



Session 34

Neural Network



Table of Content

Apa yang Akan Kita Pelajari Hari Ini?

1. History
2. Perceptron
3. Multilayer Perceptron





History





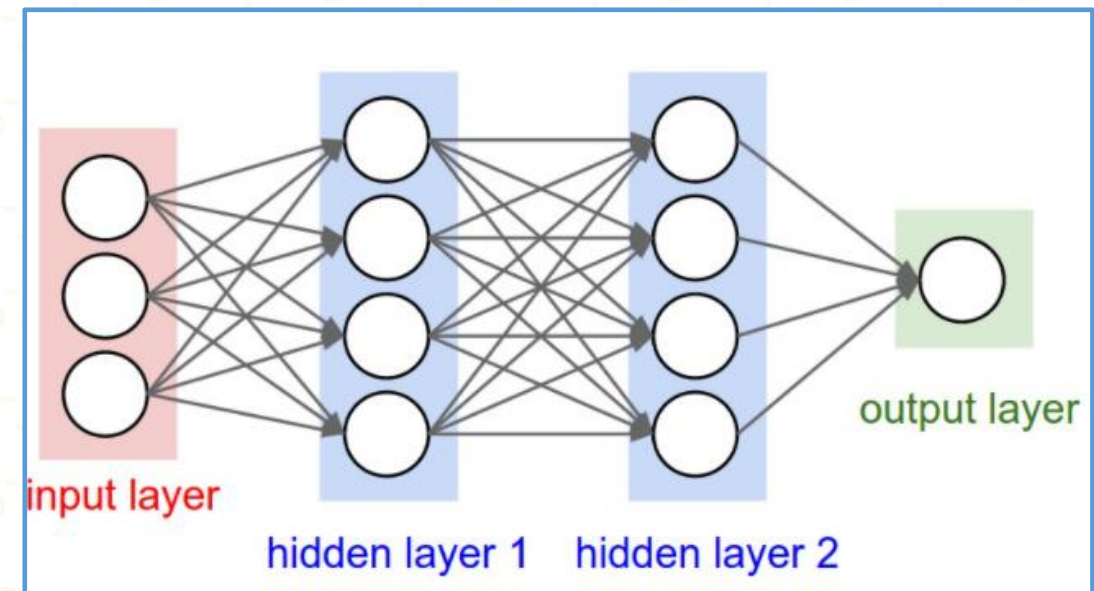
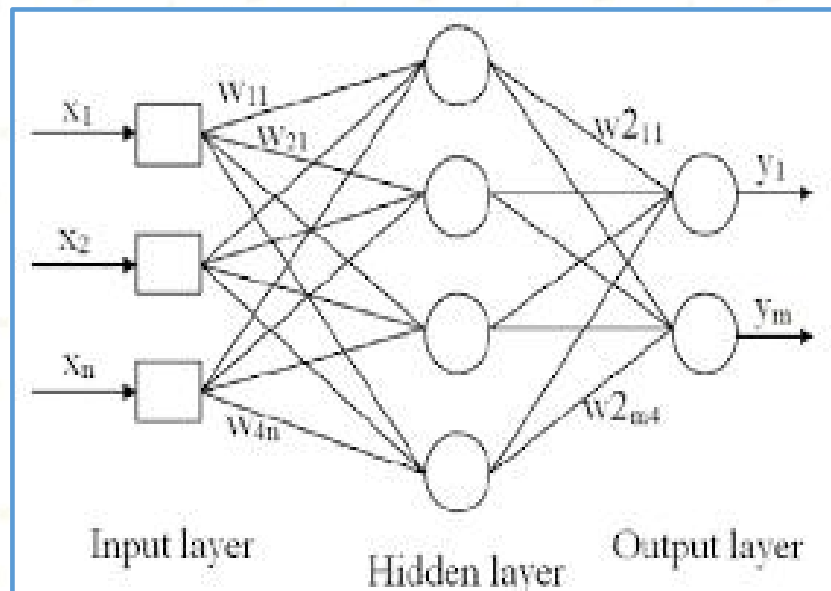
Brief History

- **History of ANN**
- 1943: McCulloch and Pitts: First mathematic model of neuron
- 1958: Rosenblatt: Perceptron - Single layer NN
- 1986: Rumelhart: Back Propagation algorithm
- 1995: Y. LeCun, Y. Bengio, et al.: Convolutional neural network
- 2006: G. E. Hinton, et al.: Deep belief nets.



Artificial Neural Network

- Artificial Neural Network (Jaringan saraf tiruan) terdiri dari kumpulan unit pemrosesan sederhana yang berkomunikasi dengan mengirimkan sinyal satu sama lain melalui sejumlah besar koneksi berbobot.
- Model yang terinspirasi oleh bagaimana neuron dalam otak manusia bekerja

