NAMA = AZIZAH RAHMA ASRI
NIM = 1103213025
UTS DEEP LEARNING

Analisis Model MLP untuk Regresi dan Klasifikasi pada Dataset RegresiUTSTelkom

1. Modifikasi Arsitektur MLP jika Mengalami Underfitting

Underfitting menunjukkan bahwa model MLP (Multi-Layer Perceptron) dengan arsitektur 256-128-64 tidak cukup kompleks untuk mempelajari pola dalam dataset. Beberapa modifikasi yang dapat dilakukan:
- Menambah jumlah neuron per layer, misalnya menjadi 512-256-128, agar kapasitas model meningkat.
- Menambah jumlah hidden layer, seperti 512-256-128-64, agar model mampu mempelajari representasi yang lebih dalam.
- Mengurangi atau menghapus regularisasi seperti dropout atau L2 jika terlalu besar, karena dapat menyebabkan informasi terlalu banyak dibuang.
- Mengganti fungsi aktivasi menjadi LeakyReLU atau ELU agar dapat mengatasi masalah "dead neurons".

Perubahan tersebut bertujuan menurunkan bias dan meningkatkan kapasitas model. Namun, sesuai prinsip bias-variance tradeoff, peningkatan kapasitas dapat meningkatkan varians dan risiko overfitting, sehingga perlu disertai dengan validasi silang atau early stopping.

2. Alternatif Loss Function Selain MSE

Selain MSE (Mean Squared Error), beberapa loss function yang cocok untuk regresi antara lain:

- MAE (Mean Absolute Error): robust terhadap outlier, namun gradiennya konstan dan tidak smooth.
- Huber Loss: gabungan antara MSE dan MAE, stabil untuk data noisy, tetapi memerlukan hyperparameter delta.
- Log-Cosh Loss: mirip MSE tapi lebih tahan terhadap outlier, namun perhitungan lebih lambat.

Situasi:
- MSE cocok untuk data tanpa outlier.
- MAE lebih baik jika terdapat banyak outlier.
- Huber cocok jika outlier tidak dominan.

NAMA = AZIZAH RAHMA ASRI
NIM = 1103213025
UTS DEEP LEARNING

- Log-Cosh untuk distribusi normal yang noisy.

Pemilihan loss function bergantung pada distribusi target dan sensitivitas terhadap outlier.

3. Pengaruh Range Nilai Fitur Terhadap Pelatihan MLP

Perbedaan range antar fitur (misalnya 0–1 vs 100–1000) menyebabkan ketidakseimbangan saat proses pembelajaran karena:

- Gradien dari fitur dengan skala besar menjadi dominan saat update bobot.
- Fitur dengan skala kecil cenderung diabaikan (vanishing gradient).
- Algoritma optimisasi seperti SGD akan memiliki langkah besar di arah fitur berskala besar, membuat loss landscape tidak stabil.

Secara matematis, dalam update bobot:
$w_{new} = w_{old} - \eta * \partial L/\partial w$
Jika x besar, $\partial L/\partial w$ menjadi besar dan membuat update tidak proporsional.

Solusi terbaik adalah melakukan feature scaling (misalnya dengan StandardScaler atau MinMaxScaler) agar semua fitur memiliki kontribusi yang seimbang.

4. Mengukur Kontribusi Fitur Tanpa Mengetahui Namanya

Beberapa metode teknikal untuk mengukur kontribusi fitur:
- Permutation Importance: mengacak nilai suatu fitur dan mengukur penurunan performa. Kelemahannya adalah tidak stabil jika fitur saling berkorelasi.
- SHAP (SHapley Additive exPlanations): berdasarkan teori game, mengukur kontribusi fitur per prediksi. Kuat secara teoritis tapi mahal secara komputasi.
- Weight Analysis: menganalisis bobot layer pertama pada MLP. Hanya efektif jika tidak ada banyak interaksi non-linear antar fitur.
- Partial Dependence Plot (PDP): menunjukkan pengaruh satu fitur terhadap output. Terbatas jika ada korelasi antar fitur.

Setiap metode memiliki kelebihan dan keterbatasan, dan sering digunakan secara komplementer.

NAMA = AZIZAH RAHMA ASRI
NIM = 1103213025
UTS DEEP LEARNING

5. Eksperimen untuk Memilih Learning Rate dan Batch Size

Untuk mendapatkan learning rate dan batch size optimal, dilakukan eksperimen sebagai berikut:

- Gunakan grid search atau pencarian manual dengan learning rate seperti [1e-4, 1e-3, 1e-2, 1e-1].
- Amati loss curve untuk melihat stabilitas dan kecepatan konvergensi.
- Batch size diuji antara 16, 32, 64, hingga 128 untuk melihat pengaruh terhadap noise dan efisiensi.

Tradeoff:
- Learning rate kecil: stabil tapi lambat. Besar: cepat tapi bisa divergen.
- Batch size kecil: cepat adaptasi tapi noisy. Besar: stabil tapi risiko local minima.

Gunakan learning rate scheduler (misalnya ReduceLROnPlateau) dan early stopping untuk hasil terbaik.