Nama : Arfara Yema Samgusdian
NIM : 1103202004
Kelas : TK-44-04

Zero to Mastery Learn PyTorch for Deep Learning

03. PyTorch Computer Vision

Pada bagian ini, kita akan menerapkan alur kerja PyTorch pada masalah computer vision. Fokusnya akan pada penggunaan beberapa library PyTorch khusus untuk computer vision. Rencananya melibatkan langkah-langkah seperti mengimpor library, mempersiapkan dataset FashionMNIST, membangun model dasar untuk klasifikasi multi-kelas, membuat prediksi, mengevaluasi model, menyiapkan kode yang bersifat perangkat-agnostik, dan kemudian meningkatkan model dengan menambahkan lapisan non-linear dan menggunakan Convolutional Neural Network (CNN). Selain itu, kita akan membandingkan model yang dibangun, mengevaluasi model terbaik, membuat matriks kebingungan, dan menyimpan model yang paling baik performanya.

Penting untuk diingat bahwa computer vision digunakan dalam berbagai konteks, seperti di aplikasi kamera ponsel, mobil modern untuk menghindari tabrakan, pabrik untuk mendeteksi cacat produk, dan kamera keamanan untuk mendeteksi intruder. Dengan mengggunakan library PyTorch seperti torchvision, kita dapat dengan mudah mengakses dataset, model arsitektur, dan transformasi gambar yang umumnya digunakan dalam masalah computer vision.

kita mulai bekerja dengan dataset FashionMNIST dalam konteks computer vision. FashionMNIST berisi gambar berwarna abu-abu dari 10 jenis pakaian yang berbeda, menjadikannya masalah klasifikasi multi-kelas. Langkah pertama adalah mendownload dataset menggunakan torchvision.datasets.FashionMNIST() dengan parameter seperti root folder, train/test split, dan transformasi data. Setelah itu, kita menjelajahi bentuk input dan output dari gambar, melihat jumlah sampel, serta melihat kelas-kelas pakaian yang ada.

Selanjutnya, kita melakukan visualisasi data dengan menampilkan salah satu gambar dari dataset menggunakan matplotlib. Setelah mempersiapkan dataset, langkah selanjutnya adalah menggunakan DataLoader untuk mempermudah proses loading data ke dalam model. DataLoader membagi dataset menjadi batch atau mini-batch, meningkatkan efisiensi komputasi dan memberikan model lebih banyak kesempatan untuk memperbarui parameter dengan lebih seringnya gradient descent.

kita membangun model dasar (baseline model) dengan menggunakan nn.Flatten() sebagai lapisan pertama, yang berfungsi untuk meratakan dimensi tensor ke dalam vektor fitur. Model ini akan menjadi dasar untuk model-model berikutnya yang lebih kompleks.

Pada bagian ini, kita melakukan serangkaian langkah untuk mengevaluasi beberapa model yang telah kita bangun. Pertama, kita membuat sebuah fungsi yang dapat menerima model yang telah dilatih, DataLoader, fungsi loss, dan fungsi akurasi untuk melakukan prediksi menggunakan model tersebut. Selanjutnya, kita menyiapkan kode yang dapat berjalan pada perangkat yang tersedia, apakah itu CPU atau GPU.

Kemudian, kita membangun dua model tambahan, yaitu model_1 dengan penambahan fungsi non-linear (nn.ReLU()) dan model_2 dengan struktur Convolutional Neural Network (CNN). Model CNN yang digunakan mengikuti arsitektur TinyVGG.

Setelah semua model dilatih, kita membandingkan hasilnya, termasuk waktu pelatihan untuk masing-masing model. Dengan membandingkan performa dan waktu pelatihan, kita dapat memilih model terbaik.

Langkah selanjutnya adalah membuat dan mengevaluasi prediksi acak menggunakan model terbaik. Kita juga membuat sebuah matriks kebingungan (confusion matrix) untuk melihat di mana model mengalami kesulitan dalam mengklasifikasikan prediksi.

Seluruh proses ini dilakukan dengan menggunakan PyTorch, dan hasilnya divisualisasikan untuk memberikan pemahaman yang lebih baik tentang kinerja model.