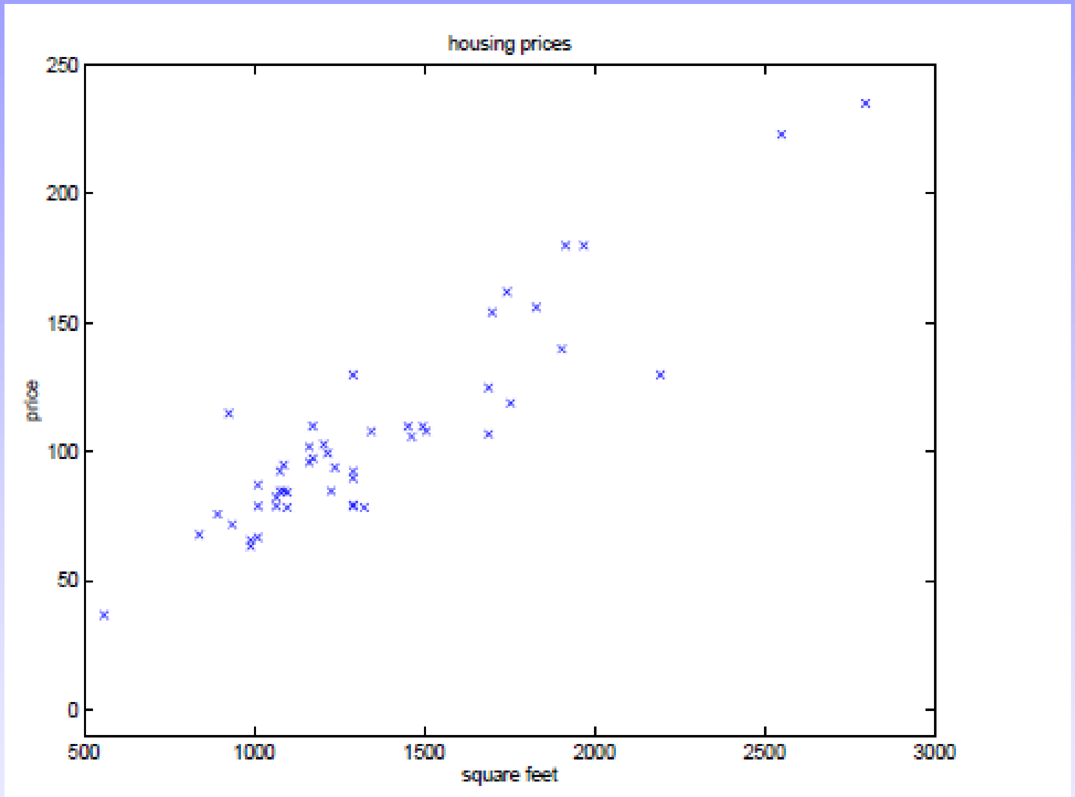
רגרסיה לינארית (Linear Regression)

- בעיית למידה מודרכת.
- נניח שיש קבוצת נתונים של שטח המגורים ומחיר הבית ב- 47 בתים.

Living area ($feet^2$)	Price (1000$s)
560	37
1012	79
893	76
2196	130
936	72
⋮	⋮

רגרסיה לינארית (Linear Regression)

אפשר לצייר את הנתונים:



האם אפשר לחזות את המחיר של בתים אחרים כפונקציה של שטח המגורים שלהם בהינתן נתונים אלה?

רגרסיה לינארית (Linear Regression)

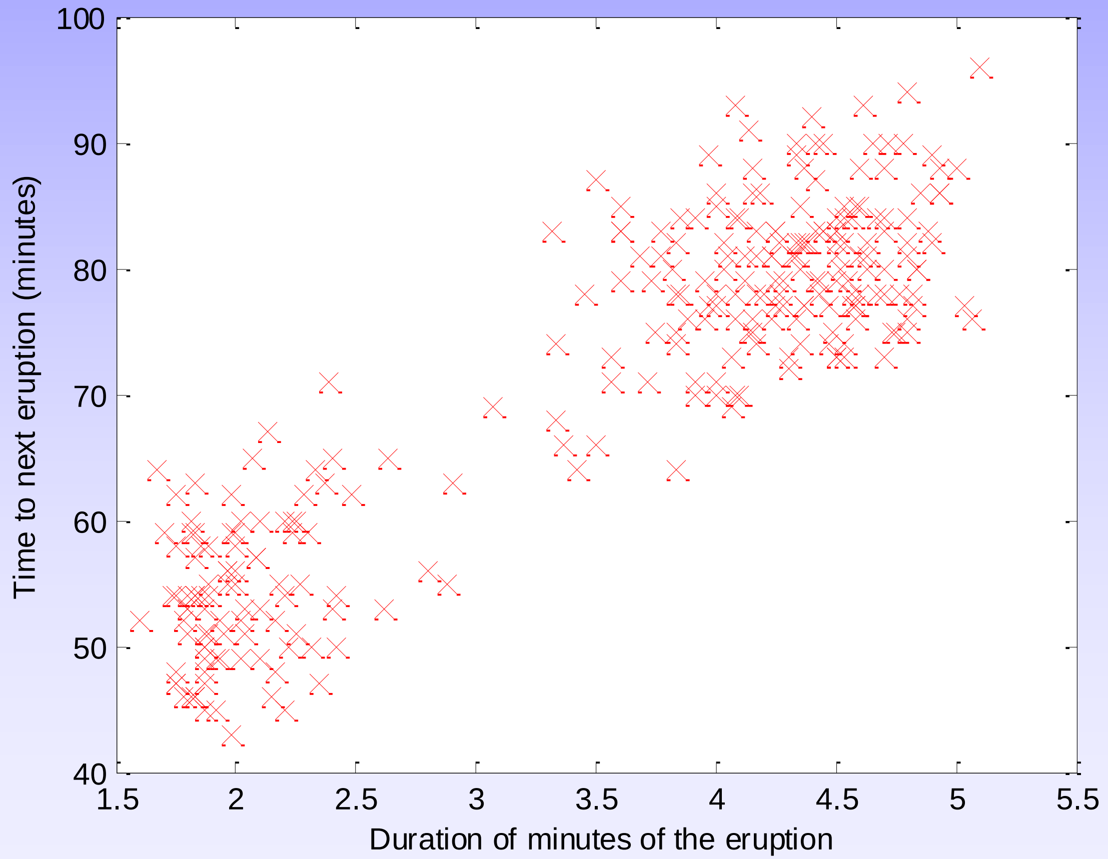
- בעייה נוספת: **הגייזר הנאמן (old faithful)| הוא תופעת טבע**

