מבוא לבינה מלאכותית - תרגול 1.

רגב יחזקאל אימרה.

4 בנובמבר 2025

יש הרבה סוגי למידה בעולם, הנה חלקם:

1. למידה מפוקחת: לכל דגימה יש את התווית שלה (label).
   (א) סיווג: חיזוי קטגוריות.
   (ב) רגרסיה: חיזוי ערך רציף.
2. למידה לא מפוקחת: יש דגימות, אין תוויות.
   (א) אשכול: חלוקת דגימות לקבוצות.
   (ב) הורדת מימדים: הפחתת כמות הפיצ'רים של המודל.
   (ג) זיהוי אנומליות: גילוי דגימות חריגות מבין סך כל הדגימות.
3. למידה באמצעות חיזוקים: המודל לומד דרך ניסוי וטעייה. המודל מבצע פעולות ומקבל חיזוקים (חיוביים או שליליים) ומשפר את המדיניות שלו.
4. למידה מונחית עצמית: המערכת מייצרת לעצמה תוויות מתוך הנתונים כדי ללמוד ייצוגים יותר טובים.
5. למידה גנרטיבית: המערכת מייצרת נתונים חדשים הדומים לנתונים המקוריים עליהם אומנה.

בקורס הזה נתמקד בעיקר בלמידה מפוקחת. למעוניינים, המחלקה למתמטיקה מציעה קורס בלמידה לא מפוקחת, והמחלקה למדעי המחשב מציעה קורסים בשער סוגי הלמידה.

חלק I

מודלי רגרסיה.

1 רגרסיה לינארית.

לאלו שלא זוכרים, בלמידה מפוקחת יש שתי נישות מרכזיות: קלסיפיקציה ורגרסיה.

הגדרה 1.1. מודל **רגרסיה** הוא מודל סטיטיסטי הנועד להעריך קשר בין משתנים.

דוגמה 2.1. דוגמאות למודל רגרסיה:

1. חיזוי מחיר של בית לפי גודל ומיקום.
2. חיזוי ציון של סטודנט לפי מספר שעות למידה.
3. חיזוי הטמפרטורה של מחר על סמך נתוני מזגל האוויר של הימים האחרונים.

איך עובד מודל רגרסיה?

1) הנחות המודל: נניח יש לנו אוסף של N דגימות והתיוגים שלהן, $\{(x_i,y_i)\}_{i=1}^N$ כאשר $x_i\in\mathbb{R}^d$ וכן $y\in\mathbb{R}$. רגרסיה לינארית, כשמה כן היא, מניחה כי הקשר בין x ו־y הוא לינארי, כלומר שקיימים $d+1$ סקלרים b, $\{w_i\}_{i=1}^d$ כך שמתקיים $y_i=\sum_{n=1}^d x_{ij}w_j+b=w^Tx_i+b$. נשאלת השאלה, איך אנחנו מגלים מיהם הסקלרים w_i,b הנ"ל?

2) פונקציית הפסד: לכל דגימה (x_i,y_i) המודל חוזה ערך כלשהו $\hat{y}_i$. נגדיר את השגיאה של המודל להיות $error_i=y_i-\hat{y}_i$. אנחנו מעוניינים במספר אחד שיהיה תלוי בכל השגיאות שלנו ושיגיד לנו כמה רע המודל שלנו. יש לכך כמה בחירות טבעיות:

- סכום השגיאות: $\Sigma(y_i-\hat{y}_i)$. הבעיה היא שפה אם יש לי שגיאה חיובים ושגיאה שלילית הן יתבטלו, מה שיכול להוביל לדיווחים שגויים.
- סכום ערכים מוחלטים: $\Sigma\left|y_i-\hat{y}_i\right|$. יותר טוב, אבל פה הבעיה היא שזה לא גזיר - הרי אם אנחנו רוצים למזער את השגיאה הכוללת שלנו נצטרך לגזור אותה.
- סכום השגיאות בריבוע: $\Sigma(y_i-\hat{y}_i)^2$. מעולה בשבילנו. זה גזיר ונותן יותר משקל ככל שהשגיאה גדלה.

