# Foundations Of Neural Networks and Deep Learning

**Day-4**

**SME: Gagan P**

**Contact: gaganp000999@gmail.com**

# recap:

## 1.What is the primary goal of Linear Regression?

- A. To classify data into categories
- B. To predict a continuous output based on input features
- C. To cluster similar data points
- D. To reduce dimensionality of data

# recap:

## 1.What is the primary goal of Linear Regression?

- A. To classify data into categories
- B. **To predict a continuous output based on input features**
- C. To cluster similar data points
- D. To reduce dimensionality of data

**What type of machine learning algorithm is Linear Regression?**

A. Unsupervised learning
B. Reinforcement learning
C. Supervised learning
D. Semi-supervised learning

**What type of machine learning algorithm is Linear Regression?**

A. Unsupervised learning
B. Reinforcement learning
C. **Supervised learning**
D. Semi-supervised learning

**1. In the hypothesis function $h(x) = wx + b$, what does 'b' represent?**

A. Weight
B. Feature
C. Bias (intercept)
D. Slope

**1. In the hypothesis function $h(x) = wx + b$, what does 'b' represent?**

A. Weight
B. Feature
**C. Bias (intercept)**
D. Slope

# What is the general form of the hypothesis function in multivariate linear regression?

- A. $y = mx + c$
- B. $h(x) = w_1x_1 + w_2x_2 + \ldots + w_nx_n + b$
- C. $y = x^2 + c$
- D. $h(x) = w/x + b$

# What is the general form of the hypothesis function in multivariable linear regression?

- A. $y = mx + c$
- **B. $h(x) = w_1x_1 + w_2x_2 + \ldots + w_nx_n + b$**
- C. $y = x^2 + c$
- D. $h(x) = w/x + b$

**What is the purpose of the cost (loss) function?**

A. Measure accuracy
B. Measure average error of predictions
C. Add noise to data
D. Store training examples

# What is the purpose of the cost (loss) function?

A. Measure accuracy
**B. Measure average error of predictions**
C. Add noise to data
D. Store training examples

**What does MSE stand for in the context of cost functions?**

A. Mean Standard Error
B. Maximum Squared Error
C. Mean Squared Error
D. Minimum Statistical Error

**What does MSE stand for in the context of cost functions?**

A. Mean Standard Error
B. Maximum Squared Error
**C. Mean Squared Error**
D. Minimum Statistical Error

**A model with lower loss is considered:**

A. Worse than a model with higher loss
B. Better than a model with higher loss
C. The same as a model with higher loss
D. Invalid

## A model with lower loss is considered:

A. Worse than a model with higher loss
**B. Better than a model with higher loss**
C. The same as a model with higher loss
D. Invalid

**If the loss is too high, what should the model do?**

A. Keep weights and bias the same

B. Remove features from the dataset

C. Adjust weights and bias to decrease the loss

D. Increase the learning rate infinitely

**If the loss is too high, what should the model do?**

A. Keep weights and bias the same

B. Remove features from the dataset

C. **Adjust weights and bias to decrease the loss**

D. Increase the learning rate infinitely

**What is the ultimate goal when training a linear regression model?**

A. To maximize the loss function

B. To reach the local minima of the loss function

C. To increase the number of features

D. To make the model more complex

**What is the ultimate goal when training a linear regression model?**

A. To maximize the loss function

**B. To reach the local minima of the loss function**

C. To increase the number of features

D. To make the model more complex

# day 4 - gradient descent

How does a model learn the correct weights and bias?

How does it reduce the loss with every step?

# Answer: Gradient Descent

# Training The Model

- Finding w and b that minimize the cost function
-  The cost surface might look like a hill — we need to go downhill

```python
# Dataset: Area (sqft) vs Price (lakhs)

X = np.array([500, 800, 1000, 1200, 1500, 1800, 2000, 2500])
y=np.array([ 70, 100, 120, 140, 170, 200, 220, 270])
```

# now use the first 3 samples from each of X and y and make a table for w and loss function

| w | loss_value |
| --- | --- |
| 0.06 | |
| 0.1 | |
| 0.14 | |

**take bias=20**

# how do we represent a prediction in terms of a single feature

$$y = w \cdot x + b$$

bias

weights feature

# Mean Squared Error( MSE )

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - (w^T x_i + b))^2$$

$n$ = number of data points

$y_i$ = true value (actual output)

$\hat{y}_i$ = predicted value from your hypothesis

$y_i - \hat{y}_i$ = error (residual)

| X | y | y_pred | y-y_pred | (y - y_pred )**2 |
|---|---|---|---|---|
| 500 | 70 | | | |
| 800 | 100 | | | |
| 1000 | 120 | | | |

**mse= ⅓ (sum of all (y-y_pred)**2)**

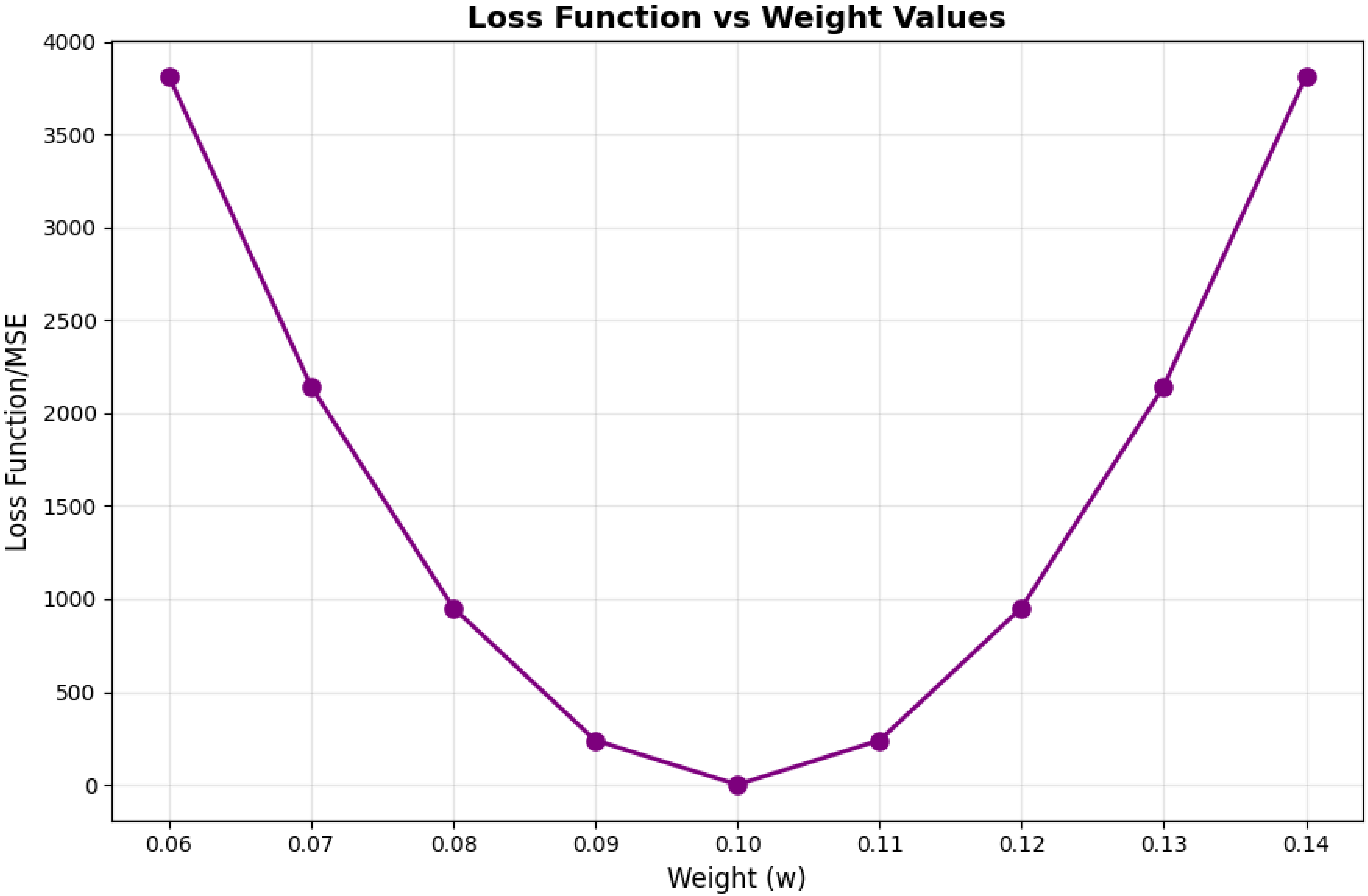
# use this mse here to fill this table for different value of w

## mse= ⅓ (sum of all (y-y_pred)**2)

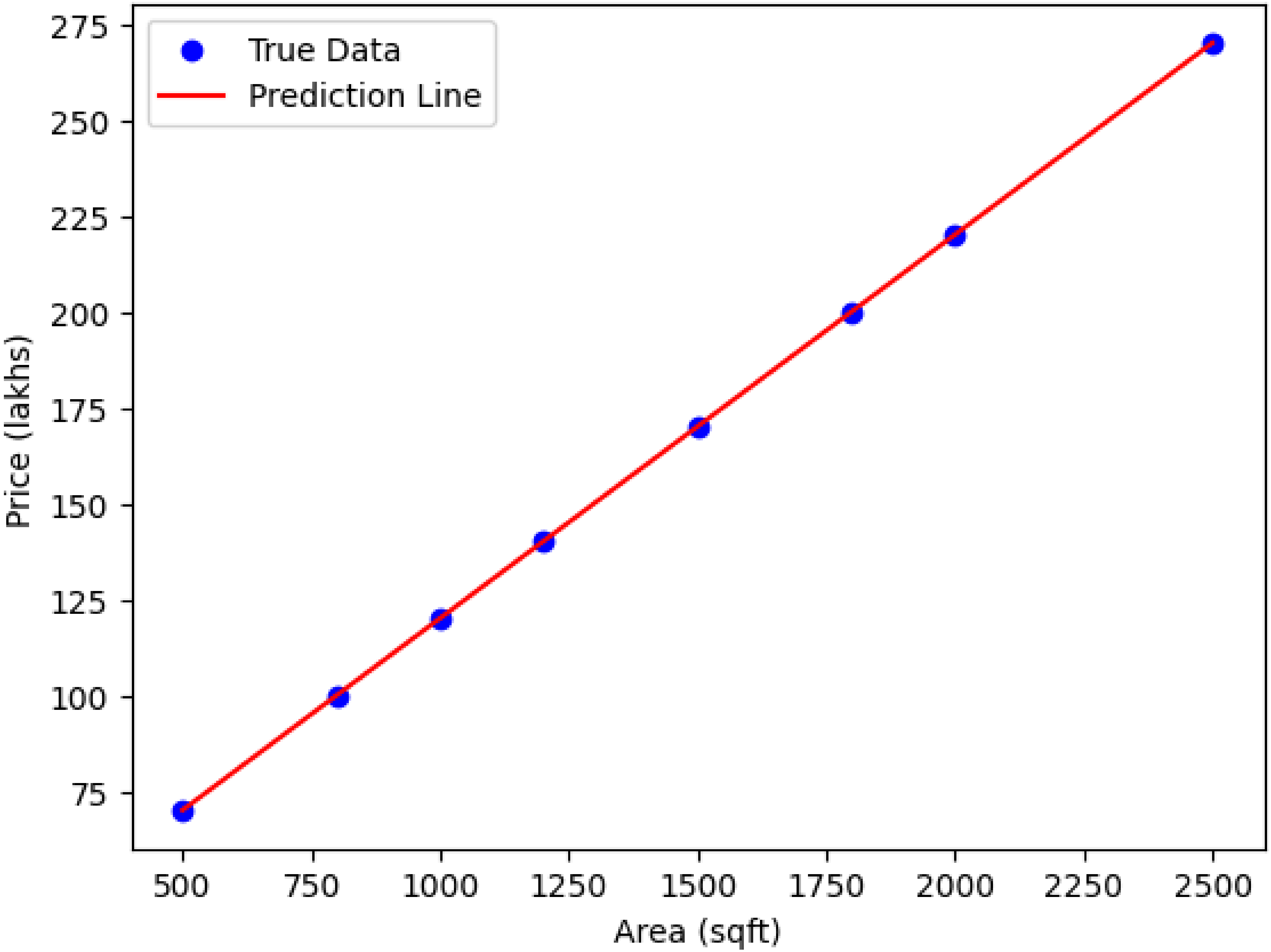
| X | y | y_pred | y-y_pred | (y - y_pred )**2 |
|---|---|---|---|---|
| 500 | 70 | | | |
| 800 | 100 | | | |
| 1000 | 120 | | | |

