

Analisa

1. Arsitektur CNN dengan X lapisan konvolusi menghasilkan akurasi training 98% tetapi akurasi validasi 62%. Jelaskan fenomena vanishing gradient yang mungkin terjadi pada lapisan awal, dan bagaimana cara memitigasinya! Mengapa penambahan Batch Normalization setelah lapisan konvolusi ke-Y justru memperburuk generalisasi, serta strategi alternatif untuk menstabilkan pembelajaran?

Vanishing Gradient:

Vanishing gradient terjadi ketika gradien yang dihitung selama proses backpropagation menjadi sangat kecil, terutama pada lapisan-lapisan awal jaringan. Hal ini menghambat pembaruan bobot pada lapisan pertama, sehingga pelatihan menjadi sangat lambat atau bahkan terhenti.

Cara Mengatasi:

- Fungsi Aktivasi Non-Saturasi: Menggunakan fungsi aktivasi seperti ReLU atau Leaky ReLU yang tidak mengalami masalah saturasi pada nilai positif, sehingga gradien bisa mengalir dengan baik pada lapisan-lapisan awal.
- Inisialisasi Bobot yang Tepat: Penggunaan inisialisasi bobot yang baik, seperti He initialization, dapat mengurangi masalah vanishing gradient, karena ini membantu menjaga distribusi gradien yang tepat selama pelatihan.

Pengaruh Batch Normalization:

Batch Normalization (BN) dapat membantu normalisasi hasil lapisan sebelumnya, tetapi kadang malah merugikan jika tidak diterapkan dengan tepat pada lapisan tertentu, terutama lapisan awal. Penambahan BN pada lapisan konvolusi tertentu mungkin menyebabkan gangguan pada proses pembelajaran dan justru memperburuk hasil yang didapat.

Alternatif untuk Menstabilkan Pembelajaran:

- Learning Rate Scheduler: Menggunakan pengatur learning rate secara bertahap dapat membantu model menghindari ketidakstabilan dan overfitting.
- Dropout: Menambahkan dropout pada lapisan fully connected bisa mencegah overfitting, menjaga model tetap stabil dan lebih efektif.

2. Ketika melatih CNN dari nol, loss training stagnan di nilai tinggi setelah XXX(3 digit epoch) epoch. Identifikasi tiga penyebab potensial terkait laju pembelajaran (learning rate), inisialisasi berat, atau kompleksitas model! Mengapa penggunaan Cyclic Learning Rate dapat membantu model keluar dari local minima, dan bagaimana momentum pada optimizer SGD memengaruhi konvergensi?

Masalah Stagnasi Loss pada Epoch yang Tinggi

Penyebab Utama:

- **Learning Rate Terlalu Tinggi:** Jika learning rate terlalu tinggi, model mungkin melewati titik minimum global dan kesulitan untuk menurunkan loss lebih lanjut.
- **Inisialisasi Bobot yang Tidak Optimal:** Jika bobot model tidak diinisialisasi dengan benar, model mungkin terjebak dalam minima lokal atau gagal untuk belajar dengan baik.
- **Model Terlalu Kompleks:** Model yang memiliki banyak parameter tanpa cukup data untuk dilatih bisa mengalami kesulitan dalam proses pelatihan, menyebabkan stagnasi pada loss.

Cyclic Learning Rate (CLR):

Dengan Cyclic Learning Rate, learning rate secara bergantian naik dan turun selama pelatihan. Teknik ini membantu model keluar dari minima lokal dan mencari solusi yang lebih baik di ruang parameter.

Momentum pada SGD:

Momentum membantu model bergerak lebih cepat menuju solusi optimal, mengurangi fluktuasi atau osilasi dalam proses pelatihan, dan memastikan konvergensi yang lebih stabil dan cepat.

3. Pada klasifikasi spesies ikan menggunakan CNN, penggunaan fungsi aktivasi ReLU tidak menunjukkan peningkatan akurasi setelah 50 epoch, meskipun learning rate telah dioptimasi. Jelaskan fenomena dying ReLU yang mungkin terjadi dan bagaimana hal ini mengganggu aliran gradien selama backpropagation!

Fenomena Dying ReLU:

Dying ReLU adalah masalah ketika banyak neuron dengan fungsi aktivasi ReLU tidak pernah teraktifkan, karena nilai masukan ke neuron tersebut terlalu negatif. Hal ini menyebabkan keluaran ReLU tetap 0 sepanjang waktu, menghambat aliran gradien selama backpropagation, yang pada akhirnya mengganggu pelatihan.

