

# Gradient Descent



- **A LOT OF  
MATH**
- **SOME MORE  
MATH**



Gradient Descent (מורד הגרדיאנט) הוא  
אלגוריתם אופטימיזציה  
משמש למזעור פונקציית העלות.

מחשב את השיפוע של פונקציית העלות  
ו"צועד" \מעדכן את פרמטרי המודל בכיוון  
הירידה (ההפוך לשיפוע), כדי להתקרב בהדרגה  
לנקודת המינימום ולהגיע לפתרון האופטימלי

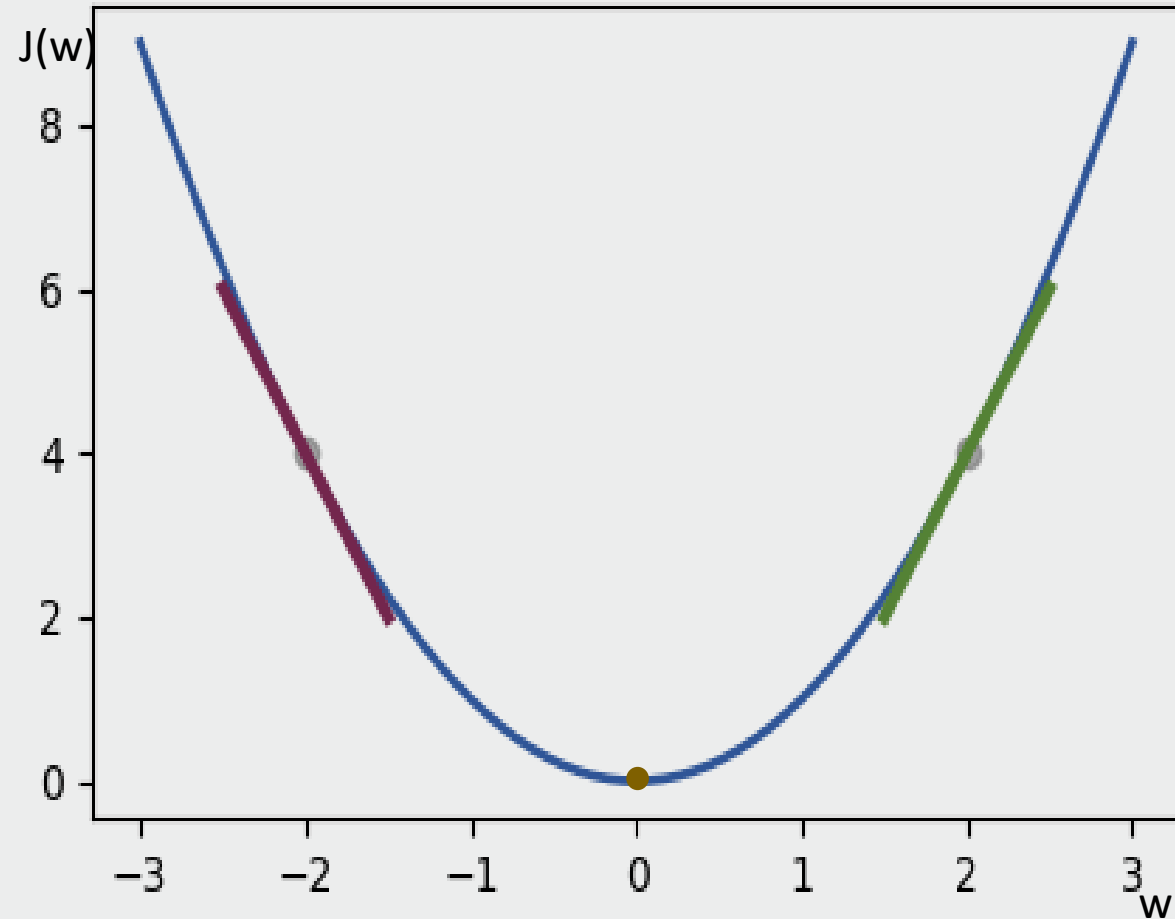
\*רלוונטי לתחום ה ML וה DL  
לא רק לרגרסיה לינארית

## גודל הצעד:

צעד קטן מידי יהיה לא יעיל.  
צעד גדול מידי יכול לפספס את  
נקודת המינימום



**כיוון הצעד:** ההתקדמות תמיד  
בירידה, במטרה להגיד למינימום.  
כיוון הצעד הוא בעצם שיפוע  
הפונקציה בנקודה.  
שיפוע הפונק' - זו נגזרת.



