

Laporan Model Tensor Flow Dengan Dataset MNIST

Convolutional Neural Network (CNN) merupakan salah satu pendekatan yang paling efektif untuk menangani tugas-tugas klasifikasi citra, termasuk pengenalan angka tulisan tangan. Dalam analisis ini, framework TensorFlow digunakan untuk membangun, melatih, dan mengevaluasi model CNN pada dataset MNIST. TensorFlow, melalui modul tensorflow.keras, menyediakan beragam alat untuk merancang arsitektur model, mengatur proses pelatihan, dan memantau performa model secara efisien.

Dataset MNIST terdiri dari gambar grayscale berukuran 28x28 piksel yang mewakili angka 0 hingga 9. Setiap gambar dilabeli sesuai dengan angka yang direpresentasikannya. Dataset ini terbagi menjadi data pelatihan dan pengujian untuk memungkinkan evaluasi model yang komprehensif. Sebelum pelatihan, gambar-gambar ini diubah menjadi tensor menggunakan TensorFlow dan dinormalisasi agar berada dalam rentang [0, 1]. Dengan pendekatan ini, TensorFlow memastikan bahwa data siap untuk digunakan dalam arsitektur CNN yang dirancang.

Dataset dan Preprocessing

Dataset MNIST terdiri dari gambar grayscale dengan resolusi 28x28 piksel, yang masing-masing mewakili angka 0-9. Dataset terbagi menjadi:

- 60.000 data pelatihan dan 10.000 data pengujian.
- Data gambar dinormalisasi dengan membagi nilai pikselnya dengan 255 agar berada pada rentang [0, 1].
- Gambar diubah bentuknya menjadi tensor berukuran (28, 28, 1) untuk kompatibilitas dengan lapisan konvolusi.

Arsitektur Model

Model CNN terdiri dari komponen berikut:

- Lapisan Konvolusi: Tiga lapisan dengan 32, 64, dan 64 filter, masing-masing menggunakan fungsi aktivasi ReLU dan padding 'same'.
- Pooling: Dua jenis pooling dibandingkan, yaitu MaxPooling2D dan AveragePooling2D.
- Lapisan Dense: Satu lapisan fully connected dengan 64 unit dan satu lapisan output dengan 10 unit (softmax untuk klasifikasi).
- Optimizers: Optimizer yang dibandingkan adalah SGD, RMSprop, dan Adam.

Strategi Training

Model dilatih menggunakan callback:

- EarlyStopping: Untuk menghentikan pelatihan jika tidak ada perbaikan pada val_loss setelah 5 epoch.
- ReduceLROnPlateau: Untuk mengurangi learning rate secara adaptif saat val_loss stagnan selama 3 epoch.

Hiperparameter

- Ukuran Kernel: (3,3) dan (5,5).
- Tipe Pooling: MaxPooling2D dan AveragePooling2D.
- Optimizer: SGD, RMSprop, dan Adam.
- Jumlah Epoch: 5 dan 100.

Hasil

Setelah melakukan serangkaian eksperimen pada model CNN dengan berbagai kombinasi hiperparameter, beberapa pola penting dapat diidentifikasi. Berikut adalah hasil analisis mendalam berdasarkan setiap parameter utama

a. Akurasi Pelatihan dan Validasi

Dalam semua eksperimen, model mencapai akurasi pelatihan yang tinggi, mendekati atau bahkan melebihi 99% pada konfigurasi tertentu dengan 100 epoch. Namun, perbedaan yang lebih signifikan muncul saat membandingkan akurasi validasi. Model dengan konfigurasi tertentu menunjukkan performa yang konsisten pada data validasi, sementara beberapa konfigurasi mengalami kesenjangan antara akurasi pelatihan dan validasi, menandakan potensi overfitting.

Pada konfigurasi terbaik (kernel (5,5), MaxPooling2D, optimizer Adam, dan 100 epoch), akurasi validasi mencapai sekitar 99%, hampir setara dengan akurasi pelatihan. Ini menunjukkan bahwa model berhasil belajar secara efektif dari data pelatihan tanpa kehilangan generalisasi pada data pengujian.

b. Efek Ukuran Kernel

- Kernel (3,3): Ukuran kernel ini menghasilkan model yang lebih ringan dengan jumlah parameter yang lebih sedikit, memungkinkan pelatihan yang lebih cepat. Namun, model cenderung membutuhkan lebih banyak epoch untuk mencapai performa optimal dibandingkan kernel (5,5).
- Kernel (5,5): Ukuran kernel yang lebih besar memungkinkan model menangkap pola yang lebih luas dalam gambar, yang tampaknya meningkatkan akurasi validasi, terutama saat dikombinasikan dengan optimizer yang lebih adaptif seperti Adam. Dengan kernel ini, model mencapai akurasi validasi tertinggi secara konsisten.

