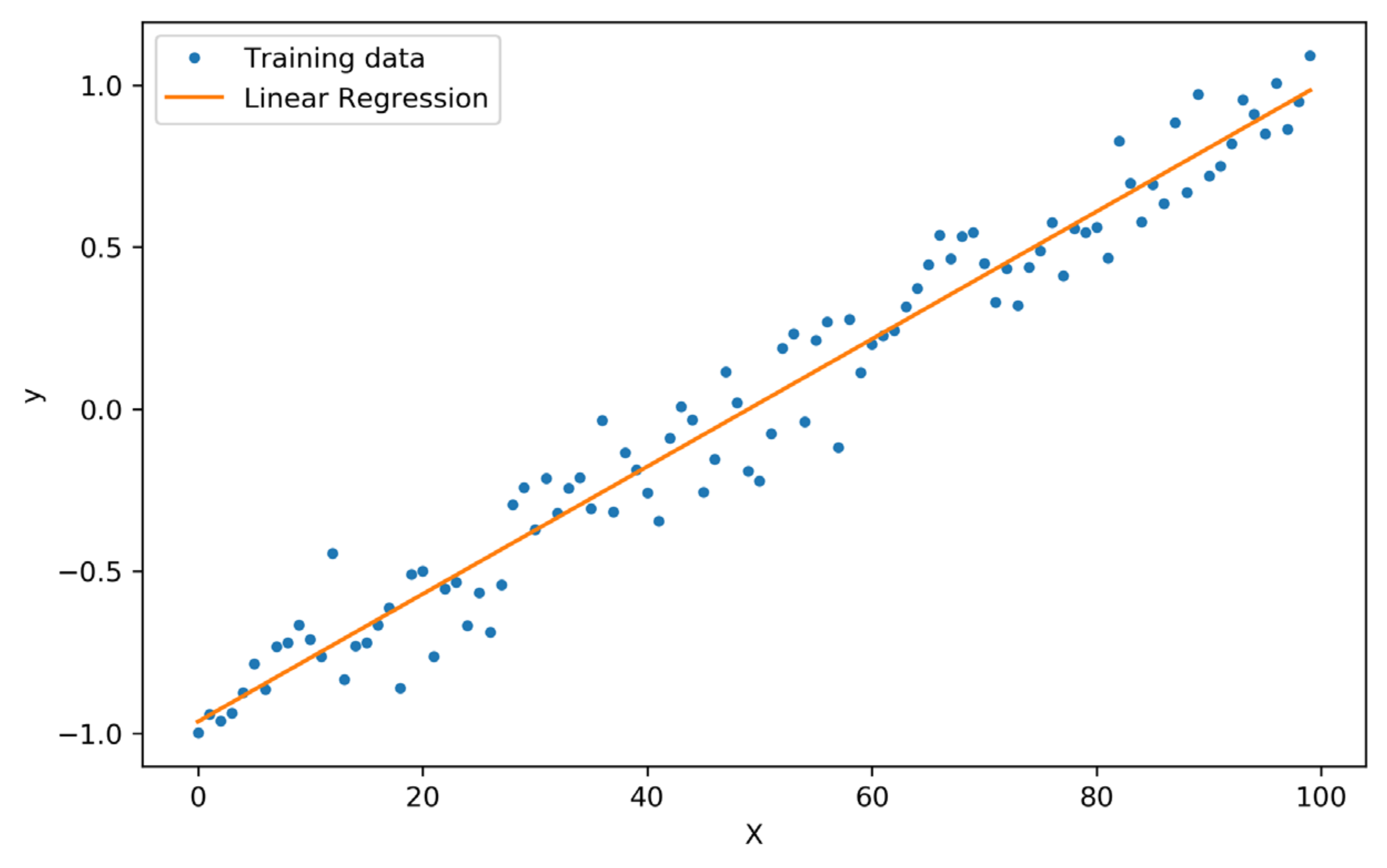
LINEAR REGRESSION

MOHAMMAD GHODDOSI

REGRESSION


Training data
Linear Regression
1.0
0.5
0.0
−0.5
−1.0
y
0
20
40
60
80
100
X

LINE EQUATION

- Standard form

$$ax + by = c$$

- Point-Slope Form

$$y - y1 = m(x - x1)$$

- Intercept Form

$$\frac{x}{a} + \frac{y}{b} = 1$$

- Slope-Intercept Form

$$y = mx + b$$

LINE EQUATION

- Standard form (3 variables)

