Learning with Artificial Neural Networks

Practical Work 04 – Deep Neural Networks

Professor: Andres Perez-Uribe
Assistants: Yasaman Izadmehr, Simon Walther
Emails : prenom.nom@heig-vd.ch

Goals:

- Having a better understanding of the differences between a shallow and a deep neural network
- Understanding the fondamentals of convolutional neural networks
- Learning the basics of the Keras framework

1. Introduction

In this practical work, you will use three differents way to classify digits images from the MNIST dataset : MLP, MLP from Histogram Of Gradients (HOG) and convolutional neural network (CNN). Then you will create a CNN to classify images of fashion categories from the Fashion MNIST dataset.
Furthermore, you will now make use of a framework called Keras. Keras is a high-level neural networks library, written in Python and capable of running on top of TensorFlow.

2. Digit recognition from raw data

The objective of this exercise is to train a shallow neural network using the raw pixel data of the MNIST digit database. Each digit input is an image of 28x28 pixels and there are 10 classes: digits 0 to 9.
Study the notebook MLP_from_raw_data.ipynb and play with the code to solve this benchmark classification task. Compare the results obtained by various neural network configurations and diverse parameters. Select a final model (e.g., the one with better performance) for analysis (please, see below the summary of work to know what to include in the report).

3. Digit recognition from features of the input data

The objective of this exercise is to train a shallow neural network using features computed from the raw pixel data of the MNIST digit database. Instead of using as input the 28x28 pixel images, we compute the Histogram of gradients (HOG) features of parts of the image (e.g., sliding windows) and use those features as inputs to the neural network.
Study the notebook MLP_from_HOG.ipynb and play with the code. Compare the results obtained by using various neural network configurations, e.g., diverse parameters of the learning and of the feature extraction phase. Finally compare the results with the previous section and select a final model (e.g., the one with better performanceI) for analysis (please, see below the summary of work to know what to include in the report).

4. Convolutional neural network digit recognition

The objective of this exercise is to train a deep convolutional neural network capable of "automatically" determining the features (i.e., via the set of convolutional kernels trained by supervised learning) that allow it to properly recognize the digits 0 to 9. Study the notebook CNN.ipynb and play with the code. Compare the results obtained by using various neural network configurations, filter sizes, number of filters per layer, using or not dropout and compare the obtained results with the previous two sections. Select a final model (e.g., the one with better performancel) for analysis (please, see below the summary of work to know what to include in the report).

5. Fashion MNIST experiment

The last part of this practical work is to use your newly acquired knowledge on another dataset. This dataset aims to be an harder dataset than MNIST while remaining completely interchangeable with it : it keeps the seems image size, it is still grayscale and has the same number of classes.
The task is to train a convolutional neural network to classify images into differents fashion categories. For this experiment, we don't provide you a notebook but you can reuse most of the code from the CNN notebook.

Report

1. What is the learning algorithm being used to optimize the weights of the neural networks? What are the parameters (arguments) being used by that algorithm? What cost function is being used please, give the equation(s)
2. Model complexity: for each experiment (shallow network learning from raw data, shallow network learning from features, CNN, and Fashion MNIST), select a neural network topology and describe the inputs, indicate how many are they, and how many outputs. Compute the number of weights of each model (e.g., how many weights between the input and the hidden layer, how many weights between each pair of layers, biases, etc..) and explain how do you get to the total number of weights.
3. Do the deep neural networks have much more "capacity" (i.e., do they have more weights?) than the shallow ones? explain with one example
4. Test every notebook for at least three different meaningful cases (e.g., for the MLP exploiting raw data, test different models varying the number of hidden neurons, for the feature-based model, test pix_p_cell 4 and 7, and number of orientations or number of hidden neurons, for the CNN, try different number of neurons in the feed-forward part) describe the model and present the performance of the system (e.g., plot of the evolution of the error, final evaluation scores and confusion matrices). Comment the differences in results. Are there particular digits that are frequently confused?