

Nama : Eko Putra Nugraha
NIM : 1103213212

Analisis Pengaruh Hyperparameter pada Performa Model CNN1

Kode ini melakukan eksperimen untuk melatih model Convolutional Neural Network (CNN) dengan berbagai kombinasi hyperparameter, yang bertujuan untuk memahami dampaknya terhadap performa model. Hyperparameter yang dieksplorasi meliputi ukuran kernel konvolusi, tipe pooling (max atau average), jumlah epoch pelatihan, dan jenis optimizer yang digunakan.

Ukuran Kernel Konvolusi: Eksperimen menguji ukuran kernel 3x3, 5x5, dan 7x7. Ukuran kernel yang lebih kecil (3x3) cenderung lebih baik dalam menangkap fitur-fitur detail pada gambar, sementara kernel yang lebih besar (misalnya 7x7) dapat menangkap pola yang lebih besar dan kompleks. Pemilihan ukuran kernel yang tepat bergantung pada karakteristik dataset dan kompleksitas fitur yang perlu dipelajari. Dari grafik yang dihasilkan, kita dapat melihat bagaimana ukuran kernel yang berbeda mempengaruhi konvergensi dan fluktuasi nilai loss.

Tipe Pooling: Dua jenis pooling diuji, yaitu Max Pooling dan Average Pooling. Max Pooling mengambil nilai maksimum dari setiap receptive field, yang membantu menyoroti fitur-fitur yang paling menonjol. Sementara itu, Average Pooling mengambil rata-rata nilai dalam setiap receptive field, yang dapat membantu menghaluskan fitur dan mengurangi noise. Pemilihan jenis pooling yang tepat juga bergantung pada karakteristik dataset. Dalam eksperimen ini, perbedaan antara kedua tipe pooling dapat diamati dalam grafik loss, dan beberapa konfigurasi mungkin lebih baik dalam satu pooling dibandingkan lainnya.

Jumlah Epoch: Kode ini menguji beberapa jumlah epoch (5, 50, 100, 250, dan 350). Epoch yang lebih sedikit mungkin tidak cukup untuk model mempelajari pola yang rumit, sehingga dapat mengakibatkan underfitting. Sementara itu, epoch yang terlalu banyak dapat menyebabkan overfitting, di mana model terlalu cocok dengan data training dan gagal dalam melakukan generalisasi pada data validasi. Grafik loss menunjukkan bagaimana train dan validation loss berubah seiring dengan bertambahnya jumlah epoch, dan dapat membantu dalam menentukan titik optimal untuk berhenti melakukan pelatihan model.

Jenis Optimizer: Tiga jenis optimizer diuji dalam kode ini, yaitu Stochastic Gradient Descent (SGD), RMSprop, dan Adam. SGD adalah optimizer paling dasar, sementara RMSprop dan Adam adalah varian yang lebih canggih yang menggunakan learning rate adaptif. Optimizer yang berbeda memiliki tingkat konvergensi yang berbeda dan mungkin lebih baik pada kondisi tertentu. Dari hasil grafik loss, kita dapat membandingkan bagaimana setiap optimizer mempengaruhi kecepatan konvergensi dan performa keseluruhan model.

Kesimpulan:

Eksperimen ini menunjukkan bahwa hyperparameter memiliki dampak signifikan pada performa model CNN. Tidak ada satu konfigurasi hyperparameter yang optimal untuk semua kondisi, dan pemilihan hyperparameter yang tepat bergantung pada dataset dan tugas yang dihadapi. Berdasarkan visualisasi hasil, kita dapat menganalisis dan memilih kombinasi hyperparameter yang menghasilkan performa terbaik berdasarkan nilai train loss dan validation loss. Selain itu, kita juga dapat memahami efek *early stopping* di mana training akan dihentikan jika validation loss tidak mengalami penurunan, sebagai salah satu strategi untuk mencegah terjadinya *overfitting*. Eksperimen ini merupakan langkah penting dalam proses pengembangan model machine learning.

dan memberikan pemahaman mendalam tentang bagaimana hyperparameter mempengaruhi kemampuan model dalam mempelajari dan menggeneralisasi data.

Analisis Pengaruh Hyperparameter pada Performa Model CNN dengan Dataset MNIST

Kode ini melakukan eksperimen mendalam untuk melatih model Convolutional Neural Network (CNN) pada dataset MNIST, yang terdiri dari gambar-gambar digit tulisan tangan. Tujuan eksperimen ini adalah untuk mengamati bagaimana berbagai kombinasi hyperparameter mempengaruhi performa model, dan juga untuk melakukan visualisasi terhadap aktivasi, gradien, dan bobot model. Hyperparameter yang dieksplorasi dalam kode ini meliputi ukuran kernel konvolusi, tipe pooling, jumlah epoch, dan jenis optimizer.