הגדרה 3.1. פונקציית הפסד (loss function) היא פונקציה הממפה את התחזית של המודל ואת הערך האמיתי למספר ממשי, המייצג את מידת הטעות או העלות של התחזית.

באופן טבעי, ככל שפונקציית ההפסד מניבה ערך נמוך יותר עבור תחזיות המודל, כך ביצועי המודל טובים יותר. מטרתנו היא למזער את ערך פונקציית ההפסד.

הגדרה 4.1. נגדיר את ה־loss function **טעות הריבועית הממוצעת** (MSE) להיות

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

כאשר θ הם הפרמטרים של המודל שלנו שאנחנו מעוניינים ללמוד (ללמוד = למזער את ה־$loss$ שלנו), ו־$\hat{y}_i = f(x_i;\theta)$ זה הערך שהמודל פולט לדגימה x_i עם פרמטרים θ. נרצה למצוא $\hat{\theta} = \arg\min_\theta \mathcal{L}(\theta)$.

זוהי פונקציית הפסד סטנדרטית בבעיות רגרסיה בכללית לאו דווקא ברגרסיה לינארית.

3) תהליך הלמידה: נציין לפרוטוקול כי קיים פתרון סגור לבעיה הזו: נכתוב את הדאטה שלנו בכתיב מטריציוני:

- $X \in \mathbb{R}^{N\times(d+1)}$ המוגדר להיות $X = \begin{pmatrix} 1 & - & x_1^T & - \\ 1 & - & x_2^T & - \\ \vdots & & \vdots & \\ 1 & - & x_N^T & - \end{pmatrix}$.
- $y \in \mathbb{R}^N$ ־ כל הערכים שלנו בצורת וקטור $y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}$.
- $\beta \in \mathbb{R}^{d+1}$ כל הפרמטרים שאנחנו רוצים ללמוד $\beta = \begin{pmatrix} b \\ w_1 \\ \vdots \\ w_d \end{pmatrix}$.

כלומר המודל שלנו הופך להיות

$$\hat{y} = X\beta$$

ופונקציית ההפסד שלנו היא

$$\mathcal{L}(\beta) = \frac{1}{N}\|y - X\beta\|_2^2$$

כדי למזער, נגזור ביחס ל־β ונשווה לאפס:

$$\nabla_\beta \mathcal{L} = -\frac{2}{N}X^T(y - X\beta)$$

כלומר נקבל

$$X^T y = X^T X\beta$$

ולפי גאוס הפתרון יהיה

$$\hat{\beta} = \left(X^T X\right)^{-1} X^T y$$

הבעיה: לרוב, המטריצה $X^T X$ לא הפיכה, לכן נצטרך להיות יותר חכמים בחיפוש שלנו אחרי $\hat{\beta}$.

הפתרון: ניעזר באלגוריתם מורד הגרדיאנט:

אלגוריתם 5.1. אלגוריתם מורד הגרדיאנט:

- בחרו נקודת התחלה θ_0.
- חשבו את $\nabla_\theta \mathcal{L}(\theta_0)$.
- בחרו גודל צעד $\eta > 0$.
- חשבו $\theta_{k+1} = \beta_k - \eta\nabla_\theta \mathcal{L}(\theta_0)$.

בצורת כתיבה יותר כללית נכתוב

$$\theta^+ = \theta - \eta\nabla_\theta \mathcal{L}(\theta)$$

מוטיבציה: הגרדיאנט של פונקציה הוא כיוון העליה הכי מהיר שלה, לכן לחסר מהמיקום הנוכחי כפולה שלילית של הגרדיאנט תוריד אותנו אל עבר מינימום מקומי. לפיכך הוספה של כפולה שלילית של הגרדיאנט של פונקציית ההפסד ל־θ אמורה להוביל לאורך זמן לירידה בערך הMSE.

חשוב! צעד חשוב באלגוריתם מורד הגרדיאנט הוא בחירה נכונה של גודל הצעד: גודל צעד גדול מדי יגרום לכך שבחיים לא נתכנס, וגודל צעד קטן מדי יגרום לכך שנתכנס מקצב מאוד איטי (לצייר).

בדר"כ בוחרים $\eta = 0.05$, ואם צריך משנים את הערך במהלך האימון.

