CS 7150: Deep Learning — Summer-Full 2020 — Paul Hand

HW 3
Due: Wednesday July 15, 2020 at 11:59 PM Eastern time via Gradescope

Name: Saurabh Vaidya
Collaborators: Sumeet Gajjar

You may consult any and all resources. Note that 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 PyTorch, TensorFlow, or any other packages you like. You may use code available on the internet as a starting point.

Question 1. *Catastrophic Forgetting*

(a) For this problem, you may work with either the CIFAR-10 dataset or the MNIST dataset. Split the 10 categories into two sets of 5 categories, A and B. Train a CNN or MLP on task A. Then continue training that net on task B. Plot classification error on the test sets of tasks A and B versus amount of training.

Response:

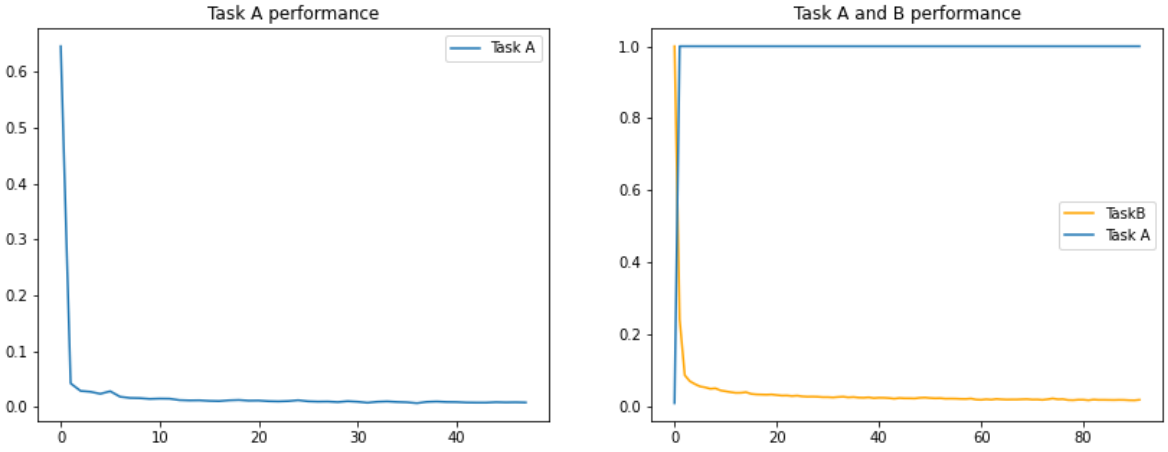

Figure 1: MNIST Task A vs Task A+B Error rates vs amount of training

Answer 1.

(b) Implement Elastic Weight Consolidation for learning task B given your solution to task A. Recall that EWC involves adding a penalty proportional to $\sum_{i=1}^{n} F_i \|\theta_i - \theta^*_{A,i}\|^2$, where $F_i = \mathbb{E}_{(x,y)}\left[\left(\partial_{\theta_i} \log p(y \mid x, \theta)\Big|_{\theta=\theta^*_A}\right)^2\right]$, and $\theta^*_A \in \mathbb{R}^n$ are the parameters learned from task A. Plot classification error on the test sets of tasks A and B versus amount of training.

Response:

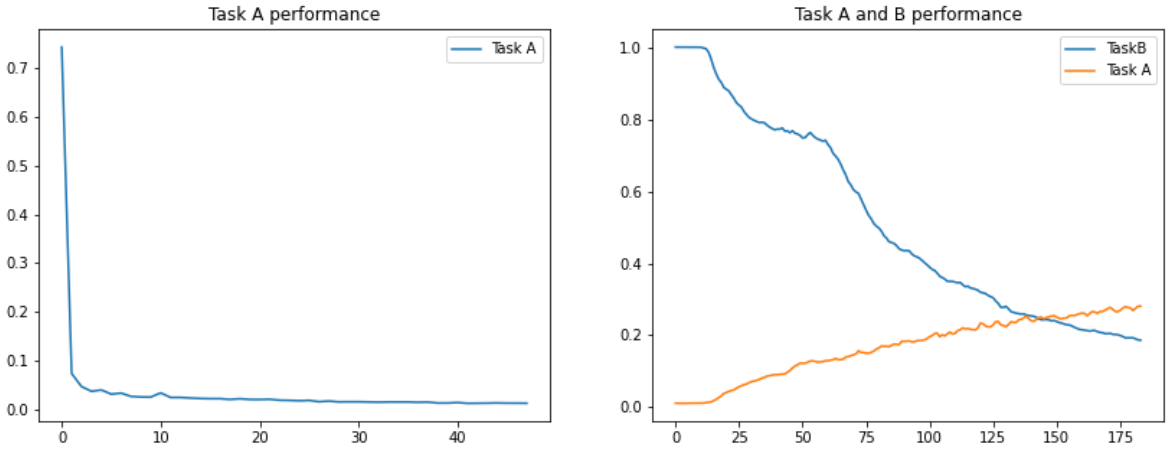

Figure 2: Task A vs Task A+B Error rates vs amount of training after EWC

Answer 2.

