Spring 2022 CS 795 Large Scale Optimization for Machine Learning Assignment 4

Due date: April 5, 2022, 11:59:59 pm

HuayuZhou(hzhou8) and Pankaj Kumar Jatav(pjatav)

Announcement Please use LaTeX to type your solution. Only PDF document generated by LaTeX is accepted. In each team, at the very beginning of the PDF document, please include you and your teammate's name and GMU NetID. The submitted file name should be `netid1_netid2_hw4.pdf`.

1 Experiments: Optimization Algorithms in Neural Networks Training (35 points, including 15 bonus points)

Read the tutorial [1] to learn how to set up the training and test datasets, deep neural networks, optimization and perform training and testing. In this problem, you will be asked to train a shallow neural network on MNIST dataset [2] using different optimization methods.

- (10 points) Load MNIST training and test dataset into your workspace, build a 3-layer neural network model (e.g., you can configure the architecture by yourself, such as number of neurons, activation functions, convolutional layers, etc.).

  We have used the 3-Layer convolutional neural network for this task. Below is the architecture of our CNN model. We used different architecture of different no of activation units and learning rate of optimizer, but we found out that learning rate 0.01 and CNN below give best accuracy with low computation as the model is simple.

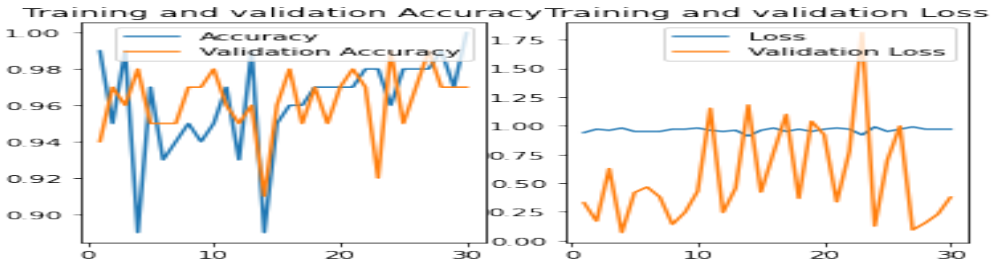
  Here's the architecture info:

  (1)the first layer is a conv2D, specifically, (1, 16, kernel$_s ize = (5, 5), stride = (1, 1), padding = (2, 2)) ReLU()), MaxPool2d(kernel_size = 2, stride = 2, padding = 0, dilation = 1, ceil_mode = False))$.**(2)the second layer**$aconv2D, specifically, (16, 32, kernel_size = (5, 5), stride = (1, 1), padding = (2, 2)), ReLU()); MaxPool2d(kernel_size = 2, stride = 2, padding = 0, dilation = 1, ceil_mode = False))$.**(3)the third layer is the output**.

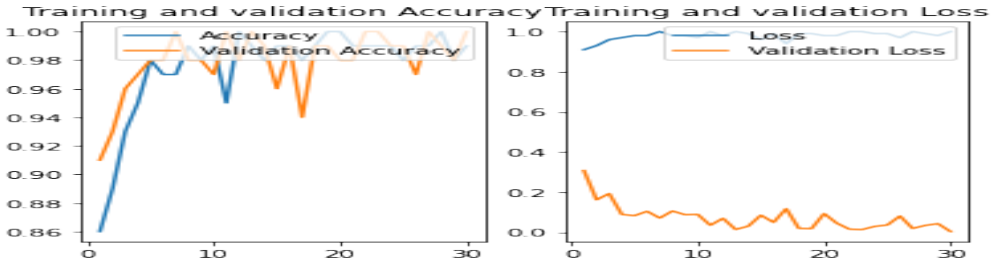
[1] `https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html`
[2] `http://yann.lecun.com/exdb/mnist/`

- (10 points) Implement Adagrad and SGD algorithms. You can tune the hyperparameters (initilization, batch size, learning rate, etc.) and use automatic differentiation. However, you are not allowed to use the built-in implementations of optimizers in pytorch. Plot the training and testing curve versus epochs.

