Image and Video Processing - Lab 3: Deep Learning (Fully Connected Networks)

Friday 13th May, 2022

The goal of this lab is to implement a Deep Learning-based classifier made of fully-connected layers. More in detail, the classifier should be able to correctly classify the content of a set of images containing handwritten digits. ***Note:*** *to work at home you need at least Matlab r2017b so that you can use the Neural Network toolbox.*

1. Load the *DigitDataset* which is made of 28×28 pixels digit images:
   - Obtain the path of the dataset with the command `fullfile(matlabroot,'toolbox','nnet','nndemos','nndatasets','DigitDataset')`
   - Load the dataset at the path you obtained at the previous step with the command `imageDatastore`; in order to correctly label the elements in the dataset, pass the following parameters to the function: `'IncludeSubfolders',true` and `'LabelSource','foldernames'`
   - Split the dataset in training and validation sub-datasets by using the function `splitEachLabel`. For each class, use 75% of the elements for training and the remainder for validation (**Note**: use the `randomize` option to randomly shuffle the elements)
2. Using the Matlab Neural Network toolbox, design a fully-connected network for digit classification. The network should contain:
   - 3 fully-connected layers (e.g. with output size 100 for the first two ones and of size 10 for the last one)
   - *ReLu* activation units (one for each layer)

   The network should be trained using Stochastic Gradient Descent (SGD) and the softmax cross-entropy loss function. Initially set the batch size to 200, the learning rate to 0.001, and train for a total of 30 epochs (**Note**: in order to avoid automatic early-stop during training, pass the parameter `'ValidationPatience',Inf` to the `TrainingOptions` function).
3. Now you can test the performance of the network:
   - Evaluate the validation accuracy of the network on the validation dataset (**Hint**: compute the percentage of correct predictions over the total size of the validation dataset)
   - Discuss the behavior of the accuracy and the loss during training
   - Select an image from the validation dataset, show the image, and use the network to obtain the classification of that image: is the classification correct?
   - Discuss the network training process behavior when changing:
     - the number of fully-connected layers (e.g. reduce to 2, increase to 4)
     - the size of the fully-connected layers (e.g. reduce the number of output neurons)
     - the batch size (e.g. reduce to 50)
     - the learning rate (e.g. increase to 0.1
   - Finally, change the network parameters (e.g. number of layers, amount of training data) to purposely make it overfit