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

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

(a) Provide and justify a closed-form expression for the minimizer y^*.

Answer 3.

$$\mathcal{L} = \min_{y} \frac{1}{n} \sum_{i=1}^{n} |y^* - x_i|^2 \tag{2}$$

$$\frac{\partial \mathcal{L}}{\partial y} = 0 \tag{3}$$

$$\frac{1}{n} \sum_{i=1}^{n} 2 * (y^* - x_i) = 0 \tag{4}$$

$$\sum_{i=1}^{n} (y^* - x_i) = 0 \tag{5}$$

$$\sum_{i=1}^{n} y^* - \sum_{i=1}^{n} x_i = 0 \tag{6}$$

$$ny^* = \sum_{i=1}^{n} x_i \tag{7}$$

$$y^* = \frac{1}{n} \sum_{i=1}^{n} x_i \tag{8}$$

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 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:

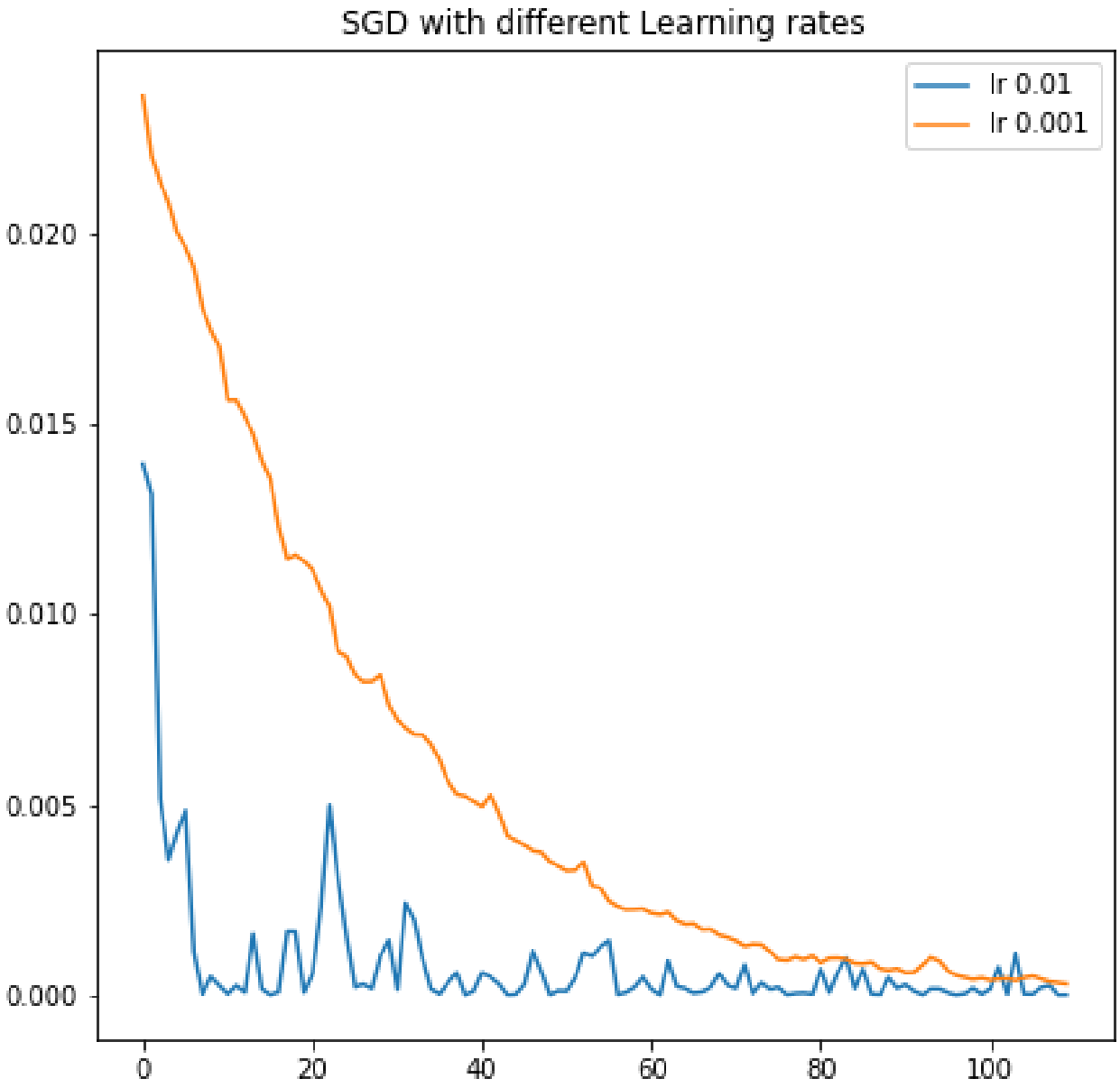

Figure 3: SGD with LR 0.01 and 0.001

Answer 4.