Machine Learning

Practical work 02 - Introduction to Neural Networks and Supervised Learning

Teacher: A. Perez-Uribe (Email: andres.perez-uribe@heig-vd.ch)
Assistants: H. Satizabal (Email: hector-fabio.satizabal-mejia@heig-vd.ch),
J. Despraz (Email: jeremie.despraz@heig-vd.ch)

Introduction

During the previous practical work sessions, we provided you with a series of notebooks to explore the workings of an artificial neuron (e.g., the Perceptron model), neural networks (multi-layer perceptrons or MLPs) and Backpropagation, which allows those system to "learn from examples".

The subsequent practical work consisted in applying a methodology for evaluating the performance of the trained neural networks on new data (e.g., hold-out validation and cross-validation), with the aim of selecting a final model to deal with the problem at hand.

The selection of a model consists in finding the model of appropriate complexity (e.g., the right number of parameters or synaptic weights) and configuration (e.g., type of activation function, learning rate, momentum, training iterations, etc).

Practical work

To evaluate a given neural network topology (e.g., 1 hidden layer, 2 hidden neurons, tanh as activation function) and configuration (e.g., learning rate = 0.001, momentum = 0.7) we need to split the dataset to define a training and a test subsets, for instance by randomly choosing 80% of the data for training, and 20% for test.

Then, we train the neural network iterating over the training data several times (a certain number of epochs).

However, performing this only once might not give a good idea of the generalization capability of the trained neural network, since the 80%-20% split for training and testing is random. Therefore, we usually split the dataset several times (e.g., N_SPLITS) and compute a mean of error to assess the performance of the trained neural network.

But, every time we train and test a neural network on a given dataset, we start with random weights, thus evaluating it based on a single trial is not enough and, therefore, we have to train several times (N_INITS) the neural network randomly initializing the weights every time.

1. Explore the "hold_out_validation" notebook

Q1. Determine where all the aforementioned parameters are defined.

Observe that we run the evaluation procedure on four different problems. Each problem is a two-class two-dimensional problem, where the two sets overlap more and more (the synthetic datasets are randomly generated using variances of 0.4, 0.5, 0.6 and 0.7).

Q2. What are the cyan and red curves in those plots ? Why are they different ?

Q3. What happens with the training and test errors (MSE) when the 2 sets overlap significantly ?

Q4. Why does the red curves sometimes indicate a higher error than the cyan ones ?

Q5. What is the boxplot summarizing the validation errors of the preceding experiments showing ?

2. Explore the "cross_validation" notebook

Q1. Determine where all the aforementioned parameters are defined.

Q2. What is the difference between hold-out and cross-validation ? What is the new parameter that has to be defined for cross-validation ?

Q3. Observe the boxplots summarizing the validation errors obtained using the cross-validation method and compare them with those obtained by hold-out validation.

3. Speaker recognition experiments

You will be provided with a database of vowels spoken by men, women and children (of 3, 5, and 7 years old). The task will be to train artificial neural networks to recognize the speakers and evaluate those models performance (e.g., by cross-validation).

The file vowels.zip contains the sounds in WAV format. They have been collected by the team of Prof. Peter Assmann of the School of Behavioral and Brain Sciences at the University of Texas at Dallas. You will also find a set of synthetic sounds corresponding to each vowel and each speaker. Please read the file 0_README.txt for more information.

Typical features of speech are timbre, pitch, intonation and tempo. For this practical work, we will use the Mel-Frequency Cepstrum Coefficients (MFCC) which have been found to be very useful for speech recognition. In sound processing, the mel-frequency cepstrum (MFC) is a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency. You can compute the MFCC coefficients using available software like *Jaudio* or *praat*, or compute them using *python-speech-features*, a set of python modules for speech/signal processing.

We provide an example notebook showing you how to read the wave files and how to compute the MFCC coefficients using the *python-speech-features* package. Consider that each given sound is splitted into multiple temporal "windows" and that we compute the MFCC for each window. Thus, for every sound, you will get 13 MFCC coefficients per window. You can now compute features characterizing those values, e.g., the mean and the standard deviation.

Report

0. For each of the following experiments, provide a brief description of the number of observations of each class, the features being used to train the model, the detailed procedure for selecting the final model (e.g., ''We used the `<my_notebook>.ipynb` notebook''), the description of the final model and its evaluation (provide the cross-validation results, the confusion matrix and the F-score). Comment your results.

1. Man vs Woman. Use only the natural voices of men and women to train a neural network that recognizes the gender of the speaker.

2. Woman vs. children. Proceed as explained in 0 but using only woman and children voices.

3. Man vs. Woman vs. children. Proceed as explained in 0.

4. Natural vs. synthetic voices. Proceed as explained in 0.

5. Design a final experiment of your choice (e.g., using your own voice). Proceed as explained in 0.

Summary for the organization:

- Submit the report and notebook(s) of the practical work before Thursday 18.04.2019, 23h55 via Cyberlearn.
- Modality: PDF report (max. 6 pages + cover page)
- The file name must contain the number of the practical work, followed by the names of the team members in alphabetical order, for example:

  `MLG_PW2_dupont_muller.pdf`
- Write also the name of the team members in the body of the report.
- Only one submission per team.