**Intro to machine learning- solution 3-**

Aviv Avraham

208295691

| | |
|---|---|
| **Introduction to Machine Learning** | **Spring Semester** |

# Homework 3: April 24, 2023

*Due: May 9, 2023*

# Theory Questions

**Remark:** Throughout this exercise, when we write a norm $\|\cdot\|$ we refer to the $\ell_2$-norm.

1. **(15 points) Step-size Perceptron.** Consider the modification of Perceptron algorithm with the following update rule:

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \eta_t y_t \mathbf{x}_t$$

whenever $\hat{y}_t \neq y_t$ ($\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t$ otherwise). Assume that data is separable with margin $\gamma > 0$ and that $\|\mathbf{x}_t\| = 1$ for all $t$. For simplicity assume that the algorithm makes $M$ mistakes at the first $M$ rounds, afterwhich it has no mistakes. For $\eta_t = \frac{1}{\sqrt{t}}$, show that the number of mistakes step-size Perceptron makes is at most $\frac{4}{\gamma^2}\log(\frac{1}{\gamma})$. (Hint: use the fact that if $x \leq a\log(x)$ then $x \leq 2a\log(a)$). It's okay if you obtain a bound with slightly different constants, but the asymptotic dependence on $\gamma$ should be tight.

ננתח את כמות השגיאות בדומה למה שראינו בכיתה.

<u>חלק ראשון</u>- נראה שהמכפלה הפנימית עם $w^*$ עולה.

$$w_{t+1}w^* = (w_t + \eta_t y_t x_t)w^* = w_t w^* + \frac{1}{\sqrt{t}} y_t x_t w^* \underset{margin\ assumption}{\geq} w_t w^* + \frac{\gamma}{\sqrt{t}}$$

מכאן נובע שאחרי M טעויות ראשונות, ($w^* = 0$ בהתחלה):

$$w_M w^* \geq \sum_{t=1}^{M} \frac{\gamma}{\sqrt{t}} \geq M\frac{\gamma}{\sqrt{M}} = \sqrt{M}\,\gamma$$

<u>חלק שני</u>- נחסום את הנורמה של $w_t$ מלמעלה:

$$\|w_{t+1}\|^2 = \|w_t + \eta_t y_t x_t\|^2$$
$$= \|w_t\|^2 + 2\eta_t y_t x_t + \|\eta_t y_t x_t\|^2 \underset{(2\eta_t y_t x_t < 0)}{\leq} \|w_t\|^2 + \left\|\frac{1}{\sqrt{t}} y_t x_t\right\|^2 \underset{(\|w_t\| = 1)}{=} \|w_t\|^2 + \frac{1}{t}$$

מכאן נובע שאחרי M טעויות ראשונות, ($w^* = 0$ בהתחלה):

$$\|w_M\|^2 \leq \sum_{t=1}^{M} \frac{1}{t} = H_M \leq \ln M + 1 \approx \log M$$

לסיכום, אחרי M טעויות (שהנחנו כי מופיעות ראשונות): $\|w_M\|^2 \leq \log M$ ו- $w_M w^* \geq \sqrt{M}\,\gamma$

נפעיל את א"ש קושי שוורץ:

$$\sqrt{M}\,\gamma \leq w_M w^* \leq \|w_M\|\|w^*\| \leq \sqrt{\log M}$$

ומכאן:

$$\sqrt{M} \leq \frac{1}{\gamma}\sqrt{\log M}$$

נעלה בריבוע ונקבל:

$$M \leq \frac{1}{\gamma^2}\log M$$

ובעזרת הרמז המופיע בשאלה:

$$M \leq \frac{2}{\gamma^2}\log\frac{1}{\gamma^2}$$

ומכאן:

$$M \leq \frac{4}{\gamma^2}\log\frac{1}{\gamma}$$

כנדרש.

2. **(15 points) Convex functions.**

(a) Let $f : \mathbb{R}^n \to \mathbb{R}$ a convex function, $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$. Show that, $g(\mathbf{x}) = f(A\mathbf{x} + b)$ is convex.

(b) Consider $m$ convex functions $f_1(\mathbf{x}), \dots, f_m(\mathbf{x})$, where $f_i : \mathbb{R}^d \to \mathbb{R}$. Now define a new function $g(\mathbf{x}) = \max_i f_i(\mathbf{x})$. Prove that $g(\mathbf{x})$ is a convex function. (Note that from (a) and (b) you can conclude that the hinge loss over linear classifiers is convex.)

(c) Let $\ell_{log} : \mathbb{R} \to \mathbb{R}$ be the log loss, defined by

$$\ell_{\log}(z) = \log_2\left(1 + e^{-z}\right)$$

Show that $\ell_{log}$ is convex, and conclude that the function $f : \mathbb{R}^d \to \mathbb{R}$ defined by $f(\mathbf{w}) = \ell_{log}(y\mathbf{w}\cdot\mathbf{x})$ is convex with respect to $\mathbf{w}$.

a

$$g(x) = f(Ax+b), f\ is\ convex$$

יהיו $x, y \in \mathbb{R}^n$ ו- $\alpha \in [0,1]$ כלשהם.

$$g(\alpha x + (1-\alpha)y) = by\ definition\ f(A(\alpha x + (1-\alpha)y) + b) =$$

$$f\big(\alpha(Ax+b) + (1-\alpha)(Ay+b)\big) \le f\ is\ convex\colon\ \alpha\, f(Ax+b) + (1-\alpha)f(Ay+b) =$$

$$by\ definition\colon \alpha\ g(x) + (1-\alpha)g(y)$$

ומכאן שg קמורה.

b

יהיו m פונקציות קמורות. נגדיר: $g(x) = \max_i f_i(x)$ .

