

Deep Learning Sekuensial: RNN, LSTM, dan GRU

Arsitektur Gerbang Memori, Masalah Vanishing Gradient, dan Peramalan Daya EBT Surya/Angin

Ir. Novalio Daratha, S.T., M.Sc., Ph.D.

Program Studi Teknik Elektro – Fakultas Teknik
Universitas Bengkulu

Minggu 11 / Semester Ganjil 2026

Agenda Perkuliahan Minggu ke-11

Capaian Pembelajaran (Sub-CPMK 10)

Mahasiswa mampu menyusun tensor 3D sekuensial, memahami mekanisme gerbang arsitektur LSTM dan GRU, mengatasi masalah kelenyapan gradien (*vanishing gradient*), serta melatih jaringan *Deep Learning* untuk peramalan daya pembangkit EBT fluktuatif.

- 1 Keterbatasan Jaringan Saraf Konvensional (MLP)
- 2 Recurrent Neural Network (RNN) Standar
- 3 Masalah Vanishing & Exploding Gradient
- 4 Arsitektur Long Short-Term Memory (LSTM)
- 5 Arsitektur Gated Recurrent Unit (GRU)
- 6 Implementasi PyTorch: Peramalan Daya PLTS

Dari MLP ke Recurrent Neural Network (RNN)

Mengapa MLP Kurang Efektif?

- Mengasumsikan seluruh sampel input independen satu sama lain.
- Tidak memiliki memori internal untuk mempertahankan informasi riwayat urutan temporal yang panjang.

Mekanisme RNN

Memiliki status tersembunyi (*Hidden State* h_t) yang berulang (*recurrent loop*):

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t + b_h) \quad (1)$$

$$\hat{y}_t = W_{hy}h_t + b_y \quad (2)$$

Arsitektur Long Short-Term Memory (LSTM)

Mengatasi *vanishing gradient* dengan jalur sel memori bebas hambatan (*Constant Error Carousel*):

Mekanisme 3 Gerbang LSTM

Forget Gate: $f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$ (Membuang memori usang)

Input Gate: $i_t = \sigma(W_i[h_{t-1}, x_t] + b_i)$, $\tilde{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_c)$

Cell State: $C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$ (Memperbarui akumulasi memori)

Output Gate: $o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$, $h_t = o_t \odot \tanh(C_t)$

Gated Recurrent Unit (GRU): Varian Lebih Ringkas

Menggabungkan *Cell State* dan *Hidden State* serta mereduksi jumlah gerbang menjadi 2:

Persamaan GRU

Reset Gate: $r_t = \sigma(W_r[h_{t-1}, x_t])$

Update Gate: $z_t = \sigma(W_z[h_{t-1}, x_t])$

Candidate: $\tilde{h}_t = \tanh(W[r_t \odot h_{t-1}, x_t])$

Hidden State: $h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$

Perbandingan Rekayasa: LSTM vs GRU

- **GRU**: Parameter 25% lebih sedikit, konvergensi lebih cepat pada dataset terbatas (misal data sensor mikrogrid).
- **LSTM**: Lebih tangguh menangani ketergantungan temporal jangka panjang yang sangat rumit.

Implementasi PyTorch: Arsitektur LSTM Model

```
import torch
import torch.nn as nn

class SolarLSTM(nn.Module):
    def __init__(self, n_features=4, hidden_size=64, num_layers=2, pred_len=24):
        super(SolarLSTM, self).__init__()
        self.lstm = nn.LSTM(input_size=n_features, hidden_size=hidden_size,
                             num_layers=num_layers, batch_first=True, dropout=0.2)
        self.fc = nn.Linear(hidden_size, pred_len)

    def forward(self, x):
        # Input x shape: (batch_size, seq_len, n_features)
        lstm_out, _ = self.lstm(x)
        # Ambil hidden state terakhir dari sequence
        last_hidden = lstm_out[:, -1, :]
        out = self.fc(last_hidden) # Output: (batch_size, pred_len)
        return out
```

Rangkuman & Persiapan Minggu ke-12

Poin Kunci Minggu 11

- LSTM dan GRU mengatasi kelemahan mendasar RNN klasik dalam menangkap memori jangka panjang.
- Gerbang non-linier bertindak sebagai saklar kendali aliran informasi internal jaringan.
- Penskalaan data 3D Tensor (N, T, D) wajib dinormalisasi secara hati-hati agar gradien stabil.

Materi Minggu Depan (Minggu ke-12)

Arsitektur Sekuensial Lanjut: 1D-Temporal CNN, Encoder-Decoder (Seq2Seq), dan Attention Mechanism / Transformer untuk peramalan multi-bus simultan.

Terima Kasih

Pertanyaan & Diskusi Deep Learning Sekuensial

Laboratorium Sistem Tenaga & Komputasi Cerdas
Jurusan Teknik Elektro – Universitas Bengkulu