Examination for the "Inverse problems and machine learning" module
==========
The examination consists of the following steps.

1. Make sure you have access to a computer with a CUDA capable GPU. On that computer, follow instructions in "Installation.txt" to set-up your environment and install PyTorch, ODL, ASTRA Toolbox and SciKit-image. Note that instructions are written for Linux, but you should be able to use Windows as well.

2. Work through the examples in the iPython notebook "learned_reconstruction_pytorch.ipynb". This notebook includes the following:
(a) Simulate 2D parallel-beam tomographic data from 28 x 28 pixel images representing MNIST handwritten digits. Note: The reason for choosing such small 2D images is to ensure the training (for the learned methods, see (c) below) is doable on modest computing hardware within reasonable time.
(b) Set-up and perform reconstruction using two model based methods: Filtered back-projection (FBP) and total variation
(c) Set-up and perform reconstruction using two learned methods, both trained against suitable supervised data with the square 2-norm as loss. The two learned methods are: (i) learned post-processing of FBP (=convolutional neural network trained to denoise/restore a FBP reconstruction) and (ii) learned gradient descent (=unrolled gradient descent), which is described in slides 23-24 (=pages 83-85) for lecture2.

3. Implement learned primal-dual (without memory), see description in slides 25-26 (=pages 86, 88-91) for lecture2. The solution (training + test) should be contained in a iPython notebook that can be executed in an environment that is set-up as in (1) above. Make sure you train and test on same data as unrolled gradient descent in 2(c)(ii) above, see the iPython notebook "learned_reconstruction_pytorch.ipynb" for that. Share you solution by providing me (Ozan Öktem, e-mail: ozan@kth.se) with a link to a suitable repository (like GitHub) for the iPython notebook that contains the learned primal-dual implementation.