מבוא ללמידה חישובית מטלה 4

Theory Questions

1. **(15 points) SVM with multiple classes.** One limitation of the standard SVM is that it can only handle binary classification. Here is one extension to handle multiple classes. Let $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ and now let $y_1, \ldots, y_n \in [K]$, where $[K] = \{1, 2, \ldots, K\}$. We will find a separate classifier $\mathbf{w}_j$ for each one of the classes $j \in [K]$, and we will focus on the case of no bias ($b = 0$). Define the following loss function (known as the *multiclass hinge-loss*):

$$\ell(\mathbf{w}_1, \ldots, \mathbf{w}_K, \mathbf{x}_i, y_i) = \max_{j \in [K]} (\mathbf{w}_j \cdot \mathbf{x}_i - \mathbf{w}_{y_i} \cdot \mathbf{x}_i + \mathbb{1}(j \neq y_i)),$$

where $\mathbb{1}(\cdot)$ denotes the indicator function. Define the following multiclass SVM problem:

$$f(\mathbf{w}_1, \ldots, \mathbf{w}_K) = \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{w}_1, \ldots, \mathbf{w}_K, \mathbf{x}_i, y_i)$$

After learning all the $\mathbf{w}_j$, $j \in [K]$, classification of a new point $\mathbf{x}$ is done by $\arg\max_{j \in [K]} \mathbf{w}_j \cdot \mathbf{x}$. The rationale of the loss function is that we want the "score" of the true label, $\mathbf{w}_{y_i} \cdot \mathbf{x}_i$, to be larger by at least 1 than the "score" of each other label, $\mathbf{w}_j \cdot \mathbf{x}_i$. Therefore, we pay a loss if $\mathbf{w}_{y_i} \cdot \mathbf{x}_i - \mathbf{w}_j \cdot \mathbf{x}_i \leq 1$, for $j \neq y_i$.

Consider the case where the data is linearly separable. Namely, there exists $\mathbf{w}_1^*, ..., \mathbf{w}_K^*$ such that $y_i = argmax_y \mathbf{w}_y^* \cdot \mathbf{x}_i$ for all i. Show that any minimizer of $f(\mathbf{w}_1, ..., \mathbf{w}_K)$ will have zero classification error.

נרצה להראות שלכל מינימייזר של $f(w_1, \ldots, w_K)$ יתקיים
ש classification error=0. כלומר שהסיווגים נכונים.
נשים לב:

$$\min_{w_1,\ldots,w_K} f(w_1, \ldots, w_K) = \min \frac{1}{n} \sum_{i=1}^{n} \ell(w_1, \ldots, w_K, x_i, y_i)$$

$$= \min_{w_1,\ldots,w_K} \frac{1}{n} \sum_{i=1}^{n} \max_{j \in [K]} (w_j \cdot x_i - w_{y_i} \cdot x_i + 1(j \neq y_i))$$

נשים לב שכאשר $j=y_i$ יתקבל בביטוי בתוך ה max:

$$w_j x_i - w_j x_i + 0 = 0$$

כאשר $j \neq y_i$ יתקבל הביטוי בתוך ה max:

$$w_j \cdot x_i - w_{y_i} \cdot x_i + 1$$

לכן בסה"כ נוכל לכתוב

$$\min_{w_1,\dots,w_k} \frac{1}{n}\sum_{i=1}^{n} \left(\max_{[k] \ni j \neq y_i} \{ w_j \cdot x_i - w_{y_i} \cdot x_i + 1, \ 0 \} \right)$$

כמובן ברור כי

$$\max_{[k] \ni j \neq y_i} \{ w_j \cdot x_i - w_{y_i} \cdot x_i + 1, \ 0 \}$$

כעת נראה שקיימים $w_1,\dots,w_k$ כך שלכל i מתקיים

$$\max_{[k] \ni j \neq y_i} \{ w_j \cdot x_i - w_{y_i} \cdot x_i + 1, \ 0 \} \leq 0$$

ומהגדרת אותו max זה יתקיים אמ"מ

$$\max_{[k] \ni j \neq y_i} (w_j \cdot x_i - w_{y_i} \cdot x_i + 1) \leq 0$$

כי אם ה max הוא הקטן, נדרוש א"ש על כל איבר

$$\Longleftrightarrow$$

$$\forall j \in [k], j \neq y_i \quad w_j \cdot x_i - w_{y_i} \cdot x_i + 1 \leq 0$$

$$\Longleftrightarrow$$

$$\forall j \in [k], j \neq y_i \quad w_j \cdot x_i - w_{y_i} \cdot x_i \leq -1$$

$$\Longleftrightarrow$$

$$\forall j \in [k], j \neq y_i \quad w_{y_i} \cdot x_i - w_j \cdot x_i \geq 1$$

בסעיף הבא נראה קיום של $w_1,\dots,w_k$ שמקיימים זאת.

נשים לב שמההנחה שה-data ניתן להפרדה לינארית אז קיימים $w_1^*,\dots,w_k^*$ כך ש $y_i=\text{argmax}_y\, w_y^*\cdot x_i$ לכל i

לכן לכל $y_i\neq j\in[k]$ מתקיים $w_{y_i}^*\cdot x_i > w_j^*\cdot x_i$.