$$ax + by = c$$

- Point-Slope Form (3 variables)

$$y - y1 = m(x - x1)$$

- Intercept Form (2 variables)

$$\frac{x}{a} + \frac{y}{b} = 1$$

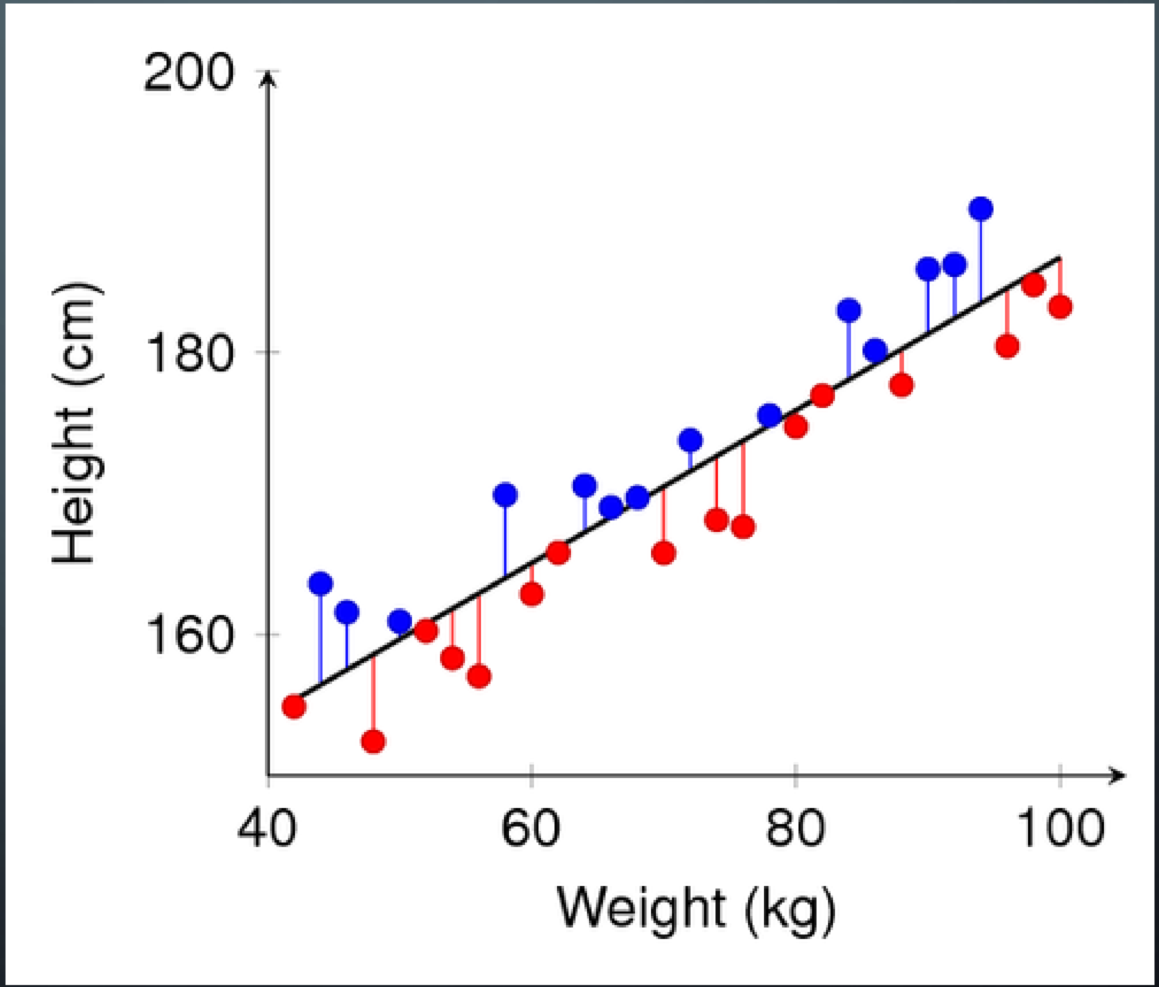
- Slope-Intercept Form (2 variables)

$$y = mx + b$$

LINE EQUATION

- If we have more than one feature, use all:
- We want to predict y using x1, x2, x3, x4
  - $\mathrm{y} = w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 + b$
- If we have n features:
  - $\mathrm{y} = \sum_{i=1}^{n} w_i x_i + b$

