COMP3420 — AI for Text and Vision

Week 03 Lecture 1: Convolutional Networks for Image Classification

Diego Mollá

COMP3420 2023H1

Abstract

This lecture will focus on the basic principles of convolutional networks. Convolutional networks revolutionised the use of deep learning for image classification, and some of the most popular image classification systems are based on the principles of convolutional networks. In this lecture we will only cover simple convolutional networks. In a subsequent lecture, we will look at some of the most popular architectures that use convolutional networks for image classification.

Update March 2, 2023

Contents

1 Convolutional Networks **1**
1.1 Convolution 2
1.2 Pooling 5

2 Convolutional Networks in Keras **6**

Reading

- Deep Learning with Python, 2nd Edition, Chapter 8.
- Practical Machine Learning for Computer Vision, Chapter 3.

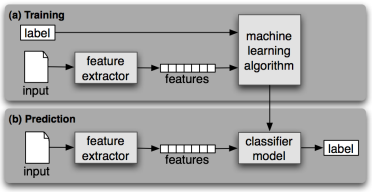
1 Convolutional Networks

Remember: Supervised Machine Learning

Given
Training data annotated with class information.

Goal
Build a *model* which will allow classification of new data.

Method



(figure from NLTK book)

- Feature extraction: Convert samples into vectors.
- Training: Automatically learn a model.
- Classification: Apply the model on new data.

(Caption for figure from NLTK book)
Figure 1.1: Supervised Classification. (a) During training, a feature extractor is used to convert each input value to a feature set. These feature sets, which capture the basic information about each input that should be used to classify it, are discussed in the next section. Pairs of feature sets and labels are fed into the machine learning algorithm to generate a model. (b) During prediction, the same feature extractor is used to convert unseen inputs to feature sets. These feature sets are then fed into the model, which generates predicted labels.

1.1 Convolution

Convolution layer vs a densely connected layer

- A densely connected layer is able to detect information that is global to the entire image.
- Often, however, we want to detect information that is specific to parts of the image.
- Convolution layers focus on specific regions of the image and are able to detect local patterns.

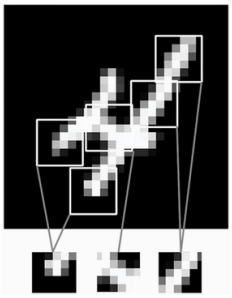
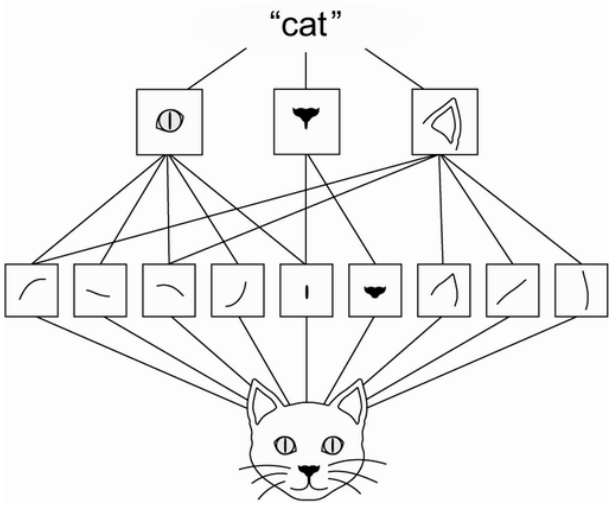


Figure 8.1 of "Deep Learning with Python", 2nd edition: Images can be broken into local patterns such as edges, textures, and so on.

Key Characteristics of ConvNets

- They can learn patterns based on specific regions of the image.
- The patterns they learn are invariant: After learning a certain pattern in one part of the image, a convnet can recognise it anywhere in the image.
- When we cascade convnet layers, they can learn spatial hierarchies of patterns.

Convolutional Networks can Extract Useful Features



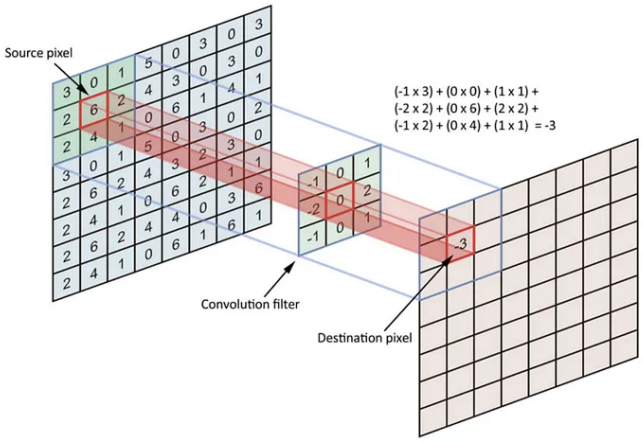
(Figure 8.2 from Deep Learning with Python)

(Caption of Figure 8.2 of book "Deep Learning with Python", 2nd Edition)

The visual world forms a spatial hierarchy of visual modules: elementary lines or textures combine into simple objects such as eyes or ears, which combine into high-level concepts such as "cat."