כלומר

$$w_{y_i}^*\cdot x_i - w_j^*\cdot x_i > 0.$$

נגדיר:

$$w_k=\frac{1}{\min_{i,j\neq y_i} w_{y_i}^*\cdot x_i - w_j^*\cdot x_i}\cdot w_k^*$$

ל $k\in[k]$. נשים לב שלכל $n\geq i\geq 1$, $y_i\neq j$ מתקיים:

$$w_{y_i}\cdot x_i - w_j\cdot x_i=\frac{1}{\min_{i,j\neq y_i} w_{y_i}^*\cdot x_i - w_j^*\cdot x_i}\,w_{y_i}^*\cdot x_i$$

$$-\frac{1}{\min_{i,j\neq y_i} w_{y_i}^*\cdot x_i - w_j^*\cdot x_i}\,w_j^*\cdot x_i=\frac{w_{y_i}^*\cdot x_i - w_j^*\cdot x_i}{\min_{i,j\neq y_i} w_{y_i}^*\cdot x_i - w_j^*\cdot x_i}\geq 1 \quad (*)$$

(*) הסבר: כאונו $w_{y_i}^*\cdot x_i - w_j^*\cdot x_i>0$, ולכן המונה והמכנה חיוביים, כמו כן, המכנה הוא מציאת המינימום של המונה על פני i,j ולכן בהכרח קטן ממנו.

כאמור קבלנו שקיימים $w_1,\dots,w_k$ כך ש:

$$\forall j\in[k], j\neq y_i \quad w_{y_i}\cdot x_i - w_j\cdot x_i\geq 1$$

ולכן לכל i:

$$\max_{j\neq y_i\in[k]}\{w_j\cdot x_i - w_{y_i}\cdot x_i+1,\ 0\}=0$$

כאינו ששקם

$$\max_{j\neq y_i \in [k]}\{w_j\cdot x_i - w_{y_i}\cdot x_i+1,\ 0\} \geq 0$$

לכן בהכרח:

$$\max_{j\neq y_i \in [k]}\{w_j\cdot x_i - w_{y_i}\cdot x_i+1,\ 0\} = 0$$

כלומר $\min_{w_1,\dots,w_k} f(w_1,\dots,w_k) = 0$

שכן $f(w_1,\dots,w_k)\geq 0$ ומצאנו $w_1,\dots,w_k$ עבורם
$f(w_1,\dots,w_k)=0$ לכן בפרט עבור $w_1,\dots,w_k$ שממזערים את
f (הרי מחפשים מינימום) יתקבל 0.

כעת נראה ש $\min_{w_1,\dots,w_k} f(w_1,\dots,w_k)=0$ גורר
classification error $=0$.

יהיו $w_1,\dots,w_k$ שממזערים את f אז:

$$f(w_1,\dots,w_k) = \frac{1}{n}\sum_{i=1}^{n}\max_{j\neq y_i \in [k]}\{w_j\cdot x_i - w_{y_i}\cdot x_i+1,\ 0\} = 0$$

הפונקציה מופיעה בתחילת השאלה

נשים לב שהמחוברים בסכום אי שליליים לכן מכך שסכומם
0 נסיק שכל אחד מהם הוא 0 כלומר לכל $1\leq i\leq n$:

$$\max_{j\neq y_i \in [k]}\{w_j\cdot x_i - w_{y_i}\cdot x_i+1,\ 0\} = 0$$

כלומר:

$$w_j \cdot x_i - w_{y_i} \cdot x_i + 1 \leq 0 \Rightarrow w_j \cdot x_i - w_{y_i} \cdot x_i \leq -1$$

$$\Rightarrow w_{y_i} \cdot x_i - w_j \cdot x_i \geq 1 \Rightarrow w_{y_i} \cdot x_i \geq w_j \cdot x_i + 1$$

$$\Rightarrow w_{y_i} x_i > w_j \cdot x_i \qquad \forall 1 \leq i \leq n$$

לכן:

$$\underset{j \in [k]}{\operatorname{argmax}}\ w_j \cdot x_i = y_i$$

כאשר y_i הם הלייבלים ולכן הסיווג נעשה ללא שגיאה

כלומר classification error=0 כנדרש.

2. **(10 points) Expressivity of ReLU networks.** Consider the ReLU activation function:

$$h(x) = \max\{x, 0\}$$

Show that the maximum function $f(x_1, x_2) = \max\{x_1, x_2\}$ can be implemented using a neural network with one hidden layer and ReLU activations. You can assume that there is no activation after the last layer.

(Hint: It is possible to implement the function using a hidden layer with 4 neurons. You may find the following identity useful: $\max\{x_1, x_2\} = \frac{x_1+x_2}{2} + \frac{|x_1-x_2|}{2}$.)

