Convolutional Neural Networks

Deep Learning for Image classification

Vitor Greati[1]

[1] Federal University of Rio Grande do Norte

Table of Contents

Basic concepts

- Correlation and convolution

Convolutional Neural Networks (CNNs)

- Definitions
- Motivation
- Layers in CNNs
    - Layer types

Spatial Processes and Filters

A **spatial domain process** on an image f is defined as

$$g = T[f]$$

where g is the output image, T is an operation that acts on a neighborhood centered at each (x, y) in the domain of f, and $g(x, y) = T[f, (x, y)]$.

A **spatial filter** is made of two components:

- a neighborhood, commonly an odd dimensions' square;
- an operation over the intensities in the neighborhood.

A filtered image is generated as the center of the filter visits each pixel.

Correlation

Correlation and convolution are operations characterized by the sum of products of the elements of the filter with the elements of the image.The difference between them is subtle.

Consider that w is a filter of dimension $m \times n$. For the sake of simplicity, guaranteeing a central pixel, m and n are odd natural numbers. The **correlation** of w with an image f is given by:

$$w(x,y) \star f(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t) \qquad (1)$$

where $a = \frac{m-1}{2}$ and $b = \frac{n-1}{2}$.

Convolution

The **convolution**, in turn, first demands w to be rotated by 180^o, then performs the same task as correlation. The following equation takes the same $m \times n$ filter w and convolves it with the image f. The rotation is performed in f for the sake of notational simplicity, which produces the same result as rotating the filter.

$$w(x,y) \bigstar f(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x-s, y-t) \tag{2}$$

For Deep Learning, correlation is commonly preferred and is generally reffered as convolution, even if the mask is not actually rotated.

Table of Contents

Basic concepts

- Correlation and convolution

Convolutional Neural Networks (CNNs)

- Definitions
- Motivation
- Layers in CNNs
  - Layer types

Definitions

Convolutional Neural Networks

Convolutional Neural Networks are neural networks that swaps in a specialized convolutional layer in place of fully connected layer for at least one of the layers in the network.

CNNs don't use fully connected layers until the very last layer(s) in the network.

Each layer in the CNN applies different sets of filters and combines the results, feeding the next layer. During the training phase, the CNN **learns values for these filters**.

Main benefits

Local invariance

Allows classifying an image as having certain object regardless the location of the object in the image, by means of *pooling layers*.

Compositionality

Each filter composes a local path of lower-level features into a higher-level representation, allowing the network to learn richier features.

Defining filters hand

Filters are generally **handmade** in order to achieve some specific effect, like

- detecting edges;
- blurring;
- sharpening;
- smoothing.

The question is: is there a way to **automatically** define these filters aiming to detect objects and classify images?

The convolutional way

By applying convolution filters, nonlinear activation functions, pooling and backpropagation, Convolutional Neural Networks are able to learn filters that can detect edges and blob-like structures in lower level layers in the network, and then use the edges and structures for eventually detecting high-level objects in the deeper layers.

Why not simple MLPs?

- MLPs do not scale well as image sizes increase.
- Does not achieve high accuracies in complex problems (see CIFAR-10 dataset).

Overview

3D arrangement

Layers in a CNN are organized in a 3D arrangement, where dimensions correspond to the **width** and **height** of the image and the **depth**, which is, for example, the number of channels in the image or the amount of filters in the layer.

These layers are not fully-connected like in MLPs: one layer connects only a subset of its neurons to the subsequent layer, a property called **local connectivity**, which is important for avoiding huge numbers of parameters in the network.

The output layer has dimensions $1 \times 1 \times N$, where N is the number of classes, and the output vector represents the score for each class.

Layer types

Examples of layer types are:

- **Convolutional** (`CONV`)
- **Activation** (`ACT` or `RELU`)
- **Pooling** (`POOL`)
- **Fully-connected** (`FC`)
- Batch Normalization (`BN`)
- Dropout (`DO`)

Describing a CNN architecture

Use the general notation:

```
TYPE1 => TYPE2 => TYPE3 => ...=> TYPEN
```

For example:

```
INPUT => CONV => RELU => FC => SOFTMAX
```