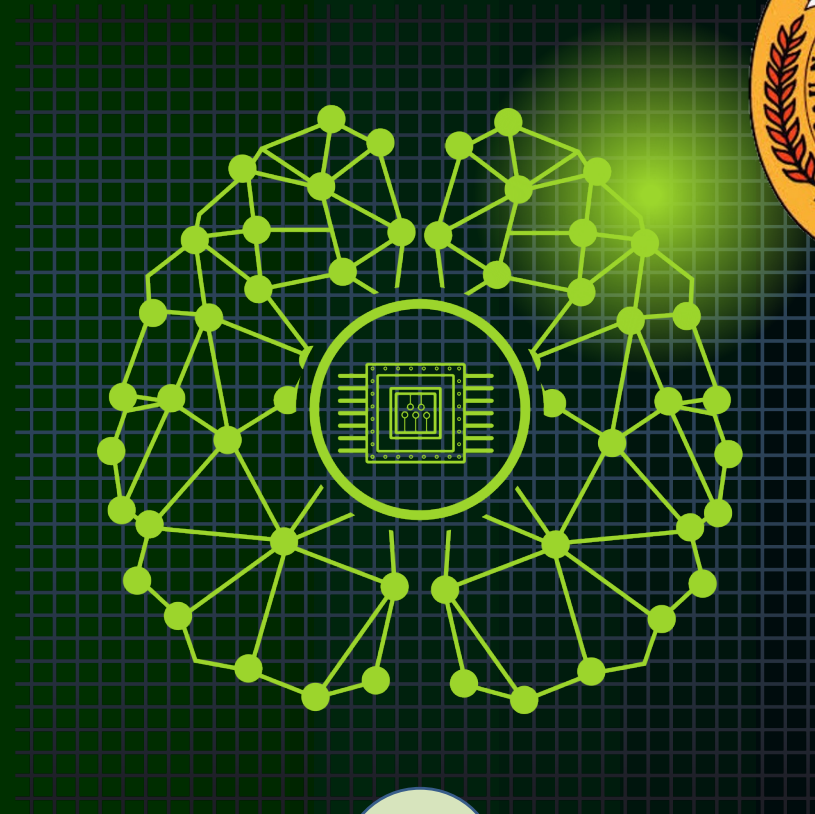


EIE6012

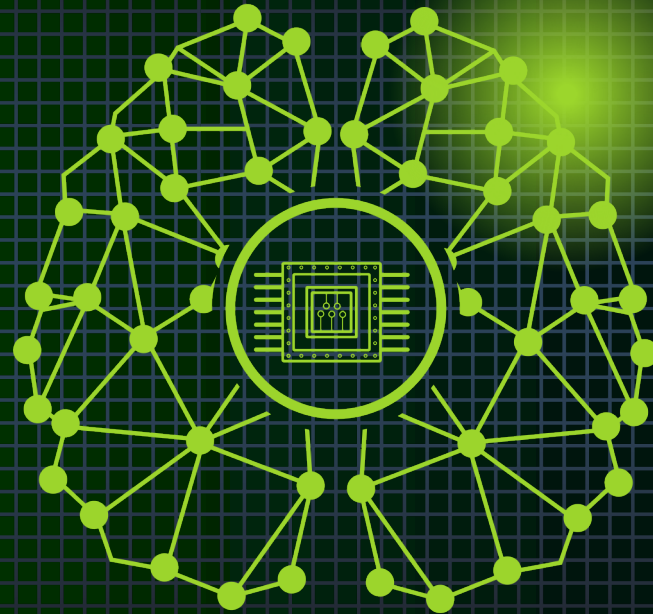
MACHINE LEARNING

ENDANG SRI RAHAYU



6

DEEP LEARNING



Tujuan Pembelajaran

Setelah mengikuti kuliah ini, mahasiswa diharapkan mampu:

- Memahami konsep dasar Deep Learning dan hubungannya dengan Machine Learning serta Artificial Intelligence.
- Menjelaskan struktur dasar Artificial Neural Network (ANN) dan proses training-nya.
- Mengenal istilah penting: neuron, layer, weight, bias, activation function.
- Membangun dan melatih model jaringan saraf sederhana menggunakan Python (Keras).

