

Chapter 15: Summary

Diedrick Darrell Darmadi - 1103223031

Bab 15 membahas bagaimana menangani data berurutan (*sequential data*) menggunakan **Recurrent Neural Networks (RNNs)** dan variasinya seperti **LSTM** dan **GRU**. Bab ini dimulai dengan menjelaskan mengapa data sekuensial tidak bisa diproses dengan cara yang sama seperti data tabular atau gambar. Pada data seperti teks, deret waktu (*time series*), log aktivitas, sinyal sensor, dan audio, nilai pada waktu berikutnya bergantung pada nilai sebelumnya. RNN dirancang untuk kondisi ini karena memiliki *state* internal yang diteruskan dari waktu ke waktu, memungkinkan model mengingat informasi sejarah saat memproses input baru.

Bab ini kemudian menjelaskan cara kerja **RNN dasar**, yang terdiri dari *cell* yang menerima input saat ini dan *hidden state* dari langkah sebelumnya. Namun, RNN klasik terbukti sulit dilatih pada sekuens panjang karena mengalami **vanishing gradient**, sehingga model sulit mengingat informasi jauh ke belakang. Untuk mengatasi hal ini, dikembangkan arsitektur **Long Short-Term Memory (LSTM)**, yang memperkenalkan tiga *gate* (input, forget, output gate) dan sebuah *cell state* panjang yang memungkinkan jaringan mempertahankan atau menghapus informasi sesuai kebutuhan. LSTM mampu mempelajari ketergantungan jangka panjang sehingga sangat efektif untuk teks, audio, dan *time series* kompleks.

Selain LSTM, bab ini juga memperkenalkan **Gated Recurrent Unit (GRU)**, versi yang lebih sederhana tetapi sering memiliki performa setara atau lebih baik. GRU menggabungkan beberapa gerbang menjadi struktur yang lebih ringkas, sehingga lebih cepat dilatih dan cocok untuk dataset yang lebih besar atau waktu pelatihan yang terbatas. Bab ini juga membahas penggunaan **Bidirectional RNN**, yang memproses urutan dari dua arah (maju dan mundur), sehingga model dapat memahami konteks dari masa lalu maupun masa depan. Pendekatan ini sangat berguna untuk masalah seperti analisis sentimen, named entity recognition, atau tugas NLP lain yang membutuhkan konteks penuh.

Setelah memahami arsitektur dasar, bab ini memperkenalkan pipeline preprocessing untuk teks dan sequence, termasuk tokenisasi, padding, windowing untuk *time series*, serta penggunaan lapisan **Embedding** untuk memetakan kata ke vektor. Layer embedding sangat penting karena representasi one-hot terlalu besar dan tidak efisien. Bab ini menjelaskan bagaimana embedding belajar representasi semantik kata secara otomatis selama pelatihan.

Bab ini juga mengajarkan cara membangun berbagai jenis model sequence-to-sequence (seq2seq), misalnya memprediksi nilai masa depan pada *time series*, mendeteksi anomali, atau menghasilkan teks sederhana. Untuk tugas prediksi *time series*, bab ini memperkenalkan *sliding window*, *input-output sequences*, dan *multi-step forecasting*. Untuk tugas NLP dasar, bab ini menunjukkan bagaimana membangun model untuk analisis sentimen, prediksi kata berikutnya, dan klasifikasi teks.

Bab 15 juga menjelaskan konsep penting seperti **stateful vs stateless RNN**, di mana *stateful RNN* mempertahankan state antar batch sehingga cocok untuk data streaming atau deret sangat panjang. Namun, model *stateful* lebih sulit dilatih dan memerlukan pengelolaan batch yang konsisten. Sebaliknya, *stateless RNN* lebih mudah digunakan dan cukup untuk sebagian besar aplikasi.

Pada bagian akhir, bab ini membahas teknik pelatihan RNN yang efektif, termasuk **Truncated Backpropagation Through Time (TBPTT)**, cara menghindari exploding gradient menggunakan **gradient clipping**, serta tips penggunaan dropout khusus RNN (*recurrent dropout*) agar model tidak

overfitting. Bab ini juga menyoroti bahwa meskipun RNN dan turunannya sangat kuat, mereka masih memiliki keterbatasan untuk urutan panjang, sehingga model berbasis **Attention** dan **Transformer**