| w | loss_value |
|---|---|
| 0.06 | |
| 0.1 | |
| 0.14 | |

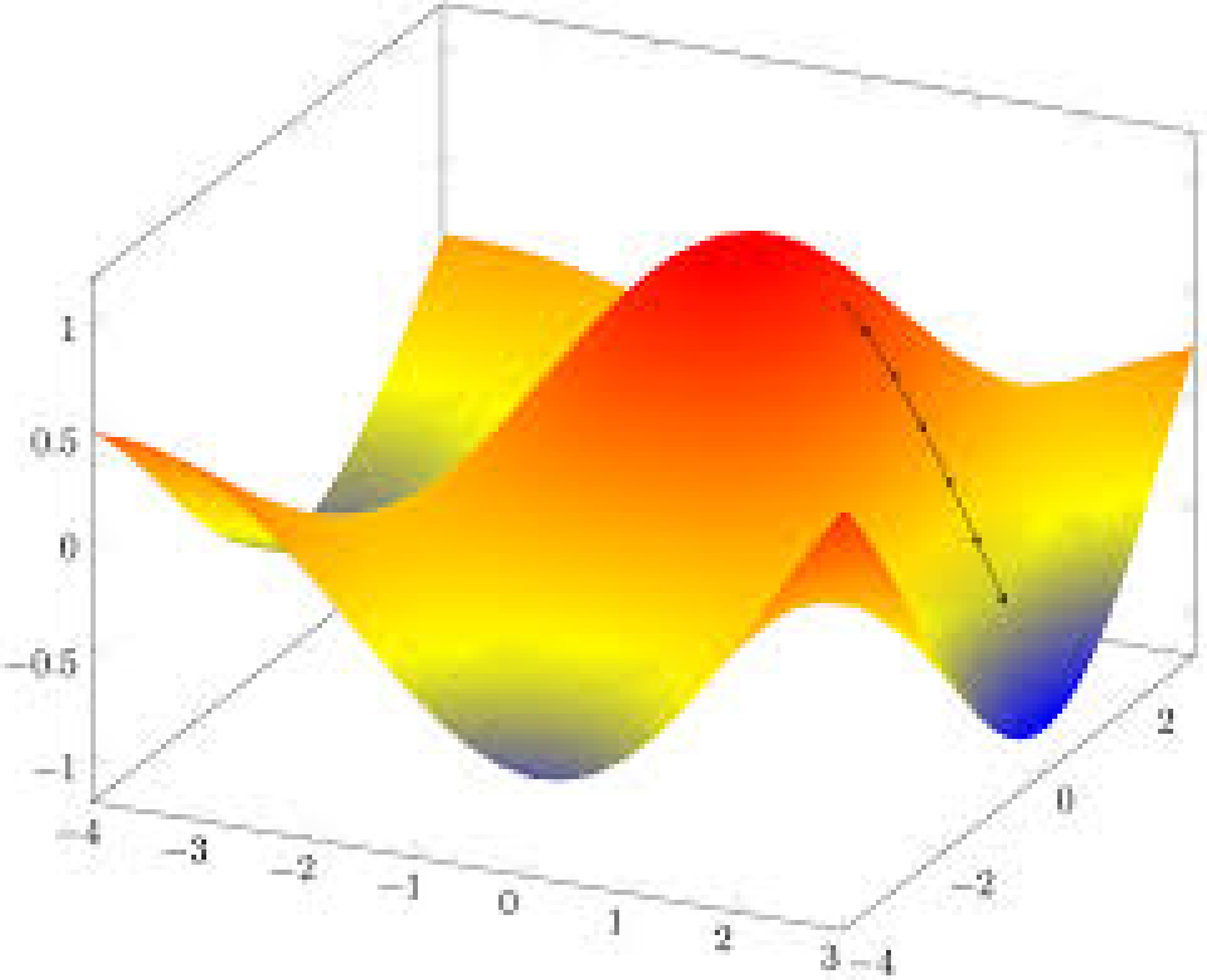
# if you plot all the points



# at w=0.1 and b=20

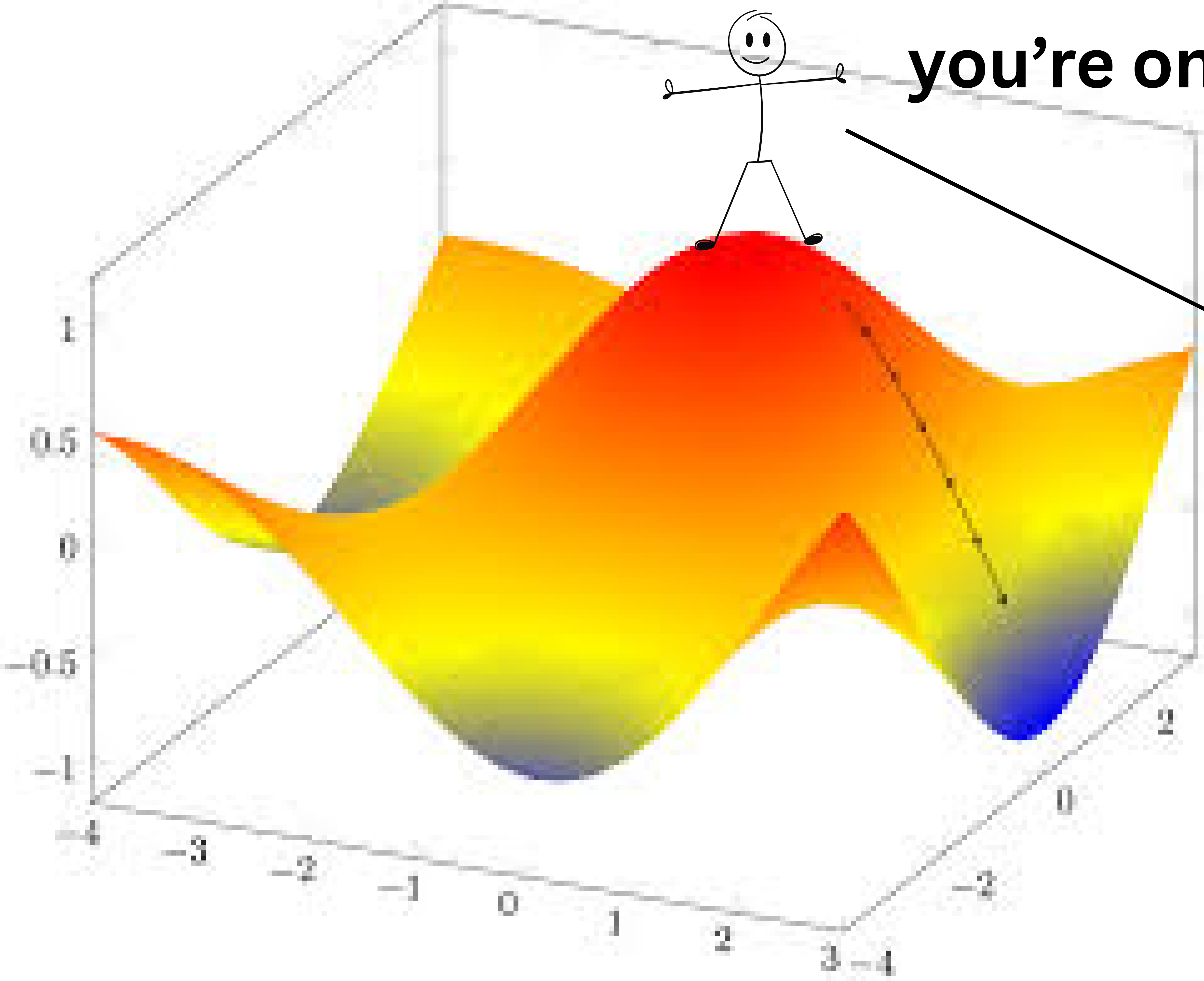


what if we used more than 1 feature, how does the loss function vs weights look like?



