מבוא לבינה חישובית מועד ב'

שאלה 1

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 γ should be tight.

$$w_{t+1}\cdot w^* = (w_t + \eta_t y_t x_t) w^* = (w_t + \frac{1}{\sqrt{t}} y_t x_t) w^*$$

$$= w_t \cdot w^* + \frac{1}{\sqrt{t}} y_t w^* \cdot x_t \geq w_t \cdot w^* + \frac{1}{\sqrt{t}} \cdot \gamma$$

$y_t w^* \cdot x_t \geq \gamma$

אחרי M טעויות מתקבל:

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

נוסף עפ"י $\|w_t\|_2 \cdot \gamma$:

אחרי כל טעות:

$$\|w_{t+1}\|_2^2 = \|w_t + \frac{1}{\sqrt{t}} y_t x_t\|_2^2 = \|w_t\|_2^2 + \underbrace{\frac{2}{\sqrt{t}} y_t x_t w_t}_{<0 \text{ כי טעות}} + \frac{1}{t}\|x_t\|$$

$$\leq \|w_t\|_2^2 + \frac{1}{t}\|x_t\| \quad , \|x_t\| = 1$$

אחרי M טעויות:

$$\|w_{t+1}\|_2^2 \leq \sum_{t=1}^{M} \frac{1}{t}$$

סה"כ:

$$\|w_{t+1}\|_2 \leq \sqrt{\sum_{t=1}^{n} \frac{1}{t}} \sim \sqrt{\log(n)}$$

טור הרמוני

ובה"כ

$$w_t \cdot w^* \geq \gamma\sqrt{n}, \quad \|w_t\|_2 \leq \sqrt{\log(n)} \qquad \|w^*\|_2 = 1$$

ומא"ש קושי שוורץ:

$$\gamma\sqrt{n} \leq w_t \cdot w^* \leq \|w_t\|_2 \|w^*\|_2 \leq \sqrt{\log(n)} \cdot 1$$

כלומר

$$\gamma\sqrt{n} \leq \sqrt{\log(n)} \Rightarrow \sqrt{n} \leq \frac{1}{\gamma}\sqrt{\log(n)}$$

$$n \leq \frac{1}{\gamma^2}\log(n) \Rightarrow n \leq \frac{2}{\gamma^2} \cdot \log\left(\frac{1}{\gamma^2}\right)$$

כנ"ל

$$n \leq \frac{4}{\gamma^2}\log\left(\frac{1}{\gamma}\right)$$

כמובן

$$\log_2(x^2) = 2\log(x)$$

שאלה 2

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 $f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})$, 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 ℓ_{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) יהי $x,y\in\tilde{\mathbb{R}^n}$ כך ש: $\lambda\in[0,1]$

$$g(\lambda x+(1-\lambda)y)=f(A(\lambda x+(1-\lambda)y)+b)$$

$$=f(A\lambda x+A(1-\lambda)y+b)=f(\lambda Ax+(1-\lambda)\cdot Ay+b)$$

$$=f(\lambda Ax+(1-\lambda)Ay+((1-\lambda)+\lambda)b)$$

$$=f(\lambda(Ax+b)+(1-\lambda)(Ay+b))$$

מתקיים $Ax+b, Ay+b\in\tilde{\mathbb{R}^n}$ לפי מה שהוכחנו מהקבץ

$$\leqslant \lambda f(Ax+b)+(1-\lambda)f(Ay+b)=\lambda g(x)+(1-\lambda)g(y)$$

כמבוקש. קיבלנו שקיים $x,y\in\mathbb{R}^n$ עבור כל $\lambda\in[0,1]$:

$$g(\lambda x+(1-\lambda)y)\leqslant\lambda g(x)+(1-\lambda)g(y)$$

סיימנו g קמורה לפי ההגדרה.

(b) יהיו $x, y \in \mathbb{R}^n$ ויהי $\lambda \in [0,1]$ שכך מתקיים לכל f_i:

$$\forall i \in [n] \quad f_i(\lambda x + (1-\lambda)y) \leq \lambda f_i(x) + (1-\lambda) f_i(y)$$

נסמן:

$$g(\lambda x + (1-\lambda)y) = \max_i f_i(\lambda x + (1-\lambda)y)$$

$$\leq \max_i \{\lambda f_i(x) + (1-\lambda) f_i(y)\}$$

$$\leq \lambda \max_i \{f_i(x)\} + (1-\lambda) \max_i \{f_i(y)\}$$

$$\leq \lambda g(x) + (1-\lambda) g(y)$$

לכן g קמורה.

(c) נסיק כי $\log$ קמורה:

$$\log'(2) = \frac{1}{\ln(2) \cdot (1+e^{-2})} \cdot -e^{-2} = -\frac{e^{-2}}{\ln(2) \cdot (1+e^{-2})}$$

$$= -\ln(2)^{-1} \cdot \frac{e^{-2}}{1+e^{-2}}$$

$$\log''(2) = -\ln(2)^{-1} \cdot \left[\frac{-e^{-2}(1+e^{-2}) - e^{-2} \cdot (-e^{-2})}{(1+e^{-2})^2}\right]$$

$$= -\ln(2)^{-1} \cdot \left[\frac{-e^{-2} - e^{-2\cdot 2} + e^{-2\cdot 2}}{(1+e^{-2})^2}\right] = -\ln(2)^{-1} \cdot \frac{-e^{-2}}{(1+e^{-2})^2}$$

מתקיים $-\ln(2)^{-1} < 0$, $-e^{-2} < 0$, $(1+e^{-2})^2 > 0$

לכן כפל המספרים מוביל לכך שמתקיים

$$\log''(2) > 0$$

לכן $\log$ קמורה. (משפט שהוכחנו)

כנ"ל:

$$f(w) = \log_b(y^{w \cdot x}) \underset{x,w \in \mathbb{R}^n}{=} \log_b((y^x)^w)$$

יהיו $z_1, z_2 \in \mathbb{R}^n$ ויהי $\lambda \in [0,1]$ כך:

$$f(\lambda z_1 + (1-\lambda) z_2) = \log_b((y^x)^{(\lambda z_1 + (1-\lambda) z_2)})$$

$$= \log_b(y^{x\lambda z_1 + y^{x(1-\lambda) z_2}})$$

$$= \log_b(\lambda(y^{x z_1}) + (1-\lambda)(y^{x z_2}))$$

מכיוון שלוגריתם (בסיס b) הוא קמור:

$$\leq \lambda \cdot \log_b(y^{x z_1}) + (1-\lambda)\log_b(y^{x z_2})$$

$$= \lambda \cdot f(z_1) + (1-\lambda) \cdot f(z_2)$$

לכן f הוא פונקציה קמורה.

שאלה 3

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: $\sum_{i=1}^n \Delta(h_{\mathbf{w}}(\bar{\mathbf{x}}^i), \mathbf{y}$
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.

(א) אוכיח ש $\sum_{i=1}^{n} l(h_w(\bar{x}^i), y^i)$ היא פונקציה קמורה. מתקיים:

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

נשים לב שאנו מבצעים סכום של הביטויים:

$$\max\{0, 1 - sgn(y_j - y_r) w \cdot (x_j - x_r)\}$$

ברור כי 0 היא פונקציה לינארית ולכן קמורה, כמו כן:

$$1-\text{sgn}(y_j-y_r)w\cdot(x_j-x_r)$$

$$=(x_j-x_r)\cdot(-\text{sgn}(y_j-y_r))w+1$$

כלומר ניתן לכתוב את הביטוי כ $mw+n$ ולכן זו
פונקציה לינארית גם ולכן קמורה.
לכן בסה"כ מקסימום על הפונקציות האלה יתן פונקציה
קמורה. ובסה"כ מתקבל שהפונקציה $\ell(h_w(x),y)$ ובפרט
גם $\ell(h_w(x^i),y^i)$ קמורה ולכן הפונקציה
$\sum_{i=1}^{m}\ell(h_w(x^i),y^i)$ קמורה כסכום של פונקציות קמורות.

(ט) נכתוב בפונקציות:

$$\Delta(h_w(\bar{x}),y)=\frac{2}{k(k-1)}\sum_{j=1}^{k-1}\sum_{r=j+1}^{k}1\{sgn(h_w(\bar{x})_j-h_w(\bar{x})_r)\neq sgn(y_j-y_r)\}$$

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

כעת נפריד למקרים:

1) אם $sgn(y_j-y_r)w\cdot(h_w(\bar{x}_j)-h_w(\bar{x}_r))\geq 0$ אז בהכרח $sgn(y_j-y_r)=sgn(h_w(\bar{x}_j)-h_w(\bar{x}_r))$ ולכן נוסיף 0 ב-$\Delta(h_w(\bar{x}),y)$, ולעומת זאת ב-$\ell(h_w(\bar{x}),y)$ נוסיף ביטוי ≥ 0.

2) אם $x=sgn(y_j-y_r)w\cdot(h_w(\bar{x}_j)-h_w(\bar{x}_r))<0$ אז בהכרח $sgn(y_j-y_r)\neq sgn(h_w(\bar{x}_j)-h_w(\bar{x}_r))$ ולכן נוסיף 1 לסכום ב-$\Delta(h_w(\bar{x}),y)$, ולעומת זאת ב-$\ell(h_w(\bar{x}),y)$ מתקבל:

$$\max\{0,1-x\}\underset{x<0}{=}\max\{0,1+y\}\underset{y>0}{=}1+y$$

כלומר נוסיף ביטוי שגדול מ-1.

בסה"כ קיבלנו שכל מחובר בסכום $\Delta(h_w(\bar{x}),y)$ קטן מכל מחובר בסכום $\ell(h_w(\bar{x}),y)$ כשהמקדם $\frac{2}{k(k-1)}$ זהה בשתי הפונקציות וגם האינדקסים בסכומים.

לכן בהכרח:

$$\Delta(h_w(\bar{x}),y)\leq\ell(h_w(\bar{x}),y)\quad\forall x\in\mathbb{R}^d,\ x\in X^k,\ y\in\mathbb{R}^k$$

כנדרש.

(c) הdata ניתן להפרדה לכן:

$\exists w^* \in R^d,\ \gamma > 0$ and:

$sgn(y_j^i - y_r^i) w^* \cdot (x_j^i - x_r^i) \geq \gamma > 0$ (עבור $\gamma > 0$ כלשהו)

$\forall 1 \leq i \leq n,\ 1 \leq j < r < k$

$\forall 1 \leq i \leq n,\ 1 \leq j < r < k$

נגדיר $\bar{w} = \frac{1}{\gamma} w^*$ ואז

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

ה$\bar{w}$ הזה ישים 0 בפונקצית hinge loss כאשר:

$$\forall_i : \max\{0, 1 - sgn(y_j^i - y_r^i) \bar{w} \cdot (x_j^i - x_r^i)\} = 0$$

אכן ראינו ש

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

לכן

$$1 - sgn(y_j^i - y_r^i) \bar{w} \cdot (x_j^i - x_r^i) \leq 0$$

ונקבל

$$\max\{0, 1 - sgn(y_j^i - y_r^i) \bar{w} \cdot (x_j^i - x_r^i)\} = 0$$

לכל הdata. לכן w_{min} שממזער את הhinge loss בהכרח ישים 0 (כלומר zero loss). נותר להראות שאכן הw_{min} מסווג את הdata נכון:

$$\text{zero hinge loss} \Longrightarrow sgn(y_j^i - y_r^i) w_{min} \cdot (x_j^i - x_r^i) \geq 1$$

$\Longrightarrow sgn(y_j^i - y_r^i)\,(x_j^i - x_r^i)\cdot w_{min} \geq 1$

כלומר הסימנים של $(y_j^i - y_r^i)$ ושל

$(x_j^i - x_r^i)\cdot w_{min}$ מסכימים לכן w_{min} מסווג נכון את הdata.

בסה"כ הראנו שניתן להשיג שגיאה 0 עם hinge loss

ומשפט קודם $\Delta(h_{\mathbf{w}}(\bar{\mathbf{x}}), \mathbf{y}) \leq \ell(h_{\mathbf{w}}(\bar{\mathbf{x}}), \mathbf{y})$ for all $\mathbf{w} \in \mathbb{R}^d, \bar{\mathbf{x}} \in \mathcal{X}^k, \mathbf{y} \in \mathbb{R}^k$.

לכן ניתן להשיג שגיאה 0 גם עם kendall-Tau loss

נזכיר:

hinge loss =

$\sum_{i=1}^{n} \ell(h_{\mathbf{w}}(\bar{\mathbf{x}}^i), \mathbf{y}^i)$ where:

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

kendall-Tau loss =

$\sum_{i=1}^{n} \Delta(h_{\mathbf{w}}(\bar{\mathbf{x}}^i), \mathbf{y}^i)$. where:

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

שאלה 4

4. **(15 points) Gradient Descent on Smooth Functions.** We say that a continuously differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is β-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, β-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 β-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!)

נתון שפונקציה β-חלקה לכן מהגדרה לכל $x_{t+1}, x_t \in \mathbb{R}^n$ מתקיים כי:

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

וכיוון ש: $x_{t+1} = x_t - \eta \nabla f(x_t)$

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

ומכאן

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

$$= -\eta \nabla f(x_t)^T(\nabla f(x_t)) + \eta^2 \cdot \frac{\beta}{2} \cdot \|\nabla f(x_t)\|^2$$

$$= -\eta \|\nabla f(x_t)\|^2 + \eta^2 \cdot \frac{\beta}{2} \cdot \|\nabla f(x_t)\|^2$$

כלומר:

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

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

$$\sum_{t=0}^{\infty} \|\nabla f(x_t)\|^2 \cdot \eta\left(1-\frac{\eta\beta}{2}\right) = \eta\left(1-\frac{\eta\beta}{2}\right)\cdot \sum_{t=0}^{\infty} \nabla f(x_t)$$

$$\leq \sum_{t=0}^{\infty} f(x_t)-f(x_{t+1}) = f(x_0)-f(x_T+1) \qquad T\in\mathbb{N}$$

טור טלסקופי

עבור $0<\eta<\frac{2}{\beta}$ מתקיים $1-\frac{\eta\beta}{2}>0$ ולכן $\eta\left(1-\frac{\eta\beta}{2}\right)>0$

ואז:

$$\sum_{t=0}^{\infty} \eta\left(1-\frac{\eta\beta}{2}\right)\|\nabla f(x_t)\|^2 \leq f(x_0)-f(x_{T+1})$$

$$\Longrightarrow \sum_{t=0}^{\infty} \|\nabla f(x_t)\|^2 \leq \frac{f(x_0)-f(x_{T+1})}{\eta\left(1-\frac{\eta\beta}{2}\right)}$$

נתון שf חיובית ולכן

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

קבוע חיובי

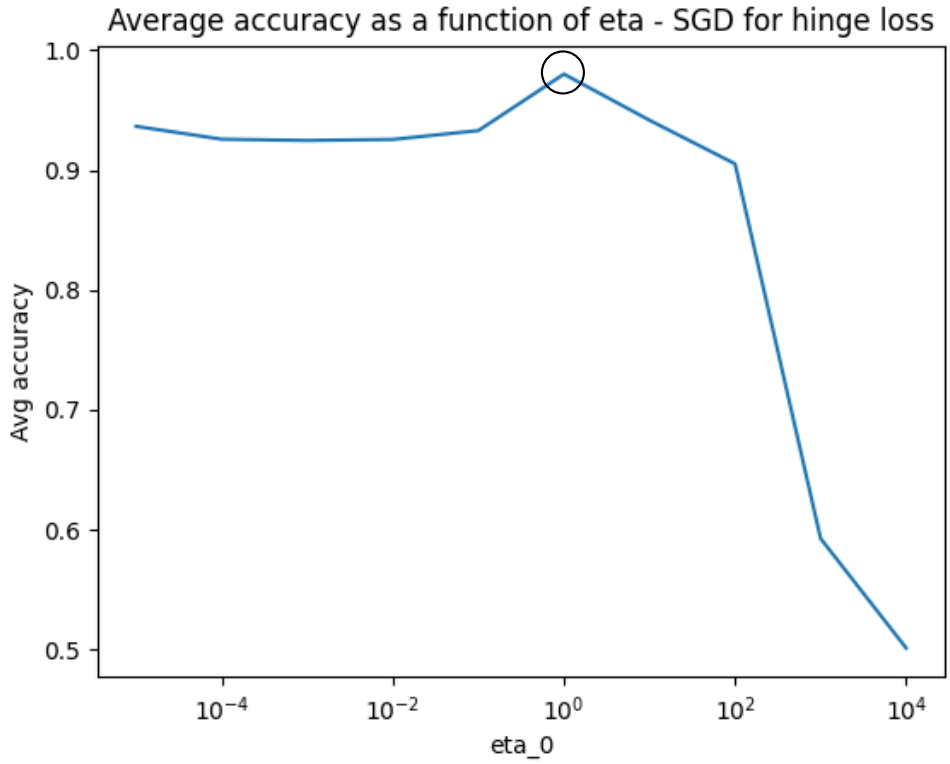
כמו כן $\|\nabla f(x_t)\|^2 \geq 0$ ולכן לפי הרמז נסיק $\lim_{t\to\infty} \|\nabla f(x_t)\|^2 = 0$ כנדרש

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:

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

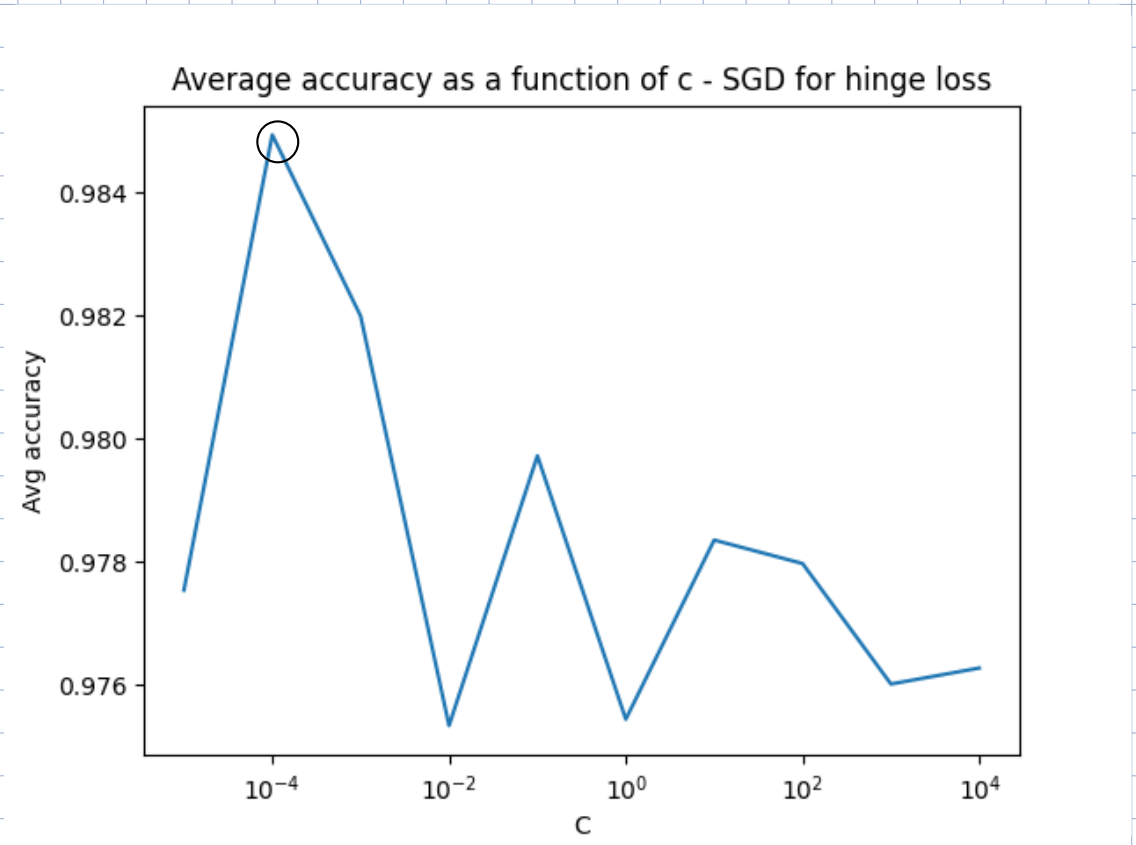
and $\mathbf{w}_{t+1} = (1 - \eta_t)\mathbf{w}_t$ otherwise, where $\eta_t = \eta_0/t$, and η_0 is a constant. Implement an SGD function that accepts the samples and their labels, C, η_0 and T, and runs T gradient updates as specified above. In the questions that follow, make sure your graphs are meaningful. 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 η_0, assuming $T = 1000$ and $C = 1$. For each possible η_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 η_0 by averaging the accuracy on the validation set across 10 runs. Plot the average accuracy on the validation set, as a function of η_0.




```
The best eta is: 1
```

(b) **(5 points)** Now, cross-validate on the validation set to find the best C given the best η_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.



The best C is: 0.0001

(c) **(5 points)** Using the best C, η_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.

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

```
The accuracy of the best classifier is: 99.2835209825998 %
```

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 η_0, assuming $T = 1000$. For each possible η_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 η_0 by averaging the accuracy on the validation set across 10 runs. Plot the average accuracy on the validation set, as a function of η_0.

(a) נחשב גרדיאנט:

$$\frac{\partial \ell_{log}}{\partial w_i} = \frac{\partial\left(\log\left(1+e^{-y_i w x_i}\right)\right)}{\partial w_j}$$

$$= \frac{1}{1+e^{-y_i w \cdot x_i}} \cdot e^{-y_i w \cdot x} \cdot (-y_i x_i)$$

כמוכן:

$$1 - \frac{1}{1+e^{-y_i w x_i}} = 1 - \text{expit}(y_i w x_i) = \frac{\cancel{1}+e^{-y_i w x_i} - \cancel{1}}{1+e^{-y_i w x_i}}$$

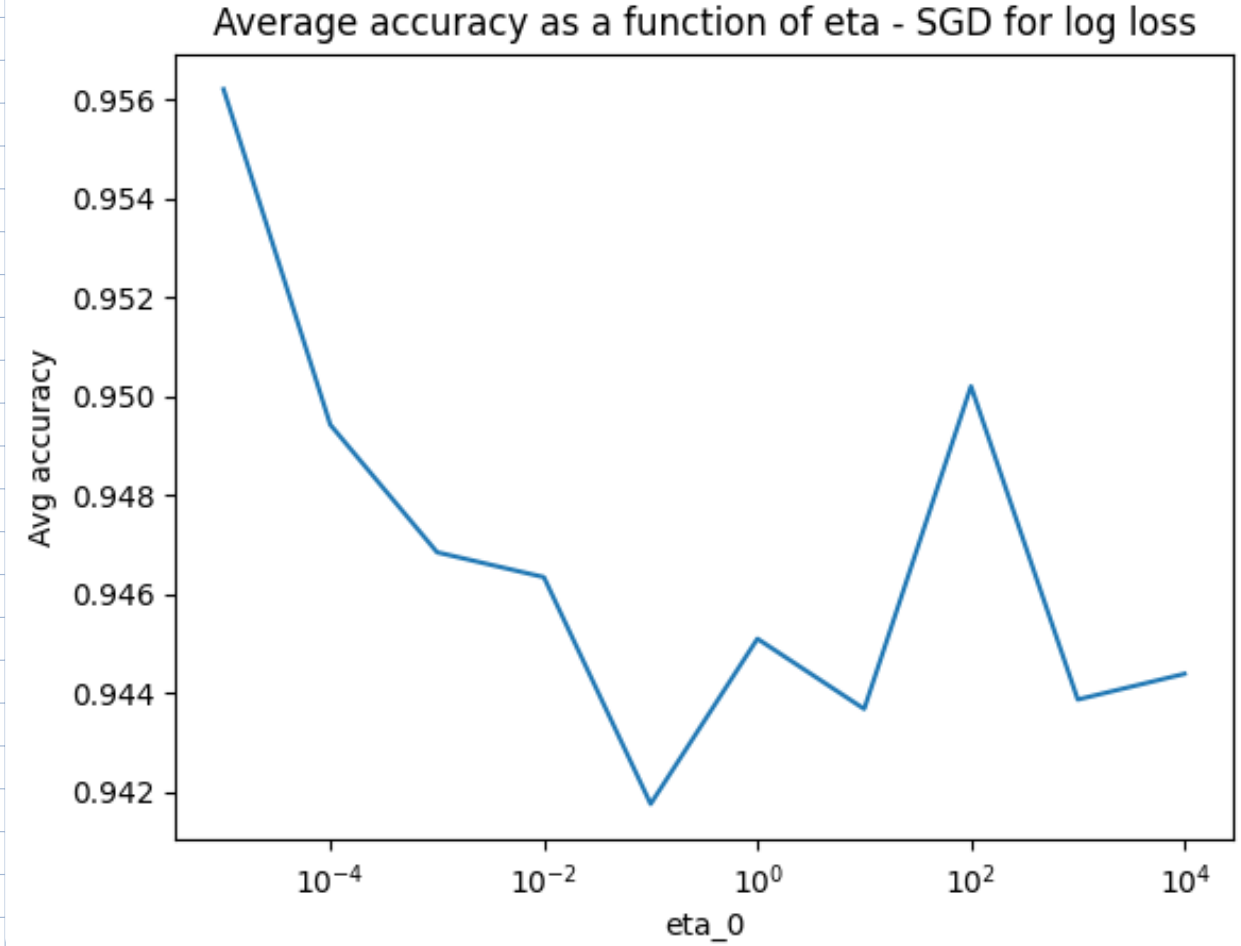
$$= \frac{1}{1+e^{-y_i w x_i}} \cdot e^{-y_i w x_i}$$

לכן השתמשתי בקוד בפונקציה expit, בסה"כ:

$$\nabla \ell_{log} = \left(\frac{e^{-y_i w_1 x}}{1+e^{-y_i w_1 \cdot x_i}} \cdot (-y_i x_i), \ldots, \frac{e^{-y_i w_t x}}{1+e^{-y_i w_t \cdot x_i}} \cdot (-y_i x_i)\right)$$

$$w_{t+1} = w_t - \eta_t \cdot \nabla \ell_{log}$$

:הפרמטר הטוב ביותר




```
The best eta is: 1e-05
```

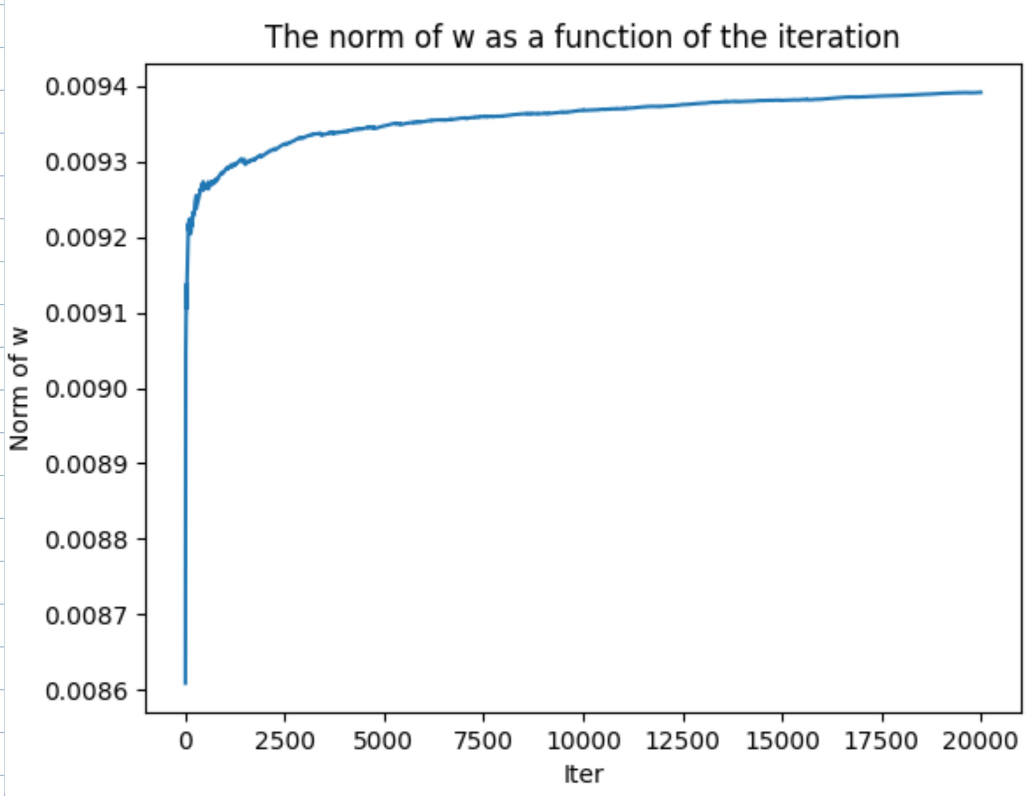
:הפלט

(b) **(5 points)** Using the best η_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?


```
The accuracy of the best classifier is: 97.54350051177073 %
```

(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.

(C)



שים לב שהנורמה גדלה, בהתחלה הנורמה גדלה בקצב מהיר
יותר שכן בהתחלה הצעדים גדולים יותר, לקראת סוף האיטרציות
מבצעים צעדים קטנים יותר ולכן השינוי בנורמה קטן.