



קריית החינוך  
פארק המדע  
בית לערכים  
למצוינות ולחדשנות

# Gradient descent

גלעד מרקמן

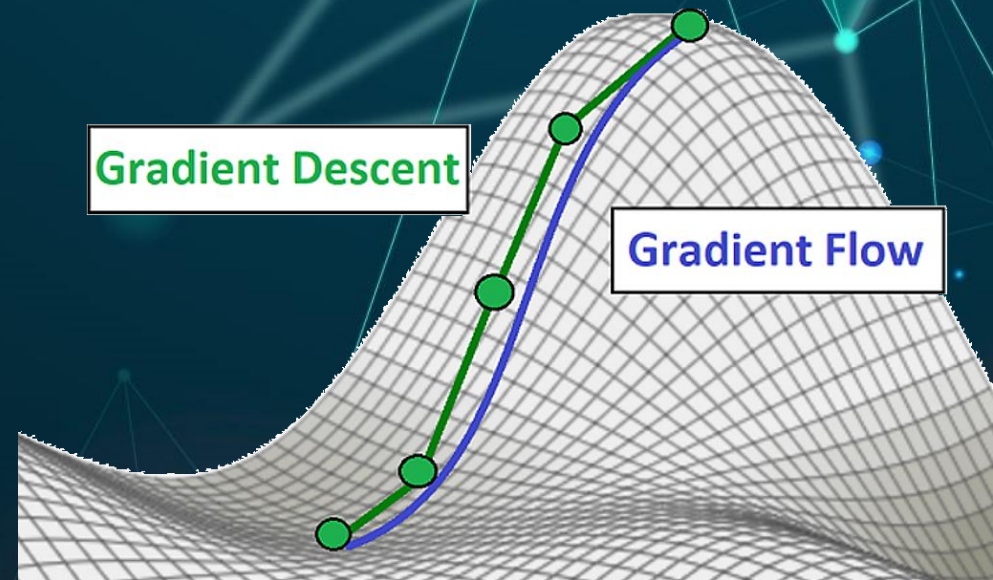
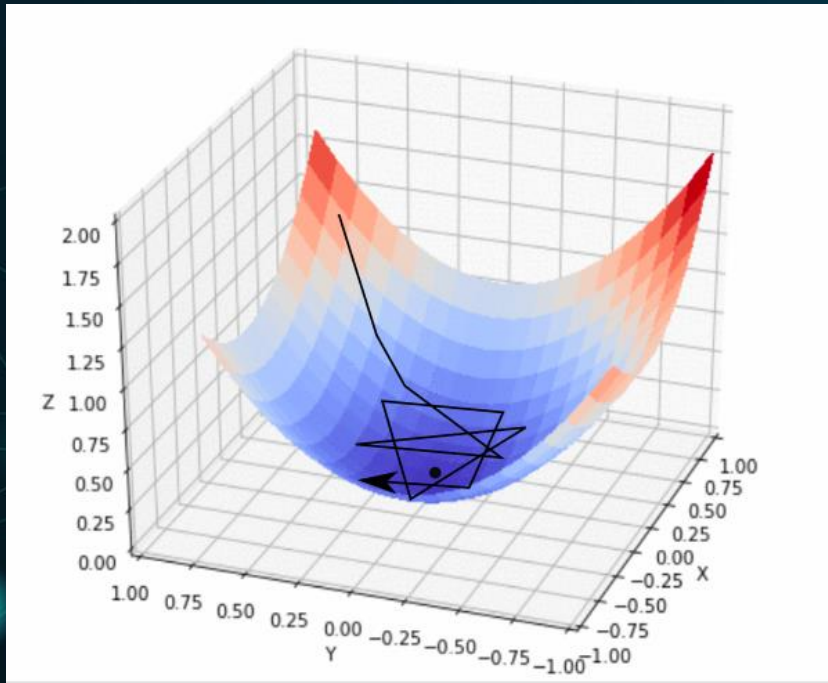
[Google Colab notebook](#)



