1 Convolutional Networks

This is a post written for the AI Helsinki study group *Image and Video Statistics*. It is based on the Chapter 9 of the book *Deep Learning*.

1.1 Definition

1.2 Use of Convolutional Networks

1.3 Benefits

1.4 Examples

1.5 Sparse Connectivity

1.6 Growing Receptive Fields

1.7 Parameter Sharing

1.8 Convolutional Network Components

1.9 Max Pooling

Pictures: Without Shift, Shifted

1.10 Example of Learned Invariances

1.11 Pooling with Down Sampling

1.12 Examples of Architectures

1.13 Convolution with Strides

1.14 Zero Pading Enables Deeper Networks

1.15 Comparison of Local Connections, Convolution, and Full Connections

Pictures: Local, Convolution, FC

1.16 Partial Connectivity Between Channels

1.17 Tiled Convolution

Pictures: Local Connection, Tiled Convolution, Traditional Convolution

1.18 Recurent Convolutional Network

1.19 Gabor Functions (optional)

1.20 Gabor-like Learned Kernels (optional)

2 Tensors

We adopt the Deep Learning book's convention of calling multidimensional arrays of real numbers as tensors.

Namely, 0-D tensors are just real numbers, 1-D tensors are arrays

$$T = (T_1, \dots, T_n)$$

of real numbers, 2-D tensors are arrays

$$T = ((T_{1,1}, \dots, T_{1,n}), \dots (T_{m,1}, \dots, T_{m,n}))$$

of arrays of real numbers (with mutually equal lenght), and so on.

Basically our tensors can be wieved as D-dimensional grids of real numbers. Example in 2-D:

$$((1,2,3),(4,5,6),(7,8,9)) \quad \simeq \quad \begin{array}{c|c|c} 1 & 2 & 3 \\ \hline 4 & 5 & 6 \\ \hline 7 & 8 & 9 \end{array}$$

There are also more elaborate ways of defining tensors, that take into account the type of the tensor, but these are not needed for our purposes.

3 References/Links