

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 sangat kecil diteruskan kembali melalui banyak lapisan, menyebabkan pembelajaran lambat atau bahkan berhenti di lapisan awal. Hal ini sering terjadi pada arsitektur CNN yang dalam tanpa inisialisasi bobot atau fungsi aktivasi yang tepat.

Batch Normalization setelah lapisan tertentu bisa memperburuk generalisasi karena mengganggu distribusi representasi fitur yang telah stabil, terutama bila batch size kecil. Strategi alternatif termasuk Residual Connection (seperti pada ResNet), pemilihan fungsi aktivasi seperti ReLU/LeakyReLU, dan penggunaan pre-trained model sebagai feature extractor.

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?

Penyebab potensial:

- Learning rate terlalu kecil sehingga model tidak cukup belajar di setiap langkah.
- Inisialisasi bobot yang buruk menyebabkan propagasi gradien tidak efektif.
- Model terlalu kompleks menyebabkan kesulitan optimasi awal.

Cyclic Learning Rate memungkinkan model mengeksplorasi ruang parameter lebih luas, mendorong keluar dari local minima atau saddle points. Momentum dalam SGD membantu mempertahankan arah gradien yang stabil, mempercepat konvergensi dan menghindari osilasi tajam.

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!

Dying ReLU terjadi ketika neuron terus-menerus menghasilkan nol ( $\text{output} \leq 0$ ), membuat gradiennya nol juga. Neuron yang mengalami hal ini tidak lagi berkontribusi dalam pembelajaran karena gradien tidak diperbarui. Solusinya meliputi: mengganti ReLU dengan LeakyReLU atau ELU, inisialisasi bobot yang lebih baik, atau normalisasi batch input.

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!

Class-weighted loss bisa gagal jika:

- Representasi visual Spesies X terlalu mirip dengan kelas lain.
- Ukuran dataset Spesies X terlalu kecil atau terlalu bervariasi.
- Model arsitektur tidak cukup kompleks atau belum mampu menangkap fitur spesifik kelas tersebut.

Solusi bisa berupa augmentasi data khusus untuk Spesies X, menambahkan loss auxiliary, atau menggunakan teknik focal loss.

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.

Overfitting terjadi ketika model terlalu menyesuaikan diri dengan data latih, termasuk noise atau outlier, sehingga performa pada data validasi menurun. Penambahan kapasitas seperti layer berlebihan, neuron terlalu banyak, atau tidak adanya regularisasi menyebabkan model terlalu kompleks.

Tiga kesalahan desain umum:

- Tidak menggunakan Dropout atau Batch Normalization.
- Layer terlalu banyak tanpa penyesuaian data atau batch size.
- Tidak melakukan early stopping atau data augmentation untuk mencegah overfitting.