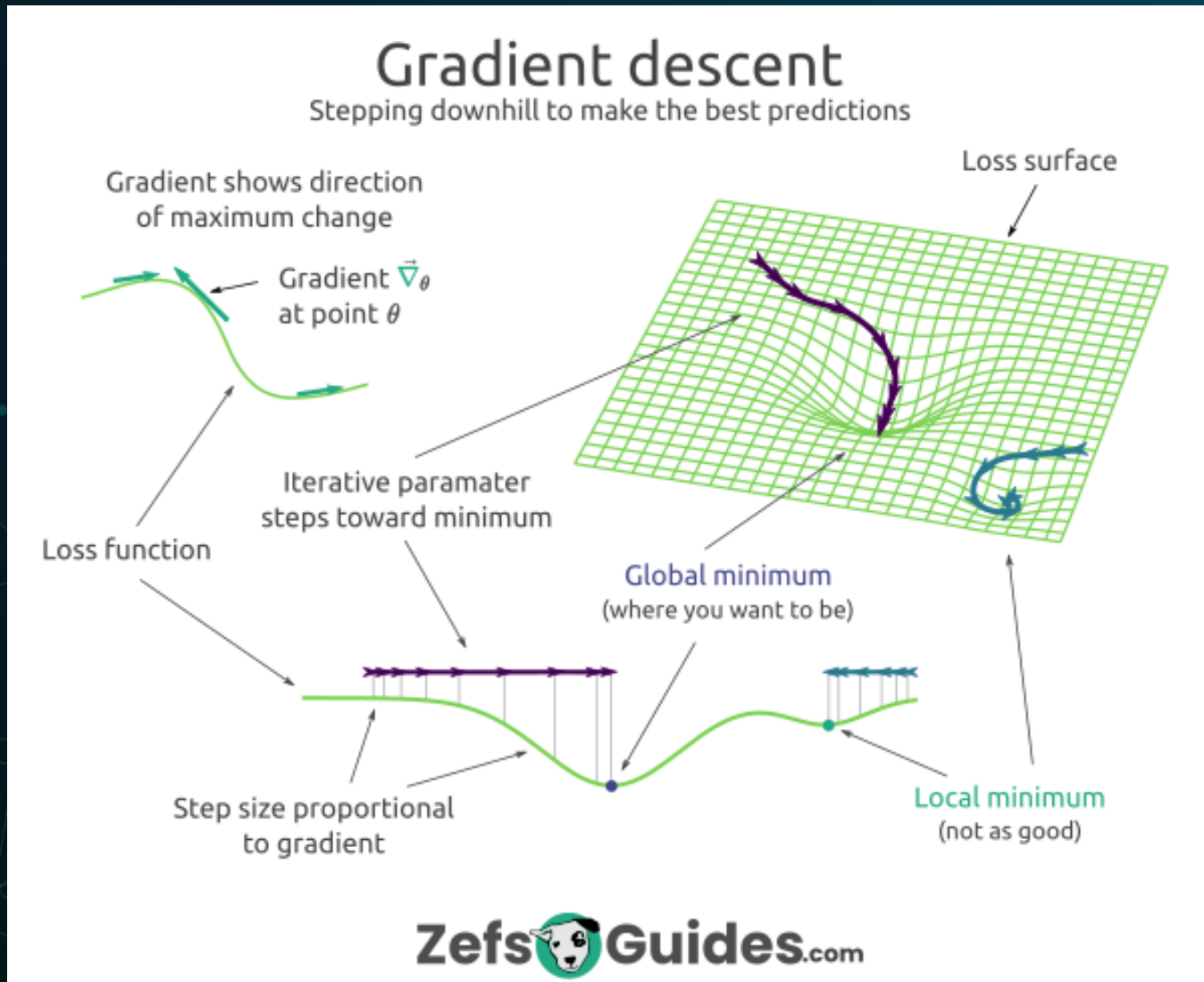


# מציאת מינימום לפונקציה במספר משתנים

- פונקציה במשתנה אחד מתארת קו במישור. פונקציה בשני משתנים מתארת משטח במרחב. פונקציה בשלושה מימדים ומעלה כבר לא ניתן להציג באופן גרפי.
- גם לפונקציות במספר משתנים ניתן למצוא מינימום מקומי בעזרת אלגוריתם SGD.



# אלגוריתם Gradient descent בשני משתנים



- האלגוריתם במקרה של מספר מימדים / משתנים זהה לאלגוריתם במשתנה אחד.

- בכל איטרציה אנחנו מחשבים את הנגזרת החלקית לפי כל אחד מן המשתנים.

- מעדכנים את המשתנה בהתאם לנגזרת החלקית שלו לפי אותה נוסחה.