יהיו $x, y \in \mathbb{R}^n$ ו- $\alpha \in [0,1]$ כלשהם.

$$\begin{aligned} g(\alpha x + (1-\alpha)y) & \\ &= by\ definition\colon\ \max_i f_i(\alpha x + (1-\alpha)y) \\ &\le assuming\ i\ is\ \arg max, f_i\ is\ convex\colon\ \alpha\ f_i(x) + (1-\alpha)f_i(y) \\ &\le\ \ \alpha\ \max_i f_i(x) + (1-\alpha)\max_i f_i(y) =\ \alpha\ g(x) + (1-\alpha)g(y) \end{aligned}$$

ומכאן שg קמורה.

c

$$\ell_{log}(z) =\ \log_2(1+e^{-z})$$

נשתמש במשפט מחדוא 1א שאומר שפונקציה קמורה אמ"מ הנגזרת השנייה שלה אי שלילית.

נגזור פעם אחת:

$$(\ell_{log}(z))' = \log_2(1+e^{-z})' = \frac{-e^{-z}}{ln2(1+e^{-z})}$$

נגזור פעם שנייה:

$$\frac{e^{-z}\left(ln2(1+e^{-z})\right)+e^{-z}(-ln2e^{-z})}{ln2(1+e^{-z})^2}=\frac{e^{-z}}{ln2(1+e^{-z})^2}>0$$

המונה חיובי כי$e^x$ פונקציה חיובית.

המכנה חיובי כי $ln2$ > 0 והגורם השני הוא בריבוע.

קיבלנו כי הנגזרת השנייה חיובית ולכן הפונקציה קמורה כנדרש.

חלק שני:

נראה כי

$$f(w)=\ell_{log}(yw\cdot\ x)$$

קמורה.

יהיו $v,w\in\mathbb{R}^d$ ו- $\alpha\in[0,1]$ כלשהם.

$$f(\alpha v+(1-\alpha)w)=\ell_{log}(y(\alpha v+(1-\alpha)w)\cdot\ x)$$
$$=\ dot\ product\ is\ distributive\ and\ scalar\ multiplication$$

$$=\ell_{log}\left((\alpha yv+(1-\alpha)yw)\cdot\ x\right)=\ \ dot\ product\ is\ distributive\ and\ commutative\ =$$

$$\ell_{log}(\alpha yv\cdot\ x+(1-\alpha)yw\ \cdot x\ )$$
$$\leq jensen\ inequality\ as\ we\ proofed\ that\ \ell_{log}\ is\ a\ convex\ function$$

$$\leq\alpha\ \ell_{log}(yv\cdot\ x\ )+(1-\alpha)\ell_{log}(yw\ \cdot x\ )=f(v)+(1-\alpha)f(w)$$

3. **(20 points) Ranking**. In this question, we consider a new learning task in which the objective is to rank items. Assume items are elements of $\mathcal{X} \subseteq \mathbb{R}^d$, and you are given a training set of $n$ lists of $k$ items each, and for each list you receive a "label" vector corresponding to the correct ranking of its items. More formally, you receive a training set

$$S = \left\{ \left( (\mathbf{x}_1^i, \mathbf{x}_2^i, \ldots, \mathbf{x}_k^i), \mathbf{y}^i \right) \right\}_{i=1}^n$$

such that for all $1 \le i \le n$, $\mathbf{y}^i \in \mathbb{R}^k$ assigns a value for each item in $\bar{\mathbf{x}}^i = (\mathbf{x}_1^i, \mathbf{x}_2^i, \ldots, \mathbf{x}_k^i)$, interpreted as a ranking of the items. Your goal is to learn a ranking function $h : \mathcal{X}^k \to \mathbb{R}^k$ which correctly ranks the lists of items from $S$. The *Kendall-Tau* loss between two rankings $\mathbf{y}', \mathbf{y}$ is defined as follows:

$$\Delta(\mathbf{y}', \mathbf{y}) = \frac{2}{k(k-1)} \sum_{j=1}^{k-1} \sum_{r=j+1}^{k} \mathbf{1} \left\{ sgn(y'_j - y'_r) \neq sgn(y_j - y_r) \right\}$$

Note that this function averages the total number of pairs of items which are in different order in $\mathbf{y}'$ compared to $\mathbf{y}$. Assume you are trying to learn a linear ranking function, i.e. a function of the form

$$h_{\mathbf{w}}((\mathbf{x}_1, \ldots, \mathbf{x}_k)) = (\mathbf{w} \cdot \mathbf{x}_1, \ldots, \mathbf{w} \cdot \mathbf{x}_k)$$

for some $\mathbf{w} \in \mathbb{R}^d$, and your goal is to minimize the Kendall-Tau loss over $S$: $\sum_{i=1}^n \Delta(h_{\mathbf{w}}(\bar{\mathbf{x}}^i), \mathbf{y}^i)$. Since this function is hard to optimize, you instead optimize the surrogate "hinge" loss $\sum_{i=1}^n \ell(h_{\mathbf{w}}(\bar{\mathbf{x}}^i), \mathbf{y}^i)$ where:

$$\ell(h_{\mathbf{w}}(\bar{\mathbf{x}}), \mathbf{y}) = \frac{2}{k(k-1)} \sum_{j=1}^{k-1} \sum_{r=j+1}^{k} \max\{0, 1 - sgn(y_j - y_r)\mathbf{w} \cdot (\mathbf{x}_j - \mathbf{x}_r)\}$$

(a) Prove that the hinge loss described above for the ranking objective is convex in $\mathbf{w}$.

(b) Prove that the hinge loss upper-bounds the Kendall-Tau loss, i.e. that $\Delta(h_{\mathbf{w}}(\bar{\mathbf{x}}), \mathbf{y}) \le \ell(h_{\mathbf{w}}(\bar{\mathbf{x}}), \mathbf{y})$ for all $\mathbf{w} \in \mathbb{R}^d, \bar{\mathbf{x}} \in \mathcal{X}^k, \mathbf{y} \in \mathbb{R}^k$.

(c) Prove that if the data is separable with a margin $\gamma > 0$ (i.e. when there exists $\mathbf{w}^\star \in \mathbb{R}^d$ and $\gamma > 0$ such that $sgn(y_j^i - y_r^i)\mathbf{w}^\star \cdot (\mathbf{x}_j^i - \mathbf{x}_r^i) \ge \gamma$ for all $1 \le i \le n$ and all $1 \le j < r \le k$), minimizing the hinge loss will result in a ranking function which minimizes the Kendall-Tau loss.

a

$$\ell(h_w(\bar{x}), y) = \frac{2}{k(k-1)} \sum_{j=1}^{k-1} \sum_{r=j+1}^{k} \max\{0, 1 - sgn(y_j - y_r) w \cdot (x_j - x_r)\}$$

יהיו $\bar{x}$ ו- $y$ קבועים כלשהם. נבחין כי $sgn(y_j - y_r)$ הוא קבוע, עבור $j$ ו$n$ קבועים, ושווה ל0 או1 .
באופן דומה גם $(x_j - x_r)$ קבוע מאותה סיבה. לכן, וכפי שראינו בכתה פונקציה לינארית, $1 -$ $sgn(y_j - y_r) w \cdot (x_j - x_r)$ היא קמורה, בנוסף לפי תרגיל 2 סעיף שני, מקסימום בין פונקציות קמורות היא פונקציה קמורה ולכן $\max\{0, 1 - sgn(y_j - y_r) w \cdot (x_j - x_r)\}$ קמורה. (נבחין כי פונקציה קבועה, פונקציה האפס קמורה גם היא).

ולבסוף ומכיוון שחיבור של פונקציות קמורות הינה פונקציה קמורה נקבל הדרוש.

*b*

יהי $x \in \mathcal{X}^k$, $w \in \mathbb{R}^d$ , ו- $y \in \mathbb{R}^k$ צ"ל כי :

$$\triangle\, (h_w(\bar{x}), y) \le \ell(h_w(\bar{x}), y)$$

$$\frac{2}{k(k-1)}\sum_{j=1}^{k-1}\sum_{r=j+1}^{k}\mathbb{I}\left\{sgn\left(h_w(\bar{x})_j-h_w(\bar{x})_r\right)\neq sgn\left(y_j-y_r\right)\right\}$$
$$\leq\frac{2}{k(k-1)}\sum_{j=1}^{k-1}\sum_{r=j+1}^{k}\max\left\{0,1-sgn\left(y_j-y_r\right)w\cdot\left(x_j-x_r\right)\right\}$$

נראה כי לכל איבר בסכום, מתקיים:

$$\mathbb{I}\left\{sgn\left(h_w(\bar{x})_j-h_w(\bar{x})_r\right)\neq sgn\left(y_j-y_r\right)\right\}\leq\max\left\{0,1-sgn\left(y_j-y_r\right)w\cdot\left(x_j-x_r\right)\right\}$$

ראשית כל, הפונקציה בצד שמאל הינה אינדיקטור, דהיינו ערכיה ב $\{0,1\}$ ומכיוון שהפונקציה בצד ימין ערכה הוא לפחות 0 , נבחין כי נותר לנו לבדוק מה קורה כאשר הפונקציה מצד שמאל שווה ל1 .

וזה קורה כאשר $sgn\left(h_w(\bar{x})_j-h_w(\bar{x})_r\right)\neq sgn\left(y_j-y_r\right)$ . נניח כי $sgn\left(y_j-y_r\right)=1$

וכי $sgn\left(h_w(\bar{x})_j-h_w(\bar{x})_r\right)=-1$ .

בנוסף נבחין כי: $h_w(\bar{x})_j-h_w(\bar{x})_r = by\ definition:\quad w\cdot x_j-w\cdot x_r =$
$distributive:\quad w\cdot\left(x_j-x_r\right)<0\ by\ assumption$

$$sgn\left(y_j-y_r\right)w\cdot\left(x_j-x_r\right)=w\cdot\left(x_j-x_r\right)<0$$

ולכן- $1-sgn\left(y_j-y_r\right)w\cdot\left(x_j-x_r\right)\geq 1$

ולכן

$$\mathbb{I}\left\{sgn\left(h_w(\bar{x})_j-h_w(\bar{x})_r\right)\neq sgn\left(y_j-y_r\right)\right\}=1\leq\max\left\{0,1-sgn\left(y_j-y_r\right)w\cdot\left(x_j-x_r\right)\right\}$$

כנדרש.

כעת נניח כי $sgn\left(y_j-y_r\right)=-1$ וכי $sgn\left(h_w(\bar{x})_j-h_w(\bar{x})_r\right)=1$ .

ואז $w\cdot\left(x_j-x_r\right)\geq 0$ ולכן $sgn\left(y_j-y_r\right)w\cdot\left(x_j-x_r\right)\leq 0$ ולכן:

$$1-sgn\left(y_j-y_r\right)w\cdot\left(x_j-x_r\right)\geq 1$$

ולכן

$$\mathbb{I}\left\{sgn\left(h_w(\bar{x})_j-h_w(\bar{x})_r\right)\neq sgn\left(y_j-y_r\right)\right\}=1\leq\max\left\{0,1-sgn\left(y_j-y_r\right)w\cdot\left(x_j-x_r\right)\right\}$$

כנדרש.

c

יהי $w^*\in\mathbb{R}^d$ אותו הווקטור שמפריד את הנתונים ב$\gamma$ margin .

דהיינו מתקיים:

$$sgn\left({y_j}^i-{y_r}^i\right)w^*\cdot\left({x_j}^i-{x_r}^i\right)\geq\gamma$$

לכל $1\leq i\leq n$ ו- $1\leq j<r\leq k$ .

נבחין כי עבור $w=\frac{w^*}{\gamma}$ מתקיים:

$$sgn\left({y_j}^i-{y_r}^i\right)w\cdot\left({x_j}^i-{x_r}^i\right)\geq 1$$

ולכן: $\max\{0,1-sgn\left(y_j-y_r\right)w\cdot\left(x_j-x_r\right)\}=0$

נתבונן ב- $\sum_{i=1}^{n}\ell\left(h_w(\overline{x^i}),y^i\right)$

$$\frac{2}{k(k-1)}\sum_{j=1}^{k-1}\sum_{r=j+1}^{k}\max\{0,1-sgn\left(y_j-y_r\right)w\cdot\left(x_j-x_r\right)\}=\frac{2}{k(k-1)}\sum_{j=1}^{k-1}\sum_{r=j+1}^{k}0=0$$

ולכן כל אופטימיזציה של הhinge lost תגיע לאפס, (כי מצאנו w שמגיע לאפס) וזהו הערך המינימלי האפשרי. נבחין כי מסעיף קודם, מזעור הhinge lost לאפס גורר מזעור של הtau Kendell loss כי הוא חסום מלמטה ע"י 0 (כסכום של אינדיקטורים) וחסום מלמעלה ע"י סעיף קודם בhinge lost שהצלחנו למזער אותו ל0 ולכן אותו מזעור ממזער את ה tau-Kendall לאפס גם כן.

4. **(15 points) Gradient Descent on Smooth Functions.** We say that a continuously differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is $\beta$-smooth if for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$

$$f(\mathbf{y}) \leq f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) + \frac{\beta}{2} \|\mathbf{x} - \mathbf{y}\|^2$$

In words, $\beta$-smoothness of a function $f$ means that at every point $\mathbf{x}$, $f$ is upper bounded by a qaudratic function which coincides with $f$ at $\mathbf{x}$.

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a $\beta$-smooth and non-negative function (i.e., $f(\mathbf{x}) \geq 0$ for all $\mathbf{x} \in \mathbb{R}^n$). Consider the (non-stochastic) gradient descent algorithm applied on $f$ with constant step size $\eta > 0$:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta \nabla f(\mathbf{x}_t)$$

Assume that gradient descent is initialized at some point $\mathbf{x}_0$. Show that if $\eta < \frac{2}{\beta}$ then

$$\lim_{t \to \infty} \|\nabla f(\mathbf{x}_t)\| = 0$$

(Hint: Use the smoothness definition with points $\mathbf{x}_{t+1}$ and $\mathbf{x}_t$ to show that $\sum_{t=0}^{\infty} \|\nabla f(\mathbf{x}_t)\|^2 < \infty$ and recall that for a sequence $a_n \geq 0$, $\sum_{n=1}^{\infty} a_n < \infty$ implies $\lim_{n \to \infty} a_n = 0$. Note that $f$ is not assumed to be convex!)

לפי עצת הרמז נשתמש בהגדרת ה $\beta - smooth$ עבור $f$ האי שלילית הנתונה.

נזכיר כי $gradient\ descent\ algorithm$ גוזר ש: $x_{t+1} = x_t - \eta\ \Lambda f(x_t)$

(*) ולכן: $x_{t+1} - x_t = -\eta\ \Lambda f(x_t)$

נציב בהגדרה את $x_{t+1}$ ואת $x_t$:

$$f(x_{t+1}) \leq f(x_t) + \Lambda f(x_t)^T \cdot (x_{t+1} - x_t) + \frac{\beta}{2} \|x_t - x_{t+1}\|^2$$

נציב את (*):

$$f(x_{t+1}) - f(x_t) \leq \Lambda f(x_t)^T \cdot (-\eta\ \Lambda f(x_t)) + \frac{\beta}{2} \|\eta\ \Lambda f(x_t)\|^2$$

לפי הגדרות ה dot product והנורמה:

$$f(x_{t+1}) - f(x_t) \leq -\eta \|\ \Lambda f(x_t)\|^2 + \frac{\beta}{2} \eta^{\ 2} \|\Lambda f(x_t)\|^2$$

$$f(x_{t+1}) - f(x_t) \leq \eta \left(\frac{\beta}{2}\eta - 1\right) \|\Lambda f(x_t)\|^2$$

מאחר שנתון כי $\eta < \frac{\beta}{2}$ נקבל כי $0 > \left(\frac{\beta}{2}\eta - 1\right)$ ולכן$(\eta > 0)$ :

$$\frac{f(x_{t+1}) - f(x_t)}{\eta\left(\frac{\beta}{2}\eta - 1\right)} \geq \|\Lambda f(x_t)\|^2$$

שוב לפי עצת הרמז נתבונן ב:

$$\sum_{t=0}^{\infty}\|\Lambda f(x_t)\|^2 \leq by\ previous\ line\ \sum_{t=0}^{\infty}\frac{f(x_{t+1})-f(x_t)}{\eta\left(\frac{\beta}{2}\eta-1\right)}$$

$$= \eta, \frac{\beta}{2}, are\ constants\ \frac{1}{\eta\left(\frac{\beta}{2}\eta-1\right)}\sum_{t=0}^{\infty}f(x_{t+1})-f(x_t)$$

$$= \ changing$$

$$- location\ \eta, \frac{\beta}{2}, are\ constants\ \frac{1}{\eta\left(1-\frac{\beta}{2}\eta\right)}\sum_{t=0}^{\infty}f(x_t)-f(x_{t+1})$$

$$= \frac{1}{\eta\left(1-\frac{\beta}{2}\eta\right)}\lim_{t\to\infty}\sum_{t=0}^{\infty}f(x_t)-f(x_{t+1}) = \ telescope\ series$$

$$= \frac{f(x_0)}{\eta\left(1-\frac{\beta}{2}\eta\right)} = constant < \ \infty$$

נבחין כי הסדרה $\|\Lambda f(x_t)\|^2$ הינה חיובית (מהגדרת הנורמה), ולכן וכפי שרשום ברמז,

$\sum_{t=0}^{\infty}\|\Lambda f(x_t)\|^2 < \infty$ גורר כי $\lim_{t\to\infty}\|\Lambda f(x_t)\|^2 < \infty$ ומחדוא 1 אנו יודעים כי $\lim_{t\to\infty}\|\Lambda f(x_t)\| < \infty$ כנדרש.

# *Programming assignment:*

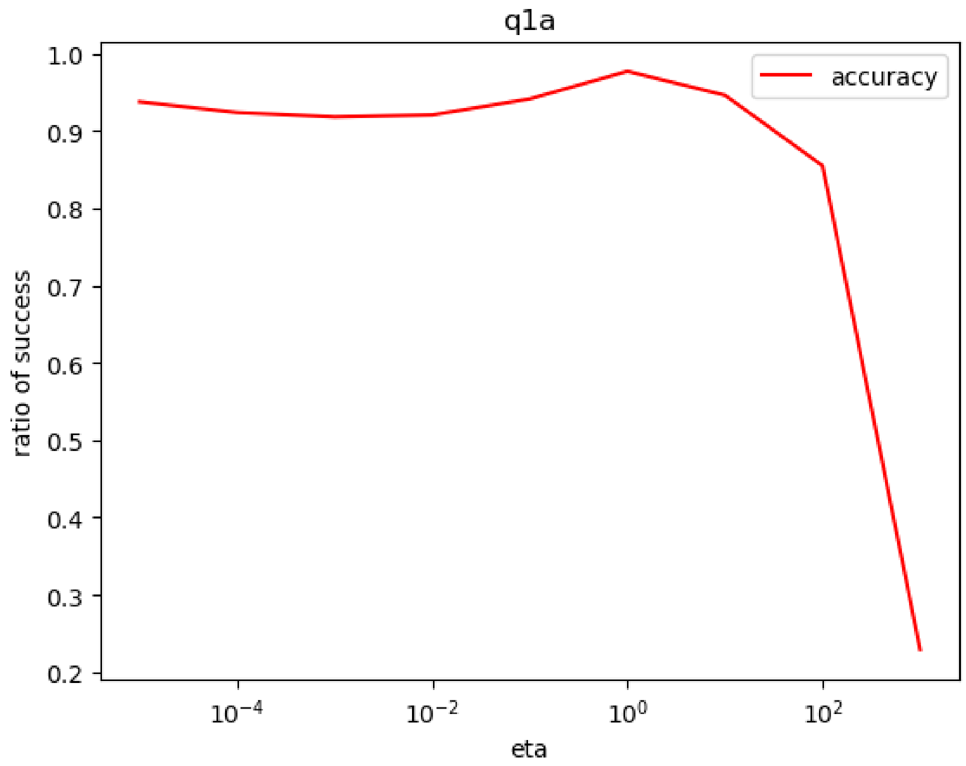
1. **(20 points) SGD for Hinge loss.** We will continue working with the MNIST data set. The file template (`skeleton_sgd.py`), contains the code to load the training, validation and test sets for the digits 0 and 8 from the MNIST data. In this exercise we will optimize the Hinge loss with $L2$-regularization ($\ell(\mathbf{w}, \mathbf{x}, y) = C(\max\{0, 1 - y\langle \mathbf{w}, \mathbf{x}\rangle\}) + 0.5\|\mathbf{w}\|^2$), using the stochastic gradient descent implementation discussed in class. Namely, we initialize $\mathbf{w}_1 = 0$, and at each iteration $t = 1, \ldots$ we sample $i$ uniformly; and if $y_i \mathbf{w}_t \cdot x_i < 1$, we update:

