

דף נוסחאות

נוטציות

משתנים אקראיים

- x - אותיות לא מוטות לועזיות או יווניות - משתנים אקראיים.
- \mathbf{x} - אותיות לא מוטות מודגשות לועזיות או יווניות - וקטורים אקראיים.

Sets (קבוצות)

נסמן קבוצה של איברים באופן הבא:

- $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}\}$ - סדרה של n וקטורים.

אלגברה לינארית

- x - אותיות סטנדרטיות (italic lower case) לועזיות או יווניות - סקלרים.
- \mathbf{x} - אותיות מודגשות - וקטורי עמודה
- \mathbf{x}^\top - וקטורי שורה
- x_i - האיבר ה- i בוקטור \mathbf{x} .
- (בכתב יד נשתמש בחץ (במקום באותיות מדגשות) בכדי לסמן וקטורים: \vec{x}).
- $\langle \mathbf{x}, \mathbf{y} \rangle (= \mathbf{x}^\top \mathbf{y} = \sum_i x_i y_i)$ - המכפלה הפנימית הסטנדרטית בין \mathbf{x} ל \mathbf{y} .
- $\|\mathbf{x}\|_2 (= \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle})$ - הנורמה הסטנדרטית (נורמת l_2) של הוקטור \mathbf{x} .
- $\|\mathbf{x}\|_l (= \sqrt[l]{\sum_i |x_i|^l})$ - נורמת l של \mathbf{x}
- \mathbf{A} - אותיות לועזיות גדולות מודגשות (bold) (capittal) - מטריצה
- \mathbf{A}^\top - המטריצה Transposed \mathbf{A} (המטריצה המשוחלפת).
- $A_{i,j}$ - האיבר ה- j שורה ה- i של \mathbf{A} .
- $A_{i,:}$ - השורה ה- i של \mathbf{A} .
- $A_{:,i}$ - העמודה ה- i של \mathbf{A} .

נגזרות

נגזרת מטריצית שימושית

$$\frac{\partial}{\partial \mathbf{A}} \mathbf{x}^\top \mathbf{A} \mathbf{x} = \mathbf{x} \mathbf{x}^\top$$

נגזרות וקטוריות שימושיות

$$\frac{\partial}{\partial \mathbf{x}} \mathbf{a}^\top \mathbf{x} = \mathbf{a}$$

$$\frac{\partial}{\partial \mathbf{x}} \mathbf{A} \mathbf{x} = \mathbf{A}$$

$$\frac{\partial}{\partial \mathbf{x}} \mathbf{x}^\top \mathbf{A} \mathbf{x} = (\mathbf{A} + \mathbf{A}^\top) \mathbf{x}$$

$$\frac{\partial}{\partial \mathbf{x}} \|\mathbf{A} \mathbf{x} + \mathbf{b}\|_2^2 = 2\mathbf{A}^\top (\mathbf{A} \mathbf{x} + \mathbf{b})$$

בעיית אופטימיזציה

בעיות מהצורה:

$$\begin{aligned} \theta^* = \arg \min_{\theta} f(\theta) \\ \text{subject to } g_i(\theta) \leq 0, \quad i = 1, \dots, m \\ h_j(\theta) = 0, \quad j = 1, \dots, p \end{aligned}$$

כאשר $g_i(\theta) \geq 0$ נקראים אילוצי אי-שוויון.

ו $h_i(\theta) = 0$ נקראים אילוצי שוויון.

הפונקציה $f(\theta)$ נקראת ה **objective**.

כמות האילוצים יכולה להשתנות ובמקרים רבים לא יופיעו אילוצים כלל.

בעיות חיזוי

בבעיות חיזוי ננסה למצוא חזאי לערכו של משתנה/וקטור אקראי y על סמך משתנה אקראי/וקטור x :

$$\hat{y} = h(x)$$

הערכת ביצועים

אנו נרצה לבחור את החזאי אשר ימעזר את **פונקציית המחיר (cost) $C(h)$** אשר נותנת ציון לכל חזאי (לרוב הציון מוגדר כאשר ציון נמוך יותר הוא טוב יותר):

$$h^* = \arg \min_h C(h)$$

מקרה פרטי של פונקציית מחיר הינה המקרה של **פונקציית סיכון (risk)**. פונקציית סיכון היא פונקציה מהצורה של:

$$R(h) = \mathbb{E} [l(h(x), y)]$$

הפונקציה l מוכנה **פונקציית הפסד (loss)**.