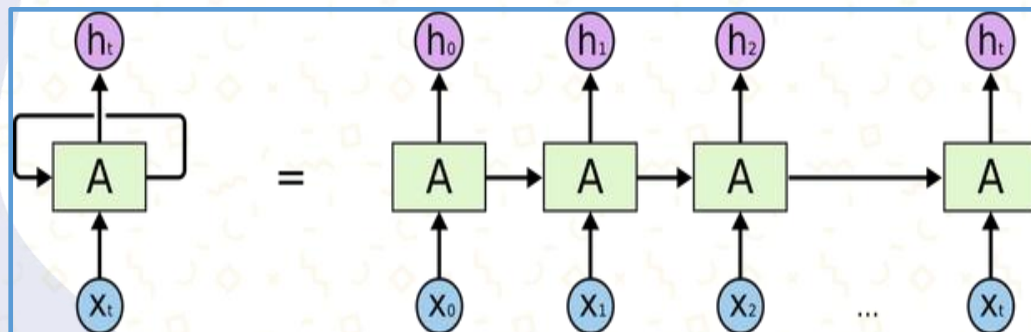
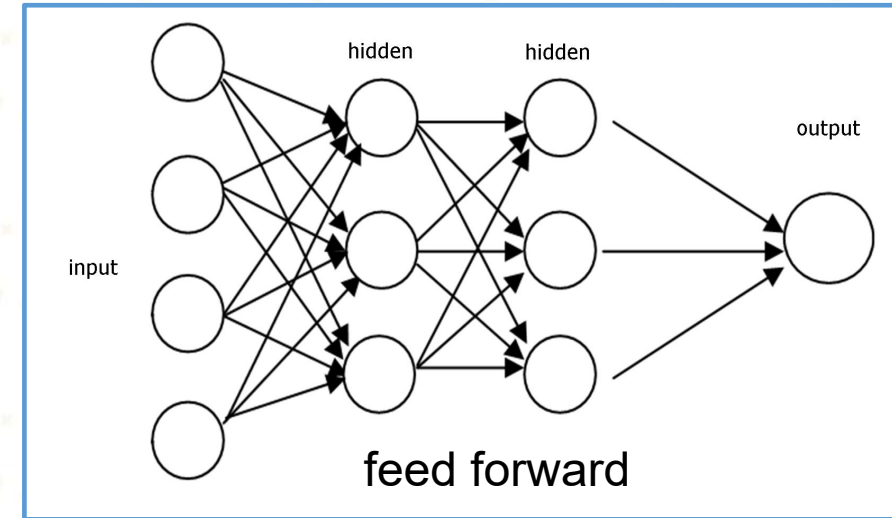




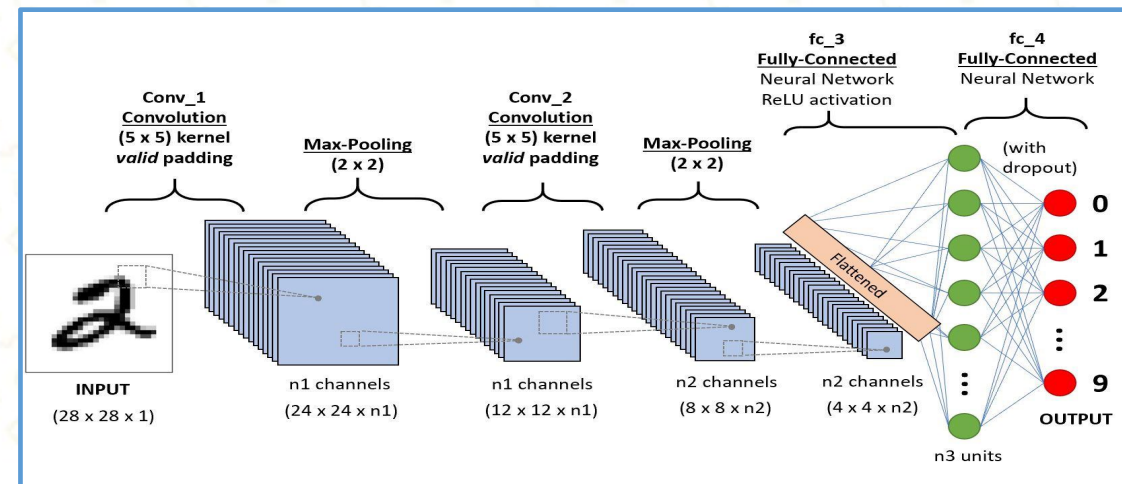
Types of ANN

Types of ANN

- Feedforward neural network
- Recurrent neural network (RNN)
- Convolutional neural network (CNN)
- etc,



RNN

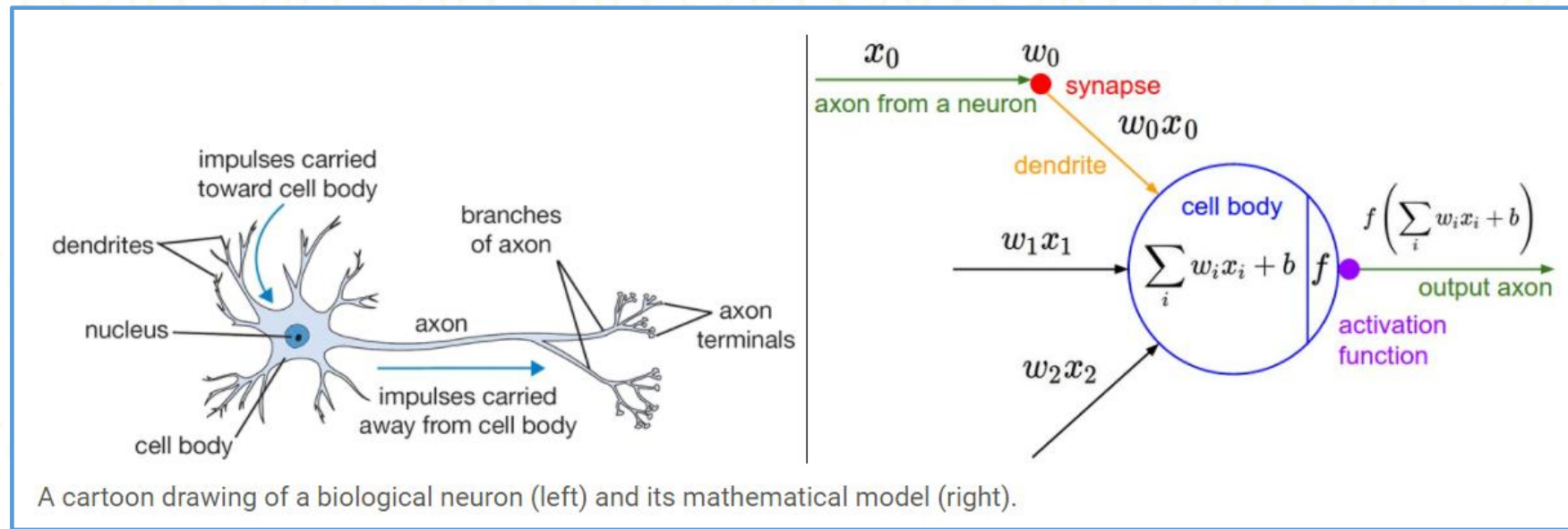


CNN



Biological motivation

- Neural Networks awalnya terinspirasi oleh tujuan pemodelan sistem saraf biologis.
- Unit komputasi dasar otak adalah neuron.
- Sekitar 86 miliar neuron dapat ditemukan di sistem saraf manusia