  We implemented the Adagrad and SGD optimizer classes and used both of them for mnist dataset. We got higest 0.99 accuracy on test set using Adagrad and 1.0 using the SGD. Below are the plots for the same.



(a) ADAGRAD Training and Testing curve versus epochs



(b) SGD Training and Testing curve versus epochs

Figure 1: Training and Testing curve versus epochs for loss and accuracy

Please find the code for the same.

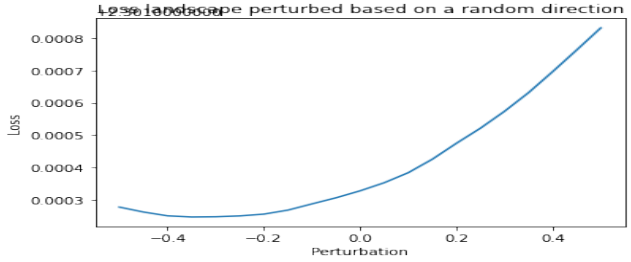
- (bonus points: 15 points) Write a program to check whether the final model your found is a local minima/local maxima/saddle points. Explain your observation.

Proof. Here's our understanding of hessian matrix:
(1) $\nabla^2 f(x) > 0$ implies that x is a local min.
(2) $\nabla^2 f(x) < 0$ implies that x is a local max.
(3) $\nabla^2 f(x)$ has both positive and negative
eigenvalues means that x is a saddle point.

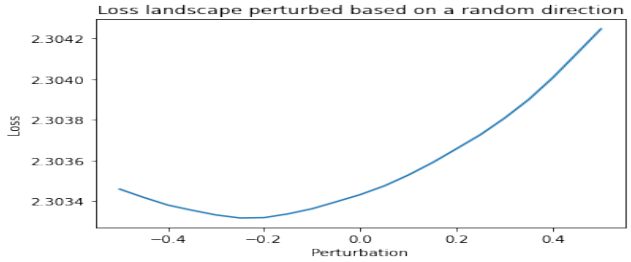
We compute the hession for both models(with adagrad and sgd). We used pytoarch library to compute the hessian named PyHessian.

We notice that for both models hessian value was positive. So both best model were at local minima.

Below are the plot for loss Perturbation using random direction



(a) ADAGRAD Loss Perturbation



(b) SGD Loss Perturbation

Figure 2: Loss Perturbation

□

Please find the code for the same.

For this particular problem, please provide a PDF version of your code.

2 Mirror Descent for Smooth Convex Functions (40 points)

In this question, we will derive the iteration complexity of mirror descent for smooth convex function. For simplicity, we consider the unconstrained problem. Formally, we prove

Theorem 1. *Let Φ be a mirror map that is ρ-strongly convex on $\mathbb{R}^d$ with respect to $\|\cdot\|$, function f be convex and L-smooth with respect to $\|\cdot\|$. Then mirror descent with learning rate $\eta = \rho/L$ satisfies*

$$f(x_t) - f(x_*) \leq \frac{LD_\Phi(x_*, x_1)}{\rho(t-1)}. \tag{1}$$

Note that $\|\cdot\|$ can be any norm, which is not necessarily 2-norm. By L-smooth with respect to $\|\cdot\|$, we mean $\|\nabla f(x) - \nabla f(y)\|_* \leq L\|x-y\|$ for any x, y, where $\|\cdot\|$ denotes the dual norm of $\|\cdot\|$ [3]. This is also equivalent to for any x, y, $f(y) \leq f(x) + \langle \nabla f(x), y - x\rangle + \frac{L}{2}\|y-x\|^2$.

