

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!**

Jawaban:

Jika MLP dengan arsitektur 3 hidden layer (256-128-64) mengalami underfitting, itu berarti model tidak cukup kompleks untuk menangkap pola dalam data. Untuk mengatasinya, beberapa modifikasi yang direkomendasikan:

- **Menambah jumlah neuron per layer**
Misalnya: (512-256-128). Ini meningkatkan kapasitas model untuk mempelajari pola yang lebih kompleks. Alasan: menambah neuron memperluas ruang representasi, mengurangi bias.
- **Menambah depth (layer tambahan)**
Misalnya: (512-256-128-64). Layer tambahan memungkinkan model mempelajari pola non-linear yang lebih dalam. Alasan: meningkatkan kompleksitas struktural untuk memecahkan masalah bias tinggi.
- **Mengurangi regularisasi/dropout**
Jika dropout terlalu tinggi, kurangi (misal dari 0.5 ke 0.3). Alasan: dropout berlebihan malah mempersulit model belajar, meningkatkan bias.
- **Menggunakan activation function yang lebih ekspresif**
Misalnya mengganti ReLU dengan LeakyReLU/ELU untuk menghindari dead neurons. Alasan: mengurangi risiko "gradient dying" yang menghambat pembelajaran.
- **Training lebih lama (epoch ditambah)**
Karena underfitting = belum cukup belajar, epoch tambahan membantu memaksimalkan pelatihan.

Bias-variance tradeoff: Perubahan di atas meningkatkan kapasitas model (menurunkan bias), tapi hati-hati agar tidak menyebabkan overfitting (variance naik). Solusinya: pantau validasi loss dan gunakan teknik early stopping.

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!

Jawaban:

Selain Mean Squared Error (MSE), loss function yang mungkin cocok adalah:

- **Mean Absolute Error (MAE)**
 - Kelebihan: Lebih robust terhadap outlier karena menghitung selisih absolut.
 - Kekurangan: Gradien konstan yang bisa membuat training lambat saat mendekati minimum.
 - Lebih unggul: Jika dataset punya outlier yang besar karena MSE terlalu penalizing untuk outlier.
- **Huber Loss**
 - Kelebihan: Gabungan MSE dan MAE—bertindak seperti MSE untuk error kecil dan MAE untuk error besar.
 - Kekurangan: Perlu menentukan parameter delta yang memisahkan error kecil dan besar.
 - Lebih unggul: Jika kita ingin loss yang smooth tapi tetap tahan terhadap outlier.
- **Log-Cosh Loss**
 - Kelebihan: Mirip MSE untuk error kecil, mirip MAE untuk error besar, plus lebih smooth.
 - Kekurangan: Komputasi sedikit lebih mahal.
 - Lebih unggul: Ketika kita ingin keseimbangan antara robust dan stabil.

Perbandingan dengan MSE: MSE sangat sensitif terhadap outlier karena kuadrat error memperbesar penalti. Alternatif seperti MAE atau Huber lebih cocok jika distribusi error tidak normal atau ada outlier dominan.

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!

Jawaban:

Jika satu fitur bernilai 0-1 dan yang lain 100-1000, MLP akan kesulitan karena:

- **Mekanisme matematis yang terdampak:**
 - **Forward pass:** Neuron dengan input berskala besar akan mendominasi output karena $z = w \cdot x + b$. Fitur besar memaksa model lebih fokus ke fitur tersebut.

- **Backpropagation:** Gradien untuk fitur besar akan lebih besar, memicu gradient explosion atau menyebabkan weight update tidak seimbang.
- **Weight update (SGD step):** $w = w - \eta \cdot \frac{\partial w}{\partial L}$. Jika satu fitur skala jauh lebih besar, bobotnya akan ter-update terlalu agresif, menyebabkan pelatihan tidak stabil.

Solusi: Lakukan scaling/normalisasi (misalnya StandardScaler atau MinMaxScaler) untuk memastikan semua fitur punya skala seragam, sehingga semua gradient berkontribusi seimbang selama training.

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!

Jawaban:

Tanpa tahu nama fitur, kita tetap bisa mengukur kontribusi relatif dengan:

- **Permutation Importance**
 - Acak satu fitur di test set.
 - Hitung drop performa model.
- Semakin besar drop-nya, semakin penting fitur tersebut.
 - Kelebihan: Tidak bergantung pada jenis model.
 - Keterbatasan: Mahal secara komputasi, bisa bias jika fitur saling berkorelasi.
- **Weight Analysis** (khusus MLP linear layer pertama)

Lihat magnitude bobot di input layer. Fitur dengan bobot besar dianggap lebih berpengaruh.

 - Kelebihan: Cepat dan mudah.
 - Keterbatasan: Tidak menangkap interaksi non-linear; bobot kecil belum tentu tidak penting.
- **SHAP (SHapley Additive exPlanations)**

Memberi kontribusi lokal (per data point) dan global.

 - Kelebihan: Cocok untuk model kompleks, menjelaskan dampak fitur secara matematis.
 - Keterbatasan: Komputasi berat dan kompleks untuk model besar.

Ideal: Kombinasi Permutation + SHAP untuk dapat gambaran global & lokal yang lengkap.

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

Jawaban:

1. Grid Search:

- Learning rate: [0.001, 0.01, 0.1]
 - Batch size: [32, 64, 128]
- Cek kombinasi performa di validation set.

2. Learning Rate Finder:

Naikkan learning rate secara eksponensial selama beberapa batch. Plot loss vs learning rate untuk menemukan titik optimal (lereng tercuram).

3. Cyclic LR Scheduler:

Tes learning rate dinamis dalam range (min-max) untuk lihat performa fluktuatif.

4. Batch Size Tuning:

Mulai dari kecil (32), naik bertahap sambil cek:

- Kecepatan training
- Stabilitas loss
- Akurasi final

Tradeoff:

- **Learning Rate:**
 - Terlalu besar → instabil, loss zig-zag atau divergen.
 - Terlalu kecil → lambat konvergen, waktu pelatihan lebih lama.
- **Batch Size:**
 - Kecil → update lebih sering, generalisasi lebih baik (tapi noisy & lambat).
 - Besar → stabil, cepat, tapi rawan local minima dan butuh memori lebih besar.

Optimal: Cari kombinasi learning rate cukup besar (stabil tapi cepat konvergen) dengan batch size yang muat di GPU tapi masih mampu generalisasi baik. Early stopping dan monitoring validation loss wajib dipakai untuk mencegah overfitting.