Natural Computing
Assignment 5

Christoph Schmidl
s4226887
c.schmidl@student.ru.nl

Koen Vijverberg
s4132858
koen.vijverberg@student.ru.nl

Alex Kolmus
s4125304
alex.kolmus@student.ru.nl

April 22, 2017

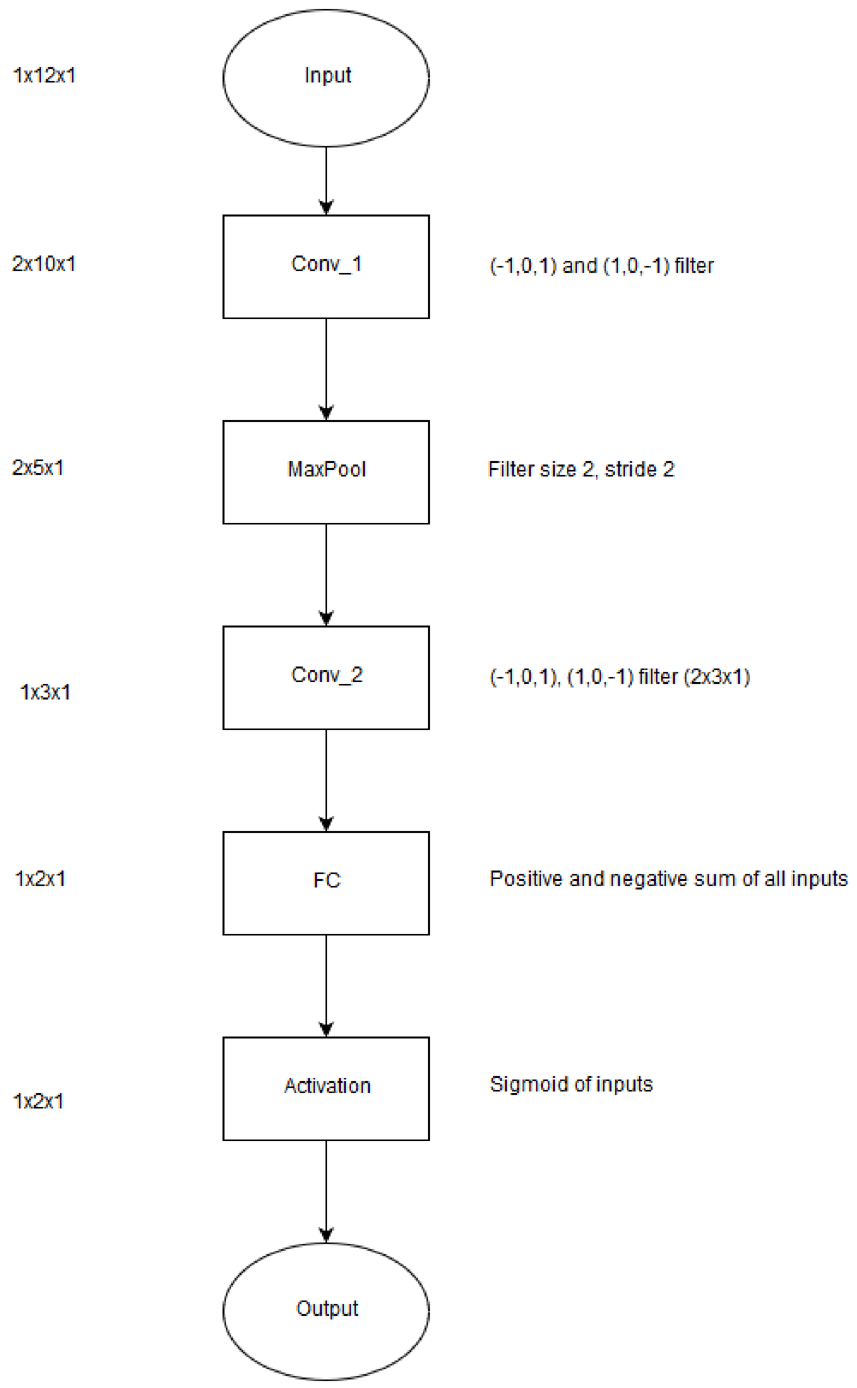
Exercises on Convolutional Neural Networks

1. Consider a convolutional neural network (CNN) with one-dimensional (1D) input. While two dimensional (2D) CNNs can be used for example for grayscale images, one-dimensional CNNs could be used for time-series such as temperature or humidity readings. Concepts for the 1D-case are equivalent to 2D networks.
We interpret data in our network as three-dimensional array where a row denotes a feature map (also called filter, kernel), a column denotes a single dimension of the observation, and the depth of the array represents different observations. As we will only work with single input vector, the depth will always be one. Let the following CNN be given:

   - Input I: Matrix of size 1 x 12 x 1. We therefore have an input with twelve dimensions consisting of a single feature map.
   - First convolutional layer with filters $F_0^1 = (-1, 0, 1)$ and $F_1^1 = (1, 0, -1)$ that generates two output feature maps from a single input feature map. Use *valid* mode for convolutions. Mode *valid* returns output of length max(M,N) - min(M,N) + 1, where M is the size of the array and N = 3 is the size of the filter. The convolution product is only given for points where the signals overlap completely. Values outside the signal boundary have no effect.
   - Max-pooling layer with stride 2 and filter size 2. Note that max-pooling pools each feature map separately.
   - Convolutional layer with convolutional kernel $F_0^2 = (-1, 0, 1), (1, 0, -1)$ of size 2 x 3 x 1.
   - Fully connected layer that maps all inputs to two outputs. The first output is calculated as the negative sum of all its inputs, and the second layer is calculated as the positive sum of all its inputs.
   - Sigmoidal activation function

   (a) Draw the network.
   Solution:

   On the left side are the expected data dims, on the right side the operations performed.



(b) Calculate the response of the CNN for the two inputs (0;0;0;0;1;1;1;1;0;0;0;0) and (1;1;1;1;0;0;0;0;1;1;1;1).
Solution:

Filter sizes are applied in order mentioned in the exercise. For the fully-connected (FC) layer the positive sum is computed first.

Input	(0;0;0;0;1;1;1;1;0;0;0;0)
Conv_1:	(0;0;1;1;0;0;-1;-1;0;0) (0;0;-1;-1;0;0;1;1;0;0)
MaxPool:	(0;1;0;-1;0) (0;-1;0;1;0)
Conv_2:	(0;-4;0)
FC:	(-4;4)
Activation:	(0.018;0.982)

Input	(1;1;1;1;0;0;0;0;1;1;1;1)
Conv_1:	(0;0;-1;-1;0;0;1;1;0;0) (0;0;1;1;0;0;-1;-1;0;0)
MaxPool:	(0;-1;0;1;0) (0;1;0;-1;0)
Conv_2:	(0;4;0)
FC:	(4;-4)
Activation:	(0.982;0.018)

2. Convolutional Networks in practice. To start:

   - download and extract the archive 'assignment_cnn.tar.gz' from blackboard;
   - follow the instructions inside the 'README.txt'

   If you are unable to successfully setup your environment and run the ipython notebook then you can download the virtualbox image (hosted here: `https://surfdrive.surf.nl/files/index.php/s/EMYHsFb8X7mk11V`) where everythin is already configures.
   Consider the practical assignment of:

   (a) implementing layers of a CNN (ConvolutionalNeuralNetworks.ipynb)
   Solution:
   See submitted archive file.

   (b) modifying a simple architecture of a CNN to achieve higher accuracy on the MNIST dataset (mnist.ipynb)
   Solution:
   See submitted archive file