Computer Science 497J/597J (Deep Learning)

Lab 1 (10 points)
Due Wednesday, October 11th at 10:00pm

Read all of the instructions. Late work will not be accepted.

Description

This lab is designed to introduce you to the core concepts of Tensorflow. You should have already watched the Lab 1 video lectures prior to starting this lab. This assignment is to be completed in "lab groups." You should pick groups of 2-4, and plan to stick in these groups for Lab 2 and on. As a lab group, you will be implementing (and submitting code), as well as producing and submitting a short writeup. Work is to be done together as a group, not in parallel with different members working on different parts. If you don't finish during the class period, please arrange a time with your lab group members to finish up together by the deadline. Note: you may opt to complete only Part 1, leaving you eligible for 8 of the 10 possible points.

Filenames and Submitting via Github

Each lab group (not each individual) will submit one set of files. For this first lab, pick one of the group members to host the files in his/her individual repository. I can create a shared Github lab repository for the groups you form that can be used for the remaining weeks. The actual files should be in the following location (relative to the root of the repository), and spelling, capitalization and spacing matter:

- `lab1/lab1.txt` Your writeup as a plaintext file, with answers to the questions later in this document (see the three Writeup sections).
- `lab1/lab1_pt1.py` Your code for part 1 of the assignment.
- `lab1/lab1_pt2.py` Your code for part 2 of the assignment.

Writeup (Lab 1)

Add the following answers (labeled L.1, L.2, etc.) to your writeup, `lab1.txt`:

L.1) List the lab group members who were present for all parts of this lab (including any follow-up after the class period).

L.2) List any lab group members who were not present for at least some parts of this lab (including any follow-up after class), and please clarify the parts they missed.

L.3) How much time, if any, did you need **outside of Monday's class time** to complete this assignment. Put "no additional time" if you completed the lab within the scheduled class period.

L.4) If your group prefers that this assignment **not** be graded, please let me know here.

Part 1: Optimizing Rosenbrock (8 points)

Skills: creating graphs, running graphs, variables, optimizers

Overview

You will use Tensorflow to minimize the non-convex Rosenbrock function, $f : \mathbb{R}^2 \to \mathbb{R}$, where

$$f(w) = (a - w_1)^2 + b(w_2 - w_1^2)^2 \tag{1}$$

Filename and commandline arguments

You should implement a Python program named `lab1_pt1.py`, which takes four arguments:

1. `a`: (float) the constant a in Eqn. 1.
2. `b`: (float) the constant b in Eqn. 1.
3. `lr`: (float) the learning rate (step size)
4. `optimizer`: (string) the optimization algorithm to use, from the set:
   - "gd" (for gradient descent)
   - "gdm" (for gradient descent with momentum [set the momentum to 0.9])
   - "adam" (for Adam)

And example call is

```
python lab1_pt2.py 1.0 100.0 0.1 gdm
```

Behavior and output

- Initialize `tf.float32` variable $w \in \mathbb{R}^2$ to the constant $[0\ 0]^T$.
- Training iterations should continue until one of three criteria is met:
  - the objective value $f(w)$ is less than 0.0001 (i.e. "converged"), or
  - the objective value is `inf` (i.e. diverged; hint: see Numpy's `isinf`), or
  - the objective value is `nan` (i.e. diverged; hint: see Numpy's `isnan`).
- After each iteration (starting with iteration 0), print the iteration number, colon, space and then the objective value as a floating point to 10 decimal places; e.g.:

```
0: 1.0000000000
1: 0.9960040450
2: 0.9920239449
3: 0.9880599976
```

Writeup (Part 1)

Add the following answers (labeled 1.1, 1.2, etc.) to your writeup, `lab1.txt`:

1.1) If you run your code with $a = 1$, $b = 100$, $lr = 0.01$ and optimization set to "gd", what is the iteration number on which you first converge or diverge[1]?

[1] I.e. the first iteration number whose objective value is < 0.0001 or nan or inf.

1.2) If you run your code with $a = 1$, $b = 100$, $lr = 0.001$ and optimizer set to "gd", what is the iteration number on which you first converge or diverge?

1.3) If you run your code with $a = 1$, $b = 100$, $lr = 0.001$ and optimizer set to "gdm", what is the iteration number on which you first converge or diverge?

1.4) If you run your code with $a = 1$, $b = 100$, $lr = 0.001$ and optimizer set to "adam", what is the iteration number on which you first converge or diverge?

1.5) If you run your code with $a = 1$, $b = 100$, $lr = 0.0001$ and optimizer set to "gd", what is the iteration number on which you first converge or diverge?

1.6) If you run your code with $a = 1$, $b = 100$, $lr = 0.0001$ and optimizer set to "gdm", what is the iteration number on which you first converge or diverge?

Part 2: Polynomial Feature Expansion (2 points)

Skills: building graphs, using graphs, vector tensors, placeholders.

Overview

Instead of using a non-linear model (like deep learning models), one can perform non-linear regression or classification by non-linearly transforming your features and then using a linear method on the transformed features. Polynomial feature expansion is one typical example. When a kth order expansion is applied to a scalar input feature x, it yields the vector of powers up to k: $[x \ x^2 \ x^3 \ \cdots \ x^K]^T$. In this part of the assignment you will build a computational graph to perform this expansion, and then run it for each input in an input file of scalar features, outputting the corresponding expanded features. It is overkill to use Tensorflow to do this, but it provides a convenient excuse to develop some of the skills cited above.

Filename and commandline arguments

You should implement a Python program named `lab1_pt2.py`, which takes two arguments:

1. `filename`: (string) the name of an input file of features to transform, where each row contains a single floating point number
2. `K`: (int) the order of the polynomial expansion

Behavior and output

Your program should first construct a Tensorflow graph that maps scalar (`float32`) features to vectors in $\mathbb{R}^K$ containing the polynomial expansion of that feature, then you should read through the input and run the graph for each input feature. Finally, your program should write the expanded features to standard output as floating point values with five decimal places. For example, if the input file contains the sequence

```
1.0
2.0
3.0
```

```
3.14159
-1.5
```

and you run your program with $K = 4$, your program should output only

```
1.00000 1.00000 1.00000 1.00000
2.00000 4.00000 8.00000 16.00000
3.00000 9.00000 27.00000 81.00000
3.14159 9.86959 31.00620 97.40878
-1.50000 2.25000 -3.37500 5.06250
```

Writeup (Part 2)

Add the following answer (labeled 2.1) to your writeup, `lab1.txt`:

2.1) For a test file with the following four lines (and $K = 4$):

```
4.2135
0.9939
-3.522
1.0
-10.0
```

please put here your program's output (without any additional formatting).

Academic Honesty

To remind you: you must not share code with anyone other than your lab partners and your professor: you must not look at any one else's code or show anyone else your code. You cannot take, in part or in whole, any code from any outside source, including the internet, nor can you post your code to it. If you need help from any other groups, all involved parties *must* step away from the computer and *discuss* strategies and approaches - never code specifics. I am available for help during office hours. I am also available via Piazza (do not wait until the last minute to send a message). If you participate in academic dishonesty, you will fail the course.