with s = 2 and z = 3, the scheme reduces the top-1 and top-5 error rates by 0.4% and 0.3%, respectively, as compared with the non-overlapping scheme with s = 2 and z = 2, which produces output of equivalent dimensions. They also found that during training that models with overlapping pooling were more difficult to overfit.

1. **Enhanced Data (Data Augmentation )**

In deep learning, when the amount of data is not large enough, there are generally 4 solutions:

  Data augmentation- artificially increase the size of the training set-create a batch of "new" data from existing data by means of translation, flipping, noise

  Regularization——The relatively small amount of data will cause the model to overfit, making the training error small and the test error particularly large. By adding a regular term after the Loss Function , the overfitting can be suppressed. The disadvantage is that a need is introduced Manually adjusted hyper-parameter.

  Dropout- also a regularization method. But different from the above, it is achieved by randomly setting the output of some neurons to zero

  Unsupervised Pre-training- use Auto-Encoder or RBM's convolution form to do unsupervised pre-training layer by layer, and finally add a classification layer to do supervised Fine-Tuning