

## CNN

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 terjadi ketika gradien yang dihitung selama backpropagation menjadi sangat kecil, sehingga parameter di lapisan awal hampir tidak diperbarui.

Penyebab:

- Lapisan awal menerima gradien yang sangat kecil karena efek berantai dari turunan fungsi aktivasi dan pembobotan.
- Gradien  $\rightarrow$  mendekati nol seiring bertambahnya kedalaman jaringan  $\rightarrow$  lambatnya pembelajaran lapisan awal.

Mitigasi:

- Gunakan Batch Normalization (BN) untuk menstabilkan distribusi input ke setiap lapisan.
- Gunakan fungsi aktivasi ReLU atau variannya (Leaky ReLU, ELU) yang tidak menyaturasi seperti sigmoid/tanh.
- Gunakan residual connections (ResNet) agar gradien dapat "melompati" lapisan.

Mengapa BN Justru Memperburuk Generalisasi di Lapisan ke-Y?

- BN menyisipkan statistik mini-batch (mean & std), yang bisa menyebabkan data leakage atau ketidakstabilan distribusi jika batch terlalu kecil.
- Setelah lapisan tertentu (ke-Y), input mungkin telah cukup stabil; menambahkan BN bisa membuat model terlalu tergantung pada data mini-batch (overfitting terhadap noise).
- BN mengurangi regularisasi jika digunakan berlebihan.

Alternatif Strategi Stabilisasi:

- Dropout untuk regularisasi.
- Weight decay (L2 regularization).
- Layer Normalization jika BN tidak cocok karena batch size kecil.
- Early Stopping untuk menghindari overtraining.

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?

Tiga Penyebab Potensial:

1. Learning rate terlalu kecil → langkah pembaruan parameter sangat lambat → stagnasi.
2. Inisialisasi bobot buruk → bisa masuk ke simetri atau saddle point → sulit keluar dari zona loss tinggi.
3. Model terlalu kompleks → overparameterized dan cenderung ke saddle point/local minima → susah beranjak tanpa gangguan besar.

Cyclic Learning Rate (CLR) Membantu?

- CLR mengayunkan learning rate dari rendah ke tinggi secara periodik.
- Saat learning rate tinggi, model bisa keluar dari local minima/saddle point dan mencari solusi yang lebih baik.
- Mencegah stagnasi karena selalu ada variasi dalam eksplorasi ruang parameter.

Momentum pada SGD:

- Momentum menambahkan akumulasi gradien masa lalu → mempercepat konvergensi ke arah yang konsisten.
- Membantu melompati lembah kecil (shallow minima) dan mengurangi osilasi → pembelajaran 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:

- Terjadi ketika neuron ReLU hanya menghasilkan output 0 terus-menerus.
- Karena  $\text{ReLU} = \max(0, x)$ , saat bobot awal atau gradien memaksa banyak output  $< 0$ , neuron tidak aktif.
- Gradien untuk neuron itu = 0 → tidak belajar selamanya → bagian jaringan “mati”.

Dampak:

- Gradien tidak mengalir ke lapisan sebelumnya.
- Menghambat pembelajaran fitur.
- Akurasi stagnan karena kapasitas jaringan efektif menurun.

Solusi:

- Gunakan Leaky ReLU, Parametric ReLU, atau ELU → mereka tetap memberi gradien meski untuk input negatif.
- Perhatikan inisialisasi bobot (mis. He initialization untuk ReLU).
- Batch Normalization bisa membantu menjaga aktivasi tetap di zona hidup.

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. Analisis mengapa class-weighted loss function gagal meningkatkan kinerja Spesies X, dan identifikasi tiga faktor penyebab potensial terkait karakteristik data dan arsitektur model!

Mengapa Class-Weighted Loss Gagal?

- Class-weight hanya memberi penalti lebih besar untuk kesalahan pada Spesies X, tetapi tidak membantu jika data tersebut ambigu, noisy, atau tidak representatif.

Tiga Penyebab Potensial:

1. Jumlah data Spesies X sangat sedikit → model kesulitan belajar distribusi kelas tersebut.
2. Fitur visual Spesies X mirip dengan kelas lain → model bingung, class-weight tidak cukup untuk membedakan.
3. Augmentasi data tidak menambah variasi representatif untuk Spesies X → distribusi tetap sempit.

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:

- Model belajar terlalu spesifik pada data training (noise, outlier, detail kecil).
- Tidak mampu menggeneralisasi ke data validasi/test.
- Training acc tinggi, validasi acc turun drastis.

Mengapa Menambah Kapasitas Tidak Selalu Baik?

- Terlalu banyak parameter → model hafal data training.
- Data tidak cukup banyak/beragam untuk melatih model besar → underutilized capacity.
- Membuat optimisasi lebih sulit dan lambat.

Tiga Kesalahan Desain yang Memicu Overfitting:

- Terlalu banyak filter/lapisan tanpa regularisasi (dropout, batch norm).
- Kernel size besar tanpa reduksi dimensi (pooling) → eksplosif pertumbuhan parameter.

- Fully connected layer besar di akhir → overparameterisasi pada bagian output.