באגן הגייזרים העילי שבפארק ילוסטון, ארה"ב, שנחשב לגייזר המפורסם בעולם. שמו ניתן לו מכיוון שניתן לחזות את הזמן עד להתפרצות הבאה בדיוק של עד 10 דקות. הסיבה לכך היא כמויות מים קבועות שזורמות במרווחי זמן קבועים לתוך מאגרי המים של הגייזר.

- בהתפרצויותיו, הגייזר פולט בין כ־14,000-36,000 ליטרים של מים לגובה של עד 35 מ' למשך של דקה וחצי ועד לחמש דקות.
- מרווח הזמן בין התפרצויותיו של הגייזר הוא כ־65 עד כ־92 דקות כשהממוצע הוא 91 דקות. בהתאם למשך ההתפרצות האחרונה, ניתן להעריך את מועד ההתפרצות הבאה, עד כדי סטייה של כ־10 דקות.
- https://www.youtube.com/watch?v=wE8NDuzt8eg

רגרסיה לינארית (Linear Regression)

- קבוצת הנתונים כאן היא 272 נקודות, המתארות את פרק הזמן עד להתפרצות הבאה, כפונקציה של משך ההתפרצות הנוכחית.



רגרסיה לינארית (Linear Regression)

נסמן את קבוצת האימון:

$x^{(i)}$ - משתני הקלט, תכונות הקלט.

$y^{(i)}$ - משתנה הפלט או המטרה, אותו אנו רוצים לחזות (מחיר הבית, הזמן עד להתפרצות הבאה).

הזוג $\left(x^{(i)}, y^{(i)}\right)$ נקרא דוגמת אימון.

קבוצת m דוגמאות האימון, קבוצת הנתונים על-פיהם אנו רוצים ללמוד, $\left\{\left(x^{(i)}, y^{(i)}\right); \quad i = 1, 2, \ldots, m\right\}$ נקראת קבוצת האימון

(i) – בסימון הנ"ל הוא פשוט אינדקס המציין את הסדר בקבוצת האימון.

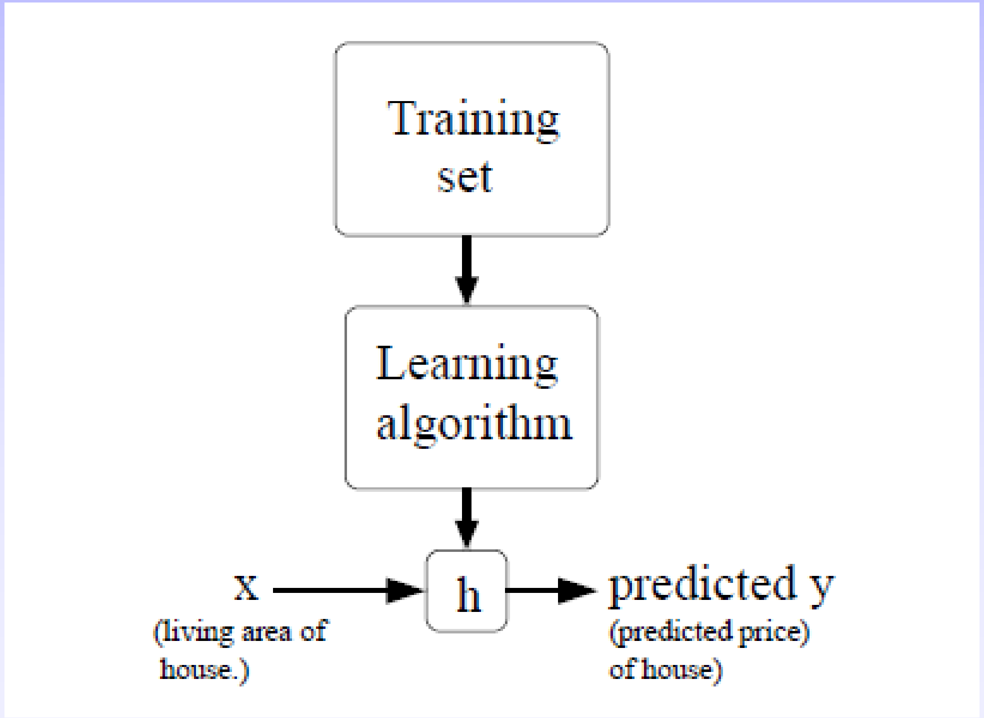
X – המרחב של ערכי הקלט, y – המרחב של ערכי הפלט.

