

1. Fenomena Vanishing Gradient dan Penggunaan Batch Normalization

Ketika arsitektur CNN dengan banyak lapisan konvolusi mencapai akurasi training 98% tetapi akurasi validasi hanya 62%, ini menunjukkan masalah overfitting yang serius.

Salah satu faktor yang berkontribusi adalah vanishing gradient.

Fenomena Vanishing Gradient pada Lapisan Awal:

Vanishing gradient terjadi ketika gradien yang digunakan untuk memperbarui bobot selama backpropagation menjadi sangat kecil saat mencapai lapisan-lapisan awal.

Akibatnya:

- Lapisan awal belajar sangat lambat atau bahkan tidak belajar sama sekali
- Model gagal mengekstrak fitur penting di lapisan awal
- Representasi fitur dasar seperti tepi dan tekstur pada ikan tidak optimal

Cara Mitigasi Vanishing Gradient:

1. Residual Connections (Skip Connections): Memungkinkan gradien mengalir langsung dari lapisan output ke lapisan input tanpa melewati semua lapisan, seperti pada arsitektur ResNet
2. Inisialisasi Bobot yang Tepat: Menggunakan metode seperti He initialization untuk fungsi aktivasi ReLU
3. Fungsi Aktivasi Alternatif: Menggunakan Leaky ReLU, PReLU, atau ELU yang memiliki gradien tidak nol untuk input negatif
4. Normalisasi Gradien: Teknik seperti gradient clipping untuk mencegah gradien terlalu kecil atau terlalu besar

Mengapa Batch Normalization Dapat Memperburuk Generalisasi:

Batch Normalization pada lapisan ke-Y justru dapat memperburuk generalisasi karena:

- Ketergantungan pada Statistik Batch: Jika batch size kecil atau data tidak distribusi merata, statistik batch menjadi tidak stabil
- Covariate Shift Internal: BN mungkin menyebabkan perubahan distribusi fitur yang terlalu agresif selama pelatihan
- Overconfidence: BN dapat membuat model terlalu yakin dengan prediksinya, mengurangi regularisasi alami

Strategi Alternatif untuk Menstabilkan Pembelajaran:

1. Layer Normalization: Normalisasi berdasarkan neuron dalam layer yang sama, tidak bergantung pada batch size
2. Group Normalization: Mengelompokkan channel dan menormalisasi dalam grup, lebih stabil untuk batch size kecil
3. Weight Standardization: Normalisasi bobot kernel konvolusi secara langsung
4. Dropout Spasial: Mematikan seluruh feature map alih-alih neuron individual

2. Stagnasi Loss Training

Ketika loss training stagnan di nilai tinggi setelah ratusan epoch, ini mengindikasikan masalah yang menghambat pembelajaran model.

Tiga Penyebab Potensial:

1. Masalah Laju Pembelajaran (Learning Rate):

- Learning Rate Terlalu Tinggi: Menyebabkan osilasi dan tidak dapat konvergen
- Learning Rate Terlalu Rendah: Terjebak pada plateau dan progress sangat lambat
- Decay Schedule Tidak Tepat: Penurunan learning rate yang tidak sesuai dengan dinamika pelatihan

2. Inisialisasi Bobot:

- Vanishing/Exploding Signal: Inisialisasi yang buruk menyebabkan sinyal menjadi sangat kecil atau besar saat forward/backward pass
- Distribusi Tidak Sesuai: Inisialisasi yang tidak sesuai dengan arsitektur dan fungsi aktivasi
- Simetri Terperangkap: Neuron dengan inisialisasi identik menyebabkan update yang identik (symmetric weight problem)

3. Kompleksitas Model:

- Kapasitas Tidak Mencukupi: Model terlalu sederhana untuk mempelajari pola kompleks pada dataset ikan
- Bottleneck Tidak Tepat: Dimensi feature map terlalu kecil di tengah arsitektur
- Ketidaksesuaian Arsitektur: Pemilihan layer yang tidak sesuai dengan karakteristik data ikan

Cyclic Learning Rate dan Local Minima:

Cyclic Learning Rate (CLR) dapat membantu model keluar dari local minima karena:

- Eksplorasi Lanskap Loss: Variasi periodik dalam learning rate memungkinkan model menjelajahi berbagai area di lanskap loss
- Escape Mechanism: Peningkatan sesaat pada learning rate membantu model "melompati" local minima
- Ensemble Effect: CLR secara implisit menghasilkan efek ensemble dari berbagai titik di lanskap optimasi

Pengaruh Momentum pada Optimizer SGD:

- Kecepatan Konvergensi: Momentum mempercepat konvergensi dengan mengakumulasi gradien dari iterasi sebelumnya
- Mengatasi Noise: Meredam fluktuasi gradien dan menghaluskan trajectory optimasi
- Escape from Saddle Points: Membantu model melewati saddle points yang merupakan tantangan dalam deep learning
- Adaptasi Gradient Direction: Memberikan "inersia" pada arah update, membantu dalam area lanskap loss yang curam

3. Fenomena Dying ReLU

Ketika penggunaan fungsi aktivasi ReLU tidak menunjukkan peningkatan akurasi setelah 50 epoch pada klasifikasi spesies ikan, ini bisa mengindikasikan masalah dying ReLU.

Fenomena Dying ReLU:

Dying ReLU terjadi ketika neuron dengan aktivasi ReLU terus-menerus menghasilkan output nol untuk setiap input, sehingga:

- Neuron tersebut menjadi "mati" dan tidak aktif lagi
- Tidak ada gradien yang mengalir melalui neuron ini selama backpropagation (karena turunan ReLU untuk $x \leq 0$ adalah 0)
- Kapasitas model berkurang karena sebagian neuron tidak berkontribusi

Bagaimana Dying ReLU Mengganggu Aliran Gradien:

1. Hilangnya Jalur Pembelajaran: Ketika neuron "mati", tidak ada gradien yang mengalir melaluinya, sehingga bobot tidak diperbarui
2. Perambatan Masalah: Neuron "mati" dapat menyebabkan neuron di lapisan sebelumnya juga mendapatkan gradien yang lebih kecil
3. Pengurangan Kapasitas Efektif: Model kehilangan kapasitas representasionalnya secara progresif
4. Ketergantungan pada Input: Fitur ikan tertentu mungkin hanya mengaktifkan sedikit neuron, menyebabkan banyak bagian model tidak terlatih

Solusi untuk Mengatasi Dying ReLU:

1. Fungsi Aktivasi Alternatif: Menggunakan Leaky ReLU ($\alpha=0.01$), PReLU (parametric α), atau ELU
2. Inisialisasi Bobot yang Tepat: Menggunakan He initialization yang dirancang untuk ReLU
3. Learning Rate yang Lebih Kecil: Mengurangi kemungkinan bobot bergerak terlalu jauh ke area negatif
4. Batch Normalization: Menjaga distribusi input ke ReLU agar tidak terlalu condong ke arah negative

4. Masalah Kinerja AUC-ROC pada Spesies Tertentu

Dalam klasifikasi spesies ikan menggunakan CNN, ketika grafik AUC-ROC menunjukkan satu kelas (Spesies X) stagnan di 0.55 sementara kelas lain mencapai >0.85 , ini mengindikasikan tantangan spesifik pada kelas tersebut.

Mengapa Class-Weighted Loss Function Gagal Meningkatkan Kinerja:

Class-weighted loss function gagal karena:

- Ketidakseimbangan Bukan Satu-satunya Masalah: Meskipun class weighting membantu ketidakseimbangan jumlah sampel, ini tidak mengatasi masalah representasi dan kompleksitas fitur
- Penekanan yang Berlebihan: Bobot yang terlalu tinggi dapat menyebabkan overfitting pada sampel kelas minoritas
- Trade-off Performa: Peningkatan performa pada satu kelas dapat mengorbankan performa kelas lain

Tiga Faktor Penyebab Potensial:

1. Karakteristik Data:

- Variabilitas Intra-kelas Tinggi: Spesies X mungkin memiliki variasi penampilan yang sangat besar (warna, ukuran, postur)
- Fitur Pembeda yang Samar: Fitur yang membedakan Spesies X dari spesies lain sangat halus atau tidak konsisten
- Kualitas Gambar Tidak Konsisten: Gambar Spesies X mungkin memiliki pencahayaan, sudut, atau resolusi yang bervariasi