Ukuran Kernel Konvolusi: Eksperimen ini menguji dua ukuran kernel yang berbeda, yaitu 3x3 dan 5x5. Kernel 3x3 cenderung lebih baik dalam menangkap fitur-fitur lokal yang detail seperti tepi dan sudut pada digit tulisan tangan, sedangkan kernel 5x5 dapat menangkap pola yang lebih besar tetapi berisiko kehilangan detail halus. Dalam dataset MNIST, di mana digit sering kali terdiri dari garis dan kurva yang sederhana, kita mengamati bagaimana ukuran kernel mempengaruhi kemampuan model dalam mempelajari fitur-fitur penting. Analisis dari visualisasi loss, akurasi, serta gradien dan aktivasi akan memperjelas bagaimana ukuran kernel berkontribusi terhadap kemampuan model untuk belajar.

Tipe Pooling: Kode ini menguji dua jenis pooling, yaitu max pooling dan average pooling. Max pooling mengambil nilai maksimum dari setiap receptive field, yang membantu menyoroti fitur yang paling penting dan mengurangi noise, sementara average pooling mengambil rata-rata nilai yang bisa menghaluskan fitur dan mengurangi sensitivitas terhadap perubahan kecil dalam input. Penggunaan max atau average pooling dapat mempengaruhi bagaimana model menggeneralisasi representasi fitur. Dalam grafik loss dan akurasi, kita dapat melihat dampak masing-masing pooling terhadap kemampuan model dalam mengenali digit-digit MNIST. Selain itu, melalui visualisasi aktivasi lapisan pooling, kita dapat lebih memahami bagaimana setiap jenis pooling memodifikasi representasi fitur yang dipelajari.

Jumlah Epoch: Kode ini menguji jumlah epoch yang berbeda, yaitu 5, 50, dan 100. Jumlah epoch yang sedikit dapat menyebabkan underfitting, dimana model belum sepenuhnya mempelajari pola yang ada di dalam data. Sementara itu, terlalu banyak epoch dapat menyebabkan overfitting, dimana model menjadi terlalu spesifik terhadap data training dan gagal melakukan generalisasi dengan baik pada data testing. Oleh karena itu, pemilihan jumlah epoch yang tepat sangat penting untuk mendapatkan model yang optimal. Dari grafik loss dan akurasi, kita dapat mengamati bagaimana training dan testing loss dan akurasi berubah seiring dengan peningkatan epoch, dan juga bagaimana early stopping diimplementasikan untuk menghindari overfitting.

Jenis Optimizer: Dalam kode ini, hanya optimizer Adam yang diuji. Adam adalah algoritma optimasi yang adaptif dan sering kali menunjukkan kinerja yang baik dalam berbagai masalah, termasuk pengolahan citra. Berbeda dengan SGD yang menggunakan learning rate yang tetap, Adam secara otomatis menyesuaikan learning rate pada setiap parameter, yang dapat mempercepat proses pelatihan dan memberikan konvergensi yang lebih baik. Dalam konteks eksperimen ini, kita mengevaluasi bagaimana optimizer Adam dapat menghasilkan model yang optimal untuk task klasifikasi digit MNIST.

Visualisasi Aktivasi, Gradien, dan Bobot: Kode ini juga melakukan visualisasi yang komprehensif untuk memahami bagaimana model bekerja. Visualisasi distribusi aktivasi pada berbagai lapisan ReLU membantu kita melihat bagaimana output dari lapisan-lapisan tersebut tersebar dan bagaimana informasi diproses. Plot aliran gradien memberikan insight mengenai bagaimana gradien mengalir melalui jaringan, yang penting untuk memastikan pelatihan berjalan dengan baik. Visualisasi distribusi bobot dari lapisan konvolusi dan *fully connected* menunjukkan bagaimana model mempelajari fitur-fitur yang penting. Selain itu, saliency map digunakan untuk mengetahui bagian mana dari gambar yang paling mempengaruhi keputusan model, memberikan pemahaman lebih lanjut mengenai bagaimana model mengenali digit.

Kesimpulan:

Eksperimen ini memberikan pemahaman mendalam tentang bagaimana berbagai hyperparameter mempengaruhi performa model CNN pada dataset MNIST. Dengan membandingkan loss, akurasi, dan visualisasi internal model, kita dapat menentukan kombinasi hyperparameter yang menghasilkan kinerja terbaik. Analisis ini tidak hanya berfokus pada performa model (loss dan akurasi), tetapi juga pada mekanisme internal model melalui visualisasi aktivasi, gradien, dan bobot. Dengan pemahaman yang komprehensif ini, kita dapat lebih baik merancang dan melatih model CNN untuk berbagai aplikasi pengolahan citra lainnya.