הגדרה 6.1. **היפרפרמטר** הוא פרמטר שקובעים על מנת להגדיר חלק כלשהו במודל.

במקרה שלנו קצב הלמידה הוא היפר פרמטר, כי הוא קובע כמה מהר מורד מגרדיאנט מתכנס ולכן הוא קובע האם יהיה למודל אימון איטי או מהיר.

הערה 7.1. בעיה מרכזית של מורד הגרדיאנט היא שאנחנו יכולים להיתקע במינימום מקומי (לצייר!) לא משנה כמה פעמים נריץ מורד הגרדיאנט. לכן ישבו אנשים חכמים והמציאו אלגוריתמים אחרים הידועים כתור $optimizers$ (נרחיב עליהם בהמשך הקורס אולי), כגון

1. SGD שזה לעשות GD ולהוסיף רעש גאוסיאני לצעד שלנו.
2. $Momentum$ שזה ככל שאנחנו מתקדמים בציר אחד יותר זמן, נתקדם בו יותר מהר.
3. $Nesterov$ שזה כמו $Momentum$ אבל קצת יותר חכם.
4. $AdaGrad$ שמוריד את ה־$learning\ rate$ ככל שמשיכים לאמן.
5. RMS שמוריד גם הוא את ה־lr אבל מונע מזה לקטון יותר מדי.
6. $Adam$ שמשלב RMS ו־$Momentum$.
7. ולאחרונה גם $Moun$.

מסקנה: כלומר התחום עדיין פעיל ואין באמת שיטה הכי טובה למצוא את הפרמטרים הכי טובים אם אין פתרון סגור.

שאלה 8.1. מתי עוצרים את האימון?

פתרון: יש כמה אפשרויות:

1. כאשר $\|\nabla\mathcal{L}\| < \varepsilon$.
2. גגמר התקציב־ נניח מסיבה מסוימת יש לי רק שעה על המחשב להריץ את האלגוריתם־ מה שיצא אני מרוצה.
3. אם חשוב להגיע לערך כלשהו של המודל (פרטים עוד רגע) ומצליחים, עוצרים.
4. כשעשיתי מספר צעדים שהגדרתי מראש.

2 מטריקות הערכת המודל.

עכשיו אחרי שאימנו את המודל שלנו, איך אנחנו יודעים בפועל כמה טוב הוא? רק בגלל שדחפנו את הדאטה שלנו למודל ואמרנו תלמד לא אומר שהוא למד. אנחנו צריכים מדד כלשהו שיאמר לנו כמה טוב המודל מכליל את הלמידה שלו, כלומר אנחנו דרך כלשהי להעריך את המודל שלנו.

הרעיון הוא מאוד פשוט: במקום לדחוף את כל הדאטה שלנו למודל נחלק את הדאטה שלנו לשתי קבוצות זרות: **קבוצת אימון** (train set) **וקבוצת מבחן** (test set). ככה נוכל למדוד את השגיאה של המודל על דאטה שהוא עוד לא ראהת ושאנחנו יודעים מה הערך האמיתי שלו.

ניזכר כי

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{1}$$

מודד את המרחק בריבוע הממוצע של החיזויים (פרדיקציות) שלנו מהערכים האמיתיים, לכן אם נקטין את ה־MSE שלנו נקבל מודל טוב יותר. הבעיה היחידה היא שהיחידות מידה של MSE הן היחידות מידה של y בריבוע, לכן כדי לתקן את זה נגדיר

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{2}$$

כיוון שה־$RMSE$ באותו יחידות מידה כמו של y, יהיה יותר קל לפרש אותו ולהחליט לפיו האם המודל טוב או לא טוב.

עוד אפשרות היא להגדיר

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|^2 \tag{3}$$

היתרון של MAE זה שהוא רגיש לאוטליירים (נקודות ששונות משמעותית משאר הדאטה) ולפיכך הוא בחירה טובה אם לדאטה שלי יש לעיתים ערכים קיצוניים.

עוד אפשרות נקראת R^2 והיא הדרך הכי שכיחה לסכם כמה טוב המודל שלי:

$$R^2 = 1 - \frac{\Sigma_i(y_i - \hat{y}_i)^2}{\Sigma_i(y_i - \bar{y})^2} \tag{4}$$

כאשר $\bar{y} = \frac{1}{N}\Sigma_i y_i$ זה הממוצע של y.