MSE COST FUNCTION



MSE COST FUNCTION

- We need a performance measure (Tom Mitchell)
- We can use idea of euclidean distance
  - 2d : $d(P,Q) = \sqrt{(P_x - Q_x)^2 + \left(P_y - Q_y\right)^2}$
  - Md: $d(P,Q) = \sqrt{(P_1 - Q_1)^2 + (P_2 - Q_2)^2 + \ \dots (P_M - Q_M)^2}$

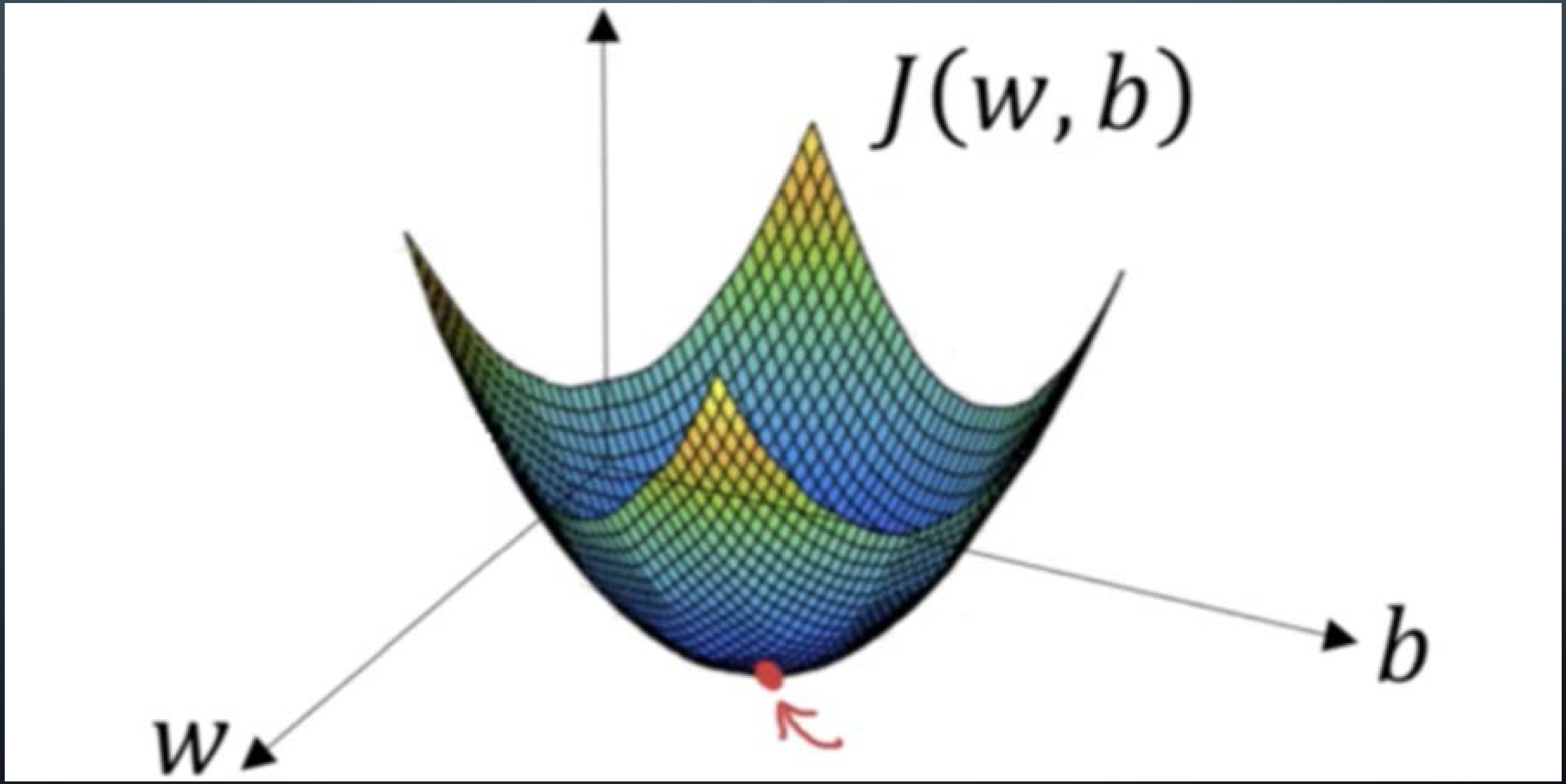
