Lab-7

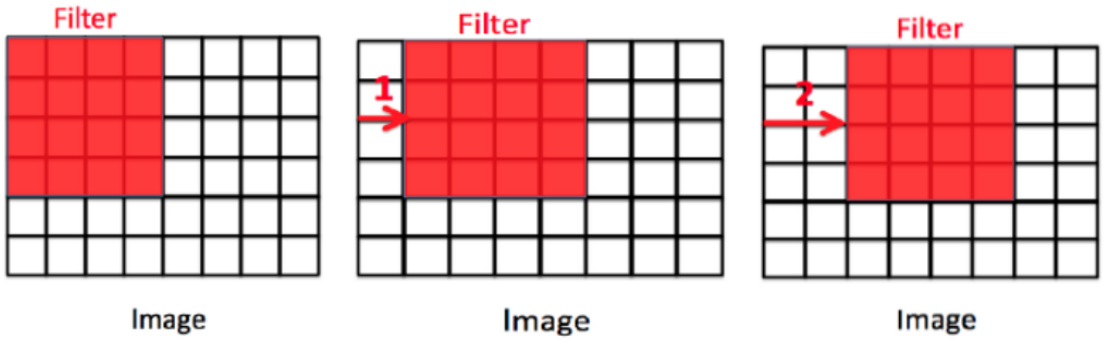
Rashmi S

21BDA02

2.What is Stride, Padding & Pooling? Explain with an example

Stride

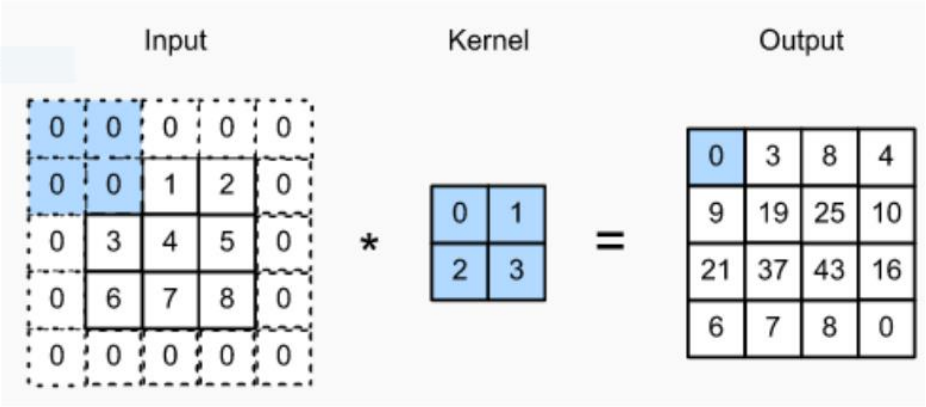
When the array is created, the pixels are shifted over to the input matrix. The number of pixels turning to the input matrix is known as the strides. When the number of strides is 1, we move the filters to 1 pixel at a time. Similarly, when the number of strides is 2, we carry the filters to 2 pixels, and so on. They are essential because they control the convolution of the filter against the input, i.e., Strides are responsible for regulating the features that could be missed while flattening the image. They denote the number of steps we are moving in each convolution. The following figure shows how the convolution would work.



Padding

The padding plays a vital role in creating CNN. After the convolution operation, the original size of the image is shrunk. Also, in the image classification task, there are multiple convolution layers after which our original image is shrunk after every step, which we don't want.

Secondly, when the kernel moves over the original image, it passes through the middle layer more times than the edge layers, due to which there occurs an overlap.
To overcome this problem, a new concept was introduced named padding. It is an additional layer that can add to the borders of an image while preserving the size of the original picture. For example:

