

Muhammad Aizar Yazid

1103223097

Analisis MLP

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!

Jawaban :

Jika model dengan arsitektur 3 layer (256-128-64) menghasilkan underfitting, beberapa modifikasi yang bisa dilakukan untuk meningkatkan kinerja model adalah sebagai berikut:

- Menambah jumlah unit pada hidden layers: Underfitting bisa terjadi jika model terlalu sederhana untuk menangkap kompleksitas data. Menambah ukuran hidden layer (misalnya, dari 256-128-64 menjadi 512-256-128) akan memberi model kapasitas lebih besar untuk mempelajari pola-pola yang lebih rumit.
- Menambah jumlah hidden layer: Selain menambah jumlah unit, kita juga bisa menambah jumlah hidden layer, misalnya menambah menjadi 4 atau 5 layer. Hal ini dapat membantu model menangkap hubungan yang lebih kompleks di dalam data.
- Mengurangi Dropout: Dropout digunakan untuk mengatasi overfitting, namun jika model underfitting, kita dapat mengurangi tingkat dropout atau bahkan menghapus beberapa layer dropout, terutama pada model yang lebih dalam.
- Menambah Epochs: Kadang-kadang underfitting disebabkan oleh pelatihan yang tidak cukup lama. Menambah jumlah epoch atau menyesuaikan learning rate juga bisa membantu model belajar lebih baik.

Alasan perubahan: Modifikasi-modifikasi ini bertujuan untuk mengurangi bias yang tinggi, yang sering kali menjadi penyebab utama underfitting. Namun, kita perlu hati-hati agar tidak memperkenalkan overfitting yang bisa meningkatkan variance model.

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!

Jawaban :

Selain MSE (Mean Squared Error), beberapa fungsi loss alternatif yang bisa digunakan antara lain:

Fungsi Loss	Kelebihan	Kekurangan	Situasi Spesifik
Mean Absolute Error (MAE)	Lebih robust terhadap outlier, tidak terlalu menghukum kesalahan besar	Gradiennya tidak kontinu pada nol, memperlambat konvergensi	Cocok untuk dataset dengan banyak outlier atau data yang memiliki distribusi tidak normal
Huber Loss	Kombinasi terbaik antara MSE dan MAE, robust terhadap outlier	Memerlukan parameter delta yang harus disesuaikan	Menguntungkan ketika ada sedikit outlier, menjaga stabilitas model
Log-Cosh Loss	Menghitung logaritma dari nilai error, lebih smooth dari MSE dan MAE	Tidak sepopuler MSE, perlu pemahaman lebih dalam	Bisa digunakan pada masalah regresi di mana model harus lebih sensitif terhadap error kecil

Untuk dataset ini, Huber Loss mungkin lebih baik, terutama jika terdapat beberapa outlier yang perlu dihindari namun masih membutuhkan 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!

Jawaban :

Fitur dengan rentang nilai yang lebih besar (misalnya 100-1000) akan memiliki gradien yang lebih besar selama proses pelatihan dibandingkan dengan fitur yang memiliki rentang lebih kecil (misalnya 0-1).

Mengapa ini terjadi?

- Gradien dihitung selama proses backpropagation dan digunakan untuk memperbarui bobot model. Gradien ini dihasilkan dari turunan fungsi loss terhadap parameter (berat).
- Jika sebuah fitur memiliki rentang yang lebih besar, perubahan kecil dalam nilai fitur tersebut akan menghasilkan perubahan besar dalam output model, yang akan memengaruhi gradien. Sebaliknya, untuk fitur dengan rentang kecil, perubahan dalam nilai fitur akan menghasilkan perubahan yang lebih kecil pada output, yang mengarah pada gradien yang lebih kecil.

Konsekuensinya:

- Bobot yang terkait dengan fitur yang memiliki rentang lebih besar cenderung diperbarui lebih besar daripada fitur yang memiliki rentang kecil. Hal ini dapat menyebabkan pembaruan bobot yang tidak seimbang. Dalam beberapa kasus, ini bisa menyebabkan model lebih "memperhatikan" fitur dengan rentang besar

dan mengabaikan fitur dengan rentang lebih kecil, padahal kedua fitur tersebut mungkin sama-sama penting.

Kesimpulan:

Perbedaan rentang fitur dapat menyebabkan masalah dalam pelatihan MLP, seperti pembaruan bobot yang tidak seimbang dan kecepatan konvergensi yang buruk. Menggunakan teknik normalisasi atau standarisasi adalah cara yang efektif untuk mengurangi masalah ini dan memastikan model dapat belajar dengan baik dari semua fitur, terlepas dari skala awal mereka.

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!

Jawaban:

Mengukur kontribusi relatif dari setiap fitur terhadap prediksi model sangat penting untuk memahami bagaimana model mengambil keputusan, serta untuk meningkatkan interpretabilitas dan transparansi model. Meskipun kita tidak tahu nama atau jenis fitur dalam dataset, ada beberapa metode yang dapat digunakan untuk mengukur kontribusi fitur tanpa informasi eksplisit mengenai fitur tersebut.

1. Permutation Importance

Permutation Importance adalah metode yang digunakan untuk menilai pentingnya fitur dengan cara mengacak (permute) nilai dari satu fitur dan mengukur seberapa besar penurunan kinerja model setelah peracikan tersebut. Konsep dasar dari metode ini adalah bahwa jika sebuah fitur memiliki kontribusi yang besar terhadap prediksi model, maka mengacak nilai dari fitur tersebut akan menyebabkan penurunan yang signifikan dalam kinerja model.

