מגישים:
עומר ג'אע - 318256120
יותם פלדמן - 315717181

Introduction to machine learning
Exercise 2

Winter 2024

Submission guidelines, **read and follow carefully**:

- The exercise **must** be submitted in pairs.
- Submit via Moodle.
- The submission should include two separate files:
  1. A pdf file that includes your answers to all the questions.
  2. The code files for the python question. You must submit a copy of the shell python file provided for this exercise in Moodle, with the required functions implemented by you. **Do not change the name of this file!** In addition, you can also submit other code files that are used by the shell file.
- Your python code should follow the course python guidelines. See the Moodle website for guidelines and python resources.
- Before you submit, **make sure that your code works in the course environment**, as explained in the guidelines. Specifically, **make sure that the test `simple_test` provided in the shell file works**.
- You may only use python modules that are explicitly allowed in the exercise or in the guidelines. If you are wondering whether you can use another module, ask a question in the exercise forum. No module containing machine learning algorithms will be allowed.
- For questions, use the exercise forum, or if they are not of public interest, send them via the course requests system.
- Grading: Q.1 (python code): 10 points, Q.2: 20 points, Q.3: 15 points, Q.4: 19 points, Q.5: 18 points, Q.6: 18 points

Question 1. Implement the soft-SVM algorithm that we learned in class in python. The shell file "softsvm.py" is provided for this exercise in Moodle. It contains an empty implementation of the function required below. You should implement it and submit according to the submission instructions.

```
def softsvm(l, trainX, trainy)
```

The input parameters are:

- l - the parameter λ of the soft SVM algorithm.
- trainX - a 2-D matrix of size $m \times d$, where m is the sample size and d is the dimension of the examples. Row i in this matrix is a vector with d coordinates that describes an example x_i from the training sample.
- trainy - a column vector of length m. The i's number in this vector is the label $y_i \in \{-1, 1\}$ from the training sample.

The function returns the linear predictor w which is a column vector in $\mathbb{R}^d$.

- You may assume all the input parameters are legal.
- We will use the library cvxopt for our Quadratic Program solver.

Instructions for using cvxopt:

- First, you will need to define the matrices H, u, A, and v which correspond to the vectors and matrices with the same names in the quadratic programming problem you learned in class. Those matrices should be cvxopt matrices, check how to create cvxopt matrices or convert numpy arrays to cvxopt matrices here: http://cvxopt.org/userguide/matrices.html
- In order to conserve memory, use sparse matrices when possible.
- Run sol = cvxopt.solvers.qp(H, u, -A, -v) to solve the quadratic programming problem. Here, we pass A and v with a minus sign, since this solver assumes the constraints are $Az \leq v$, while in class we assumed they were $Az \geq v$. The solution of the quadratic program is provided in sol["x"].
- See the note at the end of the exercise regarding a possible error and how to solve it.

Question 2. In this question, you will run your soft SVM implementation on data from the MNIST dataset you saw in exercise 1. For this task, we took a subset of this dataset which include and digits 3 and 7, and the goal of the predictor is to distinguish between the two digits. You can load the dataset, which is already divided to train and test, from the file EX2q2_mnist.npz on the course website.

Run two experiments on this data set. In the first experiment, use a sample size of 100. To generate this small sample, draw it randomly from the provided training sample. Repeat the "small sample" experiment 10 times, and when you report the results, average over these 10 experiments, and plot also error bars which show the maximum and minimum values you got over all experiments. Run your soft-SVM implementation with each of the following values of λ: $\lambda = 10^n$, for $n \in \{1, \ldots, 10\}$.

In the second experiment, use a sample size of 1000, which you should also draw randomly from the training set. Run your soft-SVM implementation with each of the following values of λ: $\lambda = 10^n$, for $n \in \{1, 3, 5, 8\}$. To make the running time feasible, you should run this experiment only once for each value of λ.

(a) Submit a plot of the training error and test error of the small sample size results as a function of λ (plot λ on a logarithmic scale), with one line for the train error and another line for the test error. Each line should show an average of the 10 experiments, and error bars which show the maximum and minimum values you got over all experiments.

(b) Add to the plot the points describing the training error and test error of the large sample size. For this part, don't draw lines between the points in this case, only show each point individually, since you tested values of λ which are quite far away from each other.

