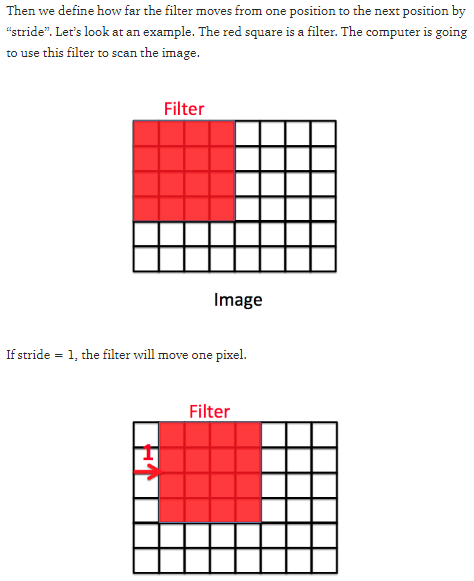
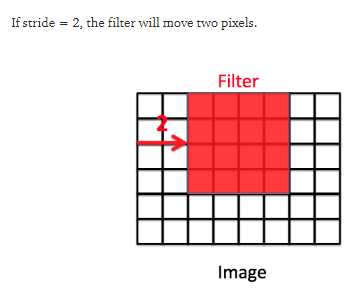
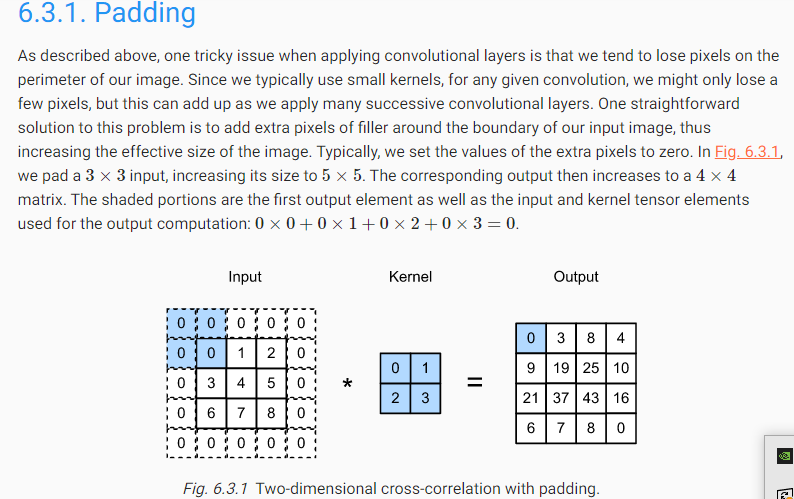
What is stride, paddling and pooling? Explain with example.

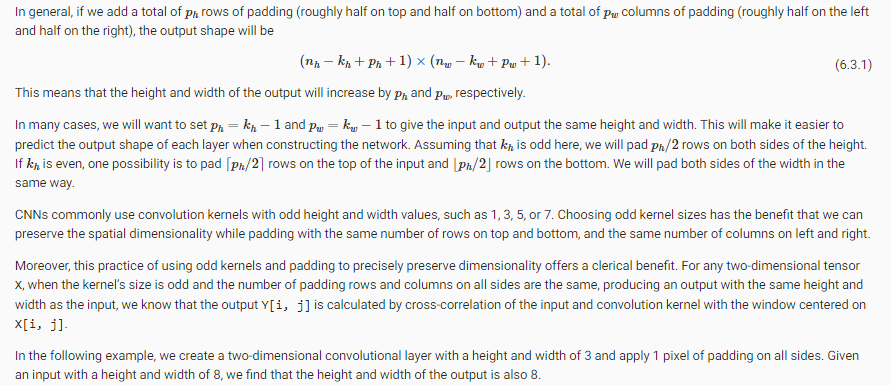
Stride: is the amount by which the filter slides over the image. For example, to slide the CONV filter one pixel at a time, then the stride = 1. If we want to jump two pixels at a time, then stride = 2. Strides of 3 or more are uncommon and rare in practice. Jumping pixels will produce smaller output volumes spatially.

A Stride of 1 will make the output image roughly the same height and width of the input image. While a stride of 2 will make the output image roughly about half of the input image size. I say roughly because it depends on what you set the padding parameter to do with the edge of the image.





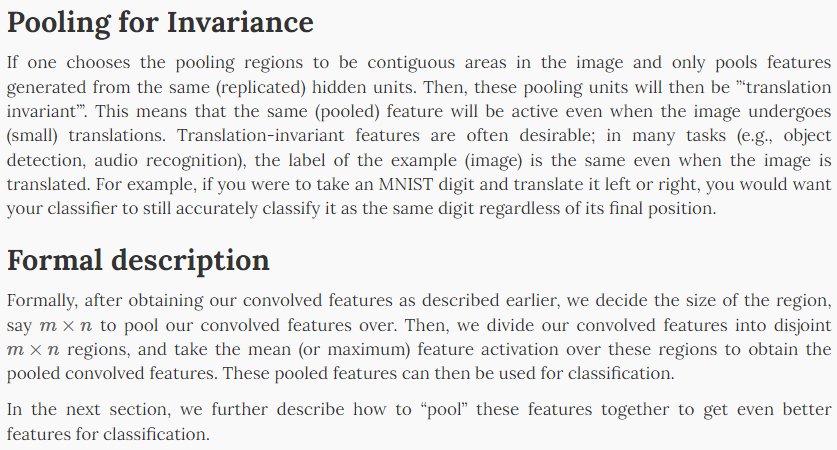




Pooling: Overview

After obtaining features using convolution, we would next like to use them for classification. In theory, one could use all the extracted features with a classifier such as a softmax classifier, but this can be computationally challenging. Consider for instance images of size 96x96 pixels, and suppose we have learned 400 features over 8x8 inputs. Each convolution results in an output of size (96−8+1)∗(96−8+1)=7921, and since we have 400 features, this results in a vector of 892∗400=3,168,400 features per example. Learning a classifier with inputs having 3+ million features can be unwieldy, and can also be prone to over-fitting.

To address this, first recall that we decided to obtain convolved features because images have the “stationarity” property, which implies that features that are useful in one region are also likely to be useful for other regions. Thus, to describe a large image, one natural approach is to aggregate statistics of these features at various locations. For example, one could compute the mean (or max) value of a particular feature over a region of the image. These summary statistics are much lower in dimension (compared to using all of the extracted features) and can also improve results (less over-fitting). We aggregation operation is called this operation ”‘pooling”’, or sometimes ”‘mean pooling”’ or ”‘max pooling”’ (depending on the pooling operation applied).



What is overfitting? How to overcome overfitting in an ml model?

Overfitting occurs when you achieve a good fit of your model on the training data, while it does not generalize well on new, unseen data. In other words, the model learned patterns specific to the training data, which are irrelevant in other data.

We can identify overfitting by looking at validation metrics, like loss or accuracy. Usually, the validation metric stops improving after a certain number of epochs and begins to decrease afterward. The training metric continues to improve because the model seeks to find the best fit for the training data.

There are several manners in which we can reduce overfitting in deep learning models. The best option is to get more training data. Unfortunately, in real-world situations, you often do not have this possibility due to time, budget or technical constraints.

Another way to reduce overfitting is to lower the capacity of the model to memorize the training data. As such, the model will need to focus on the relevant patterns in the training data, which results in better generalization. In this post, we’ll discuss three options to achieve this.