CA2 - Training Perceptron and Adaline classifiers on Pima indians data

Lever oppgave

Leveres innen 2. mars innen 12:00 **Poeng** 0 **Må leveres** en filopplasting
Filtyper pdf og py **Tilgjengelig** etter 16. feb. i 0:00

Background

You will train a number of simple classification models based on the **Pima Indians Diabetes data**.

You are supposed to use the *perceptron* and *adaline* algorithms (i.e. the **Perceptron** class and **AdalineGD** class) introduced in Chapter 2 of the course textbook.

(Note: you are **not** supposed to use any of the *scikit learn* ML-tools to solve the problem!). If you use scikit-learn for any part of the compulsory assignment it will be rejected.

Data

The data are stored in our GitLab repository:

https://gitlab.com/nmbu.no/emner/DAT200/v2020/-/tree/master/compulsory%20assignments/CA2 (https://gitlab.com/nmbu.no/emner/DAT200/v2020/-/tree/master/compulsory%20assignments/CA2)

The dataset describes the medical records for Pima Indians and whether or not each patient will have an onset of diabetes within five years. The Pima Indians Diabetes data has a total of 768 instances (rows) and 9 features (columns).

Feature description:

column 0 = Number of times pregnant - *continuous*

column 1 = Plasma glucose concentration a 2 hours in an oral glucose tolerance test - *continuous*

column 2 = Diastolic blood pressure (mm Hg) - *continuous*

column 3 = Triceps skin fold thickness (mm) - *continuous*

column 4 = 2-Hour serum insulin (mu U/ml) - *continuous*

column 5= Body mass index (weight in kg/(height in m)^2) - *continuous*

column 6 = Diabetes pedigree function - *continuous*

column 7 = Age (years) - *continuous*

column 8 = Class variable (1:tested positive for diabetes, 0: tested negative for diabetes) - *categorical*

Preprocessing

After removing suspect patients (see Task 2 below), split the dataset into a **training set** (first 400 rows) and **test set** (the remaining samples). The test set will serve as “unseen” data for which we will **predict the classes and compute the test classification accuracy**. Don't forget to scale/standardise the features in data.

Important: For both the Perceptron and the AdalineGD algorithms it is required that you set the **learning rate** to **0.0001** for **all models**.

Modelling and Prediction

For each algorithm, (Perceptron and Adaline), you should compute a total of 400 models based on different pairwise combinations of:

- **Subsets of the data** (8 in total)
- **Number of epochs for model training** (50 in total)

This is how we end up with (8 x 50) = 400 models for one algorithm. Since we will study **two** different algorithms, a total of 800 models will be trained.

The subsets of the data (8 in total) are specified as follows:

- first 50 rows of training data (after removing suspect patients as decribed in Task 2.)
- first 100 rows of training data (after removing suspect patients as decribed in Task 2.)
- first 150 rows of training data (after removing suspect patients as decribed in Task 2.)
- first 200 rows of training data (after removing suspect patients as decribed in Task 2.)
- first 250 rows of training data (after removing suspect patients as decribed in Task 2.)
- first 300 rows of training data (after removing suspect patients as decribed in Task 2.)
- first 350 rows of training data (after removing suspect patients as decribed in Task 2.)
- all 400 rows of training data (after removing suspect patients as decribed in Task 2.)

The number of epochs (50 in total) are specified as follows:

- 1 epoch
- 2 epochs
- 3 epochs

- ...
- 49 epochs
- 50 epochs

The resulting 400 models obtained from the different combinations of subsets and number of epochs are therefore:

(Model 1) Train a model using **first 50 rows** of data with **1 epoch**

(Model 2) Train a model using **first 50 rows** of data with **2 epochs**

(Model 3) Train a model using **first 50 rows** of data with **3 epochs**

...

(Model 50) Train a model using **first 50 rows** of data with **50 epochs**

(Model 51) Train a model **using first 100 rows** of data with **1 epoch**

(Model 52) Train a model **using first 100 rows** of data with **2 epochs**

...

(Model 100) Train a model using **first 100 rows** of data with **50 epochs**

...

(Model 399) Train a model using **all 400 rows** of data with **49 epochs**

(Model 400) Train a model using **all 400 rows** of data with **50 epochs**

Tasks

Make sure you: **a)** describe briefly what you intend to do using markdown cells; **b)** comment your code properly but briefly, such that the reader can easily understand what the code is doing.

1. Visualise the raw data with appropriate plots and inspect it for possible outliers or inconsistencies. Comment briefly on what you see and how this will impact the performance of the perceptron and adaline. For this use no more than three sentences.