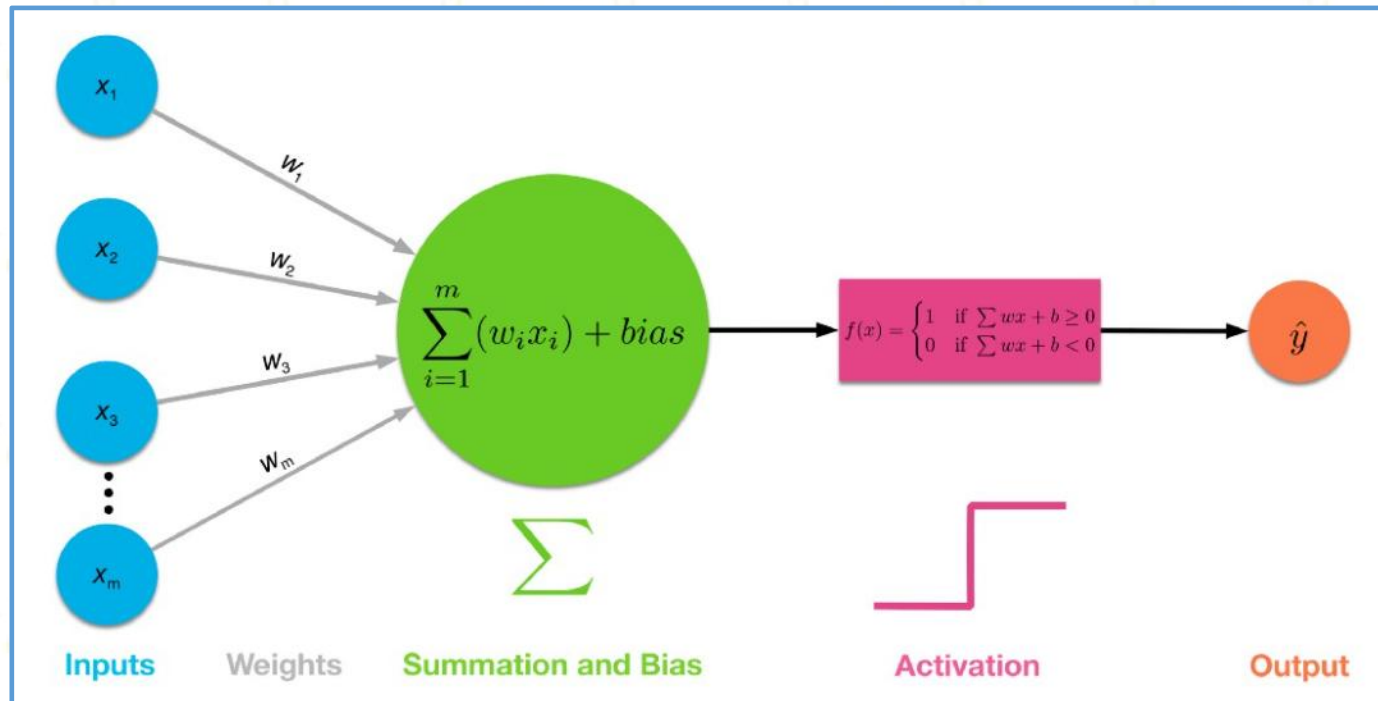
Perceptron





Perceptron

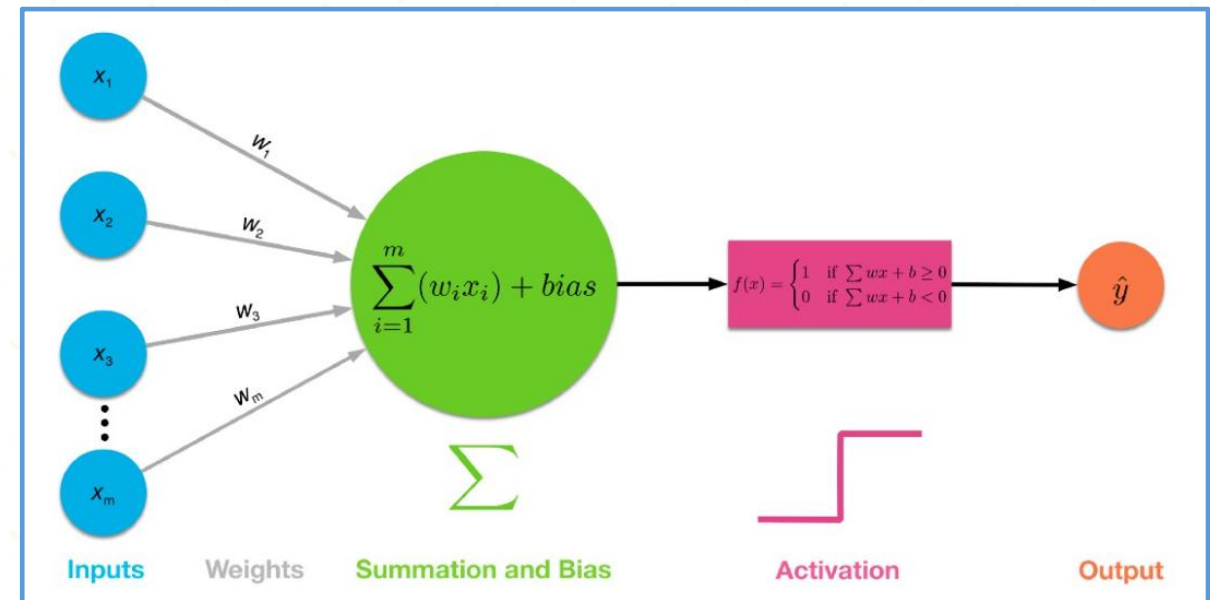
- Perceptron adalah pengklasifikasi linier (linear classifier).
- Frank Rosenblatt, seorang psikolog Amerika, mengusulkan model perceptron klasik pada tahun 1958





