

Analisis dan Penjelasan Kodingan

Struktur dan Fungsi Model

Kodingan ini menggunakan arsitektur **CNN (Convolutional Neural Network)** yang dirancang untuk memproses dataset **FashionMNIST**, sebuah dataset yang terdiri dari gambar pakaian dengan 10 kelas. Model memiliki dua lapisan konvolusi (conv1 dan conv2) dengan ukuran kernel yang dapat diatur melalui hyperparameter `kernel_size`. Setiap lapisan konvolusi diikuti oleh pooling layer, yang dapat berupa **MaxPooling** atau **AveragePooling**, tergantung nilai hyperparameter `pooling_type`. Dua lapisan fully connected (fc1 dan fc2) digunakan untuk memproses fitur hasil ekstraksi menjadi output berupa prediksi kelas. Kombinasi fungsi aktivasi **ReLU** digunakan untuk menambahkan elemen non-linearitas.

Fungsi forward mendefinisikan alur data mulai dari lapisan konvolusi dan pooling hingga transformasi ke dimensi vektor dan melalui lapisan fully connected. Padding pada lapisan konvolusi diatur sedemikian rupa agar ukuran output tetap stabil sesuai dengan input.

Proses Pelatihan dan Evaluasi

Proses pelatihan dimulai dengan inisialisasi model sesuai kombinasi hyperparameter dari `ParameterGrid`. Dataset **FashionMNIST** diproses dengan transformasi seperti konversi ke tensor dan normalisasi untuk memastikan data terstandarisasi. Data dibagi menjadi dua bagian: data latih (`train_loader`) dan data validasi (`val_loader`), dengan ukuran batch 64 untuk efisiensi komputasi.

Dua algoritma optimasi diterapkan:

- **SGD**: Learning rate 0.01 dengan momentum 0.9 untuk mempercepat konvergensi.
- **Adam**: Learning rate 0.001 untuk stabilitas dan kemampuan adaptasi yang lebih baik.

Scheduler `StepLR` diterapkan untuk mengurangi learning rate setiap 10 epoch dengan faktor 0.1. Selain itu, fitur **Early Stopping** digunakan untuk menghentikan pelatihan jika nilai `val_loss` tidak membaik setelah beberapa epoch berturut-turut (berdasarkan parameter `patience`). Fitur ini juga mencakup penghentian manual pada epoch tertentu (`stop_epochs`) untuk simulasi skenario tambahan.

Evaluasi Hyperparameter dan Output

Output menunjukkan performa model pada kombinasi hyperparameter yang berbeda. Misalnya, dengan `kernel_size=3`, `pooling_type=MaxPool`, dan optimizer **SGD**, Val Loss menurun secara konsisten dari 0.3647 ke 0.2268 hanya dalam 13 epoch sebelum Early Stopping diaktifkan. Kombinasi ini menunjukkan kemampuan untuk mengekstraksi fitur lebih efektif dengan jumlah epoch yang lebih sedikit. Di sisi lain, penggunaan pooling **AvgPool** menghasilkan penurunan loss yang lebih lambat tetapi lebih stabil, mencerminkan peran `AvgPool` sebagai metode regularisasi.

Kesimpulan

Arsitektur dan kombinasi hyperparameter yang digunakan menunjukkan fleksibilitas dan efektivitas model dalam menyelesaikan tugas klasifikasi. Kombinasi `kernel_size=3`, `pooling_type=MaxPool`, dan optimizer **SGD** memberikan hasil terbaik dengan penurunan Val Loss yang cepat dan akurat. Namun, mekanisme Early Stopping berhasil mencegah overfitting, terutama pada kombinasi dengan jumlah epoch yang tinggi. Dengan demikian, kodingan ini dirancang dengan baik untuk mengevaluasi berbagai konfigurasi hyperparameter secara efisien.