

Analisis Deep Learning: Evaluasi Model MLP untuk Regresi dan Klasifikasi

Dhanendra Nivadirrokhman (1103220113)

1. Pendahuluan

Multi-Layer Perceptron (MLP) merupakan salah satu arsitektur dasar dalam deep learning yang digunakan secara luas, baik untuk tugas regresi maupun klasifikasi. Laporan ini menyajikan evaluasi terhadap model MLP dengan arsitektur 256–128–64 yang digunakan dalam tugas regresi pada dataset tanpa label fitur yang jelas (feat_1, feat_2, dst). Selain itu, dilakukan eksplorasi terhadap strategi tuning model, evaluasi performa, serta pentingnya pre-processing data.

2. Penanganan Underfitting pada Model MLP

Ketika model mengalami underfitting, artinya kompleksitas model belum cukup untuk menangkap pola dari data pelatihan. Untuk mengatasi hal ini, beberapa strategi yang dapat dilakukan antara lain:

- a) **Penambahan Neuron atau Layer**
Menambah jumlah neuron atau layer dapat meningkatkan kapasitas representasi model terhadap relasi non-linear antar fitur. Namun, perlu diwaspadai potensi overfitting jika terlalu kompleks.
- b) **Pengurangan Regularisasi**
Regularisasi seperti dropout atau penalti L2 dapat menyebabkan underfitting jika diterapkan secara agresif. Mengurangi dropout rate atau parameter regularisasi dapat memberikan fleksibilitas lebih kepada model.
- c) **Eksperimen Fungsi Aktivasi**
Penggunaan fungsi aktivasi alternatif seperti LeakyReLU atau ELU dapat menghindari permasalahan *dying ReLU* dan meningkatkan aliran gradien, terutama pada fitur dengan nilai negatif.

3. Evaluasi Loss Function

Dalam tugas regresi, pemilihan loss function sangat berpengaruh terhadap sensitivitas model terhadap error. Tabel berikut membandingkan beberapa loss function populer:

Loss Function	Kelebihan	Kekurangan	Cocok Untuk
MSE	Mudah dihitung dan terdiferensiasi	Sangat sensitif terhadap outlier	Data bersih dengan distribusi normal
MAE	Lebih tahan terhadap outlier	Tidak halus di nol (non-differentiable)	Dataset dengan banyak outlier
Huber	Kombinasi kekuatan MSE dan MAE	Memerlukan parameter delta	Skala error sedang

Log-Cosh	Lebih smooth dibanding MSE, toleran terhadap outlier	Sedikit lebih mahal komputasi	Data dengan distribusi tidak normal
-----------------	--	-------------------------------	-------------------------------------

4. Pentingnya Feature Scaling

Model MLP sangat sensitif terhadap skala fitur. Fitur dengan rentang nilai yang besar dapat mendominasi pembaruan bobot, menyebabkan pelatihan tidak optimal. Oleh karena itu, scaling fitur menjadi langkah penting dalam pre-processing data.

Beberapa teknik scaling yang umum digunakan:

- **Standardisasi (Z-score normalization)**
- **Normalisasi Min-Max**
- **Robust Scaling** (tahan terhadap outlier)

Scaling membantu menyamakan skala antar fitur sehingga proses pelatihan menjadi lebih stabil dan cepat konvergen.

5. Evaluasi Kontribusi Fitur Tanpa Nama

Meskipun fitur tidak memiliki label deskriptif (mis. feat_1, feat_2, dst), kita tetap dapat mengevaluasi kontribusinya menggunakan metode berikut:

- Analisis Bobot Layer Pertama**
Mengamati bobot dari layer input ke hidden layer pertama dapat memberikan indikasi awal tentang pentingnya fitur. Namun, pendekatan ini bersifat linear dan tidak mempertimbangkan kompleksitas jaringan.
- SHAP (SHapley Additive exPlanations)**
SHAP memberikan nilai kontribusi tiap fitur terhadap hasil prediksi, baik secara lokal (per sample) maupun global. Metode ini sangat cocok untuk model non-linear seperti MLP dan mendukung interpretasi model secara mendalam.

6. Eksperimen Hyperparameter: Learning Rate & Batch Size

Untuk mendapatkan konfigurasi training yang optimal, dilakukan eksperimen terhadap kombinasi **learning rate** dan **batch size**. Parameter-parameter yang diuji antara lain:

- **Learning Rate:** 0.0001, 0.001, 0.01
- **Batch Size:** 32, 64, 128, 256

Perbandingan efeknya:

Parameter	Nilai Rendah	Nilai Tinggi
Learning Rate	Stabil namun lambat konvergen	Cepat belajar namun risiko overshoot
Batch Size	Lebih noisy (efek regularisasi alami)	Lebih stabil tapi rawan overfitting

Metode seperti **random search** atau **Keras Tuner** direkomendasikan untuk eksplorasi parameter yang lebih efisien.

7. Hasil Evaluasi Model

Berikut adalah hasil evaluasi dari model MLP yang telah dilatih:

- **Akurasi Validasi Terakhir:** 79.03%
- **Akurasi Validasi Tertinggi:** 79.34%
- **Durasi Total Pelatihan:** 45 menit 27 detik

F1-Score Per Kelas:

Kelas	Precision	Recall	F1-Score
0	0.78	0.73	0.75
1	0.72	0.77	0.75

- **Akurasi Keseluruhan:** 75%

8. Kesimpulan

Model MLP menunjukkan performa yang cukup baik untuk tugas prediktif yang diberikan, dengan akurasi validasi mencapai ~79% dan akurasi keseluruhan 75%. Beberapa poin penting yang dapat disimpulkan:

- Penyesuaian arsitektur dan hyperparameter masih dapat meningkatkan performa model.
- Proses pre-processing seperti scaling fitur sangat penting untuk efisiensi pelatihan.
- Interpretabilitas model dapat ditingkatkan menggunakan metode seperti SHAP untuk memahami pengaruh masing-masing fitur.
- Eksperimen sistematis terhadap parameter pelatihan perlu dilakukan untuk mendapatkan konfigurasi optimal.

Secara keseluruhan, model saat ini sudah cukup layak digunakan sebagai baseline dalam pengembangan lebih lanjut.