Perceptron

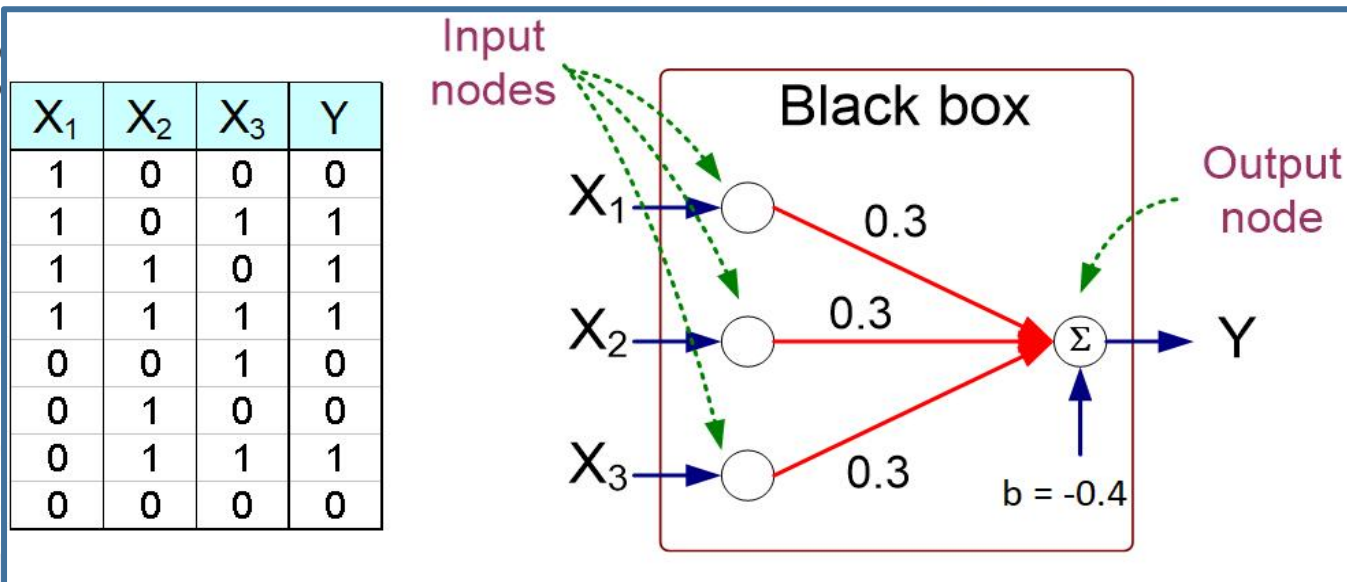
- Digunakan untuk mengklasifikasikan data input yang diberikan.
- The perceptron terdiri dari 4 bagian
 - Input values atau one input layer
 - Weights dan Bias
 - Net sum
 - Activation Function



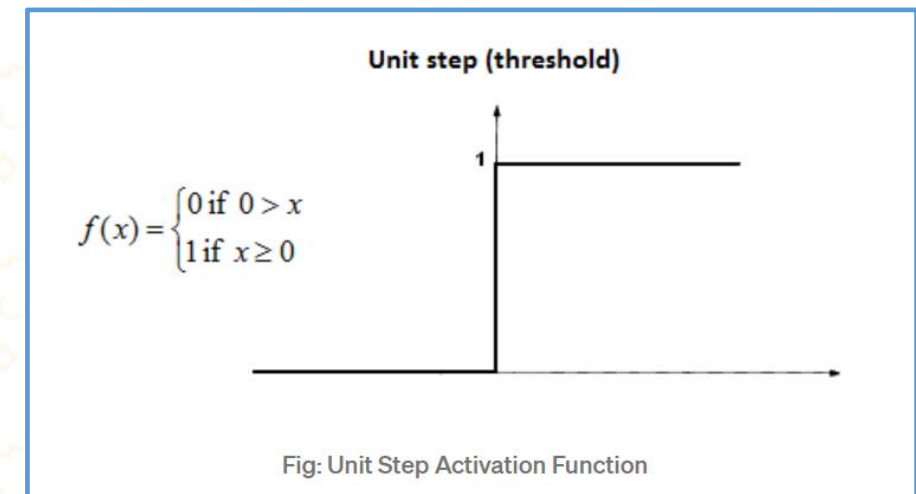


Perceptron (how does it work?)

- Semua inputs x dikalikan dengan weights w
- Jumlahkan semua hasil perkalian tersebut (kita sebut *Weighted Sum*)
- Mengaplikasikan activation function ke weighted sum



