CS 7150: Deep Learning — Spring 2021— Paul Hand

HW 2 — revised
Due: Friday March 5, 2021 at 5:00 PM Eastern time via Gradescope

Names: [Put Your Names Here]

You will submit this homework in groups of 2. You may consult any and all resources. Note that some of these questions are somewhat vague by design. Part of your task is to make reasonable decisions in interpreting the questions. Your responses should convey understanding, be written with an appropriate amount of precision, and be succinct. Where possible, you should make precise statements. For questions that require coding, you may either type your results with figures into this tex file, or you may append a pdf of output of a Jupyter notebook that is organized similarly. You may use code available on the internet as a starting point.

Question 1. *Perform experiments that show that BatchNorm can (a) accelerate convergence of gradient based neural network training algorithms, (b) can permit larger learning rates, and (c) can lead to better neural network performance. You may select any context you like for your experiments.*

Clearly explain your experimental setup, results, and interpretation.

Response:

Question 2. *Stochastic Gradient Descent*

In this problem, you will build your own solver of Stochastic Gradient Descent. Do not use built-in solvers from any deep learning packages. In this problem, you will use stochastic gradient descent to solve

$$\min_y \frac{1}{n} \sum_{i=1}^{n} |y - x_i|^2, \tag{1}$$

where x_i is a real number for $i = 1 \dots n$.

(a) Using calculus, derive a closed-form expression for the minimizer y^*.

Response:

(b) Generate points $x_i \sim \text{Uniform}[0, 1]$ for $i = 1 \dots 100$. Use Stochastic Gradient Descent with a constant learning rate to solve (1). Use $G(y) = \frac{d}{dy}|y - x_i|^2$ for a randomly chosen $i \in \{1 \dots n\}$. Create a plot of MSE error (relative to y^*) versus iteration number for two different learning rates. Make sure your plot clearly shows that SGD with the larger learning rate leads to faster initial convergence and a larger terminal error range than SGD with the smaller learning rate.

Response:

Question 3. *Momentum, RMSProp, and Adam — revised*

Define

$$f_1(x,y) = x^2 + 0.1y^2,$$
$$f_2(x,y) = \frac{(x-y)^2}{2} + 0.1\frac{(x+y)^2}{2}.$$

In this problem you will minimize these two functions using four optimizers: GD, GD with momentum 0.9, RMSProp, and Adam with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. For each, use learning rate 10^{-3}. When optimizing f_1, initialize at $(1,1)$. When optimizing f_2, initialize at $(\sqrt{2},0)$.

(a) Before numerically solving these 8 optimization problems, guess the ranking (from fastest to slowest) of the rates of convergence of the 8 solutions. Explain your reasoning. You might guess that some converge at the same rate as others. (This subproblem will be graded only on effort and not on correctness).

Response:

(b) Numerically solve the 8 optimization problems described above. Plot the objective as a function of iteration count. Perform at least 2000 iterations. You are encouraged to use the built-in optimizers for these algorithms in PyTorch.

Response:

(c) For each optimizer, comment on whether convergence rate was the same for f_1 and f_2. Why was it the same or why was it different? Explain. You might find it useful break up the explanation in terms of a first phase and a second phase.

Response: