

Analisa – Klasifikasi Ikan

- 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?**

**Jawaban:**

Fenomena ini menunjukkan gejala overfitting yang ekstrem, di mana model mampu menghafal data training tetapi gagal memprediksi data baru. Salah satu faktor yang bisa terjadi adalah vanishing gradient, terutama pada lapisan awal jaringan. Vanishing gradient terjadi ketika gradien (turunan) yang dihitung selama backpropagation menjadi sangat kecil saat melewati banyak lapisan, sehingga lapisan awal hampir tidak mengalami pembaruan bobot. Hal ini umum terjadi jika arsitektur terlalu dalam tanpa mitigasi.

Untuk memitigasi vanishing gradient, beberapa strategi yang efektif adalah:

- Menggunakan fungsi aktivasi seperti ReLU atau varian lainnya yang tidak “menekan” gradien secara ekstrem.
- Mengimplementasikan skip connections (seperti pada ResNet) yang membantu gradien mengalir lebih mudah ke lapisan awal.
- Memakai inisialisasi bobot yang baik (misalnya He initialization untuk ReLU).

Penambahan Batch Normalization setelah lapisan konvolusi ke-Y kadang malah memperburuk generalisasi karena meskipun BatchNorm membantu menstabilkan distribusi aktivasi (yang mempercepat konvergensi), ia juga bisa menyebabkan over-parameterization jika digunakan secara berlebihan, sehingga model semakin kompleks dan rentan overfitting.

Strategi alternatif untuk menstabilkan pembelajaran adalah:

- Menggunakan dropout untuk regularisasi.
- Memasukkan data augmentation agar model belajar lebih variatif.
- Memilih arsitektur yang lebih sederhana atau menggunakan teknik early stopping untuk mencegah overfitting lebih awal.

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?

**Jawaban:**

Ada tiga penyebab potensial:

1. Laju Pembelajaran (Learning Rate) Tidak Tepat:  
Jika learning rate terlalu kecil, pembaruan bobot jadi lambat dan model sulit keluar dari plateau. Jika terlalu besar, model bisa melompat-lompat dan gagal menemukan minimum.
2. Inisialisasi Berat (Weight Initialization) Buruk:  
Jika bobot awal terlalu kecil atau besar, aktivasi bisa menjadi tidak stabil. Hal ini memperburuk masalah vanishing/exploding gradient, terutama pada model dalam.
3. Kompleksitas Model Tidak Sesuai:  
Model yang terlalu sederhana mungkin tidak mampu mempelajari pola yang kompleks, sementara model terlalu kompleks bisa terperangkap di local minima atau saddle points.

Cyclic Learning Rate (CLR) memungkinkan learning rate naik turun selama pelatihan, yang memudahkan model keluar dari local minima atau saddle points yang dangkal. Dengan naik turunnya learning rate, model punya kesempatan untuk mengeksplorasi berbagai solusi yang lebih optimal.

Momentum membantu mempercepat konvergensi dengan “menyimpan” arah gradien sebelumnya. Ini mencegah model berhenti pada valley kecil dan membantu menjaga laju pembelajaran tetap stabil dalam arah yang konsisten, mempercepat pelatihan dan menghindari osilasi.

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!

**Jawaban:**

Ini kemungkinan besar disebabkan oleh fenomena dying ReLU, di mana unit ReLU menjadi “mati” dan selalu output 0 karena nilai inputnya negatif secara permanen. Hal ini menghambat aliran gradien selama backpropagation karena turunan ReLU untuk input negatif adalah 0. Akibatnya, bobot yang terhubung ke neuron mati tidak akan pernah diperbarui, sehingga sebagian model “lumpuh”.

Solusi yang dapat diterapkan antara lain:

- Mengganti ReLU dengan aktivasi lain seperti LeakyReLU, ELU, atau PReLU yang tetap memberikan gradien kecil meskipun input negatif.
- Memeriksa inisialisasi bobot agar tidak menyebabkan terlalu banyak neuron menjadi mati pada awal pelatihan.
- Menggunakan Batch Normalization sebelum ReLU untuk menjaga distribusi input tetap stabil.

**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!**

**Jawaban:**

Ini mengindikasikan bahwa model kesulitan membedakan Spesies X dari kelas lain. Meskipun sudah menggunakan *class-weighted loss*, performa tetap stagnan karena beberapa faktor:

- Ketidakseimbangan Data Ekstrem:  
Jika jumlah sampel Spesies X terlalu sedikit atau memiliki variasi sangat kecil, model tetap kesulitan meskipun bobot loss diperbesar.
- Fitur yang Kurang Informatif:  
Ciri visual Spesies X mungkin sangat mirip dengan spesies lain sehingga sulit dipisahkan oleh model hanya berdasarkan fitur konvolusi.
- Arsitektur Model Kurang Sesuai:  
Model yang terlalu sederhana atau tidak cukup dalam mungkin tidak mampu mempelajari fitur high-level yang diperlukan untuk membedakan Spesies X.

Penyebab lain bisa mencakup kualitas data yang buruk (misalnya gambar blur atau noise) atau masalah label (mislabeling).

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

**Jawaban:**

Ini adalah ciri klasik *overfitting*. Model yang lebih kompleks punya kapasitas besar untuk mempelajari detail data training, bahkan sampai noise-nya. Namun, ini justru merusak kemampuannya untuk generalisasi ke data baru. Penambahan kapasitas tidak selalu meningkatkan generalisasi karena:

- Model yang terlalu dalam/terlalu banyak parameter dapat *memorize* data tanpa belajar pola yang benar-benar berguna.
- Over-parameterization memperbesar risiko *overfitting* terutama jika data training terbatas.
- Regularisasi yang kurang memadai akan memperburuk situasi.

Tiga kesalahan desain arsitektur yang memicu degradasi:

1. Penambahan Lapisan Tanpa Regularisasi:  
Menambah lapisan konvolusi/dense tanpa disertai dropout atau batch normalization.
2. Kernel Besar Berlebihan:  
Menggunakan kernel konvolusi yang sangat besar yang membuat model belajar terlalu spesifik.
3. Pooling Kurang Efektif:  
Salah desain pada pooling (misalnya mengurangi pooling untuk mempertahankan resolusi tinggi terlalu lama), yang menyebabkan model belajar terlalu detail pada fitur spatial yang tidak signifikan.