Assignment 3. CNN with MATLAB

This assignment concerns a programming exercise with MATLAB.

Your task will be to train a neural network to recognize handwritten digits presented in the MNIST dataset.

Installation

1. Acquire and install the MATLAB 2017 b or 2018 a.
2. Download new MATLAB packages from https://www.mathworks.com/matlabcentral/fileexchange/59133-neural-network-toolbox-tm--model-for-alexnet-network.
3. Open MATLAB and start a new script, e.g., AlexNetModel.m
4. Explore the CNN method using the support material from https://www.mathworks.com/help/nnet/ref/alexnet.html
5. In a command window of MATLAB, enter `alexnet` and then `netLayers`
6. If installation went without errors, the following screen should show up:

```
Command Window
>> alexnet

ans =

  SeriesNetwork with properties:

    Layers: [25×1 nnet.cnn.layer.Layer]

>> net.Layers

ans =

  25x1 Layer array with layers:

     1   'data'     Image Input                  227x227x3 images with 'zerocenter' normalization
     2   'conv1'    Convolution                  96 11x11x3 convolutions with stride [4  4] and padding [0  0]
     3   'relu1'    ReLU                         ReLU
     4   'norm1'    Cross Channel Normalization  cross channel normalization with 5 channels per element
     5   'pool1'    Max Pooling                  3x3 max pooling with stride [2  2] and padding [0  0]
     6   'conv2'    Convolution                  256 5x5x48 convolutions with stride [1  1] and padding [2  2]
     7   'relu2'    ReLU                         ReLU
     8   'norm2'    Cross Channel Normalization  cross channel normalization with 5 channels per element
     9   'pool2'    Max Pooling                  3x3 max pooling with stride [2  2] and padding [0  0]
    10   'conv3'    Convolution                  384 3x3x256 convolutions with stride [1  1] and padding [1  1]
    11   'relu3'    ReLU                         ReLU
    12   'conv4'    Convolution                  384 3x3x192 convolutions with stride [1  1] and padding [1  1]
    13   'relu4'    ReLU                         ReLU
    14   'conv5'    Convolution                  256 3x3x192 convolutions with stride [1  1] and padding [1  1]
    15   'relu5'    ReLU                         ReLU
    16   'pool5'    Max Pooling                  3x3 max pooling with stride [2  2] and padding [0  0]
    17   'fc6'      Fully Connected              4096 fully connected layer
    18   'relu6'    ReLU                         ReLU
    19   'drop6'    Dropout                      50% dropout
    20   'fc7'      Fully Connected              4096 fully connected layer
    21   'relu7'    ReLU                         ReLU
    22   'drop7'    Dropout                      50% dropout
    23   'fc8'      Fully Connected              1000 fully connected layer
    24   'prob'     Softmax                      softmax
    25   'output'   Classification Output        crossentropyex with 'tench', 'goldfish', and 998 other classes
fx >>
```

AlexNet: testing and experiments

You will be following the description as presented in the tutorial: https://www.mathworks.com/help/nnet/ref/alexnet.html and perform the following steps:

1. Classify an image
2. Perform feature extraction
3. Use the features extracted from the training images as predictor variables
4. Classify test images
5. Transfer learning – transfer layers to new network
6. Train the new network
7. Create a new dataset by using, for example, a camera:

```
camera = webcam; % Connect to the camera
picture = camera.snapshot; % Take a picture
```

8. Collect analytical data for training progress.

In conclusion, analyze the results.

Submission

You can work in small groups up to three students.

Submit the report to the Beachboard by October 18, 11:59pm.