

Pertanyaan Analisa MLP dari datasets RegresiUTSTelkom.csv

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

Jawaban & Analisa:

1. Apabila menggunakan MLP dengan 3 hidden layer (256-128-64) menghasilkan underfitting pada dataset RegresiUTSTelkom, modifikasi yang bisa dilakukan adalah:
 - a. Menambahkan jumlah neuron per layer dengan contoh menjadi (512-256-64). Hal tersebut berfungsi untuk mengurangi bias dikarenakan representasi meningkat sehingga akan cocok untuk dataset yang besar.
 - b. Menambahkan jumlah layer sehingga model dapat merepresentasikan fungsi yang lebih kompleks.
 - c. Dapat mengganti activation function yang sebelumnya ReLU menjadi LeakyReLU atau ELU sehingga bisa menghindari “dying ReLU problem” yang dapat menyebabkan underfitting.
 - d. Mengurangi regularisasi seperti dropout atau L2, dikarenakan apabila regularisasi terlalu besar, maka model yang dibuat akan menjadi terlalu kaku atau bias tinggi.

Sehingga apabila mengalami underfitting, dapat diketahui bahwa model memiliki bias tinggi, dan variance yang rendah. Sehingga untuk mengurangi bias, dapat dilakukan peningkatan kompleksitas dengan cara penambahan jumlah layer atau neuron, mengganti activation function, atau mengurangi regularisasi yang berlebihan.

Namun dengan meningkatkan kompleksitas model, dapat menyebabkan overfitting atau variance yang tinggi sehingga kuncinya adalah dengan mencari trade-off atau keseimbangan antara kapasitas model sehingga cukup untuk menangkap pola dan regularisasi sehingga cukup untuk mencegah overfitting.

2. Loss function yang mungkin cocok untuk dataset RegresiUTSTelkom adalah:
 - a. MAE (Mean Absolute Error):
 - Kelebihan:

- Lebih robust terhadap outlier dibandingkan dengan MSE dikarenakan tidak perlu mengkuadratkan selisih
 - Menghasilkan estimasi yang lebih stabil apabila data mengandung noise ekstrem.
 - Kekurangan:
 - Gradien MAE lebih konstan sehingga dapat menyebabkan training yang lebih lambat atau tidak stabil
 - Digunakan pada saat:
 - MAE akan lebih unggul dari MSE apabila dataset memiliki banyak outlier dan apabila ingin meminimalkan kesalahan absolut.
- b. Huber Loss:
- Kelebihan:
 - Merupakan kombinasi dari MSE dan MAE sehingga kuadrat untuk error lebih kecil dan absolut untuk error besar.
 - Lebih stabil terhadap outlier dibandingkan dengan MSE
 - Kekurangan:
 - Perlu menyesuaikan hyperparameter delta untuk menentukan batas transisi dari MSE ke MAE
 - Digunakan pada saat:
 - Dataset memiliki outlier namun tetap ingin menggunakan keunggulan dari MSE untuk error yang lebih kecil
- c. Log-Cosh Loss
- Kelebihan:
 - Menggabungkan MSE untuk error yang kecil dan MAE untuk error yang besar.
 - Optimasi lebih stabil dibandingkan dengan MAE.
 - Lebih robust terhadap outlier dibandingkan dengan MSE, namun menjaga sensitivitas terhadap error kecil.
 - Kekurangan:
 - Melibatkan fungsi log dan cosh sehingga sedikit lebih kompleks secara komputasi dibandingkan dengan MSE atau MAE
 - Kurang intuitif dalam interpretasi dibandingkan MSE dan MAE
 - Digunakan pada saat:
 - Datasets dengan nilai target yang sebagian besar normal, namun terdapat beberapa outlier
 - Apabila ingin model lebih stabil, namun tidak terlalu sensitif terhadap nilai ekstrem.
3. Perbedaan skala fitur antara 0-1 dengan 100-1000 adalah fitur dengan range besar (100-1000) akan menghasilkan nilai aktivasi yang lebih besar dibandingkan fitur dengan range kecil (0-1). Sehingga gradien dari fitur besar akan lebih dominan, akibatnya fitur kecil bisa diabaikan oleh model.