בדוגמאות אלה: $X, y \in R$

רגרסיה לינארית (Linear Regression)

באופן יותר פורמלי – המטרה שלנו, בהינתן קבוצת האימון, ללמוד פונקציה $h: X \rightarrow y$ כך ש- $h(x)$ יהיה חזאי טוב לערך המתאים של y.

מסיבות היסטוריות הפונקציה $h(x)$ נקראת **השערה** (hypothesis).



רגרסיה לינארית (Linear Regression)

- כאשר משתנה המטרה הוא רציף כמו בדוגמאות הקודמות, בעיית הלמידה נקראת בעיית רגרסיה.

- כאשר y יכול לקבל רק מספר קטן של ערכים בדידים הבעייה נקראת בעיית סווג (Classification problem)

רגרסיה לינארית (Linear Regression)

כדי להפוך את הבעייה ליותר מעניינת, נניח שקבוצת האימון עשירה יותר ואנו יודעים גם את מספר חדרי השינה בכל בית:

Living area (feet^2)	#bedrooms	Price (1000\$s)
560	2	37
1012	3	79
893	3	76
2196	4	130
936	3	72
⋮	⋮	⋮

כאן משתנה ה- x הוא דו-מימדי- כלומר וקטורים דו-מימדיים ב- R^2

רגרסיה לינארית (Linear Regression)

$x_1^{(i)}$ השטח של הבית ה- i.

$x_2^{(i)}$ - מספר חדרי השינה בבית ה- i

הערה: אילו תכונות ומהו מספרן ניתנים לבחירה על-ידי המשתמש.

נניח כרגע שהתכונות ידועות.

כדי לבצע למידה מודרכת צריך להחליט כיצד נייצג את ההשערה h (או הפונקציה h) באמצעות המחשב.

בחירה התחלתית: נקרב את y על-ידי פונקציה לינארית של x.

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

θ – פרמטרים או משקלות, מבצעים פרמטריזציה של מרחב הפונקציות מ- X ל- Y.

רגרסיה לינארית (Linear Regression)

פונקציה לינארית של x:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

כדי לפשט עוד יותר: נציג את x_0 בתור 1, כך ש:

$$h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n = \sum_{i=0}^{n} \theta_i x_i = \theta^T x$$

כאשר θ ו- x הם וקטורים, ו- n הוא מספר משתני הכניסה (מבלי למנות את x_0)

רגרסיה לינארית (Linear Regression)

עתה, בהינתן קבוצת האימון, כיצד נבחר או נלמד את הפרמטרים θ?

$$h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n = \sum_{i=0}^{n} \theta_i x_i = \theta^T x$$

נציע שיטה הגיונית: נבחר את הפרמטרים θ כך ש- $h_\theta(x)$ יהיה קרוב ככל האפשר ל- y, לפחות עבור דוגמאות האימון אותן יש לנו.

באופן פורמלי נגדיר פונקציה המודדת לכל ערך של הוקטור θ כמה קרובים ערכי ה- $h_\theta(x^{(i)})$ לערכי $y^{(i)}$ המתאימים.

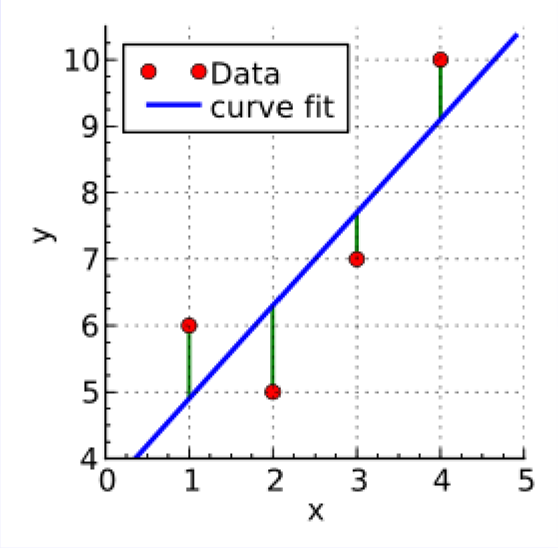
נגדיר פונקציית מחיר:

רגרסיה לינארית (Linear Regression)

נגדיר פונקציית מחיר:

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

הפונקציה נקראת **פונקציית מחיר בריבועים פחותים** של מודל הרגרסיה (Least-square cost function).