פונקציות הפסד (פונקציות סיכון) נפוצות

Common For	Loss Name	Risk Name	Loss Function	Optimal Predictor
Classification	Zero-One Loss	Misclassification Rate	$l(y_1, y_2) = I\{y_1 \neq y_2\}$	$h^*(x) = \arg \max_y p_{y x}(y x)$
Regression	L_1	Mean Absolute Error	$l(y_1, y_2) = y_1 - y_2 $	Median: $h^*(x) = \hat{y}$ s.t. $F_{y x}(\hat{y} x) = 0.5$
Regression	L_2	Mean Squared Error (MSE)	$l(y_1, y_2) = (y_1 - y_2)^2$	$h^*(x) = \mathbf{E}[y x]$

Supervised learning

בעיות חיזוי בהם הפילוג של המשתנים לא ידוע אך יש בידינו מדגם: $\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^N$

סוגי supervised learning

- בעיות סיווג (y): **classification** דיסקרטי וסופי.
- בעיות רגרסיה (y): **regression** רציף.

גישות לפתרון בעיות supervised learning

ניתן להבחין בין 3 גישות לפתרון בעיות supervised learning:

- גישה דיסקרימינטיבית: $\mathcal{D} \rightarrow h(\mathbf{x})$
- גישה גנרטיבית: $\mathcal{D} \rightarrow p_{\mathbf{x},y}(\mathbf{x}, y) \rightarrow p_{y|\mathbf{x}}(y|\mathbf{x}) \rightarrow h(\mathbf{x})$
- גישה דיסקרימינטיבית הסתברותית: $\mathcal{D} \rightarrow p_{y|\mathbf{x}}(y|\mathbf{x}) \rightarrow h(\mathbf{x})$

גישה דיסקרימינטיבית

Empirical Risk Minimization

שיטה אשר משתמשת במודל פרמטרי לפונקציית החיזוי, בה נחפש את הפרמטרים של המודל אשר ממזערים את הסיכון האמפירי (הסיכון שמשוערך על המדגם):

$$h^* = \arg \min_h \frac{1}{N} \sum_{i=1}^N l(h(\mathbf{x}^{(i)}; \theta), y^{(i)})$$

כאשר המודל הפרמטרי יכול להיות כל מודל, לדוגמא, מודל לינארי או רשת נוירונים.

Ridge Regression

בעיית LLS אשר מוסיפים לה איבר רגולריזציה L_2 :

$$\theta^* = \arg \min_{\theta} \frac{1}{N} \sum_i (\theta^\top \mathbf{x}^{(i)} - y^{(i)})^2 + \lambda \|\theta\|_2^2$$

גם כאן יש פתרון סגור:

$$\theta = (\mathbf{X}^\top \mathbf{X} + N\lambda \mathbf{I})^{-1} \mathbf{X}^\top \mathbf{y}$$

Least Absolute Shrinkage and Selection Operator ((LASSO

בעיית LLS אשר מוסיפים לה איבר רגולריזציה L_1 :

$$\theta^* = \arg \min_{\theta} \frac{1}{N} \sum_i (\theta^\top \mathbf{x}^{(i)} - y^{(i)})^2 + \lambda \sum_j |\theta_j|$$

אין פתרון סגור אך ניתן לפתרון באופן יעיל על ידי שיטות איטרטיביות כגון gradient descent.

(Linear Least Squares (LLS

LLS הוא מקרה פרטי של ERM שבו המודל הפרמטרי הוא לינארי ופונקציית הסיכון היא MSE:

$$h(\mathbf{x}; \theta) = \mathbf{x}^\top \theta$$

$$\theta^* = \arg \min_{\theta} \frac{1}{N} \sum_i (\theta^\top \mathbf{x}^{(i)} - y^{(i)})^2$$

במקרה זה ישנו פתרון סגור אשר נתון על ידי:

$$\theta = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$$

כאשר \mathbf{X} היא מטריצת המדידות אשר מוגדרת:

$$\mathbf{X} = (\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(N)})^\top$$

ו \mathbf{y} היא מטריצת התוויות:

$$\mathbf{y} = (y^{(1)}, y^{(2)}, \dots, y^{(N)})^\top$$

(K-NN (K-Nearest Neighbours

K-NN הינו אלגוריתם דיסקרימינטיבי לפתרון בעיות סיווג. באלגוריתם זה החיזויים נעשים ישירות על פי המדגם באופן הבא:

בהינתן \mathbf{x} מסוים:

1. נבחר את K הדגימות בעלות ה $\mathbf{x}^{(i)}$ הקרובים ביותר ל \mathbf{x} . (לרוב נשתמש במרחק אוקלידי, אך ניתן גם לבחור פונקציות מחיר אחרות).
2. תוצאת החיזוי תהיה התווית השכיחה ביותר (majority vote) מבין K התוויות של הדגימות שנבחרו בשלב 1.

במקרה של שיוון:

- במקרה של שיוויון בשלב 2, נשווה גם את המרחק הממוצע בין ה \mathbf{x} -ים השייכים לכל תווית. אנו נבחר בתווית בעלת המרחק הממוצע הקצר ביותר.
- במקרה של שיוון גם בין המרחקים הממוצעים, נבחר אקראית.

K-NN לבעיות רגרסיה

ניתן להשתמש באלגוריתם זה גם לפתרון בעיות רגרסיה אם כי פתרון זה יהיה לרוב פחות יעיל. בבעיות רגרסיה ניתן למצב על התוויות במקום לבחור את תווית השכיחה.

