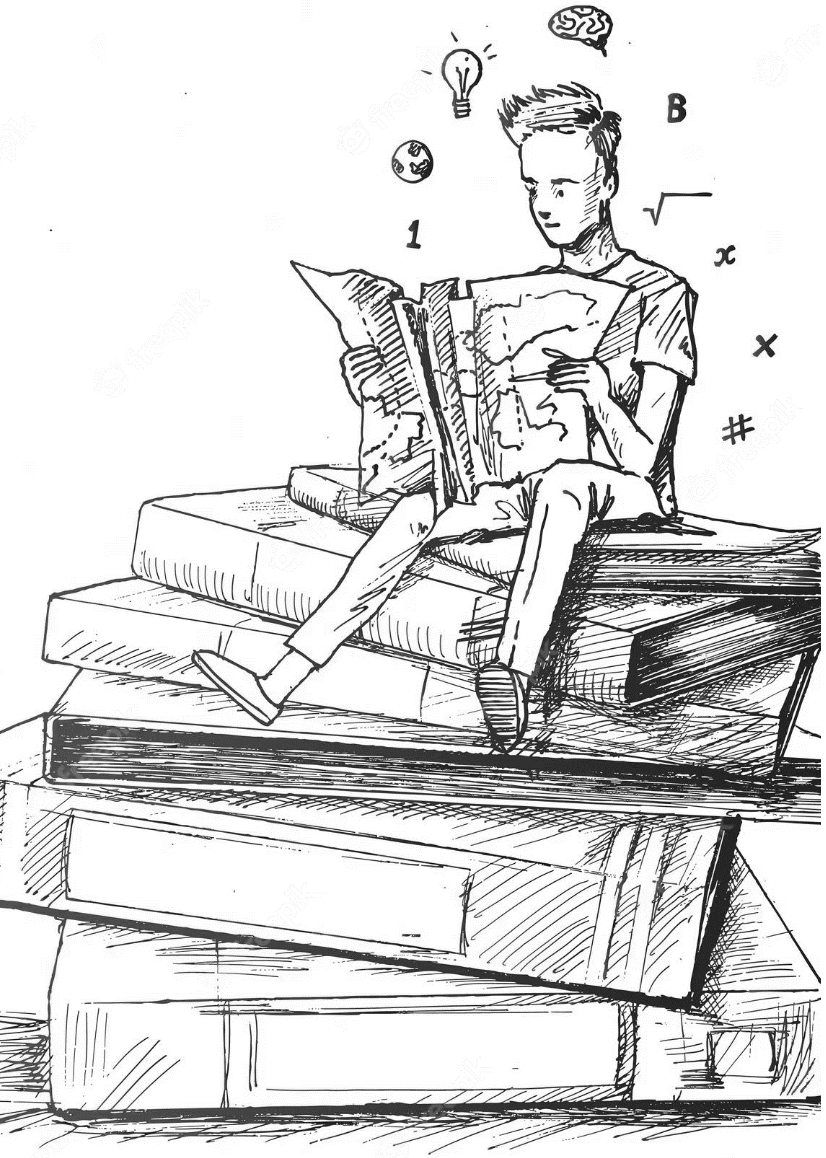
Indonesia AI
AI for Everyone, AI for Indonesia

x Ilma Aliya Fiddien

Mathematics in Deep Learning

Backward Pass

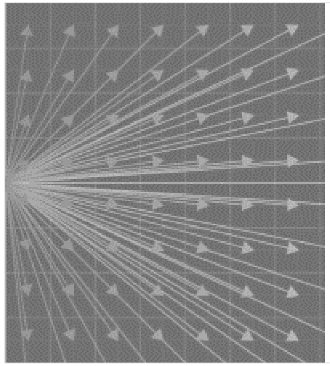
in Feedforward Neural Network



Learning Objective

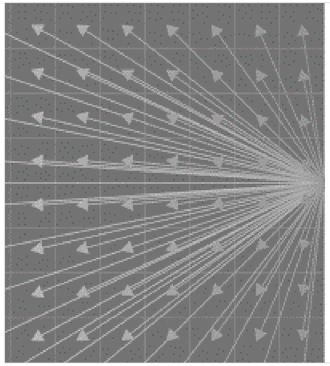
Understand the essential mathematical concepts to gain **a deeper understanding** of the underlying algorithm of artificial neural networks (ANN)

Outline

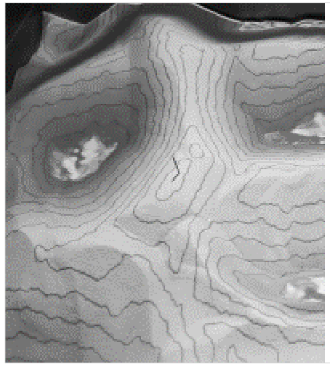


Revise:
Forward Pass

Weights & biases
Tensor operations

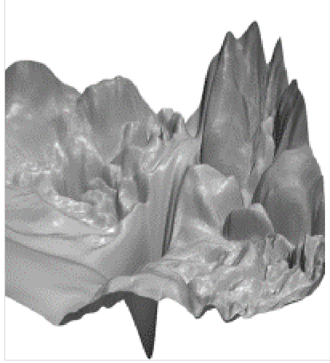


Overview:
Backward Pass



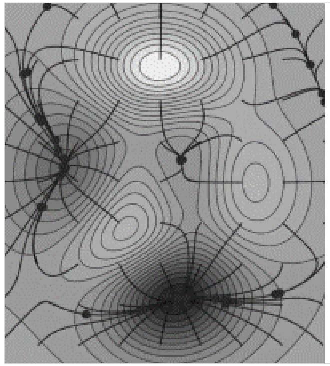
Differential Calculus

Derivative | Partial Derivatives
Gradient | Jacobian
Chain Rule
Extreme Points



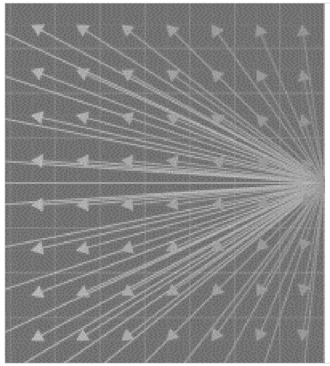
Cost Function

Loss Function
Error Function

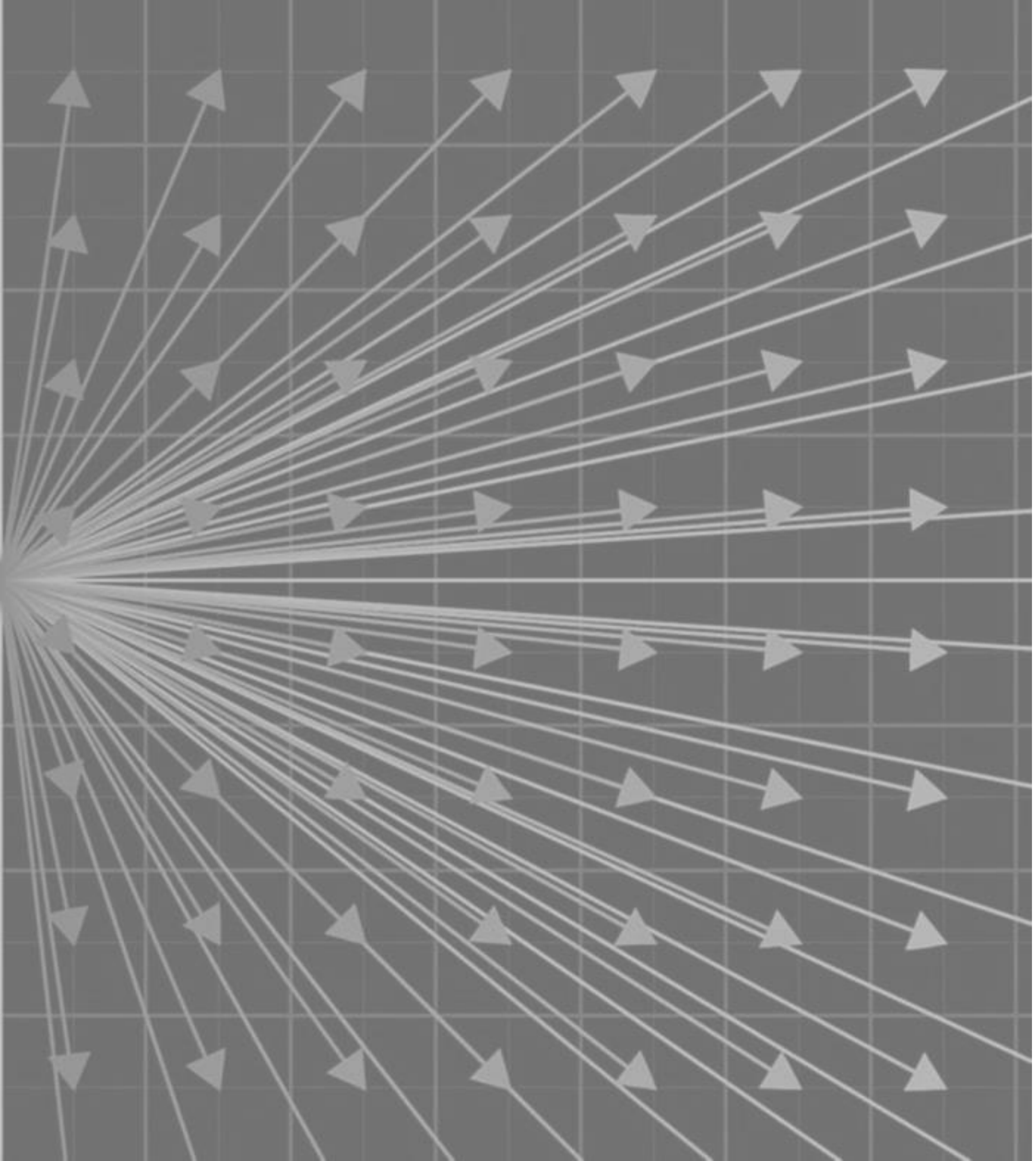


Gradient Descent

& Stochastic Gradient Descent



Backward Pass

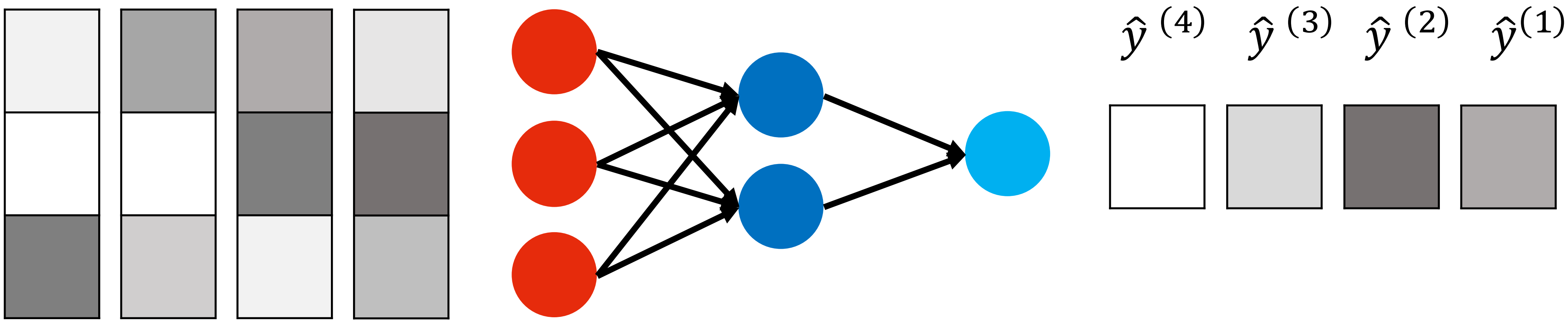


Revise: **Forward Pass**

Weights & biases

Tensor operations

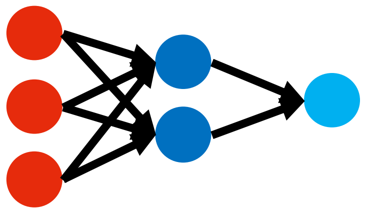
Forward Pass →



$$\sigma_1(X^{(n)} \cdot W_1 + b_1) = A_1$$

$$\sigma_2(A_1 \cdot W_2 + b_2) = A_2 = \hat{y}^{(n)}$$

Tensor Operations

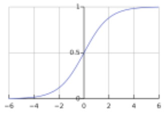


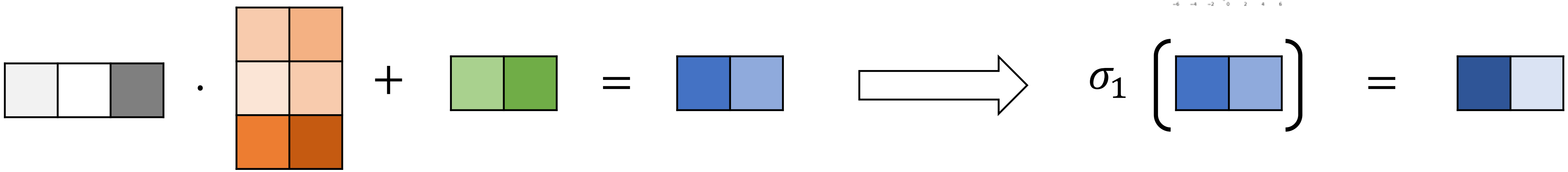
$$a_1(X^{(n)} \cdot W_1 + b_1) = A_1$$

dim$(X^{(4)})$ =
(1, 3)

dim(W_1) =
(3,2)

dim(b_1) =
(1,2)



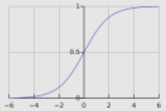


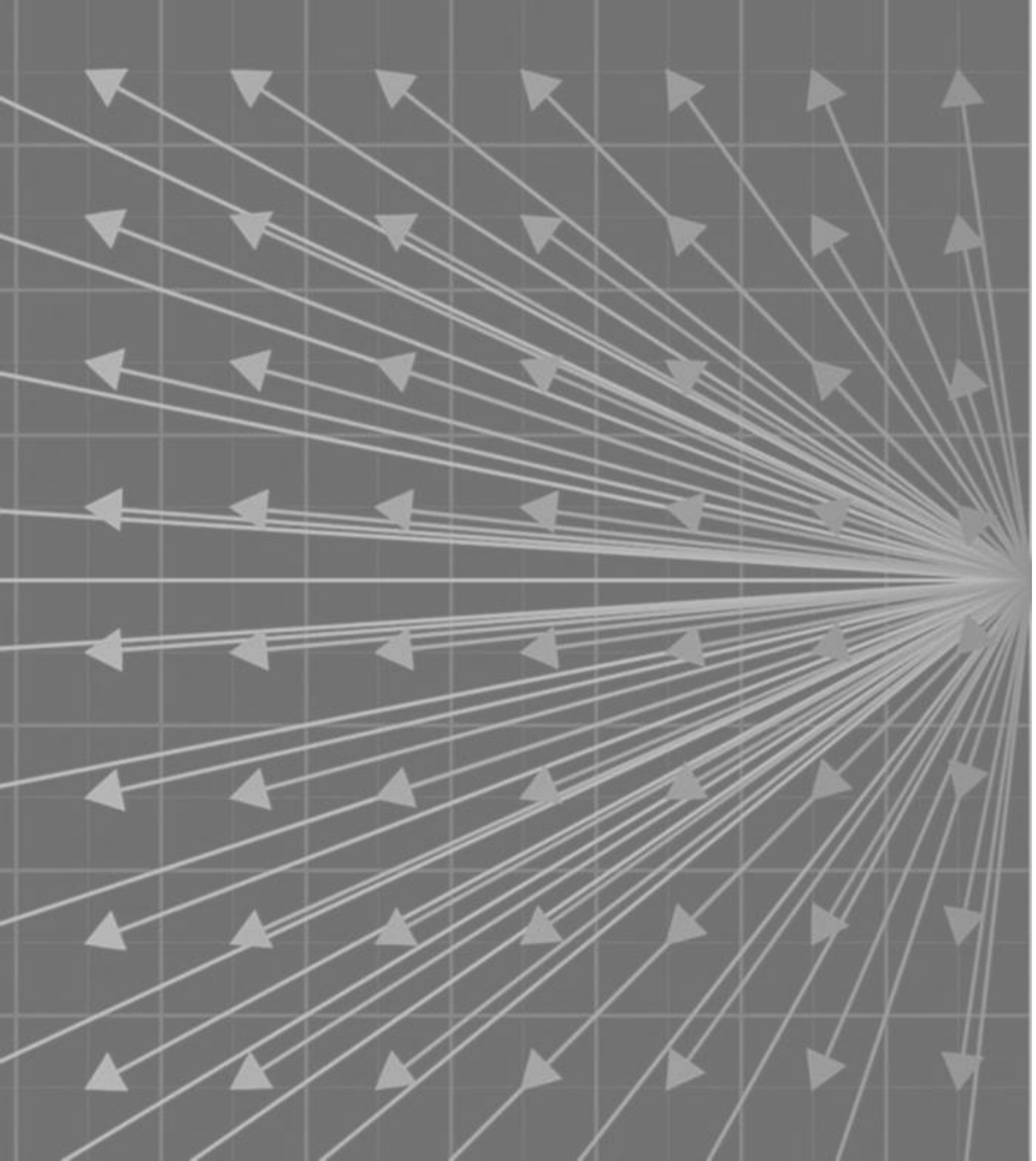
$$a_2(A_1 \cdot W_2 + b_2) = A_2 = \hat{y}^{(n)}$$

dim(A_1) =
(1, 2)

dim(W_2) =
(2,1)