## מהי נגזרת?

**נגזרת** של פונקציה בנקודה מסוימת מתארת את **קצב השינוי** של הפונקציה באותה נקודה, או במילים אחרות, את **שיפוע המשיק** לגרף הפונקציה בנקודה זו.

## סימונים מקובלים

הנגזרת של הפונקציה  $f(x)$  מסומנת לעיתים ב  $f'(x)$  ולעיתים ב  $\frac{d}{dx}$

עקרונות גזירה בסיסיים

$$\frac{d}{dx}(x^n) = nx^{n-1}$$

$$\frac{d}{dx}(cx^n) = cnx^{n-1}$$

חוק שרשרת

$$\frac{d}{dx}f(g(x)) = f'(g(x))g'(x)$$

פעולות חשבון פשוטות בין וקטור וסקאלר:

(חיבור\חיסור\חילוק\כפל)

הפעולה מבוצעת על כל אחד מאברי הוקטור

התוצאה היא וקטור

$$\begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix} + C = \begin{bmatrix} a_1 + c \\ \vdots \\ a_n + c \end{bmatrix}$$

פעולות חשבון פשוטות בין שני וקטורים:

(חיבור\חיסור\חילוק\כפל)

הפעולה מבוצעת על פי מיקום בין אברי הוקטורים.

חשוב לשים לב להתאמה בצורה

התוצאה היא וקטור

$$\begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix} + \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} = \begin{bmatrix} a_1 + b_1 \\ \vdots \\ a_n + b_n \end{bmatrix}$$

$$\vec{a} \cdot \vec{b} = \sum_{i=1}^n a_i \times b_i$$

מכפלה סקאלרית : כפי שמשתמע משמה מניבה סקאלר.  
היא תוצאה של תוצר נקודה בין שני וקטורים  
התוצאה היא סכום סדרת מכפלות האיברים.

בפייתון ניתן לחשבה בעזרת `numpy.dot()`



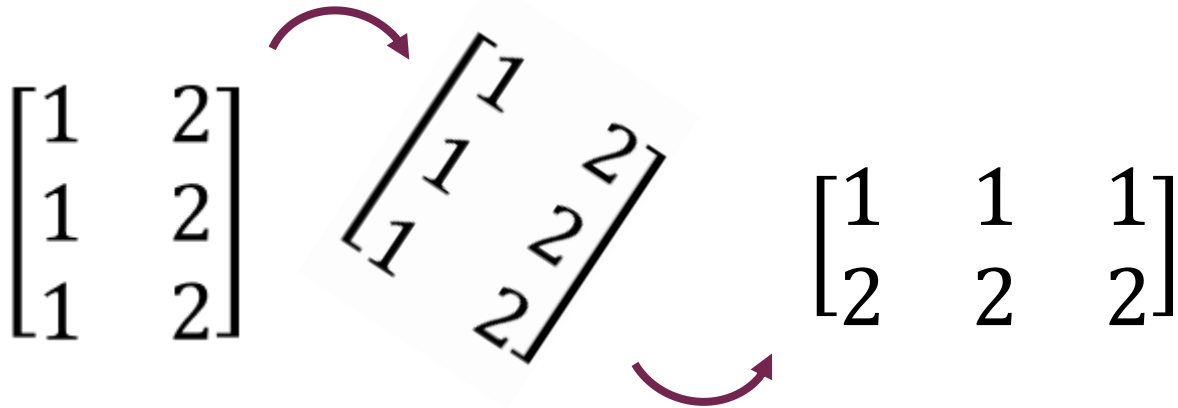
בכפל בין מטריצות צריך לשים לב שהמימדים הפנימיים מתאימים.

