Machine Learning
COMP 5630/ COMP 6630/ COMP 6630 - D01

Instructor: Dr. Shubhra (“Santu”) Karmaker
TA 1: Dongji Feng
TA 2: Souvika Sarkar
Department of Computer Science and Software Engineering
Auburn University
Fall, 2022

September 8, 2022

Assignment #3
Perceptrons

Submission Instructions

This assignment is due Tuesday, September 15, 2022, at 11:59pm. Please submit your solutions via Canvas (https://auburn.instructure.com/). You should submit your assignment as a typeset PDF. Please do not include scanned or photographed equations as they are difficult for us to grade.

Late Submission Policy

The late submission policy for assignments will be as follows unless otherwise specified:

1. 75% credit within 0-48 hours after the submission deadline.
2. 50% credit within 48-96 hours after the submission deadline.
3. 0% credit after 96 hours after the submission deadline.

Tasks

1 Linear Separators [50 pts]

Recall that a linear separator in 2 dimensions is given a function of the form:

$$y = \phi(w_1 x_1 + w_2 x_2 + b)$$

where $x_i \in \{0, 1\}$, and transfer function:

$$\phi(z) = \begin{cases} 0 & \text{if, } z \leq 0 \\ 1 & \text{otherwise} \end{cases}$$

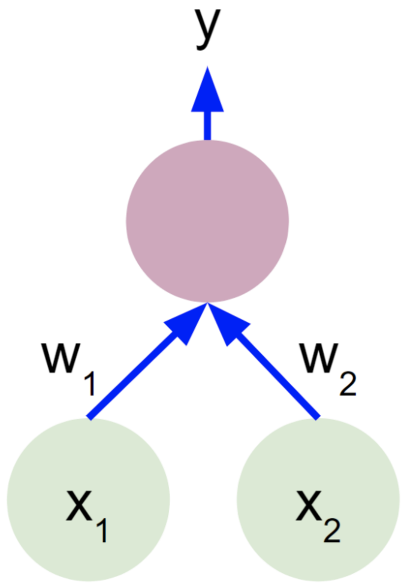

Figure 1: Linear Separator for question 1.

a. [**5 pts**] Complete the following truth table for the AND operation.

x_1	x_2	AND
0	0	
0	1	
1	0	
1	1	

Table 1: Logical AND operation

b. [**5 pts**] Provide values of w_1, w_2 and b to use the linear separator for logical AND.

c. [**5 pts**] Complete the following truth table 2 for the OR operation.

x_1	x_2	OR
0	0	
0	1	
1	0	
1	1	

Table 2: Logical OR operation

d. [**5 pts**] Provide values of w_1, w_2 and b to use the linear separator for logical OR.

e. [**5 pts**] Complete the following truth table 3 for the XOR operation.

f. [**10 pts**] Prove that the linear separator depicted in Figure 1 cannot be used to create logical XOR.

x_1	x_2	XOR
0	0	
0	1	
1	0	
1	1	

Table 3: Logical XOR operation

g. **[15 pts]** Suppose, we want to implement a classifier that takes two input values, where each value is either 0, 1 or 2, and outputs a 1 if at least one of the two inputs has value 2; otherwise it outputs a 0. Can this function be learned by a Perceptron? If so, construct a Perceptron that does it; if not, why not.

2 Programming Assignment: Perceptrons [50 pts]

In this assignment, we are going to see if we can teach a computer to distinguish living things from non-living things. More precisely, you will implement the perceptron algorithm to detect whether or not an image contains an animal or not.