(c) Based on what we learned in class, what would you expect the results to look like? Do the results you got match your expectations? In your answer address the following issues:

- Which sample size should get a smaller training error? What about test error? Do the results match your expectations?
- What should be the trend in the *training error* as a function of λ (decreasing/increasing/other)? Why? Do the results (for the small sample size) match your expectations?
- What should be the trend in the *test error* as a function of λ (decreasing/increasing/other)? Why? Do the results (for the small sample size) match your expectations?

Question 3. Let $x \in \mathbb{R}$, $\mathcal{Y} \in \{0, 1\}$. Given a text string c which describes a mathematical condition, define $h_c : x \to \mathcal{Y}$ as follows: $h_c(x) := \mathbb{I}[x \text{ satisfies } c]$. For instance, if c is the condition "x is a natural number", then $h_c(x) = \mathbb{I}[x \in \mathbb{N}]$. For any $n \in \mathbb{N}$, define the hypothesis class $\mathcal{H}_n \subseteq \mathcal{Y}^x$ as follows:

$$\mathcal{H}_n := \{h_c \mid c \text{ is a condition which can be described using at most } n \text{ characters}\}$$

For instance, the condition c given above is in $\mathcal{H}_n$ for all $n \geq 21$. The text of the conditions can include only ASCII characters (there are 128 ASCII characters).

Suppose that the examples x are temperature values, and the labels y indicate whether a doctor gives medicine to a person with this temperature. Suppose that we know that the rule that the doctors use to decide whether to give the medicine has at most 10 characters and that the doctors follow this rule exactly. We get a random independent sample from the distribution D over $x \times \mathcal{Y}$ over temperatures and doctor decisions. We would like to run an ERM algorithm with the hypothesis class $\mathcal{H}_{10}$ to find a condition with an error of at most 0.1. We have unlimited computational resources.

(a) Explain why running the ERM algorithm with the hypothesis class $\mathcal{H}_n$ for $n < 10$ or for $n > 10$ may be a worse idea than running it with $\mathcal{H}_{10}$.

(b) Use PAC bounds to calculate an upper bound on the size of a sample size that would be sufficient to find, with a probability at least 0.99, a condition with an error at most 0.1 on the distribution, using an ERM algorithm $\mathcal{H}_{10}$. Give a formal description of the guarantee that you are providing.

(c) Suppose we run ERM with $\mathcal{H}_n$ on the training sample for some $n < 10$. Calculate an upper bound on the size of a sample size that would be sufficient to find, with a probability at least 0.99, a condition with an excess error at most 0.1 on the distribution, using an ERM algorithm with $\mathcal{H}_n$. Give a formal description of the guarantee that you are providing.

Question 4. Consider the hypothesis class of homogeneous linear predictors $\mathcal{H}_L^d := \{h_w \mid w \in \mathbb{R}^d\}$ where $h_w(x) = \text{sign}(\langle w, x \rangle)$, $\forall x \in \mathbb{R}^d$.
Prove that any linearly independent set of d vectors, $u_1, \ldots, u_d \in \mathbb{R}^d$, are shattered by $\mathcal{H}_L^d$.

Reminder: A set of input vectors is said to be shattered by an hypothesis class $\mathcal{H}$ if it can get all the possible labelings by predictors from $\mathcal{H}$.

Question 5. For a given training sample $S = ((x_1, y_1), \ldots, (x_m, y_m))$, consider the following **modified version** of the soft-SVM optimization problem:

$$\lambda \|\mathbf{w}\|_1^2 + \sum_{i=1}^{m} \ell_h(w, (x_i, y_i)),$$

where $\ell_h(w,(x_i,y_i)) = \max\{0, 1 - y_i\langle w, x_i\rangle\}$ is the *hinge loss* defined in the lecture. Recall that $\|\mathbf{w}\|_1$ is the ℓ_1 norm defined as $\|\mathbf{w}\|_1 := \sum_{i=1}^{d} |w(i)|$.

Express this optimization problem as a quadratic program in standard form, as we showed in class.

(a) Write a quadratic minimization problem with constraints that is equivalent to the problem above, using auxiliary variables similar to the ξ_i in the soft-SVM implementation.

(b) Write what H,u,A,v in the definition of a Quadratic Program should be set to so as to solve the minimization problem you wrote.