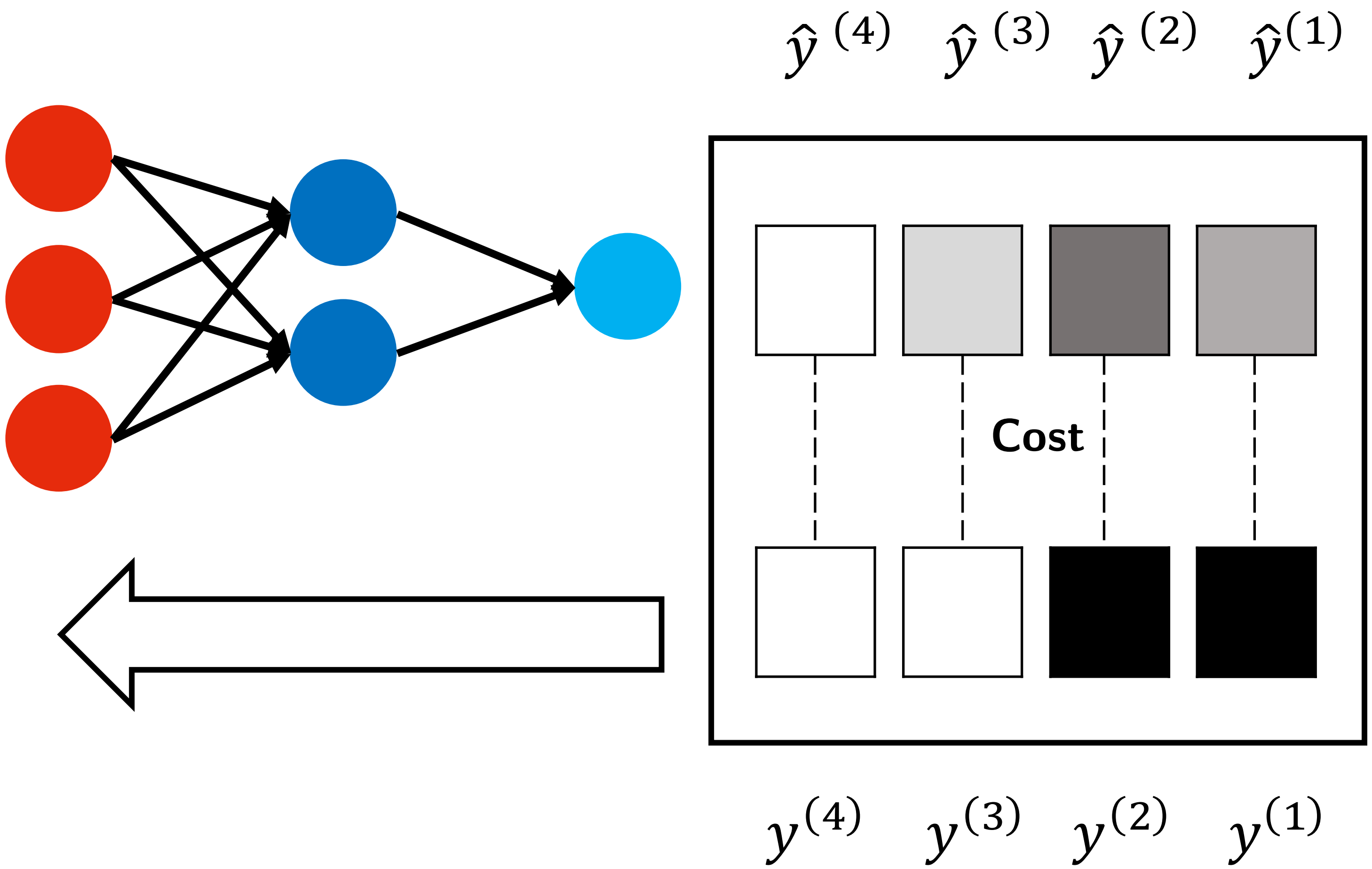
dim(b_2) =
(1,1)



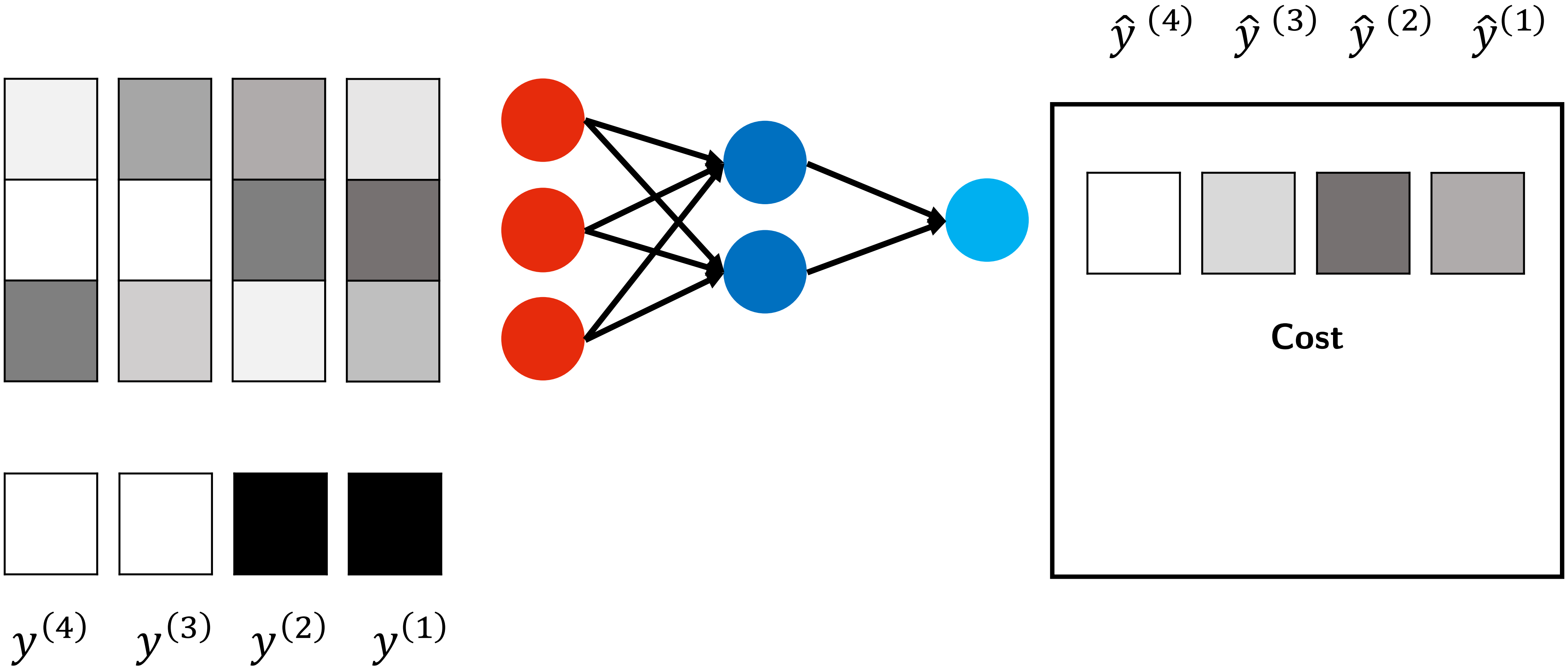


Overview: **Backward Pass**

Backward Pass ←



Backward Pass ←



Gradient Descent

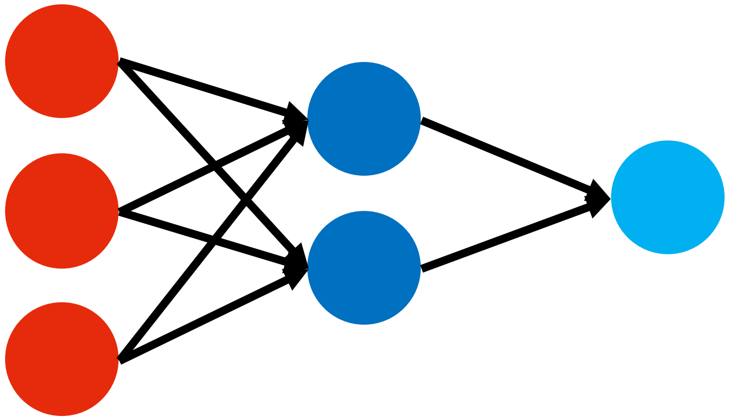
Parameter Update

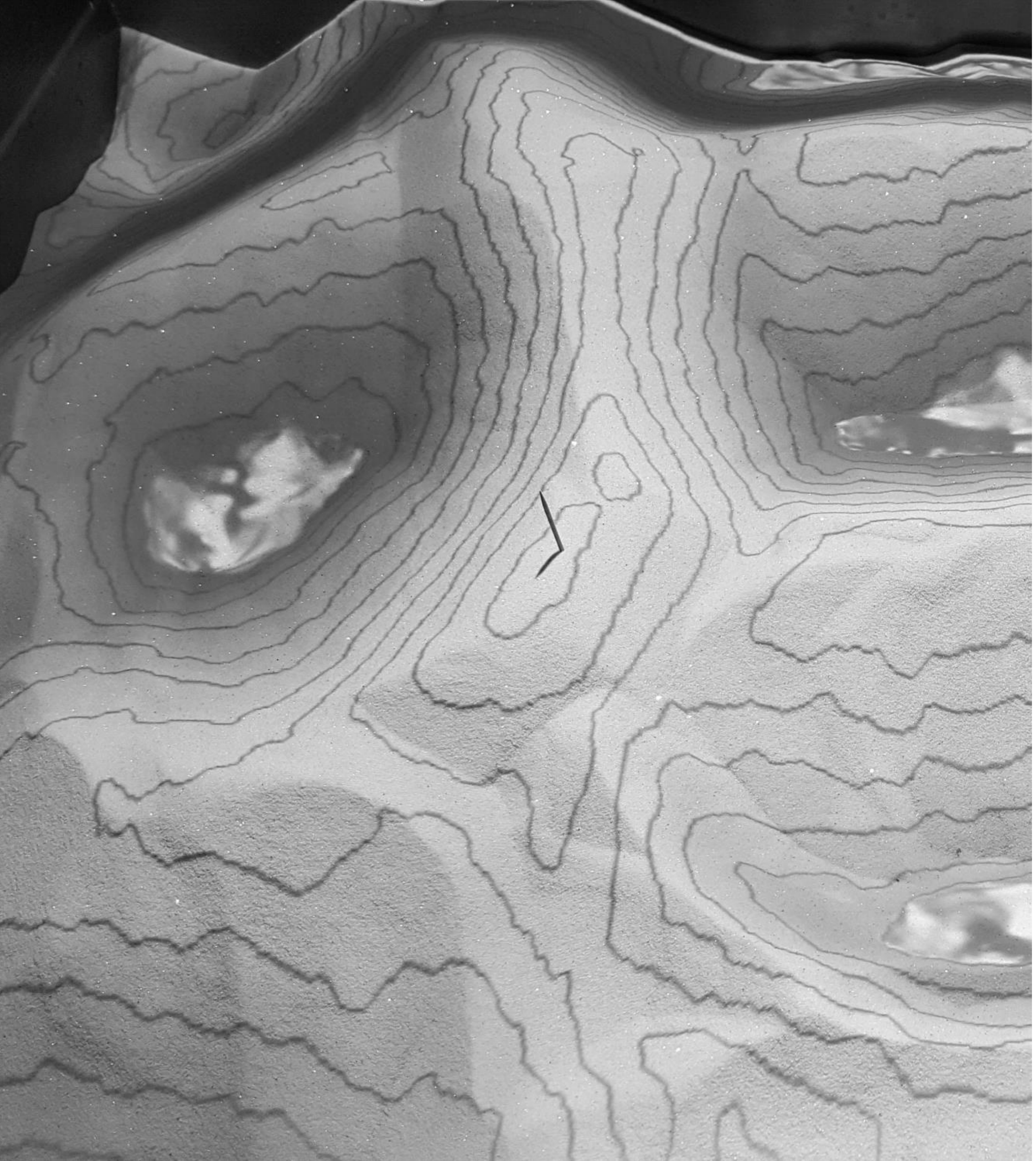
$$b_2 \leftarrow b_2 - \alpha \frac{\partial}{\partial b_2} Cost(\hat{y}, y)$$

$$W_2 \leftarrow W_2 - \alpha \frac{\partial}{\partial W_2} Cost(\hat{y}, y)$$

$$b_1 \leftarrow b_1 - \alpha \frac{\partial}{\partial b_1} Cost(\hat{y}, y)$$

$$W_1 \leftarrow W_1 - \alpha \frac{\partial}{\partial W_1} Cost(\hat{y}, y)$$





Differential Calculus

Derivative | Partial Derivatives

Gradient | Jacobian

Chain Rule

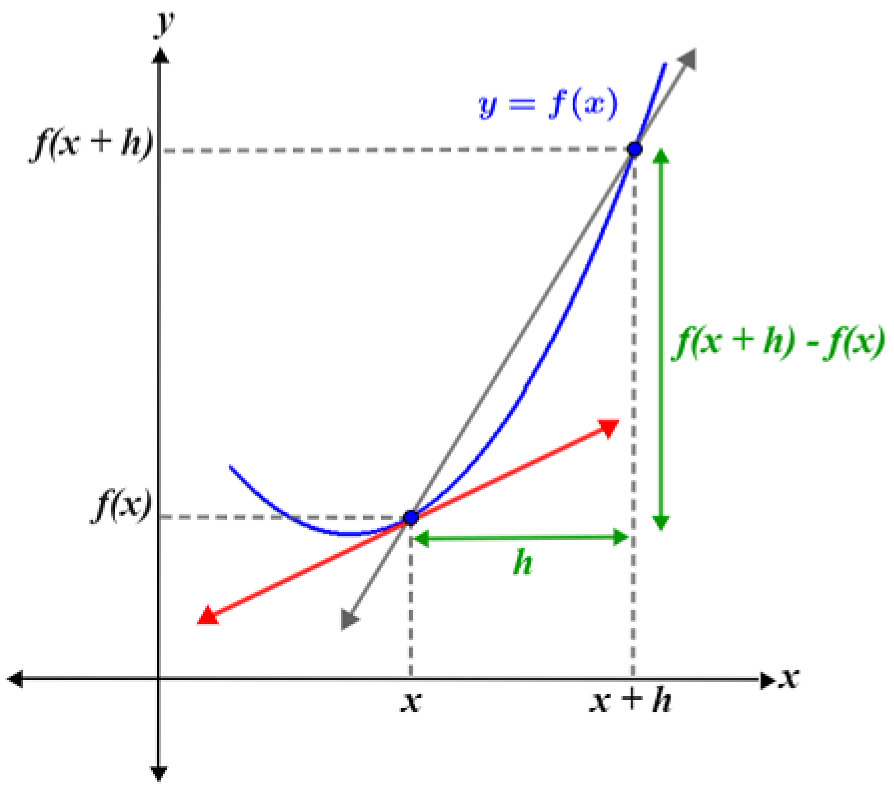
Extreme Points

Turunan (*derivative*)

$$f(x)$$

Definisi turunan dari f:

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



Contoh 1

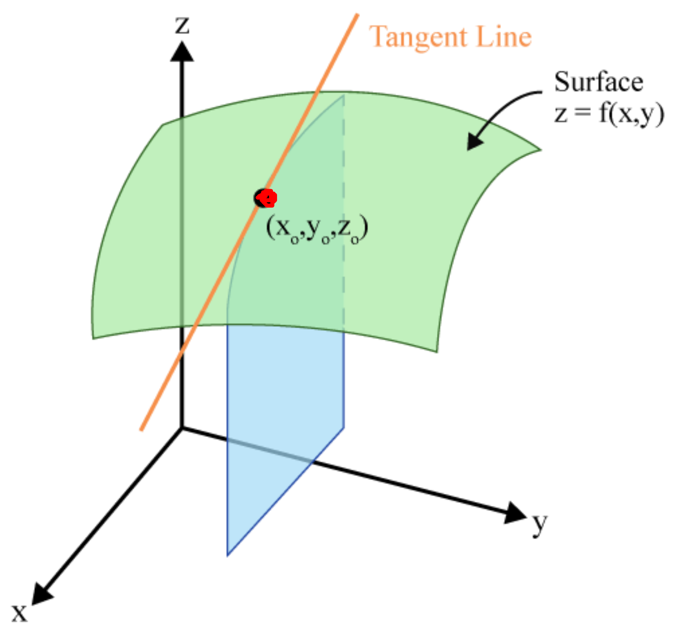
$$f(x) = x^2 + 2x^4 + 3$$

Turunan orde 1: $\frac{df}{dx} = 2x^{2-1} + 8x^{4-1} + 0 = 2x + 8x^3$

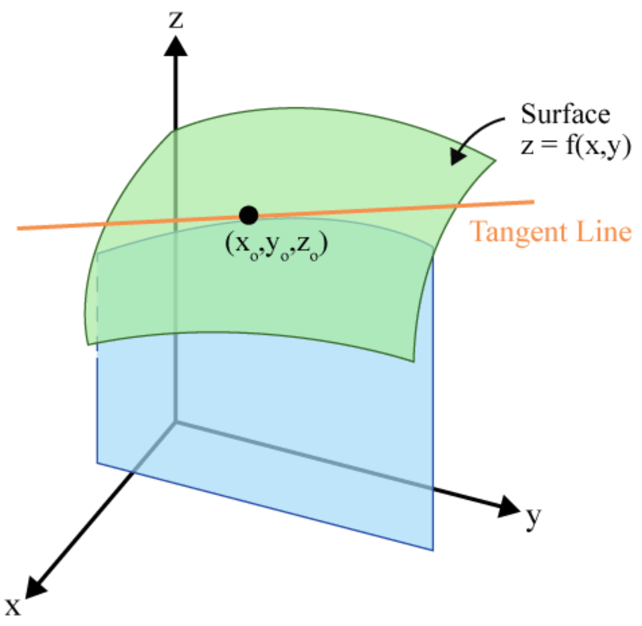
Turunan f di $x = 2$ $\frac{d}{dx}f(3) = 2(2) + 8(2)^3 = 68$