Apa itu Deep Learning?

Deep Learning (DL) adalah cabang dari Machine Learning (ML) yang menggunakan Artificial Neural Networks (ANN) dengan banyak lapisan (layers).

“Deep” berarti model memiliki lebih dari satu hidden layer, yang membuatnya mampu belajar pola yang sangat kompleks dari data.

AI → Machine Learning → Deep Learning

Level	Contoh	Tujuan
Artificial Intelligence	Sistem cerdas (misal chatbot, mobil otonom)	Meniru kecerdasan manusia
Machine Learning	Algoritma pembelajaran dari data	Membuat prediksi/pola
Deep Learning	Neural network dengan banyak layer	Mengenali pola kompleks (gambar, suara, teks)

Struktur Jaringan Saraf Tiruan (Artificial Neural Network)

Komponen utama:

1. **Neuron (Node):** unit pemroses sederhana (mirip sel otak biologis).
 2. **Layer (Lapisan):**
 1. *Input layer:* menerima data.
 2. *Hidden layer:* memproses informasi dan belajar pola.
 3. *Output layer:* menghasilkan prediksi.
 3. **Weight (Bobot):** menentukan kekuatan hubungan antar neuron.
 4. **Bias:** membantu model menyesuaikan hasil prediksi.
 5. **Activation Function:** menentukan apakah neuron aktif atau tidak (contoh: *ReLU, Sigmoid, Softmax*).
-

Proses Pembelajaran dalam Deep Learning

1. Forward Propagation:

Data mengalir dari input → hidden → output untuk menghasilkan prediksi.

2. Loss Calculation:

Hitung error antara hasil prediksi dan target sebenarnya.

3. Backward Propagation:

Model memperbarui bobot untuk meminimalkan error menggunakan *gradient descent*.

4. Iterasi:

Proses diulang ribuan kali hingga model menemukan pola terbaik.

contoh dataset MNIST (angka tulisan tangan 0–9).

```
[4] ✓ 55s
# === Import Library ===
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten
from tensorflow.keras.datasets import mnist

# === 1. Load Dataset ===
(x_train, y_train), (x_test, y_test) = mnist.load_data()

# Normalisasi nilai pixel (0-255 → 0-1)
x_train, x_test = x_train / 255.0, x_test / 255.0

# === 2. Bangun Model ===
model = Sequential([
    Flatten(input_shape=(28, 28)),      # Ubah gambar 28x28 menjadi vektor 784 elemen
    Dense(128, activation='relu'),      # Hidden layer 1
    Dense(64, activation='relu'),       # Hidden layer 2
    Dense(10, activation='softmax')     # Output layer (10 kelas: angka 0-9)
])

# === 3. Kompilasi Model ===
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

# === 4. Latih Model ===
model.fit(x_train, y_train, epochs=5, batch_size=32, validation_data=(x_test, y_test))

# === 5. Evaluasi Model ===
loss, accuracy = model.evaluate(x_test, y_test)
print(f"Akurasi Model: {accuracy:.2f}")
```

- **Flatten** mengubah gambar menjadi vektor.
- **Dense** membuat layer fully connected (semua neuron terhubung).
- **ReLU** mempercepat konvergensi (aktivasi non-linear).
- **Softmax** menghasilkan probabilitas untuk setiap kelas (0–9).

Perbandingan ML vs DL

Aspek	Machine Learning	Deep Learning
Fitur	Harus diekstraksi manual	Belajar fitur otomatis
Data	Cukup sedikit	Butuh data besar
Komputasi	Relatif ringan	Butuh GPU/komputasi tinggi
Interpretasi	Lebih mudah dijelaskan	Lebih kompleks (“black box”)
Contoh	Linear Regression, SVM	CNN, RNN, Autoencoder