2.1 Problem Statement

You are given a dataset consisting of images. Each image contains a picture of an animal or something else. Your task is to implement two algorithms to classify which images have animals in them. Using the training set, you will train a perceptron classifier that will predict the right class label for an unseen image. Use the development set to test the accuracy of your learned models. We will have a separate (unseen) test set that we will use to run your code after you turn it in. **You may use NumPy in this assignment to program your solution. Aside from that library, no other outside non-standard libraries can be used.**

2.2 The template package

The template package contains everything you need to get started on your assignment. Specifically, it contains

- **reader.py:** This file is responsible for reading in the data set. It returns the training images, labels for these images, development data, and labels for the development data.
- **main.py:** This is the main file that starts the program, and computes the accuracy, precision, recall, and F1-score using your implementation of the classifers.
- **classify.py:** This is the file where you will be doing all of your work.
- **data.zip** The dataset. When you uncompress this file, you'll find a binary object that our reader code will unpack for you.

To understand more about how to run the MP, run python3 main.py -h in your terminal. Add your code to classify.py. **Do not modify the code provided in the other files.**

2.3 The dataset

The dataset consists of 10000 32 * 32 colored images total. We have split this data set for you into 2500 development examples and 7500 training examples. The images have their RGB values scaled to range 0-1. This is a subset of the CIFAR-10 dataset, provided by Alex Krizhevsky.

The reader will supply the set of test images as one giant numpy array. The development data is in the same format. So your code must read the dimensions from the input numpy array.

In the dataset, each image is 32 * 32 and has three (RGB) color channels, yielding 32 * 32 * 3 = 3072 features.

2.4 Perceptron Model

The perceptron model is a linear function that tries to separate data into two or more classes. It does this by learning a set of weight coefficients wi and then adding a bias b. Suppose you have features x_1, ... , x_n then this can be expressed in the following fashion:

$$f_{w,b}(x) = \sum_{i=1}^{n} w_i x_i + b$$

You will use the perceptron learning algorithm to find good weight parameters wi and b such that sign $f_{w,b}(x) > 0$ when there is an animal in the image and sign $f_{w,b}(x) \leq 0$ when there is a no animal in the image.

Your function classifyPerceptron() will take as input the training data, training labels, development data, learning rate, and maximum number of iterations. It should return a list of labels for the development data. Do not hardcode values for tuning parameters inside your classifier functions.

2.5 Training and Development

Please see the textbook and lecture notes for the perceptron algorithm. You will be using a single classical perceptron whose output is either positive or negative (i.e. sign/step activation function).

- **Training:** To train the perceptron you are going to need to implement the perceptron learning algorithm on the training set. Each pixel of the image is a feature in this case. Be sure to initialize weights and biases to zero.

  Note: Use a constant learning rate (no decay) and do not shuffle the training data.

- **Development:** After you have trained your perceptron classifier, you will have your model decide whether or not each image in the development set contains animals. In order to do this take the sign of the function $f_{w,b}(x)$. If it is negative or zero then classify as 0. If it is positive then classify as 1.

Use only the training set to learn the weights.

2.6 Accuracy Reporting

Using the main.py script, generate accuracy, f1, precision and recall scores for your perceptron algorithm on the development set. Submit your code (only classify.py) and report accuracy, f1, precision and recall scores in your PDF submission.

2.7 Using Numpy

It is much easier to write fast code by using numpy operations. Your data is provided as a numpy array. Use numpy operations as much as possible, until right at the end when you choose a label for each image and produce the output list of labels.

NumPy Tips:

- Running computations on arrays tend to be faster when run using NumPy arrays. If you are having issues with timing out, then consider using a NumPy implementation
- Linear algebra operators (dot product, vector norm, matrix multiplication) are immensely simplified when using NumPy. Consider looking up methods to perform some of these operations if they are needed in your solution.
- NumPy Broadcasting may make your life easier for this assignment.

Disclaimers: This assignment re-uses some materials from the publicly available website: CMU Introduction to Machine Learning Course, 10-315, Spring 2019. I personally thank Prof. Maria-Florina Balcan for sharing her teaching materials publicly. This assignment is exculively used for instructional purposes.

We also thank University of Illinois at Urbana-Champaign for developing this assignment. Credit geoes to Justin Lizama (jlizama2@illinois.edu), Daniel Gonzales (dsgonza2@illinois.edu) and Weilin Zhang (weilinz2@illinois.edu).