Decision Trees

עץ החלטה אשר ממפה כל \mathbf{x} לעלה מסויים אשר מכיל חיזוי אשר נקבע מראש.

שיטה לבניית העץ הינה באופן חמדני אשר בכל שלב מוסיף את הפיצול הטוב ביותר תחת קריטריון מסויים.

נמספר את העלים של עץ נתון על ידי $j = 1, 2, \dots$. בעבור עלה מסויים j בעץ נגדיר:

- מספר התוויות אשר משויכות לעלה מסויים בעץ: N_j .

- הפילוג האמפירי של התוויות בעלה מסויים:

$$\hat{p}_{j,y} = \frac{1}{N_j} \sum_{i \in \mathcal{I}_j} I\{y_i = y\}$$

שני קריטריונים נפוצים הינם:

- אינדקס Gini:

$$Q_j = \sum_{y \in \{1, \dots, C\}} \hat{p}_{j,y} (1 - \hat{p}_{j,y})$$

- אנטרופיה:

$$Q_j (= H_j) = \sum_{y \in \{1, \dots, C\}} -\hat{p}_{j,y} \log_2 \hat{p}_{j,y}$$

בכל צעד נרצה לבחור את הפיצול אשר ממזער את הגודל:

$$Q_{\text{total}} = \sum_j \frac{N_j}{N} Q_j$$

Soft SVM

בעיה פרימאלית

$$\begin{aligned} \mathbf{w}^*, b^*, \{\xi_i\}^* = \\ \arg \min_{\mathbf{w}, b, \{\xi_i\}} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i \\ \text{s.t. } y^{(i)} (\mathbf{w}^\top \mathbf{x}^{(i)} + b) \geq 1 - \xi_i \quad \forall i \\ \xi_i \geq 0 \quad \forall i \end{aligned}$$

בעיה דואלית

$$\begin{aligned} \{\alpha_i\}^* = \\ \arg \max_{\{\alpha_i\}} \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} y^{(i)} y^{(j)} \alpha_i \alpha_j \mathbf{x}^{(i)\top} \mathbf{x}^{(j)} \\ \text{s.t. } 0 \leq \alpha_i \leq C \quad \forall i \\ \sum_i \alpha_i y^{(i)} = 0 \end{aligned}$$

תכונות

α_i	משוואה	תכונה
$\alpha_i = 0$	$y^{(i)} (\mathbf{w}^\top \mathbf{x}^{(i)} + b) > 1$	נקודות שמסוגלות נכון ורחוקות מה margin
$0 \leq \alpha_i \leq C$	$y^{(i)} (\mathbf{w}^\top \mathbf{x}^{(i)} + b) = 1$	נקודות על ה margin (support vectors)
$\alpha_i = C$	$y^{(i)} (\mathbf{w}^\top \mathbf{x}^{(i)} + b) = 1 - \xi_i$	נקודות שחרגות מה margin גם support (vectors)

Hard-SVM

בעיה פרימאלית

$$\begin{aligned} \mathbf{w}^*, b^* = \arg \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{s.t. } y^{(i)} (\mathbf{w}^\top \mathbf{x}^{(i)} + b) \geq 1 \quad \forall i \end{aligned}$$

בעיה דואלית

$$\begin{aligned} \{\alpha_i\}^* = \\ \arg \max_{\{\alpha_i\}} \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} y^{(i)} y^{(j)} \alpha_i \alpha_j \mathbf{x}^{(i)\top} \mathbf{x}^{(j)} \\ \text{s.t. } \alpha_i \geq 0 \quad \forall i \\ \sum_i \alpha_i y^{(i)} = 0 \end{aligned}$$

מתוך הפרמטרים α_i ניתן לשחזר את \mathbf{w} באופן הבא:

$$\mathbf{w} = \sum_i \alpha_i y^{(i)} \mathbf{x}^{(i)}$$

את b מוצאים על ידי בחירת support vector והצבה לאילוף של הבעיה הפרימאלית.

תכונות

α_i	משוואה	תכונה
$\alpha_i = 0$	$y^{(i)} (\mathbf{w}^\top \mathbf{x}^{(i)} + b) > 1$	נקודות רחוקות מה margin
$\alpha_i \geq 0$	$y^{(i)} (\mathbf{w}^\top \mathbf{x}^{(i)} + b) = 1$	נקודות על ה margin (support vectors)

שיטות לא פרמטריות

היסטוגרמה

1. מחלקים את תחום הערכים של \mathbf{x} יכול לקבל ל bins (תאים) לא חופפים אשר מכסים את כל התחום.
2. לכל תא משערכים את ההסתברות של המאורע ש \mathbf{x} נמצא בתוך התא.
3. הערך של פונקציית הצפיפות בכל תא תהיה ההסתברות המשווערכת להיות בתא חלקי גודל התא.