אלגוריתם ה- LMS (Least Mean Square)

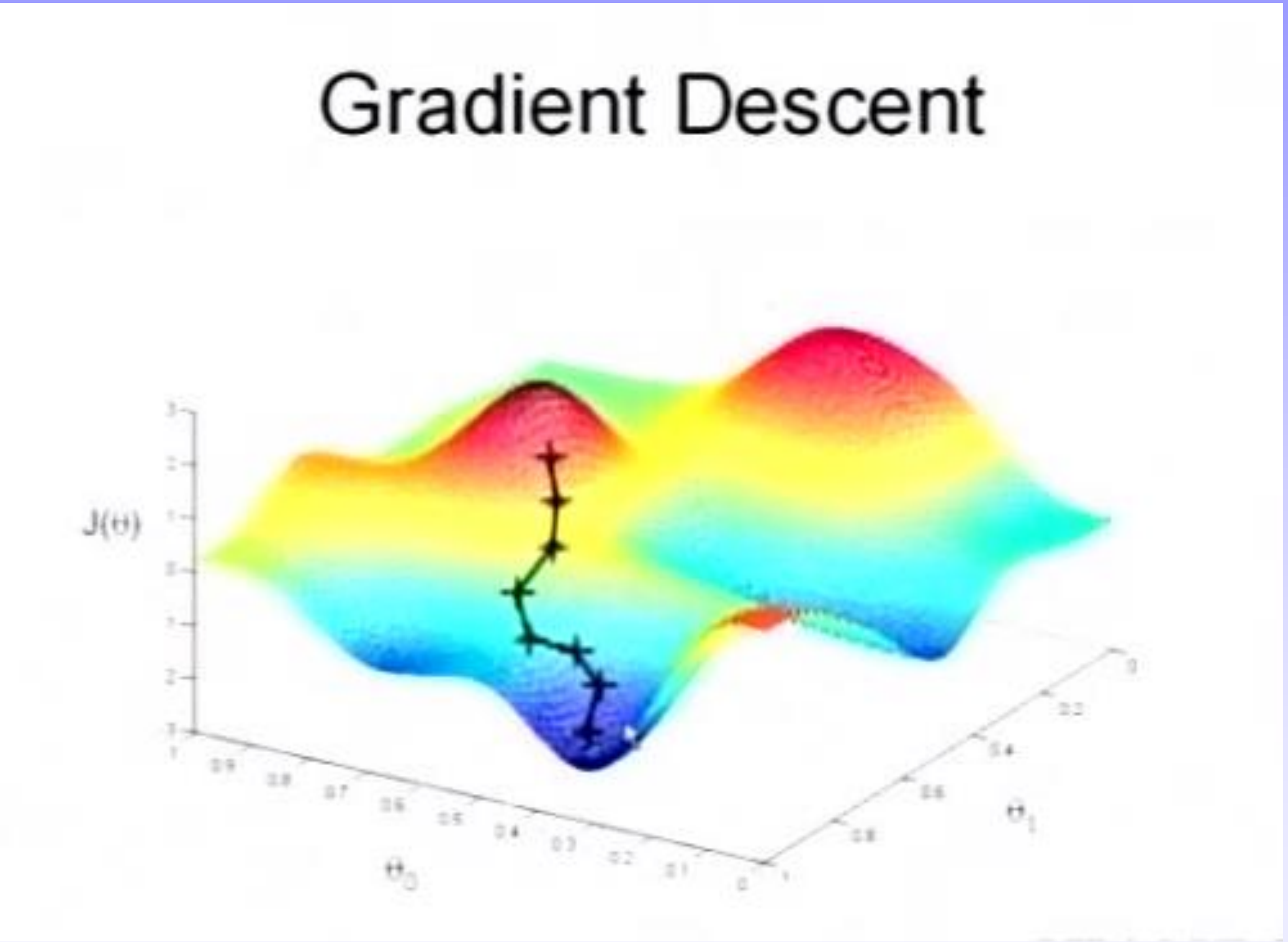
רוצים לבחור ערך θ כך שימזער את $J(\theta)$.
כדי לעשות זאת נתחיל עם ניחוש התחלתי עבור θ, ונשנה
את θ בכל חזרה, כדי להפוך את $J(\theta)$ ליותר קטן, עד שלבסוף
נתכנס לערך של θ שימזער את $J(\theta)$.

באופן ספיציפי נתבונן באלגוריתם ה- gradient descent.
אלגוריתם זה מתחיל ב- θ התחלתי כלשהו ומבצע באופן חוזר ונשנה
את העדכון:

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

(העדכון נעשה באופן סימולטני לכל ערכי $j=0,1,2,\ldots,n$)

אלגוריתם ה- Gradient descent



אלגוריתם ה- Gradient descent

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

α- נקרא קצב הלמידה.

זהו אלגוריתם המבצע באופן איטרטיבי צעד לכוון הירידה התלול ביותר של J

כדי לממש את האלגוריתם צריך לבחון מהי הנגזרת החלקית בצד ימין.

נניח שיש רק דוגמת אימון אחת (x,y).

אפשר לפיכך להתעלם מהסכום בהגדרה של J .

אלגוריתם ה- Gradient descent

הנחה : קיימת רק דוגמת אימון אחת (x,y)

$$\frac{\partial}{\partial\theta_j}J(\theta)=\frac{\partial}{\partial\theta_j}\frac{1}{2}(h_\theta(x)-y)^2=$$

$$=2\frac{1}{2}(h_\theta(x)-y)\frac{\partial}{\partial\theta_j}(h_\theta(x)-y)=$$

$$(h_\theta(x)-y)\frac{\partial}{\partial\theta_j}(\sum_{i=0}^{n}\theta_i x_i-y)=$$

$$(h_\theta(x)-y)x_j$$

אלגוריתם ה- Gradient descent

עבור דוגמת אימון יחידה מתקבל כלל העדכון הבא:

$$\theta_j := \theta_j + (y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)}$$

כלל זה נקרא **כלל עדכון ה- LMS** או כלל העדכון לפי **שערוך בריבועים פחותים** (Least mean square).
הכלל ידוע גם בתור **כלל הלמידה Widrow-Hoff**.
(באופן אינטואיטיבי – גודל העדכון פרופורציונלי לאיבר השגיאה $(y^{(i)} - h_\theta(x^{(i)}))$

אם נתקלים בדוגמא עבורה החיזוי כמעט מתאים לערך של $y^{(i)}$ יש צורך מועט לשנות את הפרמטרים – שגיאה קטנה.
עבור שגיאה גדולה - הערך של החיזוי $h_\theta(x^{(i)})$ רחוק מאוד מ- $y^{(i)}$ - נדרש שינוי גדול בפרמטרים.

