CMU 10-715: Homework 1

Perceptron Algorithm on Handwritten Digits

Released: August 31, 2022.

Due: Sept. 12, 2022, 11:59 PM.

Instructions:

- **Collaboration policy:** Collaboration on solving the homework is allowed, after you have thought about the problems on your own. It is also OK to get clarification (but not solutions) from books, again after you have thought about the problems on your own. Please don't search for answers on the web, previous years' homeworks, etc. (please ask the TAs if you are not sure if you can use a particular reference). There are two requirements: first, cite your collaborators fully and completely (e.g., "Alice explained to me what is asked in Question 4.3"). Second, write your solution *independently*: close the book and all of your notes, and send collaborators out of the room, so that the solution comes from you only.

- **Submitting your work:** On Gradescope:

  - Submit your completed python file, perceptron.py, to the assignment titled "HW1 (codes) Perceptron Algorithm on Handwritten Digits".
  - Submit your PDF report file, named [your andrew id].pdf , to the assignment titled "HW1 (report) Perceptron Algorithm on Handwritten Digits". Upon submission, please follow Gradescope instructions to match the question numbers with the page numbers of your report.

  There is no limit on the number of submissions to Gradescope.

- **Gradescope access:** The link to this course is `https://www.gradescope.com/courses/428731/`. If prompted, use the entry code: 6ZJXG3.

  - **Please ensure that your Gradescope account is registered with your andrew.cmu.edu email address** (since otherwise there may be issues compiling your grades).

- **Auto-grader:** Your code will be evaluated with Gradescope auto-grader. We encourage you to start early and use the auto-grader to check that your implementation is correct.

- **Late days:** For each homework you get three late days to be used only when anything urgent comes up. No points will be deducted for using these late days. We will consider an honor system where we will rely on you to use the late days appropriately.

- **Skeleton codes:** The python files, data.py and perceptron.py, can be found at the attachment section of the Diderot post that announces the release of this homework.

1 Perceptron Algorithm

Consider a classification problem where we have features $\mathbf{x}_i \in \mathbb{R}^d$, and labels $y_i \in \{-1, +1\}$ for samples $i \in [n]$. Let $\mathcal{H} = \{\mathbf{x} \mapsto \text{sign}(\mathbf{w}^\top \mathbf{x} + b) : \mathbf{w} \in \mathbb{R}^d, b \in \mathbb{R}\}$ be our hypothesis class with weights $\mathbf{w}$ and bias b. You will implement the perceptron:

Algorithm 1: Perceptron algorithm

Initialize parameters $\mathbf{w}_0 = 0$, $b_0 = 0$, step $t = 0$;
while $\exists i \in [n]$ *such that* $y_i(\mathbf{w}^\top \mathbf{x}_i + b) \leq 0$ **do**
  $\mathbf{w}_{t+1} = \mathbf{w}_t + y_i \mathbf{x}_i$;
  $b_{t+1} = b_t + y_i$;
  $t = t + 1$;
end
Output $\mathbf{w}_t$ and b_t

1.1 MNIST Binary Classification Data [20 points]

The first part of the assignment will be to prepare data from the MNIST database of handwritten digits. The output of this part of the homework will only be the plots of the training data. For this assignment you will use python3 in particular the numpy and matplotlib libraries.

We recommend you to use the functions available at data.py.

(a) Download the MNIST database available at this url: `http://yann.lecun.com/exdb/mnist/`, make sure to download the four files corresponding to train and test features labels. *Hint: You can either do so manually, or use the "maybe_download" method provided in data.py.*

(b) Write a function to filter the datasets to only keep the examples associated with the threes and eights. This function must also keep only the first 500 examples of the train and 500 of the test dataset.

(c) (10 points) After you filtered, plot a grid of 5x5 images of the first 25 images in the training data and include the plot in your report.

(d) Transform the feature data from images of 28x28 to flattened vectors of 784 entries, we will refer to this number as d the dimension of the features.

(e) (10 points) Plot a histogram or bar chart to show the amount of threes and eights in the filtered training data.

(f) Transform the label data from the integer labels 3 and 8 to -1, +1 respectively so that you can later use directly in the perceptron algorithm.

For this part of the homework you will only need to report the image grid and the histogram plots items (c) and (e) in your pdf submission.

1.2 Perceptron Algorithm [80 points]

In this part of the homework you will complete the perceptron class provided to you in perceptron.py by coding the methods: update, train and predict. After this you will record the performance of the perceptron algorithm in the MNIST binary classification data you constructed above **(with the 500 train and 500 test)**.

(a) (15 points) Complete the predict method which computes the perceptron predictions, it receives the features $\mathbf{X} \in \mathbb{R}^{n \times d}$, and outputs its predictions $\hat{y} \in \mathbb{R}^n$. *Hint: y_hat.shape* $== (n,)$.

(b) (15 points) Complete the update method which updates the weights $\mathbf{w}$ and bias b of the classifier, it receives the features of a single example $\mathbf{x} \in \mathbb{R}^d$ and its label $y \in \{-1, +1\}$. *Hint: x.shape* $== (784,)$.

(c) (20 points) Complete the train method which receives the train and test data and executes the perceptron algorithm for 2000 iterations. *Hint: This method will be evaluated using the Autograder on linearly separable data (not CIFAR), check that your perceptron can perfectly classify data with this property.*

(d) (20 points) Use the trajectories attribute from the perceptron class that stores the train and test accuracy of the perceptron algorithm when you execute the train method.

  (a) (10 points) Plot the trajectories. *Hint: trajectories may fluctuate. You are allowed (but not required) to plot them with a rolling average so as to see the trend more easily.*

  (b) (5 points) Report the final train accuracy.

  (c) (5 points) Report the final test accuracy.

(e) (10 points) Describe the plot of the train and test accuracy trajectories. In particular, explain the difference between the two trajectories. Provide your ideas on the origin of the generalization gap.

The perceptron class will be graded through **Gradescope's Autograder**. This Autograder will ask for your perceptron.py file:

- Be sure you only use numpy in this file.
- Only include the Perceptron class.
- Do not change method names of the methods, their inputs and outputs.
- Comply with the assertions provided in the Perceptron class.
- Your data wrangling and plot functions must be in separate files that you are not required to send.

- To avoid compatibility issues, we recommend you use python version 3.6 or higher and numpy version 1.22.0 .

Besides, your response for parts (d) and (e) should be included in your pdf submission.