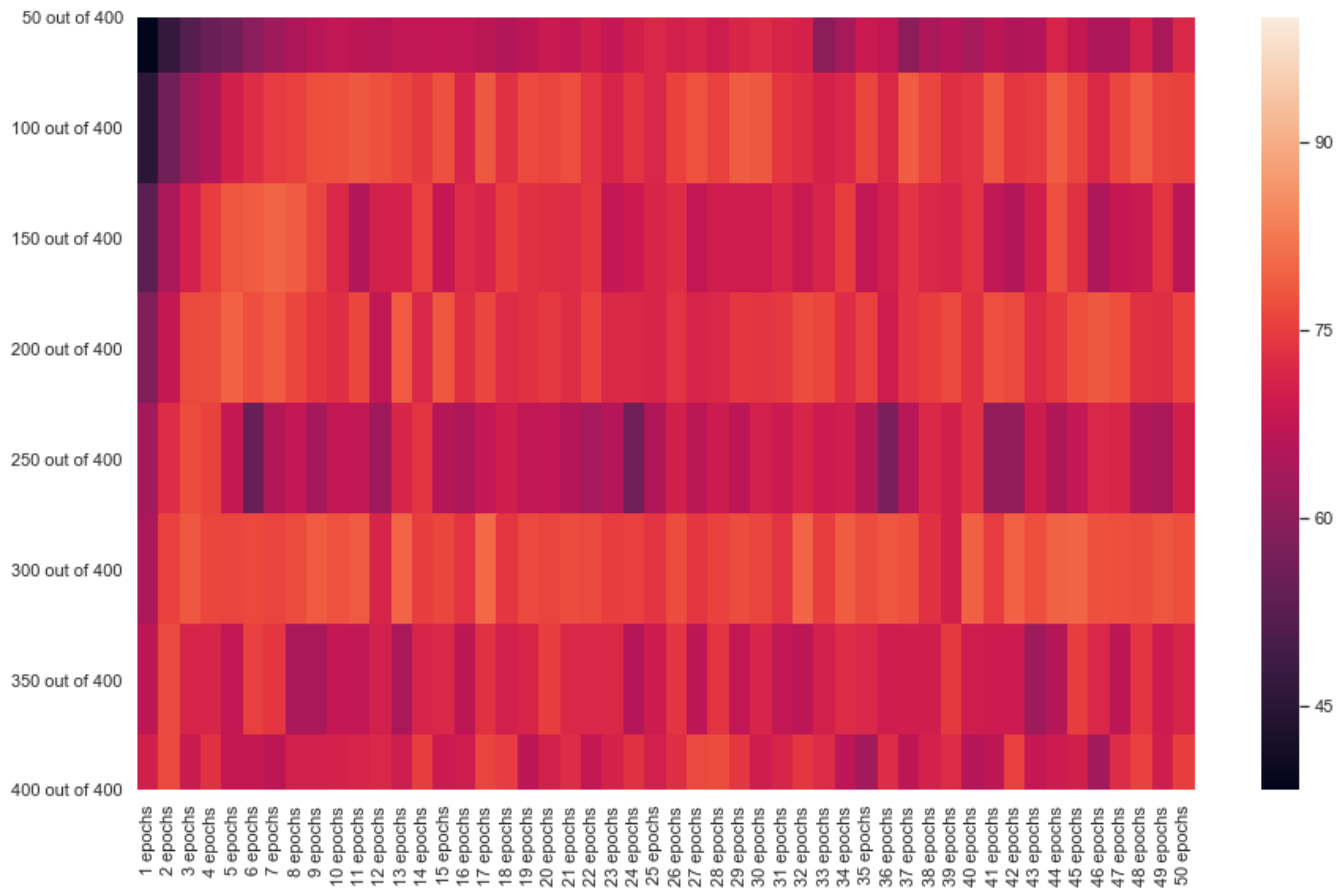
2. Suspect patients: the data says that some patients had a Diastolic blood pressure equal to zero and/or body mass index equal to zero. This is physically not possible. It seems like these parameters were not measured or missing and therefore set to zero. Identify patients with these values and remove them from the data. After removing there should be **729 patients left**.