נרשום זאת בעבור המקרה של משתנה אקראי סקלרי. נסמן ב B את מספר התאים וב l_b ו r_b את הגבול השמאלי והימני בהתאמה של התא ה b . ההסטוגרמה תהיה נתונה על ידי:

$$\hat{p}_{\mathbf{x},\mathcal{D}}(\mathbf{x}) = \begin{cases} \frac{1}{\text{size of bin } 1} \hat{p}_{\{\mathbf{x} \text{ in bin } 1\},\mathcal{D}} & \mathbf{x} \text{ in bin } 1 \\ \vdots \\ \frac{1}{\text{size of bin } B} \hat{p}_{\{\mathbf{x} \text{ in bin } B\},\mathcal{D}} & \mathbf{x} \text{ in bin } B \end{cases}$$

$$= \begin{cases} \frac{1}{N(r_1-l_1)} \sum_{i=1}^N I\{l_1 \leq \mathbf{x}^{(i)} < r_1\} & l_1 \leq \mathbf{x} < r_1 \\ \vdots \\ \frac{1}{N(r_B-l_B)} \sum_{i=1}^N I\{l_B \leq \mathbf{x}^{(i)} < r_B\} & l_B \leq \mathbf{x} < r_B \end{cases}$$

הערות:

- בחירת התאים משפיעה באופן משמעותי על תוצאת השערוך של ה PDF.
- כלל אצבע: לחלק את טווח הערכים ל- \sqrt{N} תאים בגודל אחיד.

(Kernel Density Estimation (KDE

$$\hat{p}_{\mathbf{x},\phi,\mathcal{D}}(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^N \phi(\mathbf{x} - \mathbf{x}^{(i)})$$

כאשר $\phi(\mathbf{x})$ מכונה **פונקציית גרעין (kernel)** או **Parzan window**. פונקציות גרעין נפוצות הינן:

- חלון מרובע:

$$\phi_h(\mathbf{x}) = \frac{1}{h^D} I\{|x_j| \leq \frac{h}{2} \quad \forall j\}$$

- גאוסיאן:

$$\phi_\sigma(\mathbf{x}) = \frac{1}{\sqrt{2\pi}\sigma^D} \exp\left(-\frac{\|\mathbf{x}\|_2^2}{2\sigma^2}\right)$$

כאשר h או σ הם hyper-parameters של המודל.

כלל אצבע לבחירת רוחב הגרעין במקרה הגאואסי הסקלרי הינו $1.06 \text{ std}(\mathbf{x}) N^{-\frac{1}{5}}$, $\sigma = \left(\frac{4 \cdot \text{std}(\mathbf{x})^5}{3N}\right)^{\frac{1}{5}}$, כאשר $\text{std}(\mathbf{x})$ הינה הסטיית תקן של \mathbf{x} (אשר לרוב תהיה משוערכת גם היא מתוך המדגם)

שיטות פרמטריות

בשיטה הפרמטרית נציע מודל פרמטרי לפילוג המשותף של x ו y . שתי שיטות נפוצות למציאת הפרמטרים הינם:

MLE

$$\hat{\theta}_{MLE} = \arg \max_{\theta} \mathcal{L}(\theta; \mathcal{D}) = \arg \min_{\theta} - \sum_i \log \left(p_{\mathbf{x},y}(\mathbf{x}^{(i)}, y^{(i)}; \theta) \right)$$

לרוב נוח לנסות לבנו את המודל כמכפלה של שני פונקציות.

$$p_{\mathbf{x},y}(\mathbf{x}, y) = p_{\mathbf{x}|y}(\mathbf{x}|y)p_y(y)$$

MAP

$$\hat{\theta}_{MAP} = \arg \max_{\theta} p_{\theta|\mathcal{D}}(\theta|\mathcal{D}) = \arg \min_{\theta} - \log(p_{\theta}(\theta)) - \sum_i \log \left(p_{\mathbf{x},y|\theta}(\mathbf{x}^{(i)}, y^{(i)}|\theta) \right)$$

Quadratic Discriminant Analysis (QDA)

QDA דומה ל LDA רק שכאן ישנה מטריצה Σ_c לכל מחלקה:

$$p_{\mathbf{x}|y}(\mathbf{x}|y) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_y|}} e^{-\frac{1}{2}(\mathbf{x}-\mu_y)^T \Sigma_y^{-1}(\mathbf{x}-\mu_y)}$$

הפרמטר Σ_c נתון על ידי:

$$\Sigma_c = \frac{1}{|\mathcal{I}_c|} \sum_{i \in \mathcal{I}_c} (\mathbf{x}^{(i)} - \mu_{y^{(i)}}) (\mathbf{x}^{(i)} - \mu_{y^{(i)}})^T$$

במקרה של misclassification rate בינארי, המשערך נתון על ידי:

$$h(x) = \begin{cases} 1 & \mathbf{x}^T C \mathbf{x} + \mathbf{a}^T \mathbf{x} + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

