

Nama: Guntur Eka Wardana

NIM: 1103174102

Technical Report: PyTorch Computer Vision

Pendahuluan

Dalam bagian ini, kita akan mengimplementasikan alur kerja PyTorch pada masalah computer vision dengan fokus pada pemanfaatan beberapa library PyTorch khusus untuk domain ini. Rencananya melibatkan langkah-langkah seperti impor library, persiapan dataset FashionMNIST, konstruksi model dasar untuk klasifikasi multi-kelas, prediksi, evaluasi model, kode perangkat-agnostik, dan peningkatan model dengan lapisan non-linear serta penggunaan Convolutional Neural Network (CNN). Pada akhirnya, kita akan membandingkan model, mengevaluasi model terbaik, membuat matriks kebingungan, dan menyimpan model dengan performa terbaik.

Penggunaan Library PyTorch dalam Computer Vision

Pertama-tama, kita akan memahami dasar-dasar penggunaan PyTorch dalam konteks computer vision. Modul-modul khusus seperti torchvision akan menjadi fokus utama, mempermudah proses pengolahan data gambar dan pengembangan model.

Persiapan Dataset FashionMNIST

Kita akan memulai dengan bekerja pada dataset FashionMNIST, yang berisi gambar berwarna abu-abu dari 10 jenis pakaian berbeda. Langkah awal melibatkan pengunduhan dataset menggunakan **`torchvision.datasets.FashionMNIST()`**, dengan pengaturan root folder, train/test split, dan transformasi data. Setelahnya, kita akan menjelajahi bentuk input dan output gambar, serta melihat kelas-kelas pakaian yang ada.

Visualisasi Data dan Persiapan DataLoader

Visualisasi data dilakukan dengan menampilkan satu gambar dari dataset menggunakan matplotlib. Setelah persiapan dataset, kita akan 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.

Model Dasar dan Pengembangan Model

Model dasar (Model 0) dibangun dengan menggunakan **`nn.Flatten()`** sebagai lapisan pertama, meratakan dimensi tensor ke dalam vektor fitur. Model ini akan menjadi dasar untuk model-model berikutnya yang lebih kompleks. Langkah selanjutnya adalah mengembangkan dua model tambahan: model_1 dengan penambahan fungsi non-linear (**`nn.ReLU()`**), dan model_2 dengan struktur Convolutional Neural Network (CNN) mengikuti arsitektur TinyVGG.

Evaluasi Model

Proses evaluasi melibatkan serangkaian langkah, termasuk pembuatan fungsi untuk prediksi menggunakan model yang telah dilatih, penyediaan kode yang dapat berjalan pada perangkat yang tersedia (CPU atau GPU), dan membandingkan hasil serta waktu pelatihan dari beberapa model. Selanjutnya, prediksi acak dilakukan menggunakan model terbaik, dan matriks kebingungan dibuat untuk menganalisis kesulitan model dalam mengklasifikasikan prediksi.

Kesimpulan

Keseluruhan proses ini dilakukan dengan menggunakan PyTorch, menekankan keterlibatan library seperti torchvision untuk mempermudah tugas-tugas spesifik dalam computer vision. Analisis hasil dan visualisasi memberikan pemahaman yang lebih baik tentang kinerja model. Dengan langkah-langkah ini, kita dapat membangun, mengoptimalkan, dan mengevaluasi model untuk tugas computer vision dengan efisien menggunakan PyTorch.