$$y = f(0.3 \cdot x_1 + 0.3 \cdot x_2 + 0.3 \cdot x_3 - 0.4)$$





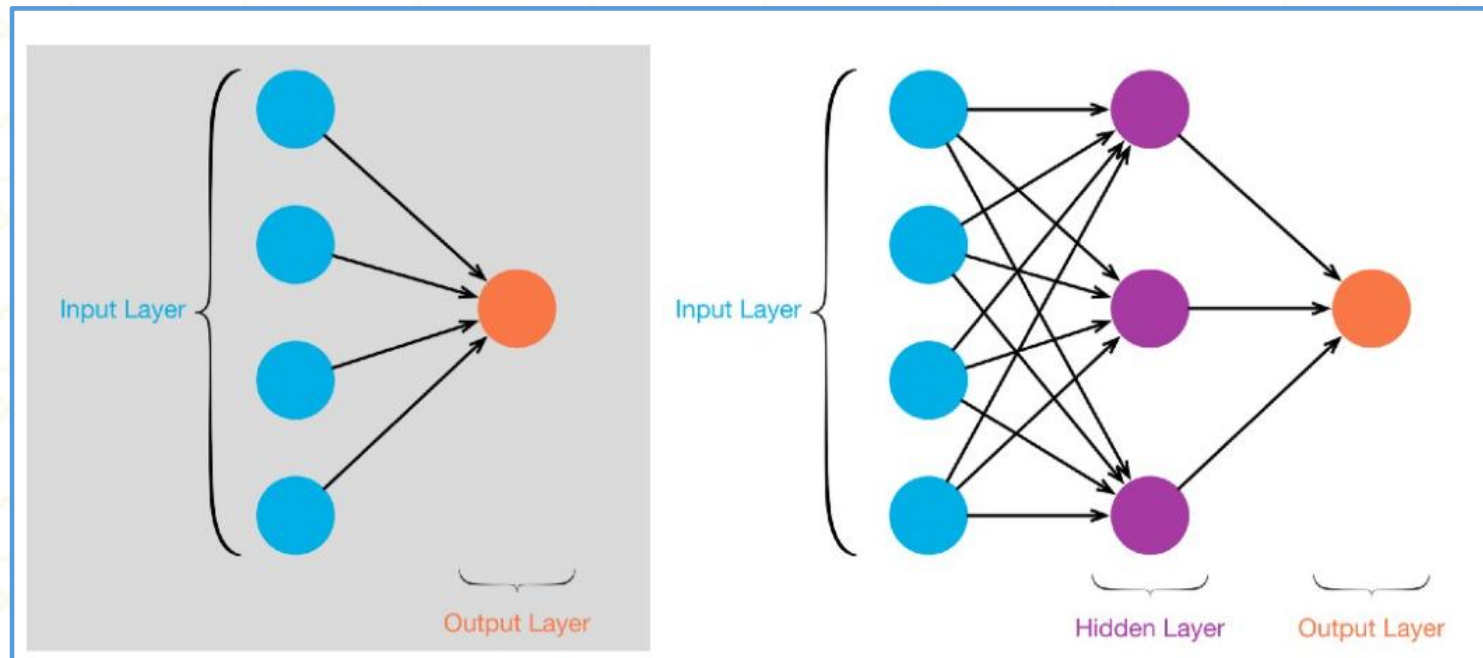
Multilayer Perceptron





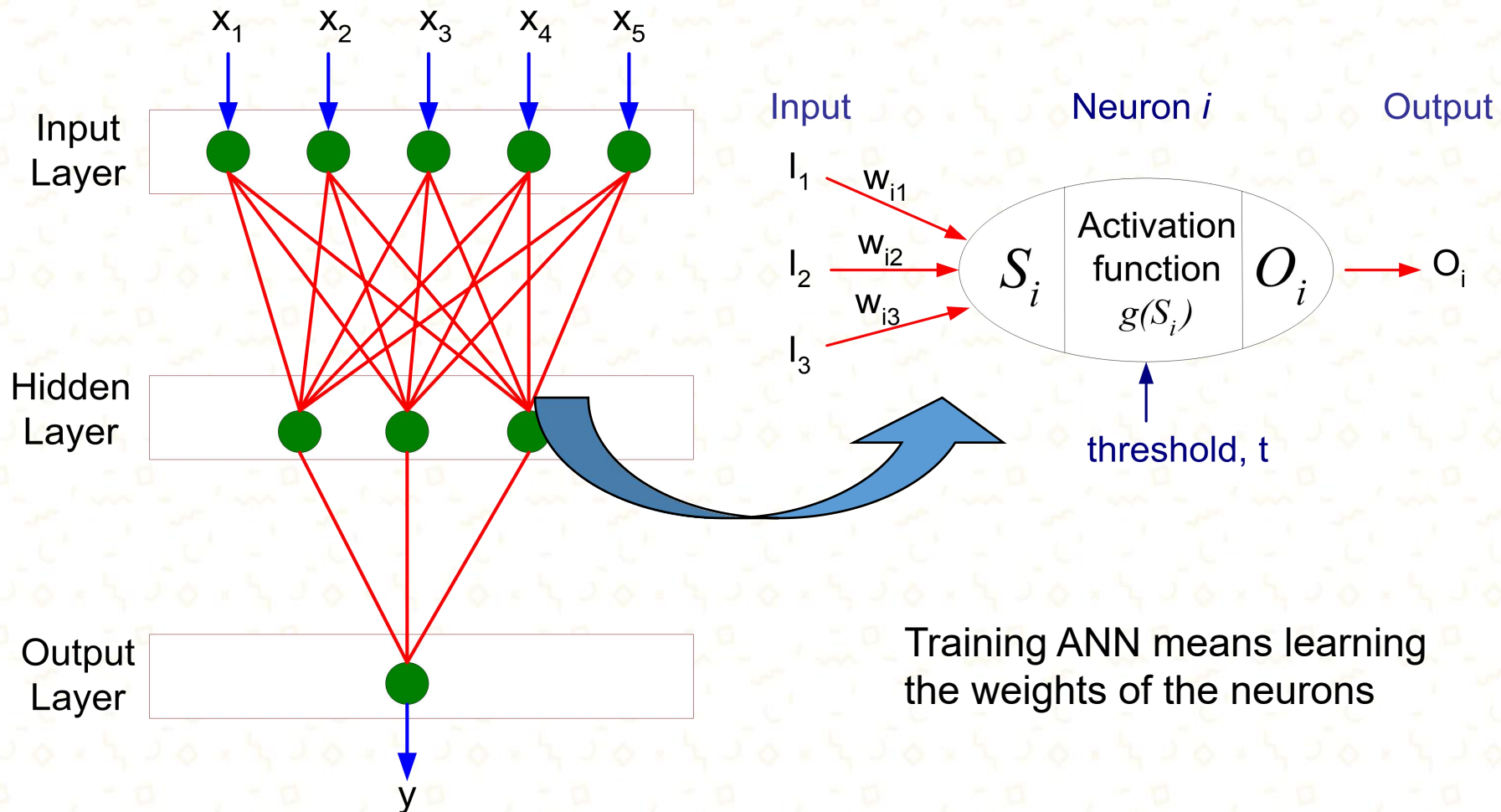
Multilayer perceptron

- Model ini terdiri dari tiga jenis layer — input layer, hidden layer, output layer.
- Kecuali node input, setiap node adalah neuron yang menggunakan fungsi aktivasi nonlinier.
- MLP menggunakan backpropagation untuk training-nya.