אלגוריתם ה- Gradient descent

קיימות שתי דרכים לבצע את השיטה עבור יותר מדוגמא אחת:

1. Batch gradient descent: השיטה עובדת על כל אחת מהדוגמאות עבור כל קבוצת האימון בכל צעד.

Repeat until convergence {

$$\theta_j := \theta_j + \sum_{i=1}^{m} (y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)}$$

}

ניתן לראות כי הגודל בסכימה בכלל העדכון הנ"ל הוא: $\frac{\partial}{\partial \theta_j} J(\theta)$ כאשר J היא ההגדרה המקורית של פונקציית המחיר

אלגוריתם ה- Gradient descent

2. Incremental or stochastic gradient descent –

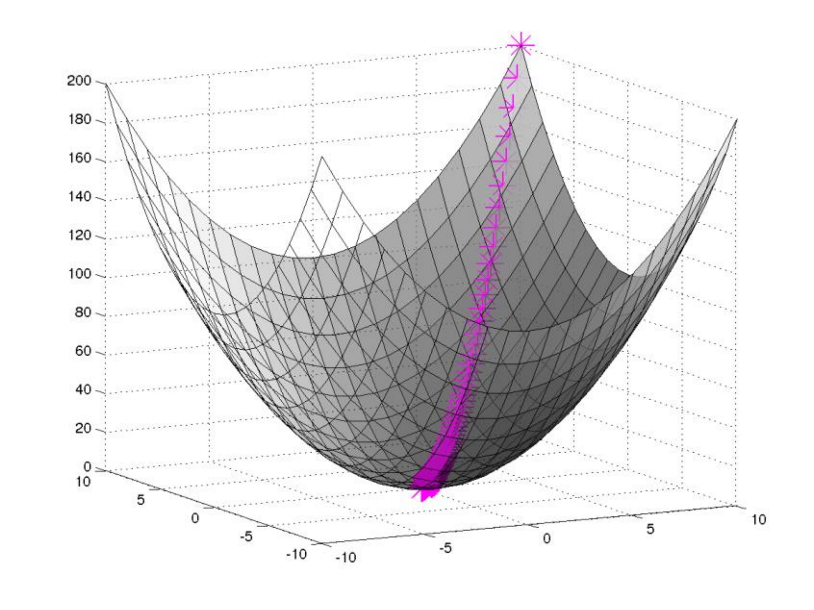
השיטה עוברת בכל צעד על דוגמת אימון אחת בלבד, ומעדכנת את וקטור הפרמטרים בהתאם לשגיאה:

$$
\begin{aligned}
&Loop\ \{\\
&\quad for\ i = 1\ to\ m\ \{\\
&\quad\quad \theta_j := \theta_j + (y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)}, \quad \forall j\\
&\quad \}\\
&\}
\end{aligned}
$$

- האלגוריתם מתעדכן מייד ולא רק אחרי מעבר על m דוגמאות האימון – יתרון כאשר m הוא גדול.
- בדרך-כלל מגיע קרוב יותר למינימום הרבה יותר מהר מאשר גירסת ה- batch.
- ייתכן שלא תהיה התכנסות למינימום, והפרמטרים θ יתנדנדו סביב המינימום של J(θ), אבל באופן מעשי ערכים סביב המינימום יהיו קרובים מספיק למינימום.

אלגוריתם ה- Gradient descent

הערה: שיטת ה- gradient descent עשויה להיות רגישה למינימה מקומיות (local minima),
באופן כללי עבור בעיית האופטימיזציה המתוארת כאן יש רק מינימום גלובלי אחד. הפונקציה J היא פונקציה קוודרטית קמורה.לפיכך אם α לא יותר מדי גדולה, ה- gradient descent תמיד מתכנס למינימום גלובלי.



ביטוי סגור למזעור פונקציית המחיר J(θ)

אלגוריתם ה- gradient descent מספק דרך אחת למזעור J.
נדון עתה בדרך נוספת המאפשרת למזער את J ללא שימוש באלגוריתם איטרטיבי, אלא באופן ישיר באמצעות חישוב הנגזרת של J והשוואתה לאפס.
לצורך כך וכדי להימנע מכתיבה רבה,נגדיר נגזרות מטריציות:
נניח כי $f : R^{mxn} \to R$ היא פונקציה הממפה מטריצות mxn לישר הממשי, ונגדיר:

$$\nabla_A f(A) = \begin{bmatrix} \frac{\partial f}{\partial A_{11}} & .. & \frac{\partial f}{\partial A_{1n}} \\ : & : & : \\ \frac{\partial f}{\partial A_{m1}} & .. & \frac{\partial f}{\partial A_{mn}} \end{bmatrix}$$