האינטואיציה היא שהמונה מודד עד כמה רחוקות התחזיות מהאמת, בעוד שהמכנה מודד עד כמה הנתונים רחוקים מניבוי הממוצע בלבד.

אם המודל מושלם אז $R^2 = 1$, אם המודל מוצע (תרתי משמע חוזה כל הזמן את הממוצע) אז $R^2 = 1$, ואם $R^2 < 0$ אז המודל ממש גרוע.

דרך נוספת למדוד בעין כמה טוב המודל שלנו היא על ידי לשרטט את y מול $\hat{y}$: ככל שהמודל יותר טוב נצפה מהזוג $(y_i, \hat{y}_i)$ להיות קרוב יותר לישר $y = x$. לזה קוראים scatter plot.

ניתן גם לשרטט את השארית $Residual_i = y_i - \hat{y}_u$ וככל שהמודל יותר טוב נצפה שהשאריות יהיו קרובות ל־0.

הערה 1.2. חלוקה טובה של הדאטה ל־train \ test היא $80\%\backslash 20\%$ בהתאמה, נרחיב על כך בהמשך.

תרגיל 2.2. פתחו את כלל האימון של אלגוריתם מורד הגרדיאנט לרגרסיה לינארית.

פתרון. ניזכר כי כלל העדכון הוא $\theta^+ = \theta - \eta\nabla\mathcal{L}(\theta)$ וכן ברגרסיה לינארית $\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$. אם נציב $\hat{y}_i = w^T x_i + b$ נקבל

$$\mathcal{L}(w,b) = \frac{1}{N}\sum_{i=1}^{N}(y_i - (w^T x_i + b))^2$$

נגזור ביחס לכל אחד מהפרמטרים:

$$\frac{\partial\mathcal{L}}{\partial w} = \frac{\partial}{\partial w}\frac{1}{N}\sum_{i=1}^{N}(y_i - (w^T x_i + b))^2 = \frac{2}{N}\sum_{i=1}^{N} -x_i(y_i - (w^T x_i + b)) = \frac{2}{N}\sum_{i=1}^{N} -x_i(y_i - \hat{y}_i) = \frac{2}{N}\sum_{i=1}^{N} x_i(\hat{y}_i - y_i)$$
$$\frac{\partial\mathcal{L}}{\partial b} = \frac{\partial}{\partial w}\frac{1}{N}\sum_{i=1}^{N}(y_i - (w^T x_i + b))^2 = -\frac{2}{N}\sum_{i=1}^{N}(y_i - (w^T x_i + b)) = -\frac{2}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i) = \frac{2}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)$$

נהוג להגדיר $\mathbb{R}^{N\times d} \ni X = \begin{pmatrix} -x_1^T- \\ \vdots \\ -x_N^T- \end{pmatrix}$ וכן $\mathbb{R}^N \ni y = \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix}, \mathbb{R}^N \ni \hat{y} = \begin{pmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_N \end{pmatrix}$ ולכן לפי כפל מטריצות $\sum_{i=1}^{N} x_i(\hat{y}_i - y_i) = X^T(\hat{y} - y)$ כלומר

$$\frac{\partial\mathcal{L}}{\partial w} = \frac{2}{N}X^T(\hat{y} - y)$$
$$\frac{\partial\mathcal{L}}{\partial b} = \frac{2}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)$$

ולכן נקבל

$$w^+ = w - \eta\frac{2}{N}X^T(\hat{y} - y)$$
$$b^+ = b - \eta\frac{2}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)$$

הערה 3.2. אם נסמן במקום $X = \begin{pmatrix} 1 & - & x_1^T & - \\ 1 & - & x_2^T & - \\ \vdots & & \vdots & \\ 1 & - & x_N^T & - \end{pmatrix}, y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}, \beta = \begin{pmatrix} b \\ w_1 \\ \vdots \\ w_d \end{pmatrix}$ נוכל לכתוב

$$\beta^+ = \beta - \eta\frac{2}{N}X^T(\hat{y} - y)$$