אם $x \geq 0$ אז:

$x = \max\{x,0\} = h(x),\quad 0 = \max\{-x,0\} = h(-x)$

$x = x - 0 = \max\{x,0\} - \max\{-x,0\}$

כיוון: (אם $x \geq 0$ אזי)

$x \geq -x \Rightarrow |x| = x = \max\{x,0\} + \max\{-x,0\} = x + 0 = x$

אם $x < 0$ אז:

$-x = \max\{-x,0\} = h(-x),\quad 0 = \max\{x,0\} = h(x)$

$0 = x - x = x + \max\{-x,0\} = \max\{x,0\}$

$\Rightarrow x = \max\{x,0\} - \max\{-x,0\}$

כיוון: (אם $x < 0$ אזי):

$x < -x \Rightarrow |x| = -x = \max\{x,0\} + \max\{-x,0\} = 0 - x = -x$

קיבלנו עבור x:

$x = \max\{x,0\} - \max\{-x,0\} = h(x) - h(-x)$

$|x| = \max\{x, -x\} = \max\{x,0\} + \max\{-x,0\} = h(x) + h(-x)$

נוכיח:

$$\max\{x_1, x_2\} = \frac{x_1 + x_2}{2} + \frac{|x_1 - x_2|}{2}$$

$$\frac{x_1 + x_2}{2} + \frac{|x_1 - x_2|}{2} = \frac{h(x_1 + x_2) - h(-x_1 - x_2)}{2} + \frac{h(x_1 - x_2) + h(-x_1 + x_2)}{2}$$

$$= \frac{1}{2}\left[h(x + x_2) - h(-x_1 - x_2) + h(x_1 - x_2) + h(-x_1 + x_2)\right]$$

נבנה רשת נוירונים תיכאורטית:

3. **(15 points) Soft SVM with ℓ^2 penalty.** Consider the following problem:

$$\min_{\boldsymbol{w}\in\mathbb{R}^d, b\in\mathbb{R}, \boldsymbol{\xi}\in\mathbb{R}^n} \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{C}{2}\sum_{i=1}^{n}\xi_i^2$$
$$\text{s.t. } y_i(\boldsymbol{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i \quad \forall i = 1, \ldots, n$$

(a) Show that a constraint of the form $\xi_i \geq 0$ will not change the problem. Meaning, show that these non-negativity constraints can be removed. That is, show that the optimal value of the objective will be the same whether or not these constraints are present.

(b) What is the Lagrangian of this problem?

(c) Minimize the Lagrangian with respect to $\boldsymbol{w}, b, \boldsymbol{\xi}$ by setting the derivative with respect to these variables to 0.

(d) What is the dual problem?

(a) אוכיח שהערך האופטימלי של פונקציית המטרה ללא אילוצי $\xi_i \geq 0$ ועם האילוצים זהה. לשם כך נסמן ערך פונקציית המטרה ללא האילוצים - V_1, ערך פו' המטרה עם האילוצים - V_2. $V_1 \neq V_2$.

אם $V_2 < V_1$ משמעות הדבר שבהוספת אילוצים הצלחנו לקבל ערך נמוך יותר ולכן טוב יותר בפונקציית המטרה מאשר פונקציית המטרה ללא האילוצים האלו וזה לא ייתכן. לכן $V_1 \leq V_2$.

אם $V_1 < V_2$ כלומר הוספת האילוצים $\xi_i \geq 0$ הניבה ערך שגדול יותר אז בהכרח בפיתרון האופטימלי שערכו V_1 היה קיים משתנה $\xi_j < 0$ ולכן האילוצים השפיעו את פונקציית המטרה. אך נשים לב:

$$V_1 = \min_{w\in\mathbb{R}^d, b\in\mathbb{R}, \xi\in\mathbb{R}^n} \frac{1}{2}\|w\|^2 + \frac{C}{2}\sum_{i=1}^{n}\xi_i^2 = \underbrace{\min_{w\in\mathbb{R}^d, b\in\mathbb{R}, \xi\in\mathbb{R}^n} \frac{1}{2}\|w\|^2 + \frac{C}{2}\cdot\xi_j^2 + \frac{C}{2}\sum_{j\neq i=1}^{n}\xi_i^2}_{(*)}$$

אבל אם נציב $\xi_j = 0$ ב(*) במקום $\xi_j < 0$ נקבל ערך קטן יותר שכן נשים לב ש ξ_j מופיע בריבוע. לכן נעדיף לקחת $\xi_j = 0$.

נראה ש $\xi_j = 0$ עדיין מקיים את האילוץ $y_j(w^T x_j + b) \geq 1 - \xi_j$

ברור שאם עבור $\xi_j>0$ כלשהו מתקיים

$y_j(w^T x_j+b)\ge 1-\xi_j$ אז עבור $\xi_j=0$ האילוץ ימשיך להתקיים

שכן רק נקטין את הביטוי באגף הימני.

הראנו שעבור $\xi_j=0$ יישמרו האילוצים ופו' המטרה תקטן ולכן

בפיתרון האופטימלי יתקיים $\xi_j=0$ בסתירה להנחה ש $\xi_j>0$ ולכן

$V_1\ge V_2$. בסה"כ הראנו $V_2\ge V_1\ge V_2$ ולכן בהכרח $V_1=V_2$,

כלומר הוספת אילוצי אי שליליות לא תשנה את ערך פונקציית המטרה.

