

1. Modifikasi Arsitektur untuk Mengatasi Underfitting

Jika model MLP dengan 3 hidden layer (256-128-64) menghasilkan underfitting, artinya model terlalu sederhana dan tidak mampu menangkap kompleksitas data. Untuk mengatasi hal ini, kita perlu meningkatkan kapasitas model dengan mempertimbangkan bias-variance tradeoff:

Modifikasi yang dapat dilakukan:

1. Menambah jumlah neuron di setiap hidden layer: Misalnya, menjadi (512-256-128).
 - Alasan: Meningkatkan jumlah neuron akan memperluas ruang hipotesis model, memungkinkannya mempelajari pola yang lebih kompleks.
 - Bias-Variance Tradeoff: Meningkatkan kapasitas model berpotensi mengurangi bias (kesalahan sistematis) tetapi dapat meningkatkan variance (sensitivitas terhadap variasi data training).
2. Menambah jumlah hidden layer: Misalnya, menjadi 4 atau 5 layer.
 - Alasan: Menambah layer memungkinkan model mempelajari representasi data yang lebih hierarkis dan abstrak.
 - Bias-Variance Tradeoff: Mirip dengan poin 1, menambah layer meningkatkan kapasitas model dan berpotensi mengurangi bias tetapi meningkatkan variance.
3. Menggunakan fungsi aktivasi non-linear yang lebih kompleks: Misalnya, mengganti ReLU dengan Swish atau Mish.
 - Alasan: Fungsi aktivasi non-linear memungkinkan model mempelajari hubungan non-linear dalam data. Fungsi yang lebih kompleks dapat menangkap pola yang lebih rumit.
 - Bias-Variance Tradeoff: Menggunakan fungsi aktivasi yang lebih kompleks dapat meningkatkan kapasitas model dan berpotensi mengurangi bias tetapi meningkatkan variance.
4. Menambahkan fitur interaksi: Membuat fitur baru yang merupakan kombinasi dari fitur-fitur yang ada.
 - Alasan: Fitur interaksi memungkinkan model mempelajari hubungan antar fitur yang mungkin tidak tertangkap secara langsung.
 - Bias-Variance Tradeoff: Menambah fitur dapat meningkatkan kapasitas model, tetapi terlalu banyak fitur dapat menyebabkan overfitting.

Penting untuk dicatat: Modifikasi ini harus dilakukan secara bertahap dan dievaluasi performanya untuk menghindari overfitting.

2. Alternatif Loss Function untuk Regresi

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

- MAE (Mean Absolute Error):
 - Kelebihan: Lebih robust terhadap outlier dibandingkan MSE.
 - Kekurangan: Gradien MAE tidak kontinu di nol, yang dapat menyebabkan masalah dalam optimasi.
 - Situasi unggul: Ketika dataset memiliki banyak outlier.
- Huber Loss:
 - Kelebihan: Menggabungkan kelebihan MSE dan MAE. Robust terhadap outlier tetapi memiliki gradien yang lebih baik daripada MAE.
 - Kekurangan: Membutuhkan parameter tambahan (delta) yang perlu di tuning.
 - Situasi unggul: Ketika dataset memiliki outlier tetapi ingin mempertahankan sifat-sifat baik MSE.
- Log-Cosh Loss:
 - Kelebihan: Mirip dengan Huber Loss, robust terhadap outlier dan memiliki gradien yang kontinu.
 - Kekurangan: Komputasinya lebih mahal daripada MSE atau MAE.
 - Situasi unggul: Ketika menginginkan loss function yang smooth dan robust terhadap outlier.

3. Pengaruh Range Fitur yang Berbeda

Range nilai fitur yang berbeda dapat memengaruhi pelatihan MLP karena:

- Fitur dengan range nilai yang lebih besar akan mendominasi: Gradien yang dihitung untuk fitur tersebut akan lebih besar, menyebabkan bobot yang terkait dengan fitur tersebut diperbarui lebih agresif.
- Konvergensi yang lebih lambat: Optimasi model akan lebih sulit karena model perlu menyeimbangkan pengaruh fitur dengan range nilai yang berbeda.

Mekanisme Matematis:

- **Gradien:** Gradien dihitung berdasarkan turunan loss function terhadap bobot. Fitur dengan range nilai yang lebih besar akan memiliki turunan yang lebih besar, menghasilkan gradien yang lebih besar.
- **Weight Update:** Bobot diperbarui berdasarkan gradien dan learning rate. Gradien yang lebih besar akan menyebabkan perubahan bobot yang lebih besar.

Solusi:

- **Standardisasi atau Normalisasi:** Menerapkan StandardScaler atau MinMaxScaler untuk menskalakan fitur ke range yang sama. Hal ini memastikan bahwa semua fitur memiliki pengaruh yang seimbang selama pelatihan.

4. Mengukur Kontribusi Relatif Fitur

Tanpa mengetahui nama fitur, kita dapat mengukur kontribusi relatif setiap fitur menggunakan teknik berikut:

- **Permutation Importance:**
 - **Metode:** Mengacak nilai-nilai suatu fitur dan mengukur penurunan performa model. Penurunan yang lebih besar menunjukkan fitur tersebut lebih penting.
 - **Keterbatasan:** Permutation Importance dapat dipengaruhi oleh korelasi antar fitur.
- **Weight Analysis:**
 - **Metode:** Menganalisis bobot yang dipelajari oleh model, khususnya pada layer pertama. Bobot yang lebih besar menunjukkan fitur tersebut lebih penting.
 - **Keterbatasan:** Interpretasi bobot bisa sulit, terutama pada model yang kompleks. Hanya berlaku untuk model linear atau layer pertama model non-linear.
- **SHAP Values (SHapley Additive exPlanations):**
 - **Metode:** Menghitung kontribusi setiap fitur terhadap prediksi individu dengan mempertimbangkan semua kemungkinan kombinasi fitur.
 - **Keterbatasan:** Komputasinya bisa mahal, terutama untuk dataset yang besar.

5. Mendesain Eksperimen untuk Learning Rate dan Batch Size

Untuk memilih learning rate dan batch size secara optimal, kita dapat melakukan eksperimen berikut:

1. Learning Rate:

- Range: Mencoba berbagai learning rate dalam skala logaritmik, misalnya $1e-1$, $1e-2$, $1e-3$, $1e-4$, $1e-5$.
- Evaluasi: Memantau loss function selama training. Learning rate yang optimal akan menghasilkan penurunan loss yang cepat dan stabil.

2. Batch Size:

- Range: Mencoba berbagai batch size, misalnya 32, 64, 128, 256.
- Evaluasi: Membandingkan performa model (akurasi, loss) dan waktu training.

Tradeoff Komputasi dan Stabilitas:

- Learning Rate: Learning rate yang besar dapat mempercepat training tetapi menyebabkan model tidak stabil dan gagal konvergen. Learning rate yang kecil akan menghasilkan training yang lebih lambat tetapi lebih stabil.
- Batch Size: Batch size yang besar dapat mempercepat training per epoch tetapi membutuhkan lebih banyak memori. Batch size yang kecil dapat menghasilkan training yang lebih lambat tetapi memungkinkan model untuk menggeneralisasi lebih baik.