כלומר $\nabla_A f(A)$ זוהי מטריצה mxn, שהאיבר ה- (i,j) שלה הוא $\frac{\partial f}{\partial A_{ij}}$

ביטוי סגור למזעור פונקציית המחיר J(θ)

העקבה של מטריצה ריבועית : A_{nxn}
מוגדרת כסכום הערכים של האלכסון הראשי

$$tr(A) = \sum_{i=1}^{n} a_{ii}$$

כלומר גם זו פונקציה הממפה מטריצות nxn לישר הממשי.
מספר תכונות של אופרטור העקבה (trace):

$$tr(AB) = tr(BA)$$

$$tr(ABC) = tr(CAB) = tr(BCA)$$

$$tr(ABCD) = tr(DABC) = tr(CDAB) = tr(BCDA)$$

$$tr(A) = tr(A^T)$$

$$tr(A+B) = tr(A) + tr(B)$$

$$tr(\alpha A) = \alpha \cdot tr(A)$$

ביטוי סגור למזעור פונקציית המחיר J(θ)

מספר עובדות על **נגזרות מטריצות** בהן נשתמש בהמשך:

$$1)\ \nabla tr(AB) = B^T$$

$$2)\ \nabla_{A^T} f(A) = (\nabla_A f(A))^T$$

$$3)\ \nabla_A tr(ABA^T C) = CAB + C^T AB^T$$

מתוך 2) ו- 3) אפשר לקבל את הכלל הבא:

$$\nabla_A tr(ABA^T C) = B^T A^T C^T + BA^T C$$

ריבועים פחותים במבט נוסף

מצויידים בכלים של נגזרות מטריצות, נמשיך עתה למצוא ביטוי סגור לערך של θ הממזער את $J(\theta)$:

נתחיל בכתיבת J בסימון מטריצי וקטורי:

נגדיר את המטריצה X:

$$X = \begin{pmatrix} -(x^{(1)})^T - \\ -(x^{(2)})^T - \\ \vdots \\ -(x^{(m)})^T - \end{pmatrix}$$

כלומר כל שורה היא דוגמת אימון, כאשר $X_{m \times (n+1)}$ מכילה את דוגמאות האימון בשורות, לדוגמא בשורה הראשונה וקטור התכונות של דוגמת האימון הראשונה וכו'.

ריבועים פחותים במבט נוסף

וכן וקטור ערכי המטרה (התיוגים)

$$y = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{pmatrix}$$

עתה, מאחר ו-

$$h_\theta(x^{(i)}) = (x^{(i)})^T \cdot \theta =$$

$$x_0^{(i)}\theta_0 + x_1^{(i)}\theta_1 + \ldots + x_n^{(i)}\theta_n$$

ניתן להראות בקלות כי:

$$X\theta - y = \begin{pmatrix} (x^{(1)})^T\theta \\ (x^{(2)})^T\theta \\ \vdots \\ (x^{(m)})^T\theta \end{pmatrix} - \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{pmatrix} = \begin{pmatrix} h_\theta(x^{(1)}) - y^{(1)} \\ h_\theta(x^{(2)}) - y^{(2)} \\ \vdots \\ h_\theta(x^{(m)}) - y^{(m)} \end{pmatrix}$$

ריבועים פחותים במבט נוסף

נשתמש בעובדה כי לכל וקטור z

$$z^T z = \sum_i z_i^2$$

ולכן:

$$\frac{1}{2}(X\theta - y)^T (X\theta - y) = \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 = J(\theta)$$

כדי למזער את J נמצא את הנגזרת שלו ביחס ל- θ:

$$\nabla_\theta J(\theta) = \nabla_\theta \frac{1}{2}(X\theta - y)^T (X\theta - y)$$

ריבועים פחותים במבט נוסף

$$\nabla_\theta J(\theta) = \nabla_\theta \frac{1}{2}(X\theta - y)^T(X\theta - y)$$

$$= \frac{1}{2}\nabla_\theta(\theta^T X^T X\theta - \theta^T X^T y - y^T X\theta + y^T y)$$

$$= \frac{1}{2}\nabla_\theta tr(\theta^T X^T X\theta - \theta^T X^T y - y^T X\theta + y^T y)$$

הסבר: tr(real number)=real number

$$= \frac{1}{2}\nabla_\theta(tr(\theta^T X^T X\theta) - 2tr(y^T X\theta))$$

הסבר: $tr(A)=tr(A^T)$ וכן השתמשנו ב –

$$\nabla_A tr(ABA^T C) = B^T A^T C^T + BA^T C$$

כאשר $A^T=\theta$, $B^T=B=X^TX$ ו- C=I.

$$= \frac{1}{2}(X^T X\theta + X^T X\theta - 2X^T y)$$

$$= X^T X\theta - X^T y$$

וכן: $\nabla_\theta tr(y^T X\theta) = y^T X$

ריבועים פחותים במבט נוסף

כדי למזער את $J(\theta)$ נשווה את הנגזרות ל- 0 ונקבל את מערכת **המשוואות הנורמליות**:

$$X^T X\theta = X^T y$$

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

רגרסיה לינארית מרובת משתנים

- הערות מימוש עבור אלגוריתם ה- Gradient Descent

1. קצב הלמידה α .