MSE COST FUNCTION

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

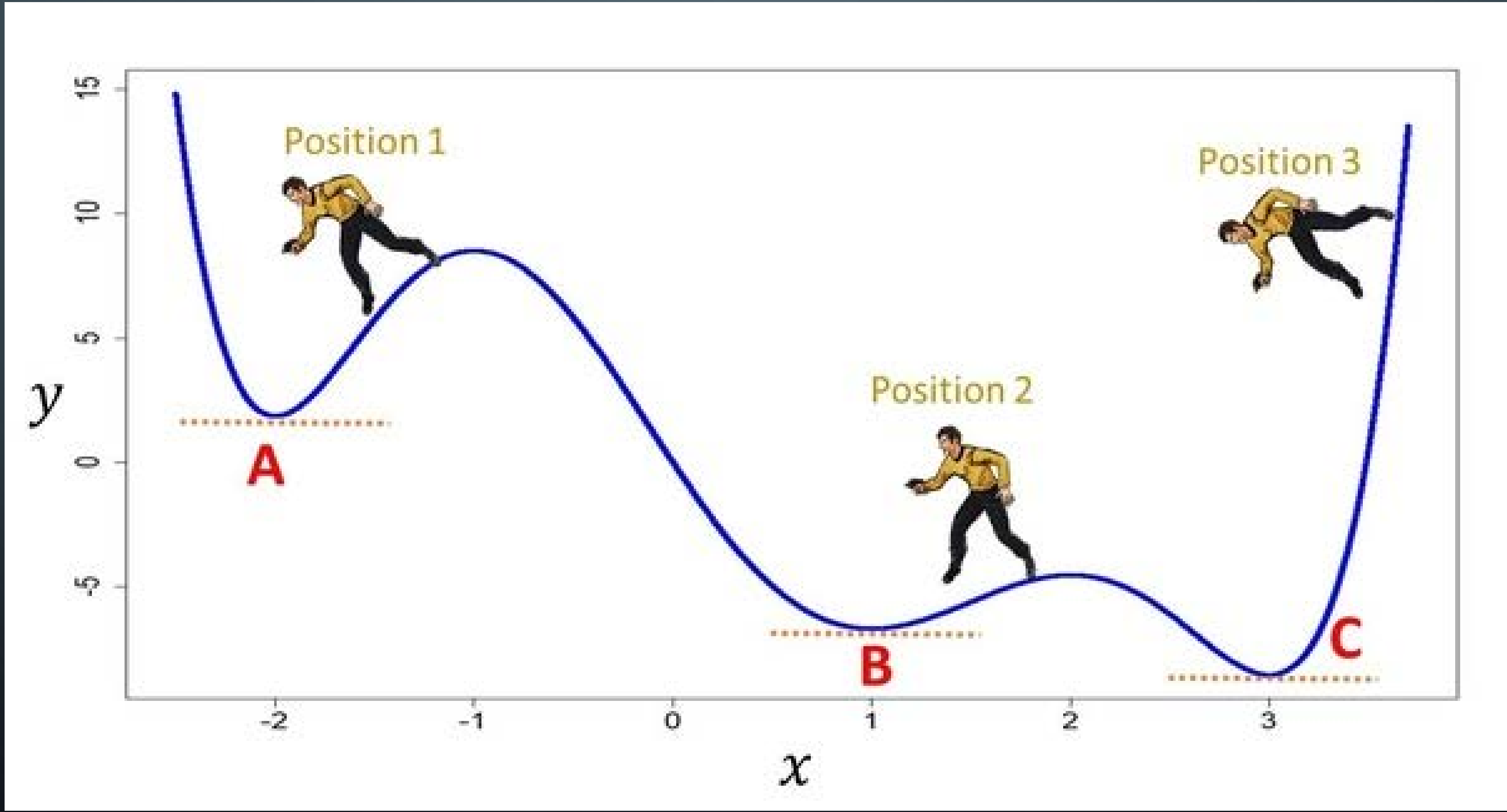
$$MSE = \frac{1}{2n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2$$

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

OPTIMIZATION



OPTIMIZATION AND DERIVATIVES (2D)



OPTIMIZATION AND DERIVATIVES (2D)

