

Ringkasan Buku 1

Nama Buku: Tensorflow in Actions

Team: Wilhelmina Arlene, Luthfiah Maulidya, Fadhilah Dwi Istiani

Chapter 4 — *Dipping Toes in Deep Learning*

Tujuan Utama Chapter 4

Chapter 4 adalah **titik transisi penting** dalam buku ini.

Jika:

- Chapter 1 = *konsep & motivasi*
- Chapter 2 = *fondasi teknis TensorFlow*
- Chapter 3 = *alat membangun model & data*

maka Chapter 4 menjawab:

"Bagaimana deep learning bekerja secara nyata, langkah demi langkah?"

Fokus chapter ini:

- mengenalkan **tiga jenis neural network utama**
- membangun model **end-to-end**
- memahami **alur data** → **model** → **output**
- bukan sekadar "pakai", tapi **paham kenapa**

STRUKTUR BESAR CHAPTER 4

Chapter 4 dibagi menjadi **3 eksperimen utama**:

1. **Fully Connected Network (FCN)**
2. **Convolutional Neural Network (CNN)**
3. **Recurrent Neural Network (RNN)**

Setiap eksperimen selalu mengikuti pola:

1. memahami data
2. membangun model
3. training
4. evaluasi
5. interpretasi hasil

4.1 Fully Connected Networks (FCN)

Apa itu Fully Connected Network?

FCN adalah:

- neural network paling dasar
- setiap neuron terhubung ke semua neuron di layer berikutnya

Struktur umum:

Input → Dense → Dense → Output

4.1.1 Memahami Data

Menjelaskan:

- **jangan langsung bikin model**
- pahami dulu:
 - bentuk data
 - fitur
 - target

Alasan:

Banyak kegagalan model bukan karena arsitektur, tapi karena **salah memahami data**.

4.1.2 Autoencoder sebagai Contoh FCN

Apa itu Autoencoder?

Autoencoder adalah model yang:

- **input = output**
- belajar membuat representasi padat (*latent representation*)

Struktur:

Input → Encoder → Latent Space → Decoder → Output

Tujuan:

- dimensionality reduction
- feature learning
- denoising

Konsep Penting Autoencoder

- bukan supervised learning klasik
- loss = perbedaan input vs output
- encoder mempelajari fitur penting data

👉 **Insight penting:**

Autoencoder menunjukkan bahwa neural network **tidak selalu untuk klasifikasi**.

4.2 Convolutional Neural Networks (CNN)

Kenapa CNN Dibutuhkan?

FCN **tidak cocok untuk gambar** karena:

- terlalu banyak parameter
- tidak mempertahankan struktur spasial

CNN mengatasi dengan:

- **local connectivity**
- **weight sharing**

4.2.1 Memahami Data Gambar

Gambar direpresentasikan sebagai:

- tensor 3D: (height, width, channels)
- batch gambar → tensor 4D

Menjelaskan:

- penting memahami **dimensi data**
- salah dimensi = error paling umum

4.2.2 Implementasi CNN

Komponen utama CNN:

1. **Convolution layer**
 - ekstraksi fitur
 - edge, texture, pattern
2. **Activation (ReLU)**
 - non-linearity
3. **Pooling**
 - reduksi dimensi
 - stabilitas fitur
4. **Dense layer**
 - klasifikasi akhir

Alur CNN Secara Konseptual

Image → Conv → ReLU → Pool → Conv → Pool → Flatten → Dense → Output

Insight Penting CNN

- CNN belajar fitur **secara hierarkis**
- layer awal → fitur sederhana
- layer akhir → fitur kompleks

4.3 Recurrent Neural Networks (RNN)

Kenapa RNN?

FCN & CNN:

- tidak memiliki memori
- tidak cocok untuk data berurutan

RNN dirancang untuk:

- time series
- teks
- data sekuensial

4.3.1 Memahami Data Sekuensial

Karakteristik data:

- ada urutan
- konteks masa lalu penting
- contoh:
 - suhu harian
 - harga saham
 - CO₂ level

4.3.2 Implementasi RNN

Konsep utama:

- hidden state
- informasi masa lalu memengaruhi output saat ini

Masalah RNN klasik:

- vanishing gradient
- sulit belajar dependensi panjang

Solusi modern:

- **LSTM**
- **GRU**

4.3.3 Studi Kasus: Prediksi CO₂

Model digunakan untuk:

- memprediksi nilai CO₂ masa depan
- berdasarkan nilai sebelumnya

Pipeline:

1. windowing data
2. membentuk sequence
3. training RNN
4. evaluasi prediksi

👉 **Pesan penting:**

RNN bukan sekadar "layer baru", tapi **cara berpikir baru tentang data.**