Assignment 1
Neural Networks 501582-3

Classification with Multilayer Perceptron MLP

Assessment information:

Weighting	15%
Deadline	**5th May 2023**
Submission Mode	Electronic via blackboard
Learning outcome assessed	• Demonstrate network abilities to classify patterns. • Evaluate practical considerations in applying neural networks in different applications.
Purpose of assessment	This assignment assesses the understanding of the MLPs and backpropagation algorithm by implementing MLP network for classification.
Marking Criteria	**Individual work**

Objectives

This assignment requires you to implement the multi-layer perceptron algorithm for supervised learning. Python programming language is recommended. Question on Python syntax? Google is your best friend.

Note: NO credit will be given for late submissions. Also No credit for any copy-paste assignments. NO credit will be given for implementing any other type of classification algorithms. **More importantly**, no credit will be given also for using existing libraries instead of implementing it by yourself. However, you are allowed to use numpy, scipy or any other libraries that you might need to implement/evaluate the algorithm or to visualise the results (e.g., matplotlib). You must provide a README file describing how to run you code to re-produce the results.

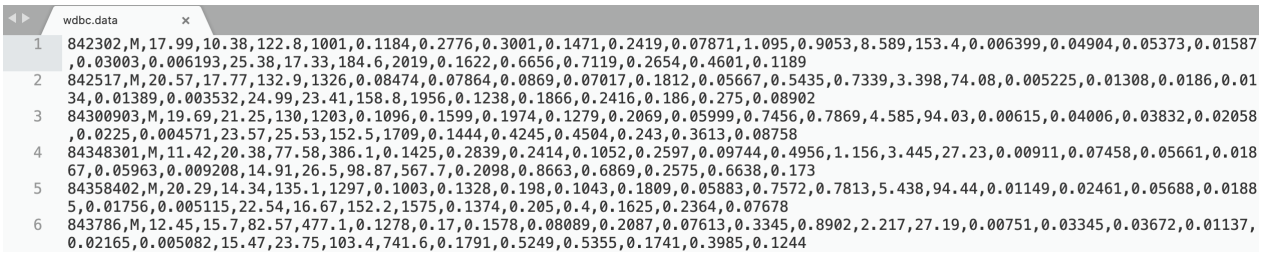
Multi-layer Perceptron for classification

Perceptron is a supervised learning algorithm for classification or regression. In supervised learning, you are given a data set of pairs, where the first element of each pair is a list of features $x \in R^d$, and the second element is a label, $y \in R = \{0,1\}$ in a (binary) classification problem, or $y \in R = R$ in a regression problem. The underlying assumption is that those data represent a function $f : R^d \rightarrow R$, and the goal is to recover f from the data.

A good predictor (classifier or regressor) does not make too many mistakes, in the case of classification; or does not make predictions that are too far off from the true label, in the case of regression. It is necessary to hold out some data, as a test set, to see whether the predictor can generalize well.

Your main task is to implement the multi-layer perceptron algorithm for binary classification. In particular, your task is to train MLP to predict whether breast cancer is benign or malignant based on a set of features. The breast cancer diagnostic dataset contains features calculated from a digital image of a fine needle aspirate (FNA) of a breast mass, and a label representing the diagnosis, i.e., benign or malignant. The features describe the characteristics of the cell nuclei

present in the image. There are 569 data points and 30 features, which consist of the mean, standard error, and "worst" (i.e., the mean of the three largest values) of ten measurements, such as radius, perimeter, area, etc. The dataset is available in the UCI machine learning repository, you can access it here. The file name is **wdbc.data**. Each line represents an instance (i.e., a patient), where the first number is the patient's ID, then the class label (B for benign or M for malignant), and then the list of 30 features. These values are separated by commas (csv file). The first instances are shown in the following screenshot.