General Structure of MLP

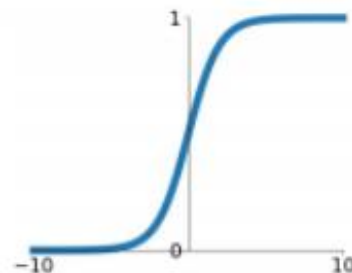




Activation function

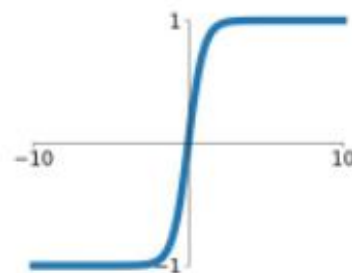
Sigmoid

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



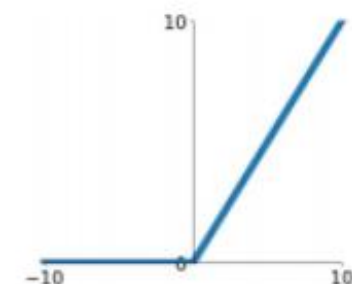
tanh

$$\tanh(x)$$



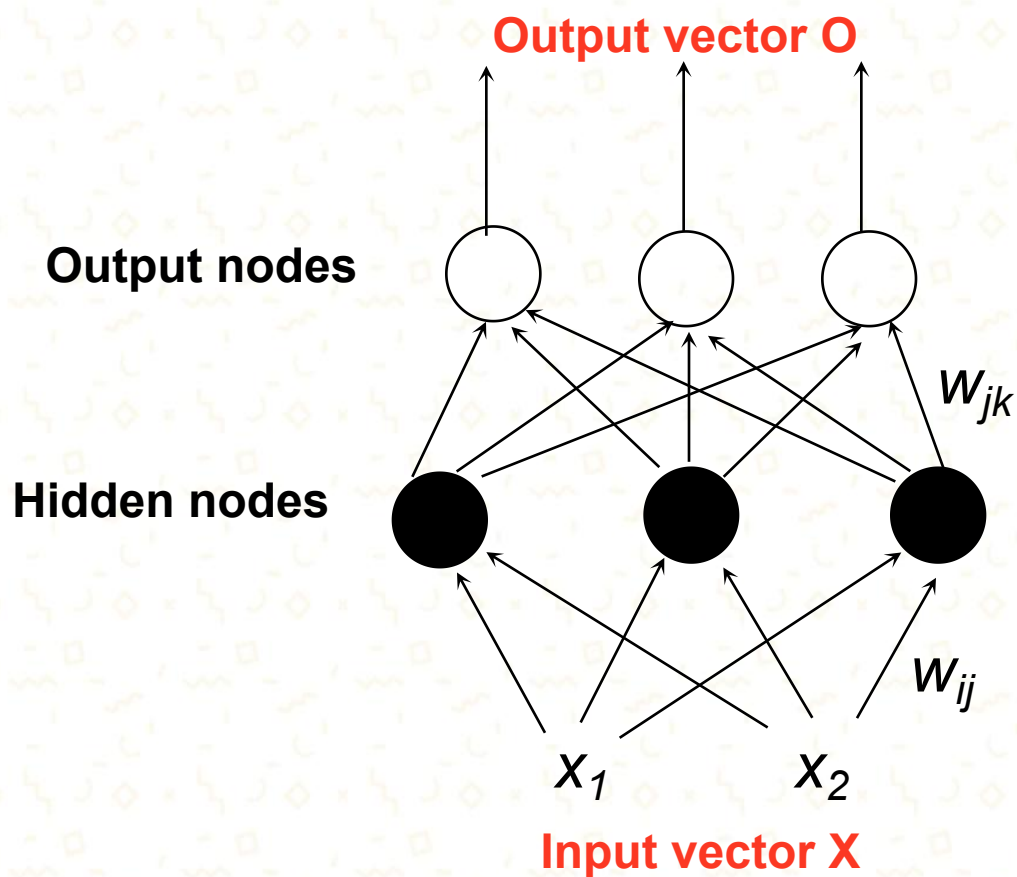
ReLU

$$\max(0, x)$$





Multilayer perceptron



$$O_k = \frac{1}{1 + e^{-\sum h_j w_{jk} + \theta_k}}$$

$$h_j = \frac{1}{1 + e^{-\sum x_i w_{ij} + \theta_j}}$$



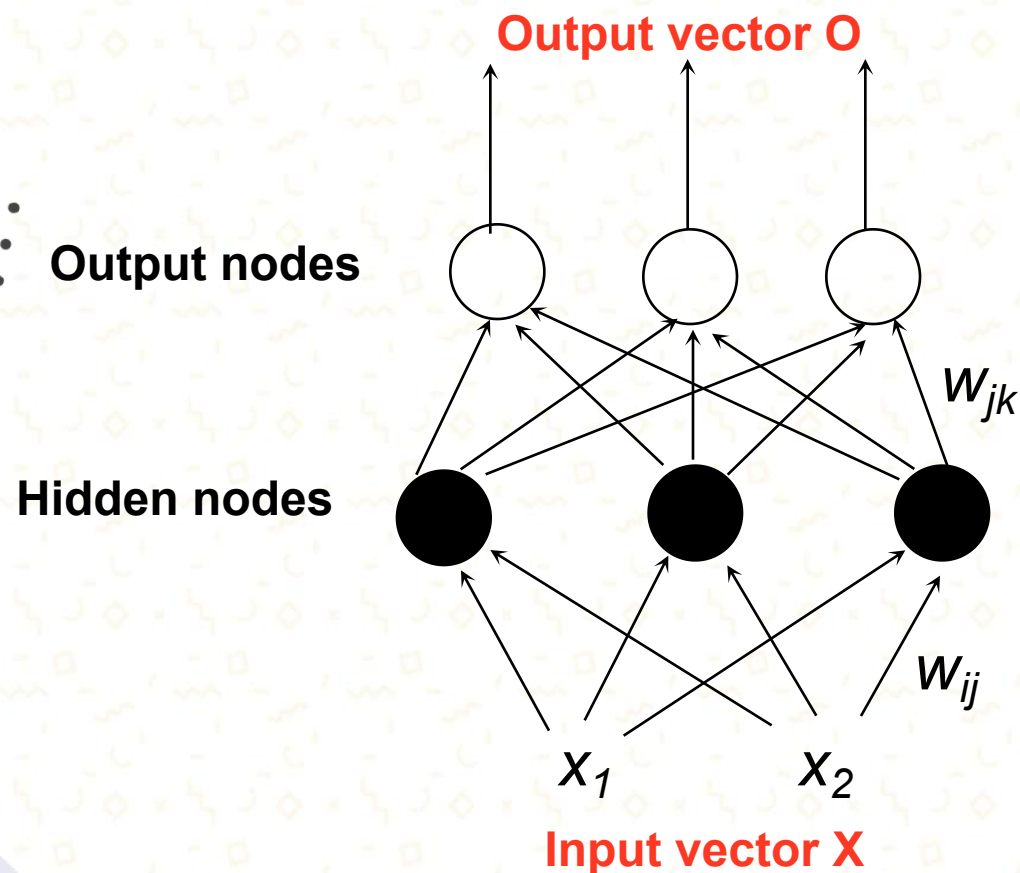
Network Training

- Tujuan dari training
 - Mendapatkan weights yang membuat hampir semua sampel dalam training data dapat diklasifikasikan dengan benar
- Langkah-langkah
