

Nama : Satya Athaya Daniswara

NIM : 1103213152

KELAS : TK4501

Analisis Komprehensif Implementasi MLP untuk Regresi

Gambaran Arsitektur

Implementasi ini menghadirkan arsitektur **Multilayer Perceptron (MLP)** yang fleksibel untuk tugas regresi. Jaringan neural ini dibangun menggunakan PyTorch dengan sejumlah pilihan desain utama yang membuatnya serbaguna dan dapat disesuaikan.

Fitur Utama Arsitektur

1. Konfigurasi Lapisan yang Fleksibel

- Mendukung jumlah lapisan tersembunyi yang bervariasi.
- Jumlah neuron per lapisan dapat disesuaikan secara seragam di seluruh lapisan tersembunyi.
- Dimensi input dapat diadaptasi sesuai ruang fitur.
- Satu neuron output untuk prediksi regresi.

2. Fleksibilitas Fungsi Aktivasi

- Mendukung berbagai fungsi aktivasi: ReLU, Sigmoid, Tanh.
- Termasuk opsi aktivasi linear melalui fungsi Identitas.
- Tidak ada fungsi aktivasi di lapisan output (sesuai untuk regresi).

Analisis Implementasi Pelatihan

Implementasi pelatihan menunjukkan perhatian yang baik terhadap praktik deep learning modern:

Aspek Positif Implementasi

1. Manajemen Data

- Konversi tensor yang tepat untuk kompatibilitas dengan PyTorch.
- Pemrosesan batch melalui DataLoader.
- Skala data dan prapemrosesan.
- Pembagian data train-test untuk validasi.

2. Desain Loop Pelatihan

- Menggunakan optimizer Adam untuk pembelajaran adaptif.
- Mengimplementasikan mini-batch gradient descent.
- Melacak loss untuk pelatihan dan pengujian.
- Pelaporan progres secara berkala setiap 10 epoch.

3. Metode Evaluasi Kinerja

- Evaluasi yang komprehensif dengan berbagai metrik:
 - R-squared (R^2) untuk kesesuaian model.
 - Mean Squared Error (MSE) untuk magnitudo kesalahan.
 - Pelacakan terpisah untuk metrik pelatihan dan pengujian.

Kerangka Eksperimen

Pengaturan eksperimen menunjukkan pendekatan sistematis untuk tuning hyperparameter:

Rentang Hyperparameter

Lapisan Tersembunyi: [1, 2, 3]

Neuron per Lapisan: [4, 8, 16, 32, 64]

Fungsi Aktivasi: ['linear', 'sigmoid', 'relu', 'tanh']

Epoch: [1, 10, 25, 50, 100, 250]

Learning Rates: [10, 1, 0.1, 0.01, 0.001, 0.0001]

Batch Sizes: [16, 32, 64, 128, 256, 512]

Analisis Hasil Pelatihan

Dari hasil pelatihan yang diberikan, terlihat beberapa pola:

1. Pola Konvergensi

- Loss Awal (Epoch 10): ~0,0761
- Loss Akhir (Epoch 50): ~0,0719
- Loss Pengujian Akhir: ~0,0761

2. Stabilitas Pembelajaran

- Loss pelatihan menurun secara konsisten.
- Loss pengujian stabil, menunjukkan generalisasi yang baik.
- Kesenjangan kecil antara loss pelatihan dan pengujian menunjukkan tidak ada overfitting yang signifikan.

3. Perbandingan Beberapa Jalankan

- Tiga sesi pelatihan ditampilkan, masing-masing menunjukkan peningkatan.
- Jalankan terakhir mencapai loss pelatihan terbaik sebesar 0,0593.
- Loss pengujian tetap stabil di sekitar 0,076, menunjukkan kinerja yang robust.

Rekomendasi Perbaikan

1. Peningkatan Arsitektur

- Pertimbangkan untuk menambahkan **dropout layer** untuk regularisasi yang lebih baik.
- Tambahkan **batch normalization** di antara lapisan untuk pelatihan yang lebih stabil.
- Implementasikan **learning rate scheduling** untuk konvergensi yang lebih baik.

2. Proses Pelatihan

- Tambahkan **early stopping** untuk mencegah overfitting.
- Terapkan **k-fold cross-validation** untuk evaluasi yang lebih andal.
- Tambahkan **gradient clipping** untuk mencegah gradien meledak.

3. Pemantauan dan Visualisasi

- Tambahkan pemantauan learning rate.
- Implementasikan visualisasi distribusi bobot dan gradien.
- Tambahkan fungsionalitas plotting prediksi vs aktual.

Analisis Kinerja

Model menunjukkan karakteristik kinerja yang menjanjikan:

1. Efisiensi Pelatihan

- Penurunan loss yang konsisten di setiap epoch.
- Tingkat konvergensi yang baik dengan learning rate yang dipilih.
- Ukuran batch yang masuk akal untuk efisiensi memori.

2. Generalisasi

- Kesenjangan kecil antara loss pelatihan dan pengujian.
- Loss pengujian stabil di seluruh epoch.
- Tidak ada tanda-tanda overfitting yang signifikan.

Kesimpulan

Implementasi ini mewakili fondasi yang solid untuk tugas regresi berbasis MLP. Arsitektur ini terstruktur dengan baik dan fleksibel, sementara proses pelatihannya mengikuti praktik terbaik.

Kerangka eksperimen memungkinkan optimisasi hyperparameter yang sistematis, dan hasilnya menunjukkan konvergensi dan generalisasi yang baik.

Kekuatan utama terletak pada desain modular dan metrik evaluasi yang komprehensif. Area untuk perbaikan terutama berfokus pada penambahan teknik pelatihan modern seperti dropout dan batch normalization, serta peningkatan kemampuan pemantauan.