UNIVERSITÄT HEIDELBERG
Zukunft. Seit 1386.
IWR Interdisciplinary Center for Scientific Computing

Prof. Björn Ommer
ommer@uni-heidelberg.de

Exercise sheet 11: GANs and Inverting CNN Features

Due on *26/01/2018*, 2pm.

Question 1: Generating images with Generative Adversarial Networks (7P)

In this task you will implement a simple Generative Adversarial Network (GAN) using PyTorch.

a) Implement a Generator Network.

	Generator
Linear	$Z \rightarrow 128$
	Relu
Linear	$128 \rightarrow 784 = H \times W \times N_c$
	Sigmoid

Where Z is a vector of random noise with 100 dimensions. H, W denote the image width and height and N_c denotes to number of image channels. In case of MNIST dataset $H \times W \times N_c = 28 \times 28 \times 1$

Initialize weights and biases of the network with values $\sim \mathcal{N}(0,\, 0.075^2)$.

b) Implement a Discriminator Network.

	Discriminator
Linear	$H \times W \times N_c \rightarrow 128$
	Relu
Linear	$128 \rightarrow 1$
	Sigmoid

Initialize weights and biases of the network with values $\sim \mathcal{N}(0,\, 0.075^2)$.

c) Train your Generator and Discriminator on the MNIST dataset using a batchsize of 64 and an ADAM solver with learning rate of 0.001 for at least 10000 iterations. Visualize 64 randomly generated images every 1000 iterations. (5P)

d) Generate a set S of 2000 fake images. Pick 2 real image from each class. Find 4 nearest fake images from the set S for each of the picked images. (Use Euclidean distance on the pixel values of the images as the distance metric). Visualize and discuss the results.

You can use a provided starter code in `gan-task.ipynb`. (2P)

Question 2: Inverting the CNN features **(3P)**

You are provided with implementation of the CNN image representation inversion method by Mahendran and Vedaldi, 2014 in `inversion.py`.

Play around with the parameters of the method to better understand it.

a) Invert representations of the provided images for every convolutional and max-pooling layer of the `vgg11` network. Indices of all the layers can be printed with `print(model)`.

   Visualize the results. What do you observe when moving from shallow layers to the deep? (2P)

b) Total Variation is used as image regularization in this method. What happens with inverted image if we decrease the weight of the Total Variation `tv_lambda`? (1P)

Universität Heidelberg
Zukunft. Seit 1386.
IWR Interdisciplinary Center for Scientific Computing

Artificial Intelligence

Wintersemester 17/18

Prof. Björn Ommer
ommer@uni-heidelberg.de

Note: Submit exactly one ZIP file and one PDF file via Moodle before the deadline. The ZIP file should contain your executable code. Make sure that it runs on different operating systems and use relative paths. Non-trivial sections of your code should be explained with short comments, and variables should have self-explanatory names. The PDF file should contain your written code, all figures, explanations and answers to questions. Make sure that plots have informative axis labels, legends and captions.