Convolutional networks are shown to be able to detect specific characteristics of images so that, when arranged in layers, the lower-level layers (which are closer to the input image) detect low-level information such as edges and textures, and the higher-level layers (which are closer to the final output layer) detect higher-level information such as parts of a face etc.

The Convolution Operation



(`https://towardsdatascience.com/simple-introduction-to-convolutional-neural-networks-cdf8d3077bac`)

The convolution operation will multiply every element of the convolution filter with every element of a region of the image. The sum of the products will represent the new value in the transformed image.

The filter then slides to process the next region of the image, and so on until the entire image has been processed.

- A convolution is a filter that applies to a specific part of the image.

- This filter is basically like a neuron in a densely connected layer that takes as input a part of the image only.
- The convolution filter is then slid across the image.

Anatomy of a Convolution

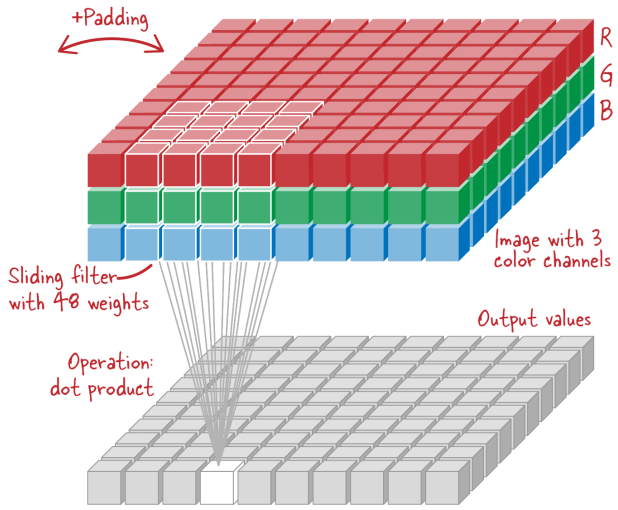
Padding. We often want to pad the original image, with a set value (e.g. 0), so that we can apply the convolution filter to the edges of the image. If there is no padding, the resulting image after the convolution is slightly smaller.

Kernel size. The shape of the filter. A filter with size x will process parts of the image with shape (x, x, c), where c is the number of channels.

Stride. The stride is the number of cells that we skip between each pass of the sliding filter.

Activation. Often, we want to add an activation function that will apply after the convolution operation.

Anatomy of Convolution

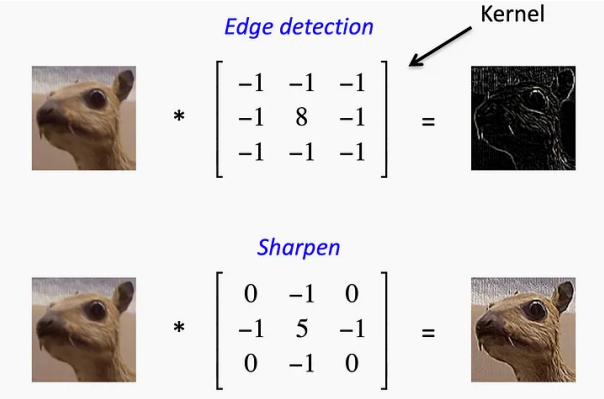


(Figure 3.9 of "Practical Machine Learning for Computer Vision")

In this image, the convolution uses a filter with kernel size 3, with a total of $4 \times 4 \times 3 = 48$ weights.