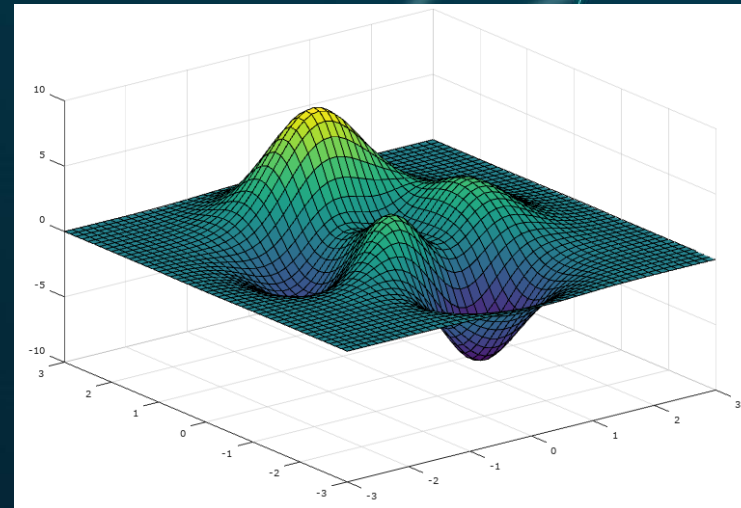
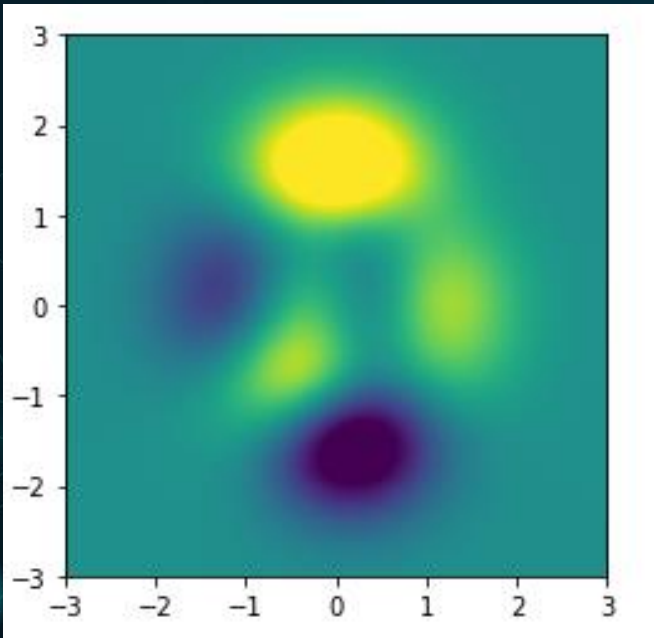
# הדגמה למציאת מינימום בפונקציה עם שני משתנים

• נתונה הפונקציה הבאה  $peaks()$ :

$$Z = 3(1 - x)^2 * e^{-x^2 - (y+1)^2} - 10 * \left(\frac{x}{5} - x^3 - y^5\right) * e^{-x^2 - y^2} - \frac{1}{3} e^{-(x+1)^2 - y^2}$$

• From Udemy, A deep understanding of deep learning, Mike X Cohen

• ניתן להציג את הפונקציה באמצעות הגרף הבא, כאשר צבע צהוב מתאר נקודה גבוהה וצבע כחול נקודה נמוכה.





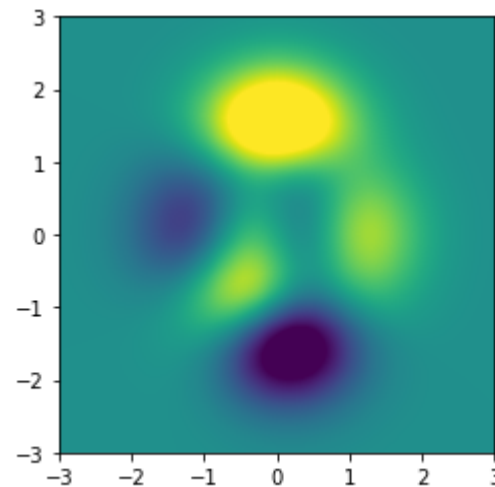
קריית החינוך  
פארק המדע  
בית לערכים  
למצוינות ולחדשנות

# הכנת הנתונים – הדפסת הגרף

```
def peaks(x,y):  
    # expand to a 2D mesh  
    x,y = np.meshgrid(x,y)  
  
    z = 3*(1-x)**2 * np.exp(-(x**2) - (y+1)**2) \  
        - 10*(x/5 - x**3 - y**5) * np.exp(-x**2-y**2) \  
        - 1/3*np.exp(-(x+1)**2 - y**2)  
    return z
```

בגרף מימין:  
צבע צהוב – מתאר ערך גבוה / גדול.  
צבע כחול – מתאר ערך נמוך / קטן

```
# create the landscape  
x = np.linspace(-3,3,201)  
y = np.linspace(-3,3,201)  
  
z = peaks(x,y)  
  
# let's have a look!  
plt.imshow(z,extent=[x[0],x[-1],y[0],y[-1]],vmin=-5,vmax=5,origin='lower')  
plt.show()
```





קריית החינוך  
פארק המדע  
בית לערכים  
למציאות ולחדשנות

# מציאת המינימום ב 2D

```
def Peaks(x, y):  
    return 3*(1-x)**2 * torch.exp(-(x**2) - (y+1)**2) \  
        - 10*(x/5 - x**3 - y**5) * torch.exp(-x**2-y**2) \  
        - 1/3*torch.exp(-(x+1)**2 - y**2)
```

