

## Zero to Mastery Learn PyTorch for Deep Learning Course Summary

### 02. PyTorch Neural Network Classification

Nama : Ihsan Ridho Asmoro

Nim : 1103204184

Dalam modul ini, kita akan belajar dasar-dasar masalah klasifikasi dalam machine learning. Klasifikasi itu seperti menebak, apakah sesuatu termasuk dalam satu kelompok atau tidak. Ada tiga jenis masalah klasifikasi utama: yang hanya punya dua kategori, yang punya banyak kategori, dan yang bisa masuk ke beberapa kategori sekaligus.

Kita juga akan lihat bagaimana jaringan saraf bekerja untuk menyelesaikan masalah klasifikasi. Sebelum masuk ke koding, kita akan bahas bagaimana struktur jaringan saraf untuk klasifikasi, seperti jumlah lapisan, neuron per lapisan, dan jenis aktivasi yang digunakan.

Langkah pertama adalah mempersiapkan data untuk klasifikasi. Kita akan buat data yang berisi dua lingkaran dengan warna berbeda. Setelah itu, kita tampilkan datanya dalam bentuk scatter plot untuk melihat pola distribusi datanya.

Setelah data siap, kita ubah bentuknya agar sesuai dengan kebutuhan dan konversi menjadi tensor untuk digunakan oleh PyTorch. Selanjutnya, kita bagi data menjadi dua bagian: 80% untuk pelatihan dan 20% untuk pengujian.

Selanjutnya, kita bangun modelnya. Kode yang kita buat harus bisa berjalan baik di CPU maupun GPU. Kita buat modelnya dengan menggunakan dua lapisan linear untuk menangani input dan output data.

Kemudian, kita atur fungsi-fungsi penting seperti fungsi forward untuk menghitung langkah maju model, fungsi kerugian yang digunakan untuk melatih model, dan optimizer untuk mengoptimalkan parameter model.

Setelah model selesai, kita mulai proses pelatihan. Kita konfirmasi terlebih dahulu bahwa model kita bisa menghasilkan label prediksi yang sesuai. Setelah itu, kita bangun loop pelatihan dan pengujian. Pada setiap epoch, model diperbarui menggunakan optimizer, dan hasil pelatihan dan pengujian dicetak setiap 10 epoch untuk memantau perkembangan.

Meskipun model sudah melalui pelatihan, performanya masih belum memuaskan. Akurasi stagnan di sekitar 50%, menandakan bahwa model ini belum lebih baik dari prediksi acak. Kita perlu melakukan evaluasi lebih lanjut dan mungkin melakukan penyesuaian untuk meningkatkan kinerja model pada tugas klasifikasi ini.

Selanjutnya, kita melakukan evaluasi awal dengan menggunakan metrik-metrik yang telah diperoleh. Hasilnya menunjukkan bahwa model cenderung melakukan prediksi acak. Untuk mendapatkan pemahaman yang lebih dalam, kita lakukan analisis visual dengan membuat decision boundary. Decision boundary ini menunjukkan bahwa model cenderung memisahkan data dengan garis lurus, yang tidak sesuai dengan distribusi data kita yang berbentuk lingkaran. Hal ini menjelaskan performa rendah model, karena model hanya bisa memotong data menjadi dua bagian sejajar tanpa dapat menangkap pola yang sesuai.

Kemudian, kita identifikasi bahwa kondisi ini adalah underfitting, di mana model tidak mampu menangkap pola yang ada dalam data. Perbaikan model dilakukan dengan menambah kompleksitas model atau menerapkan teknik-teknik khusus.

Untuk meningkatkan model dari perspektif model, kita coba pendekatan dengan menambah lapisan dan unit tersembunyi, serta melatih model untuk lebih lama. Meskipun kita telah membuat model yang lebih kompleks, hasilnya masih belum menunjukkan peningkatan yang signifikan.

Selanjutnya, kita eksplorasi apakah model kita gagal karena tidak mampu memodelkan data lingkaran dengan baik. Untuk menguji ini, kita buat data linier dan latih model untuk melihat apakah bisa memodelkan data tersebut. Hasilnya menunjukkan bahwa model yang dirancang untuk menangani data linier lebih baik daripada model sebelumnya yang hanya mampu memodelkan data lingkaran.

Kemudian, kita menyadari bahwa kita perlu memperkenalkan non-linearitas ke dalam model untuk mengatasi masalah data lingkaran. Kita tambahkan fungsi aktivasi non-linear, seperti ReLU, dan latih model dengan data baru yang berisi lingkaran merah dan biru. Evaluasi model menunjukkan peningkatan, dan model mampu memahami pola non-linear dari data.

Untuk memahami lebih dalam apa yang dilakukan oleh fungsi aktivasi non-linear, kita replikasi beberapa fungsi aktivasi, seperti ReLU dan Sigmoid, dan melihat visualisasinya. ReLU mengubah nilai negatif menjadi 0 dan membiarkan nilai positif tidak berubah, sedangkan Sigmoid menghasilkan nilai antara 0 dan 1.

Setelah itu, kita menggabungkan semua konsep dalam model PyTorch untuk klasifikasi multi-kelas. Kita buat data multi-kelas menggunakan metode `make_blobs()` dan bangun model yang mampu menangani data multi-kelas. Fungsi loss yang digunakan adalah `nn.CrossEntropyLoss()`, dan kita tetap menggunakan SGD untuk mengoptimalkan parameter model.

Sebelum melatih model, kita lakukan langkah maju sekali untuk melihat apakah semuanya berfungsi. Output yang kita dapatkan disebut logits, dan kita konversi logits tersebut menjadi probabilitas prediksi menggunakan softmax.

Setelah semua persiapan selesai, kita tulis loop pelatihan dan pengujian untuk meningkatkan dan mengevaluasi model. Kita konversi output model (logits) menjadi probabilitas prediksi dan kemudian menjadi label prediksi. Setelah itu, kita membuat beberapa prediksi dan memvisualisasikannya, serta mengevaluasi akurasi.

Terakhir, kita bahas lebih banyak metrik evaluasi klasifikasi yang bisa digunakan, seperti presisi, recall, F1-score, confusion matrix, dan classification report. Metrik-metrik ini memberikan wawasan lebih rinci tentang kinerja model klasifikasi.

Semua langkah-langkah ini memberikan dasar yang kokoh untuk memahami dan mengembangkan model klasifikasi dalam machine learning menggunakan PyTorch.