$$\mathbf{w}_{t+1} = (1 - \eta_t)\mathbf{w}_t + \eta_t C y_i \mathbf{x}_i$$

and $\mathbf{w}_{t+1} = (1 - \eta_t)\mathbf{w}_t$ otherwise, where $\eta_t = \eta_0 / t$, and $\eta_0$ is a constant. Implement an SGD function that accepts the samples and their labels, $C$, $\eta_0$ and $T$, and runs $T$ gradient updates as specified above. In the questions that follow, make sure your graphs are meaningful. Consider using `set_xlim` or `set_ylim` to concentrate only on a relevant range of values.

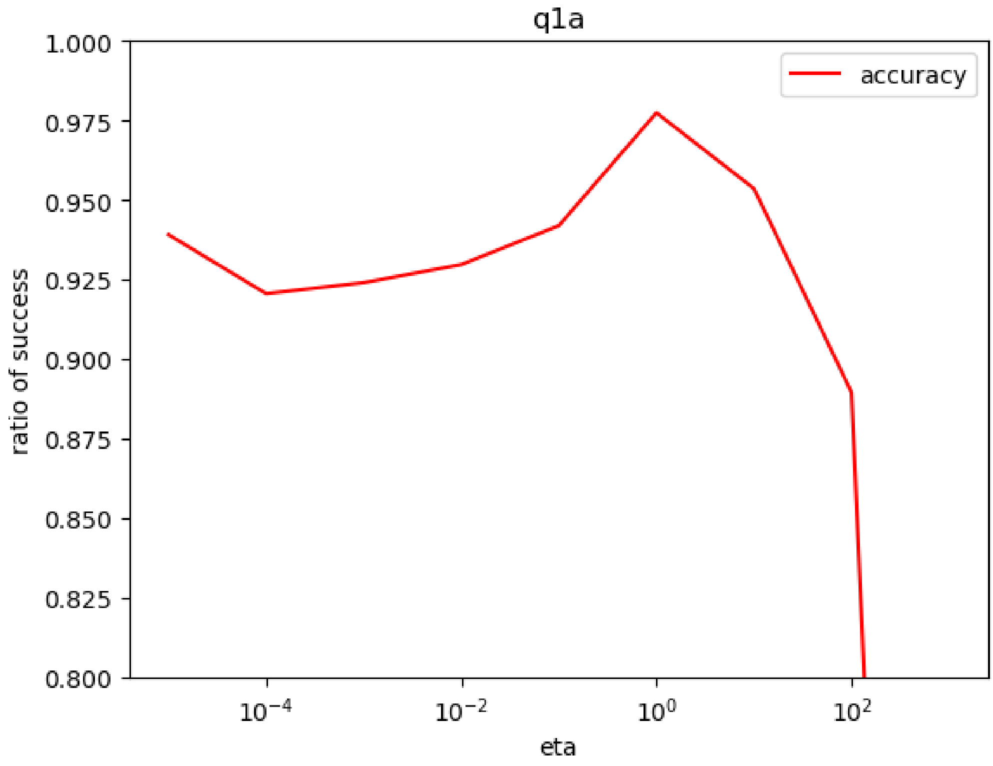
   (a) **(5 points)** Train the classifier on the training set. Use cross-validation on the validation set to find the best $\eta_0$, assuming $T = 1000$ and $C = 1$. For each possible $\eta_0$ (for example, you can search on the log scale $\eta_0 = 10^{-5}, 10^{-4}, \ldots, 10^4, 10^5$ and increase resolution if needed), assess the performance of $\eta_0$ by averaging the accuracy on the validation set across 10 runs. Plot the average accuracy on the validation set, as a function of $\eta_0$.

*First, I trained the classifier on the training set. Then I used cross-validation on the validation set with eta0 on the log scale, and plotted the result:*




*As we can see, there is a slight pick around eta = 1.*

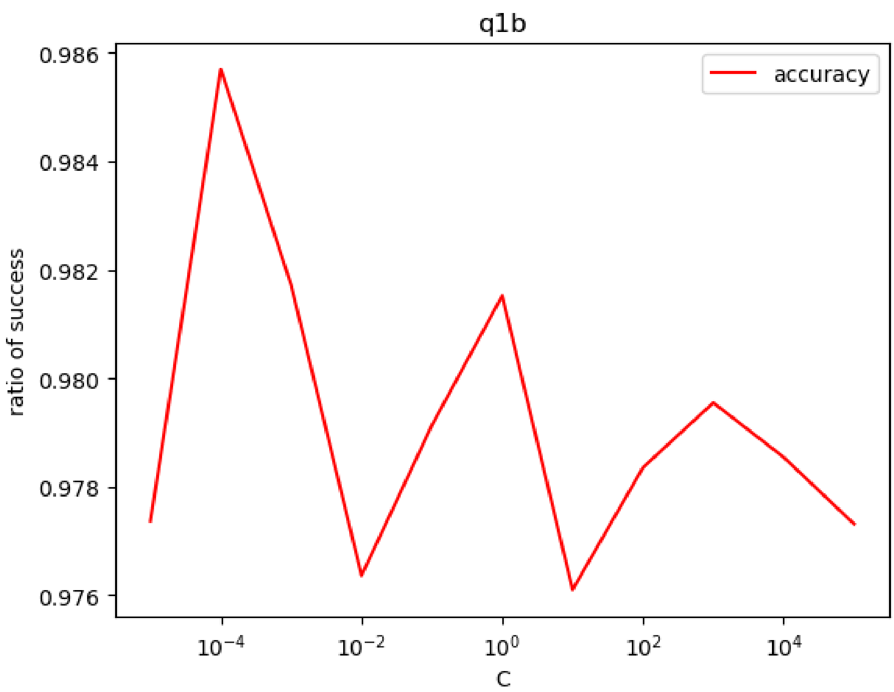
*I reset the y limit to observe the result better:*