Turunan orde 2: $\frac{d^2f}{dx^2} = 2 + 24x^2$

Turunan parsial (*partial derivative*)



Slope of the surface in the x-direction



Slope of the surface in the y-direction

Calcworkshop.com

Sumber gambar: Calc Workshop

Contoh 2

$$f(x, y) = x^2 + 3y^4$$

Turunan parsial orde 1: $\frac{\partial f}{\partial x} = 2x$ $\frac{\partial f}{\partial y} = 12y^3$

Turunan parsial f terhadap x di $(3, 1)$: $\frac{\partial}{\partial x} f(3,1) = 2(3) = 6$

Turunan parsial f terhadap y di $(3, 1)$: $\frac{\partial}{\partial y} f(3,1) = 12(1)^3 = 12$

Turunan parsial orde 2: $\frac{\partial^2 f}{\partial x^2} = 2$ $\frac{\partial^2 f}{\partial y^2} = 36y^2$

Gradient suatu fungsi

$$f\colon \mathbb{R}^2 \to \mathbb{R}$$
$$\nabla f\colon \mathbb{R}^2 \to \mathbb{R}^2$$

generalisasi

$$g\colon \mathbb{R}^m \to \mathbb{R}$$
$$\nabla g\colon \mathbb{R}^{\mathrm{m}} \to \mathbb{R}^m$$

Gradient:

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

Penulisan lain:

$$\nabla f = \frac{\partial f}{\partial x}\hat{\imath} + \frac{\partial f}{\partial y}\hat{\jmath}$$

Gradient:

$$\nabla g = \begin{bmatrix} \dfrac{\partial g}{\partial x_1} \\ \vdots \\ \dfrac{\partial g}{\partial x_m} \end{bmatrix}$$

Gradient suatu fungsi di suatu titik

$$f\colon \mathbb{R}^2 \to \mathbb{R}$$
$$\nabla f\colon \mathbb{R}^2 \to \mathbb{R}^2$$

generalisasi

$$g\colon \mathbb{R}^m \to \mathbb{R}$$
$$\nabla g\colon \mathbb{R}^m \to \mathbb{R}^m$$

Gradient f di titik (x, y) :

$$\nabla f(x,y) = \begin{bmatrix} \frac{\partial f}{\partial x}(x,y) \\ \frac{\partial f}{\partial y}(x,y) \end{bmatrix}$$

Gradient f di titik p:

$$\nabla g = \begin{bmatrix} \frac{\partial g}{\partial x_1} \\ \vdots \\ \frac{\partial g}{\partial x_m} \end{bmatrix}$$

dengan

$$p = (x_1, \dots, x_m)$$

Contoh 3

Gradient $f(x, y) = x^2 + 3y^4$ di titik (1,2):

$$\nabla f(1{,}2)$$

$$= 2(1)\hat{\imath} + 12(2)^3\hat{\jmath}$$

$$= \begin{bmatrix} 2(1) \\ 12(2)^3 \end{bmatrix} = \begin{bmatrix} 2 \\ 96 \end{bmatrix}$$

Jacobian suatu fungsi

$$f\colon \mathbb{R}^2 \to \mathbb{R}^2$$
$$\nabla f\colon \mathbb{R}^2 \to \mathbb{R}^{2\times 2}$$

generalisasi ⟹

$$g\colon \mathbb{R}^m \to \mathbb{R}^n$$
$$\nabla g\colon \mathbb{R}^{\mathrm{m}} \to \mathbb{R}^{m\times n}$$

Gradient:

$$\nabla f = \begin{bmatrix} \frac{\partial}{\partial x_1} f_1 & \frac{\partial}{\partial x_1} f_2 \\ \frac{\partial}{\partial x_2} f_1 & \frac{\partial}{\partial x_2} f_2 \end{bmatrix}$$

Gradient:

$$\nabla g = \begin{bmatrix} \frac{\partial}{\partial x_1} g_1 & \cdots & \frac{\partial}{\partial x_1} g_n \\ \vdots & \ddots & \vdots \\ \frac{\partial}{\partial x_m} g_1 & \cdots & \frac{\partial}{\partial x_m} g_n \end{bmatrix}$$

Jacobian suatu fungsi di suatu titik

$$f\colon \mathbb{R}^2 \to \mathbb{R}^2$$
$$\nabla f\colon \mathbb{R}^2 \to \mathbb{R}^{2\times 2}$$

generalisasi

$$g\colon \mathbb{R}^m \to \mathbb{R}^n$$
$$\nabla g\colon \mathbb{R}^{\mathrm{m}} \to \mathbb{R}^{m\times n}$$

Gradient:

$$\nabla f(x,y) = \begin{bmatrix} \frac{\partial}{\partial x_1} f_1(x,y) & \frac{\partial}{\partial x_1} f_2(x,y) \\ \frac{\partial}{\partial x_2} f_1(x,y) & \frac{\partial}{\partial x_2} f_2(x,y) \end{bmatrix}$$

Gradient:

$$\nabla g(p) = \begin{bmatrix} \frac{\partial}{\partial x_1} g_1(p) & \cdots & \frac{\partial}{\partial x_1} g_n(p) \\ \vdots & \ddots & \vdots \\ \frac{\partial}{\partial x_m} g_1(p) & \cdots & \frac{\partial}{\partial x_m} g_n(p) \end{bmatrix}$$

dengan

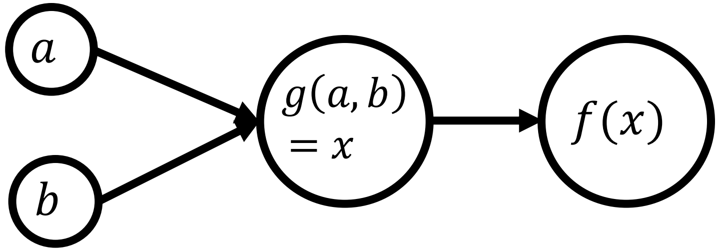
$$p = (x_1, \ldots, x_m)$$

Komposisi fungsi

$$g(a, b) = a + 2b$$

$$f(x) = x^3$$

Komposisi fungsi g lalu f:



$$\frac{\partial f}{\partial a} = ?$$

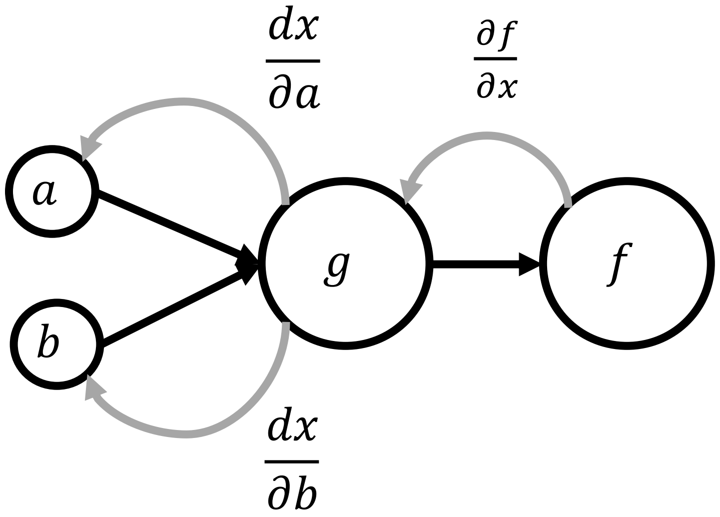
$$\frac{\partial f}{\partial b} = ?$$

Aturan rantai (*chain rule*)

Turunan parsial f terhadap a:

$$\frac{\partial f}{\partial a} = \frac{\partial f}{\partial x}\frac{dx}{\partial a} = \frac{\partial^2 f}{\partial x \partial a}$$

$$= 3x^2(1) = 3x^2$$



Turunan parsial f terhadap b:

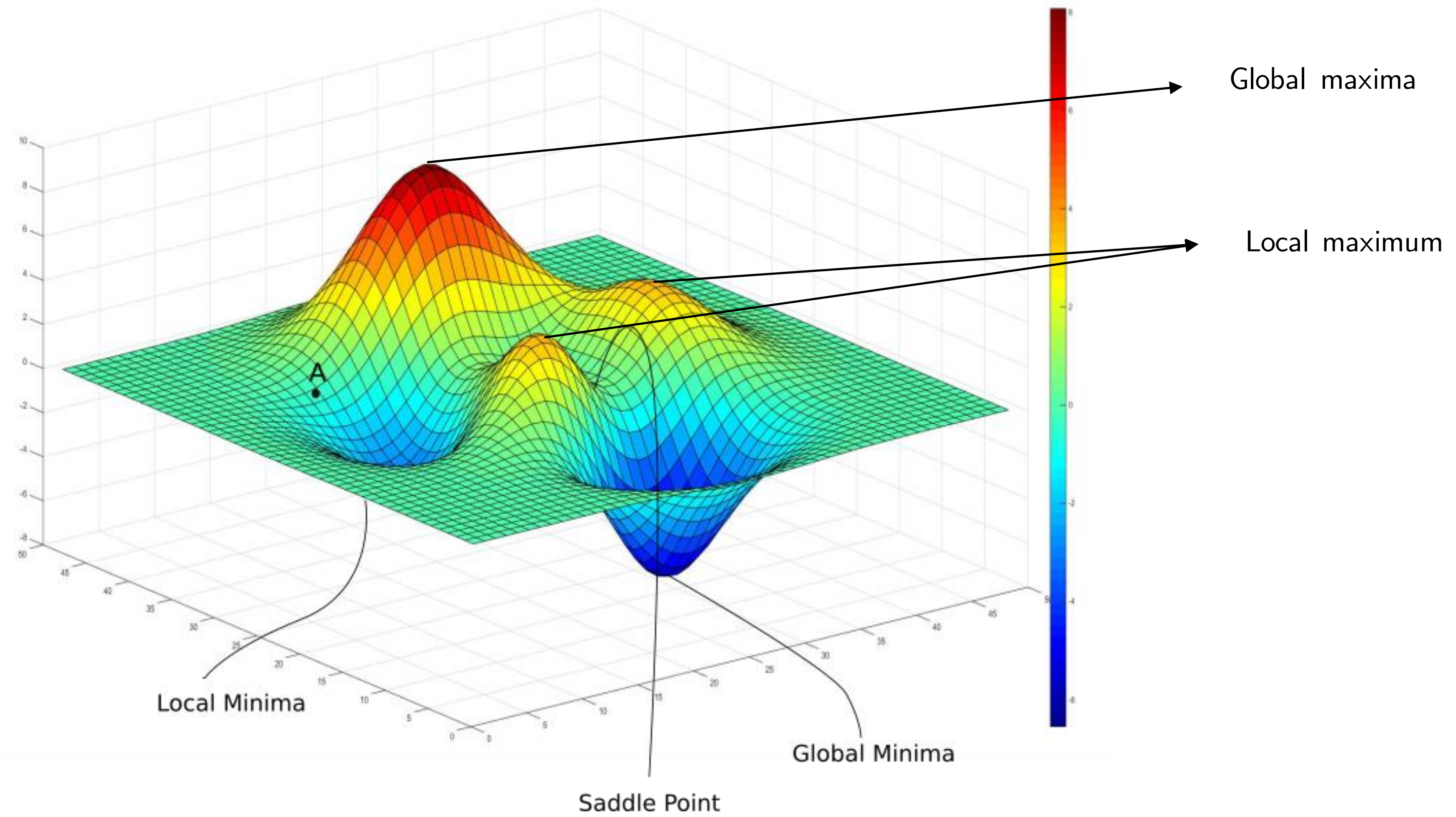
$$\frac{\partial f}{\partial b} = \frac{\partial f}{\partial x}\frac{dx}{\partial b} = \frac{\partial^2 f}{\partial x \partial b}$$

$$= 3x^2(2) = 6x^2$$

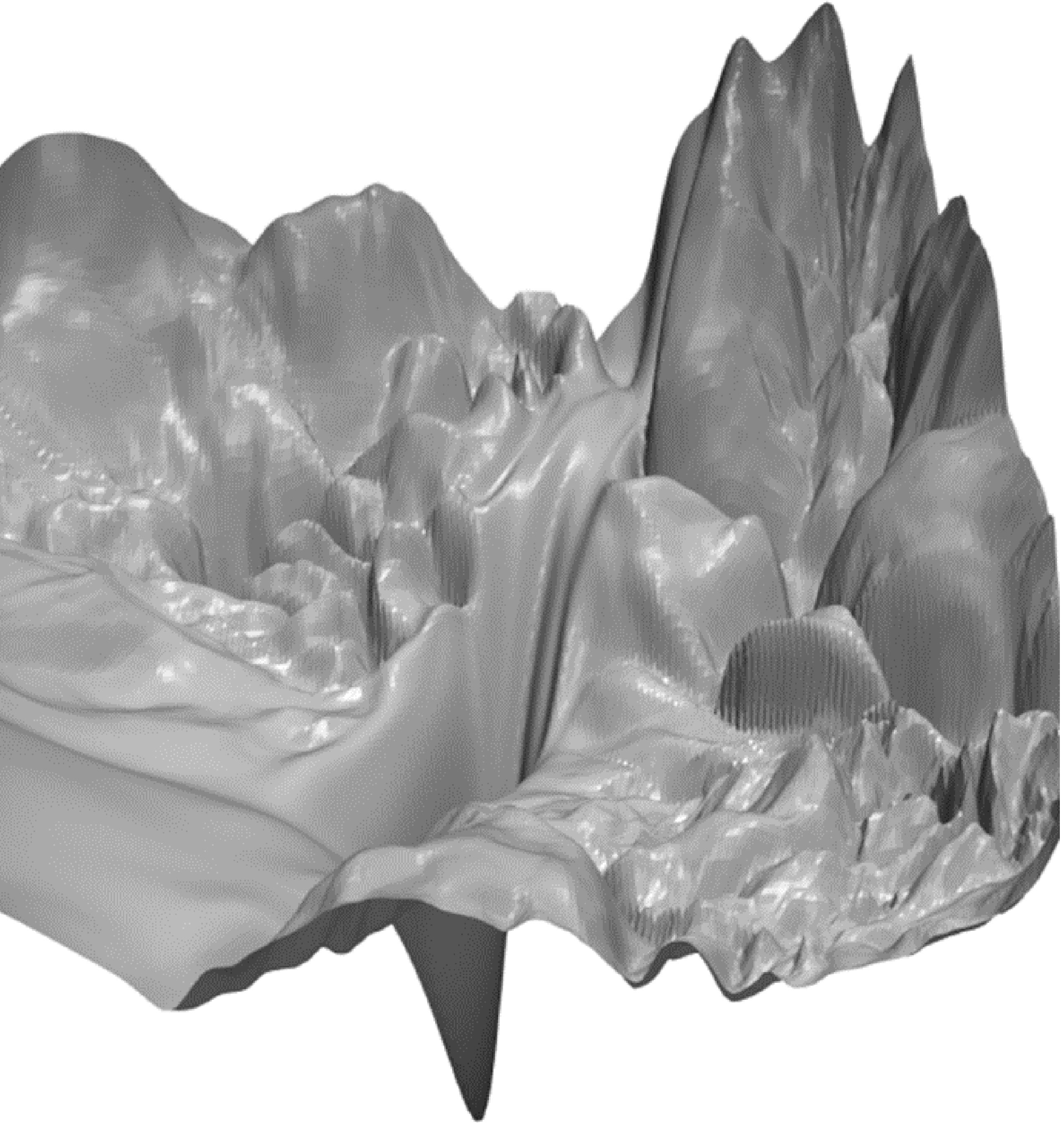
$$x = g(a, b) = a + 2b \qquad f(x) = x^3$$

$$\frac{\partial x}{\partial a} = 1 \quad \frac{\partial x}{\partial b} = 2 \qquad \frac{\partial f}{\partial x} = \frac{df}{dx} = 3x^2$$

