ASSIGNMENTS

October 21, 2022

1 PseudoInverse vs Ridge Regression - (4 p)

Discuss the difference between solving ill-posed problems by pseudo-inversion and ridge-regression on the basis of experimental benchmarks.

2 Euclidean Distance in MNIST Chars - (3 p)

Let us consider the MNIST Handwritten character recognition that can be retrieved at

`http://yann.lecun.com/exdb/mnist/`.

The assignment requires to address the following items.

1. CROSS-DISTANCE IN MNIST
   Write a Python program that access at the above data collection and returns:
   - The Euclidean distance between any two chars that can be selected from the database;
   - The *cross-distance* between patterns on the basis of their category. Basically, we need to construct the cross-distance matrix defined ad
   $$d_{i,j} = \frac{1}{|\mathcal{P}_{ij}|} \sum_{(h,\kappa)\in\mathcal{P}_{ij}} (x_{h,i} - x_{\kappa,j})^2,$$
   where $\mathcal{P}_{ij}$ are all the pairs of chars with class i and j, respectively.
2. Let us consider a recognition algorithm based on the Euclidean distance between the incoming pattern and a *centroid* computed, for each class, as
   $$\bar{x}_i = \frac{1}{\ell_i} \sum_{\kappa=1}^{\ell_i} x_{\kappa,i}.$$
   Formally, given any pattern x the algorithm determines its class on the basis of the following criterion:
   $$i^\star = \arg \min_{i=1,10} (x - \bar{x}_i)^2.$$
   Write a Python program for implementing this algorithm. Then perform experiments and make comments on its effectiveness. What about choosing different metrics?
3. Modify the above recognition algorithm by using the cosine similarity instead of the Euclidean distance. Make comments concerning the performance w.r.t. the previous case.
4. Look for graphic tools that can pre-process the handwritten chars with the purpose of modifying the thickness. Make comments on the performance of the written recognition algorithms when changing the thickness.

3 The cosine distance for text classification - (3 p)

Create your own small document collection with the purpose of discovering similarities between documents. In particular, the assignment requires to address the following items.

1. Create your own small document collection.
2. Write a Python program to construct the document internal representation based on the `tfxidf` representation[1] and the document cosine similarity between any pair of the collection.
3. Write a Python program that to retrieve the most similar element of a given document w.r.t. the document of the previously chosen data collection by using the cosine similarity.
4. Perform the same experiment as above by using the Euclidean distance.
5. Make comments on the similarity which arises when using the cosine or the Euclidean distance.

[1] I will support you by providing appropriate Python scripts.

4 Weight prediction - (5 p)

Consider the problem of predicting the weight of a person on the basis of his/her height and on the basis of his/her age as shown during my lectures. Based on what I presented the assignment is

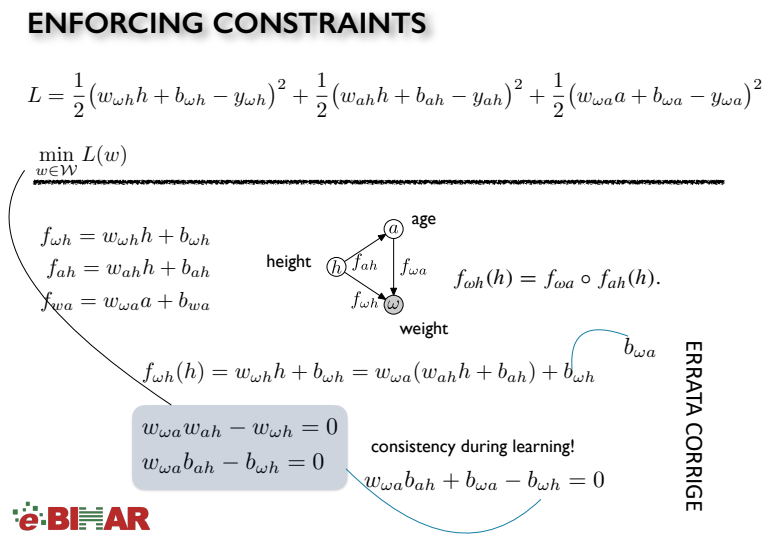

Figure 1: Slides showed during my lectures.

expected to provide an experimental assessment of the idea which needs to undergo the following steps:

1. Create yourself the data collection.
2. Provide regression experiments for the independent predictions.
3. Experiment the constraint-based learning principle.
4. Make final comments on the experiments.

5 IRIS classification (from UCI repository - (3 p))

This is perhaps the best known database to be found in the pattern recognition literature (See Duda & Hart, for example). The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are NOT linearly separable from each other.

Predicted attribute: `class of iris plant.`
Detailed information at

`https://archive.ics.uci.edu/ml/datasets/iris`

6 Separation surfaces in feedforward nets - (4 p)

Given a neural network defined by $f(w, x)$, with given weights w, we call separation surface the set $\mathcal{S} = \left\{x \in \mathbb{R}^d : f(w, x) = 0\right\}$. Write a program for drawing separation surfaces with input in $\mathbb{R}^2$ and discuss the results.