*We can deduct that the best eta0 is 1 with accuracy ~ 0.975.*

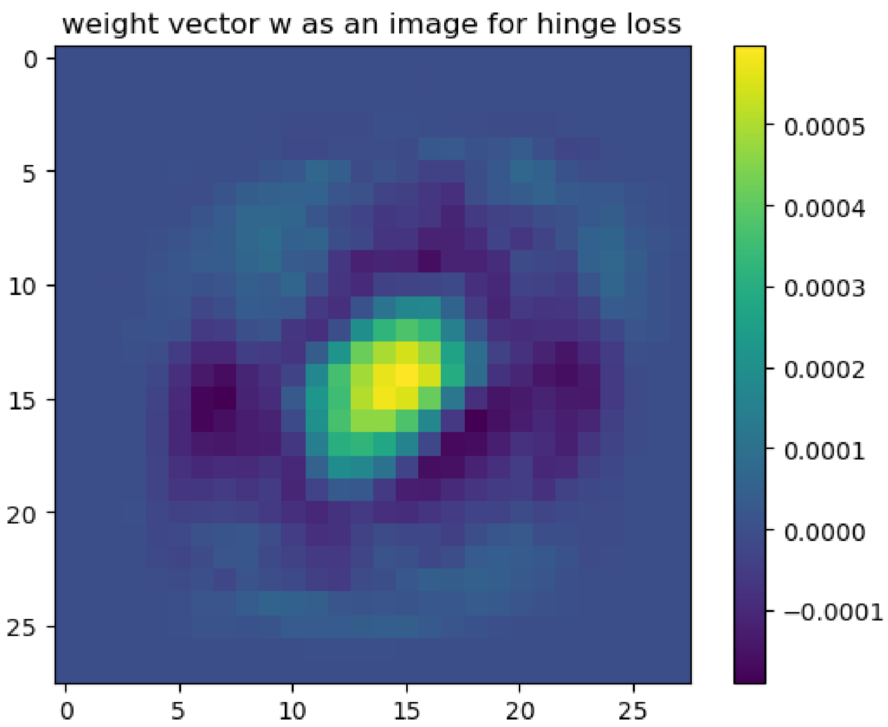
(b) **(5 points)** Now, cross-validate on the validation set to find the best $C$ given the best $\eta_0$ you found above. For each possible $C$ (again, you can search on the log scale as in section (a)), average the accuracy on the validation set across 10 runs. Plot the average accuracy on the validation set, as a function of $C$.

*Second, I trained the classifier on the training set. Then I used cross-validation on the validation set with C on the log scale, and plotted the result:*



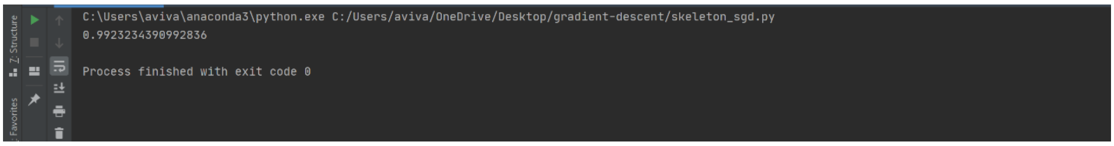
*We can deduct that the best C is* $10^{-4}$ *with accuracy ~ 0.985.*

(c) **(5 points)** Using the best $C$, $\eta_0$ you found, train the classifier, but for $T = 20000$. Show the resulting $\mathbf{w}$ as an image, e.g. using the following `matplotlib.pyplot` function: `imshow(reshape(image, (28, 28)), interpolation='nearest')`. Give an intuitive interpretation of the image you obtain.




*As we can observe, this image looks like a combination of 0 and 8, because there is a dark circle which surrounded by a bright circle that indicates the 0 figure. Moreover, the dark part makes a plus "+" sign towards the bright circle plus the very bright line in the center of the image which we can infer the center part of the 8 digit, therefore the bright colors make the 8 digit.*

(d) **(5 points)** What is the accuracy of the best classifier on the test set?




*The accuracy on the test set with eta0=1, c=*$10^{-4}$ *and T=20000 was 99.23%.*

2. **(15 points) SGD for log-loss.** In this exercise we will optimize the log loss defined as follows:

$$\ell_{log}(\mathbf{w}, \mathbf{x}, y) = \log\left(1 + e^{-y\mathbf{w}\cdot\mathbf{x}}\right)$$

(in the lecture you defined the loss with $\log_2(\cdot)$, but for optimization purposes the logarithm base doesn't matter). Derive the gradient update for this case, and implement the appropriate SGD function.

- In your computations, it is recommended to use `scipy.special.softmax` to avoid numerical issues which arise from exponentiating very large numbers.