Kesimpulannya, kernel (5,5) lebih unggul untuk tugas klasifikasi pada dataset MNIST, meskipun memerlukan waktu pelatihan yang sedikit lebih lama.

c. Perbandingan Tipe Pooling

- MaxPooling2D: Tipe pooling ini secara konsisten memberikan akurasi validasi yang lebih tinggi dibandingkan AveragePooling2D. MaxPooling2D lebih agresif dalam mengekstraksi fitur dominan, memungkinkan model untuk lebih fokus pada bagian penting dari gambar (seperti tepi dan sudut).
- AveragePooling2D: Pooling ini menghasilkan performa yang lebih stabil dalam beberapa konfigurasi, tetapi sering kalah dalam akurasi validasi dibandingkan MaxPooling2D. Hal ini mungkin disebabkan oleh pendekatannya yang lebih halus dalam merata-ratakan nilai, yang dapat mengaburkan fitur penting pada data MNIST.

Kesimpulannya, MaxPooling2D adalah pilihan yang lebih baik untuk dataset MNIST karena sifatnya yang lebih diskriminatif.

d. Performa Optimizer

- SGD: Optimizer ini menunjukkan hasil yang stabil, tetapi cenderung lebih lambat untuk mencapai performa optimal, terutama pada kombinasi kernel yang lebih besar seperti (5,5). Model sering memerlukan lebih dari 50 epoch untuk mendekati akurasi yang setara dengan optimizer lain.

- RMSprop: Optimizer ini memberikan performa yang baik dan sering konvergen lebih cepat dibandingkan SGD. Namun, dalam beberapa konfigurasi, akurasi validasi menunjukkan fluktuasi, menandakan bahwa RMSprop kurang stabil dalam menangani learning rate yang tinggi.
 - Adam: Optimizer ini memberikan performa terbaik dalam hampir semua konfigurasi. Adam berhasil memadukan kecepatan konvergensi dan stabilitas, menghasilkan akurasi validasi tertinggi tanpa overfitting. Pada kombinasi kernel (5,5) dan MaxPooling2D, Adam consistently mencapai akurasi di atas 99% dalam waktu pelatihan yang lebih singkat.
- e. Jumlah Epoch
- 5 Epoch: Dalam sebagian besar konfigurasi, model mencapai akurasi pelatihan dan validasi sekitar 97-98%. Namun, performa ini tidak cukup optimal untuk model dengan kernel (5,5) dan optimizer RMSprop atau Adam, yang memerlukan lebih banyak waktu untuk memanfaatkan kapasitas model.
 - 100 Epoch: Model mencapai performa maksimal, dengan akurasi validasi mendekati 99% pada konfigurasi optimal. Callback EarlyStopping sering menghentikan pelatihan sebelum mencapai 100 epoch, menunjukkan efektivitas callback dalam mencegah overfitting.

Penutup

Eksperimen ini berhasil menunjukkan pentingnya pemilihan konfigurasi yang tepat dalam membangun model Convolutional Neural Network (CNN) untuk tugas klasifikasi, seperti pada dataset MNIST. Melalui analisis berbagai kombinasi hiperparameter, ditemukan bahwa penggunaan kernel (5,5), MaxPooling2D, dan optimizer Adam memberikan hasil yang optimal, dengan akurasi validasi mendekati 99%. Selain itu, penerapan teknik callback seperti EarlyStopping juga berperan penting dalam meningkatkan efisiensi pelatihan dan mengurangi risiko overfitting.

Kesimpulannya, hasil eksperimen ini memberikan wawasan yang berguna dalam memilih kombinasi terbaik dari kernel, tipe pooling, dan optimizer, yang dapat diterapkan untuk berbagai tugas klasifikasi gambar serupa di masa depan. Dengan menerapkan hasil ini, diharapkan model CNN dapat mencapai performa terbaik dan lebih mudah diterapkan dalam aplikasi nyata yang membutuhkan pengenalan pola visual.