Extreme points



Sumber gambar: Paperspace



Cost Function

Loss Function

Error Function

Cost Function vs Evaluation Metrics

Cost Function

- Mengevaluasi model **ketika** proses "belajar"
- Digunakan untuk mempelajari hubungan antara input dan output

Evaluation Metrics

- Mengevaluasi model **di luar** proses "belajar"
- Digunakan untuk mengevaluasi seberapa baik hubungan antara input dan output yang telah dipelajari

Karakteristik Umum Cost Function

- Mengevaluasi model **ketika** proses "belajar"
  - v.s. *evaluation metrics*: mengevaluasi model **di luar** proses "belajar"
- Digunakan untuk mempelajari hubungan antara input dan output
  - Hanya melibatkan variabel y dan $\hat{y}$
- Fungsi yang **kontinu secara global*** dan **turunannya terdefinisikan**

Beberapa Contoh Cost Function

- Mean Absolute Error (MAE) atau L1 Loss
- Mean Squared Error (MSE) atau L2 Loss
- Root Mean Squared Error (RMSE)
- Binary Cross-Entropy Loss

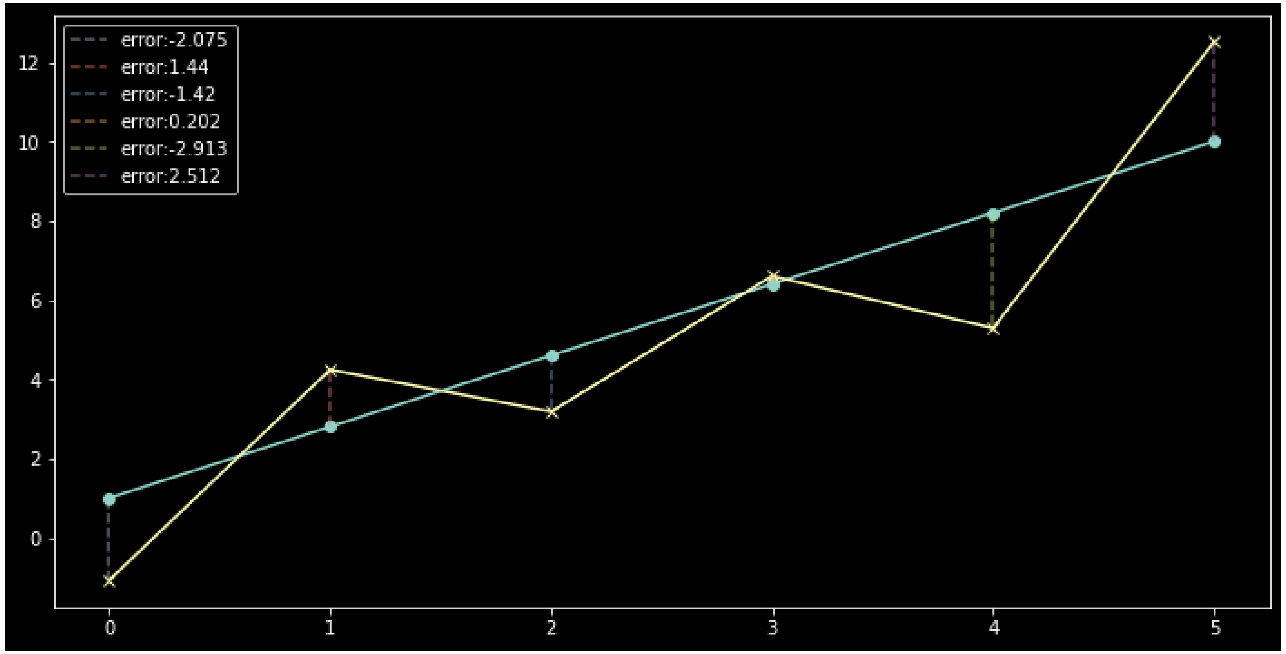
“Error”

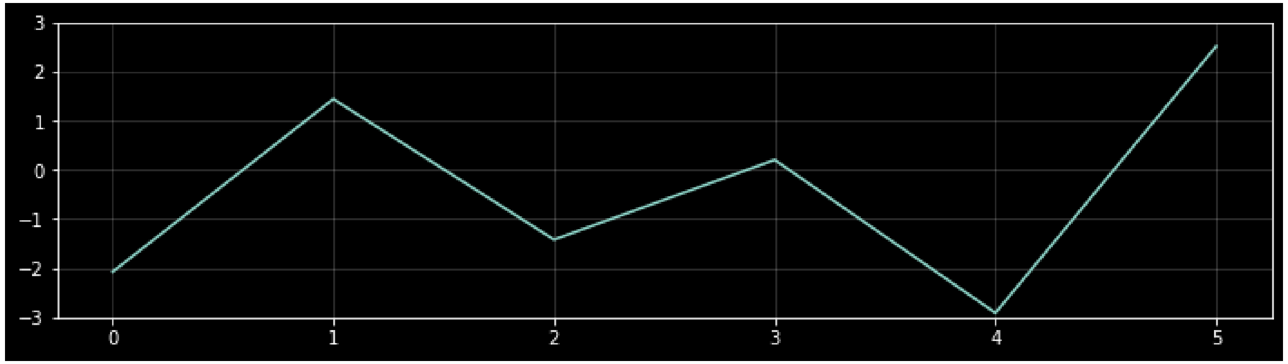
Selisih antara output asli dan output prediksi

$$error = \hat{y} - y$$

Mean Error

$$ME(y, \hat{y}) = \frac{1}{N}\sum_{i=1}^{N}(\hat{y} - y)$$



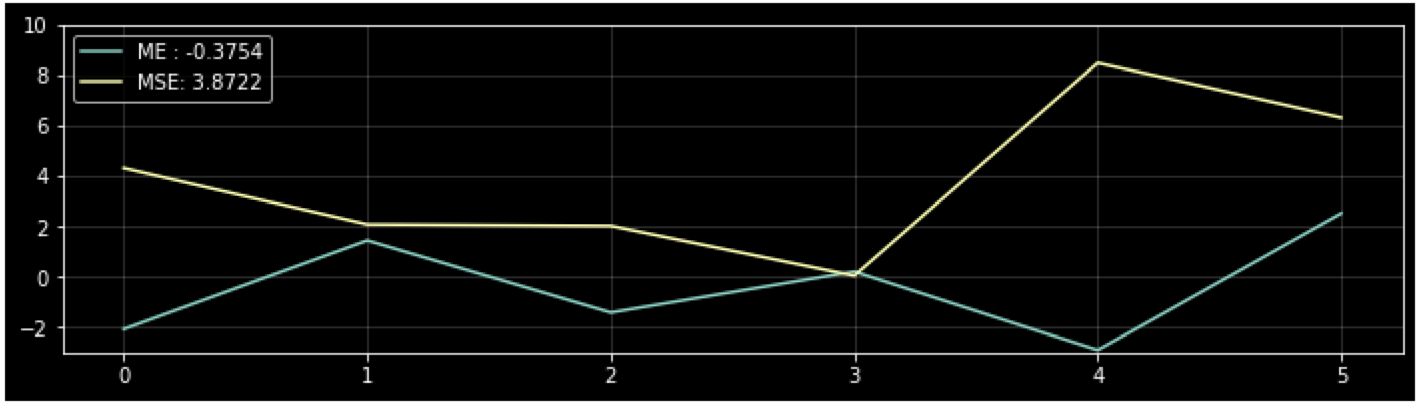


Mean Squared Error (MSE)

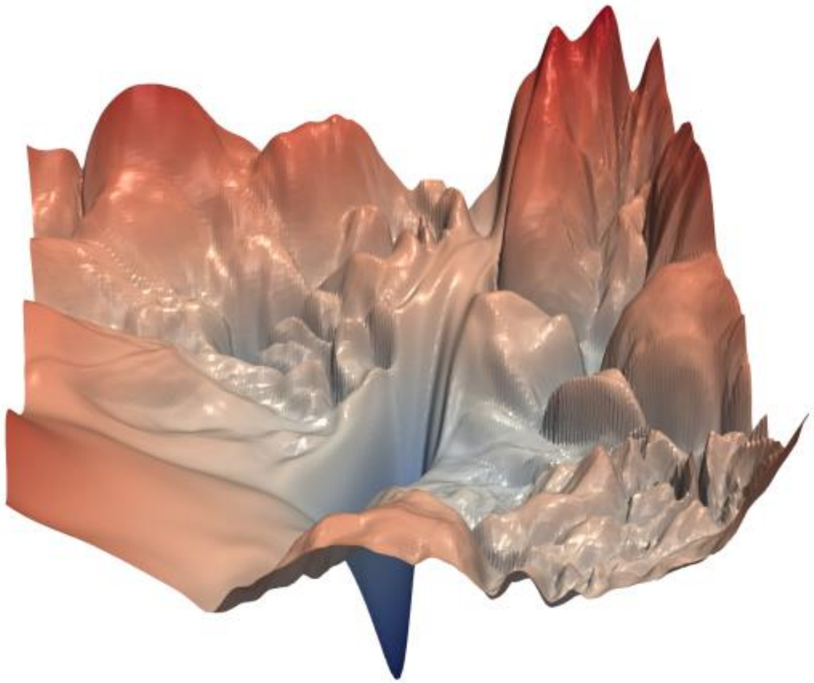
- Error negatif dan error positif tidak saling menghabiskan
- Memberikan penalti yang lebih besar untuk data outlier

$$MSE(y,\hat{y}) = \frac{1}{N}\sum_{i=1}^{N}\left(y^{(i)} - \hat{y}^{(i)}\right)^2$$

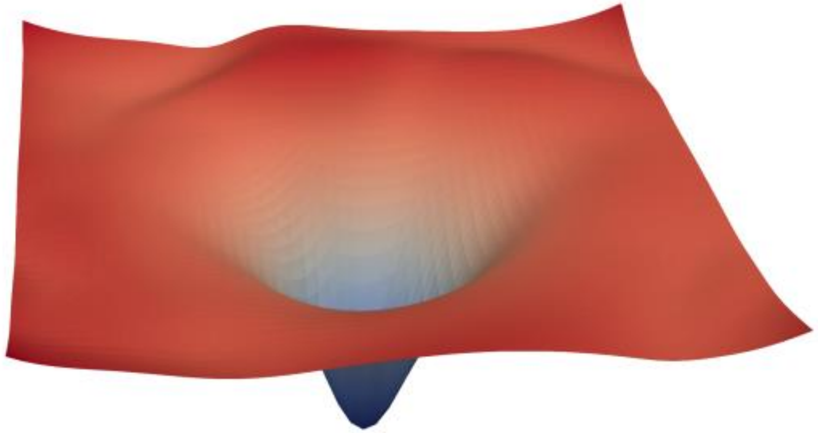
y	$\hat{y}$	E	SE
1	0.8	-0.2	0.04
1	0.9	-0.1	0.01
1	1.1	0.1	0.01
1	1.3	0.3	0.09
MSE			0.0375



Cost function surface



Permukaan dengan banyak perubahan kontur
(cenderung tidak stabil)



Permukaan dengan sedikit perubahan kontur
(lebih stabil)

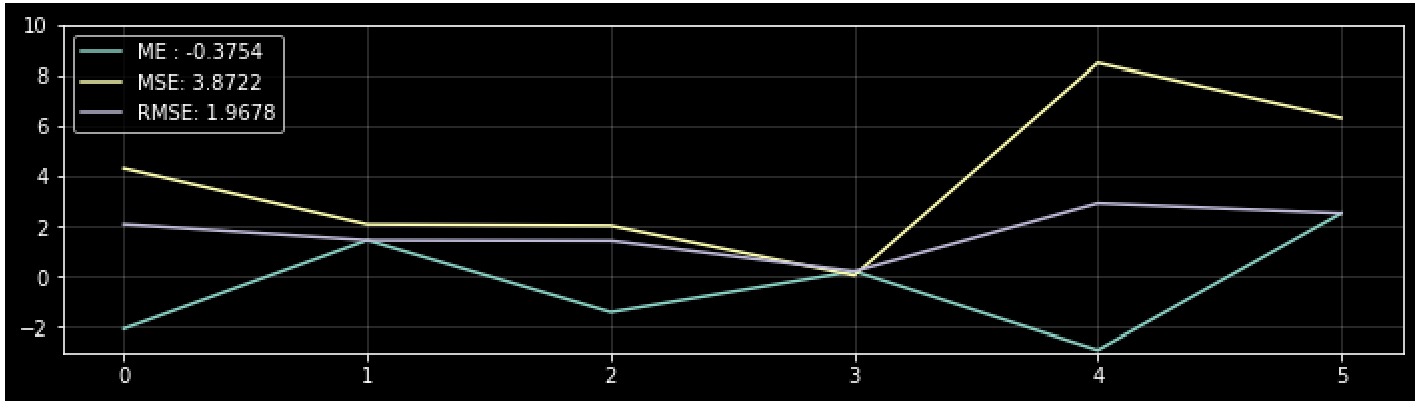
Sumber gambar: Visualizing the Loss Landscape of Neural Nets (Li et. al., 2018)

Root Mean Squared Error (RMSE)

- Error negatif dan error positif tidak saling menghabiskan
- Memberikan penalti yang sedikit lebih besar untuk data outlier
- Mengembalikan satuan error ke satuan asli data

$$RMSE(y, \hat{y}) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y^{(i)} - \hat{y}^{(i)})^2}$$

y	$\hat{y}$	E	SE
1	0.8	-0.2	0.04
1	0.9	-0.1	0.01
1	1.1	0.1	0.01
1	1.3	0.3	0.09
RMSE			0.1936

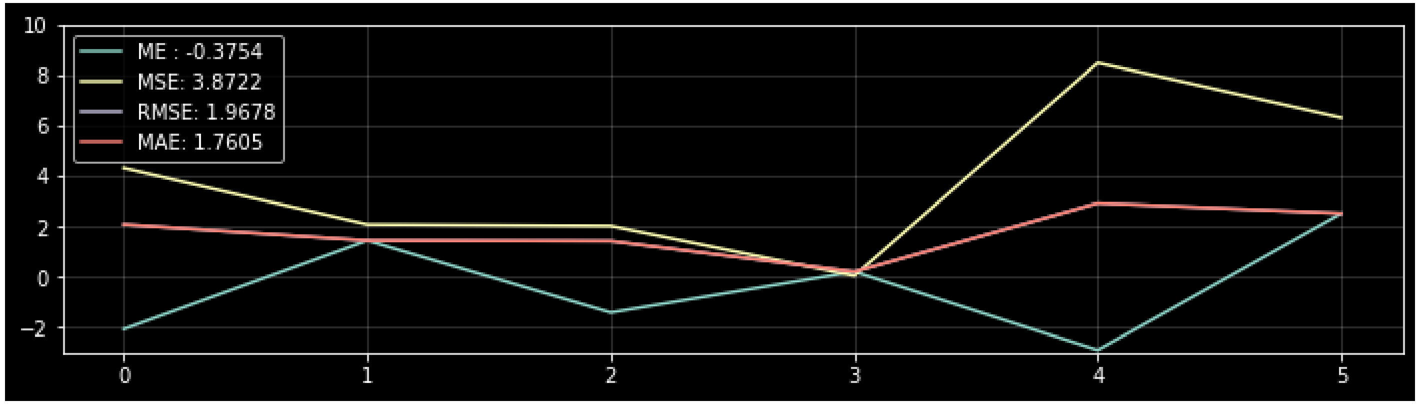


Mean Absolute Error (MAE)

- Error negatif dan error positif tidak saling menghabiskan
- Tidak memberlakukan penalti yang berbeda pada error yang kecil maupun error yang besar

$$MAE(y, \hat{y}) = \frac{1}{N}\sum_{i=1}^{N}\left|y^{(i)} - \hat{y}^{(i)}\right|$$

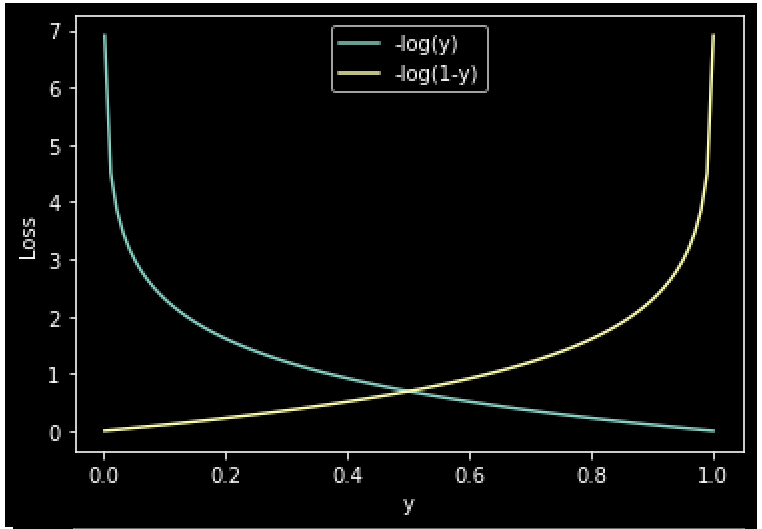
y	$\hat{y}$	E	AE
1	0.8	-0.2	0.2
1	0.9	-0.1	0.1
1	1.1	0.1	0.1
1	1.3	0.3	0.3
MAE			0.1