(b

$$\min_{w\in\mathbb{R}^d,\, b\in\mathbb{R},\, \varepsilon\in\mathbb{R}^n} \frac{1}{2}\|w\|^2 + \frac{C}{2}\sum_{i=1}^{n} \varepsilon_i^2$$

$$\text{s.t. } y_i(w^T x_i + b) \geq 1 - \varepsilon_i \quad \forall i=1,\dots,n$$

נעביר אגף ונכפיל ב-1- ונכתוב את בעיית האופטימיזציה כך:

$$\min_{w\in\mathbb{R}^d,\, b\in\mathbb{R},\, \varepsilon\in\mathbb{R}^n} \frac{1}{2}\|w\|^2 + \frac{C}{2}\sum_{i=1}^{n} \varepsilon_i^2$$

$$\text{s.t. } -y_i(w^T x_i + b) + 1 - \varepsilon_i \leq 0 \quad \forall i=1,\dots,n$$

הלגראנז'יאן:

$$\mathcal{L}(w,\varepsilon,\alpha,b) =$$

$$= \frac{1}{2}\|w\|^2 + \frac{C}{2}\cdot\sum_{i=1}^{n} \varepsilon_i^2 + \sum_{i=1}^{n} \alpha_i\left(-y_i(w^T x_i + b) + 1 - \varepsilon_i\right)$$

$$= \frac{1}{2}\|w\|^2 + \frac{C}{2}\cdot\|\varepsilon\|^2 + \sum_{i=1}^{n} -\alpha_i\varepsilon_i + \alpha_i\left(-y_i(w^T x_i + b) + 1\right)$$

$$= \frac{1}{2}\|w\|^2 + \frac{C}{2}\cdot\|\varepsilon\|^2 - \alpha\cdot\varepsilon + \sum_{i=1}^{n} \alpha_i\left(-y_i(w^T x_i + b) + 1\right)$$

c) נחשב נגזרות חלקיות של הלגרנג'יאן ונשווה ל-0

$$\nabla_w \mathcal{L} = w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0 \iff w = \sum_{i=1}^{n} \alpha_i y_i x_i$$

$$\nabla_{\xi} \mathcal{L} = \frac{2C}{2}\xi - \alpha = C\xi - \alpha \iff C\xi = \alpha \iff \xi = \frac{\alpha}{C}$$

$$\nabla_b \mathcal{L} = -\sum_{i=1}^{n} \alpha_i y_i = 0$$

נציב בלגרנג'יאן:

$$\frac{1}{2}\Big(\sum_{i=1}^{n} \alpha_i y_i x_i\Big)^2 + \frac{C}{2}\Big\|\frac{\alpha}{C}\Big\|^2 - \alpha \cdot \frac{\alpha}{C} + \sum_{i=1}^{n} \alpha_i\Big[-y_i\Big(\Big(\sum \alpha_i y_i x_i\Big)^T x_i + b\Big) + 1\Big]$$

$$= \frac{1}{2}\Big(\sum_{i=1}^{n} \alpha_i y_i x_i\Big)^2 + \frac{1}{2C}\|\alpha\|^2 - \frac{\|\alpha\|^2}{C} + \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \alpha_i y_i \Big(\sum_{i=1}^{n} \alpha_i y_i x_i\Big)^T x_i - b\sum_{i=1}^{n} \alpha_i y_i$$

נזכיר $-\sum_{i=1}^{n} \alpha_i y_i = 0$ ומתקבל:

$$\frac{1}{2}\Big(\sum_{i=1}^{n} \alpha_i y_i x_i\Big)^2 - \frac{1}{2C} \cdot \|\alpha\|^2 + \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \alpha_i y_i x_i \Big(\sum_{i=1}^{n} \alpha_i y_i x_i\Big)$$

$$= -\frac{1}{2} \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \cdot x_i \cdot x_j + \sum_{i=1}^{n} \alpha_i - \frac{1}{2C} \cdot \|\alpha\|^2$$

P) Nicecalc:

$$\max_{\alpha} \min_{w,\xi,b} \mathcal{L}(w,\xi,\alpha,b) = \max_{\alpha} g(\alpha) = \min_{\alpha} -g(\alpha)$$

Licznie ciekawie:

$$\min_{\alpha} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \cdot x_i \cdot x_j - \sum_{i=1}^{n}\alpha_i + \frac{1}{2C}\cdot\|\alpha\|^2$$

$$-\sum_{i=1}^{n}\alpha_i y_i = 0$$

4. **(15 points) Gradient of cross-entropy loss over softmax**. Let $\mathbf{y} \in \{0,1\}^d$ be a one-hot vector, i.e. $\mathbf{y}$ has a single entry which is 1 and the rest are 0. Consider the following loss function $\ell_{\mathbf{y}} : \mathbb{R}^d \to \mathbb{R}$:

$$\ell_{\mathbf{y}}(\mathbf{w}) = -\mathbf{y} \cdot \log\left(\text{softmax}(\mathbf{w})\right),$$

where $\text{softmax}(\mathbf{w}) = \frac{e^{\mathbf{w}}}{\sum_{j=1}^{d} e^{w_j}}$, is the softmax function (see slide 7 of lecture #7). In the above notation, the exponent and logarithm function operate elementwise when given a vector as an input. The function $\ell_{\mathbf{y}}$ is known as the *cross-entropy loss*, and you will encounter it in the programming assignment.