Question 6. In this exercise, we will see that without margin assumptions, the Perceptron algorithm might run for a long time, exponential in the dimension d. We define a special sample $S = ((x_1, y_1), \ldots, (x_m, y_m))$, where $m = d$, $x_i \in \mathbb{R}^d$, and $y_i \in \{-1, 1\}$, where $y_i := (-1)^{i+1}$, and $x_i \in \mathbb{R}^d$ is defined by:

$$x_i(j) = \begin{cases} (-1)^i, & j < i \\ (-1)^{i+1}, & j = i \\ 0, & j > i. \end{cases}$$

In the following steps, you will prove that for the sample above, the Perceptron performs a number of updates at least exponential in d:

(a) Prove that in any iteration t of the Perceptron on the given sample S, and for any $i \leq d$, $|w^{(t+1)}(i)| \leq t$.
Hint: use induction on the Perceptron update $w^{(t+1)} \leftarrow w^{(t)} + y_i x_i$.

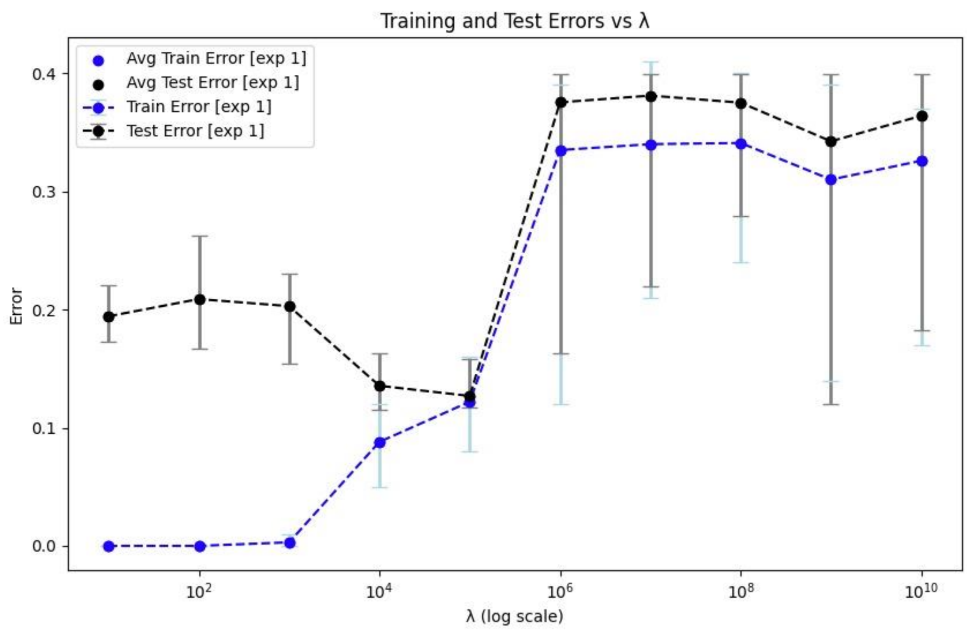
(b) Let $w^{(T)}$ be the separator that the Perceptron outputs. Prove that for every coordinate i, $|w^{(T)}(i)| \geq 2^{i-1}$. Hint: prove this by induction on i, using the fact that the separator defined by $w^{(T)}$ labels correctly all the examples in the given sample S.

(c) Conclude from the two previous items that the number of updates of the Perceptron until it stops and outputs $w^{(T)}$ is exponential in d.

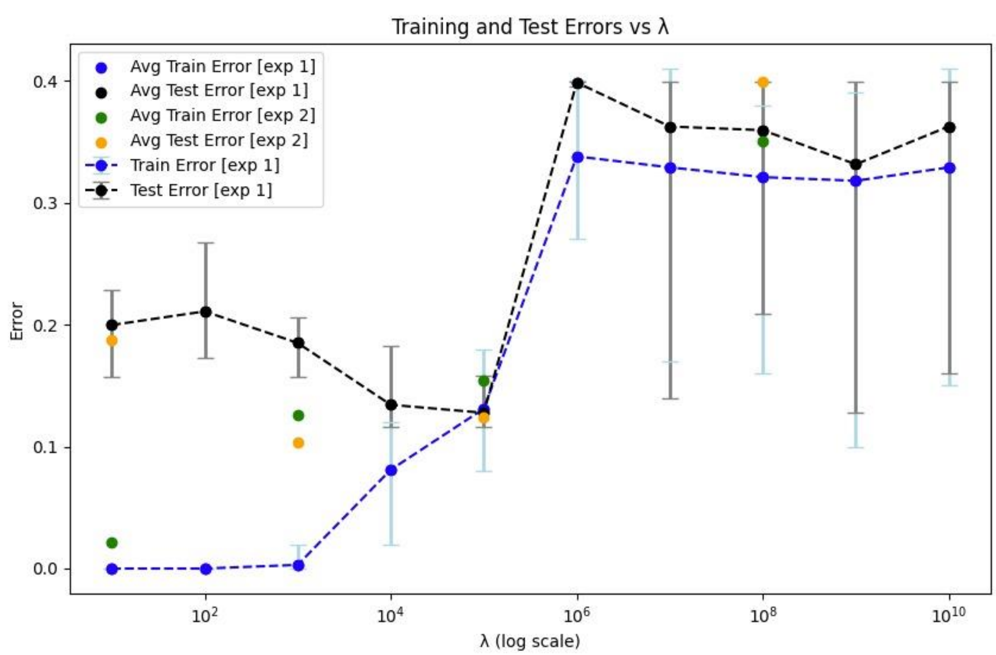
To help with intuition, you may want to implement the Perceptron algorithm defined in class (no need to submit the code), and check its behavior on the sample S for $d = 1, 2, \ldots, 10$.