Langkah-langkahnya:

1. Model dilatih pada data asli dan menghasilkan nilai kinerja (misalnya, akurasi atau MSE).
2. Nilai fitur tertentu diacak atau dipertukarkan (permute) sehingga tidak lagi relevan.
3. Model diujikan kembali pada data yang telah dipermutasi dan nilai kinerjanya diukur.
4. Jika ada penurunan kinerja yang signifikan setelah permutasi, itu menunjukkan bahwa fitur tersebut memiliki kontribusi besar terhadap model.

Kelebihan:

- Metode ini dapat diterapkan pada model non-linier dan kompleks seperti MLP, yang sulit untuk dianalisis hanya dengan melihat bobot atau koefisien.
- Hasilnya mudah dipahami karena memberikan peringkat kontribusi setiap fitur berdasarkan pengaruhnya terhadap kinerja model.

Kekurangan:

- Proses permutasi memerlukan waktu yang cukup lama, terutama jika dataset besar atau model yang digunakan kompleks.
- Sensitif terhadap interaksi fitur; jika beberapa fitur sangat bergantung satu sama lain, perhitungan kontribusi individu fitur bisa kurang akurat.

2. Weight Analysis

Metode lain yang bisa digunakan untuk mengukur kontribusi relatif fitur adalah dengan analisis bobot (weight analysis). Pada model linear atau jaringan syaraf sederhana seperti MLP, bobot yang diberikan pada setiap fitur di layer pertama dapat memberikan gambaran tentang pentingnya fitur tersebut. Dalam MLP, bobot yang lebih besar menunjukkan bahwa fitur tersebut memiliki pengaruh yang lebih besar pada hasil prediksi model.

Langkah-langkahnya:

- Dalam MLP, kita dapat melihat bobot di layer pertama, yaitu yang menghubungkan input ke hidden layer. Bobot yang besar menunjukkan bahwa fitur terkait memiliki dampak yang signifikan pada output.
- Di sisi lain, jika bobot sangat kecil (atau mendekati nol), ini bisa menunjukkan bahwa fitur tersebut tidak banyak berkontribusi pada keputusan model.

Kelebihan:

- Mudah untuk dihitung dan diterapkan, terutama pada model sederhana atau pada model yang menggunakan fully connected layers di mana bobot dapat dilihat langsung.

Kekurangan:

- MLP dan model non-linier lainnya memiliki kesulitan dalam interpretasi bobot secara langsung karena adanya aktivasi non-linier, sehingga bobot tidak selalu mencerminkan kontribusi aktual dari fitur pada prediksi.
- Metode ini lebih terbatas jika model memiliki banyak layer atau jika terdapat banyak interaksi antara fitur yang sulit ditangkap hanya dengan melihat bobot.

Kesimpulan:

Untuk mengukur kontribusi relatif setiap fitur terhadap prediksi model tanpa mengetahui nama fitur, kita dapat menggunakan berbagai metode seperti Permutation Importance, Weight Analysis, SHAP, LIME, atau analisis residual dan visualisasi. Setiap metode memiliki kelebihan dan kekurangannya sendiri, tergantung pada jenis model yang digunakan dan tujuan analisis. Pilihan metode yang tepat akan bergantung pada kebutuhan interpretabilitas model dan tingkat kompleksitas dataset yang dihadapi.

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

Jawaban

Ketika memilih learning rate dan batch size, ada beberapa trade-off yang perlu dipertimbangkan antara komputasi dan stabilitas pelatihan:

- **Learning Rate:**
Learning rate yang terlalu tinggi dapat menyebabkan pelatihan yang tidak stabil karena model bisa melewati solusi optimal atau bahkan divergen. Namun, jika learning rate terlalu rendah, pelatihan bisa sangat lambat dan memakan banyak waktu untuk konvergen. Di sisi lain, learning rate yang lebih besar memungkinkan pelatihan lebih cepat, tetapi perlu pengaturan yang lebih hati-hati untuk memastikan model tidak melompat ke nilai yang buruk.
- **Batch Size:**
Batch size yang lebih kecil memungkinkan pembaruan bobot yang lebih sering, yang dapat membantu model untuk lebih mempelajari pola dari data dan menghindari overfitting. Namun, ini membutuhkan waktu lebih lama untuk menyelesaikan satu epoch dan lebih banyak iterasi untuk konvergensi. Di sisi lain, batch size yang lebih besar memungkinkan pembaruan yang lebih jarang, tetapi dapat mempercepat pelatihan karena mengurangi waktu yang dibutuhkan untuk setiap iterasi.

Kesimpulan

Pemilihan learning rate dan batch size yang optimal sangat penting untuk efisiensi dan efektivitas pelatihan model. Desain eksperimen untuk memilih kombinasi yang optimal melibatkan berbagai teknik, seperti grid search, random search, dan learning rate scheduling. Selain itu, pertimbangan antara komputasi dan stabilitas pelatihan harus diambil untuk memastikan bahwa model tidak hanya konvergen dengan cepat, tetapi juga dapat menggeneralisasi dengan baik pada data baru. Evaluasi dengan cross-validation dan penggunaan metode lanjutan seperti Bayesian optimization dapat lebih lanjut meningkatkan pencarian hyperparameter yang optimal.