

**LAPORAN UTS ANALISA TABULAR TASK**

**DATASET REGRESI**



Disusun oleh :

Rafif Zain Hibatullah

(1103213163)

**PRODI S1 TEKNIK KOMPUTER**

**FAKULTAS TEKNIK ELEKTRO**

**UNIVERSITAS TELKOM BANDUNG**

**2025**

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!

- Mengurangi jumlah layer: Jika model dengan tiga layer tersebar (256-128-64) menghasilkan underfitting, berarti model tidak memiliki cukup kapasitas untuk menangkap kompleksitas dataset. Salah satu cara untuk mengatasi masalah ini adalah dengan mengurangi jumlah hidden layers menjadi dua layer, misalnya (128, 64), atau menggunakan satu layer yang lebih besar seperti (256). Tradeoff: Jika terlalu banyak mengurangi layer, model mungkin tidak cukup kuat untuk menangkap hubungan kompleks dalam data, dan bisa meningkatkan bias (underfitting).
- Menambah jumlah neuron: Jika arsitektur model saat ini terlalu sederhana dan tidak cukup untuk menangkap pola dalam data, bisa ditambah jumlah neuron per layer. Misalnya, mencoba (512, 256) atau lebih untuk memberikan kapasitas lebih pada model agar bisa menangkap pola yang lebih kompleks. Tradeoff: Jika model terlalu besar, variance bisa meningkat dan menyebabkan overfitting.
- Penyesuaian Regularisasi: Jika overfitting bukan masalah utama, maka bisa diturunkan penalti regularisasi ( $\alpha$ ) untuk mengurangi pembatasan pada model, sehingga memberikan ruang lebih banyak bagi model untuk belajar. Tradeoff: Mengurangi  $\alpha$  terlalu banyak dapat menyebabkan model terlalu bebas, yang berisiko meningkatkan overfitting.

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!

- Mean Absolute Error (MAE):

Kelebihan: MAE lebih robust terhadap outlier dibandingkan dengan MSE, karena tidak membesarkan kesalahan besar dengan mengkuadratkan kesalahan.

Kekurangan: MAE kurang sensitif terhadap kesalahan besar. Jika ingin memberikan penalti yang lebih besar terhadap kesalahan besar, MAE mungkin tidak cocok.

Situasi spesifik: MAE bisa lebih baik digunakan jika data memiliki outliers dan tidak ingin kesalahan besar terlalu mempengaruhi model.

- Huber Loss:

Kelebihan: Huber loss menggabungkan kelebihan MSE dan MAE. Untuk kesalahan kecil, Huber bersifat kuadrat seperti MSE, namun untuk kesalahan besar, ia berperilaku seperti MAE.

Kekurangan: Huber loss memerlukan parameter tambahan delta, yang mengontrol ambang batas antara MSE dan MAE.

Situasi spesifik: Huber loss sangat berguna jika dataset mengandung outliers, tetapi tetap ingin memiliki model yang sensitif terhadap kesalahan kecil.

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!

MLP sangat sensitif terhadap skala fitur. Fitur dengan skala yang besar (range 100-1000) akan mendominasi dalam perhitungan gradien, menyebabkan update bobot yang lebih besar untuk fitur dengan skala lebih besar, sementara fitur dengan skala kecil (range 0-1) akan memiliki pembaruan bobot yang lebih kecil. Perbedaan skala ini menyebabkan gradien yang tidak seimbang, yang mempengaruhi konvergensi model. Weight update menjadi tidak seimbang, dan model mungkin kesulitan dalam menyeimbangkan pengaruh masing-masing fitur dalam prediksi.

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!

Jika tidak diketahui nama fitur, namun ingin mengukur kontribusi relatif setiap fitur terhadap prediksi model, dapat digunakan beberapa metode yang bersifat model-agnostik atau yang tidak memerlukan pengetahuan langsung tentang nama fitur. Beberapa metode tersebut yang dapat digunakan:

- **Permutation Importance:**

Permutation importance mengukur penurunan performa model ketika nilai dari fitur diacak. Semakin besar penurunan performa, semakin besar kontribusi fitur tersebut terhadap prediksi model.

Keterbatasan: Permutation importance bisa sangat komputasional mahal, terutama jika dataset besar. Selain itu, ia mungkin tidak akurat jika fitur sangat berkorelasi satu sama lain.

- **Weight Analysis (untuk Model Linier):**

Jika menggunakan model linier (seperti regresi linier atau MLP dengan aktivasi linier), bisa melihat bobot yang dihasilkan oleh model untuk menilai pentingnya setiap fitur. Fitur dengan bobot yang lebih besar biasanya memiliki pengaruh lebih besar terhadap prediksi.

Keterbatasan: Untuk model non-linier seperti MLP dengan aktivasi non-linier (seperti ReLU), bobot saja tidak cukup untuk menunjukkan pentingnya fitur, karena kontribusi fitur di lapisan tersembunyi lebih kompleks.

- SHAP (SHapley Additive exPlanations):

SHAP adalah pendekatan berbasis game teori untuk mengukur kontribusi setiap fitur terhadap prediksi model. Ini memberikan gambaran yang lebih tepat dan konsisten untuk model non-linier seperti MLP.

Keterbatasan: SHAP dapat menjadi komputasional mahal dan tidak selalu cepat untuk diterapkan pada model besar.

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

Desain Eksperimen:

Learning Rate:

1. Tentukan rentang learning rate (misalnya 0.0001 hingga 0.01).
2. Gunakan RandomizedSearchCV atau GridSearchCV untuk mencoba nilai-nilai dalam rentang tersebut.
3. Plot kehilangan (loss) saat pelatihan untuk melihat apakah learning rate terlalu tinggi atau terlalu rendah (mungkin menyebabkan overfitting atau underfitting).

Batch Size:

1. Coba beberapa ukuran batch yang berbeda, seperti 32, 64, 128.
2. Batch size kecil mengarah pada lebih banyak pembaruan bobot lebih cepat, namun mungkin lebih berisik dalam gradien dan membutuhkan lebih banyak epoch untuk konvergen.
3. Batch size besar akan memberikan pembaruan yang lebih stabil tetapi lebih memakan waktu per iterasi.

Tradeoff:

- Batch size kecil lebih cepat per iterasi, namun bisa membutuhkan lebih banyak epoch untuk konvergen.
- Batch size besar memberikan pembaruan yang lebih stabil tetapi lebih lambat dalam per iterasi dan memerlukan lebih banyak memori.
- Learning rate tinggi mempercepat konvergensi tetapi bisa menyebabkan instabilitas.
- Learning rate rendah akan stabil tetapi lebih lambat dalam konvergensi.