Practical block II: Object Classification using CNN

Objective of the practical session

Learn how to perform object classification using deep learning methods

Please note that the session will be performed using Matlab and its libraries. The software is available in any PC of the laboratory room. Code using other programming languages will not be evaluated.

For this session the following toolboxes are required: Deep Learning, Computer Vision, Parallel computing, Image processing

The practical session can be done in pair of students, and to be evaluated it is sufficient that one of the two members of the group upload the delivery to the campus virtual.

1. Introduction

Convolutional neural networks are essential tools for deep learning and are especially suited for image recognition. This practical session aims at showing how to create and train a simple convolutional neural network for deep learning classification. The pipeline consists in load image data, define the network architecture, specify training options, Train the network and Predict the labels of new data and calculate the classification accuracy. The session will focus on understanding the role of the main tuning parameters available.

2. Practical session

The practical session of the block I consist in classifying synthetic images of handwritten digits using CNN. We would like to reproduce the sequence classification pipeline, illustrated in the following matlab example:

```
openExample('nnet/TrainConvNetWithRotatedImagesExample')
```

however, please download the file “CNN_classification_2020_v1.mlx” from the campus virtual. This script is a modified version of the “Train Network with Augmented Images” available in matlab.

The dataset consist of a 4-D array data-set having 5000 images of handwritten digits 28-by-28-pixel wide.

The 5000 image data-set is divided in 3 groups:
a Training-set having of "Training_n_samples" images
the remaining images are equally divided in Validation and Test sets

Data augmentation helps prevent the network from overfitting and memorizing the exact details of the training images. The function `imageDataAugmenter`, initially disables will be used to learn the impact of the data augmentation on the network performances.
Two simple networks are proposed, a first one (layers_v1) include a single convolutional layer, the second one (layers_v1) is a deeper network and consists of four convolutional layers

Session task:

1. Compute the baseline training

Execute for the first time the script, understand the plot curves and observe the computed accuracy.
At this time the data augmentation is not activated, the selected network is _v1, the number of epocs is 20, and the number of samples used for training is 4000.

Observe the Test accuracy computed at the end, observe the validation accuracy on the plot.

2. Simulating lack of data

We are simulating the lack of data, by reducing the number of images in the training set to 1000, then to 500, then to 300. Please note that the "epocs_factor" is computed to compensate the limited number of samples.

What do we observe in the performance? How the training and validation curve behave? What happens if we set "epocs_factor" to 1?

3. Influence of Batch normalization

The batch normalization improves the convergence of the training.
Set the training-set to 1000, and comment the %batchNormalizationLayer in layers_v1
Then
Set the training -set to 300, and comment the %batchNormalizationLayer in layers_v1

What do we observe in the performance? How the training and validation curve behave? What happens if we set "epocs_factor" to 1?

Then restore the "batchNormalizationLayer"

4. Influence of Batch normalization

Set the training-set to 300, and replace layers_v2 as network to train in trainNetwork(...);

How the training and validation performance change?

5. Define a data-augmentation strategy

Set the training-set to 300, and activate the line (and comment the previous line)

```
%augimds =
augmentedImageDatastore(imageSize,XTrain,YTrain,'DataAugmentation',imageA
ugmenter);
```

And in `imageDataAugmenter(...) set`

'RandRotation' only

What do you observe?

activate both 'RandRotation'and 'RandXTranslation'

What do you observe?

activate 'RandRotation', 'RandXTranslation' and 'RandYTranslation'

What do you observe?

activate 'RandRotation', 'RandXTranslation' , 'RandYTranslation', 'RandYReflection'

What do you observe?

activate 'RandRotation', 'RandXTranslation' , 'RandYTranslation', 'RandYReflection' and RandXReflection

What do you observe?

3. Deliverable

A **SHORT** report in PDF format containing around five pages must be delivered for the practical block. The matlab code of the session should be delivered as well. Compact everything in a single ZIP file.

IMPORTANT, only files delivered using the campus virtual will be evaluated. Please upload in the corresponding task your file before the dead line
IN CASE YOU HAVE NO ACCCESS TO THE CAMPUS VIRTUAL please contact the professor of the practical course to solve the issue.