[SYNTHESIS] CHAPTER 9. CONVOLUTIONAL NEURAL NETWORK

Rudresh Mishra, Ricardo Rodriguez
IMT Atlantique

I. Introduction

Convolutional neural networks (CNNs) are a Neural Network that has a grid topology like time-series data (1D grid) or image data (2D grid). The name of convolution is given because of the use of convolution in place of general matrix multiplication in at least one of their layers.

Motivation: Sparse interactions, parameter sharing and equivariant representations are three ideas that convolution allows.

- Sparse interaction is accomplished by making the kernel smaller than the input. For example in edge detection, we can use portions of images (subsets of pixels) from an input image. This will reduce the memory requirements of the model and improves its statistical efficiency.
- Parameter sharing: In traditional neural networks each weight is used exactly once when computing the output of a layer however CNN allows to use the same parameter for more than one function in a model. This reduces the number of parameters to be learnt and also reduces computational needs.
- Equivariant representations: Is a property that means that if the input changes, the output changes in the same way, which gives the property of invariance to changes in illumination, change of position, but internal representation is equivariance to these change

Applications:

a) Human pose estimation, b) Edge detection, c) Face Recognition, d) Human behaviour interpretation, e) Classfication+localization f) object detection, g) Segmentation, h) Image captioning, ...

II. CNN Structure

We define the structure of a CNN

- Input:
  - – size W1 × H1 × D1
- Requires four hyperparameters:
  - –Number of filters K
  - – their spatial extent F
  - – the stride S
  - – the amount of zero padding P
- Produces a volume of size W2 × H2 × D2 where:
  - – W2 = W1−F +2P/S+1
  - – H2 = H1−F +2P/S+1
  - – D2 = K
- With parameter sharing, it introduces F F D1 weights per filter, for a total of F F D1K weights and K biases.
- The filters depth is always equal to the input volume depth.
- The output volume depth is equal to the number of filters
- In the output volume, the d-th depth slice (of size W2 × H2 is the result of performing a valid convolution of the d-th filter over the input volume with a stride of S, and then offset by d-th bias.

Working model:

The filter matrix(weight matrix) is convolved with the input data. The result of convolving the filter over all the input volume is another volume called activation map. we feed a stack of the filter to convolve with the input which produces a stack of the activation map.

Consider a 5 x 5 whose image pixel values are 0, 1 and filter matrix 3 x 3 as shown below in Fig 1.

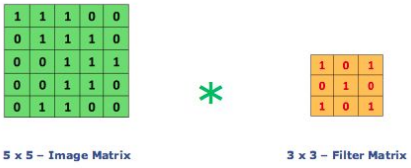

Fig 1

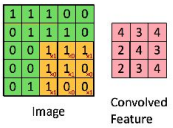

Fig 2

Then the convolution of 5 x 5 image matrix multiplied with 3 x 3 filter matrix which is called **"Feature Map"** as output shown above in Fig 2

III. Key Terminologies:

Pooling: Its the function to progressively reduce the spatial size of the representation to reduce the number of parameters and computation in the network. one of the common poling is max pooling.

Padding: The output volume decreases after each Conv layer, the absence of padding will result in the decrease of the size and result in the volume of size $1 \times 1 \times ?$ after the few-layer. in order to solve it, we add zero around input volume.

Hyperparameter: choosing hyperparameter is a key aspect, it totally depends on the kind of problem we deal with .normally we should be using small filters (e.g. 3×3 or at most 5×5), using a stride of $S = 1$.