כזכור:

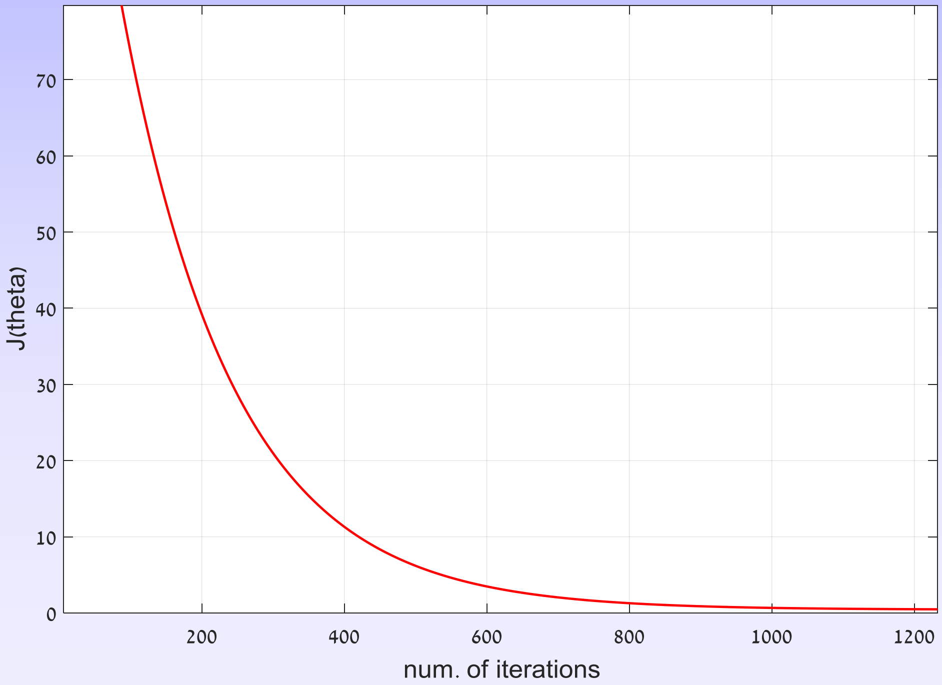
$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

במהלך ביצוע האלגוריתם נרצה לבצע debugging כדי לדאוג למהלך תקין של ה- gd.

חשיבות גדולה לבחירה נכונה של קצב הלמידה α

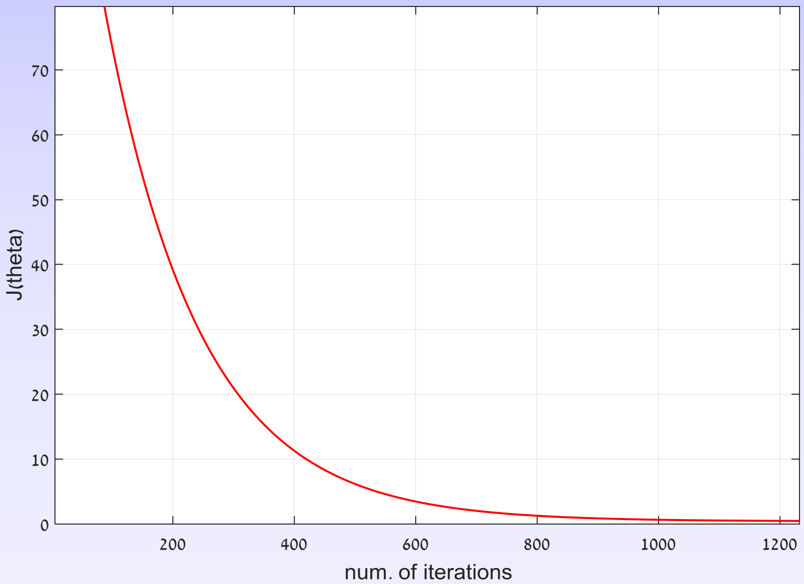
הערות מימוש עבור אלגוריתם ה- Gradient Descent

1. אחת הדרכים להבטיח עבודה נכונה של האלגוריתם:
   - ציור פונקציית העלות כתלות באיטרציות.
   - עבודה נכונה – $J(\theta)$ צריך לקטון בכל איטרציה.
   - ציור כזה יכול גם להצביע על מספר האיטרציות הדרוש כדי להגיע לערך הפרמטרים הרצוי.