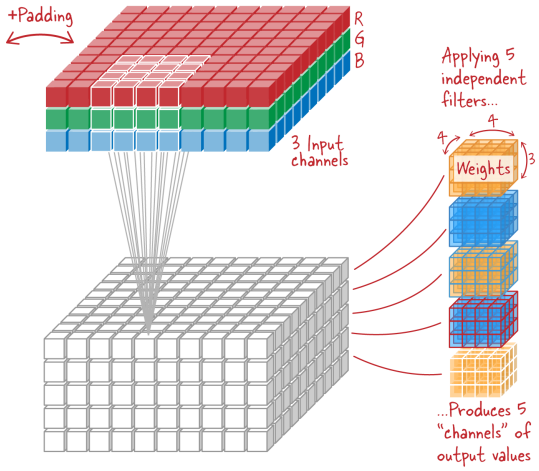
Examples of Filters

Below are examples of weights to make specific filters. In practice, the network will learn the best values during the training stage.



(https://towardsdatascience.com/simple-introduction-to-convolutional-neural-networks-cdf8d3077bac)

Using Multiple Filters



(Figure 3-11 from Computer Vision book)

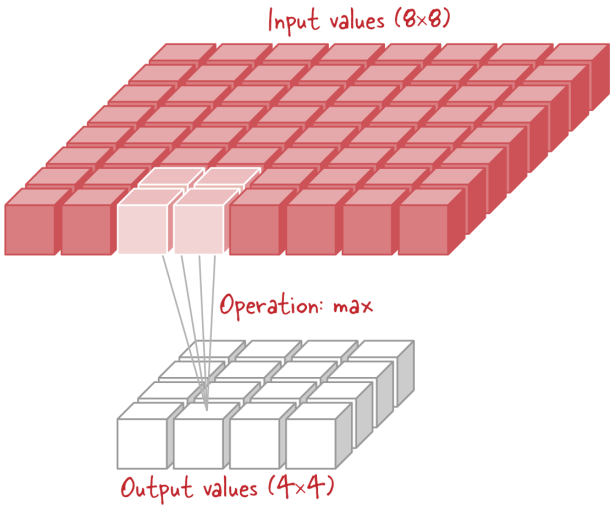
In a convolutional layer, we often apply multiple filters to the same image patch. Each filter has different weights (which are learnt during the training stage), and generates an independent number. Effectively, each filter creates a channel in the new image. In the figure above (Figure 3-22 in "Practical Machine Learning for Computer Vision"), we are applying 5 filters, each with kernel size 4 x 4 x 3.

1.2 Pooling

Pooling

- A convolution layer is often followed by a pooling layer.
- A pooling layer will combine all values of a region of the resulting image into a single number.
- Often, we use Max Pooling, that is, choose the largest value from the region.

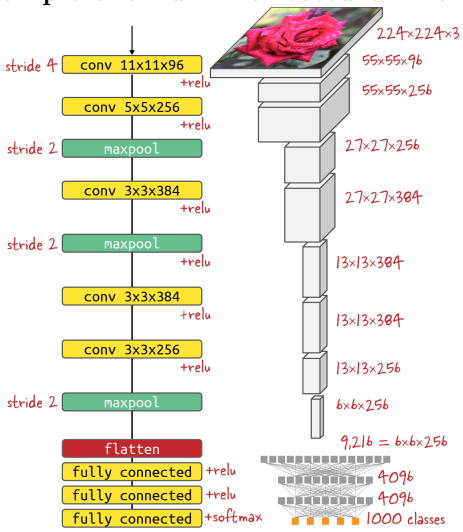
Example of Pooling



(Figure 3-14 from Computer Vision book)

(Caption of Figure 3-14 from "Practical Machine Learning for Computer Vision"):
A 2x2 max-pooling operation applied to a single channel of input data. The max is taken for every group of 2x2 input values and the operation is repeated every two values in each direction (stride 2).

Example of a Full Architecture: AlexNet



(Figure 3-16 from Computer Vision book)
AlexNet (2012) was one of the early successes of ConvNets.
(Caption of Figure 3-16 from "Practical Machine Learning for Computer Vision")
The AlexNet architecture: neural network layers are represented on the left. Feature maps (as transformed by the layers) on the right.

2 Convolutional Networks in Keras

Convolutional Networks in Keras

- This section is based on jupyter notebooks provided by the unit textbooks.

- Study these notebooks carefully.
- The notebooks also introduce important terminology that you need to understand.

Take-home Messages

1. Explain the advantages of convolutional layers vs. dense layers.
2. Explain convolutional and pooling layers.
3. Using Keras, implement image classifiers that include stacked convolutional and pooling layers.

What's Next

Assignment 1

- Submit Friday 10 March, 11:55pm

Week 4

- Advanced Convolutional Networks.
- Friday 17 March: Census Date.

Reading

- Computer Vision book, chapter 3.
- Deep Learning book, chapters 8 and 9.