מימדי התוצאה יהיו המימדים החיצוניים.

$$\begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 \\ 1 & 2 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 3 & 6 \\ 6 & 12 \\ 9 & 18 \end{bmatrix}$$

<https://matrixmultiplication.xyz/>

כאשר המימדים לא בדיוק מתאימים אפשר לעשות טרנספוזיציה (בפייתון T).


$$\begin{bmatrix} 1 & 2 \\ 1 & 2 \\ 1 & 2 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 6 \\ 6 & 12 \\ 9 & 18 \end{bmatrix}$$



כדי להבין מה קורה בחלק הזה  
צריך קודם להיזכר קצת בסטטיסטיקה:

מה שStandard Scaler עושה זה סטנדרטיזציה לנתונים.

```
from sklearn.preprocessing import StandardScaler  
# Data preprocessing  
scaler = StandardScaler()  
X_train_scaled = scaler.fit_transform(X_train)  
X_test_scaled = scaler.transform(X_test)
```

כלומר הוא מחליף כל נתון בציון התקן שלו.

לשם כך הוא מחשב את הממוצע וסטית התקן על סט האימון  
ואז משתמש בהם כדי לחשב גם את ציוני התקן של נתוני המבחן.

בסוף התהליך מתקבל סט נתונים עם שונות 1 וממוצע אפס.  
אנחנו עושים את התהליך הזה למשל כשיש יחידות מידה שונות לכל פיצ'ר  
(למשל משקל של רכב זה בק"ג וכח 100 מודדים ב... כח 100).

**ואם זה נשמע זר ומוזר- אז הנה כמה שקפים שכבר ראינו פעם:  
(מצגת "05 למי קראת נורמלי" בסטטיסטיקה )**

# חישוב: ציון תקן

## Z score

```
from sklearn.preprocessing import StandardScaler
# Data preprocessing
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

## כמה סטיות תקן מהממוצע – נמצא ערך מסויים

ציון התקן מאפשר להשוות באופן יחסי ערכים מהתפלגויות שונות.

כאשר התצפית גדולה מהממוצע (תוחלת), ציון התקן יהיה חיובי.  
כאשר התצפית קטנה מהממוצע (תוחלת), ציון התקן יהיה שלילי.  
כאשר התצפית שווה לממוצע, ציון התקן יהיה אפס.

$$Z = \frac{x_i - \mu}{\sigma}$$

```
from sklearn.preprocessing import StandardScaler
# Data preprocessing
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

# so standardization

אם נחשב לכל נתון את ציון התקן שלו, נקבל קבוצה חדשה של ציוני תקן.

הממוצע של קבוצה זו הוא תמיד 0 ,  
והשונות תמיד 1

**תהליך נקרא סטנדרטיזציה**

```
from sklearn.preprocessing import StandardScaler
# Data preprocessing
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

# standardization

## סטנדרטיזציה

Index	x	z
1	160	-1.345
2	163	-0.905
3	167	-0.318
4	170	0.122
5	175	0.856
6	180	1.589
Mean	169.17	0
SD	6.82	1

הממוצע של קבוצה זו הוא תמיד 0 ,  
והשונות תמיד 1.

לדוגמא בטבלה הבאה נתונים גבהים של 6 סטודנטים. לאחר שחושב הממוצע וחושבה סטיית התקן של גבהי הסטודנטים, חושב ציון התקן של כל אחד מגבהים. לבסוף, חושב ממוצע וחושבה סטיית תקן לציוני התקן



✓ עקרון המורד הגרדיאנטי

✓ נגזרות

✓ פעולות מתמטיות עם וקטורים,

סקאלרים ומטריצות

✓ סטנדרטיזציה