1
0.5
0
−0.5
−1
−4
−3
−2
−1
0
1
2
3
−4
−2
0
2

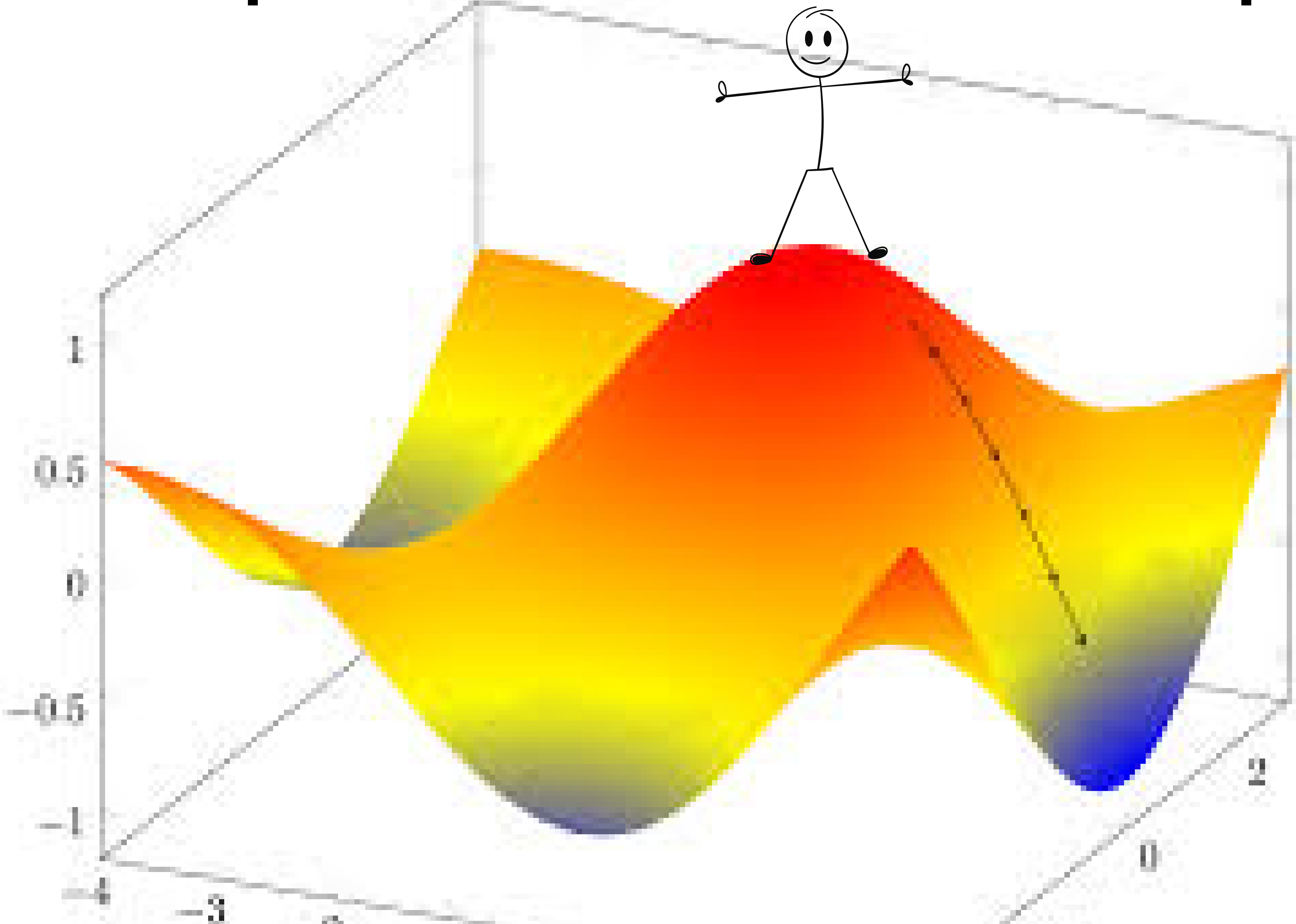
**you’re on top of the hill**



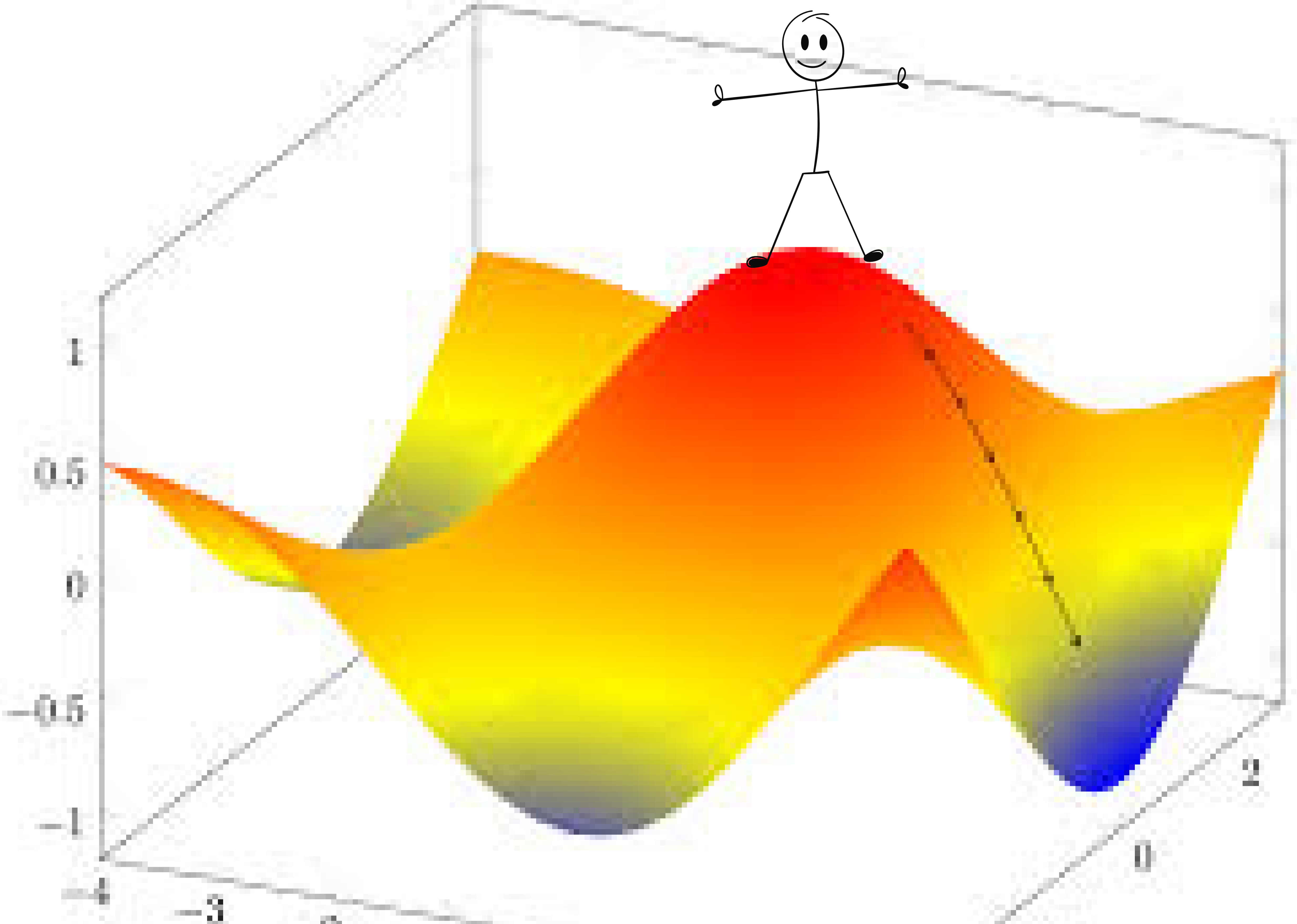

**imagine this person is weight(w)**

how do you reach the bottom of the hill?

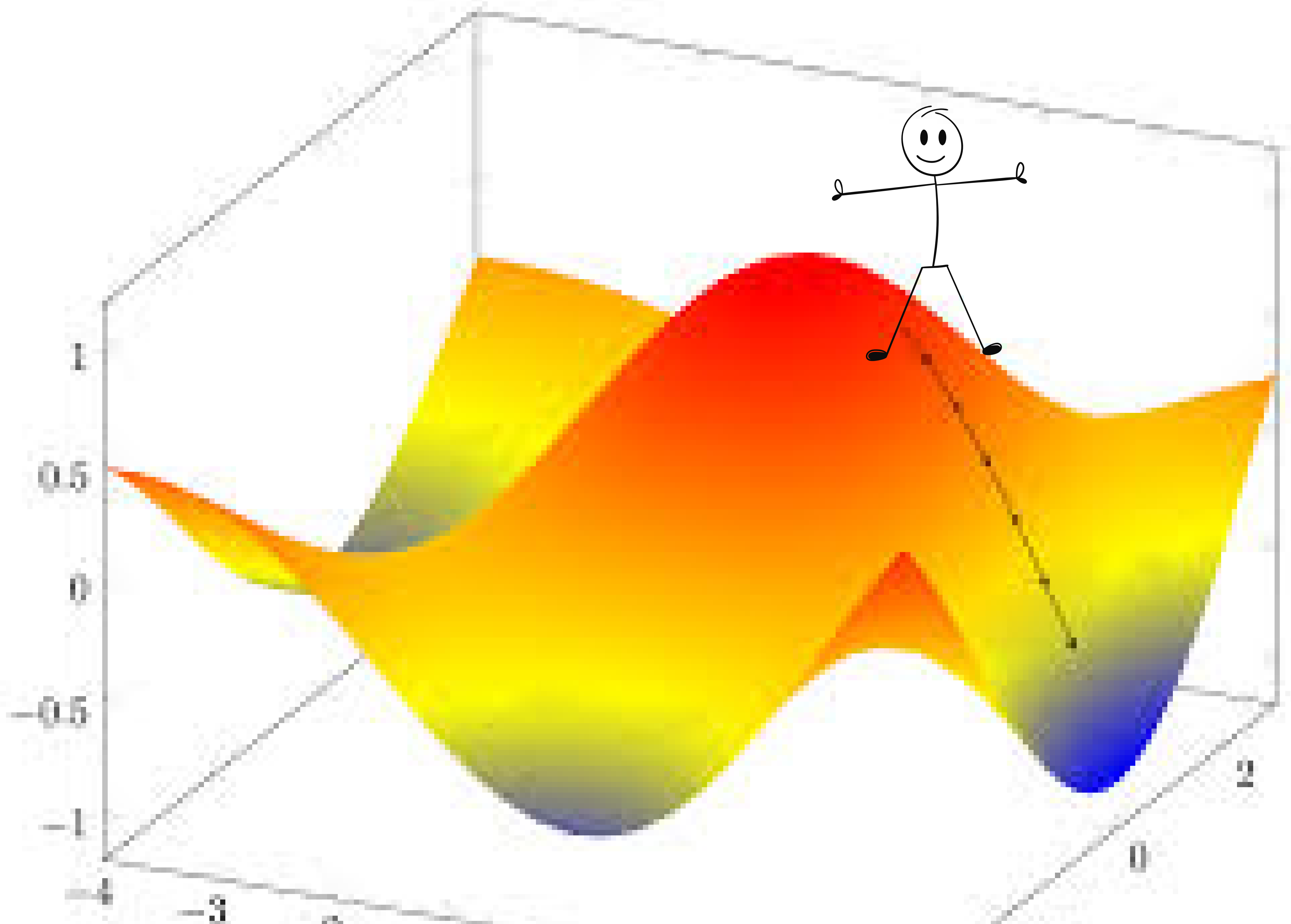
you look around the hill 360 degree and move towards the path which has the most slope