- (10 points) Prove that the descent lemma holds:

$$f(x_{s+1}) - f(x_s) \leq -\frac{1}{\eta} D_\Phi(x_s, x_{s+1}). \tag{2}$$

Proof. Given that f is a convex and L-smooth function, mirror descent with η, then

$$f(x_{s+1}) - f(x_s) \leq g_s^T(x_{s+1} - x_s) - \frac{L}{2}||x_{s+1} - x_s||^2 \tag{3}$$

by mirror update, we have

$$-\eta g_s^T = \nabla\phi(y_{s+1}) - \nabla\phi(x_s) \tag{4}$$

then

$$g_s^T = -\frac{1}{\eta}(\nabla\phi(y_{s+1}) - \nabla\phi(x_s)) \tag{5}$$

by mirror update, we also know

$$x_{s+1} = \Pi_x^\Phi(y_{s+1}) \tag{6}$$

and

$$\Pi_z^\Phi(y) = arg\min_x D_\Phi(x, y) \tag{7}$$

Therefore,

$$g_s^T(x_{s+1} - x_s) = -\frac{1}{\eta}(\nabla\Phi(y_{s+1}) - \nabla\Phi(x_s))(x_{s+1} - x_s) = -\frac{1}{\eta}[\Phi(x_s) - \Phi(x_{s+1}) - < \nabla\Phi(x_s), (x_s - x_{s+1}) >] \tag{8}$$

Because map Φ is ρ-strongly convex, then

$$D_\Phi(x_s, x_{s+1}) = \Phi(x_s) - \Phi(x_{s+1}) - < \nabla\Phi(x_s), (x_s - x_{s+1}) > \tag{9}$$

]

(8) and (9) imply that,

$$g_s^T(x_{s+1} - x_s) = -\frac{1}{\eta} D_\Phi(x_s, x_{s+1}) \tag{10}$$

plug (10) in (3), we have

$$f(x_{s+1}) - f(x_s) \leq -\frac{1}{\eta} D_\Phi(x_s, x_{s+1}) \tag{11}$$

□

[3] https://en.wikipedia.org/wiki/Dual_norm

- (10 points) Prove the following inequality:

$$f(x_s) - f(x_*) \leq \frac{1}{\eta} \left[D_\Phi(x_s, x_{s+1}) + D_\Phi(x_*, x_s) - D_\Phi(x_*, x_{s+1}) \right]. \tag{12}$$

Proof. Since function f is convex, by the definition of mirror descent and the conclusion from problem1, we have the following,

$$f(x_s) - f(x) \leq g_s^T (x_s - x) \tag{13}$$

and

$$g_s^T (x_s - x) = \frac{1}{\eta} (\nabla\Phi(x_s) - \nabla\Phi(y_{s+1}))^T (x_s - x) = \frac{1}{\eta} (D_\Phi(x, x_s) + D_\Phi(x_s, y_{s+1}) - D_\Phi(x, y_{s+1})) \tag{14}$$

And RHS of (14),

$$\leq \frac{1}{\eta} (D_\Phi(x, x_s) + D_\Phi(x_s, y_{s+1}) - D_\Phi(x, x_{s+1}) - D_\Phi(x_{s+1}, y_{s+1})) \tag{15}$$

When limit $x -> x_*$ and $x_{s+1} = y_{s+1}$, (14) and (15) imply that

$$g_s^T (x_s - x_*) \leq \frac{1}{\eta} (D_\Phi(x_*, x_s) + D_\Phi(x_s, x_{s+1}) - D_\Phi(x_*, x_{s+1})) \tag{16}$$

Finally, with limit $x -> x_*$, (13) and (16) prove that

$$f(x_s) - f(x_*) \leq \frac{1}{\eta} [D_\Phi(x_s, x_{s+1}) + D_\Phi(x_*, x_s) - D_\Phi(x_*, x_{s+1}] \tag{17}$$

□

- (10 points) Use the above results to show

$$\frac{1}{t-1} \sum_{2}^{t} [f(x_s) - f(x_*)] \leq \frac{L D_\Phi(x_*, x_1)}{\rho(t-1)}. \tag{18}$$

Proof. by the conclusion from problem2, we do telescoping from s=2 to t, then

$$\sum_{s=2}^{t} (f(x_s) - f(x_*)) \leq \sum_{s=2}^{t} \frac{1}{\eta} (D_\Phi(x_s, x_{s+1}) + D_\Phi(x_*, x_s) - D_\Phi(x_*, x_{s+1})) \tag{19}$$