Cara Mengatasi Dying ReLU:

- **Leaky ReLU:** Menggunakan Leaky ReLU memungkinkan gradien tetap mengalir meskipun nilai input ke neuron negatif, mengurangi masalah ini.
- **Inisialisasi Bobot yang Baik:** Menggunakan He Initialization yang memang didesain untuk mengatasi masalah pada ReLU dapat mengurangi kemungkinan terjadinya Dying ReLU.

4. Pada pelatihan CNN untuk klasifikasi XX spesies ikan, grafik AUC-ROC menunjukkan satu kelas (Spesies X) stagnan di 0.55 sementara kelas lain mencapai >0.85 setelah YYY epoch. Penyebab Gagalnya Class-Weighted Loss: Penggunaan class-weighted loss tidak selalu efektif jika ada faktor lain yang menghambat model dalam mendeteksi kelas minoritas. Misalnya, jika distribusi kelas sangat tidak seimbang atau model tidak cukup kuat untuk mempelajari pola kelas minoritas, class-weighted loss mungkin tidak cukup membantu.

Penyebab Gagalnya Class-Weighted Loss:

Penggunaan class-weighted loss tidak selalu efektif jika ada faktor lain yang menghambat model dalam mendeteksi kelas minoritas. Misalnya, jika distribusi kelas sangat tidak seimbang atau model tidak cukup kuat untuk mempelajari pola kelas minoritas, class-weighted loss mungkin tidak cukup membantu.

Faktor Penyebab:

- Distribusi Data yang Tidak Seimbang: Jika jumlah kelas minoritas sangat kecil, meskipun menggunakan penyesuaian bobot, model tetap kesulitan mendeteksi kelas tersebut.
 - Model Tidak Cukup Kompleks: Model mungkin tidak cukup kompleks untuk mempelajari pola dalam data kelas minoritas.
 - Learning Rate yang Tidak Tepat: Jika learning rate terlalu tinggi atau rendah, model bisa terlalu cepat berfokus pada kelas mayoritas dan tidak cukup memperhatikan kelas minoritas
5. Pada arsitektur CNN untuk klasifikasi ikan, peningkatan kompleksitas model justru menyebabkan penurunan akurasi validasi dari 85% ke 65%, meskipun akurasi training mencapai 98%. Jelaskan fenomena overfitting yang terjadi, dan mengapa penambahan kapasitas model tidak selalu meningkatkan generalisasi! Identifikasi 3 kesalahan desain arsitektur yang memicu degradasi performa

Fenomena Overfitting:

Overfitting terjadi ketika model terlalu mempelajari detail atau noise dari data pelatihan, yang mengakibatkan penurunan kinerja pada data validasi. Peningkatan kompleksitas model dengan menambah jumlah lapisan atau neuron dapat menyebabkan model beradaptasi terlalu baik pada data pelatihan, tetapi gagal menggeneralisasi ke data yang belum pernah dilihat.

Mengapa Peningkatan Kapasitas Tidak Selalu Meningkatkan Generalisasi:

- Overfitting: Menambah kapasitas model tidak selalu meningkatkan kemampuan model untuk menggeneralisasi. Model yang terlalu kompleks bisa terjebak dalam pola spesifik dari data pelatihan.
- Kurangnya Data: Data pelatihan yang terbatas dengan model yang kompleks cenderung memicu overfitting, karena model belajar pola yang tidak relevan.

- Tidak Ada Regularisasi: Tanpa teknik regularisasi seperti dropout atau L2 regularization, model yang kompleks akan cenderung belajar lebih dari sekadar pola yang ada, termasuk noise dalam data.

Kesalahan Desain Arsitektur:

- Terlalu Banyak Parameter: Menambah jumlah lapisan atau neuron tanpa pengaturan yang baik hanya meningkatkan kompleksitas dan kemungkinan overfitting.
- Kurangnya Dropout: Tanpa dropout, model lebih rentan terhadap overfitting, terutama pada lapisan fully connected.
- Inisialisasi Bobot yang Buruk: Jika bobot tidak diinisialisasi dengan baik, model akan kesulitan belajar meskipun kapasitas model besar.