

Chapter 17: Summary

Diedrick Darrell Darmadi - 1103223031

Bab 17 membahas secara mendalam konsep **Autoencoder**, yaitu arsitektur neural network yang dilatih untuk merekonstruksi inputnya sendiri. Autoencoder digunakan bukan untuk prediksi label, melainkan untuk mempelajari **representasi terkompresi** (*latent representation*) dari data. Bab ini dimulai dengan motivasi mengapa kompresi dan pemahaman struktur internal data penting, terutama pada tugas-tugas seperti pengurangan dimensi, deteksi anomali, denoising, dan generasi data baru. Tidak seperti PCA yang hanya linear, autoencoder mampu belajar representasi non-linear yang lebih kaya dan fleksibel.

Konsep dasar autoencoder terdiri dari dua bagian: **encoder** dan **decoder**. Encoder bertugas mengubah input menjadi vektor berukuran kecil yang disebut *latent vector*, sementara decoder merekonstruksi kembali input dari latent vector tersebut. Bab ini menunjukkan bahwa jika kapasitas model dibatasi, autoencoder dipaksa belajar fitur esensial dalam data, bukan sekadar menghafal. Namun, jika model terlalu besar, autoencoder dapat dengan mudah melakukan *copy* data sehingga tidak belajar representasi berguna. Oleh karena itu, arsitektur dan regularisasi sangat penting.

Bab ini membahas berbagai variasi autoencoder. Pertama adalah **Stacked Autoencoder**, yaitu autoencoder dengan banyak hidden layer untuk menangkap pola kompleks. Stacked Autoencoder dapat digunakan untuk *pretraining* jaringan dalam, terutama ketika dataset label kecil. Kemudian dibahas **Denoising Autoencoder**, yang dilatih dengan input yang sengaja diberi noise tetapi targetnya adalah input asli. Teknik ini membuat encoder belajar fitur yang lebih stabil dan kuat terhadap gangguan. Selain itu, dijelaskan pula **Sparse Autoencoder**, yang mendorong activations menjadi jarang (*sparse*) sehingga masing-masing neuron belajar mendeteksi pola spesifik. Regularisasi seperti *L1 penalty* atau *KL divergence* digunakan untuk mencapai sparsity.

Salah satu bagian paling penting dari bab ini adalah pembahasan **Variational Autoencoder (VAE)**. VAE merupakan model generatif yang mempelajari distribusi probabilistik dari data, bukan hanya merekonstruksi secara deterministik. Tidak seperti autoencoder biasa yang menghasilkan latent vector tunggal, VAE menghasilkan *mean vector* dan *variance vector* yang merepresentasikan distribusi Gaussian dari latent space. Selama pelatihan, sampel ditarik dari distribusi ini menggunakan teknik *reparameterization trick*, sehingga memungkinkan backpropagation. VAE mampu menghasilkan data baru yang mirip dengan data asli, seperti gambar digit baru atau wajah sintetis. Bab ini membahas fungsi loss VAE yang merupakan gabungan reconstruction loss dan KL divergence.

Selain VAE, bab ini juga memperkenalkan **Convolutional Autoencoder**, yang sangat efektif untuk data gambar karena menggunakan convolutional encoder dan decoder. Dengan memanfaatkan spatial structure pada gambar, convolutional AE dapat belajar representasi lebih efisien dan menghasilkan rekonstruksi lebih baik dibanding AE berbasis Dense layer. Autoencoder konvolusional digunakan untuk tugas seperti denoising gambar, deteksi cacat pada permukaan industri, dan pengurangan dimensi pada data visual.

Bab ini juga mengulas **Sequence Autoencoders**, yaitu autoencoder berbasis RNN atau LSTM yang digunakan untuk data sekuens seperti teks atau time series. Sequence Autoencoder dapat digunakan untuk deteksi anomali pada data sensor, kompresi log aktivitas, atau tugas NLP tertentu. Model ini mempelajari bagaimana menyusun kembali urutan lengkap berdasarkan embedding sekuens ringkas yang dihasilkan encoder.

Pada bagian akhir, bab ini membahas penggunaan autoencoder untuk **anomaly detection**. Karena autoencoder dilatih untuk merekonstruksi data normal, data yang tidak sesuai dengan pola biasa akan menghasilkan error rekonstruksi tinggi. Teknik ini sangat efektif untuk mendeteksi fraud, kerusakan mesin, anomali data medis, dan lain-lain. Bab ini juga membahas praktik terbaik seperti regularisasi, bottleneck size, pemilihan arsitektur, dan monitoring loss agar autoencoder tidak overfitting.