שאלה 2

א.


Training and Test Errors vs λ
Avg Train Error [exp 1]
Avg Test Error [exp 1]
Train Error [exp 1]
Test Error [exp 1]
Error
λ (log scale)

ב.


Training and Test Errors vs λ
Avg Train Error [exp 1]
Avg Test Error [exp 1]
Avg Train Error [exp 2]
Avg Test Error [exp 2]
Train Error [exp 1]
Test Error [exp 1]
Error
λ (log scale)

ג. בהתבסס על מה החומר שנלמד, מה אנחנו מצפים שהתוצאות יהיו והאם התוצאות תואמות את הציפיות שלנו?
נחלק את התשובה שלנו ל3 חלקים ובכל תשובה נרחיב על הציפיות והתוצאות.

1. איזה גודל מדגם אמור לקבל שגיאת אימון קטנה יותר? מה לגבי שגיאת על הtest?
2. מה צריכה להיות המגמה בשגיאת האימון כפונקציה של λ? למה?
   האם התוצאות עבור הניסוי הראשון תואמות את הציפיות שלך?
3. מה צריכה להיות המגמה בשגיאת הtest כפונקציה של λ? למה?
   האם התוצאות עבור הניסוי הראשון תואמות את הציפיות שלך?

פתרון:

1. ציפיות:
שגיאת האימון – בניסוי הראשון גודל המדגם היה בגודל 100 ובניסוי השני גודל המדגם היה בגודל 1000.
הציפיות שלנו הן ששגיאת האימון (בממוצע) תהייה קטנה יותר בניסוי הראשון מאשר בניסוי השני מאחר וקל יותר למצוא מפריד עבור מספר קטן יותר של דוגמאות (פחות דוגמאות לטעות עליהן), כלומר הסיכוי לטעות על מדגם האימון קטן כאשר גודל המדגם קטן.
שגיאת הtest – הציפיות שלנו במקרה זה יהיו הפוכות. מאחר שהתאמנו על יותר דוגמאות בניסוי השני, אנו מצפים שהשגיאה על הtest בניסוי השני תהייה קטנה יותר מאשר השגיאה על הtest בניסוי הראשון.
לפי הנוסחה שלמדנו בכיתה ניתן לראות שככל שכמות הדוגמאות גדלה השגיאה קטנה:

$$\mathbb{E}_{S\sim\mathcal{D}^m}[L_h(\hat{w}_S,\mathcal{D})] \leq \min_{u\in\mathbb{R}^d}\left(L_h(u,\mathcal{D})+\lambda\|u\|_2^2\right)+\frac{2R_{\mathcal{D}}^2}{\lambda m}.$$

תוצאות:
שגיאת האימון – כמו שצפינו, ניתן לראות בגרף שאכן שגיאת האימון בניסוי הראשון קטנה יותר מאשר בניסוי השני.
שגיאת הtest –כמו שצפינו, התוצאות במקרה זה הפוכות לשגיאת האימון.
ניתן לראות בגרף שאכן שגיאת האימון בניסוי הראשון גדולה יותר מאשר בניסוי השני.

2. ציפיות: למדנו בכיתה כי שככל ש-λ גדלה כך השגיאה על האימון תגדל גם.
הסיבה לכך היא כשאנחנו מגדילים את λ אנחנו נותנים משקל גדול יותר למזעור הנורמה של u, כלומר נבחר לבסוף u בעל נורמה קטנה יותר וכתוצאה מכך ה soft margin יגדל.
כאשר ה soft margin גדול יותר אנחנו פחות רגישים ל sample הספציפי ולכן השגיאה על האימון תגדל.