7 Covid-19 predictions

Based on publicly available data for Italy, write a program for prediction on the number of deaths and the number of people in intensive care. The prediction is expected to act on a given time window for the predictions. The model you are expected to use is simply that of a linear machine. However, the prediction must be enriched using polynomial and exponential models. The number of cases is predicted according to the following two assumptions:

- *Polynomial approximation*

$$f(x) = a_{x^n} + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0 \tag{1}$$

  where one estimates the coefficients after having enriched the input by

$$z_1 = x,\ z_2 = x^2,\ \ldots z_n = x^n$$

- *Exponential approximation*

$$f(x) = ae^{\alpha x} \tag{2}$$

You are assumed to learn on a training set and to make predictions on a future window.

8 Backpropagation on the `XOR` network - (4 p)

Write yourself a program for learning the weights of the `XOR` neural network. We assume that you simply define the `XOR` neural network as a wired structure within your code, without needing of a general processing of any neural network. Overall, the assignment is expected to face the following problems:

1. Write down the code. Show the optimization of the risk function by plotting its evolution with the number of epochs.
2. Draw the separation surface that is generated at different steps of learning.
3. Draw the lines associated with the neurons of the hidden layer and the separation line of the output neuron in the hidden space.

9 Code of Backpropagation - 5 p

The script on Backpropagation that I prepared was purposely designed to focus only on algorithmic issues, but it is definitely unsuitable for the adoption in other experiments apart from MNIST character recognition. In addition, it does not provide a software organization, which any program is supposed to possess. Based on that Python script, design a program for learning in multi-layered neural networks according to the following guidelines:

1. Provide a general description of the network architecture and data format so as to handle any learning experiment;
2. Design a module for carrying out the test on a given test set;
3. Design a stopping criterion based on cross-validation;
4. Design a module for plotting the results

Students are encouraged to exploit object oriented programming, but a modular organization with appropriate call to functions is also acceptable.

10 Backpropation experiments on MNIST - 5 p

Based on the Python script on Backpropagation that for experiments MNIST character recognition, extend it properly to address the following requirements:

1. Improve the script by a modular organization and save the weights to a file;
2. Design a module for carrying out the test on the given MNIST csv data on the basis of saved weight files;
3. Discuss appropriate choices of the mini-batches
4. Discuss the initialization of the weights and the network architecture;
5. Discuss the role of the stopping criterion
6. Design a module for plotting the results

11 Backpropation: The case of sigmoidal and relu non-linearities - 5 p

Based on the Python script on Backpropagation that for experiments MNIST character recognition, change it properly with the purpose of experimenting the differences between the *rectifier neuron* with respect to the *sigmoidal neuron*. The experiments are supposed to be carried out for learning simple Boolean functions.

1. Improve the script by a modular organization and prepare an interface for learning Boolean functions.
2. Prepare the training set of the chosen Boolean function.
3. Discuss the role of the depth of the network when choosing the two different non-linearities.

12 Relative entropy versus quadratic loss (4 p)

Discuss the different behavior of the relative entropy in case of premature saturation of the neurons with respect to the quadratic loss by providing both experimental evidence and theoretical description. In particular, one is expected to address the following items:

1. Set up your own experiments with neural nets without hidden layers and with hidden layers. and describe differences.
2. Extend the relative entropy to the case of targets $\{-1, +1\}$.
3. Compute the gradient in cases (quadratic and relative entropy loss) and discuss the differences when neurons are saturated.

13 Handwritten char recognition

Based on the Python scripts and on the MNIST data made available write a program for handwritten char recognition whose objectives are the following:

- Test the performances by using the Quadratic and the Entropy Loss
- Test the performance and discuss the efficiency of the following learning mode protocols:
  - batch mode
  - on-line mode
  - mini-batch mode
- Discuss the role of the weight initialization for both the Quadratic and Entropy loss

14 2-D pattern recognition

Based on the Python scripts discussed for 2-D pattern classification write a program with the objective of addressing the following points:

- Test the performances by using the Quadratic and the Entropy Loss
- Test the performance and discuss the efficiency of the following learning mode protocols:
  - batch mode
  - on-line mode
  - mini-batch mode
- Discuss the role of the weight initialization for both the Quadratic and Entropy loss

15 MNIST with variable thickness - 4 p

Suppose we process the handwritten chars of MNIST with the purpose of increasing the thickness of the characters . Discuss your guess on the performance of a simple classifier based on the Euclidean distance from reference chars for each class obtained simply by averaging the patterns of the same class. Basically, for patterns x_i^c of class c we define the pattern representer

$$\bar{x}_c = \frac{1}{\ell_c} \sum_{\kappa=1}^{\ell_c} x_\kappa^c$$

and the decision is based on

$$\arg\min d(x, \bar{x}_c)$$

The assignment is expected to analyze the problem by an experimental assessment according to the following guidelines:

- Write an algorithm for changing the thickness of a char;
- Perform the experiments when changing the thickness;
- Write a report on the performance as a function of the thickness

16 Check the correctness of Backpropagation code - 4 p

Write a Python program for checking Backpropagation correctness by computing the gradient explicitly. The assignment is expected to

- check the result of Backprop code by the explicit computation
- report the precision depending on different numerical approximations

17 Terminal attractor driven policies - 5 p

Set up an appropriate experimental environment for assessing the effectiveness of the Terminal Attractor Policies (TAP). In particular you are expected to focus on the following items:

- Analysis of different policies for updating the time quantization step.
- Analysis of TAP for batch-mode learning;
- Analysis of TAP for on-line learning