Analisa

1. Jika menggunakan model MLP dengan 3 hidden layer (256-128-64) menghasilkan underfitting pada dataset ini, modifikasi apa yang akan dilakukan pada arsitektur? Jelaskan alasan setiap perubahan dengan mempertimbangkan bias-variance tradeoff!

Jawab:

Underfitting terjadi ketika model terlalu sederhana untuk menangkap pola dari data, misalnya karena bias terlalu tinggi.

Solusi Arsitektural:

- **Menambah jumlah neuron per layer**
  Misalnya ubah arsitektur dari (256-128-64) menjadi (512-256-128).
  Alasan: Model lebih kompleks, kapasitas meningkat untuk mempelajari pola non-linear.
- **Menambah hidden layer**
  Ubah menjadi 4–5 layer, contohnya: (512-256-128-64)
  Alasan: Kedalaman jaringan bisa membantu dalam membangun representasi bertingkat dari fitur.
- **Mengurangi regularisasi (dropout, L2 penalty)**
  Alasan: Regularisasi mencegah overfitting, tapi terlalu kuat bisa menyebabkan underfitting.

Modifikasi di atas **menurunkan bias**, tetapi **meningkatkan variance**, jadi tetap perlu dipantau melalui validasi. Ini adalah Bias-Variance Tradeoff

2. Selain MSE, loss function apa yang mungkin cocok untuk dataset ini? Bandingkan kelebihan dan kekurangannya, serta situasi spesifik di mana alternatif tersebut lebih unggul daripada MSE!

Jawab:

Loss Function	Kelebihan	Kekurangan	Cocok Saat
MAE (Mean Absolute Error)	Lebih robust terhadap outlier	Gradient konstan, bisa konvergensi lambat	Outlier signifikan, tidak ingin penalti besar
Huber Loss	Kombinasi MAE dan MSE, tahan outlier	Perlu parameter delta yang harus dituning	Tradeoff antara noise & sensitivitas

Loss Function	Kelebihan	Kekurangan	Cocok Saat
Log-Cosh Loss	Smooth, mirip MSE tapi lebih tahan outlier	Perhitungan lebih kompleks	Data dengan noise moderat
Quantile Loss	Cocok untuk prediksi interval	Interpretasi lebih sulit	Ingin prediksi median / percentile tertentu

3. Jika salah satu fitur memiliki range nilai 0-1, sedangkan fitur lain 100-1000, bagaimana ini memengaruhi pelatihan MLP? Jelaskan mekanisme matematis (e.g., gradien, weight update) yang terdampak!

Jawab:

Jika fitur A berada di 0–1 dan fitur B di 100–1000:

- **Masalah:** Fitur dengan skala besar akan memiliki kontribusi dominan pada output neuron.
- **Akibat:**
    - **Gradient tidak seimbang:** Berat dari fitur besar mengalami pembaruan yang lebih besar.
    - **Learning menjadi tidak stabil:** MLP sulit konvergen karena bias distribusi gradien.
- **Mekanisme matematis:**
    - Input besar menghasilkan nilai pre-activation z = w*x + b besar.
    - Hal ini bisa menyebabkan **aktivasi saturasi** (misalnya sigmoid → gradient mendekati 0).
    - **Update weight (Δw = -lr * dL/dw)** jadi tidak proporsional.

Solusi: Normalisasi/standarisasi fitur sebelum training (MinMaxScaler, StandardScaler, dll).

4. Tanpa mengetahui nama fitur, bagaimana Anda mengukur kontribusi relatif setiap fitur terhadap prediksi model? Jelaskan metode teknikal (e.g., permutation importance, weight analysis) dan keterbatasannya!

Jawab:

Beberapa metode teknikal:

A. Permutation Feature Importance

- Mengacak 1 fitur, ukur penurunan performa.
- **Kelebihan:** Model-agnostic.
- **Keterbatasan:** Lambat, tidak bisa menangani fitur yang berkorelasi tinggi.

B. SHAP (SHapley Additive exPlanations)

- Mengukur kontribusi fitur secara individual & interaksi.
- **Kelebihan:** Interpretasi mendalam.
- **Keterbatasan:** Kompleks & mahal komputasi.

C. Weight Analysis (khusus linear layer)

- Lihat magnitude bobot input → layer pertama.
- **Kelebihan:** Cepat.
- **Keterbatasan:** Tidak memperhitungkan non-linearitas dan interaksi antar fitur.

5. Bagaimana Anda mendesain eksperimen untuk memilih learning rate dan batch size secara optimal? Sertakan analisis tradeoff antara komputasi dan stabilitas pelatihan!

Jawab:

Secara otomatis mencari learning rate optimal dengan plotting loss terhadap learning rate.

C. Tradeoff:

Parameter	Efek pada Training	Komputasi
Learning Rate	Tinggi → cepat tapi bisa divergen	Lebih sedikit iterasi
	Rendah → stabil tapi lambat	Lebih banyak iterasi
Batch Size	Kecil → lebih noisy, regularisasi alami	Lambat (banyak step)
	Besar → stabil tapi butuh memori besar	Cepat tapi mahal (GPU)

Rekomendasi: Mulai dengan lr kecil (1e-3) dan batch sedang (32/64), kemudian sesuaikan berdasarkan loss curve.