תוצאות: כמו שצפינו, ניתן לראות בגרף שכאשר λ גדלה גם שגיאת האימון גדלה.

3. ציפיות: למדנו בכיתה שישנו טרייד אוף בבחירת ה-λ :
כאשר נבחר λ גדולה – נעניש יותר עבור נורמה גדולה של u (הגורמת לsoft-margin קטן יותר), דבר העלול לגרום לאובר פיטינג.
כאשר נבחר λ קטנה – נעניש יותר עבור גודל מדגם קטן דבר שיגרום למרחב האפשרויות של האלגוריתם לרגישות לשגיאה שנוצרת ממדגם קטן. כתוצאה מכך לא ימצא מפריד לינארי בצורה טובה.
לכן אנחנו מצפים למגמת ירידה בשגיאה על ה-test עד שנגיע ל- λ האופטימלית ולאחר מכן השגיאה תחל לעלות.

תוצאות: כמו שצפינו, ניתן לראות בגרף שהשגיאה על הtest הקטנה ביותר מתקבלת לא ע"י λ גדולה ולא ע"י λ קטנה. בנוסף, ניתן לשים לב שאנו מקבלים שגיאה גדולה יותר ככל שאנו מגדילים את λ וגם שגיאה גדולה יותר כאשר λ קטנה מאוד.

שאלה 3:

(a) Explain why running the ERM algorithm with the hypothesis class $\mathcal{H}_n$ for $n < 10$ or for $n > 10$ may be a worse idea than running it with $\mathcal{H}_{10}$.

(a) הרצת אלגוריתם ERM עם סדר $n>10$ או סדר $n<10$ הוא רעיון פחות טוב:

1) סדר $n>10$: מחלקת ההיפותזות שלנו תהיה גדולה יותר ולכן זמן הריצה של ה- ERM עלול לגדול. בנוסף ההסתברות שקיים כלל התפלגות בעל שגיאה אפס על המדגם גדלה (מכיוון שמחלקת ההיפותזות גדלה) ולכן ה- ERM עלול לבחור כלל התפלגות שונה מהרצוי.

2) סדר $n<10$: מחלקת היפותזות כן אינה כוללת את כל ההתפלגות של הרופאים ולכן ERM עלול למצוא כלל התפלגות אחר אשר יתכן ויגדיל את השגיאה ההסתברותית.

(b) Use PAC bounds to calculate an upper bound on the size of a sample size that would be sufficient to find, with a probability at least 0.99, a condition with an error at most 0.1 on the distribution, using an ERM algorithm $\mathcal{H}_{10}$. Give a formal description of the guarantee that you are providing.

לפי נתון השאלה אני מניח שהכלל בו משתמש הרופא כדי להחליט האם לתת את התרופה מכיל לכל היותר 10 תווים והרופא פועל לפיו בדיוק.

לכן נקבע שאנחנו במקרה Realizable $(n=10,\ H_{10})$

נתון:

1. $\varepsilon = 0.1$
2. $1-\delta = 0.99 \Leftarrow$ $\delta = 0.01$
3. ישנם 128 תווי ASCII

עבור תנאי באורך 10 בדיוק יש 128^{10} אפשרויות שונות

עבור תנאי באורך 9 בדיוק יש 128^{9} אפשרויות שונות

$\vdots$

עבור תנאי באורך 1 בדיוק יש 128^{1} אפשרויות שונות

לכן

$$|H_{10}| = \sum_{i=1}^{10} 128^i$$

$$m \geq \frac{\ln(|H|) + \ln\left(\frac{1}{\delta}\right)}{\varepsilon} = 531.333$$

כלומר

$$\boxed{m \geq 532}$$

(c) Suppose we run ERM with $\mathcal{H}_n$ on the training sample for some $n < 10$. Calculate an upper bound on the size of a sample size that would be sufficient to find, with a probability at least 0.99, a condition with an excess error at most 0.1 on the distribution, using an ERM algorithm with $\mathcal{H}_n$. Give a formal description of the guarantee that you are providing.