For the RHS of the above equation, after the simplification, it becomes the following,

$$\sum_{s=2}^{t} D_\Phi(x_s, x_{s+1}) - D_\Phi(x_*, x_{t+1}) \tag{20}$$

based on the conclusion from problem1, we know that

$$\sum_{s=2}^{t} D_\Phi(x_s, x_{s+1}) \leq \sum_{s=2}^{t} \eta(f(x_s) - f(x_{s+1})) = \eta[f(x_2) - f(x_{t+1})] \tag{21}$$

Equations (19), (20) and (21) imply the following,

$$\sum_{s=2}^{t} (f(x_s) - f(x_*)) \leq \frac{1}{\eta} D_\Phi(x_*, x_1) \tag{22}$$

both sides of (22) divided by (t-1), and with $\eta = \frac{\rho}{L}$, we have the conclusion as below,

$$\frac{1}{t-1} \sum_{2}^{t} [f(x_s) - f(x_*)] \leq \frac{L D_\Phi(x_*, x_1)}{\rho(t-1)}. \tag{23}$$

□

- (10 points) Prove the last iterate guarantee as in Theorem 1.

3 Power Method with Noise (40 points)

Consider the algorithm of power method corrupted by noise which is presented in Algorithm 1, where

Algorithm 1 Power Method with Noise

1: **for** $t = 1, \ldots, T$ **do**
2: $\quad \tilde{x}_{t+1} = Ax_t + \zeta_t$
3: $\quad x_{t+1} = \tilde{x}_{t+1} / \|\tilde{x}_{t+1}\|$
4: **end for**
5: **return** x_T

the ζ_t is a random noise (the noise may not be independent from A and x_t). We aim to prove the following theorem.

Theorem 2. *For any positive semidefinite matrix A with top two eigen-pairs (λ_1, v_1), (λ_2, v_2), where $\lambda_1 > \lambda_2$, denote $\theta_t := \arccos(|\langle v_1, x_t \rangle|)$. Suppose that for any t, the noise ζ_t satisfies $\|\zeta_t\| \leq \epsilon(\lambda_1 - \lambda_2)/2$ for some small $\epsilon \leq 1/2$. Assume that $\cos(\theta_1) \geq \epsilon$. Then we have $\tan(\theta_T) \leq 3\epsilon$ for any $T \geq \Omega\left(\frac{\lambda_1}{\lambda_1 - \lambda_2} \log \frac{1}{\epsilon}\right)$.*

- (15 points) Prove that, when $\|\zeta_t\| \leq \min(\epsilon, \cos(\theta_t)) \cdot \frac{\lambda_1 - \lambda_2}{2}$, then

$$\tan(\theta_{t+1}) \leq \left(1 - \frac{\lambda_1 - \lambda_2}{2(\lambda_1 + \lambda_2)}\right) \tan(\theta_t) + \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2} \epsilon. \tag{24}$$

Proof. Given that $\theta_t := \arccos(|\langle v_1, x_t\rangle|)$, we imply that

$$\cos(\theta_t) = |\langle v_1, x_t\rangle| \tag{25}$$

Assume that $u_1 = v_1^{\perp}$, then

$$\tan(\theta_t) = \frac{|\langle u_1, x_t\rangle|}{|\langle v_1, x_t\rangle|} \tag{26}$$

We're going to show that $\tan(\theta_t)$ decreases multiplicatively as iteration goes up.

Given that $\|\zeta_t\| \leq \epsilon(\lambda_1 - \lambda_2)/2$, and $\epsilon \leq 1/2$, then

$$2\|\zeta_t\| \leq \epsilon(\lambda_1 - \lambda_2) \tag{27}$$

By (26), we have

$$\tan(\theta_{t+1}) = \frac{|\langle u_1, x_{t+1}\rangle|}{|\langle v_1, x_{t+1}\rangle|} \tag{28}$$

