CMPSC 448: Machine Learning. Homework 4a:
Matrices from Scratch.
Due: February 17, 2020

1. Instructions

- You cannot look at anyone else's code.
- Fill in and upload hw4a.py to gradescope.
- All code (except import statements) in hw4a.py should be inside functions (importing hw4a.py should not cause code to execute).
- Code must have comments and any constants should be stored in a variable defined near the top of your file.

Important! Read this! Your homework **may not** import any python packages. Any attempt to import a package will result in a 0 for the assignment. Do not even use the word "import" or "eval" or "exec" in your file.

2. Some Background

Since your code cannot import any python libraries, you will need a wrapper around it to test your code. For example,

```
import numpy as np
from matplotlib import pyplot as plt
import hw4a

def sigmoid(x):
    result = 1.0/(1 + np.exp(-np.array(x)))
    return result.tolist()

def huber(x):
    myval = np.array(x)
    result =  np.where(np.abs(myval) <= 1, 0.5* myval**2, np.abs(myval)-0.5)
    return result.sum()

z = np.arange(-5, 5, 0.1)
plt.plot(z, [huber(a) for a in z])
plt.show() # program will wait until you close the graph

#create your A and x
hw4a.question5(A, x, sigmoid)
```

2.1. **The sigmoid function.** The sigmoid function, defined above works on numbers, lists, and lists of lists. When x is a single number, this function returns the sigmoid of x. When x is a list or numpy vector, it applies the sigmoid function element-wise (i.e. to each element in the list/vector) and returns a list of the same dimension as x. When x is a list of lists (or numpy matrix), it applies the sigmoid function element-wise and returns a list of lists.

The derivative of the sigmoid is:

$$\frac{d\,\text{sigmoid}(x)}{dx} = \text{sigmoid}(x) * (1 - \text{sigmoid}(x))$$

2.2. **Huber Loss.** The Huber loss, defined above, is an alternative to squared loss and absolute value loss. For numbers close to 0 it acts like the squared loss and for numbers farther away, it acts like absolute value loss. You will need to figure out how to compute its derivative. The formula for huber loss is:

$$\text{huber}(x) = \begin{cases} \frac{1}{2}x^2 & \text{if } |x| \leq 1 \\ |x| - \frac{1}{2} & \text{if } |x| > 1 \end{cases}$$

When applied to a number x, it returns the huber loss of x. When applied to a vector (or list) or a matrix (or list of lists), the result is the sum of the huber losses of the elements.

3. Basic Matrix Operations

Question 1 (5 points). *In hw4a.py, fill in the function* `question_matvec(mat, vec)`. *The variable* ***mat*** *is a matrix represented as a list of lists and* ***vec*** *is a column vector represented as a python list. The output should be the matrix product between* ***mat*** *and* ***vec****.*

Error checking is important. If there is any problem with the dimensions of the vector and matrix that prevent valid multiplication, your code should raise an exception in this way: `raise Exception("Dimension Mismatch")`

Remember, your code may not import anything.

Question 2 (5 points). *In hw4a.py, fill in the function* `question_matmat(mat1, mat2)`. *The variables* ***mat1*** *and* ***mat2*** *are both matrices represented as a list of lists. The output should be the matrix product between* ***mat1*** *and* ***mat2****.*

Error checking is important. If there is any problem with the dimensions of the matrices that prevent valid multiplication, your code should raise an exception in this way: `raise Exception("Dimension Mismatch")`

Remember, your code may not import anything.

4. Matrix Calculus

Question 3 (10 points). *In hw4a.py, fill in the function* `question3(A, W, x, b)` *where A and W are matrices (list of lists), and x and b are vectors. Your function should return $\frac{\partial f}{\partial x}$, where $f(x) = (Ax - b)^T W(Ax - b)$.*

Question 4 (10 points). *In hw4a.py, fill in the function* `question4(A, C, W, x, b, d)` *where A, C, W are matrices (list of lists), and x, b, d are vectors. Your function should return $\frac{\partial f}{\partial x}$, where $f(x) = (Ax - b)^T W(Cx - d)$*

Question 5 (10 points). *In hw4a.py, fill in the function* `question5(A, x, sigmoid)` *where A is a matrix, and x and b are vectors.* ***sigmoid*** *is the sigmoid function (also known as* sigmoid*). Your function should return $\frac{\partial f}{\partial x}$, where $f(x) = ||A\sigma(x)||^2$ (the value of $||z||^2$ is the sum of the squares of the entries of z). If you have questions about the sigmoid function, read the background section on the first page before you ask.*

Remember, your code may not import anything.

Question 6 (10 points). *In hw4a.py, fill in the function* `question6(A, x, b, sigmoid)` *where A is a matrices, and x and b are vectors. Your function should return $\frac{\partial f}{\partial x}$, where $f(x) = ||\sigma(Ax) - b||^2$. If you have questions about the sigmoid function, read the background section on the first page before you ask.*

Remember, your code may not import anything.

Question 7 (10 points). *In hw4a.py, fill in the function* `question7(A, W, x, b, sigmoid)` *where A and W are matrices, and x and b are vectors. Your function should return $\frac{\partial f}{\partial x}$, where $f(x) = ||W\sigma(Ax) - b||^2$. If you have questions about the sigmoid function, read the background section on the first page before you ask.*

Remember, your code may not import anything.

Question 8 (10 points)**.** *In hw4a.py, fill in the function* `question6(A, W, x, b, sigmoid, huber)` *where* A *and* W *are matrices, and* x *and* b *are vectors. Your function should return* $\frac{\partial f}{\partial x}$*, where* $f(x) = huber(\sigma(Ax) - b)$*. If you have questions about the sigmoid function, read the background section on the first page before you ask. If you have questions about the* ***huber*** *loss function, read the background section on the first page before you ask.*

Remember, your code may not import anything.