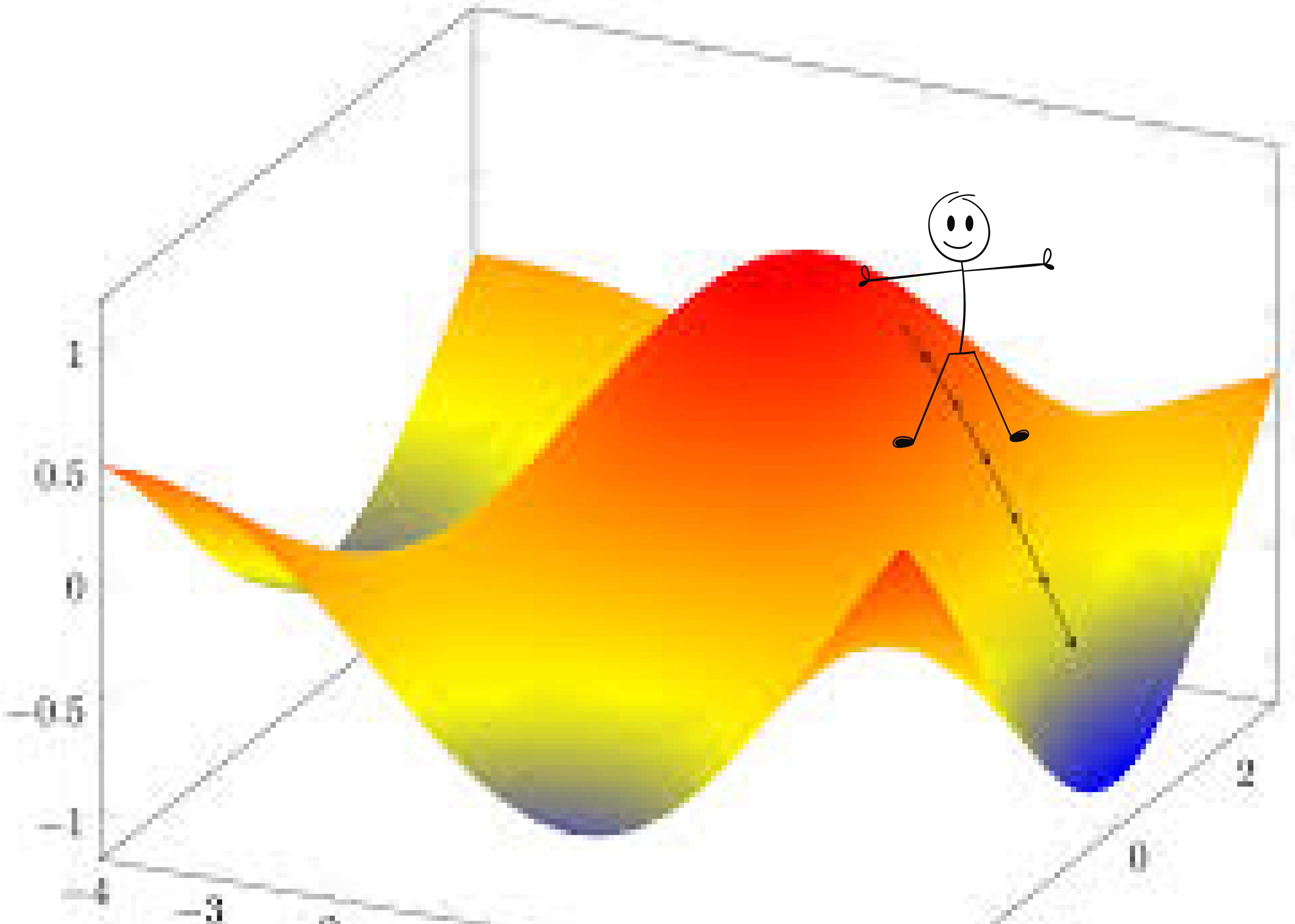
# you then take small steps



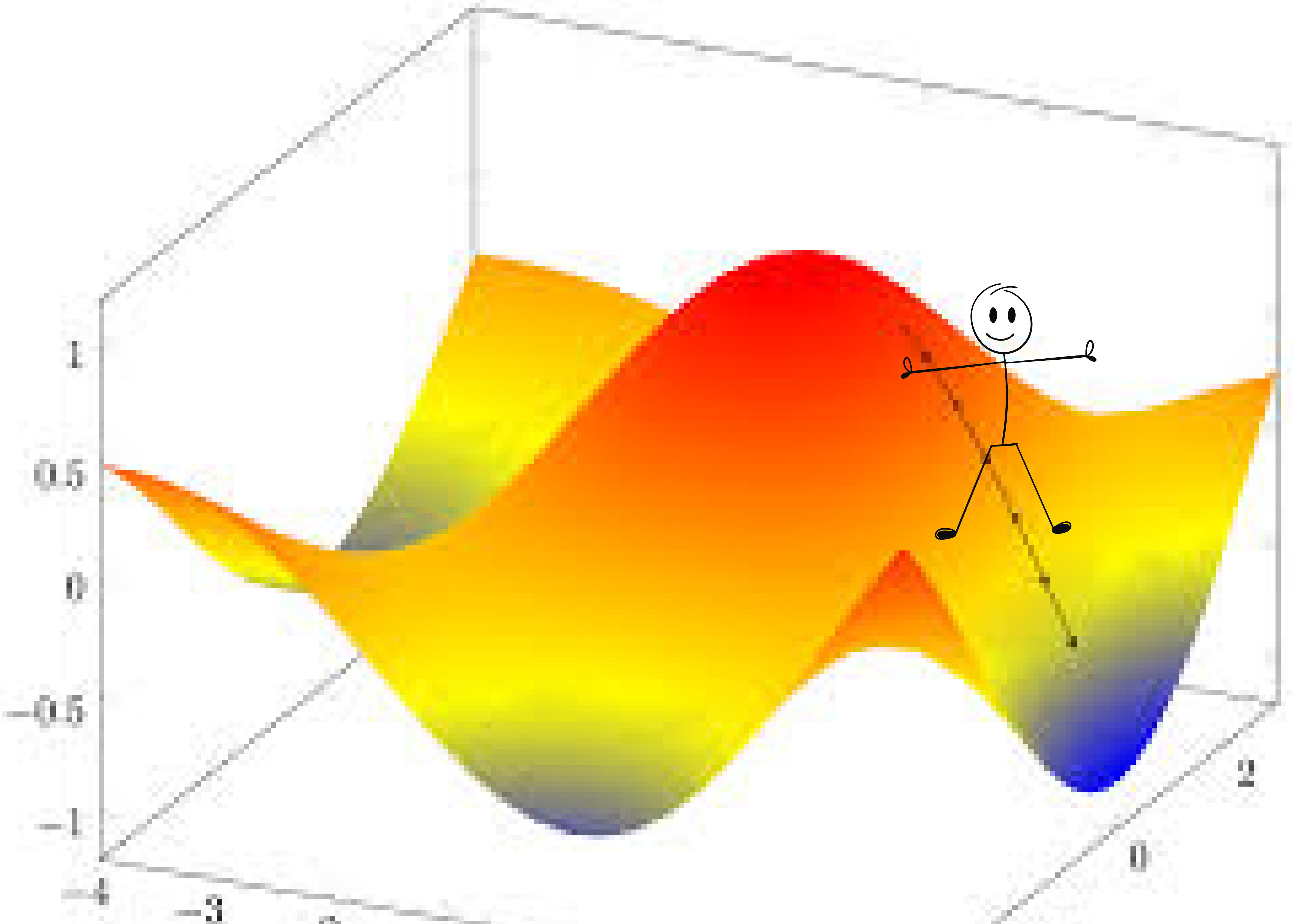
# you then take small steps



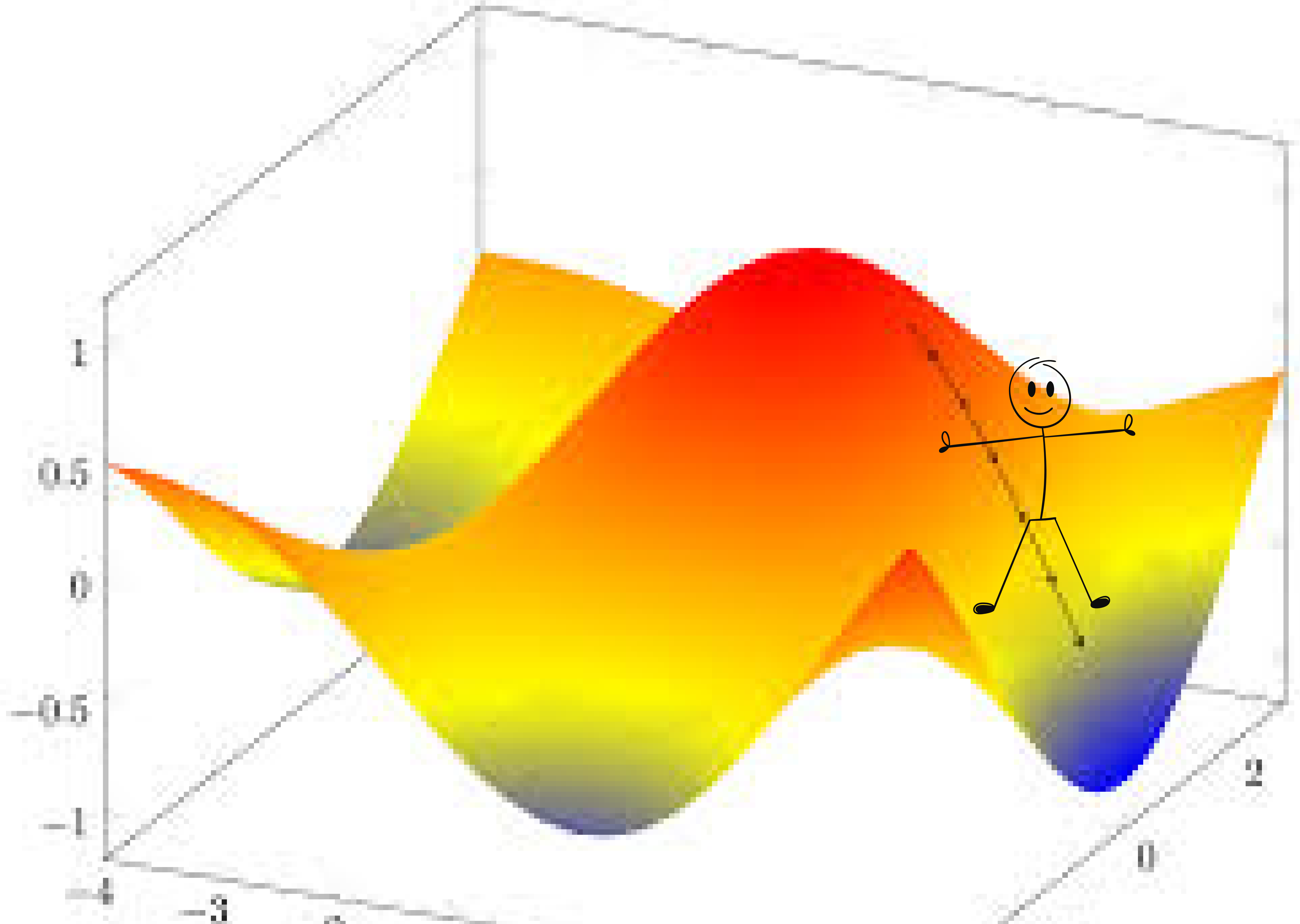
# you then take small steps



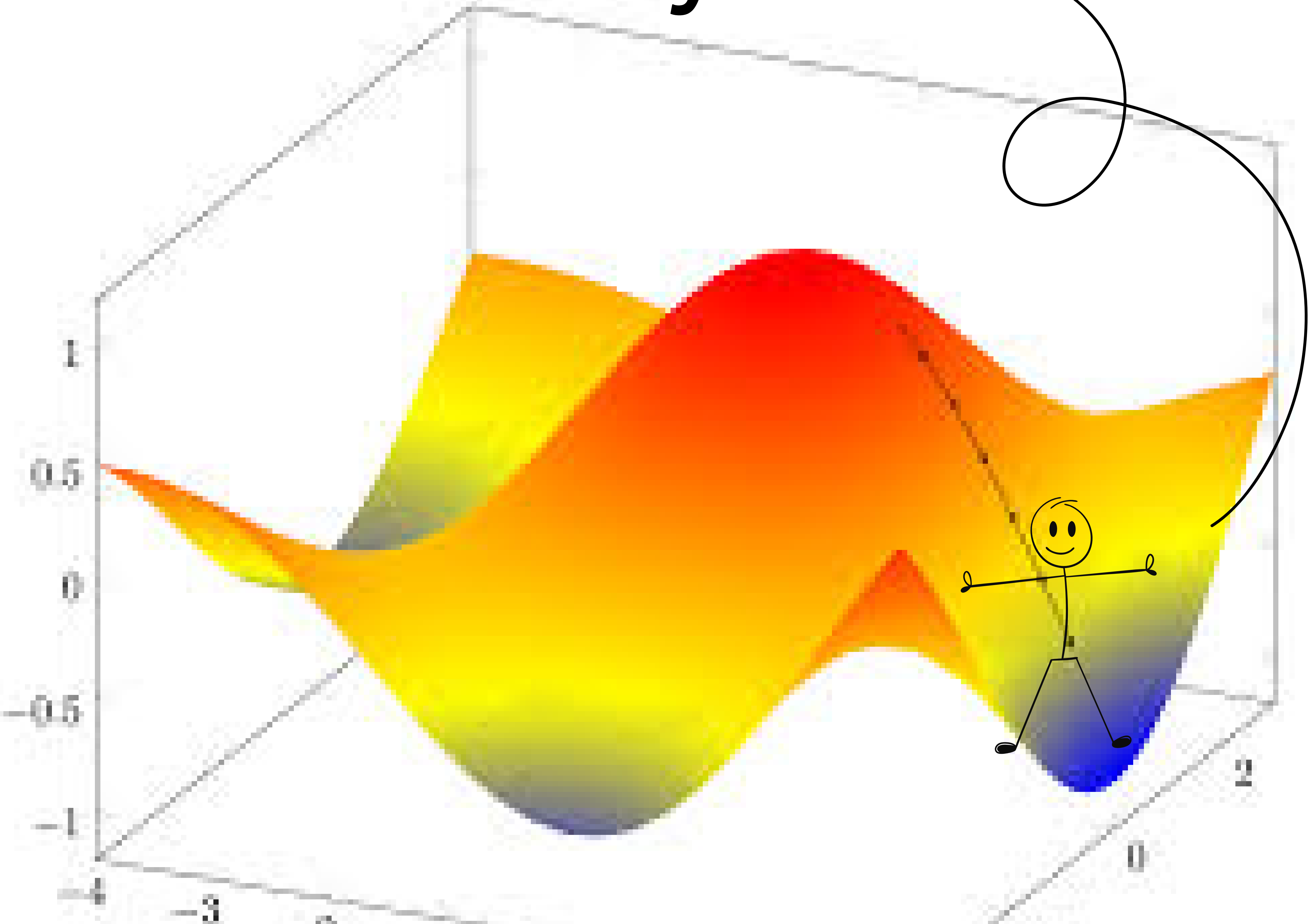
