Exercises
**Computational Intelligence Lab**
SS 2017

**Machine Learning Institute**
Dept. of Computer Science, ETH Zürich
**Prof. Dr. Thomas Hofmann**
Web http://cil.inf.ethz.ch/

# Series 9, May 11-12, 2017
# (Convolutional Neural Networks & Generative Models)

**Problem 1 (CNN computations):**

In this exercise we consider a convolutional neural network made of one convolutional layer, followed by a non-linearity ReLU, and then followed by a 3x3 average-pooling without stride, on top of which we apply a softmax. Assume that this small CNN takes as input images with two channels, and that the convolutional layer has two filters, each filter channel containing an odd number of pixels, so that we can index it with pixels $(i, j)$ for $-k \leq i, j \leq k$. Further assume that convolutions are performed with *zero-padding*, i.e. by completing $I$ with zeros in such a way that the convolution of $I$ by a filter has the same size as $I$. *Hint: each filter can be seen as an image with three channels*.

1. Write down the mathematical expression of the convolutional layer, the ReLU non-linearity, the max-pooling.
2. Write down the mathematical expression of the ReLU non-linearity and the max-pooling. What do you get by composing it with the previous question? This is the signal you get just before applying the softmax.

   In the following questions, consider a 2-channels image $I = (I_c)_{1 \leq c \leq 2}$, where each $I_c$ is of size $4 \times 4 = 16$ pixels. Assume that each of the two filters $K^l$, for $l = 1$ or $l = 2$, is a 2-channel image of size $3 \times 3 = 9$ pixels per channel. We choose the image to have pixel values given by $(I_c)_{i,j} = ij + c$, and the filters to be defined as

   $$K_c^l = \begin{pmatrix} 0 & l & 0 \\ l & -c & l \\ 0 & l & 0 \end{pmatrix},$$

   where $c \in \{1, 2\}$ is the channel and $l \in \{1, 2\}$.
3. Compute the two 4x4 1-channel convolved images $I \star K^l$ for $l \in \{1, 2\}$. (Computing all coefficients by hand is tedious but do at least the first 4 ones, in the top left corner).
4. Write down the result of applying a non-linearity ReLU to each of the convolved images.
5. Apply a 3x3 max-pooling without stride to each of the two images obtained in the previous question (compute at least the first coefficient in the top left corner).

**Problem 2 (Road Extraction from Satellite Images):**

For the third choice of semester project task, we provide a set of satellite/aerial images acquired from GoogleMaps. We also provide ground-truth images where each pixel is labeled as {road, background}. Your goal is to train a classifier to segment roads in these images, i.e. assign a label {road=1, background=0} to each pixel.

1. Download the training data from the competition website

   inclass.kaggle.com/c/cil-road-segmentation.

2. Obtain the python notebook `segment_aerial_images.ipynb` from

   github.com/dalab/lecture_cil_public/tree/master/exercises/ex9

   to see example code on how to extract the images as well as pixel labels.

   The notebook shows how to use `scikit learn` to generate features from each pixel, and finally train a linear classifier to predict whether each pixel is road or background. It also provides helper functions to visualize the images, labels and predictions.

3. As a more advanced approach, try `tf_aerial_images.py`, which demonstrates the use of a convolutional neural network in TensorFlow for the same prediction task.

**Problem 3 (Galaxy image classification):**

For the fourth choice of semester project task, we provide a set of cosmology images observed by astronomical telescopes, corrupted cosmology images and images with other content. To get you started in exploring the data, we will consider here the problem of discriminating real cosmology images from the other two groups.

1. Download the supplementary data from the competition website.

   inclass.kaggle.com/c/cil-cosmology-2017.

2. Read documentation on the supplementary data as needed for this task.

3. Obtain the Python script `features_cosmology_project.py` from

   github.com/dalab/lecture_cil_public/tree/master/exercises/ex9

   which provides example code for loading the images, extracting basic features from them, and a skeleton for evaluating machine learning models.

4. Get familiar with the cosmology data by solving a simple binary classification problem: Each image should be assigned a label from the set $\{\text{real cosmology} = 1, \text{non real cosmology} = 0\}$. For this, divide the images in the folder `labeled` into training and test set. Implement hand-crafted image features + baseline classifiers with `Scikitlearn`, or learn image features automatically with a deep learning approach using `Tensorflow`. Evaluate and contrast your models' predictive performance and get a feeling for which features are informative.