Labs
Optimization for Machine Learning
Spring 2020

EPFL
School of Computer and Communication Sciences
Martin Jaggi & Nicolas Flammarion
github.com/epfml/OptML_course

Problem Set 3, due March 13, 2020 (Gradient Descent, cont.)

Gradient Descent

Solve exercises 12, 14 and 16 from the lecture notes.

Computing Fixed Points

Gradient descent turns up in a surprising number of situations which apriori have nothing to do with optimization. In this exercise we will see how computing the fixed point of functions can be seen as a form of gradient descent. Suppose that we have a 1-Lipschitz continuous function $g : \mathbb{R} \to \mathbb{R}$ such that we want to solve for

$$g(x) = x\,.$$

A simple strategy for finding such a fixed point is to run the following algorithm: starting from an arbitary x_0, we iteratively set

$$x_{t+1} = g(x_t)\,. \tag{1}$$

Practical exercise. We will try solve for x starting from $x_0 = 1$ in the following two equations:

$$x = \log(1 + x)\,, \text{ and} \tag{2}$$

$$x = \log(2 + x)\,. \tag{3}$$

Follow the Python notebook provided here:

colab.research.google.com/github/epfml/OptML_course/blob/master/labs/ex03/notebook.ipynb

What difference do you observe in the rate of convergence between the two problems? Let's understand why this occurs.

Theoretical questions.

1. We want to re-write the update (1) as a step of gradient descent. To do this, we need to find a function f such that the gradient descent update is identical to (1):

   $$x_{t+1} = x_t - \gamma f'(x_t) = g(x_t)\,.$$

   Derive such a function f.

2. Give sufficient conditions on g to ensure convergence of procedure (1). What γ would you need to pick? *Hint: We know that gradient descent on f with fixed step-size converges if f is convex and smooth. What does this mean in terms of g?*

3. What condition does g need to satisfy to ensure *linear* convergence? Are these satisfied for problems (2) and (3) in the exercise?