

Zero to Mastery Learn PyTorch for Deep Learning Course Summary

03. PyTorch Computer Vision

Nama : Ihsan Ridho Asmoro

NIM : 1103204184

Visi komputer adalah bagaimana kita mengajarkan komputer untuk melihat dunia. Misalnya, kita bisa membuat model untuk mengenali apakah suatu gambar berisi kucing atau anjing (klasifikasi biner), atau bahkan mengenali apakah gambar tersebut berisi kucing, anjing, atau ayam (klasifikasi multi-kelas). Selain itu, visi komputer juga termasuk dalam mengidentifikasi di mana mobil muncul dalam video atau memahami bagaimana objek berbeda dalam gambar dapat dibedakan.

Langkah pertama: Mengunduh Dataset FashionMNIST

- FashionMNIST adalah kumpulan data visi komputer berisi gambar pakaian berwarna abu-abu dari 10 jenis pakaian.
- Dataset ini diunduh menggunakan PyTorch's `torchvision.datasets.FashionMNIST()` dengan beberapa parameter.

1.1 Representasi Input dan Output Model Komputer Visi

- Gambar direpresentasikan sebagai tensor dengan bentuk `[jumlah_channel_warna, tinggi, lebar]`.
- `Jumlah_channel_warna=1` menunjukkan gambar grayscale, dan untuk RGB, `jumlah_channel_warna=3`.
- Representasi CHW (Color Channels, Height, Width) atau HWC (Color Channels Last) digunakan.

1.2 Visualisasi Data

- Beberapa contoh dari dataset dapat divisualisasikan menggunakan `matplotlib`.
- Terdapat 60,000 sampel pelatihan dan 10,000 sampel pengujian.
- Terdapat 10 kelas pakaian untuk klasifikasi multi-kelas.

1. Persiapan DataLoader

- `DataLoader` membantu memuat data ke dalam model, baik selama pelatihan maupun inferensi.
- Batch atau mini-batch digunakan untuk efisiensi komputasional.
- Kode disediakan untuk membuat `DataLoader` dengan batch size tertentu.

2. Model 0: Membangun Model Dasar

- Fungsi-fungsi pelatihan, pengujian, loss, optimizer, dan metrik evaluasi disiapkan untuk tugas klasifikasi.
- Loop pelatihan dan pelatihan model pada batch data diimplementasikan.

3. Membuat Prediksi dan Mendapatkan Hasil Model 0

- Prediksi dilakukan menggunakan model 0, dan hasilnya ditampilkan.

4. Menyiapkan Kode Agnostik Perangkat (Device Agnostic)
 - Kode disediakan untuk memungkinkan penggunaan GPU jika tersedia atau menggunakan CPU jika tidak.
5. Model 1: Membangun Model dengan Non-Linearitas
 - Model 1 dibuat dengan menambahkan fungsi non-linear di antara lapisan-lapisan linear.
6. Model 2: Membangun Convolutional Neural Network (CNN)
 - CNN dibuat untuk meningkatkan kinerja model dalam menangani data visual.
7. Membandingkan Hasil Model dan Waktu Pelatihan
 - Tiga model berbeda (model_0, model_1, model_2) dilatih dan hasilnya disatukan untuk perbandingan.
 - Waktu pelatihan juga dicatat.
8. Membuat dan Menilai Prediksi Acak dengan Model Terbaik
 - Model terbaik (model_2) dipilih dan digunakan untuk membuat prediksi pada sampel uji.
9. Membuat Matriks Konfusi untuk Evaluasi Lebih Lanjut
 - Matriks konfusi digunakan sebagai metode evaluasi visual untuk memeriksa seberapa baik model memprediksi kelas.
10. Menyimpan dan Memuat Model Terbaik
 - Model terbaik disimpan dan kemudian dimuat kembali untuk memastikan hasilnya konsisten setelah penyimpanan.