With $\tilde{x}_{t+1} = Ax_t + \zeta_t$ and $x_{t+1} = \tilde{x}_{t+1}/\|\tilde{x}_{t+1}\|$, we have

$$\tan(\theta_{t+1}) = \frac{|\langle u_1, Ax_t + \zeta_t\rangle|}{|\langle v_1, Ax_t + \zeta_t\rangle|} \tag{29}$$

Define P_k as a set of $p \times p$ projection matrices Π from p dimensions to k dimensional subspaces;

Let $V \in R^{d\times k}$ have orthonormal columns, and $X \in R^{d\times p}$ have independent columns, for $p \geq k$, then

$$\cos(\Theta_k(V, X)) = \max_{\Pi\in P_k} \min_{x\in range(X\Pi), ||X||_2=1} ||V^T x|| = \max_{\Pi\in P_k} \min_{||w||_2=1, \Pi w=w} \frac{||V^T X w||}{||Xw||} \tag{30}$$

Set up $U = V^{\perp}$, we have

$$\tan(\Theta_k(V, X)) = \min_{\Pi\in P_k} \max_{x\in range(X\Pi)} \frac{||U^T x||}{||V^T x||} = \min_{\Pi\in P_k} \max_{||w||_2=1, \Pi w=w} \frac{||U^T X w||}{||V^T X w||} \tag{31}$$

Additionally, if we let V contain the first 2 eigenvectors of matrix A, then by (29), we have

$$\tan(\theta_{t+1}) = \min\max \frac{||u_1^T(Ax_t + \zeta_t)w||}{||v_1^T(Ax_t + \zeta_t)w||} \leq max \frac{||u_1^T A x_t w|| + ||u_1^T \zeta_t w||}{||v_1^T A x_t w|| - ||v_1^T \zeta_t w||} \tag{32}$$

finally,

$$\tan(\theta_{t+1}) \leq \max \frac{1}{||v_1^T x_t w||} \frac{\lambda_2 ||u_1^T x_t w|| + ||u_1^T \zeta_t w||}{\lambda_1 - \frac{||v_1^T \zeta_t w||}{||v_1^T x_t w||}} \tag{33}$$

Define $\Delta = \frac{\lambda_1 - \lambda_2}{2}$, by the assumption that $\|\zeta_t\| \leq \min(\epsilon, \cos(\theta_t)) \cdot \frac{\lambda_1 - \lambda_2}{2}$, then we have,

$$\max \frac{||v_1^T \zeta_t w||}{||v_1^T x_t w||} \leq \frac{||v_1^T \zeta_t||}{\cos(\theta_t)} \leq \frac{\lambda_1 - \lambda_2}{2} = \Delta \tag{34}$$

Similarly, using the fact that $\frac{1}{\cos(\theta_t)} \leq 1 + \tan(\theta)$, we can tell that

$$\max \frac{||u_1^T \zeta_t w||}{||v_1^T x_t w||} \leq \frac{||\zeta_t||}{\cos(\theta_t)} \leq \epsilon\Delta(1 + \tan(\theta_t)) \tag{35}$$

Plugging (35) back into (33), and with $\lambda_1 = \lambda_2 + \Delta$, we have

$$\tan(\theta_{t+1}) \leq \max \frac{||u_1^T x_t w||}{||v_1^T x_t w||} \frac{\lambda_2}{\lambda_2 + \Delta} + \frac{\epsilon\Delta(1 + \tan(\theta_t))}{\lambda_2 + \Delta} = \frac{\lambda_2 + \epsilon\Delta}{\lambda_2 + \Delta} \tan(\theta_t) + \frac{\epsilon}{\lambda_2 + \Delta} \tag{36}$$

Plugging in $\Delta = \frac{\lambda_1 - \lambda_2}{2}$ in (36), eventually, we have the following result,

$$\tan(\theta_{t+1}) \leq \left(1 - \frac{\lambda_1 - \lambda_2}{2(\lambda_1 + \lambda_2)}\right) \tan(\theta_t) + \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2}\epsilon. \tag{37}$$

□

- (10 points) Prove that when $\cos(\theta_t) \geq \epsilon$, we have $\cos(\theta_{t+1}) \geq \epsilon$.

