

## Bab 15: Processing Sequences Using RNNs and CNNs

### Tujuan Bab

Memahami bagaimana menangani data sekuensial (seperti teks, audio, atau data time series) menggunakan Recurrent Neural Networks (RNN) dan juga CNN yang dimodifikasi untuk sekuens.

### Konsep Utama

#### 1. Masalah pada Data Sekuensial

Data sekuensial memiliki urutan, sehingga perlu arsitektur yang dapat menangkap keterkaitan temporal antar elemen (misalnya, kata dalam kalimat, suhu dalam hari berturut-turut).

#### 2. Recurrent Neural Networks (RNN)

RNN mengolah input satu elemen sekuens per langkah waktu sambil mempertahankan state (memori).

Struktur dasar:

$x_t \rightarrow [\text{RNN cell}] \rightarrow h_t \rightarrow \text{output}$

↑

$h_{t-1}$

Keras RNN sederhana:

```
model = keras.models.Sequential([
    keras.layers.SimpleRNN(20, return_sequences=True, input_shape=[None, 1]),
    keras.layers.SimpleRNN(20),
    keras.layers.Dense(1)
])
```

#### 3. Masalah pada RNN Standar

- Vanishing gradient: Gradient mengecil saat propagasi ke belakang
- Sulit mengingat hubungan jangka panjang

#### 4. LSTM dan GRU

Solusi RNN yang lebih stabil:

Arsitektur	Kelebihan
LSTM	Menggunakan gate untuk mengontrol memori jangka panjang
GRU	Lebih ringan dari LSTM, dengan performa sebanding

Contoh:

```
keras.layers.LSTM(20, return_sequences=True)
keras.layers.GRU(20)
```

#### 5. Bidirectional RNN

Mengakses informasi masa lalu dan masa depan dalam sekuens.

```
keras.layers.Bidirectional(keras.layers.LSTM(20))
```

#### 6. 1D Convolution for Sequences

CNN juga bisa digunakan untuk data sekuensial, dikenal sebagai Conv1D. Lebih cepat dari RNN dan bisa efektif untuk pola lokal.

```
model = keras.models.Sequential([
    keras.layers.Conv1D(32, kernel_size=3, activation="relu", input_shape=[None, 1]),
    keras.layers.MaxPooling1D(pool_size=2),
    keras.layers.LSTM(20),
    keras.layers.Dense(1)
```

])

## 7. Embedding Layer

Digunakan untuk data kategorikal berurutan seperti teks. Setiap kata direpresentasikan sebagai vektor dens.

```
keras.layers.Embedding(input_dim=10000, output_dim=128)
```

## Proyek / Notebook Praktik

Isi:

- Prediksi time series (misalnya suhu) menggunakan RNN dan LSTM
- Membangun jaringan LSTM, GRU, dan CNN untuk data sekuensial
- Penerapan embedding untuk teks
- Analisis perbandingan performa berbagai arsitektur

## Inti Pelajaran

Konsep	Penjelasan
RNN	Cocok untuk data berurutan, tapi rawan vanishing gradient
LSTM / GRU	RNN yang lebih stabil dan efektif untuk long-term dependency
Bidirectional RNN	Menggunakan informasi dua arah
Conv1D	Alternatif cepat untuk mengenali pola lokal dalam sekuens
Embedding	Mengonversi kategori (mis. kata) menjadi vektor dens

---

## Kesimpulan

Bab ini memperkenalkan cara kerja dan implementasi berbagai arsitektur neural network untuk data sekuensial. RNN cocok untuk memproses data berurutan, namun arsitektur modern seperti LSTM, GRU, dan bahkan CNN 1D memberi hasil yang lebih stabil dan efisien. Ini membentuk fondasi kuat untuk NLP, prediksi time series, dan pemrosesan sinyal.