3. With each of the 400 models, you should **predict the classes of the unseen samples in the**

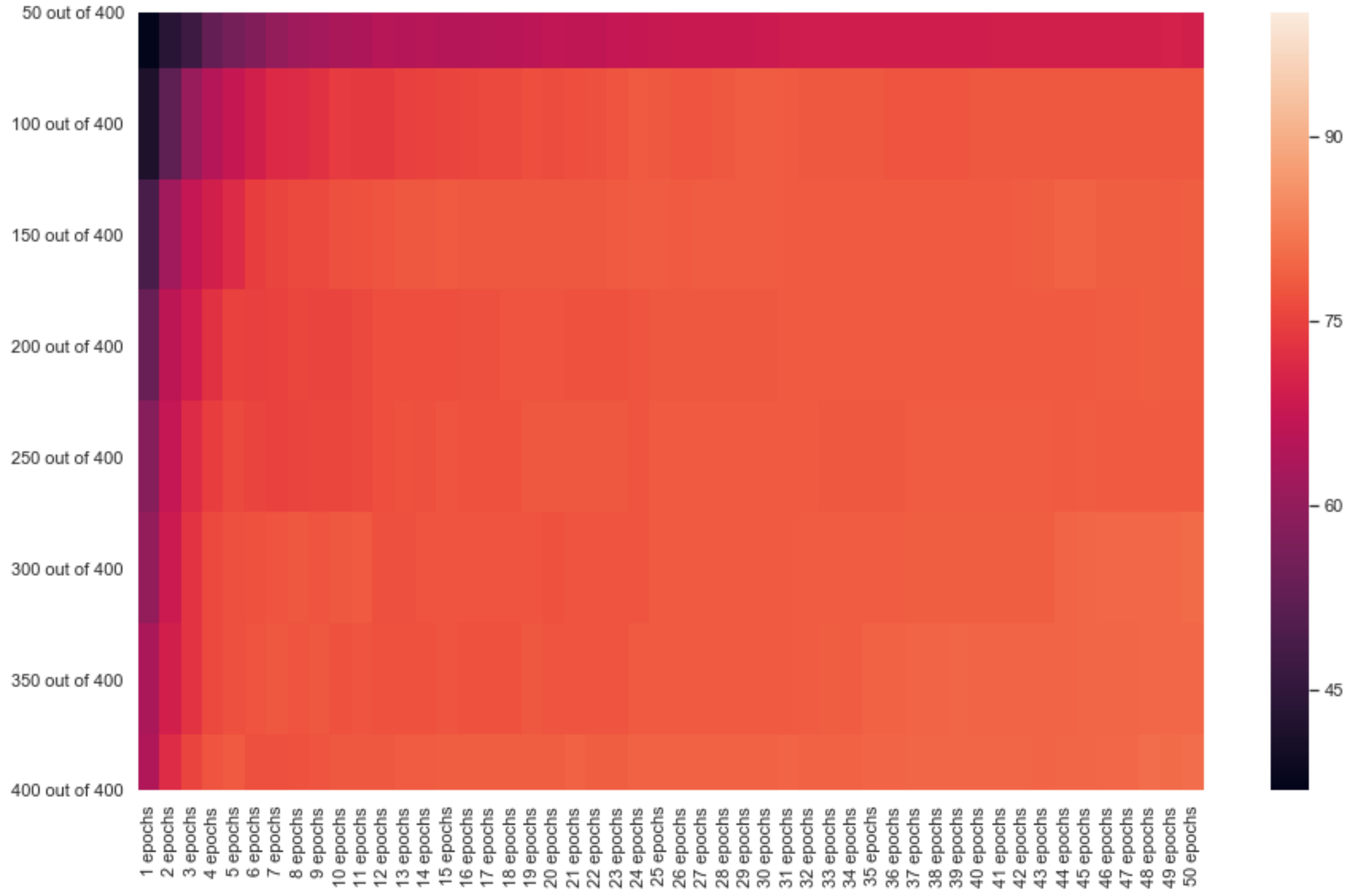
test data and compute the test set classification accuracy. Store the results in a (8 x 50) numpy array or a pandas dataframe.

4. Plot a heatmap of the results (test set classification accuracy) using Python plotting packages matplotlib or **seaborn** (https://seaborn.pydata.org/) . See below what the heatmaps should look like for the two classification algorithms.

Heat map for results computed with perceptron algorithm



Heat map for results computed with adaline algorithm



5. Provide the maximum test set classification accuracy for each, the perceptron classifier and the adaline classifier and **information on with which combination of number training data samples and number of epochs the best classification accuracy was achieved**.

6. The training time of the simpler perceptron algorithm is quite a bit longer than the training time of the adaline algorithm. What might be the reason for this?

Submission to Canvas

1. A PDF of your Jupyter notebook named "CA2 - <your name>.pdf"

2. A .py file with your code named "CA2 - <your name>.py"

Good luck!