Prove that the gradient of $\ell_{\mathbf{y}}$ with respect to $\mathbf{w}$ is given by:

$$\nabla \ell_{\mathbf{y}}(\mathbf{w}) = \text{softmax}(\mathbf{w}) - \mathbf{y}$$

$$\forall 1 \le i \le d \quad \frac{\partial \ell_y(w)}{\partial w_i} = -y_i \cdot \frac{1}{\frac{e^{w_i}}{\sum_{j=1}^{d} e^{w_j}} \cdot \ln e} \cdot \frac{e^{w_i} \cdot \sum_{j=1}^{d} e^{w_j} - e^{w_i} \cdot e^{w_i}}{\left(\sum_{j=1}^{d} e^{w_j}\right)^2}$$

$$= -y_i \cdot \frac{\sum_{j=1}^{d} e^{w_j}}{e^{w_i}} \cdot \frac{e^{w_i}\left(\sum_{j=1}^{d} e^{w_j} - e^{w_i}\right)}{\left(\sum_{j=1}^{d} e^{w_j}\right)^2}$$

$$= -y_i \cdot \frac{\sum_{j=1}^{d} e^{w_j} - e^{w_i}}{\sum_{j=1}^{d} e^{w_j}} = -y_i\left(1 - \frac{e^{w_i}}{\sum_{j=1}^{d} e^{w_j}}\right)$$

$$= -y_i + y_i \cdot \frac{e^{w_i}}{\sum_{j=1}^{d} e^{w_j}}$$

$$y_i \frac{e^{w_i}}{\sum_{j=1}^{d} e^{w_j}} = \begin{cases} \frac{e^{w_i}}{\sum_{j=1}^{d} e^{w_j}} & i=c \\ 0 & i \neq c \end{cases}$$

הנגזרת:

$$\nabla \ell_{\log}(w) = \left(\frac{\partial \ell_{\log}}{\partial w_1}, \cdots, \frac{\partial \ell_{\log}}{\partial w_d} \right)$$

$$= \left(\frac{e^{w_1}}{\sum_{j=1}^{d} e^{w_j}} - y_1, \cdots, \frac{e^{w_d}}{\sum_{j=1}^{d} e^{w_j}} - y_i \right) = \text{softmax}(w) - y$$

כאשר

חלק תכנותי

שאלה 1a - בקוד.

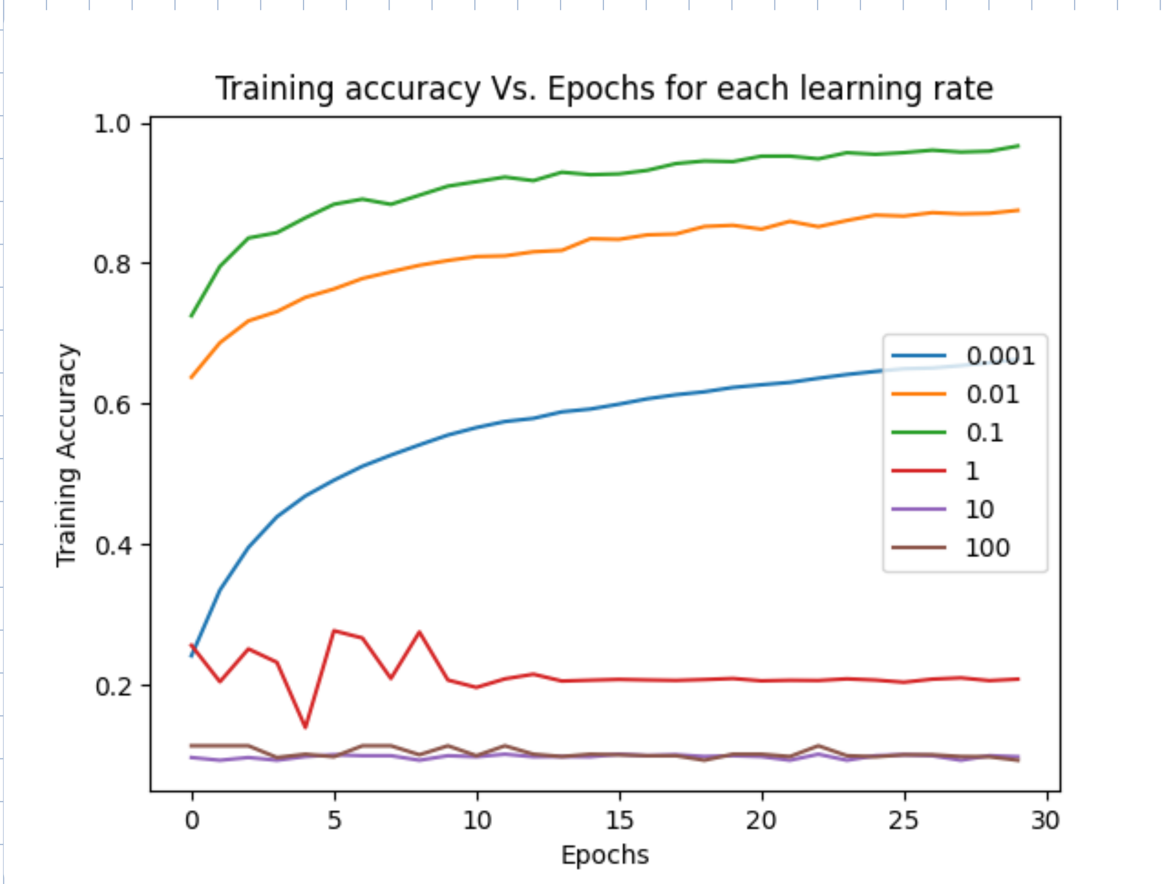
שאלה 1b:

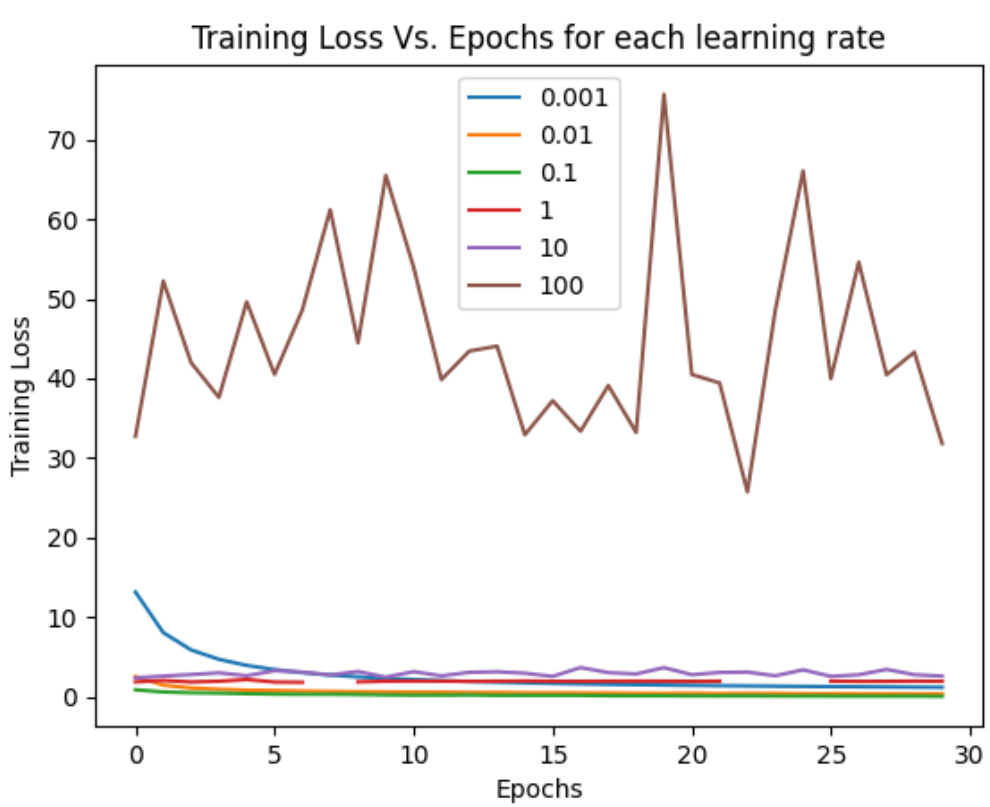
(b) **(10 points)** Train a one-hidden layer neural network as in the example given above (e.g., training set of size 10000, one hidden layer of size 40). For each learning rate in $\{0.001, 0.01, 0.1, 1, 10, 100\}$, plot the *training* accuracy, *training* loss ($\ell(\mathcal{W})$) and *test* accuracy across epochs (3 plots: each contains the curves for all learning rates). For the test accuracy you can use the *one_label_accuracy* function, for the training accuracy use the *one_hot_accuracy* function and for the training loss you can use the *loss* function. All functions are in the *Network* class.

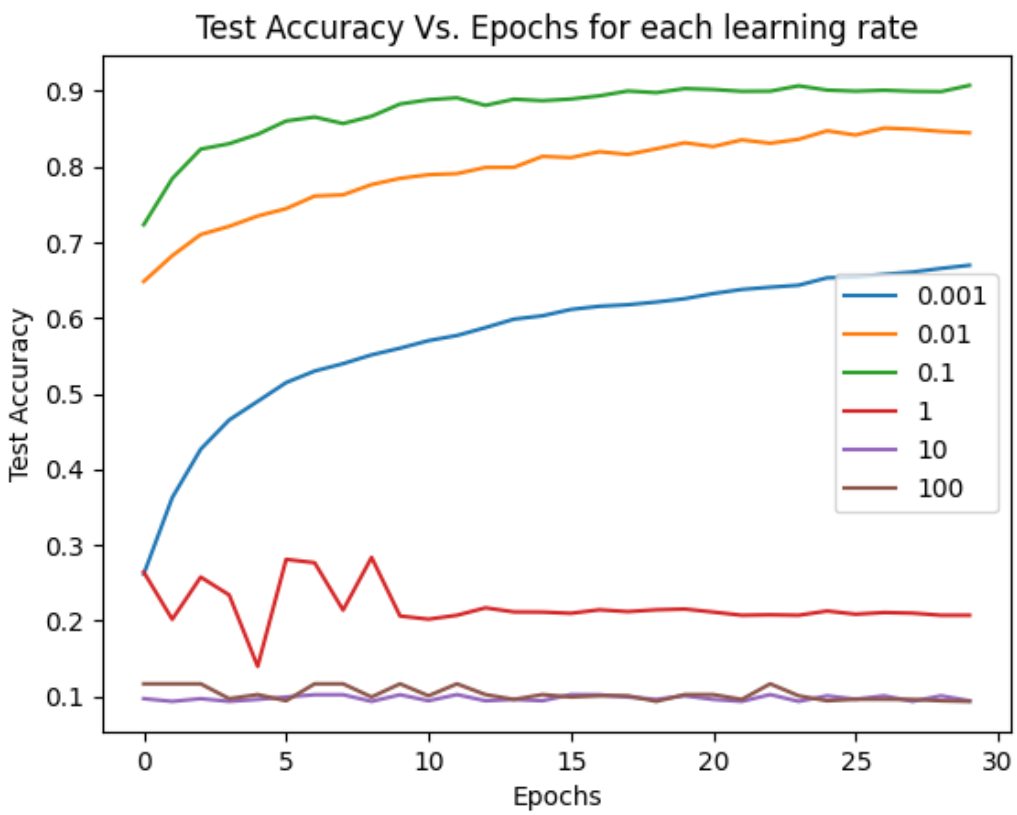
The test accuracy with leaning rate 0.1 in the final epoch should be above 80%.