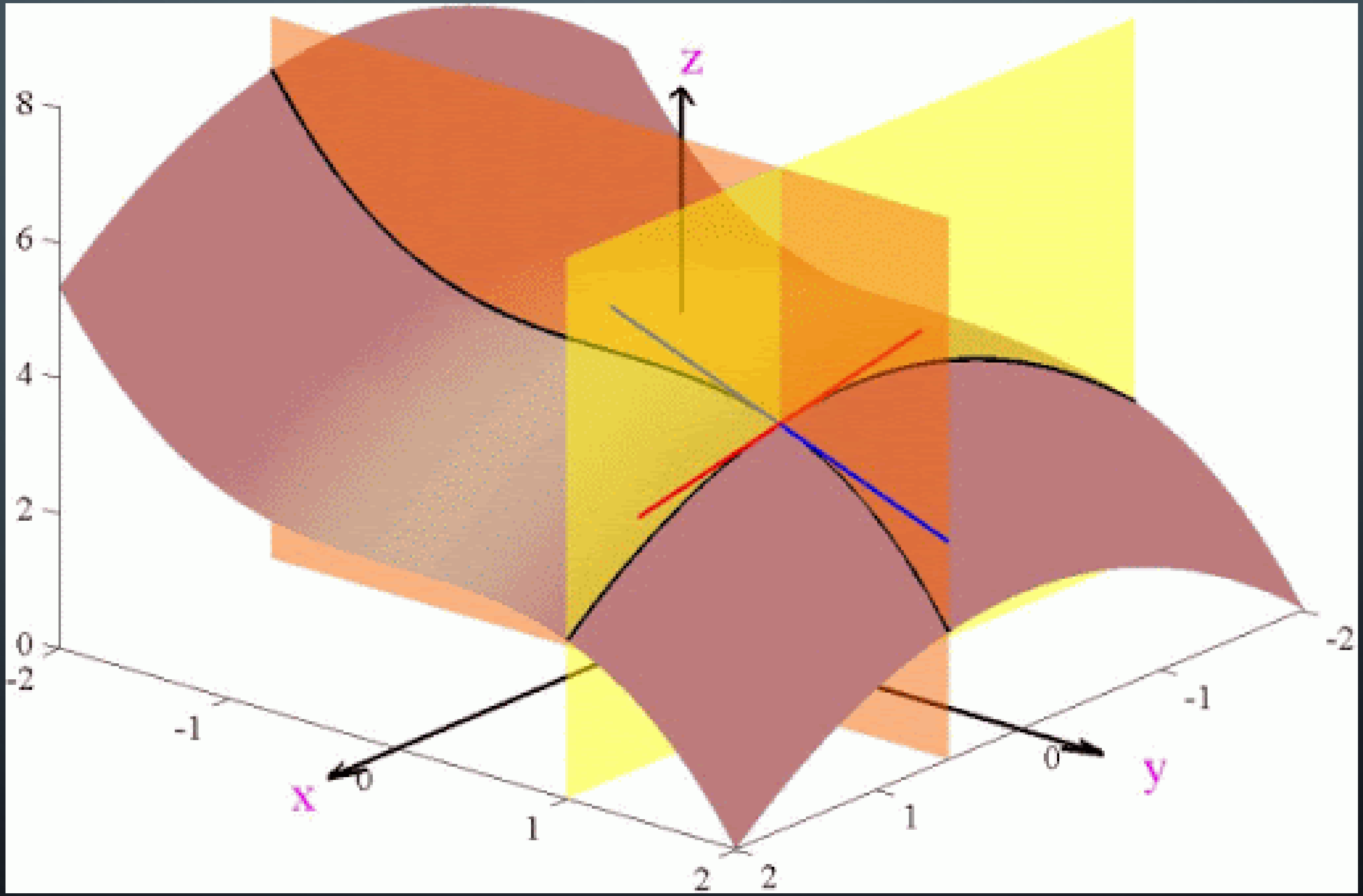
$$f'(x) = \lim_{\Delta x \to 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

$$\lim_{\Delta x \to 0} \Delta x \ * f'(x) = \lim_{\Delta x \to 0} f(x + \Delta x) - f(x)$$

$$\Delta x \ * f'(x_0) \approx f(x_0 + \Delta x) - f(x_0)$$

$$\begin{cases} \Delta f_{x_0} < 0 & \qquad \Delta x \ * f'(x_0) < 0 \\ \Delta f_{x_0} > 0 & \qquad \Delta x \ * f'(x_0) > 0 \end{cases}$$

OPTIMIZATION AND DERIVATIVES (ND)