כאשר:

$$C = \frac{1}{2}(\Sigma_0^{-1} - \Sigma_1^{-1})$$

$$\mathbf{a} = \Sigma_1^{-1} \mu_1 - \Sigma_0^{-1} \mu_0$$

$$b = \frac{1}{2} (\mu_0^T \Sigma_0^{-1} \mu_0 - \mu_1^T \Sigma_1^{-1} \mu_1) + \log \left(\frac{\sqrt{|\Sigma_0|} p_y(1)}{\sqrt{|\Sigma_1|} p_y(0)} \right)$$

Linear Discriminant Analysis (LDA)

LDA משתמש במודל הבא + MLE:

$$p_{\mathbf{x}|y}(\mathbf{x}|y) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} e^{-\frac{1}{2}(\mathbf{x}-\mu_y)^T \Sigma^{-1}(\mathbf{x}-\mu_y)}$$

לפרמטרים של המודל יש פתרון סגור. נשתמש בסימונים:

- $\mathcal{I}_c = \{i : y^{(i)} = c\}$ - זאת אומרת, אוסף האינדקסים של הדגמים במדגם שמקיימים $y^{(i)} = c$.
- $|\mathcal{I}_c|$ - מספר האינדקסים ב \mathcal{I}_c .
- μ_c - וקטורי התוחלת של הפילוג הנורמאלי $p_{\mathbf{x}|y}(\mathbf{x}|c)$.

$$\mu_c = \frac{1}{|\mathcal{I}_c|} \sum_{i \in \mathcal{I}_c} \mathbf{x}^{(i)}$$

$$\Sigma = \frac{1}{N} \sum_i (\mathbf{x}^{(i)} - \mu_{y^{(i)}}) (\mathbf{x}^{(i)} - \mu_{y^{(i)}})^T$$

את הפילוג של p_y לומדים מתוך הפילוג האמפירי של y במדגם.

במקרה של misclassification rate בינארי, המשערך נתון על ידי:

$$h(x) = \begin{cases} 1 & \mathbf{a}^T \mathbf{x} + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

כאשר:

$$\mathbf{a} = \Sigma^{-1} (\mu_1 - \mu_0)$$

$$b = \frac{1}{2} (\mu_0^T \Sigma^{-1} \mu_0 - \mu_1^T \Sigma^{-1} \mu_1) + \log \left(\frac{p_y(1)}{p_y(0)} \right)$$

גישה דיסקרימינטיבית הסתברותית

Logistic Regression

$$p_{y|x}(y|x) = \text{softmax}(F(\mathbf{x}; \boldsymbol{\theta}))_y = \frac{e^{f_y(\mathbf{x}; \boldsymbol{\theta}_y)}}{\sum_c e^{f_c(\mathbf{x}; \boldsymbol{\theta}_c)}}$$

כאשר F היא פונקציה שמאגדת לוקטור C פונקציות פרמטריות $(f_1(\mathbf{x}; \boldsymbol{\theta}_1), f_2(\mathbf{x}; \boldsymbol{\theta}_2), \dots, f_C(\mathbf{x}; \boldsymbol{\theta}_C))^T$ את הפרמטרים מוצאים בעזרת MLE ו gradient descent.

המקרה הבינארי

במקרה הבינארי ניתן להשתמש במודל:

$$p_{y|x}(1|x) = \sigma(f(\mathbf{x}; \boldsymbol{\theta}))$$
$$p_{y|x}(0|x) = 1 - \sigma(f(\mathbf{x}; \boldsymbol{\theta}))$$

כאשר $\sigma(x) = \frac{1}{1+e^{-x}}$ (סיגמואיד).

תכונת של סיגמואיד ו softmax

$$\begin{aligned} \sigma(-z) &= 1 - \sigma(z) \bullet \\ \frac{\partial}{\partial z} \log(\sigma(z)) &= 1 - \sigma(z) \bullet \\ \frac{\partial}{\partial z_j} \log(\text{softmax}(\mathbf{z}))_i &= \underbrace{\delta_{i,j}}_{=I\{i=j\}} - \text{softmax}(\mathbf{z})_j \bullet \end{aligned}$$

Gradient Descent

בעבור בעיית המינימיזציה:

$$\arg \min_{\boldsymbol{\theta}} g(\boldsymbol{\theta})$$

• מאתחלים את $\boldsymbol{\theta}^{(0)}$ בנקודה אקראית כל שהיא.

• חוזרים על צעד העדכון הבא עד שמתקיים תנאי עצירה כל שהוא:

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta \nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta}^{(t)})$$

Bagging

מקטין את ה variance לרוב בלי לפגוע הרבה ב bias.

