DATA.ML.200 Pattern Recognition and Machine Learning

Exercise Set 4: Detector evaluation and autoencoder denoising

1. **python** *ROC analysis.* (2 points)

   Load the files "detector_groundtruth.dat" and "detector_output.dat" that contain the ground-truth values (1: positive, 0:negative) and detector outputs, respectively.

   Write code that calculates and plots the ROC curve of true positive rate (recall) vs false positive rate (1-precision). Read more from the "Receiver operating characteristics" Wikipedia page.

2. **python** *Noisy MNIST Fashion classification.*

   Write code that loads and normalizes the data (see the lecture notebook). Add similar noise to the lecture example to the both training and test images:

```
noise_factor = 0.2
train_images_noisy = train_images + noise_factor * tf.
   random.normal(shape=train_images.shape)
test_images_noisy = test_images + noise_factor * tf.random
   .normal(shape=test_images.shape)

# Make sure values still in (0,1)
train_imagse_noisy = tf.clip_by_value(train_images_noisy,
   clip_value_min=0., clip_value_max=1.)
test_images_noisy = tf.clip_by_value(test_images_noisy,
   clip_value_min=0., clip_value_max=1.)
```

3. Define a CNN classifier model for the MNIST Fashion and train with clean images

4. Report classification accuracy for the clean test images (2 pts)

5. Report classification accuracy for the noisy test images (2 pts)

6. Define a CNN autoencoder model to denoise the noisy images and train it using noisy training images - plot examples (2 pts)

7. Report classification accuracy for autoencoder denoised test images (2 pts)

8. Train the original model with noisy train images and report accuracy for noisy test images (2 pts)