OPTIMIZATION AND DERIVATIVES (ND)

$$\boxed{\nabla f = \frac{\partial f}{\partial x_1} \quad \frac{\partial f}{\partial x_2} \quad \dots \quad \frac{\partial f}{\partial x_3}}$$

$$f(\vec{X}_0 + \vec{u}) \approx f(\vec{X}_0) + \nabla f(\vec{X}_0).\,\vec{u}$$

$$\|f(\vec{X}_0 + \vec{u}) - f(\vec{X}_0)\| \approx \|\nabla f(\vec{X}_0)\,.\,\vec{u}\ \|$$

$$\|f(\vec{X}_0 + \vec{u}) - f(\vec{X}_0)\| \approx \|\nabla f(\vec{X}_0)\| * \|\vec{u}\| * \cos(\theta_{X_0,u})$$

jupyter

WHY MSE + GD WORKS?

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

Repeat until convergence {

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

}

VECTOR DOT PRODUCT

$$\vec{a}.\vec{b} = [a_1 \quad a_2 \quad \dots \quad a_n].[b_1 \quad b_2 \quad \dots \quad b_n]$$

$$= a_1 b_1 + a_2 b_2 + \dots + a_n b_n$$

$$= \sum_{i=1}^{n} a_i b_i$$

MATRIX DOT PRODUCT

$$X.Y = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1k} \\ X_{21} & X_{22} & \dots & X_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n1} & X_{n2} & \cdots & X_{nk} \end{bmatrix} . \begin{bmatrix} Y_{11} & Y_{12} & \cdots & Y_{1m} \\ Y_{21} & Y_{22} & \dots & Y_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{k1} & Y_{k2} & \cdots & Y_{km} \end{bmatrix}$$

MATRIX DOT PRODUCT

$$
\begin{array}{cc}
\begin{array}{c} \overrightarrow{x_1} \\ \overrightarrow{x_2} \\ \vdots \\ \overrightarrow{x_n} \end{array}
&
\begin{bmatrix}
X_{11} & X_{12} & \cdots & X_{1k} \\
X_{21} & X_{22} & \cdots & X_{2k} \\
\vdots & \vdots & \ddots & \vdots \\
X_{n1} & X_{n2} & \cdots & X_{nk}
\end{bmatrix}
\end{array}
\cdot
\begin{array}{c}
\begin{array}{cccc} \overrightarrow{y_1} & \overrightarrow{y_2} & \cdots & \overrightarrow{y_m} \end{array} \\
\begin{bmatrix}
Y_{11} & Y_{12} & \cdots & Y_{1m} \\
Y_{21} & Y_{22} & \cdots & Y_{2m} \\
\vdots & \vdots & \ddots & \vdots \\
Y_{k1} & Y_{k2} & \cdots & Y_{km}
\end{bmatrix}
\end{array}
$$

MATRIX DOT PRODUCT

$$\begin{bmatrix} \overrightarrow{x_1}.\overrightarrow{y_1} & \overrightarrow{x_1}.\overrightarrow{y_2} & \cdots & \overrightarrow{x_1}.\overrightarrow{y_m} \\ \overrightarrow{x_2}.\overrightarrow{y_1} & \overrightarrow{x_2}.\overrightarrow{y_2} & \ldots & \overrightarrow{x_2}.\overrightarrow{y_m} \\ \vdots & \vdots & \ddots & \vdots \\ \overrightarrow{x_n}.\overrightarrow{y_1} & \overrightarrow{x_n}.\overrightarrow{y_2} & \cdots & \overrightarrow{x_n}.\overrightarrow{y_m} \end{bmatrix}$$

LINEAR REGRESSION AS VECTOR MULTIPLICATION

$$\mathrm{y} = w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 + b$$

$$\mathrm{y} = w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 + w_5$$

$$\mathrm{y} = \langle w_1 \quad w_2 \quad w_3 \quad w_4 \quad w_5 \rangle \,.\, \langle x_1 \quad x_2 \quad x_3 \quad x_4 \quad 1 \rangle$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} x_{1,1} & \cdots & x_{M,1} & 1 \\ x_{1,2} & \dots & x_{M,2} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ x_{1,N} & \cdots & x_{M,N} & 1 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_M \end{bmatrix}$$

LINEAR REGRESSION AS VECTOR MULTIPLICATION

$$H = X.W^T$$

$$J = \frac{1}{2N}(Y - H)^T.(Y - H)$$

$$\nabla J = \frac{1}{N}(Y - H)^T.X$$

jupyter