# you then take small steps



# you then take small steps

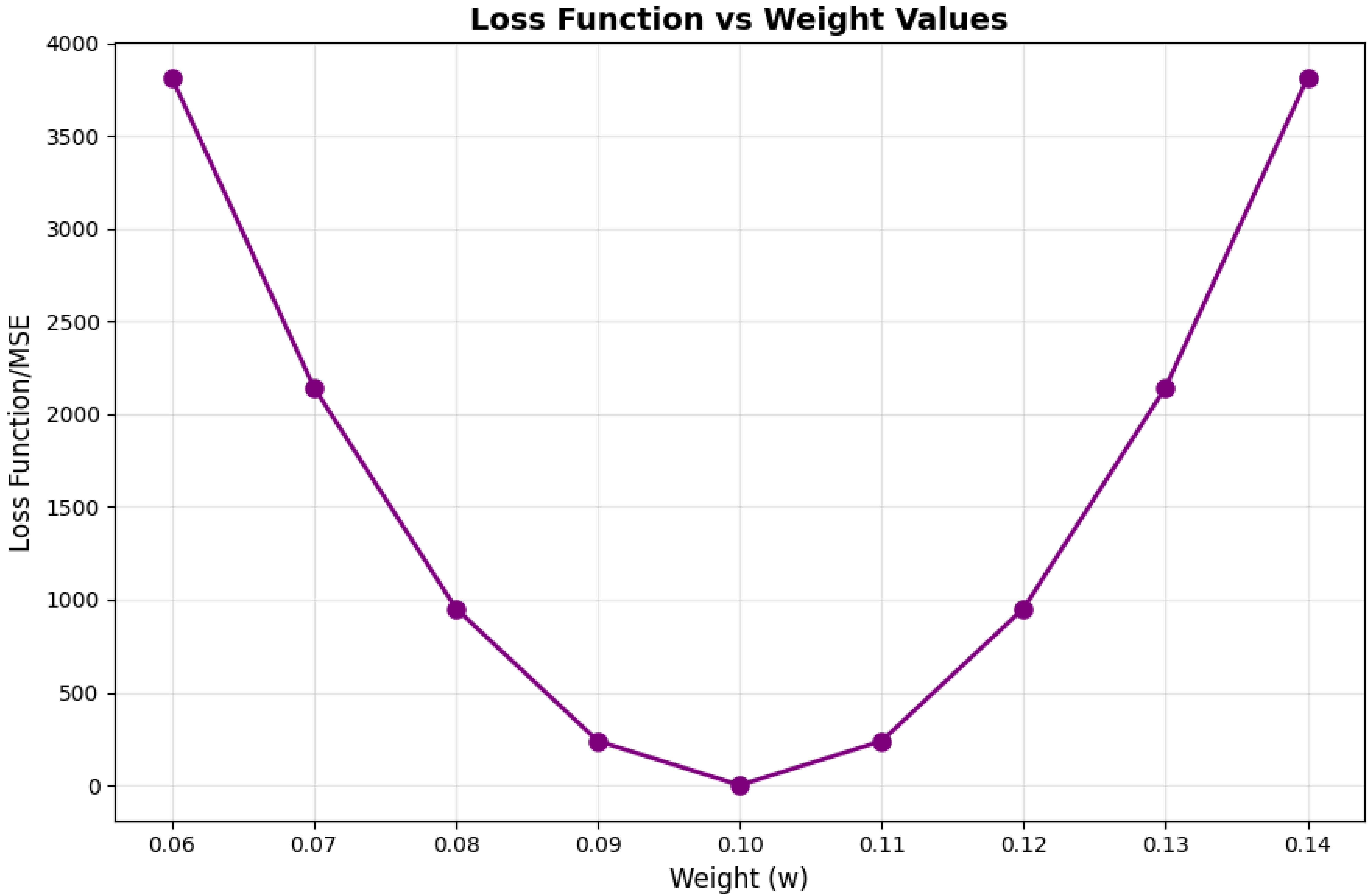


**can’t reach bottom anymore!!**



# GRADIENT DESCENT STOPPED!!

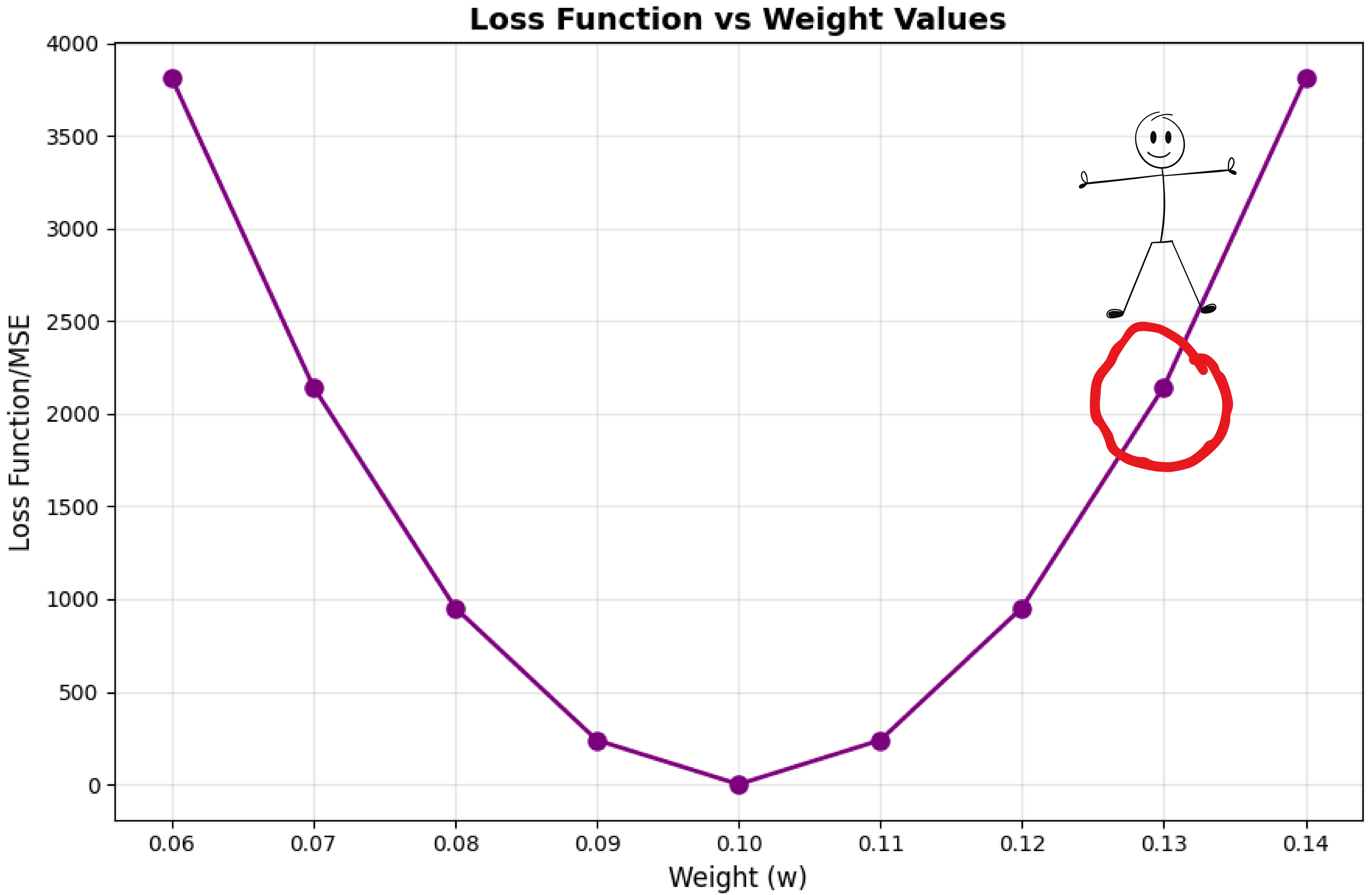
# Now let’s understand it mathematically