What happens when the learning rate is too small or too large? Explain the phenomenon.

הגרפים:




Training Loss Vs. Epochs for each learning rate
0.001
0.01
0.1
1
10
100
Training Loss
Epochs
0
10
20
30
40
50
60
70
0
5
10
15
20
25
30


Test Accuracy Vs. Epochs for each learning rate
0.001
0.01
0.1
1
10
100
Test Accuracy
Epochs
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
0
5
10
15
20
25
30

ניתן לראות שה test accuracy וה training accuracy
הטובים ביותר נקבל עבור learning rate=0.1 (הגרפים הירוקים).
עבור learning rate נמוך מדי הצעדים ב SGD יהיו קטנים
מדי ולכן יידרשו יותר צעדים על מנת להגיע לדיוק אופטימלי.
(גרפים כתום וכחול). ניתן לראות למשל עבור learning rate=0.001
שיש צורך במספר צעדים גבוה יותר על מנת למזער את פונקציית ההפסד
לעומת learning rate גבוה יותר.
עבור learning rate גבוה מדי גודל הצעד ב SGD גדול מדי
ויגרום לכך שנפספס את נקודת המינימום, כלומר נתרחק מדי מהנקודה
בעלת ההפסד המיטבי. (גרפים אדום, סגול וחום).
ניתן לראות למשל עבור learning rate=100 פונקציית ההפסד
אינה יציבה ויש הרבה עליות וירידות (כמו בחיים :)).

(c) **(5 points)** Now train the network on the whole training set and test on the whole test set:

```
>>> training_data, test_data = data.load(train_size=50000,test_size=10000)
>>> net = network.Network([784, 40, 10])
>>> net.SGD(training_data, epochs=30, mini_batch_size=10, learning_rate=0.1,
test_data=test_data)
```

Do **not** calculate the training accuracy and training loss as in the previous section (this is time consuming). What is the test accuracy in the final epoch (should be above 90%)?

הדיוק בסוף האימון הוא 94.23% בערך

```
Epoch 29 test accuracy: 0.9423
```

הערה: ה-epoch האחרון הוא 29 ולא 30 כי מתחילים מ-0.

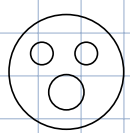
(d) **(10 points bonus)** Explore different structures and parameters and try to improve the test accuracy. Use the whole test set. In order to get full credit you should get accuracy of more than 96%. Any improvement over the previous clauses will give you 5 points. The maximum score for this homework set remains 100.

הקוד:

```
# Q1d - bonus
training_data, test_data = backprop_data.load(train_size=50000, test_size=10000)
net = backprop_network.Network([784, 800, 10])
net.SGD(training_data, 30, 10, 0.1, test_data, 1)
```

הדיוק שהתקבל:

```
Epoch 28 test accuracy: 0.9617
Epoch 29 test accuracy: 0.9617
```



הגענו ל-96.17% בערך!

שאלה 2

(a) **(5 points)** Train the network **on 10% of the training set** (otherwise the training process would take a long time) once with the default parameters given in the code, and then again using the Adam optimizer instead of SGD (see the documentation in `torch.optim.Adam` for more information), with a learning rate of 0.00005. How is the training process affected by this change? Do you achieve better training loss? What about the validation accuracy?

:SGD הרצה

```
Epoch [1/10], Step [71/71], Loss: 2.1925
Accuracy of the network on the 5000 validation images: 25.6 %
Epoch [2/10], Step [71/71], Loss: 1.8714
Accuracy of the network on the 5000 validation images: 38.0 %
Epoch [3/10], Step [71/71], Loss: 1.6779
Accuracy of the network on the 5000 validation images: 34.4 %
Epoch [4/10], Step [71/71], Loss: 1.5494
Accuracy of the network on the 5000 validation images: 39.0 %
Epoch [5/10], Step [71/71], Loss: 1.4755
Accuracy of the network on the 5000 validation images: 43.4 %
Epoch [6/10], Step [71/71], Loss: 1.4065
Accuracy of the network on the 5000 validation images: 47.2 %
Epoch [7/10], Step [71/71], Loss: 1.3365
Accuracy of the network on the 5000 validation images: 43.4 %
Epoch [8/10], Step [71/71], Loss: 1.2709
Accuracy of the network on the 5000 validation images: 48.6 %
Epoch [9/10], Step [71/71], Loss: 1.2077
Accuracy of the network on the 5000 validation images: 47.0 %
Epoch [10/10], Step [71/71], Loss: 1.1507
Accuracy of the network on the 5000 validation images: 50.8 %
```