1. ניצור p מדגמים מתוך המדגם המקורי על ידי דגימה אקראית של דגימות ממנו.
2. בעבור כל מדגם חדש נבנה חזאי.
3. החזאי הכולל יהיה קומבינציה/בחירת הרוב של כל החזאים שבנינו:

$$h(\mathbf{x}) = \frac{1}{p} \sum_{i=1}^p \tilde{h}_i(\mathbf{x}) \quad \circ \quad \text{בעבור רגרסיה:}$$
$$h(\mathbf{x}) = \text{majority}(\{\tilde{h}_1(\mathbf{x}), \tilde{h}_2(\mathbf{x}), \dots, \tilde{h}_p(\mathbf{x})\}) \quad \circ \quad \text{בעבור סיווג:}$$

AdaBoost

בעבור סיווג בינארי מקטין את ה bias, לרוב ללא פגיעה גדולה ב variance.

אלגוריתם

ב $t = 0$ נאתחל וקטור משקלים $w_i^{(t)} = \frac{1}{N}$. בכל צעד t נבצע את הפעולות הבאות:

1. נבחר את המסווג אשר ממזער את ה misclassification rate הממושקל:

$$\tilde{h}_t = \arg \min_{\tilde{h}} \sum_{i=1}^N w_i^{(t-1)} I\{y^{(i)} \neq \tilde{h}(\mathbf{x}^{(i)})\}$$

2. נחשב את המקדם α_{t+1} של המסווג:

$$\varepsilon_t = \sum_{i=1}^N w_i^{(t-1)} I\{y^{(i)} \neq \tilde{h}_t(\mathbf{x}^{(i)})\}$$
$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \varepsilon_t}{\varepsilon_t} \right)$$

3. נעדכן את וקטור המשקלים:

$$\tilde{w}_i^{(t)} = w_i^{(t-1)} \exp \left(-\alpha_t y^{(i)} \tilde{h}_t(\mathbf{x}^{(i)}) \right)$$
$$w_i^{(t)} = \frac{\tilde{w}_i^{(t)}}{\sum_{j=1}^N \tilde{w}_j^{(t)}}$$

המסווג הסופי

הסיווג הסופי נעשה על ידי קומבינציה לינארית של כל מסווגים והמשקל שלהם.

$$h(\mathbf{x}) = \text{sign} \left(\sum_{t=1}^T \alpha_t \tilde{h}_t(\mathbf{x}) \right)$$

חסם

ניתן להראות שאם מתקיים שבכל צעד שגיאת ה misclassification error הממושקלת קטנה מ $\frac{1}{2} - \gamma_t$ אז ניתן לחסום את שגיאת ה misclassification rate על המדגם באופן הבא:

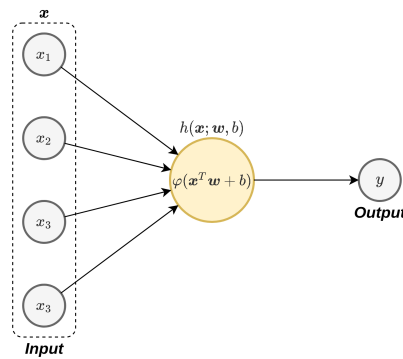
$$\frac{1}{N} \sum_i I\{h(\mathbf{x}^{(i)}) \neq y^{(i)}\} \leq \frac{1}{N} \sum_{i=1}^N \exp \left(-\sum_{t=1}^T \alpha_t y^{(i)} \tilde{h}_t(\mathbf{x}^{(i)}) \right) \leq \exp \left(-2 \sum_{t=1}^T \gamma_t^2 \right)$$

רשתות נוירונים

שיטה לבניית מודלים פרמטריים בעלי יכולת ייצוג גבוה בהשראת רשתות נוירונים ביולוגיות. רשתות נוירונים מתקבלות על ידי שירשור של מספר נוירונים כאשר כל נוירון מבצע פעולה מטמית פשוטה.

נוירון בודד

כל נוירון ברשת מבצע את הפעולה הבאה:



כאשר w ו b הם הפרמטרים של הנוירון ו $\varphi(\cdot)$ היא פונקציה שאותה יש לקבוע מראש והיא מכונה **פונקציית ההפעלה (activation function)**. בחירות נפוצות של פונקציית ההפעלה כוללות את

- הפונקציה הלוגיסטית (סיגמואיד): $\varphi(x) = \sigma(x) = \frac{1}{1+e^{-x}}$
- טנגנס היפרבולי: $\varphi(x) = \tanh(x/2)$

- פונקציית ה ReLU (Rectified Linear Unit): אשר מוגדרת $\varphi(x) = \max(x, 0)$ (זוהי פונקציית ההפעלה נפוצה ביותר כיום).