פונקציה עם טנסור

```
# initialize random weight and params  
X = torch.tensor(-1, dtype=torch.float32 ,requires_grad=True)  
Y = torch.tensor(-2, dtype=torch.float32,requires_grad=True)  
learning_rate = 0.01  
  
# init optimizer  
optimizer = torch.optim.SGD([X,Y], lr=learning_rate)  
  
for epoch in range (1000):  
    # Forward  
    Z = Peaks(X,Y)  
  
    # backward - calculate grads  
    Z.backward()  
  
    # optimize wights  
    optimizer.step()  
  
    if epoch % 100 == 0:  
        print(f"epoch= {epoch} \t X,Y = {X.item():.3f}, {Y.item():.3f} \t Z={Z:.3f} \t X_grad, Y_grad= {X.grad:.3f} , {Y.grad:.3f}")  
  
    # zero wights  
    optimizer.zero_grad()  
  
print(f"End X,Y = {X.item():.3f}, {Y.item():.3f} Z={Z:.3f}")
```

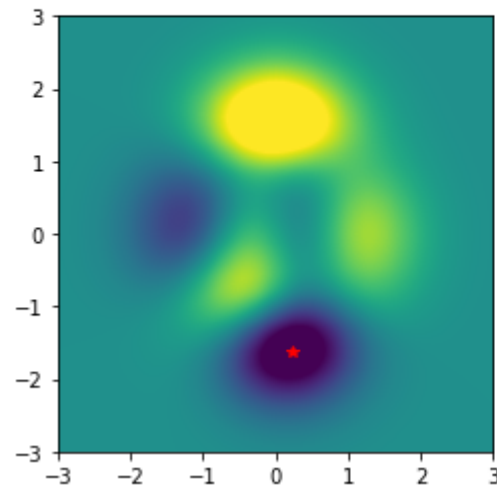


קריית החינוך  
פארק המדע  
בית לערכים  
למצוינות ולחדשנות

# תוצאות ההרצה של הקוד

```
epoch= 0      X,Y = -0.974, -1.998      Z=-0.592      X_grad, Y_grad= -2.607 , -0.226
epoch= 100    X,Y = 0.228, -1.626      Z=-6.551      X_grad, Y_grad= -0.000 , -0.000
epoch= 200    X,Y = 0.228, -1.626      Z=-6.551      X_grad, Y_grad= -0.000 , -0.000
epoch= 300    X,Y = 0.228, -1.626      Z=-6.551      X_grad, Y_grad= -0.000 , -0.000
epoch= 400    X,Y = 0.228, -1.626      Z=-6.551      X_grad, Y_grad= -0.000 , -0.000
epoch= 500    X,Y = 0.228, -1.626      Z=-6.551      X_grad, Y_grad= -0.000 , -0.000
epoch= 600    X,Y = 0.228, -1.626      Z=-6.551      X_grad, Y_grad= -0.000 , -0.000
epoch= 700    X,Y = 0.228, -1.626      Z=-6.551      X_grad, Y_grad= -0.000 , -0.000
epoch= 800    X,Y = 0.228, -1.626      Z=-6.551      X_grad, Y_grad= -0.000 , -0.000
epoch= 900    X,Y = 0.228, -1.626      Z=-6.551      X_grad, Y_grad= -0.000 , -0.000
End X,Y = 0.228, -1.626 Z=-6.551
```

```
plt.imshow(z,extent=[x[0],x[-1],y[0],y[-1]],vmin=-5,vmax=5,origin='lower')
plt.plot(X.item(), Y.item(), '*', color='red')
plt.show()
```





# תרגיל

• הריצו את הקוד שבנינו עם הנתונים הבאים. הסבירו את ההגיון לתוצאות שהתקבלו בהרצות אילו:

- Learning\_rate = 0.1; X = 0; Y = 1.5
- Learning\_rate = 0.1; X = 0; Y = 2
- Learning\_rate = 0.01; X = 0; Y = -1
- Learning\_rate = 0.01; X = -2; Y = 0
- Learning\_rate = 0.5; X = -2; Y = 0

• מצאו את נקודת המינימום של הגרף הבא:  $f(x,y) = x \cdot e^{-(x^2+y^2)}$

• דוגמאות לגרפים דו מימדיים





# כללי

- תוצאות האלגוריתם של Gradient descent תלויות בפרמטרים שאותחלו, ולא תמיד נקבל את התשובה הנכונה.
- הפרמטרים הינם:
  - קצב הלימוד learning rate
  - הערכים בהם אותחלו הפרמטרים (במקרים שלנו  $X, Y$ ).
  - קיום מינימום מקומי לעומת מינימום מוחלט.
- על כן, למידת מכונה מחייבת ניסוי וטעיה, תוך שינוי הפרמטרים עד לקבלת התוצאה הרצויה.