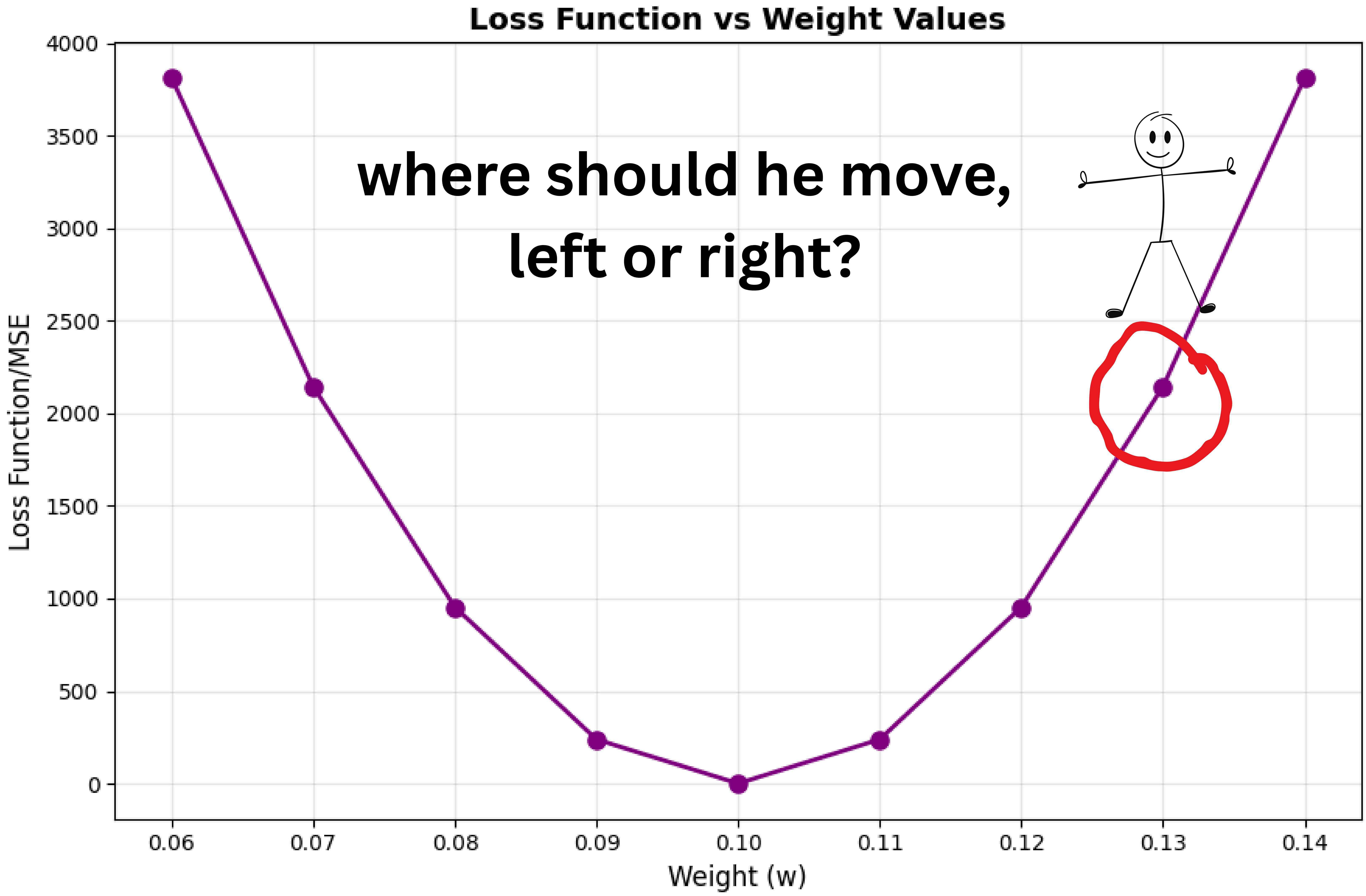
Loss Function vs Weight Values
Loss Function/MSE
Weight (w)
0
500
1000
1500
2000
2500
3000
3500
4000
0.06
0.07
0.08
0.09
0.10
0.11
0.12
0.13
0.14