 - Inisialisasi weights w_{ij} dengan nilai acak(random)
 - Masukkan (feed) sample training X ke dalam jaringan satu persatu
 - Untuk setiap unit
 - Hitung output value O dengan mengaplikasikan activation function
 - Hitung error E
 - Update weights w_{ij} dan biases





Backpropagation Learning



$$Err_k = (T_k - O_k)$$

O_k

h_j

$$\delta_k = O_k(1 - O_k)(T_k - O_k)$$
$$w_{jk} = w_{jk} + (l)\delta_k O_k$$

$$\delta_j = h_j(1 - h_j) \sum_k \delta_k w_{jk}$$
$$w_{ij} = w_{ij} + (l)\delta_j h_j$$

Given (X, T)



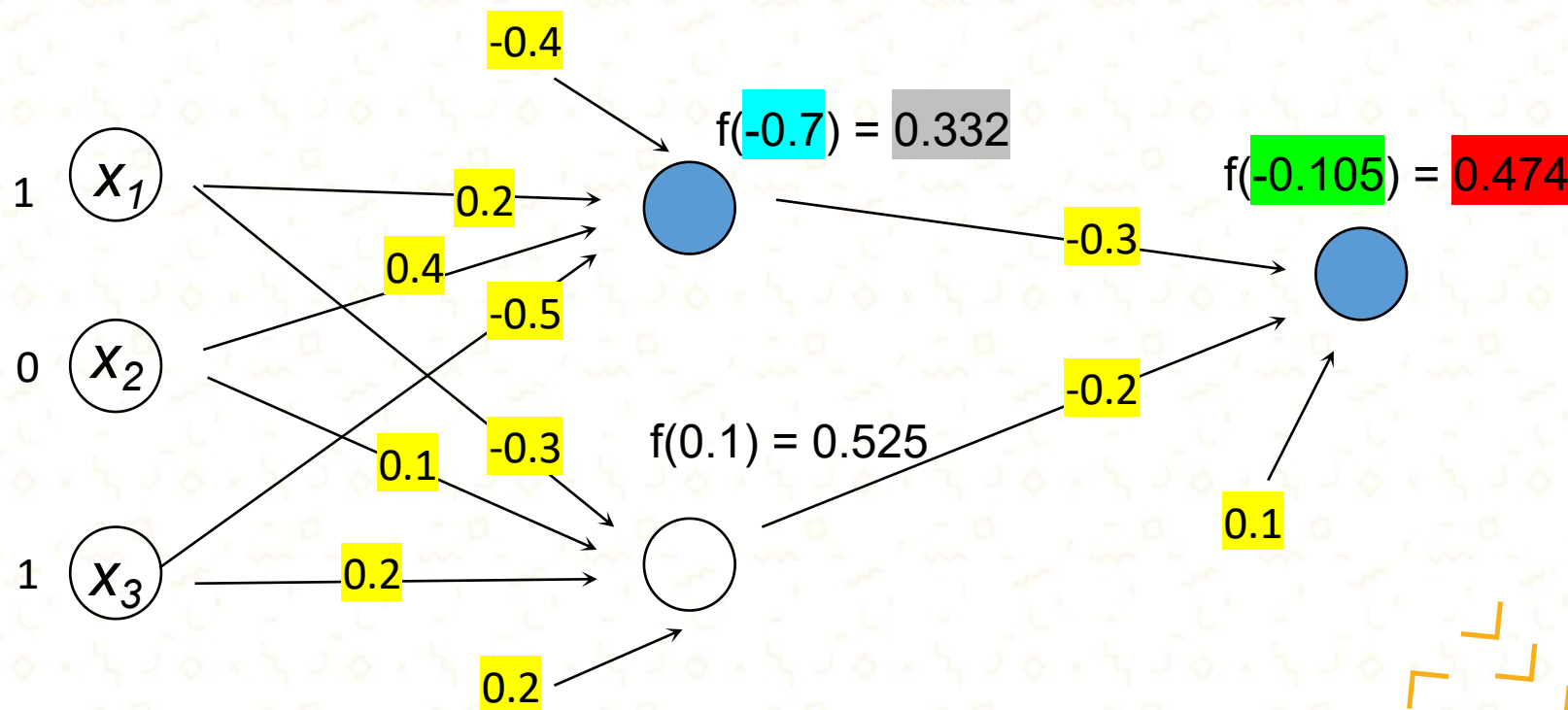
Contoh

Input vector X

X ₁	X ₂	X ₃	Y
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	0
0	1	0	0
0	1	1	1
0	0	0	0