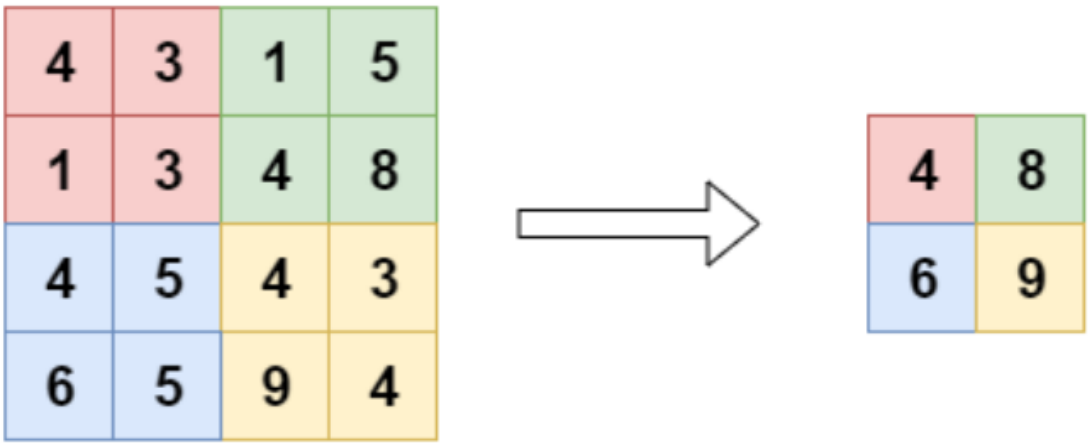


Pooling

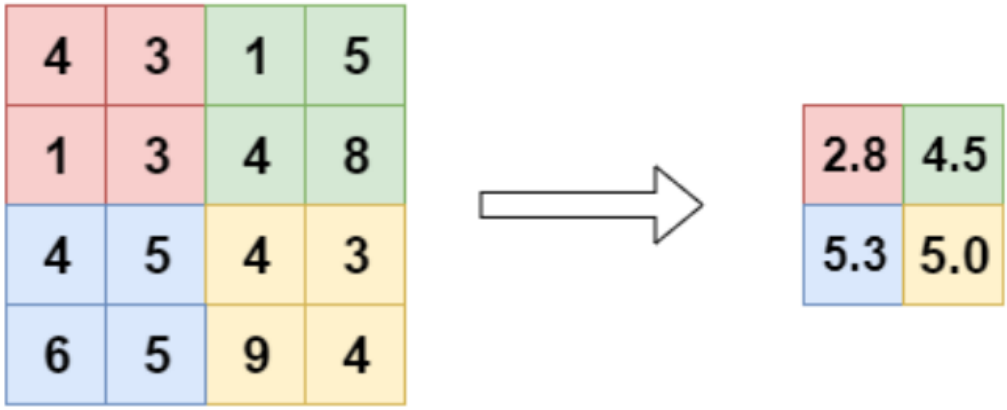
Then the picture is shrunk, the pixel density is also reduced, the downscaled image is obtained from the previous layers. Basically, its function is to progressively reduce the spatial size of the image to reduce the network complexity and computational cost. Spatial pooling is also known as down sampling or subsampling that reduces the dimensionality of each map but retains the essential feature.

Types:-

a) Max pooling



b) Avg pooling



Reference link:-
https://www.codingninjas.com/codestudio/library/convolution-layer-padding-stride-and-pooling-in-cnn

4.What is overfitting? How to overcome overfitting in an ML model?

A key challenge with overfitting, and with machine learning in general, is that we can't know how well our model will perform on new data until we actually test it.

To address this, we can split our initial dataset into separate *training* and *test* subsets.

This method can approximate of how well our model will perform on new data.

If our model does much better on the training set than on the test set, then we're likely overfitting

There are several ways

a) **Cross-validation**

   It is a powerful preventative measure against overfitting.The idea is clever: Use your initial training data to generate multiple mini train-test splits. Use these splits to tune your model.

b) **Train with more data**

c) **Remove features**

Some algorithms have built-in feature selection. For those that don't, we can manually improve their generalizability by removing irrelevant input features.

d) **Early stopping**

   Up until a certain number of iterations, new iterations improve the model. After that point, however, the model's ability to generalize can weaken as it begins to overfit the training data.

e) **Regularization**

Regularization refers to a broad range of techniques for artificially forcing your model to be simpler.

The method will depend on the type of learner you're using. For example, you could prune a decision tree, use dropout on a neural network, or add a penalty parameter to the cost function in regression.

f) **Ensembling**

Ensembles are machine learning methods for combining predictions from multiple separate models. There are a few different methods for ensembling, but the two most common are: Bagging and Boosting

Reference link :- https://elitedatascience.com/overfitting-in-machine-learning