# let’s say at the beginning of the training, the weight w=0.13 and b=20

$$y = \underbrace{0.13}_{\text{slope}}x + \underbrace{20}_{\text{intercept}}$$

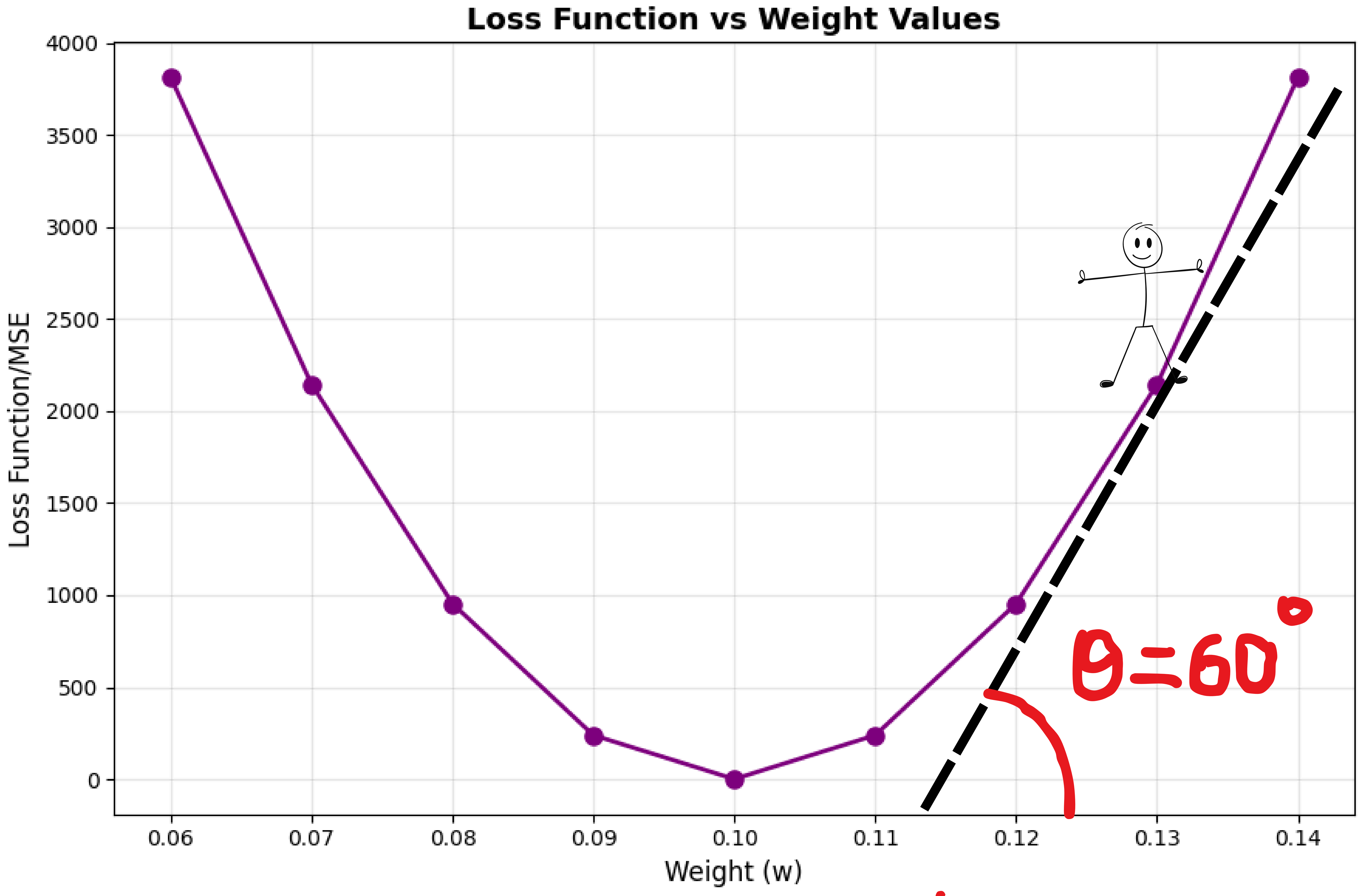


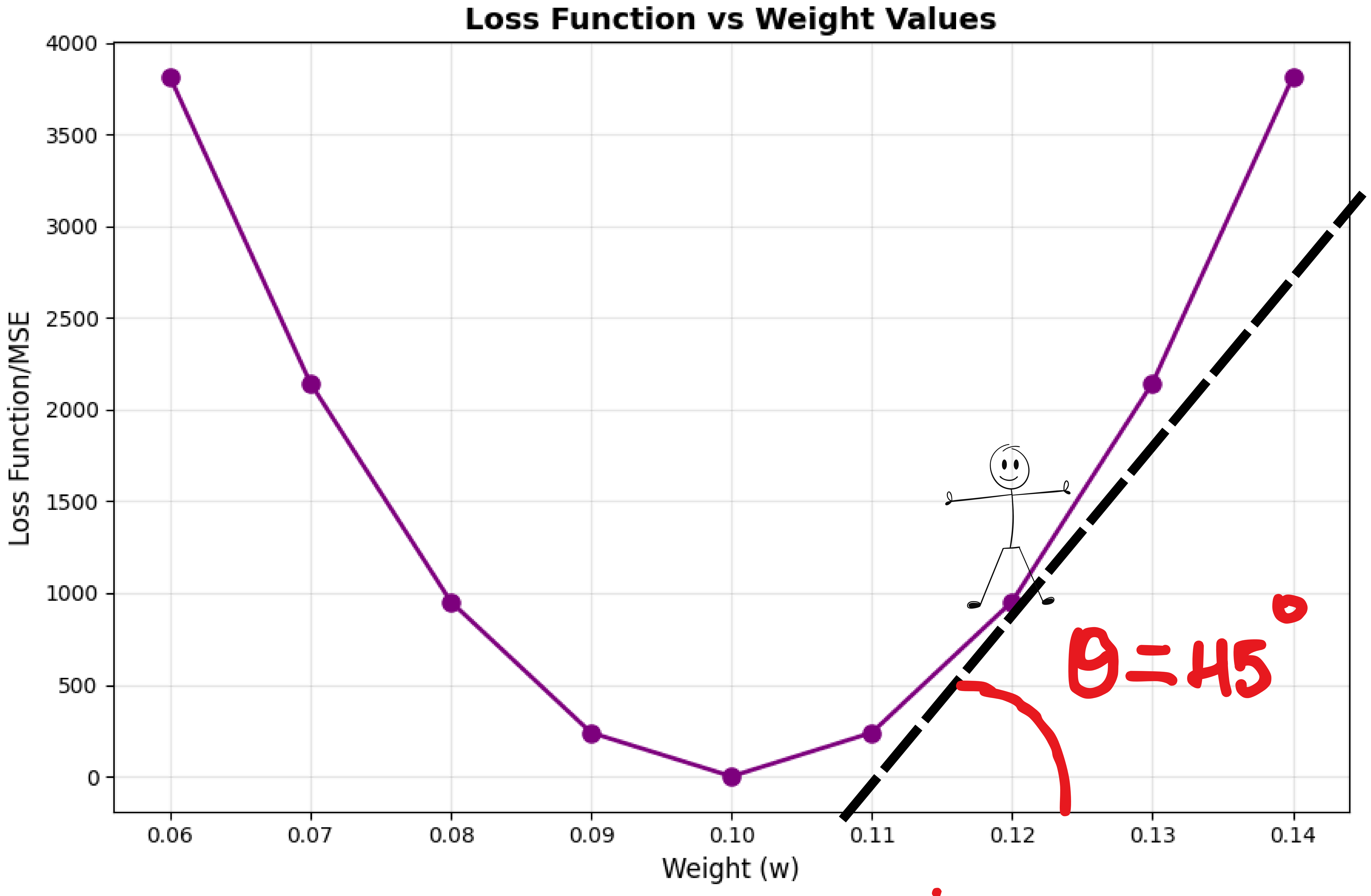
Loss Function vs Weight Values
Loss Function/MSE
Weight (w)
4000
3500
3000
2500
2000
1500
1000
500
0
0.06
0.07
0.08
0.09
0.10
0.11
0.12
0.13
0.14


Loss Function vs Weight Values
where should he move,
left or right?
Loss Function/MSE
Weight (w)
4000
3500
3000
2500
2000
1500
1000
500
0
0.06
0.07
0.08
0.09
0.10
0.11
0.12
0.13
0.14

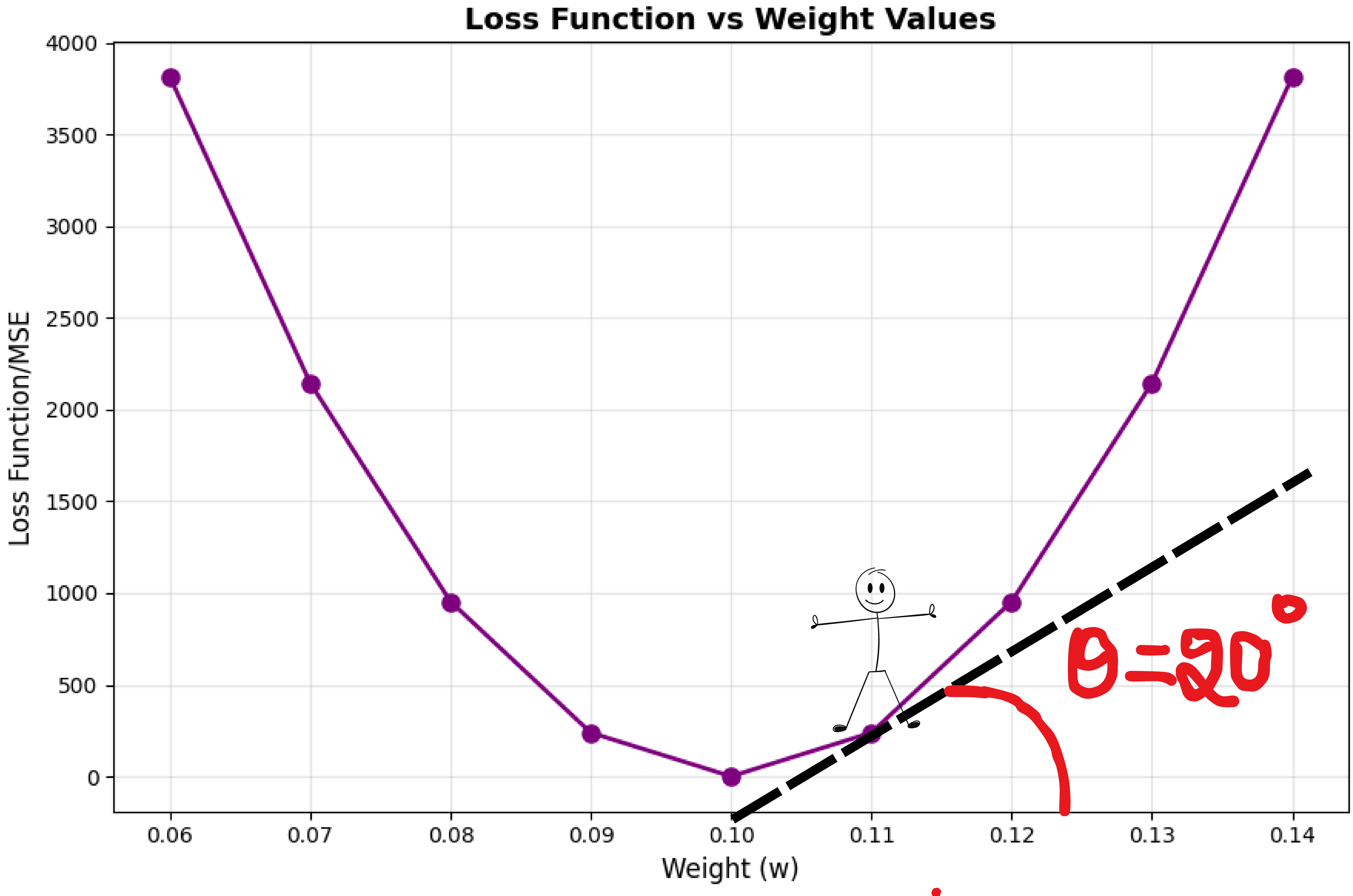
now what is the meaning of slope?
remember calculus

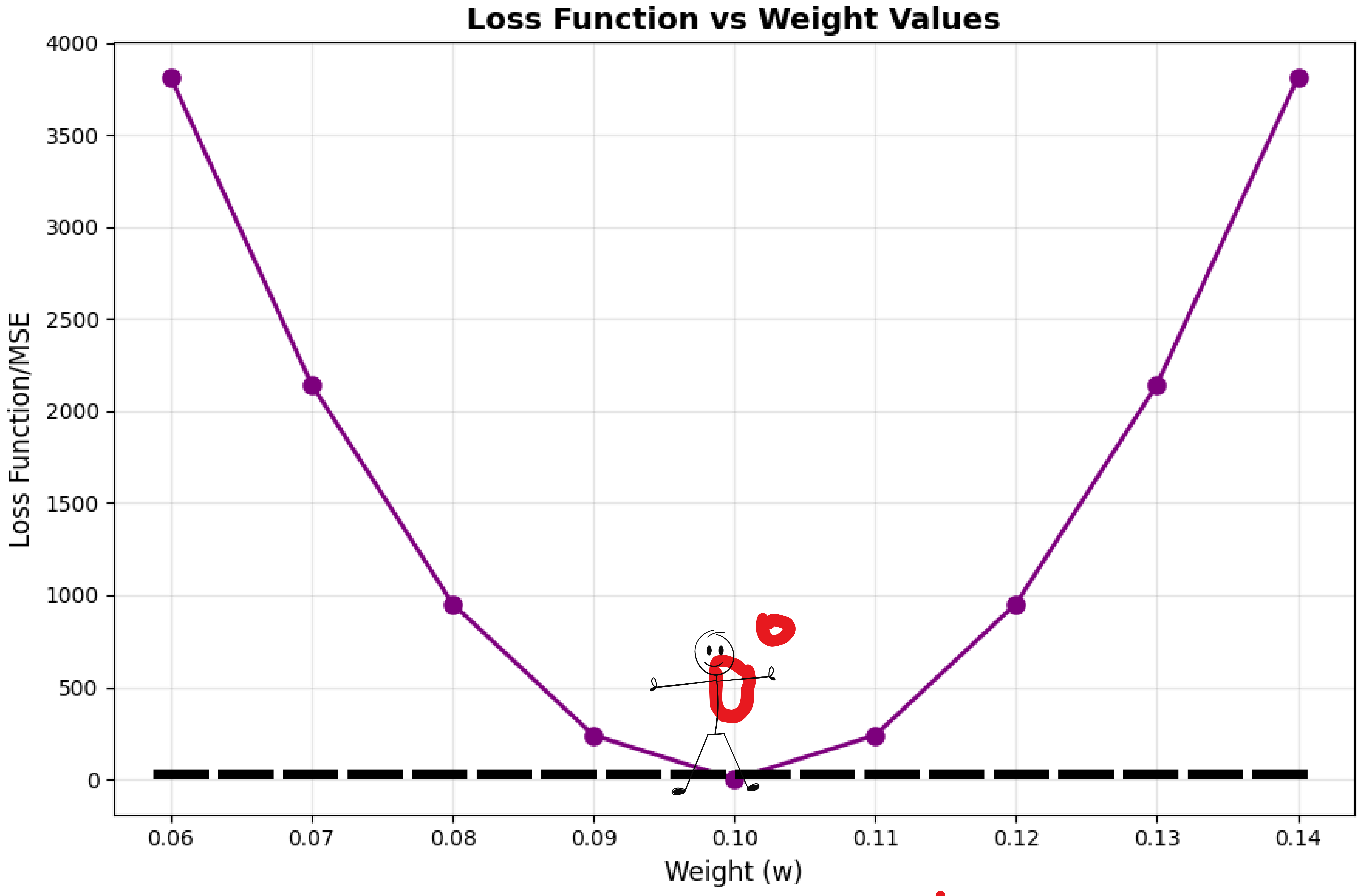
angle made by the tangent to that point!!