רשתות נוירונים

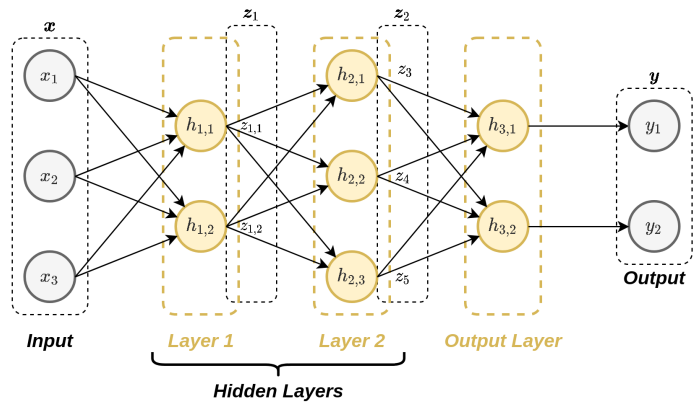
מושגים:

- **ארכיטקטורה:** המבנה של הרשת (האופן בו הנוירונים מחוברים)
- **יחידות נסתרות (hidden units):** הנוירונים אשר אינם מחוברים למוצא הרשת (אינם נמצאים בסוף הרשת).
- **רשת עמוקה (deep network):** רשת אשר מכילה מסלולים מהכניסה למוצא אשר עוברים דרך יותר מיחידה נסתרת אחת.

(Multi-Layer Perceptron (MLP

רשת שבה:

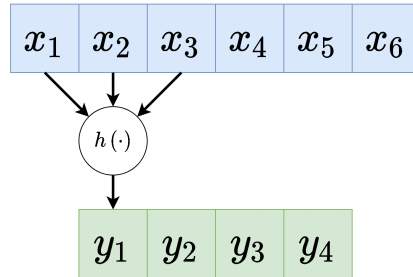
- הנוירונים מסודרים בשתיים או יותר שכבות (layers).
- השכבות הם **Fully Connected (FC) layers** (כל נוירון מוזן מכל הנוירונים שבשכבה שלפניו).



Convolutional Neural Network (CNN)

רשתות אשר מכילות שכבות קונבולוציה. שכבת קונבולוציה נבדלת משכבת FC בשני מובנים:

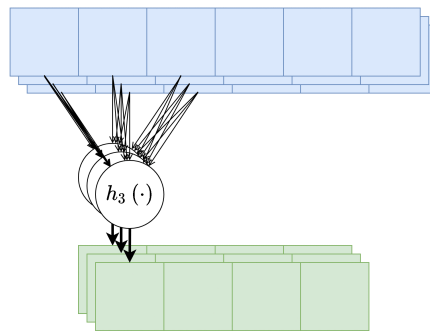
1. כל נירון בשכבה זו מוזן רק מכמות מוגבלת של ערכים הנמצאים בסביבתו הקרובה.
2. כל הנירונים בשכבה מסוימת זהים, זאת אומרת שהם משתמשים באותם הפרמטרים (תכונה המכונה **weight sharing**).



וקטור המשקלים אשר מכפיל את הערכים בכניסה לנירונים בשכבת הקונבולוציה נראה גרעין הקונבולוציה.

שכבות קונבולוציה עם מספר ערצים

שכבת קונבולוציה תכיל לרוב מספר ערצים בכניסה ומספר ערצים ביציאה. במקרה זה יהיה נירון שונה (עם גרעין קונבולוציה שונה) בעבור כל ערוץ ביציאה וכל אחד מהנירונים יפעל על כל ערוצי הכניסה

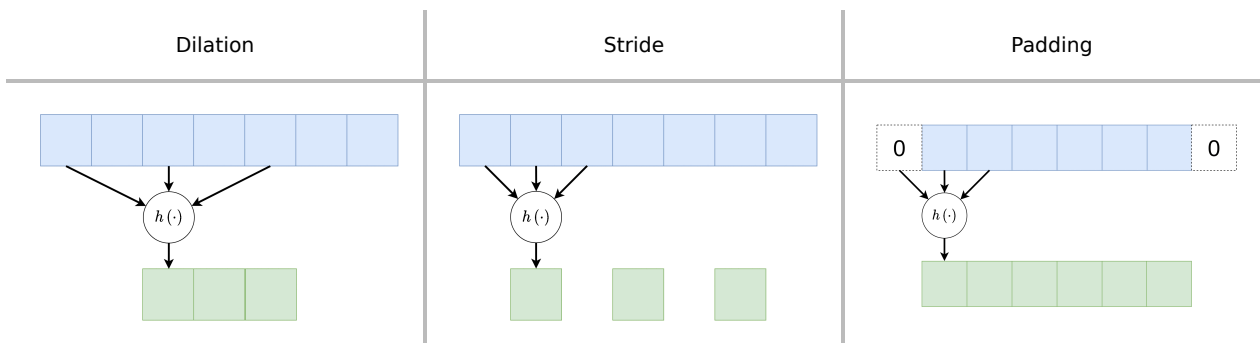


$$\underbrace{C_{in} \times C_{out} \times K}_{\text{the weights}} + \underbrace{C_{out}}_{\text{the bias}}$$

כאשר:

- C_{in} - מספר ערוצי קלט.
- C_{out} - מספר ערוצי פלט.
- K - גודל הגרעין.