2. Arsitektur Model:

- Ketidacukupan Representasi: Arsitektur tidak cukup ekspresif untuk menangkap fitur diskriminatif Spesies X
- Field of View Tidak Tepat: Ukuran reseptif field pada lapisan konvolusi tidak optimal untuk fitur Spesies X
- Ketidakesesuaian Skala Fitur: Skala fitur penting Spesies X mungkin hilang dalam pooling atau stride

3. Permasalahan Pembelajaran:

- Confounding Features: Model mungkin fokus pada fitur yang tidak relevan yang kebetulan berkorelasi dengan Spesies X
- Gradient Starvation: Kelas lain mendominasi update gradien, menyebabkan pembelajaran yang tidak seimbang
- Representasi Bersaing: Representasi yang baik untuk kelas lain mungkin membuat representasi Spesies X sulit dioptimalkan

Solusi yang Dapat Diterapkan:

1. Focal Loss: Mengurangi bobot sampel yang mudah diklasifikasi, fokus pada sampel sulit
2. Hierarchical Classification: Memisahkan klasifikasi untuk kelompok spesies yang membingungkan
3. Feature Augmentation: Augmentasi spesifik untuk Spesies X yang meningkatkan variabilitas pembelajaran
4. Arsitektur Attention: Menambahkan mekanisme attention untuk fokus pada fitur diskriminatif

5. Overfitting pada Model Kompleks

Ketika peningkatan kompleksitas model CNN menyebabkan penurunan akurasi validasi dari 85% ke 65%, meskipun akurasi training mencapai 98%, ini adalah kasus klasik overfitting.

Fenomena Overfitting:

Overfitting terjadi ketika model mempelajari noise dan detail spesifik pada data training alih-alih pola umum yang dapat digeneralisasi, sehingga:

- Model memiliki performa sangat baik pada data training
- Model memiliki performa buruk pada data yang belum pernah dilihat (validasi/test)
- Model "menghafal" data training daripada mempelajari pola umum

Mengapa Penambahan Kapasitas Model Tidak Selalu Meningkatkan Generalisasi:

- Bias-Variance Trade-off: Model kompleks memiliki variance tinggi, lebih rentan terhadap noise dalam data
- Data yang Terbatas: Ketika data training terbatas, model kompleks cenderung overfitting
- Curse of Dimensionality: Ruang fitur yang lebih tinggi membutuhkan data eksponensial lebih banyak
- Parameter Efficiency: Model yang lebih kecil dengan parameter efisien dapat mencapai generalisasi lebih baik

Tiga Kesalahan Desain Arsitektur yang Memicu Degradasi Performa:

1. Penggunaan Kapasitas yang Tidak Tepat:

- Over-parameterization: Jumlah parameter yang terlalu banyak dibandingkan dengan jumlah data training
- Layer Terlalu Dalam: Menambahkan lapisan yang tidak diperlukan tanpa mekanisme regularisasi yang memadai
- Filter Size yang Berlebihan: Ukuran filter konvolusi yang terlalu besar menangkap noise alih-alih pola umum

2. Kekurangan Mekanisme Regularisasi:

- Kurangnya Dropout: Tidak ada atau terlalu sedikit dropout di lapisan fully connected
- Regularisasi L1/L2 yang Tidak Memadai: Weight decay yang tidak cukup untuk mengontrol kompleksitas model
- Data Augmentation Terbatas: Kurangnya variasi dalam data augmentation untuk memperkaya data training

3. Masalah Arsitektur dan Aliran Informasi:

- Bottleneck yang Tidak Tepat: Pengurangan dimensi yang terlalu agresif menyebabkan hilangnya informasi penting
- Skip Connection yang Tidak Efektif: Residual/skip connection yang tidak optimal menyebabkan degradasi informasi
- Pooling yang Berlebihan: Terlalu banyak operasi pooling menyebabkan hilangnya detail spasial penting

Solusi untuk Arsitektur yang Lebih Baik:

1. Arsitektur Sesuai Dataset: Menyelaraskan kompleksitas model dengan ukuran dan kerumitan dataset
2. Progressive Regularization: Meningkatkan regularisasi seiring dengan peningkatan kapasitas model
3. Transfer Learning + Fine-tuning: Menggunakan model pre-trained dengan fine-tuning terbatas
4. Architecture Search: Eksplorasi sistematis untuk menemukan arsitektur optimal untuk data spesifik ikan