(a) **(5 points)** Train the classifier on the training set. Use cross-validation on the validation set to find the best $\eta_0$, assuming $T = 1000$. For each possible $\eta_0$ (for example, you can search on the log scale $\eta_0 = 10^{-5}, 10^{-4}, \ldots, 10^4, 10^5$ and increase resolution if needed), assess the performance of $\eta_0$ by averaging the accuracy on the validation set across 10 runs. Plot the average accuracy on the validation set, as a function of $\eta_0$.

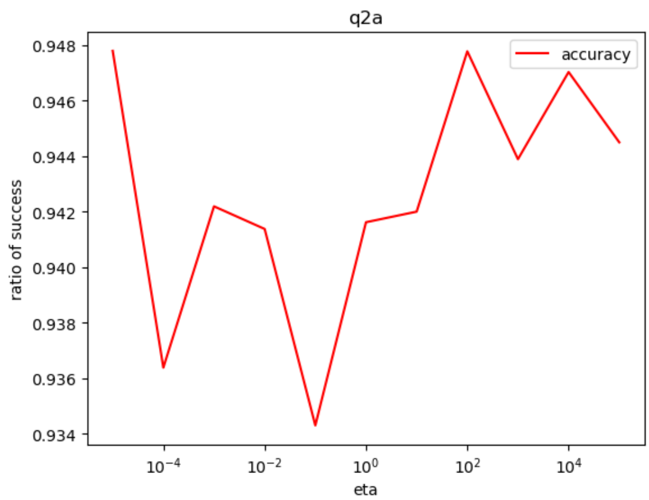
*First, we need to derive the log-loss to find the gradient update.*

$$\nabla fi(w) = \nabla(\log(1 + e^{-y_i w\cdot x_i}) = \nabla(\log(1 + e^{-y_i \sum_{j=0}^{n} w_j x_{ij}})$$

$$\frac{\partial fi(w)}{\partial w_j} = (assuming\ base\ of\ \log is\ e)\frac{(-y_i x_{ij})e^{-y_i \sum_{j=0}^{n} w_j x_{ij}}}{1 + e^{-y_i \sum_{j=0}^{n} w_j x_{ij}}}$$

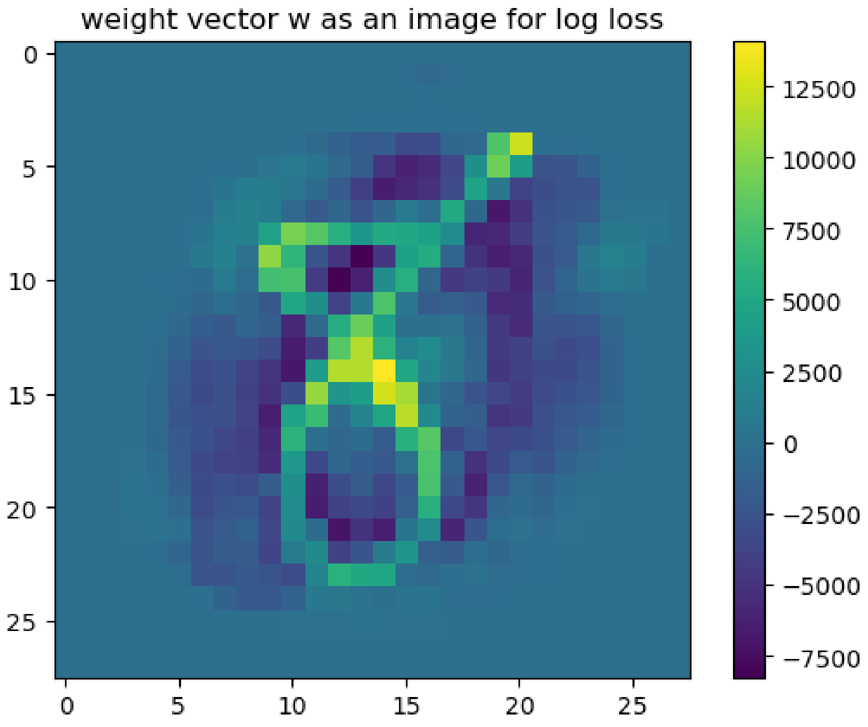
$$\nabla fi(w) = -y_i x_i\left(1 - \frac{1}{1 + e^{-y_i w\cdot x_i}}\right)$$

I used the previous calculation in my code to find the best eta0 possible:




As we can see, we can either choose eta0 = 10^-5 or 10^2, both with accurate ratio of ~ 0.948.

(b) **(5 points)** Using the best $\eta_0$ you found, train the classifier, but for $T = 20000$. Show the resulting $\mathbf{w}$ as an image. What is the accuracy of the best classifier on the test set?




we can see clearly that the bright colors displaying the 8 figure, and the dark colors displaying the 0 figure.

The accuracy of the best classifier on the test set was: 94.17%


(c) **(5 points)** Train the classifier for $T = 20000$ iterations, and plot the norm of $\mathbf{w}$ as a function of the iteration. How does the norm change as SGD progresses? Explain the phenomenon you observe.

As we can see there is a jump in the w weight vector norm value from 0 to 0.01 at the beginning of the iterations and afterwards, the w norm is staying at 0.0108. It makes sense that the w norm converges.

That because we are decreasing eta on each step

and therefore, we change w very little from

each iteration to another.