MASTER OF SCIENCE
IN ENGINEERING

HES-SO

Machine Learning

Practical work 11 - Recurrent Neural Networks

Teachers: A. Perez-Uribe (Email: andres.perez-uribe@heig-vd.ch) & J. Hennebert
Assistant: H. Satizabal (Email: hector-fabio.satizabal-mejia@heig-vd.ch)

Summary for the organization:

- Submit a report before Monday 18.12.17 23h55 via Moodle.
- Modality: PDF report (max. 6 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 11_dupont_muller_smith.pdf.
- Put also the name of the team members in the body of the report.
- Only one submission per team.

0. Notebooks and libraries

Download the notebook material from the Moodle platform and the databases.

1. Types of sequence-prediction problems

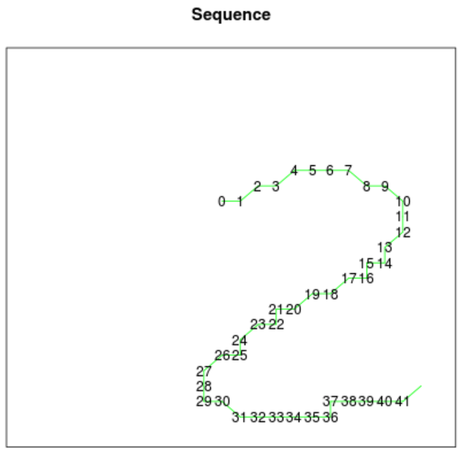
Jason Brownlee on the “Machine learning mastery” website presents a pretty simple examples of the 5 types of sequence prediction tasks that can be solved with Recurrent Neural Networks. Carefully read this page and follow the examples in Python.

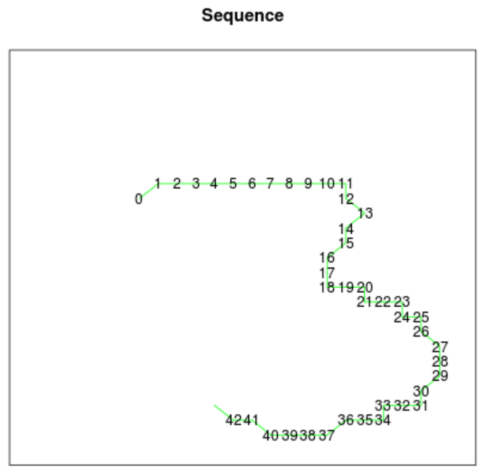
https://machinelearningmastery.com/sequence-prediction-problems-learning-lstm-recurrent-neural-networks/

Nothing to include in the report regarding this point.

2. Towards sequence recognition

The objective of this exercise is to train a so-called Simple Recurrent Neural Network capable of recognizing the digits 0 to 9. This time, the neural network does not have access to the 2D image of the digit, but to the pen stroke sequences (see example below).





The first notebook called **mnist_stroke_normalization.ipynb** is used to explore the pen stroke database and to normalize the length of the sequences, given that some digits are longer to draw than others (e.g., a one and an eight).

The second notebook called **test_stroke_dense.ipynb** trains a Multilayer Perceptron (MLP) using the sequence of strokes as input to the network. If we consider an online application, each (x,y) point of the stroke sequence arrives at a given time, thus we shall store the sequence and only when it is finished, we can ask the MLP to recognize the digit.

Run the notebooks and analyse the results obtained. To be more precise, answer to the following questions:

- What can you conclude by analyzing the histogram of stroke length ?

- What do you observe when visualizing the strokes, ordered by length ?

- Why do we need to normalize the stroke lengths to train the MLP model ?

- We run several experiments and decided to set some parameter values (number of layers, of hidden neurons, learning parameters, etc), but there might be better configurations… without an exhaustive exploration, we cannot be sure. Try other configurations and parameter values to check if you happen to find a better model.

- How do we evaluate the results of these models ?

- Is your model overfitting ? observe the plot of loss evolution for training and test.

- What do you observe in the confusion matrix ? are there digits that are frequently confused?

- What is the difference in performance (e.g., accuracy) between the models you obtained when attempting to find a better configuration ?

3. Recurrent Neural Networks

The third notebook called **test_stroke_lstm.ipynb** uses a Long Short Term Memory (LSTM) network to recognize the pen stroke sequences of the digits 0 to 9.

Run the notebook and compare the results with those obtained in the previous point.

- Try to understand the amount of parameters (weight connections) being used by the LSTM model and explain how to arrive to the values given by Keras when you create your model.

- Indicate what are the parameters (learning rate, momentum, dropout, etc) being used to train the model and the sort of activation functions being used. Give their expression.

- Does the LSTM model performing better than the MLP one ? consider the accuracy and the resulting confusion matrices.

- Is your model overfitting ?

- What is the difference in model complexity (e..g, number of weight values) between the MLP model and the LSTM model ? which one is preferable ?

4. Time-series forecasting tutorial online

Follow the tutorial on “Time-series forecasting” by Jason Brownlee on the “Machine learning mastery” website. You will learn how to prepare data, develop, and evaluate an LSTM recurrent neural network for time series forecasting. It is a fairly simple example that uses a small database to start with LSTM recurrent neural networks.

https://machinelearningmastery.com/time-series-forecasting-long-short-term-memory-network-python/

Nothing to include in the report regarding this point.

5. Sequences of strokes

The purpose of this exercise is to predict the following value of a sequence. The dataset used in this exercise contains 3-dimensional samples of horizontal and vertical pen speed and pen force of a single user writing some characters. Each character sample is a 3-dimensional pen tip velocity trajectory. This is contained in matrix format, with 3 columns and T rows where T is the length of the character sample. The original dataset was published by Ben H. Williams from the School of Informatics at the Univesity of Edinburg, an can be downloaded from the UCI machine learning repository (http://archive.ics.uci.edu/ml/datasets/Character+Trajectories).

Complete the given notebook in order to train 3 parallel models for predicting each of the 3-dimensional variables (x, y, f) using the recent history of the 3-D sequence. For instance, the first model will predict:

x_hat (t+1) as a function of x(t), y(t), f(t), x(t-1), y(t-1), f(t-1), … x(t-n), y(t-n), f(t-n)

- Build a table with the information from three of your best performant models. Fill it with:
  - Number of parameters
  - Number of epochs
  - RMSE for the training dataset
  - RMSE for the testing dataset
- Show some examples of trajectories reconstructed with both methods, for trajectories in the training and testing datasets.
  - How do you obtain better trajectories? forecasting only one step? or using the model's output as input to reconstruct the whole time-series. Why?