Hidden nodes

Output nodes



- Net Input $I_j = 0.2 + 0 + -0.5 - 0.4 = -0.7$
- Output $O_j = (1/(1+e^{0.7})) = 0.332$
- Net Input $I_k = -0.3(0.332) - (0.2)(0.525) + 0.1 = -0.105$
- Output $O_k = (1/(1+e^{0.105})) = 0.474$



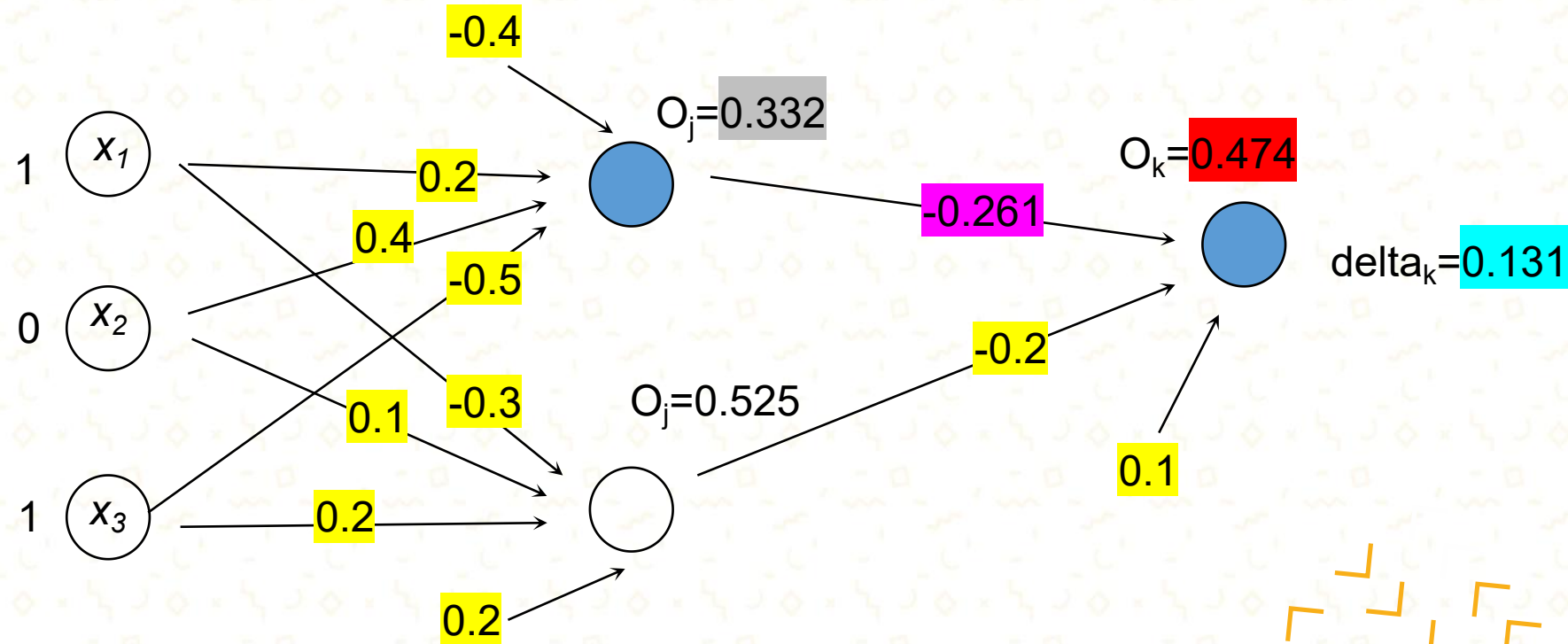
Contoh

Input vector X

X ₁	X ₂	X ₃	Y
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	0
0	1	0	0
0	1	1	1
0	0	0	0

Hidden nodes

Output nodes



- $\text{delta}_k = (0.474)(1 - 0.474)(1 - 0.474) = 0.131$
- $w_{jk} = -0.3 + (0.9)(0.131)(0.332) = -0.261$
- Dan seterusnya..



Discussion on NN

- Keuntungan
 - **Robust** -berfungsi baik ketika training set mengandung error
 - Output bisa discrete, real-valued, atau vector
- Criticism
 - Waktu yang lama saat training
 - Sulit untuk dipahami





MLP in Sklearn

- Sklearn menyediakan class MLPClassifier

```
class sklearn.neural_network.MLPClassifier(hidden_layer_sizes=(100), activation='relu', *, solver='adam', alpha=0.0001,
batch_size='auto', learning_rate='constant', learning_rate_init=0.001, power_t=0.5, max_iter=200, shuffle=True, random_state=None,
tol=0.0001, verbose=False, warm_start=False, momentum=0.9, nesterovs_momentum=True, early_stopping=False,
validation_fraction=0.1, beta_1=0.9, beta_2=0.999, epsilon=1e-08, n_iter_no_change=10, max_fun=15000)
```

[source]

Hyperparameter	Description
hidden_layer_sizes	Jumlah hidden layer dan node-nya
activation	Activation function
max_iter	Jumlah iterasi



Lets Practice!



Thank
YOU