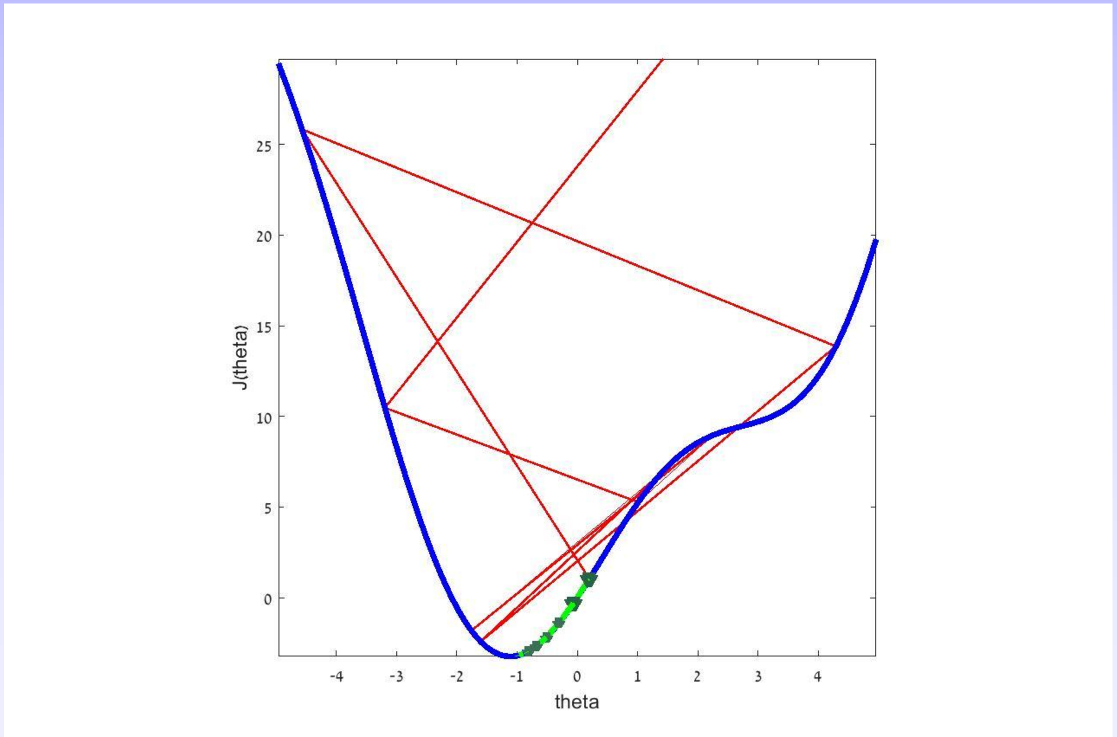
הערות מימוש עבור אלגוריתם ה- Gradient Descent

- מספר האיטרציות הדרושות תלוי ביישום ובבעייה, ויכול להשתנות במידה רבה.
- דרך לבדיקה – האם $J\left(\theta\right)$ קטן בפחות מערך סף מסויים בין שתי איטרציות.
- הבעייה – קשה לקבוע את ערך הסף, ולכן עדיף להתבונן בציור של ערך $J\left(\theta\right)$ כתלות במספר האיטרציות.



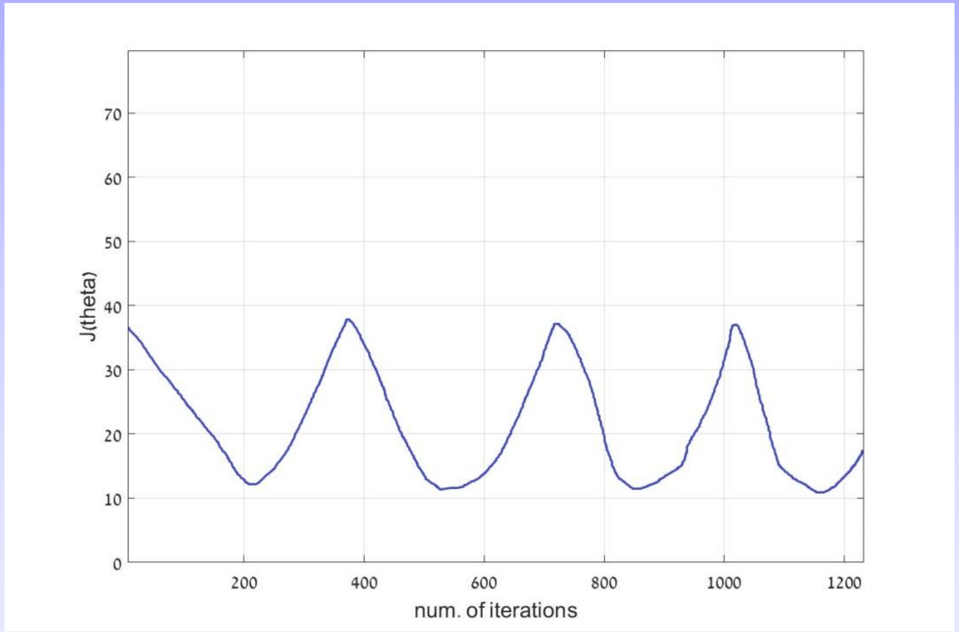
הערות מימוש עבור אלגוריתם ה- Gradient Descent

- בתהליך האופטימיזציה, עבור α לא מספיק קטן, עשויה להיווצר התבדרות **(-)** כפי שמודגם בציור,
- עבור ערך מתאים של α האלגוריתם מוצא את המינימום **(-)**.



הערות מימוש עבור אלגוריתם ה- Gradient Descent

- גם עבור המצב הבא הקטנת ערך α עשוי לפתור את הבעייה:



הערות מימוש עבור אלגוריתם ה- Gradient Descent

- אפשר להוכיח מתימטית שעבור ערך α מספיק קטן אלגוריתם ה- gd יורד בכל איטרציה, ומתכנס למינימום מקומי של J(θ).
- אם α יותר מדי קטן, ההתכנסות עלולה להיות מאוד אטית.

לסיכום:

- אם α גדול מדי - J(θ) לא יורד בכל איטרציה, עשוי לא להתכנס.
- אם α יותר מדי קטן – התכנסות אטית.
- כדי לבחור ערך α נכון – צריך לנסות תחום של ערכים כמו לדוגמא: α=1e-3, 5e-3,1e-2, 5e-2,1e-1,5e-1,1, 1e1 וכו', תוך בחינת הגרף של J(θ) כתלות ב- θ.