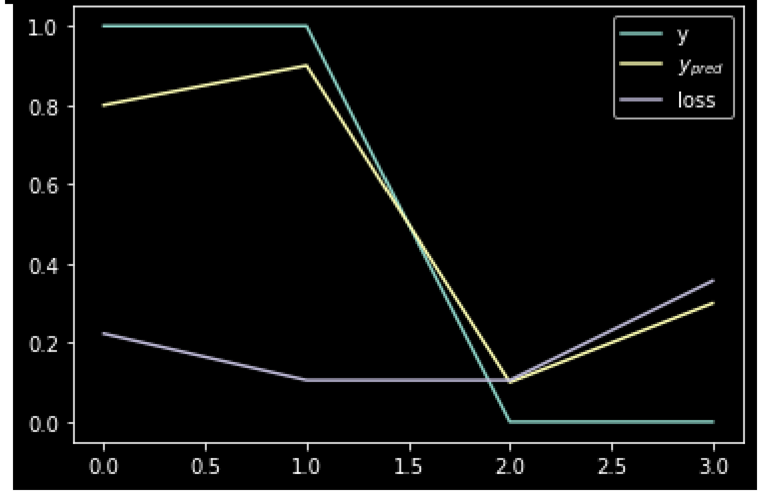
Negative Log Likelihood/ Binary Cross Entropy (BCE)

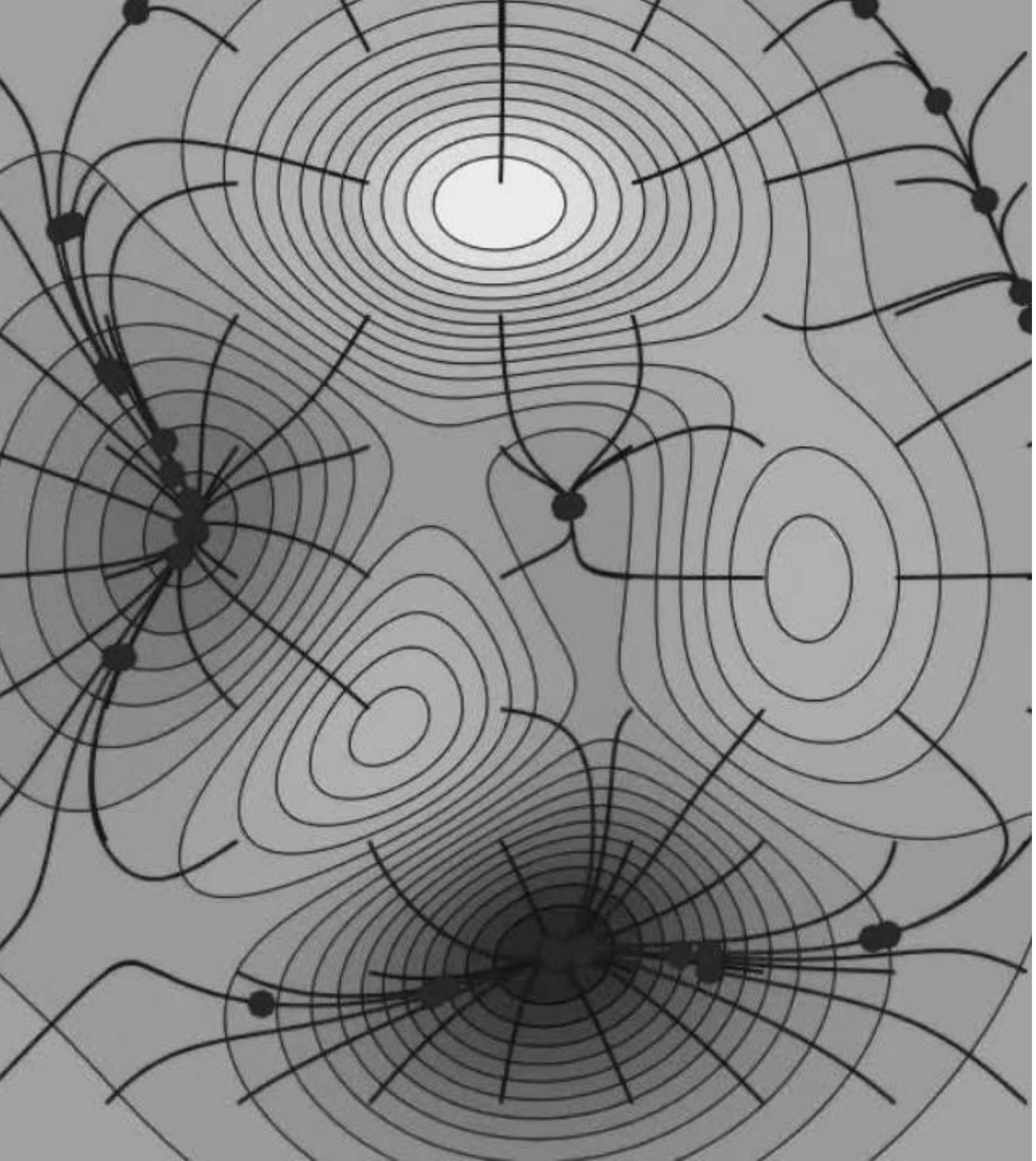
$$BCE(y,\hat{y}) = -\frac{1}{N}\sum_{i=1}^{N} y^{(i)}\log\left(\hat{y}^{(i)}\right) + \left(1-y^{(i)}\right)\log\left(1-\hat{y}^{(i)}\right)$$

- Negatif rata-rata dari log peluang-diperbaiki sebuah data



y	$\hat{y}$	$\hat{y}_{corrected}$	$log(\hat{y})$	$ylog(\hat{y})$	$(1-y)log(\hat{y}_{corrected})$
1	0.8	0.8	-0.223	-0.223	
1	0.9	0.9	-0.105	-0.105	
0	0.1	0.9	-2.303		-0.105
0	0.3	0.7	-1.204		-0.357
BCE					0.1976



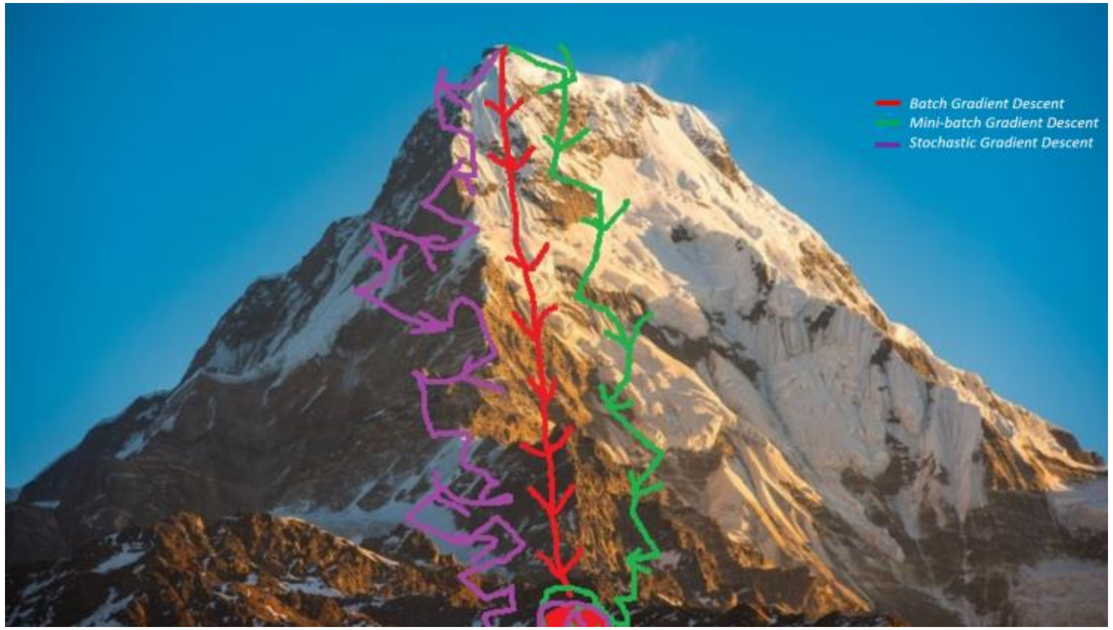


Gradient Descent

& Stochastic Gradient Descent

Gradient Descent

- Algoritma optimisasi conveks iteratif berorde satu
- Bertujuan untuk mencari **minimum lokal** dari suatu fungsi terdiferensiasi (*cost function*)



Sumber gambar: Imad Dabbura, Towards Data Science

Cost Function Optimisation

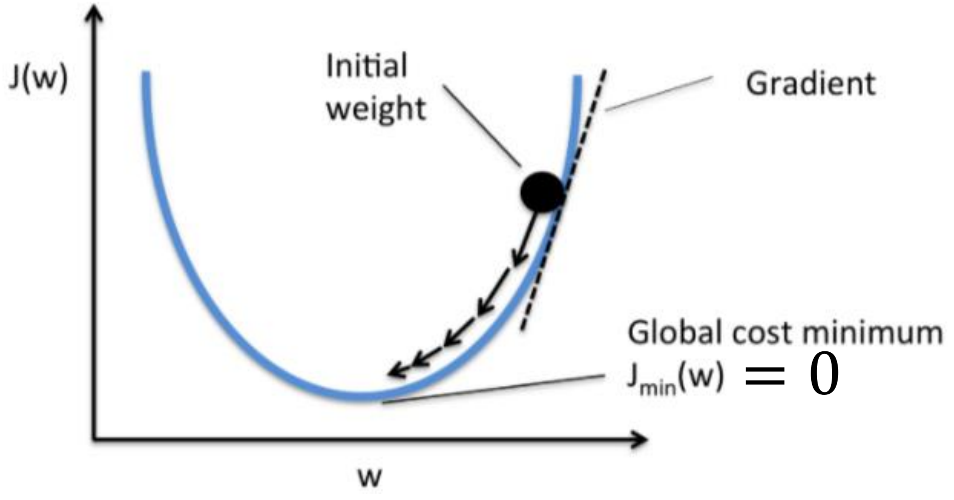
- Tujuan latihan: Meminimalkan cost

$$\min J(W) \sim \min_{W} Cost(\hat{y}, y)$$

$$W = \text{weights \& biases}$$

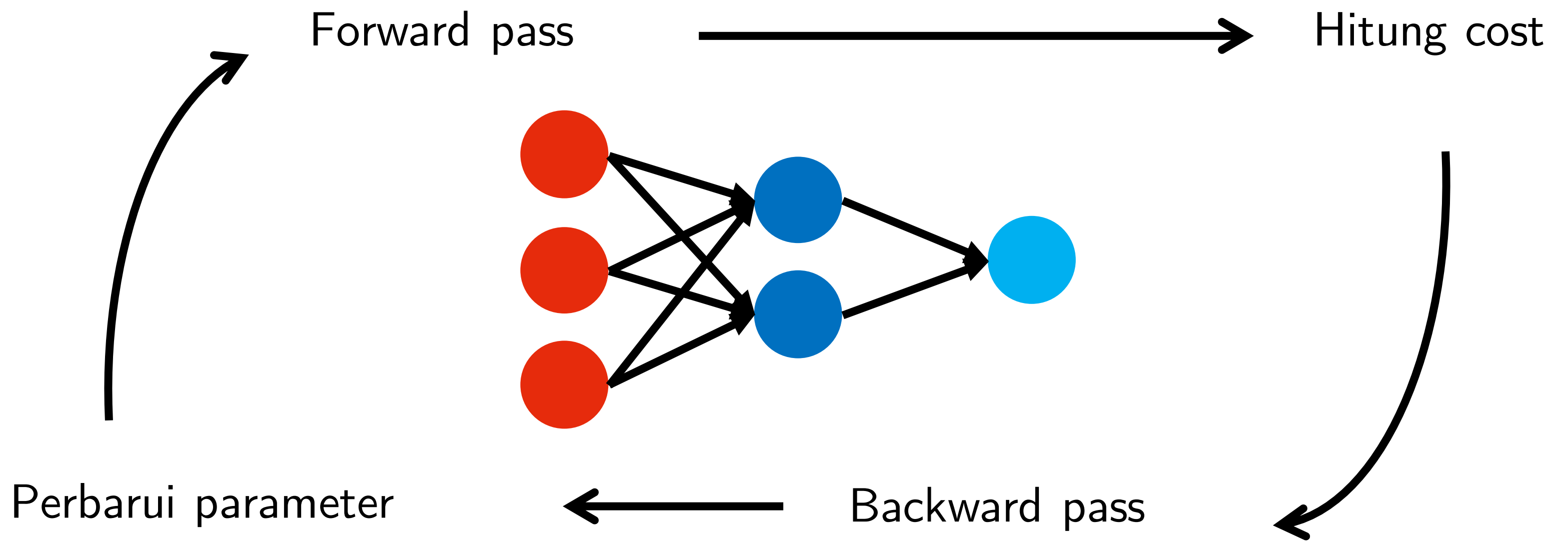
- Pilih W sedemikian sehingga $Cost(\hat{y}, y)$ minimum
  - Nilai $\hat{y}$ semakin mendekati y

