Functional Programming - Lab Class Exercises 6

Frank C Langbein
frank@langbein.org

Version 1.4.0

This lab class set simply provides simple example code for the topics captured in the lecture. Suggestions for expanding the code are given, but these are optional and no solutions will be provided.

1. Sobel Gradient

`lab6-sobel.py` is python code based on (Py)Torch that computes the discrete Sobel gradient, which consists of two convolutions and calculating a norm. Similar convolutions can be implemented in the same way.

You could try to turn this into a neural network to learn the Sobel gradient operation from a large set of training examples (generated with this code) or even try to make the network learn to detect edges, using the Canny edge detector. For what we covered in the lecture, this is quite advanced, but shows a different example of how a program could be trained/optimised to implement an operation.

You can also try to convert the code to a tensorflow implementation.

2. Image Classification with CNNs

`lab6-cnn.py` is a simple CNN to classify images for a standard dataset. It shows how to realise basic CNNs in tensorflow.

The network is not optimal and you could try to modify it to perform better on the training dataset. Increasing the number of epochs alone is not enough, but the network structure needs to be adjusted (e.g. by adding more convolutional layers or filters, etc).

An advanced option may also be to expand the network into an autoencoder.