Dampak pada mekanisme pelatihan adalah:

- a. Forward Pass (Perhitungan Aktivasi)
Jika x besar, maka z besar sehingga aktivasi seperti ReLU bisa berada pada area ekstrem dan akan mengarah pada saturasi aktivasi yang akan menurunkan sensitivitas terhadap perubahan input.
 - b. Backward pass (Perhitungan Gradien)
Fitur besar akan menghasilkan gradien yang besar sehingga bobotnya akan diperbarui menjadi lebih besar. Fitur kecil akan menghasilkan gradien kecil sehingga bobotnya diperbarui lambat atau bahkan menjadi tidak signifikan. Sehingga kedua hal tersebut akan menyebabkan pembelajaran antar fitur menjadi tidak seimbang, maka hasil model akan bias terhadap fitur yang berskala besar.
 - c. Weight Update (Pembaruan bobot)
Apabila gradien dari fitur berskala kecil memiliki nilai yang sangat kecil, maka bobotnya hampir tidak akan berubah sehingga akan menghambat konvergensi
4. Untuk mengukur kontribusi relative setiap fitur terhadap model adalah dengan menggunakan metode:
- a. Permutation Feature Importance (PFI):
 - Cara kerja:
 - Metode akan mengacak nilai fitur satu per satu, kemudian mengukur penurunan performa model. Fitur yang kontribusinya besar akan menyebabkan penurunan performa signifikan saat diacak.
 - Kelebihan:
 - Bisa digunakan untuk semua model (MLP, CNN, dll).
 - Mengukur kontribusi nyata terhadap prediksi.
 - Keterbatasan:
 - Lambat secara komputasi terutama jika jumlah fitur besar.
 - Tidak akurat apabila fitur saling berkorelasi.
 - b. Weight Analysis:
 - Cara kerja:
 - Mengamati besar bobot yang menghubungkan input ke hidden layer.
 - Fitur dengan bobot yang lebih besar akan dianggap lebih berkontribusi
 - Kelebihan:
 - Cepat dikarenakan langsung dari arsitektur model.
 - Bisa diterapkan tanpa retraining.
 - Keterbatasan:
 - Tidak mempertimbangkan interaksi antar fitur.
 - Tidak akurat untuk jaringan yang dalam dikarenakan bobot tersebar.
 - c. SHAP (Shapley Additive exPlanations)
 - Cara kerja:
 - Mengukur kontribusi marginal fitur terhadap prediksi dengan mempertimbangkan semua kombinasi subset fitur.
 - Kelebihan:
 - Akurasi tinggi, dan adil serta mempertimbangkan interaksi.

- Memberikan kontribusi per prediksi
 - Keterbatasan:
 - Komputasi sangat mahal, terutama untuk model yang besar.
 - Implementasi kompleks jika tanpa Pustaka tambahan.
5. Eksperimen untuk memilih learning rate dan batch size secara optimal adalah dengan cara:
- a. Grid search atau Random search eksperimen:
 - Menentukan rentang nilai learning rate dan batch size, kemudian melakukan pelatihan untuk setiap kombinasi dari learning rate dan batch size. Lakukan pencatatan metrik seperti loss validasi, akurasi, waktu training, dan stabilitas loss.
 - b. Learning rate finder
 - Dimulai dari learning rate sangat kecil, dan naik eksponensial tiap batch hingga 1, perhatikan plot loss pada learning rate sehingga titik dimana loss mulai turun menjadi kandidat learning rate optimal.

Untuk analisis trade-off antara stabilitas dan komputasi pelatihan dilakukan dengan cara:

- a. Learning Rate:
 - LR kecil maka komputasi menjadi lambat dan stabilitas menjadi stagnan.
 - LR besar maka komputasi menjadi cepat per langkah dan stabilitas menjadi tidak stabil atau divergen.
 - LR optimal maka komputasi menjadi efisien dan stabilitas menjadi stabil sekaligus cepat.
- b. Batch Size:
 - Batch Size kecil maka komputasi menjadi lambat per epoch dikarenakan banyaknya step dan stabilitas menjadi loss fluktuatif.
 - Batch Size besar maka komputasi menjadi lebih cepat per epoch dan stabilitas menjadi stabil namun bisa terjadi overfit.
 - Batch Size optimal maka komputasi menjadi efisien dan stabilitas menjadi cukup stabil.

Sehingga kesimpulannya

- LR kecil: stabil, namun mahal dari segi waktu
- LR besar: lebih cepat namun beresiko tidak stabil
- Batch Size kecil: lebih akurat dan generalisasi bagus, namun lambat dan noisy
- Batch Size besar: lebih cepat, loss lebih halus, namun bisa terjadi overfit dan boros memori.