תכונות נוספות

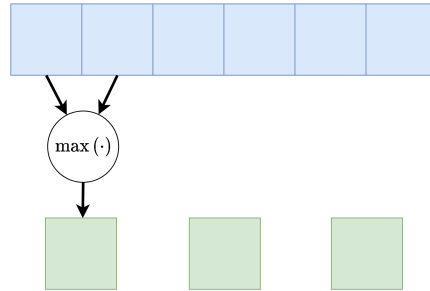


שכבות Pooling

שכבות אשר משמשות להקטנת המספר האיברים שעליהם הרשת עובדת. שתי שכבות pooling נפוצות הינן:

1. **Average pooling**: אשר פועל כל פעם על חלון מסויים ומחזיר את הממוצע שלו.
2. **Max pooling**: אשר פועל כל פעם על חלון מסויים ומחזיר את הערך המקסימאלי בחלון הנתון.

שכבות pooling מוגדרות על ידי שני פרמטרים, גודל החלון וה $stride$ (גודל הצעד שבו הם זזות). (לרוב ה- $stride$ יהיה זהה לגודל החלון).



בשכבה זאת אין פרמטרים נלמדים.

Unsupervised Learning

PCA

PCA הוא אלגוריתם לינארי להורדת המימד של הוקטור \mathbf{x} לוקטור קצר יותר באורך K , אשר ממזער את שיגאת השיחזור הריבועית הממוצעת:

- Encoding (קידוד): $\mathbf{z} = \mathbf{T}^\top (\mathbf{x} - \bar{\mathbf{x}})$
- Decoding (שיחזור): $\tilde{\mathbf{x}} = \mathbf{T}\mathbf{z} + \bar{\mathbf{x}}$

כאשר:

- $\bar{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}^{(i)}$
- \mathbf{X} היא מטריצת המדידות אשר מוגדרת: $\mathbf{X} = (\mathbf{x}^{(1)} - \bar{\mathbf{x}}, \mathbf{x}^{(2)} - \bar{\mathbf{x}}, \dots, \mathbf{x}^{(N)} - \bar{\mathbf{x}})^\top$
- \mathbf{T} היא מטריצה אשר עמודותיה הן K הוקטורים העצמיים של המטריצה $\mathbf{X}^\top \mathbf{X}$ אשר מתאימים לערכים העצמיים הגדולים ביותר.

K-Means

אלגוריתם אשכול אשר מבצע אישכול ל K קבוצות. סימונים:

- K - מספר האשכולות (גודל אשר נקבע מראש).
- \mathcal{I}_k - אוסף האינדקסים של האשכול ה- k . לדוגמא: $\mathcal{I}_5 = \{3, 6, 9, 13\}$
- $|\mathcal{I}_k|$ - גודל האשכול ה- k (מספר הפרטים בקבוצה)
- $\{\mathcal{I}_k\}_{k=1}^K$ - חלוקה מסויימת לאשכולות

K-means מנסה לפתור את בעיית האופטימיזציה הבאה:

$$\arg \min_{\{\mathcal{I}_j\}_{j=1}^K} \frac{1}{N} \sum_{k=1}^K \frac{1}{2|\mathcal{I}_k|} \sum_{i,j \in \mathcal{I}_k} \|\mathbf{x}^{(j)} - \mathbf{x}^{(i)}\|_2^2 = \arg \min_{\{\mathcal{I}_j\}_{j=1}^K} \frac{1}{N} \sum_{k=1}^K \sum_{i \in \mathcal{I}_k} \|\mathbf{x}^{(i)} - \boldsymbol{\mu}_k\|_2^2$$

כאשר:

$$\boldsymbol{\mu}_k = \frac{1}{|\mathcal{I}_k|} \sum_{i \in \mathcal{I}_k} \mathbf{x}^{(i)}$$

האלגוריתם

האלגוריתם מאותחל בצעד $t = 0$ על ידי בחירה אקראית של K מרכזי מסה: $\{\boldsymbol{\mu}_k\}_{k=1}^K$.

בכל צעד t מבצעים את שתי הפעולות הבאות:

2. עדכון של מרכזי המסה המסה על פי:

$$\boldsymbol{\mu}_k = \frac{1}{|\mathcal{I}_k|} \sum_{i \in \mathcal{I}_k} \mathbf{x}^{(i)}$$

(אם $|\mathcal{I}_k| = 0$ אז משאירים אותו ללא שינוי)

1. עדכון מחדש את החלוקה לאשכולות $\{\mathcal{I}_k\}_{k=1}^K$ כך שכל דגימה משוייכת למרכז המסה הקרוב עליה ביותר. כלומר אנו נשייך את כל דגימה \mathbf{x} לפי:

$$k = \arg \min_{k \in [1, K]} \|\mathbf{x} - \boldsymbol{\mu}_k\|_2^2$$

(במקרה של שני מרכזים במרחק זהה נבחר בזה בעל האינדקס הנמוך יותר).

תנאי העצירה של האלגוריתם הינו כשהאשכולות מפסיקים להשתנות. אחת הדרכים הנפוצות לאיתחול של $\{\boldsymbol{\mu}_k\}_{k=1}^K$ היא לבחור k נקודות מתוך המדגם.