MSE | MASTER OF SCIENCE IN ENGINEERING

HES-SO

Machine Learning

Practical work 11 - Convolutional Neural Networks

Teachers: A. Perez-Uribe (Email: andres.perez-uribe@heig-vd.ch) & J. Hennebert
Assistants: Benoît Hohl (Email: benoit.hohl@heig-vd.ch) and Simon Walther (Email: simon.walther@heig-vd.ch)

Summary for the organization:

- Submit the solutions of the practical work before Tuesday 04.12.23 23h59 via Moodle.
- Modality: PDF report (max. 10 pages)
- The file name must contain the number of the practical work, followed by the names of the team members by alphabetical order, for example 10_dupont_muller_smith.pdf.
- Put also the name of the team members in the body of the notebook (or report).
- Only one submission per team.

Notebooks and libraries

Download the notebook material from the Moodle platform and use Google colab, or install TensorFlow and Keras libraries. Keras is a high-level neural networks library, written in Python and capable of running on top of TensorFlow.

Setting up the Google Colab environment

1. Create a Google account if you do not already have one
2. Upload the following files to your Google Drive account. Put them in a folder called "Colab Notebooks":
   - MIP_from_raw_data.ipynb
   - MIP_from_HOG.ipynb
   - CNN.ipynb
3. Open the notebooks using colab (right click->Open with->Colaboratory)
4. On the menu, click on **Runtime** and then on **Change runtime type**
5. Select:
   - Runtime type = Python 3
   - Hardware accelerator = GPU
6. You are now ready to play with the notebooks

Exercices

This practical work has the objective of introducing the use of Convolutional Neural Networks to train classification models (e.g., digit recognition from images), to compare those models with classical “shallow” neural networks exploiting the raw images and features of the raw images, and finally to setup a model selection methodology. For each point below, use the available annotated data to create an independent test dataset (e.g., 20% of the whole dataset), a training dataset (e.g., 80% of the remaining data or 64% of the dataset) and a validation dataset (e.g., 16% of the dataset).

1. 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. Model selection: Compare the performance obtained by various neural network configurations and diverse parameters (hyper-parameter tuning). During this phase use a train dataset and evaluate the models using a validation dataset. 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) and evaluate it using the test dataset.

2. 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. Model selection: Compare the performance obtained by various neural network configurations and diverse parameters (e.g., learning and feature extraction parameters). During this phase use a train dataset and evaluate the models using a validation dataset. 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) and evaluate it using the test dataset.

3. 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. Model selection: Compare the performance obtained by various neural network configurations and diverse parameters (filter sizes, number of filters per layer, using or not dropout, etc). During this phase use a train dataset and evaluate the models using a validation dataset. 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), evaluate it using the test dataset, and compare the obtained results with the previous two sections.

Summary of work to include in the report

1. What is the learning algorithm being used to train the neural networks? What are the parameters (arguments) being used by that algorithm? What cost function is being used? please, give the equation(s) and describe (e.g., please include your code for this part) how did you create the training, validation and test datasets.

2. Model complexity: for each experiment (shallow network learning from raw data, shallow network learning from features and CNN), 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?

Comments:

Très intéressante exploration des CNNs

Do the deep neural networks have much more “capacity” (i.e., do they have more weights?) than the shallow ones? Explain with one example:
Comme vous avez pu le constater avec vos calculs, un deep neural network (type CNN) n'a pas forcément plus de poids qu'un nn avec un seule couche cachée 795’010>60317... C'est parce que t ous les neurons d'une couche sont connectés à tous les neurones de la prochaine couche dans le cas d'un simple MLP.

Très bon rapport, félicitations !

Grade: Pass