wdbc.data

```
1 842302,M,17.99,10.38,122.8,1001,0.1184,0.2776,0.3001,0.1471,0.2419,0.07871,1.095,0.9053,8.589,153.4,0.006399,0.04904,0.05373,0.01587
  ,0.03003,0.006193,25.38,17.33,184.6,2019,0.1622,0.6656,0.7119,0.2654,0.4601,0.1189
2 842517,M,20.57,17.77,132.9,1326,0.08474,0.07864,0.0869,0.07017,0.1812,0.05667,0.5435,0.7339,3.398,74.08,0.005225,0.01308,0.0186,0.01
  34,0.01389,0.003532,24.99,23.41,158.8,1956,0.1238,0.1866,0.2416,0.186,0.275,0.08902
3 84300903,M,19.69,21.25,130,1203,0.1096,0.1599,0.1974,0.1279,0.2069,0.05999,0.7456,0.7869,4.585,94.03,0.00615,0.04006,0.03832,0.02058
  ,0.0225,0.004571,23.57,25.53,152.5,1709,0.1444,0.4245,0.4504,0.243,0.3613,0.08758
4 84348301,M,11.42,20.38,77.58,386.1,0.1425,0.2839,0.2414,0.1052,0.2597,0.09744,0.4956,1.156,3.445,27.23,0.00911,0.07458,0.05661,0.018
  67,0.05963,0.009208,14.91,26.5,98.87,567.7,0.2098,0.8663,0.6869,0.2575,0.6638,0.173
5 84358402,M,20.29,14.34,135.1,1297,0.1003,0.1328,0.198,0.1043,0.1809,0.05883,0.7572,0.7813,5.438,94.44,0.01149,0.02461,0.05688,0.0188
  5,0.01756,0.005115,22.54,16.67,152.2,1575,0.1374,0.205,0.4,0.1625,0.2364,0.07678
6 843786,M,12.45,15.7,82.57,477.1,0.1278,0.17,0.1578,0.08089,0.2087,0.07613,0.3345,0.8902,2.217,27.19,0.00751,0.03345,0.03672,0.01137,
  0.02165,0.005082,15.47,23.75,103.4,741.6,0.1791,0.5249,0.5355,0.1741,0.3985,0.1244
```

Relevant information about the dataset:

- Number of instances: 569
- Number of attributes: 32 (ID, diagnosis, 30 real-valued input features)

Attribute information:

- ID number
- Diagnosis (M = malignant, B = benign)

Ten real-valued features are computed for each cell nucleus:

a) radius (mean of distances from centre to points on the perimeter)
b) texture (standard deviation of Gray-scale values)
c) perimeter
d) area
e) smoothness (local variation in radius lengths)
f) compactness (perimeter^2 / area - 1.0)
g) concavity (severity of concave portions of the contour)
h) concave points (number of concave portions of the contour)
i) symmetry
j) fractal dimension ("coastline approximation" - 1)

Tasks/Questions:

- Write a program to load the instances from the training file **wdbc.data**.
- Split the dataset into a train set (80%) and a test set (20%). Use the train set to train the model. Note: you can use scikit-learn built-in tools for this task and ensure the balance between the class labels in the two sets.
- Implement the multi-layer perceptron neural network for breast cancer classification. Design the layers of the network (input, one hidden and output layer) according to the required task. Include a figure in your report for the MLP structure.
- Considering squared-loss (or cross-entropy) error function with the backpropagation algorithm, train the network to perform the classification task using the labelled training instances. Plot a figure for train/test classification accuracy for the training epochs. According to your plot, what would be the ideal number of iterations to terminate the training? Briefly report your findings.
- Explain the purpose of the learning rate hyperparameter that is used in the update rule of the backpropagation algorithm. Train the MLP with one hidden layer with different learning rates [1.0, 0.5, 0.1, 0.01] and plot the train/test loss for each value. Briefly explain the results.
- Vary the number of nodes *k* in the hidden layer with [5, 10, 15, 20, 25, 30] and evaluate the performance of your MLP classifier. Plot your results in a graph where the x-axis corresponds

to the value of k and the y-axis corresponds to the classification accuracy. What trends can be observed from the graph? Briefly report your findings.

Preamble

- Install *python 3*.
- Install linear algebra libraries numpy and scipy
- For figures and visualisations, you can install *matplotlib* [1].
- You might need to install scikit-learn to perform some evaluations on the NN[2].

Submission Instructions:

Submit:

a) The Python code files.
b) A pdf discussing the assignment's questions.

Good Luck,
Dr Huda Hakami

[1] https://matplotlib.org/stable/users/installing.html
[2] https://scikit-learn.org/stable/install.html

Good luck,
Dr Huda Hakami