Homework 1

Balint Negyesi

17-02-2022

Due date: 3rd of March, 2022, before the start of the lecture
Submission via email: b.negyesi@tudelft.nl.
Submissions should consist of an offline, compilable copy of the jupyter notebook and potentially a *short* `.pdf` report of your findings.

In this homework we are concerned with the classification task of handwritten digits. We aim to fit a statistical model which, given an input image of a handwritten digit, recognizes the digit and assigns the image to the proper class. For data we use the well-known MNIST database. The data can be downloaded by executing the following code snippet in a Google Colab notebook.

```
from torchvision.datasets import MNIST
dataset = MNIST('', train=False, download=True)

X = dataset.data.detach().numpy()
y = dataset.targets.detach().numpy()
```

This data is already preprocessed into the regression matrix X and the dependent variable y. In this example the regression matrix is an $\mathbb{R}^{M\times 28\times 28}$ *tensor* consisting of $M = 10000$ samples of 28×28 pixel images. Each sample corresponds to a different measurement. In every sample's image, each pixel's *value* corresponds to a *regressor* – these are the explanatory variables of the classification model. The true labels are found in the vector $y \in \mathbb{R}^{10000}$. In order to display a single image of the dataset you can deploy the `matplotlib` library. For instance, the following code snippet displays the first image in the dataset.

```
import matplotlib.pyplot as plt

idx = 0
img = X[idx, :, :]

plt.imshow(img)
plt.title('true_label:' + str(y[idx]))
plt.show()
```

Using this dataset, and supplementary material provided on the Iris classification problem, solve the following exercises:

Ex. 1 Formulate a binary Single Layer Perceptron model which classifies images based on whether it is an `8` or not. Report on the classification accuracy. In order to avoid infinite loops, replace your convergence condition with a `max_iteration` $\leq 10^4$ criterion in your iteration scheme.
Extend your model in such a way that it distinguishes between *all* digits, i.e. given input image it assigns it a label in $\{0, 1, \ldots, 9\}$. *Hint: recall how we extended binary logistic regression to the multinomial case during the lecture.*

Ex. 2 Formulate a multinomial logistic regression model which distinguishes between digits. Fit it accordingly on the data using Newton's method. (You can use the corresponding module of scikit but be careful not to apply regularization.)

Ex. 3 Use the pre-specified example's neural network class, and train both a shallow and deep neural network of given widths and depths. (Consider the provided `.train()` method as a *black-box* optimizer, and leave its specification as is.) Try different *activation* functions and justify your choices.

Ex. 4 Compare the classification accuracy across the latter 3 methods. Gather confusion matrices. Highlight a few wrongly classified images and see if you can find a pattern.