$$J(w) = w^2$$



Sumber gambar: Analytics Vidhya

Learning algorithm: Gradient Descent

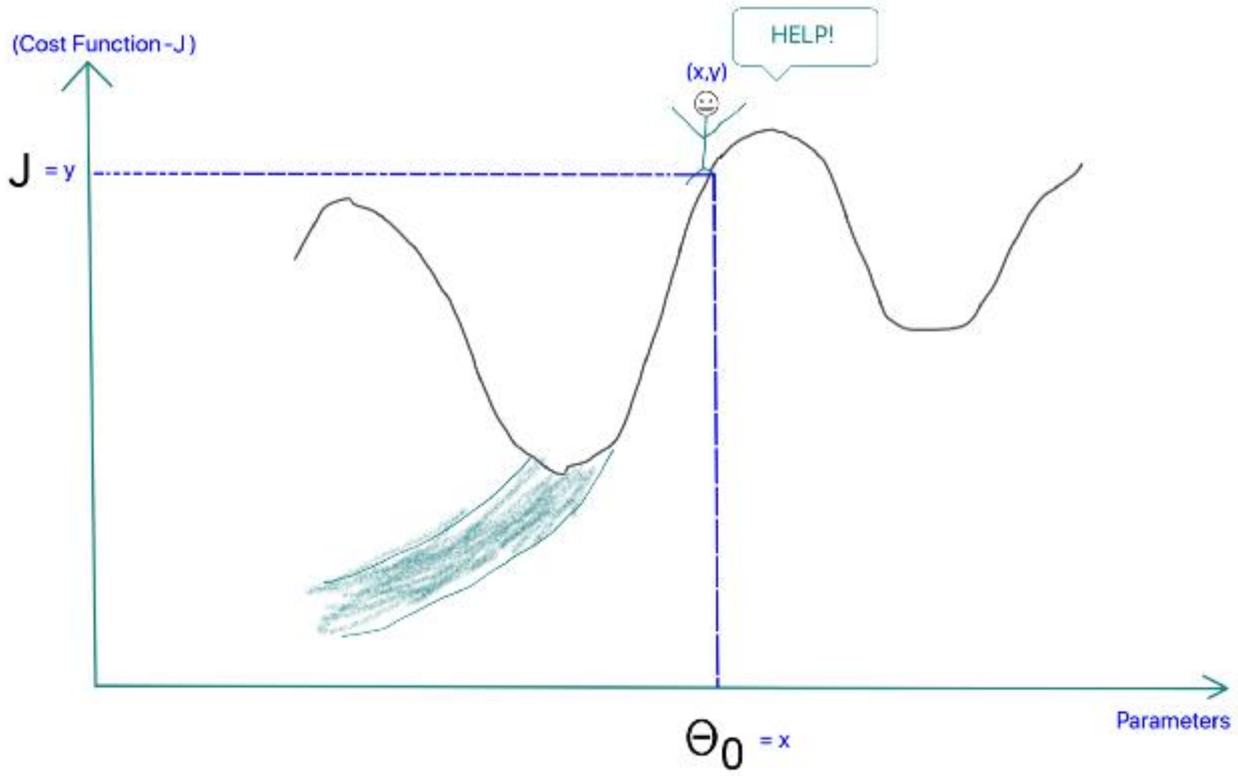


Perbaruan parameter: $W \coloneqq W - \alpha \nabla J(W)$

Learning rate $\boldsymbol{\alpha}$

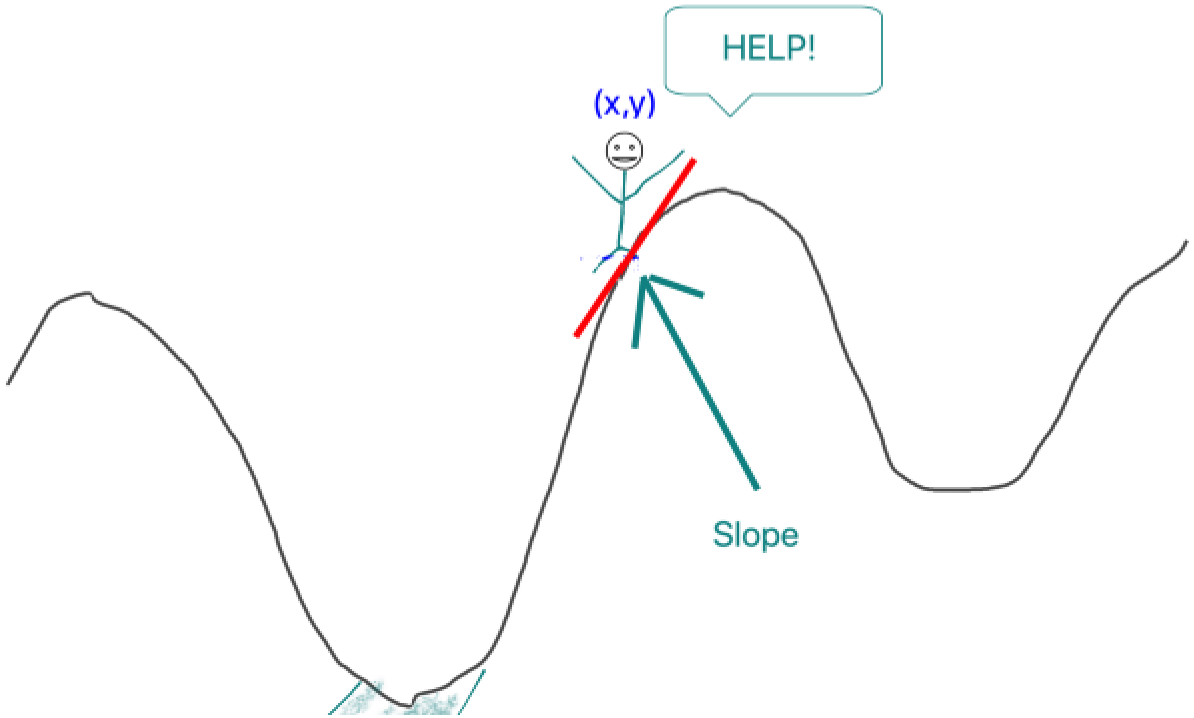
- $\boldsymbol{\alpha > 0}$
- Mengatur seberapa besar porsi dari gradient $\nabla J(W)$ yang diambil untuk mengubah parameter W (yang akan digunakan di iterasi latihan selanjutnya)
- Mengatur seberapa cepat model harus berlatih
- Mengatur seberapa sensitif respon parameter model terhadap data yang baru saja ia lihat