בגלל ש $n<10$ לא מובטח לי ש D is realizable by H_n

לכן אנחנו במקרה האגנוסטי

$$m \geq \frac{2\log(|H_n|) + 2\log\left(\frac{2}{\delta}\right)}{\varepsilon^2}$$

בגלל ש n אינו קבוע אז $|H_n|$ תלוי ב- n.

$$|H_n| = \sum_{i=1}^{n} 128^i$$

לכן

$$\boxed{m \geq \frac{2\ln\left(\sum_{i=1}^{n} 128^i\right) + 2\ln\left(\frac{2}{0.01}\right)}{0.01^2}}$$

לדוגמה $n=9$:

$$m \geq \frac{2\ln\left(\sum_{i=1}^{9} 128^i\right) + 2\ln\left(\frac{2}{0.01}\right)}{0.01^2} = 9794.88$$

$$m \geq 9795$$

Question 4. Consider the hypothesis class of homogeneous linear predictors $\mathcal{H}_L^d := \{h_w \mid w \in \mathbb{R}^d\}$ where $h_w(x) = \text{sign}(\langle w, x\rangle),\ \forall x \in \mathbb{R}^d$.
Prove that any linearly independent set of d vectors, $u_1, \dots, u_d \in \mathbb{R}^d$, are shattered by $\mathcal{H}_L^d$.

Reminder: A set of input vectors is said to be shattered by an hypothesis class $\mathcal{H}$ if it can get all the possible labelings by predictors from $\mathcal{H}$.

הוכחה:

יהיו $u_1,\dots,u_d \in \mathbb{R}^d$ וקטורים בלתי תלויים.

על מנת להוכיח ש $u_1,\dots,u_d \in \mathbb{R}^d$ הם shattered ע"י מחלקת ההיפותזות

נראה כי לכל וקטור u_i קיימת תווית $y_i \in \{-1,1\}$, (קבוצת כל התוויות האפשריות)

למעשה אנחנו מחפשים וקטור $w \in \mathbb{R}^d$ כך שלכל $1 \le i \le d$ מתקיים:

$$\text{וגם}\begin{cases} \langle w,u_i\rangle > 0 & \text{אז} \quad y_i = 1 \quad \text{אם} \\ \langle w,u_i\rangle < 0 & \text{אז} \quad y_i = -1 \quad \text{אם} \end{cases}$$

ונשים לב כי ישנם d וקטורים בלתי תלויים ולכן הם בסיס של $\mathbb{R}^d$.

כלומר ניתן להציג כל וקטור ב-$\mathbb{R}^d$ ע"י צירוף לינארי שלהם.

לכן למערכת משוואות קיים פתרון (n משוואות עם n נעלמים) כנדרש.

לדוגמה עבור $u_1 = \begin{pmatrix}1\\0\\0\end{pmatrix}$, $u_2 = \begin{pmatrix}0\\1\\0\end{pmatrix}$, $u_3 = \begin{pmatrix}0\\0\\1\end{pmatrix}$

$y_1 = 1$ $\quad y_2 = -1$ $\quad y_3 = 1$

$$\begin{cases} w^T \cdot u_1 > 0 & , y_1 = 1 \\ w^T u_2 < 0 & y_2 = -1 \\ w^T \cdot u_3 > 0 & y_3 = 1 \end{cases}$$

ניתן לבחור $w = \begin{pmatrix}1\\-1\\1\end{pmatrix}$

שאלה 5, סעיף ג':

objective function:

$$\min_{\substack{s\in\mathbb{R}^d \\ \varepsilon\in\mathbb{R}^m}} \lambda\cdot t^2 + \sum_{i=1}^{m}\varepsilon_i$$

S.t:

1) $\|w_1\|_1^2$: $\forall i\in\{1,\dots,d\}$

2: $w_i \le r_i$, 3: $w_i \ge -r_i$ $\left(|w_i| \le r_i\right)$, 5: $t \ge \sum_{j=1}^{d} r_j$

$r_i - w_i \ge 0$ $\quad$ $w_i + r_i \ge 0$ $\quad$ $t - \sum_{j=1}^{d} r_j \ge 0$

2) hing loss: $\forall i\in\{1,\dots,m\}$ 4: $\varepsilon_i \ge 0$ $\quad$ 1: $y_i\langle w, x_i\rangle \ge 1-\varepsilon_i$

$y_i\langle w, x_i\rangle + \varepsilon_i \ge 1$

הסבר על 1:

$|w_i| \le r_i$ פותר את ענין l_1 norm

$t \ge \sum_{i=1}^{d} r_i$ פותר את ענין $\Sigma()$

שאלה 5, סעיף ב':

$$H = 2\lambda \begin{pmatrix} 1_{1\times1} & 0 \\ 0 & 0 \end{pmatrix}_{(1+2d+m)\times(1+2d+m)} \qquad Z = \begin{pmatrix} t_{1\times1} \\ w_{d\times1} \\ r_{d\times1} \\ \varepsilon_{m\times1} \end{pmatrix}_{(1+2d+m)\times 1}$$

$$u = \begin{pmatrix} 0_{(1+2d+m)} \\ 1_{m\times1} \end{pmatrix} \qquad X_{mat} = \begin{pmatrix} x_1^T \\ \vdots \\ x_m^T \end{pmatrix} \qquad Y_{mat} = diag\{y_1, \dots, y_m\}$$

$$A = \begin{array}{cc} & \begin{array}{cccc} t_1 & w_d & r_d & \varepsilon_m \end{array} \\ \begin{array}{c} m \; 1 \\ d \; 2 \\ d \; 3 \\ m \; 4 \\ 1 \; 5 \end{array} & \left[\begin{array}{c|c|c|c} 0_{m\times1} & Y_{mat}X_{mat\,(m\times d)} & 0_{m\times d} & I_{m\times m} \\ \hline 0_{d\times1} & -I_{d\times d} & I_{d\times d} & 0_{d\times m} \\ \hline 0_{d\times1} & I_{d\times d} & I_{d\times d} & 0_{m\times m} \\ \hline 0_{m\times1} & 0_{m\times d} & 0_{m\times d} & I_{m\times m} \\ \hline 1_{1\times1} & 0_{1\times d} & -1_{1\times d} & 0_{1\times m} \end{array}\right] \end{array}_{(2d+2m+1)\times(1+2d+m)}$$

$$V = \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{pmatrix} 1_{m\times1} \\ 0_{d\times1} \\ 0_{d\times1} \\ 0_{m\times1} \\ 0_{1\times1} \end{pmatrix}_{(2d+2m+1)\times 1}$$

שאלה 6:

$y_i := (-1)^{i+1}$

(a) בהינתן $S=\{(x_1,y_1),\dots,(x_m,y_m)\}$ נוכיח כי לכל איטרציה t של אלג' הפרספטרון על S, ולכל $i\le d$, $|w^{(t+1)}(i)|\le t$

נוכיח בעזרת אינדוקציה על מס' העדכונים t.

<u>בסיס:</u> $t=0$, לפי האלג' $w^{(1)}\leftarrow[0,\dots,0]^T$ אז $w^1(i)=0$ לכל $i\le d$ ולכן $|w^1(i)|=0\le t=0$

<u>הנחה:</u> נניח כי לכל $d=m\ge t>0$ ולכל $i\le d$, $|w^{(t)}(i)|\le t-1$.

<u>צעד:</u> נוכיח את הטענה עבור $t+1$.

נניח כי העדכון ה- $t+1$ הוא: $w^{(t+1)}\leftarrow w^{(t)}+y_ix_i$ אז צ"ל לכל $j\le d$ $|w^{(t+1)}(j)|\le t$.

$$|w^{(t+1)}(j)| = |w^{(t)}(j)+y_i x_i(j)| \overset{\text{אי שוויון המשולש}}{\le} |w^{(t)}(j)|+|y_i x_i(j)| \overset{(*)}{\le} |w^{(t)}(j)|+1 \overset{\text{ה.א}}{\le} t-1+1=t \quad \blacksquare$$

(*) המכפלה של $y_i\cdot x_i(j)$ יכולה להיות שווה ל- 1/-1/0, ובערך מוחלט יכולה להיות שווה ל- 1/0.

(b) יהי $w^{(T)}$ המפריד שאלג' הפרספטרון החזיר. כלומר (1) לכל i: $y_i\langle w^{(T)},x_i\rangle>0$

נוכיח באינדוקציה על i כי לכל i $|w^{(T)}(i)|\ge 2^{i-1}$

<u>בסיס:</u> $i=1$, לפי ההגדרה: $x_1=(1,0,\dots,0)$, $x_1(1)=(-1)^2=1$, $y_1=(-1)^{1+1}=1$

$x_1(1)=y_1=1$ $\qquad$ $\langle w^{(T)},x_1\rangle = w^{(T)}(1)$

$$|w^{(T)}(1)| = |w^{(T)}(1)\cdot y_1\cdot x_1(1)| = |y_1\cdot\langle w^{(T)},x_1\rangle| \overset{(2)}{\ge} 1 = 2^{1-1}$$

(2) לפי הרמז שבשאלה, המפריד מוגדר ע"י $w^{(T)}$ מתייג נכון עבור כל הדוגמאות שב-S.