:Adam הרצה

```
Epoch [1/10], Step [71/71], Loss: 1.8775
Accuracy of the network on the 5000 validation images: 38.6 %
Epoch [2/10], Step [71/71], Loss: 1.5067
Accuracy of the network on the 5000 validation images: 48.6 %
Epoch [3/10], Step [71/71], Loss: 1.3428
Accuracy of the network on the 5000 validation images: 49.8 %
Epoch [4/10], Step [71/71], Loss: 1.2231
Accuracy of the network on the 5000 validation images: 53.2 %
Epoch [5/10], Step [71/71], Loss: 1.0836
Accuracy of the network on the 5000 validation images: 59.6 %
Epoch [6/10], Step [71/71], Loss: 0.9696
Accuracy of the network on the 5000 validation images: 58.8 %
Epoch [7/10], Step [71/71], Loss: 0.8907
Accuracy of the network on the 5000 validation images: 63.6 %
Epoch [8/10], Step [71/71], Loss: 0.7791
Accuracy of the network on the 5000 validation images: 66.6 %
Epoch [9/10], Step [71/71], Loss: 0.7348
Accuracy of the network on the 5000 validation images: 62.4 %
Epoch [10/10], Step [71/71], Loss: 0.6338
Accuracy of the network on the 5000 validation images: 63.2 %
```

נשים לב שב Adam התקבל loss נמוך יותר
נשים לב שב SGD הגענו בepoch האחרון לאחוז דיוק
של 50.8% לעומת 63.2 ב Adam

(b) **(5 points)** Again using Adam, train the network with a learning rate of 0.005 instead of the default 0.00005. How is the training process affected?

```
Epoch [1/10], Step [71/71], Loss: 22.3386
Accuracy of the network on the 5000 validation images: 13.0 %
Epoch [2/10], Step [71/71], Loss: 2.3835
Accuracy of the network on the 5000 validation images: 13.8 %
Epoch [3/10], Step [71/71], Loss: 2.3339
Accuracy of the network on the 5000 validation images: 16.6 %
Epoch [4/10], Step [71/71], Loss: 2.3180
Accuracy of the network on the 5000 validation images: 20.8 %
Epoch [5/10], Step [71/71], Loss: 2.2471
Accuracy of the network on the 5000 validation images: 20.2 %
Epoch [6/10], Step [71/71], Loss: 2.1806
Accuracy of the network on the 5000 validation images: 21.8 %
Epoch [7/10], Step [71/71], Loss: 2.1958
Accuracy of the network on the 5000 validation images: 14.4 %
Epoch [8/10], Step [71/71], Loss: 2.1598
Accuracy of the network on the 5000 validation images: 18.8 %
Epoch [9/10], Step [71/71], Loss: 2.1917
Accuracy of the network on the 5000 validation images: 16.2 %
Epoch [10/10], Step [71/71], Loss: 2.1279
Accuracy of the network on the 5000 validation images: 20.2 %
```

נשים לב כי הloss גבוה יותר וה validation accuracy
נמוך יותר ולכן נסיק שערך זה אינו אפקטיבי לאימון
הרשת.

(c) **(5 points)** Train the network using Adam and a step size of 0.00005, but change the architecture so that it doesn't include Batch Normalization or Dropout layers (simply comment out the relevant lines in the network's definition). How does the training loss evolve when compared to the one with BatchNorm + Dropout? Do these additions seem to accelerate the learning process?

```
Epoch [1/10], Step [71/71], Loss: 2.0798
Accuracy of the network on the 5000 validation images: 30.8 %
Epoch [2/10], Step [71/71], Loss: 1.7767
Accuracy of the network on the 5000 validation images: 38.8 %
Epoch [3/10], Step [71/71], Loss: 1.6116
Accuracy of the network on the 5000 validation images: 42.2 %
Epoch [4/10], Step [71/71], Loss: 1.5089
Accuracy of the network on the 5000 validation images: 49.2 %
Epoch [5/10], Step [71/71], Loss: 1.4111
Accuracy of the network on the 5000 validation images: 51.6 %
Epoch [6/10], Step [71/71], Loss: 1.3259
Accuracy of the network on the 5000 validation images: 49.0 %
Epoch [7/10], Step [71/71], Loss: 1.2276
Accuracy of the network on the 5000 validation images: 53.4 %
Epoch [8/10], Step [71/71], Loss: 1.1299
Accuracy of the network on the 5000 validation images: 54.8 %
Epoch [9/10], Step [71/71], Loss: 1.0410
Accuracy of the network on the 5000 validation images: 58.6 %
Epoch [10/10], Step [71/71], Loss: 0.9175
Accuracy of the network on the 5000 validation images: 54.8 %
```

נשים לב שה loss גדל (בכל epoch) לעומת סעיף a וה validation
accuracy קטן ולכן נסיק שיש חשיבות ל BatchNorm + Dropout?

הערה לגבי השאלה: בכל סעיף ביצעתי את השינויים הרלוונטיים
בקוד - לא שכפלתי את הקוד כמה פעמים ולכן התוצאות מופיעות
ב pdf ☺