Loss Function vs Weight Values
Loss Function/MSE
4000
3500
3000
2500
2000
1500
1000
500
0
0.06
0.07
0.08
0.09
0.10
0.11
0.12
0.13
0.14
Weight (w)
θ=60°



Loss Function vs Weight Values
Loss Function/MSE
0
500
1000
1500
2000
2500
3000
3500
4000
0.06
0.07
0.08
0.09
0.10
0.11
0.12
0.13
0.14
Weight (w)
θ=45°

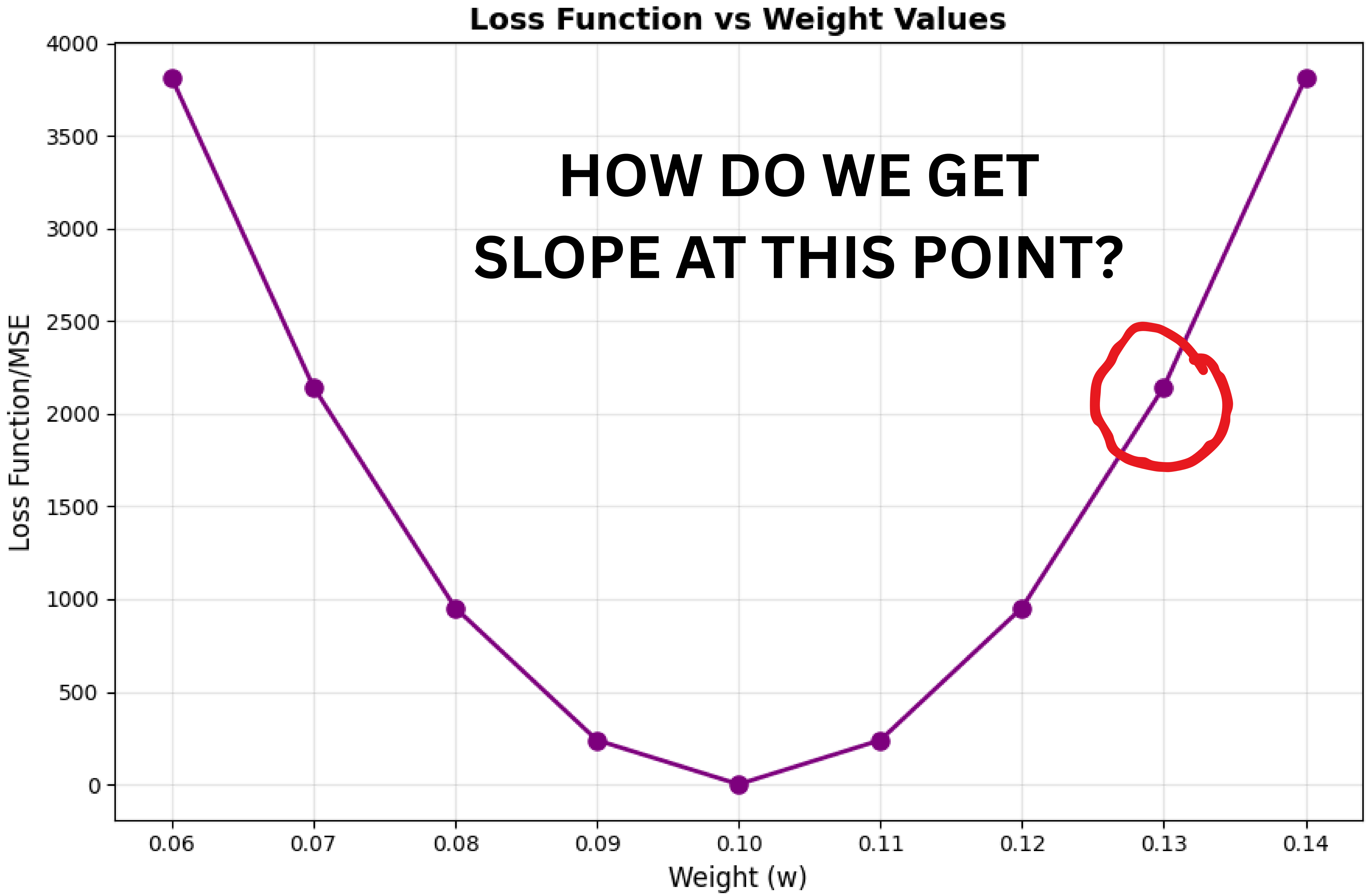

Loss Function vs Weight Values
Loss Function/MSE
Weight (w)
4000
3500
3000
2500
2000
1500
1000
500
0
0.06
0.07
0.08
0.09
0.10
0.11
0.12
0.13
0.14
θ=20°


Loss Function vs Weight Values
Loss Function/MSE
Weight (w)
4000
3500
3000
2500
2000
1500
1000
500
0
0.06
0.07
0.08
0.09
0.10
0.11
0.12
0.13
0.14

he(weight) can’t go down further, Gradient Descent is stopped and the **PERFECT value of weights are found!!**

now what is slope in terms of calculus??

DERIVATIVE!!


Loss Function vs Weight Values
HOW DO WE GET
SLOPE AT THIS POINT?
Loss Function/MSE
4000
3500
3000
2500
2000
1500
1000
500
0
0.06
0.07
0.08
0.09
0.10
0.11
0.12
0.13
0.14
Weight (w)

# derivative of Loss w.r.t Weight

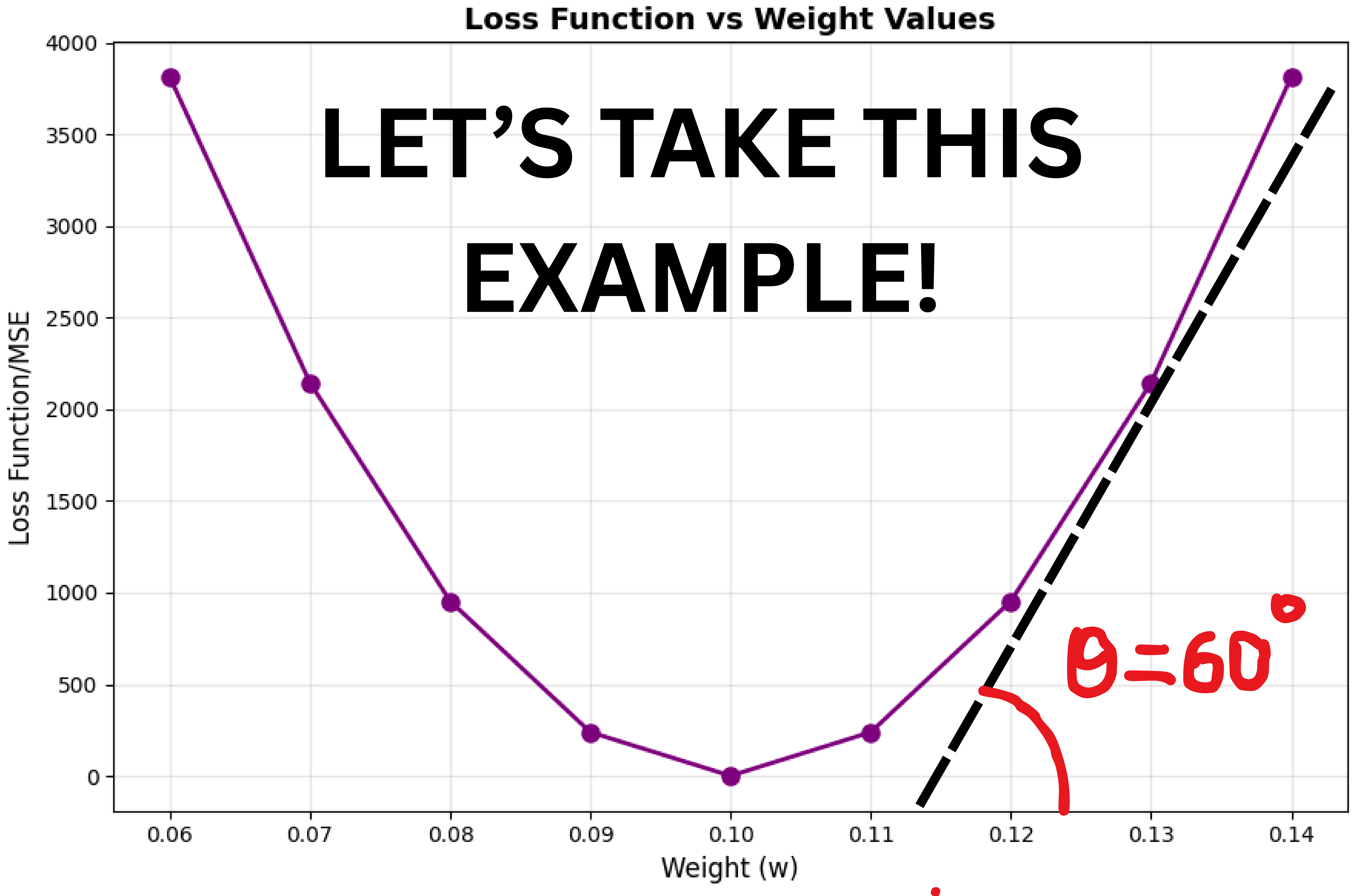
$$\frac{dL}{dw}$$

# how weights get updated

$$w := w - \alpha \frac{dL}{dw}$$

$\alpha$ = learning rate (step size)

$\frac{dL}{dw}$= partial derivatives of loss


Loss Function vs Weight Values
LET'S TAKE THIS EXAMPLE!
Loss Function/MSE
4000
3500
3000
2500
2000
1500
1000
500
0
0.06
0.07
0.08
0.09
0.10
0.11
0.12
0.13
0.14
Weight (w)
θ=60°

# Now calculate for w=0.13

## alpha=0.01

$\frac{dL}{dw}$=tan(60)

$$w := w - \alpha \frac{dL}{dw}$$

## find the derivative of L w.r.t to w and b

$$L(w,b) = \frac{1}{n}\sum_{i=1}^{n}(y_i - (wx_i + b))^2$$

$$\frac{\partial L}{\partial w} = \frac{-2}{n} \sum_{i=1}^{n} x_i (y_i - (wx_i + b))$$

- **Measures how loss changes when w changes.**
- **If gradient > 0 → loss increases → move left.**
- **If gradient < 0 → loss decreases → move right.**

# similarly for b

$$\frac{\partial L}{\partial b} = \frac{-2}{n} \sum_{i=1}^{n} (y_i - (wx_i + b))$$

# replace them here

$$w := w - \alpha \frac{\partial L}{\partial w}$$

$$b := b - \alpha \frac{\partial L}{\partial b}$$

$$\frac{\partial L}{\partial w} = \frac{-2}{n} \sum_{i=1}^{n} x_i(y_i - (wx_i + b))$$

$$\frac{\partial L}{\partial b} = \frac{-2}{n} \sum_{i=1}^{n} (y_i - (wx_i + b))$$

# code

```
for epoch in range(epochs):
    y_pred = w * X + b
    dw = (-2/n) * np.sum(X * (y - y_pred))
    db = (-2/n) * np.sum(y - y_pred)
    w = w - alpha * dw
    b = b - alpha * db
```

# hands on session