Machine Learning for Agricultural Applications

Assignment 5

Prof. Dr. Niels Landwehr
Dr. Julian Adolphs

Summer Term 2020

Release: 04.06.2020
Discussion: 11.06.2020

Task 1 – Convolution [25 points]

Write a python script that executes the convolution for an input image of height m, width l and channel size d. The filter has size $k \times k$ and also channel size d. Use random values for the image and filter entries. Use (only) the numpy library.

a) Start with the convolution without padding.

b) Now include padding with one margin of zeros around the input.

c) After the convolution execute an average pooling with kernel size 2 and stride 2.

d) Check your result by calculating the convolution and average pooling manually and comparing with the script output for $k, m, l, d = 2, 3, 3, 2$ and the image and convolution kernels

$$\text{image} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad \begin{bmatrix} 1 & 0 & 2 \\ 1 & 2 & 0 \\ 2 & 1 & 2 \end{bmatrix}, \qquad \text{conv.kernel} = \begin{bmatrix} 0 & 0 \\ 2 & 0 \end{bmatrix}, \quad \begin{bmatrix} 1 & 2 \\ 2 & 0 \end{bmatrix}$$

`Solution: ex5_convPaddPool.py`

d) Die gegeben Test-Matrizen mit np.random.seed(1) und m=3, l=3, d= 2, k=2, p=1.

```
image[:, 1:-1, 1:-1] = np.random.randint( 0, 3, size = (d, m-2*p, l-2*p))
filt = np.random.randint(0, 3, size = (d, k, k))
```

$$\text{convolution} = \begin{bmatrix} 0 & 4 & 0 & 4 \\ 2 & 5 & 10 & 2 \\ 2 & 9 & 6 & 4 \\ 4 & 4 & 5 & 2 \end{bmatrix}, \qquad \text{aver.pool} = \begin{bmatrix} 2.75 & 4.00 \\ 4.75 & 4.25 \end{bmatrix}.$$

Task 2 – ConvNet with Keras [25 points]

In this exercise, we use Keras to train a convolutional neural net. Load the fashion_mnist data set, which is included in Keras (Assignment 4). Use the sequential API in Keras to create a ConvNet, that is, a neural network consisting of multiple convolution layers followed by multiple fully connected layers. Use at least 3 conv-layers, 1 pooling layer and 3 dense layers. Train the model on the training part of the fashion_mnist data set and

evaluate it on the test part of the data set. Try to improve the model to reach a test accuracy of at least 92% (use metrics = 'accuracy'). Play with the number of conv layers, dense layers, nodes and number of training epochs.

Hint: Use `batch_size = 32` to speed up calculations.

`Solution:` `ex5_keras_cnn.py`
