

Laporan Analisis Model CNN Dengan Dataset MNIST

Dalam pengenalan pola dan klasifikasi citra, model Convolutional Neural Network (CNN) telah menjadi salah satu metode yang paling efektif. Laporan ini berfokus pada analisis performa CNN dalam mengenali angka pada dataset MNIST, sebuah kumpulan data standar untuk tugas klasifikasi digit. Melalui kombinasi berbagai parameter, seperti ukuran kernel, jenis pooling, optimizer, dan jumlah epoch, dilakukan eksperimen untuk mengevaluasi dampak masing-masing parameter terhadap akurasi model. Pendekatan early stopping juga digunakan untuk mencegah overfitting dan meningkatkan efisiensi pelatihan.

a. Deskripsi Dataset

Dataset MNIST berisi gambar digit angka yang masing-masing memiliki resolusi 28x28 piksel dalam format grayscale. Dataset ini terdiri dari dua bagian utama, yaitu data pelatihan yang terdiri dari 60.000 gambar dan data pengujian yang terdiri dari 10.000 gambar. Transformasi data diterapkan untuk memastikan setiap gambar berada dalam format tensor yang sesuai dan dinormalisasi dengan rata-rata 0.1307 dan standar deviasi 0.3081. Transformasi ini membantu dalam proses pelatihan model agar konvergen lebih cepat dan stabil.

b. Model Arsitektur CNN

Arsitektur CNN yang digunakan dirancang untuk menyelesaikan tugas klasifikasi multi-kelas. Model ini terdiri dari dua lapisan convolusi, masing-masing dengan kernel berukuran variabel (3 atau 5) dan padding yang menyesuaikan ukuran kernel. Setelah setiap lapisan convolusi, diterapkan pooling yang dapat berupa MaxPooling atau AveragePooling, tergantung pada parameter yang diuji. Model ini juga memiliki dua lapisan fully connected, yaitu FC1 dengan 128 unit dan FC2 yang menghasilkan output 10 unit, sesuai dengan jumlah kelas pada dataset MNIST. Aktivasi ReLU digunakan pada setiap lapisan convolusi dan fully connected untuk memastikan non-linearitas. Output akhir dari model adalah log softmax, yang sesuai untuk tugas klasifikasi multi-kelas.

c. Optimizer dan Scheduler

Untuk mengoptimalkan parameter model, tiga metode optimasi digunakan, yaitu SGD dengan momentum, RMSProp, dan Adam. Setiap optimizer memiliki karakteristik unik yang memengaruhi kecepatan konvergensi dan stabilitas pelatihan. Selain itu, digunakan scheduler ReduceLROnPlateau untuk menyesuaikan learning rate berdasarkan perubahan loss. Scheduler ini membantu mengurangi learning rate ketika performa model stagnan, yang sering kali meningkatkan konvergensi.

d. Strategi Early Stopping

Strategi early stopping diterapkan untuk menghentikan proses pelatihan sebelum mencapai jumlah epoch maksimum jika akurasi pada data pengujian tidak meningkat dalam lima epoch berturut-turut. Pendekatan ini bertujuan untuk menghindari overfitting dan mengurangi waktu pelatihan yang tidak diperlukan.

e. Parameter yang Diuji

Eksperimen dilakukan dengan menguji berbagai parameter untuk memahami pengaruhnya terhadap performa model:

- Ukuran kernel: 3 dan 5.
- Jenis pooling: MaxPooling dan AveragePooling.
- Optimizer: SGD, RMSProp, dan Adam.
- Jumlah epoch: 5 dan 100, dengan early stopping diterapkan.

f. Hasil dan Analisis

Struktur pengujian dirancang untuk mengevaluasi performa model berdasarkan metrik-metrik seperti train loss, test loss, dan akurasi. Train loss yang menurun menunjukkan bahwa model berhasil belajar dari data, sedangkan test loss yang menurun dan akurasi yang meningkat mencerminkan kemampuan generalisasi model. Visualisasi grafik menunjukkan pola-pola ini dengan jelas, di mana pelatihan dihentikan lebih awal jika tidak ada peningkatan lebih lanjut pada akurasi. Observasi utama menunjukkan bahwa kernel berukuran 5 cenderung memberikan performa lebih baik, terutama ketika digunakan bersama MaxPooling. Optimizer Adam menunjukkan konvergensi tercepat dengan train loss yang rendah, sementara SGD dengan momentum lebih stabil namun memerlukan lebih banyak epoch. Pelatihan dengan 100 epoch memberikan hasil terbaik, tetapi early stopping sering kali menghentikan pelatihan sebelum mencapai batas maksimum epoch.

g. Kesimpulan

Berdasarkan hasil eksperimen, kombinasi parameter terbaik terdiri dari kernel berukuran 5, pooling MaxPooling, optimizer Adam, dan jumlah epoch hingga 100 dengan early stopping. Penggunaan scheduler dan strategi early stopping terbukti sangat efektif dalam menghindari overfitting dan mengurangi waktu pelatihan.