בנוסף, ראינו בכיתה כי $\langle w^{(T)},x_i\rangle>0 \Rightarrow \langle w^{(T)},x_i\rangle\ge 1$, לכן מ-(1) נקבל כי לכל i $\langle w^{(T)},x_i\rangle\ge 1$.

<u>הנחה:</u> נניח כי לכל $d=m\ge i>1$ $|w^{(T)}(i)|\ge 2^{i-1}$

<u>צעד:</u> נוכיח את הטענה עבור $i+1$, כלומר נוכיח ש- $|w^{(T)}(i+1)|\ge 2^{i+1-1}=2^i$

נסתכל על x_{i+1} ו- y_{i+1}: לפי ההגדרה, $y_{(i+1)}=(-1)^{i+1+1}=(-1)^{i+2}$, $x_{(i+1)}=(\overbrace{(-1)^{i+1},\dots}^{<(i+1)},\overbrace{(-1)^{i+2}}^{=(i+1)},\overbrace{0,\dots,0}^{>(i+1)})$

$y_i=(-1)^{i+1}$, $x_i=((-1)^i,\dots,(-1)^{i+1},0,\dots,0)$

$$1 \overset{(2)}{\le} y_i \langle w^{(T)}, x_i \rangle = (-1)^{i+1} \cdot \left((-1)^i \sum_{j=1}^{i-1} w^{(T)}(j) + w^{(T)}(i) \cdot (-1)^{i+1} + 0 \right)$$

$$1 \overset{(2)}{\le} y_{(i+1)} \langle w^{(T)}, x_{(i+1)} \rangle = (-1)^{i+2} \cdot \left((-1)^{i+1} \sum_{j=1}^{i} w^{(T)}(j) + w^{(T)}(i+1) \cdot (-1)^{i+2} \right) = (-1)^{i+2} \cdot \left((-1)^{i+1} \sum_{j=1}^{i-1} \left(w^{(T)}(j) + w^{T}(i)\right) + w^{(T)}(i+1) \cdot (-1)^{i+2} \right)$$

נחלק למקרים (זוגי ואי-זוגי):

(i זוגי) $$2 \le y_i \langle w^{(T)}, x_i \rangle + y_{(i+1)} \langle w^{(T)}, x_{(i+1)} \rangle = -1 \cdot \left(1 \cdot \sum_{j=1}^{i-1} w^{(T)}(j) - w^{(T)}(i) \right) + 1 \left(-\sum_{j=1}^{i-1} \left(w^{(T)}(j) + w^{T}(i)\right) + w^{(T)}(i+1) \right)$$

$$= -2 \cdot \sum_{j=1}^{i-1} w^{(T)}(j) + w^{(T)}(i+1)$$

(i אי-זוגי) $$2 \le y_i \langle w^{(T)}, x_i \rangle + y_{(i+1)} \langle w^{(T)}, x_{(i+1)} \rangle = 1 \cdot \left(-\sum_{j=1}^{i-1} w^{(T)}(j) + w^{(T)}(i) \right) - 1 \left(\sum_{j=1}^{i-1} \left(w^{(T)}(j) + w^{T}(i)\right) - w^{(T)}(i+1) \right)$$

$$= -2 \cdot \sum_{j=1}^{i-1} w^{(T)}(j) + w^{(T)}(i+1)$$

נשים לב שעבור שני המקרים אנו מקבלים אותה תוצאה.

$$2 \le -2 \cdot \sum_{j=1}^{i-1} w^{(T)}(j) + w^{(T)}(i+1)$$

$$\Downarrow$$

$$w^{T}(i+1) \ge 2 + 2 \sum_{j=1}^{i-1} w^{(T)}(j) \overset{(\text{ה.א})}{\ge} 2 + 2 \cdot \sum_{j=1}^{i-1} 2^{j-1} = 2 + 2 \cdot \Big(\overbrace{\frac{1(2^{i-1}-1)}{1}}^{\text{סכום סדרה הנדסית}} \Big) = 2 + 2^i - 2 = 2^i \quad \blacksquare$$

(c) בסעיף (a) הוכחנו כי לכל אי' t ולכל $i \le d$ $|w^{(t+1)}(i)| \le t$.

בסעיף (b) הוכחנו כי לכל $i \le d$ $|w^{(T)}(i)| \ge 2^{i-1}$.

לכן לפי (a): $T \ge |w^{(T)}(d)|$

ולפי (b): $|w^{T}(d)| \ge 2^{d-1}$

לכן נקבל כי $T \ge 2^{d-1}$,

כלומר מס' העדכונים א' אזי הפספטרון אקספוננציאלי ב-d.