Linear Algebra and Optimization for Machine Learning – Project 2

May 27, 2022

Attached to this assignment you can find a 'Heart' dataset containing data of 270 patients. The first 13 columns represents the patients' features such as age etc., and the last column has a value +1 if the patient has a heart disease and −1 if not.

The subsequent features used are:

- age
- sex
- chest pain type (1: typical angina; 2: atypical angina; 3: non-anginal pain; 4: asymptomatic)
- resting blood pressure
- serum cholestoral in mg/dl
- if fasting blood sugar > 120 mg/dl (1 = true; 0 = false)
- resting electrocardiographic results (0: normal; 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV); 2: showing probable or definite left ventricular hypertrophy by Estes' criteria)
- maximum heart rate achieved
- exercise induced angina (1 = yes; 0 = no)
- ST depression induced by exercise relative to rest
- the slope of the peak exercise ST segment (1: upsloping; 2: flat; 3: downsloping)
- number of major vessels (0-3) colored by flourosopy
- blood disorder called thalassemia (3 = normal; 6 = fixed defect; 7 = reversable defect)

The goal of this assignment is to implement a dense feedforward neural network from scratch to perform the task of classification whether, based on the 13 features, the patient has a heart disease.
In specific, work on the following exercises:

1. Perform dimensionality reduction of the data so that you represent each data point using only two features (instead of 13). Plot the resulting data as a scatter plot (like fig. 0.1) to see what is the potential for separating the 'disease' and 'non-disease' classes using simple tools. You are allowed to perform the dimensionality reduction using a ready package like scikit-learn; explain the algorithm used. Judge if the data requires preprocessing (scaling) and, if so, perform it.
2. Implement a dense feedforward neural network with of a variable size of N numbers of hidden layers and $K_1, K_2, \ldots, K_N$ nodes per layer. The network is to perform classification. Therefore, choose suitable activation function(s) and a suitable loss function; examples have been discussed in the lecture.

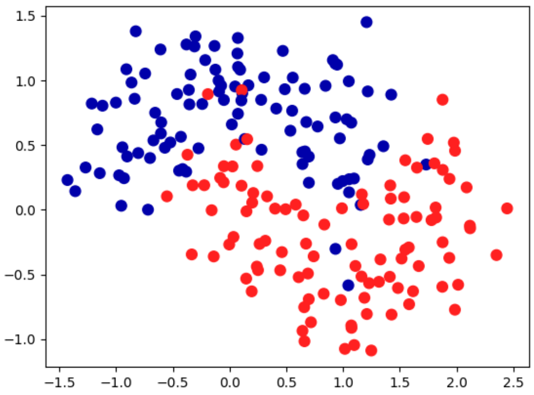

Figure 0.1: Scatter plot of a two-class data set.

3. Implement the Adam gradient method to train the neural network classifier. That means, you need to implement the forward- and back-propagation algorithm in a way that is generic for any N and $K_1, K_2, \dots, K_N$ you choose. Use a suitable stopping criterion.

4. Split the data set into training (75%) and test (25%) data (use stratified sampling to make sure that the 'disease' percentage is similar in both sets).

5. Train your network on the training set for $K_i \in \{2, 5, 10\}$ and $N \in \{2, 5, 10\}$, and for each obtained network, report on the resulting accuracy score (fraction of observations whose labels have been predicted correctly) both on training and test data. Also, for each network plot the partition of the 2D feature space into areas labelled with the two different labels (e.g., by coloring the areas).

6. If your group consists of 4 people:

   - Vary the data split into training and test data and investigate if/how the split influences the accuracy score on training and test data.
   - For each network you train, compute and plot the training/test prediction accuracies of the network after every iteration of your gradient method and plot them against time or number of iterations (choice is up to you).

Explain and motivate all the steps/algorithmic choices made in your solution. Report also on the running times and the number of gradient step iterations (or epochs, if you choose a stochastic gradient variant) you performed while training.

The ideal situation (grade 10) is that you implement all the requested methods (except for dimensionality reduction). If you are not successful, you may use an available package like e.g. JAX to define the network of your choice which will perform the differentiation for you, or use a ready made package that does both the network setup and training for you. However, this will lead to some point deductions. As a point of reference, an assignment which uses a ready package (in which you only specify a number of layers etc etc) will receive a grade 5 if all the results are discussed clearly.

Leave clear comments in your Python code – unclear assignments and their codes will result in a reduction of the grade. Also, explain if something does not work or works too slowly; if you can explain well the problem and how this could be addressed, we might subtract less points, if something does not work as expected.

The page limit for the entire assignment is 6 pages of text excluding images (font size 11). A submitted assignment consists of a PDF file with the assignment and a zipped folder including .py files + a short readme.txt file explaining how to use the code.

The deadline is June 24th, 23:59 CET, please send the assignments to: e.a.t.julien@tudelft.nl.

As mentioned in the lectures, we expect you to work in groups of 3–4; for groups of 4, please be sure to also work on exercise 6.