Cost function: $J(W)$



Sumber gambar: AI Graduate

Gradient cost function: $\nabla J(W)$

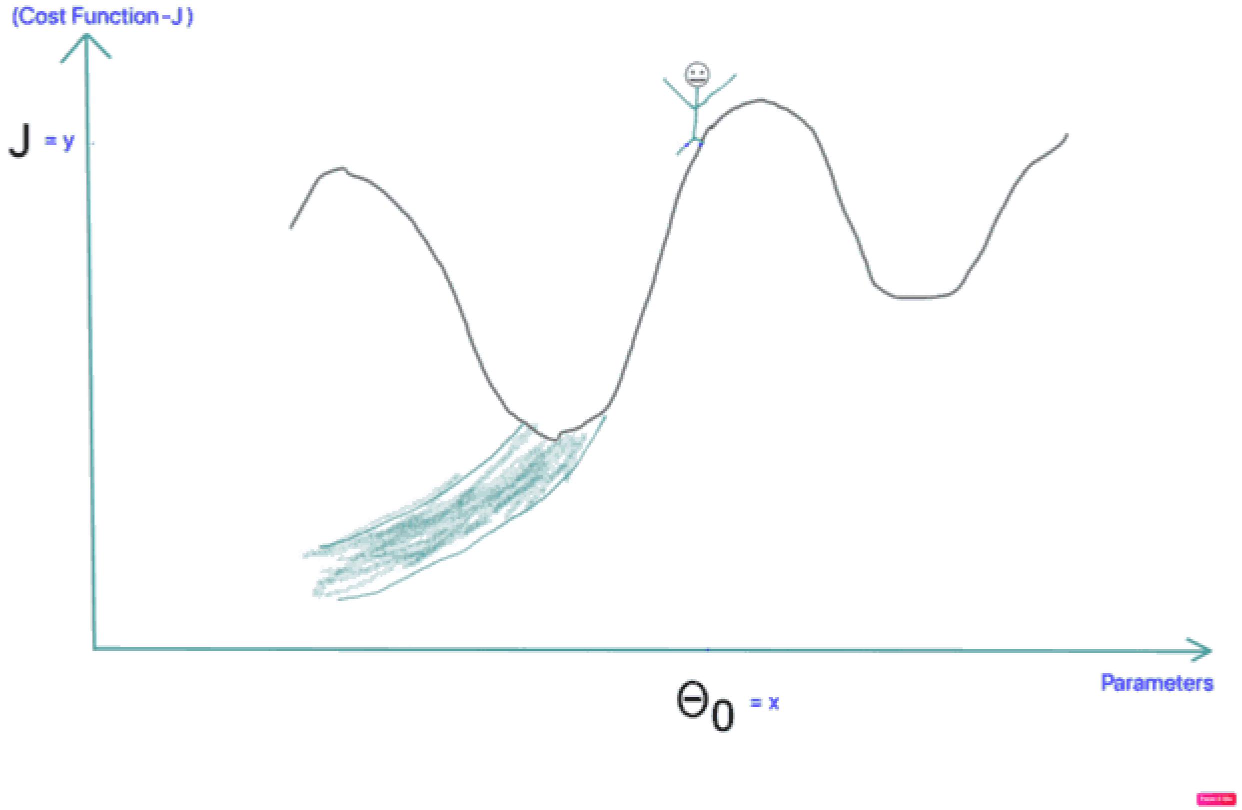


Sumber gambar: AI Graduate

α: learning rate

$0 \leq \alpha \leq 1$

$$\alpha \nabla J(W)$$



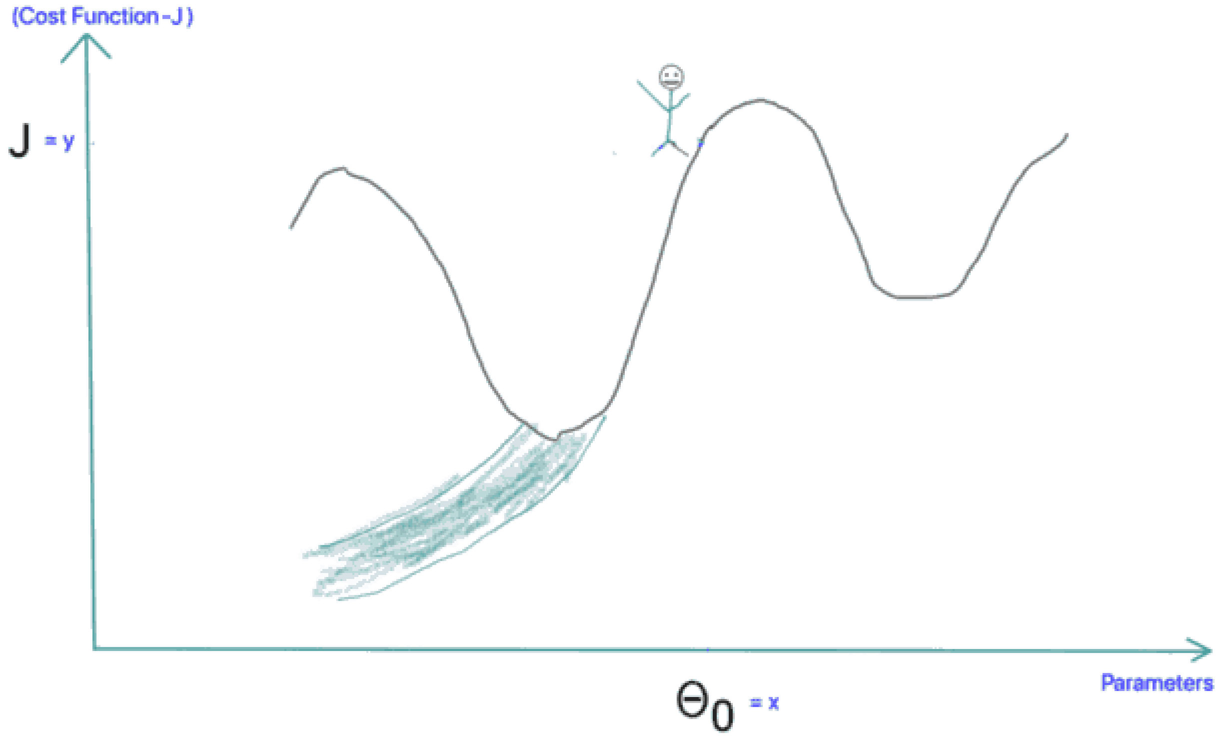
terlalu lambat

Sumber animasi: AI Graduate

α: learning rate

$0 \le \alpha \le 1$

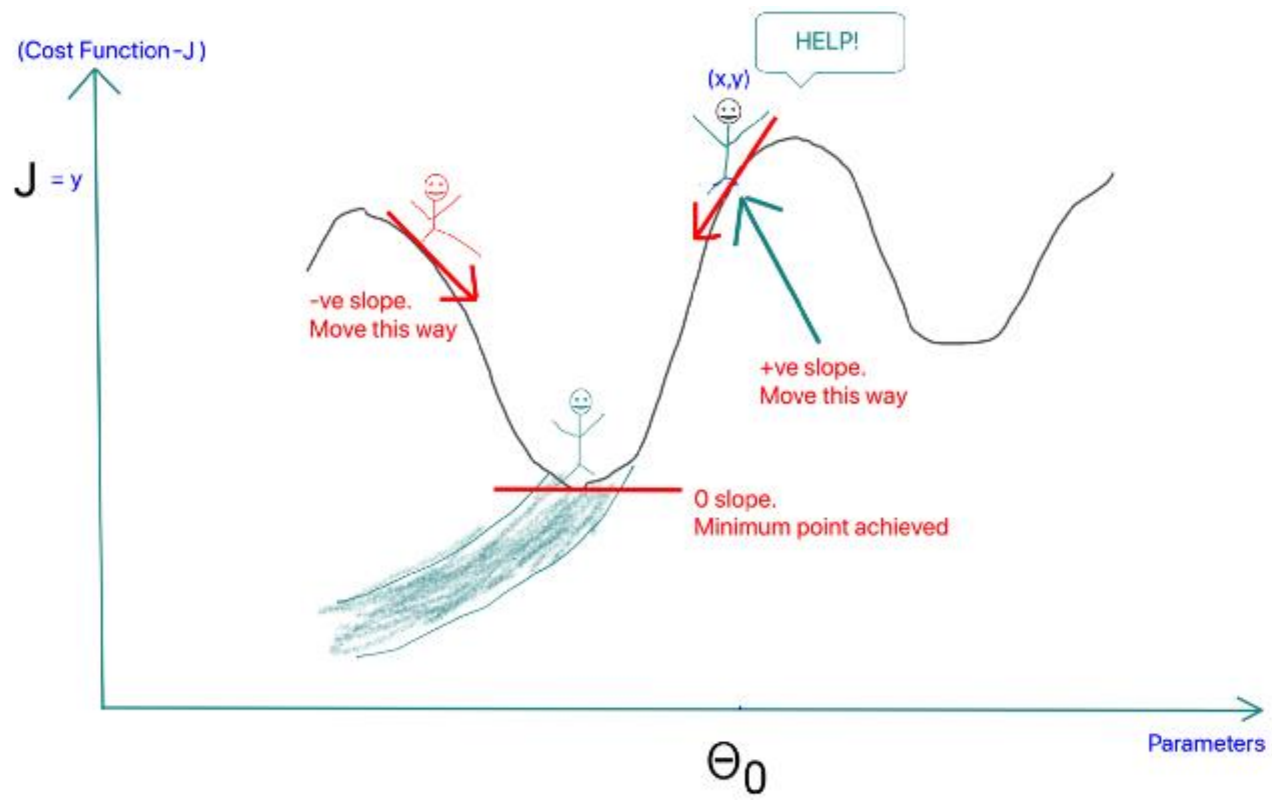
$$\alpha \nabla J(W)$$



terlalu cepat

Sumber animasi: AI Graduate

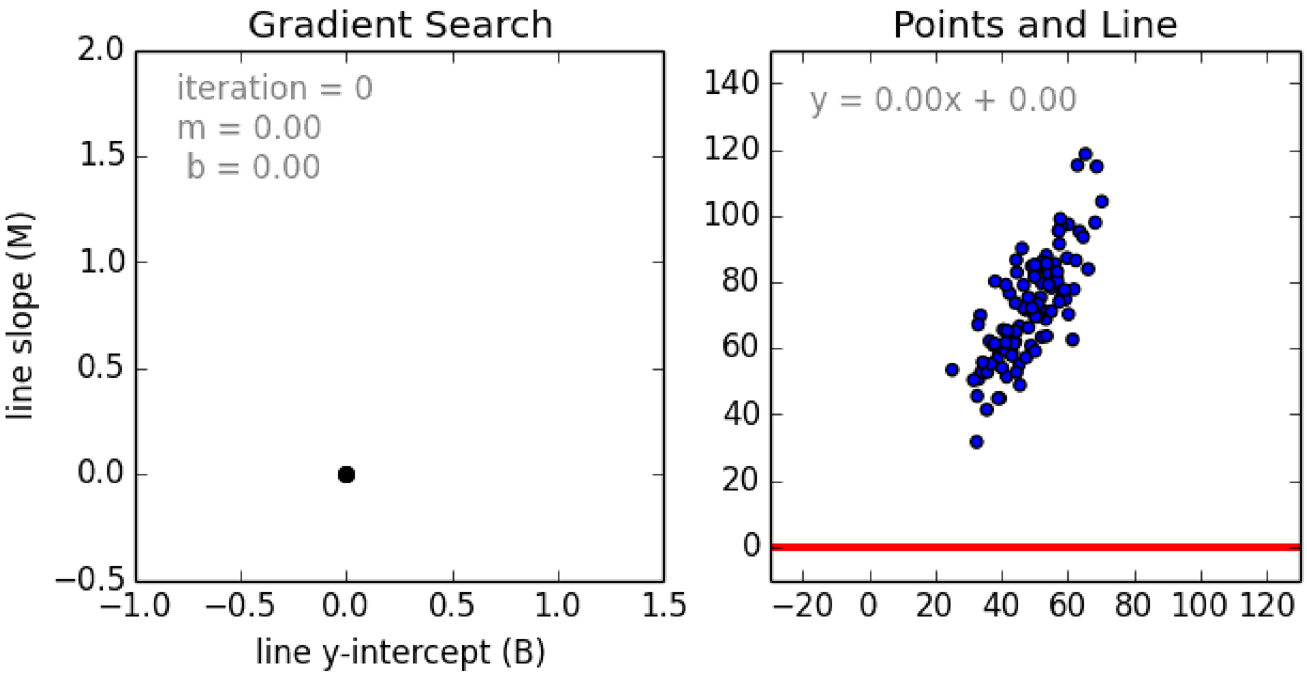
$$W := W - \alpha \nabla J(W)$$



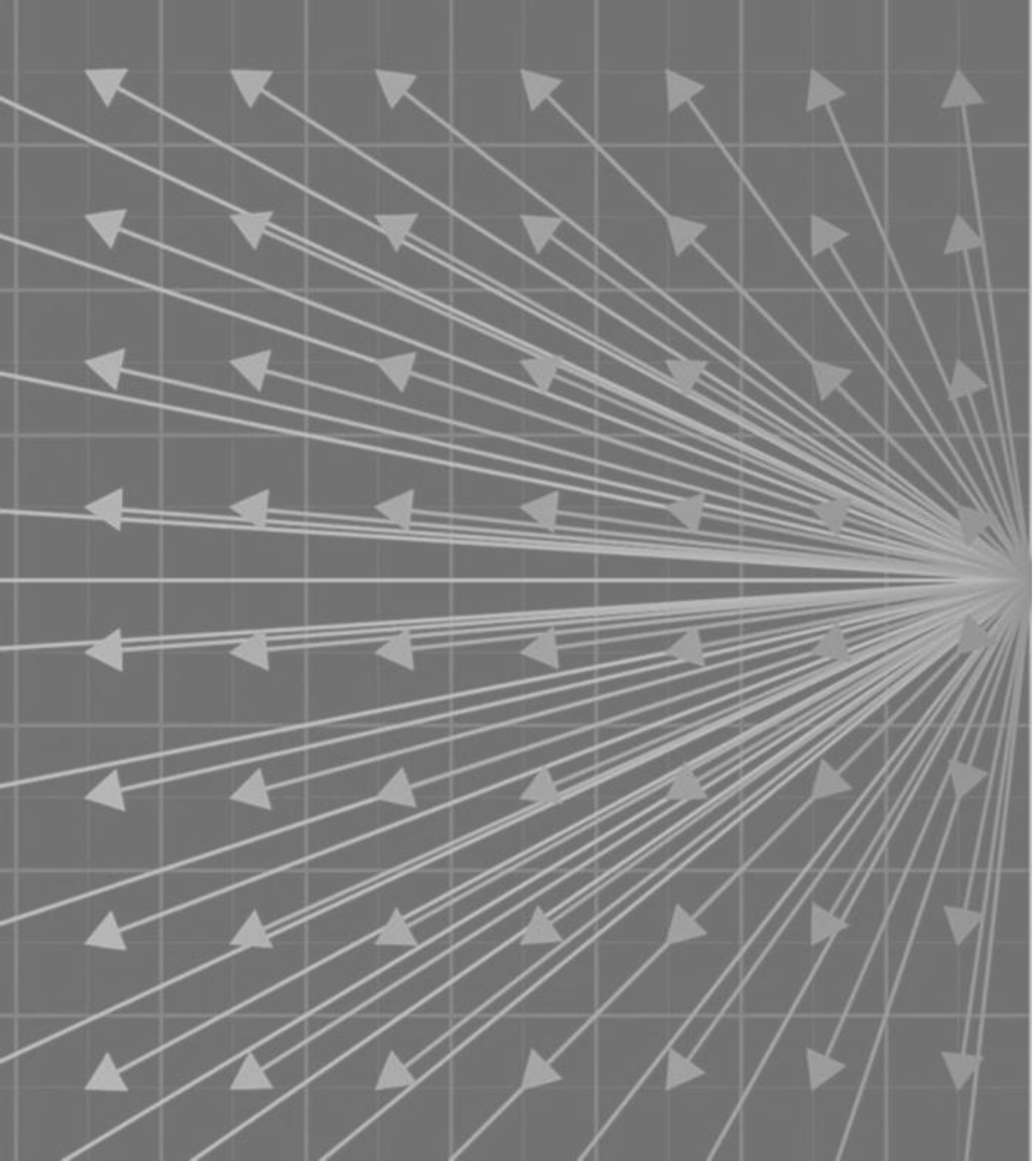
Sumber gambar: AI Graduate

Gradient Search

Model: $y = \mathrm{m}x + \mathrm{b}$



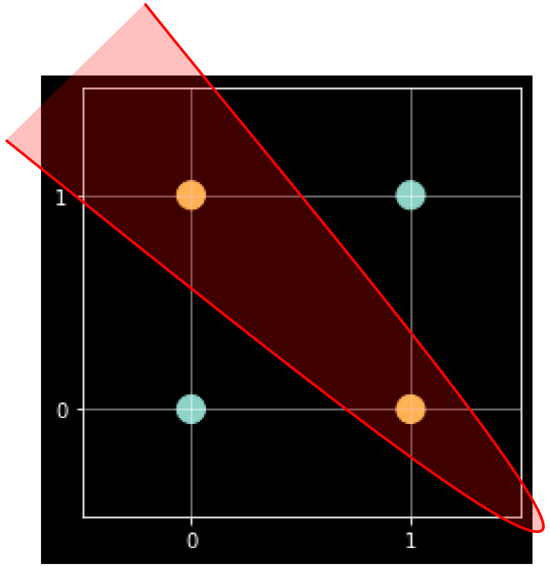
Sumber animasi: techburst.io



Backward Pass

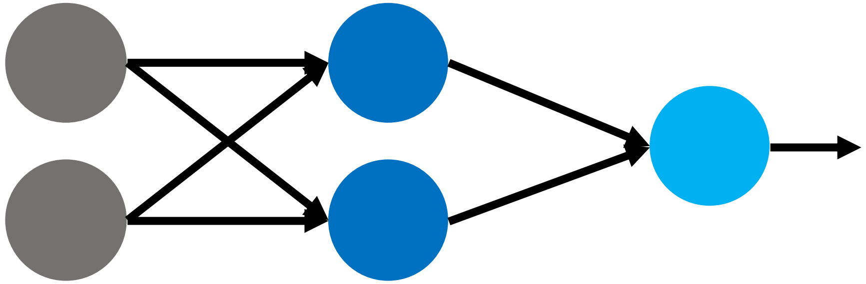
Contoh: Masalah XOR

x_1	x_2	y
0	0	0
1	0	1
0	1	1
1	1	0



Selesaikan dengan NN:

(notebook di akhir slide)



Forward pass

Input features

- $A_0 = X$

Layer 1

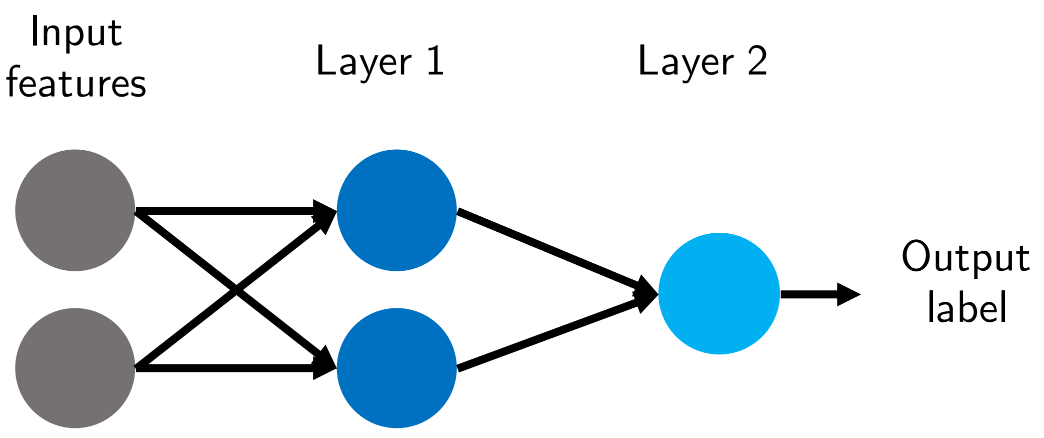
- $Z_1 = A_0 \cdot W_1 + b_1$
- $A_1 = \sigma_1(Z_1)$

Layer 2 - Output

- $Z_2 = A_1 \cdot W_2 + b_2$
- $A_2 = \sigma_2(Z_2) = \hat{y}$

Cost Function

- $J(W) = MSE(y, \hat{y})$



Forward pass: XOR problem

Input

- $A_0 = X$

Layer 1

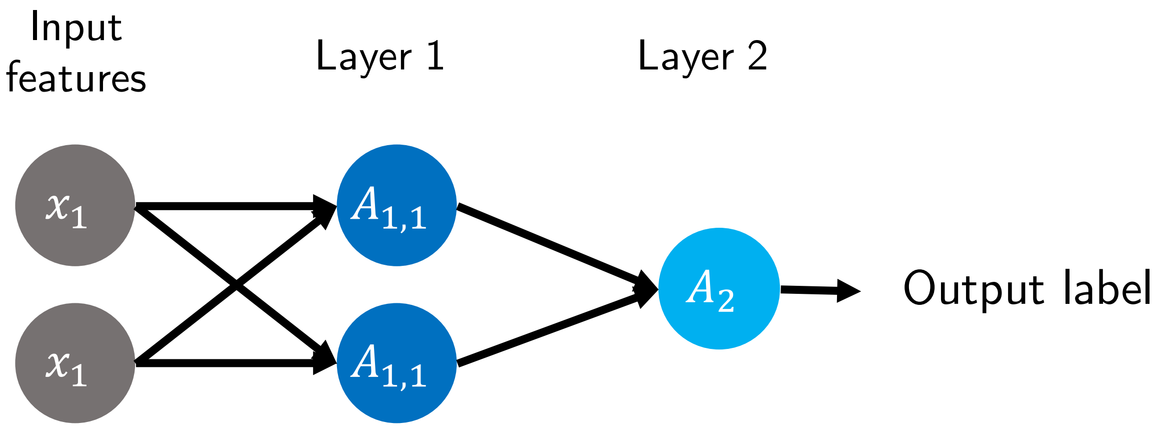
- $Z_1 = A_0 \cdot W_1 + b_1$
- $A_1 = \sigma(Z_1)$

Layer 2 - Output

- $Z_2 = A_1 \cdot W_2 + b_2$
- $A_2 = \sigma(Z_2)$

Cost Function

- $C = MSE(y, A_2)$



Turunan fungsi yang relevan:

$$\sigma(z) = sigmoid(z) = \frac{1}{1 + e^{-z}}$$

$$MSE(y, \hat{y}) = \frac{1}{N}\sum_{i=1}^{N}(\hat{y} - y)^2$$

$$\frac{d\sigma(z)}{dz} = \big(1 - \sigma(z)\big)\sigma(z)$$

$$\frac{dMSE(y, \hat{y})}{d\hat{y}} = \frac{2}{N}\sum_{i=1}^{N}(\hat{y} - y)$$

Computational Graph

Input

- $A_0 = X$

Layer 1

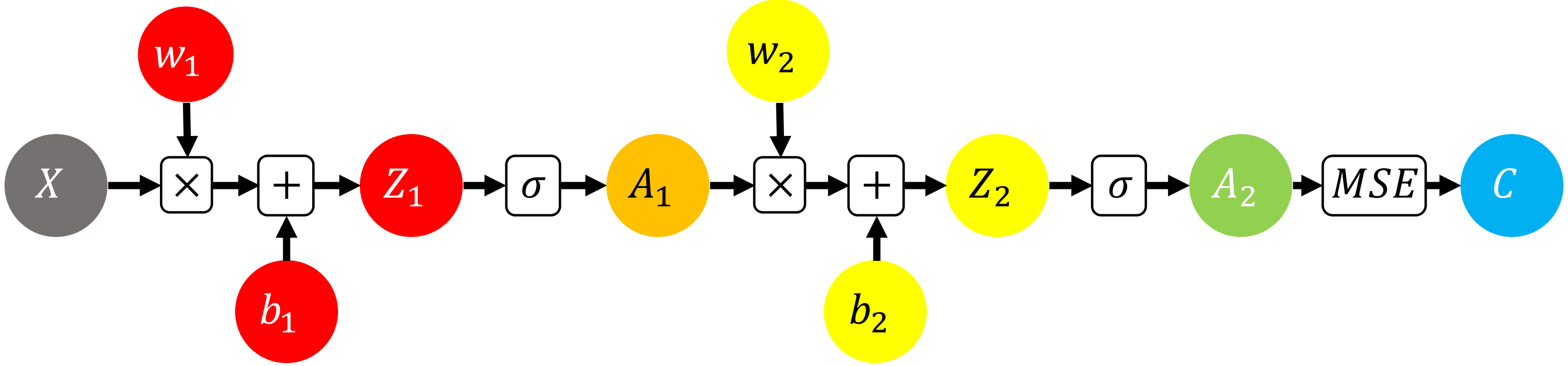
- $Z_1 = A_0 \cdot W_1 + b_1$
- $A_1 = \sigma(Z_1)$

Layer 2 - Output

- $Z_2 = A_1 \cdot W_2 + b_2$
- $A_2 = \sigma(Z_2)$

Cost Function

- $C = MSE(y, A_2)$



Computational Graph

Input

- $A_0 = X$

Layer 1

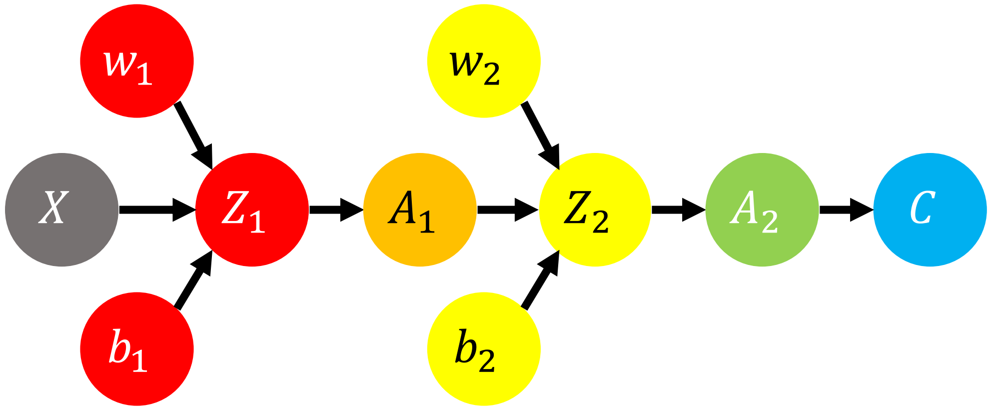
- $Z_1 = A_0 \cdot W_1 + b_1$
- $A_1 = \sigma(Z_1)$

Layer 2 - Output

- $Z_2 = A_1 \cdot W_2 + b_2$
- $A_2 = \sigma(Z_2)$

Cost Function

- $C = MSE(y, A_2)$



Forward Pass

Input

- $A_0 = X$

Layer 1

- $Z_1 = A_0 \cdot W_1 + b_1$
- $A_1 = \sigma(Z_1)$

Layer 2 - Output

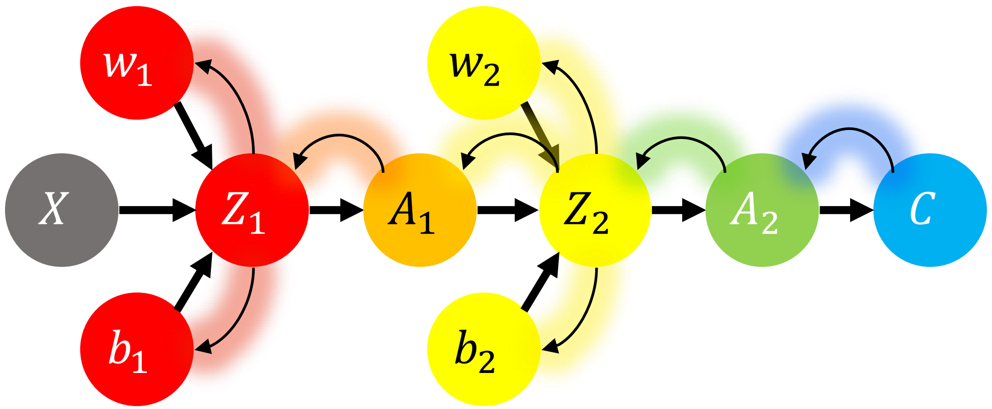
- $Z_2 = A_1 \cdot W_2 + b_2$
- $A_2 = \sigma(Z_2)$

Cost Function

- $C = MSE(y, A_2)$

Turunan/Turunan Parsial

- $\frac{\partial Z_1}{\partial W_1} = A_0 \quad \frac{\partial Z_1}{\partial b_1} = 1$
- $\frac{dA_1}{dZ_1} = \sigma'(Z_2)$
- $\frac{\partial Z_2}{\partial A_1} = W_2 \quad \frac{\partial Z_2}{\partial W_2} = A_1 \quad \frac{\partial Z_2}{\partial b_2} = 1$
- $\frac{dA_2}{dZ_2} = \sigma'(Z_2) = (1 - \sigma(Z_2))\sigma(Z_2)$
- $\frac{dC(W)}{dA_2} = \frac{2}{N}\sum(A_2 - y)$



Backward Pass

Layer 2

- $\frac{dC}{dA_2} = \frac{2}{N}\sum(A_2 - y)$
- $\frac{dC}{dZ_2} = \frac{dC}{dA_2}\frac{dA_2}{dZ_2}$

Layer 1

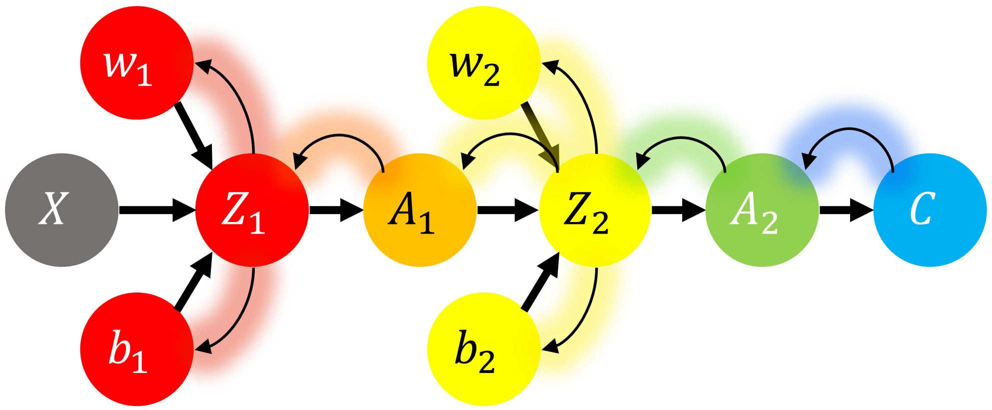
- $\frac{dC}{dA_1} = \frac{dC}{dZ_2}\frac{\partial Z_2}{\partial A_1} = \frac{dC}{dA_2}\frac{dA_2}{dZ_2}\frac{\partial Z_2}{\partial A_1}$
- $\frac{dC}{dZ_1} = \frac{dC}{dA_1}\frac{dA_1}{dZ_1} = \frac{dC}{dA_2}\frac{dA_2}{dZ_2}\frac{\partial Z_2}{\partial A_1}\frac{dA_1}{dZ_1}$