  Proof. by conclusion from the first problem, we know that, at each step t of the given Noisy Power Method,

  $$tan(\theta_{t+1}) \leq max(\epsilon, \max(\epsilon, (\frac{\lambda_2}{\lambda_1})^{\frac{1}{2}}) \tan(\theta_t)) \tag{38}$$

  And it implies that,for a small $\epsilon \leq \frac{1}{2}$, and with the assumption that $\cos(\theta_1) \geq \epsilon$ and $\cos(\theta_t) \geq \epsilon$, we obtain that,

  $$\cos(\theta_{t+1}) \geq \min(1 - \frac{\epsilon^2}{2}, \cos(\theta_1)) \geq \cos(\theta_1) \geq \epsilon \tag{39}$$

  therefore, $\cos(\theta_{t+1}) \geq \epsilon$. □

- (15 points) Use the above results to prove Theorem 2.

Proof. Let V represent the top$k$ eigenvectors of the given matrix A, and set up $\gamma = 1 - \frac{\lambda_2}{\lambda_1} = \frac{\lambda_1 - \lambda_2}{\lambda_1}$. Meanwhile, suppose that the initial subspace is X_0, and with noise $||\zeta_t|| \leq \epsilon \frac{\lambda_1 - \lambda_2}{2}$, we have to prove the following: at each stage s, for some small $\epsilon \leq \frac{1}{2}$, there exits an $S \leq \frac{1}{\gamma} \log(\frac{\tan \Theta_k(V, X_0)}{\epsilon})$, s.t., we have $\tan \Theta(V, X_S) \leq 3\epsilon$ for all stage $s \geq S$.

Since at every stage s of the algorithm,

$$\tan \Theta_k(V, X_s) \leq \max(\epsilon, \tan \Theta_k(V, X_0)) \tag{40}$$

This implies that, for $\epsilon \leq \frac{1}{2}$, we have

$$\cos \Theta_k(V, X_s) \geq \min(1 - \frac{\epsilon^2}{2}, \cos \Theta_k(V, X_0)) \geq \frac{7}{8} \cos \Theta_k(V, X_0) \tag{41}$$

So, the conclusion from problem1 applies at each stage s, then:

$$\tan \Theta_k(V, X_{s+1}) = \tan \Theta_k(V, AX_s + \zeta_s) \leq \max(\epsilon, \delta \tan \Theta_k(V, X_s)) \tag{42}$$

here, $\delta = \max(3\epsilon, (\frac{\lambda_2}{\lambda_1})^{\frac{1}{4}})$.

After $S = \log_{\frac{1}{\delta}} \frac{\tan \Theta_k(V, X_0)}{\epsilon}$ stages, the tangent will reach 3ϵ and settle down.

Meanwhile, it's observed that:

$$\log(\frac{1}{\delta}) \geq \min(\log(\frac{1}{\epsilon}), \log(\frac{\lambda_1}{\lambda_2})) \geq \min(1, \log(\frac{1}{1-r})) \geq \min(1, \gamma) = \gamma \tag{43}$$

Therefore, at every stage s, for some small $\epsilon \leq \frac{1}{2}$, then, there exists an $S \leq \frac{1}{\gamma} log(\frac{\tan \Theta_k(V, X_0)}{\epsilon})$,i.e., $S \leq \frac{\lambda_1}{\lambda_1 - \lambda_2} \log(\frac{\tan \Theta_k(V, X_0)}{\epsilon})$ s.t., we have $\tan \Theta(V, X_S) \leq 3\epsilon$ for all stage s.

In fact, based on the conclusion from problm2 that $\cos(\Theta_1) \leq \epsilon, \cos(\Theta_t) \leq \epsilon, \cos(\Theta_{t+1}) \leq \epsilon$, then $\tan \Theta_k(V, X_0) \approx 1$.

Finally, it's proved that $\tan(\Theta_T) \leq 3\epsilon$ for any $T \geq \Omega(\frac{\lambda_1}{\lambda_1 - \lambda_2} \log \frac{1}{\epsilon})$.

□