Perubahan Parameter

- $\frac{\partial C}{\partial W_2} = \frac{dC}{dZ_2}\frac{\partial Z_2}{\partial W_2}$ $\quad \frac{\partial C}{\partial b_1} = \frac{dC}{dZ_2}\frac{\partial Z_2}{\partial b_2}$
- $\frac{\partial C}{\partial W_1} = \frac{dC}{dZ_1}\frac{\partial Z_1}{\partial W_1}$ $\quad \frac{\partial C}{\partial b_1} = \frac{dC}{dZ_1}\frac{\partial Z_1}{\partial b_1}$

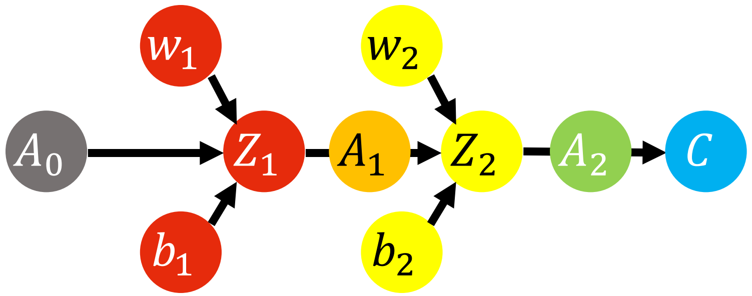
Turunan/Turunan Parsial

- $\frac{\partial Z_1}{\partial W_1} = A_0 \quad \frac{\partial Z_1}{\partial b_1} = 1$
- $\frac{dA_1}{dZ_1} = \sigma'(Z_2)$
- $\frac{\partial Z_2}{\partial A_1} = W_2 \quad \frac{\partial Z_2}{\partial W_2} = A_1 \quad \frac{\partial Z_2}{\partial b_2} = 1$
- $\frac{dA_2}{dZ_2} = \sigma'(Z_2) = \big(1 - \sigma(Z_2)\big)\sigma(Z_2)$
- $\frac{dC(W)}{dA_2} = \frac{2}{N}\sum(A_2 - y)$



Update Parameter

- $W_2 := W_1 - \alpha \frac{dC}{dW_2} = W_2 - \alpha \left(\frac{dC}{dA_2} \frac{dA_2}{dZ_2} \frac{\partial Z_2}{\partial W_2} \right)$
- $b_2 := b_2 - \alpha \frac{dC}{db_2} = b_2 - \alpha \left(\frac{dC}{dA_2} \frac{dA_2}{dZ_2} \frac{\partial Z_2}{\partial b_2} \right)$
- $W_1 := W_1 - \alpha \frac{dC}{dW_1} = W_1 - \alpha \left(\frac{dC}{dA_2} \frac{dA_2}{dZ_2} \frac{\partial Z_2}{\partial A_1} \frac{dA_1}{dZ_1} \frac{\partial Z_1}{\partial W_1} \right)$
- $b_1 := b_1 - \alpha \frac{dC}{db_1} = b_1 - \alpha \left(\frac{dC}{dA_2} \frac{dA_2}{dZ_2} \frac{\partial Z_2}{\partial A_1} \frac{dA_1}{dZ_1} \frac{\partial Z_1}{\partial b_1} \right)$



Update Parameter

- $W_2 \coloneqq W_1 - \alpha \frac{dC}{dW_2} = W_2 - \alpha \left(\frac{2}{N} \sum (A_2 - y) \frac{dA_2}{dZ_2} \frac{\partial Z_2}{\partial W_2} \right)$
- $b_2 \coloneqq b_2 - \alpha \frac{dC}{db_2} = b_2 - \alpha \left(\frac{2}{N} \sum (A_2 - y) \frac{dA_2}{dZ_2} \frac{\partial Z_2}{\partial b_2} \right)$
- $W_1 \coloneqq W_1 - \alpha \frac{dC}{dW_1} = W_1 - \alpha \left(\frac{2}{N} \sum (A_2 - y) \frac{dA_2}{dZ_2} \frac{\partial Z_2}{\partial A_1} \frac{dA_1}{dZ_1} \frac{\partial Z_1}{\partial W_1} \right)$
- $b_1 \coloneqq b_1 - \alpha \frac{dC}{db_1} = b_1 - \alpha \left(\frac{2}{N} \sum (A_2 - y) \frac{dA_2}{dZ_2} \frac{\partial Z_2}{\partial A_1} \frac{dA_1}{dZ_1} \frac{\partial Z_1}{\partial b_1} \right)$

Update Parameter

- $W_2 \coloneqq W_1 - \alpha \frac{dC}{dW_2} = W_2 - \alpha \left(\frac{2}{N} \sum (A_2 - y) \left(1 - \sigma(Z_2)\right) \sigma(Z_2) \frac{\partial Z_2}{\partial W_2} \right)$
- $b_2 \coloneqq b_2 - \alpha \frac{dC}{db_2} = b_2 - \alpha \left(\frac{2}{N} \sum (A_2 - y) \left(1 - \sigma(Z_2)\right) \sigma(Z_2) \frac{\partial Z_2}{\partial b_2} \right)$
- $W_1 \coloneqq W_1 - \alpha \frac{dC}{dW_1} = W_1 - \alpha \left(\frac{2}{N} \sum (A_2 - y) \left(1 - \sigma(Z_2)\right) \sigma(Z_2) \frac{\partial Z_2}{\partial A_1} \frac{dA_1}{dZ_1} \frac{\partial Z_1}{\partial W_1} \right)$
- $b_1 \coloneqq b_1 - \alpha \frac{dC}{db_1} = b_1 - \alpha \left(\frac{2}{N} \sum (A_2 - y) \left(1 - \sigma(Z_2)\right) \sigma(Z_2) \frac{\partial Z_2}{\partial A_1} \frac{dA_1}{dZ_1} \frac{\partial Z_1}{\partial b_1} \right)$

Update Parameter

- $W_2 := W_1 - \alpha \frac{dC}{dW_2} = W_2 - \alpha \left(\frac{2}{N} \sum (A_2 - y) \left(1 - \sigma(Z_2)\right) \sigma(Z_2) A_1 \right)$
- $b_2 := b_2 - \alpha \frac{dC}{db_2} = b_2 - \alpha \left(\frac{2}{N} \sum (A_2 - y) \left(1 - \sigma(Z_2)\right) \sigma(Z_2) 1 \right)$
- $W_1 := W_1 - \alpha \frac{dC}{dW_1} = W_1 - \alpha \left(\frac{2}{N} \sum (A_2 - y) \left(1 - \sigma(Z_2)\right) \sigma(Z_2) W_2 \frac{dA_1}{dZ_1} \frac{\partial Z_1}{\partial W_1} \right)$
- $b_1 := b_1 - \alpha \frac{dC}{db_1} = b_1 - \alpha \left(\frac{2}{N} \sum (A_2 - y) \left(1 - \sigma(Z_2)\right) \sigma(Z_2) W_2 \frac{dA_1}{dZ_1} \frac{\partial Z_1}{\partial b_1} \right)$

Update Parameter

- $W_2 \coloneqq W_1 - \alpha \frac{dC}{dW_2} = W_2 - \alpha \left(\frac{2}{N} \sum (A_2 - y) \left(1 - \sigma(Z_2)\right) \sigma(Z_2) A_1 \right)$

- $b_2 \coloneqq b_2 - \alpha \frac{dC}{db_2} = b_2 - \alpha \left(\frac{2}{N} \sum (A_2 - y) \left(1 - \sigma(Z_2)\right) \sigma(Z_2) 1 \right)$

- $W_1 \coloneqq W_1 - \alpha \frac{dC}{dW_1} = W_1 - \alpha \left(\frac{2}{N} \sum (A_2 - y) \left(1 - \sigma(Z_2)\right) \sigma(Z_2) W_2 \left(1 - \sigma(Z_1)\right) \sigma(Z_1) \frac{\partial Z_1}{\partial W_1} \right)$

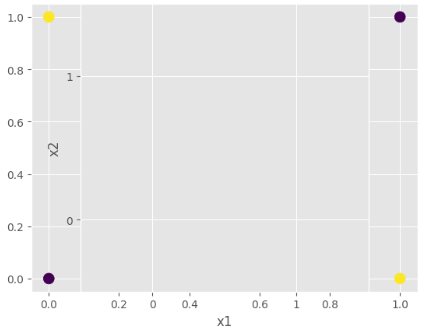
- $b_1 \coloneqq b_1 - \alpha \frac{dC}{db_1} = b_1 - \alpha \left(\frac{2}{N} \sum (A_2 - y) \left(1 - \sigma(Z_2)\right) \sigma(Z_2) W_2 \left(1 - \sigma(Z_1)\right) \sigma(Z_1) \frac{\partial Z_1}{\partial b_1} \right)$

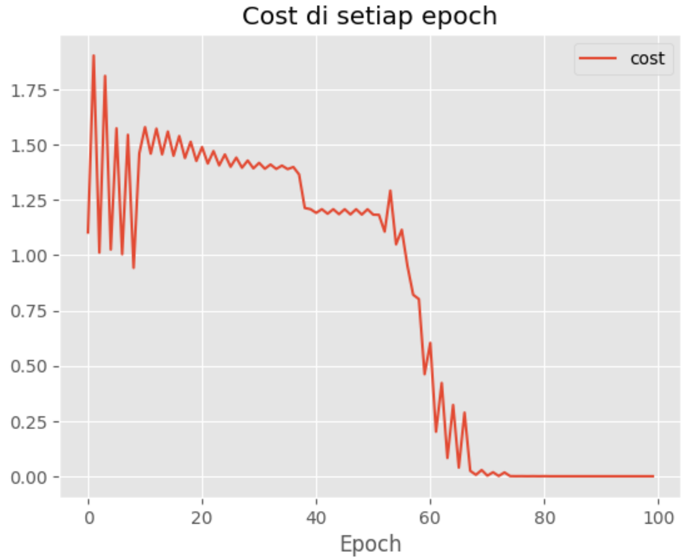
Update Parameter

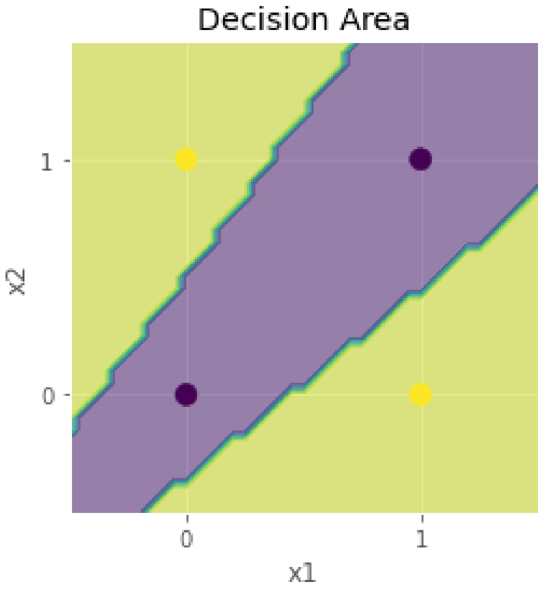
- $W_2 \coloneqq W_2 - \alpha \frac{dC}{dW_2} = W_2 - \alpha \left(\frac{2}{N} \sum (A_2 - y) \left(1 - \sigma(Z_2)\right) \sigma(Z_2) A_1 \right)$
- $b_2 \coloneqq b_2 - \alpha \frac{dC}{db_2} = b_2 - \alpha \left(\frac{2}{N} \sum (A_2 - y) \left(1 - \sigma(Z_2)\right) \sigma(Z_2) 1 \right)$
- $W_1 \coloneqq W_1 - \alpha \frac{dC}{dW_1} = W_1 - \alpha \left(\frac{2}{N} \sum (A_2 - y) \left(1 - \sigma(Z_2)\right) \sigma(Z_2) W_2 \left(1 - \sigma(Z_1)\right) \sigma(Z_1) A_0 \right)$
- $b_1 \coloneqq b_1 - \alpha \frac{dC}{db_1} = b_1 - \alpha \left(\frac{2}{N} \sum (A_2 - y) \left(1 - \sigma(Z_2)\right) \sigma(Z_2) W_2 \left(1 - \sigma(Z_1)\right) \sigma(Z_1) 1 \right)$

Contoh aplikasi: Masalah XOR

```
LogicGate = {"AND": [[0], [0], [0], [1]],
             "NAND": [[1], [1], [1], [0]],
             "OR": [[0], [1], [1], [1]],
             "XOR": [[0], [1], [1], [0]],
             "NOR": [[1], [0], [0], [0]],
             "XNOR": [[1], [0], [0], [1]]}
```







Google Colaboratory:
https://drive.google.com/file/d/1BWyxq_Hm7K1lb85qavxR2SPs623cXo96/view?usp=sharing

Open with Google Colaborat...

Futher learning...

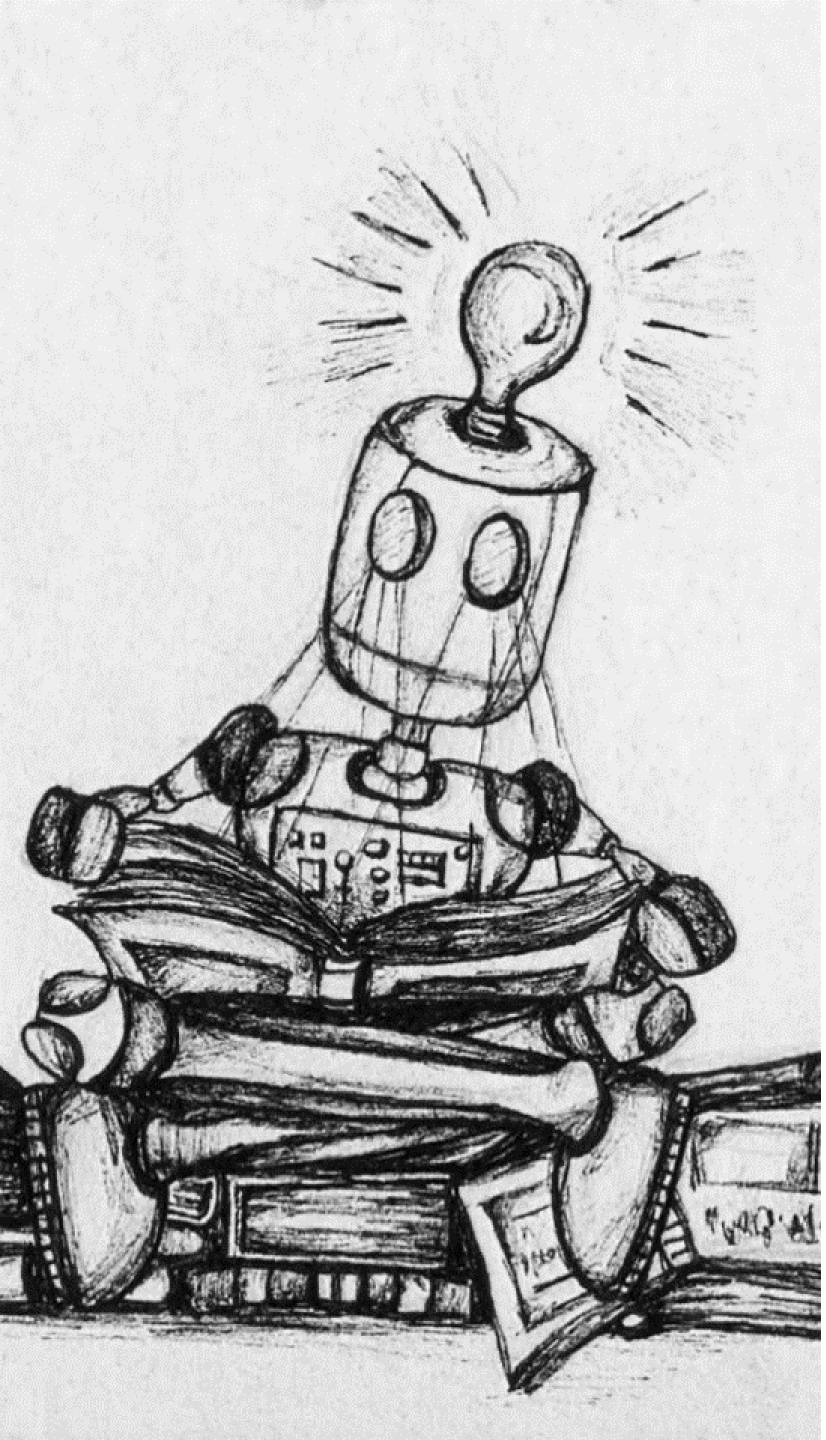
- **Deep Learning Book (Goodfellow et. al., 2016)**

  https://www.deeplearningbook.org/

- **Dive into Deep Learning:**

  Appendix: Mathematics for Deep Learning

  https://www.d2l.ai/chapter_appendix-mathematics-for-deep-learning/index.html



Thank you!

Find me on

LinkedIn : linkedin.com/in/fiddien

Website : fiddien.com