Nama: Rizki Hidayat

NIM: 1103202131

Laporan

1.Pendahuluan

Modul ini bertujuan untuk mengimplementasikan dan mengevaluasi model pembelajaran mesin menggunakan kerangka kerja PyTorch pada dataset FashionMNIST. Tujuan utama mencakup persiapan data, pembangunan model dasar, dan model yang lebih baik dengan non-linearitas, serta pengujian kinerja pada perangkat GPU. Dan juga implementasi dan analisis model Convolutional Neural Network (CNN)

2.Penjelasan

2.1 Mempersiapkan data

Modul ini menggunakan dataset FashionMNIST dan menggunakan PyTorch's DataLoader untuk membentuk set pelatihan dan pengujian.

2.2 Build Model Dasar (FashionMNISTV0)

membuat model dasar (FashionMNISTModelV0) dengan lapisan-lapisan linear dan menggunakan fungsi aktivasi ReLU pada model pertama.

2.3 Training Model 0

Model 0 menggunakan loop pelatihan, mengukur waktu pelatihan, dan mengevaluasi kinerja model pada perangkat CPU.

2.4 Model 1: Pembangunan Model dengan Non-Linearitas

Di modul ini membuat model kedua (FashionMNISTModelV1) yang memasukkan fungsi aktivasi ReLU pada lapisan-lapisan untuk mengeksplorasi non-linearitas.

2.5 Pelatihan Model 1 pada Perangkat GPU:

Pada modul ini mengubah pelatihan model kedua ke perangkat GPU jika tersedia dan mengevaluasi kinerja model pada perangkat GPU.

2.6 Evaluasi Model 1

mengukur kinerja model dengan membandingkan hasil prediksi dan label sebenarnya pada dataset pengujian.

2.7 Model 2: Membangun Convolutional Neural Network (CNN)

Membuat model 2 dengan menggunakan CNN, diamana model ini memiliki dua blok konvolusi (block_1 dan block_2) yang masing-masing terdiri dari dua layer konvolusi, fungsi aktivasi ReLU, dan max-pooling.

2.8 Inisiasi pada Model 2

Model 2 diinisialisasi dengan parameter tertentu, seperti input_shape, hidden_units, dan output_shape.

2.9 Eksplorasi nn.Conv2d() dan nn.MaxPool2d():

Pada notebook ini diberikan Contoh penggunaan nn.Conv2d() dengan variasi parameter. Dan juga demonstrasi efek nn.MaxPool2d() terhadap ukuran tensor hasil.

2.10 Training model Fungsi Loss dan Optimizer:

Fungsi loss (nn.CrossEntropyLoss()) dan optimizer (torch.optim.SGD()) ditentukan untuk melatih model.

2.11 Pelatihan dan Evaluasi Model:

Model dilatih dan dievaluasi selama beberapa epoch menggunakan data pelatihan dan pengujian. Waktu pelatihan diukur untuk evaluasi efisiensi pelatihan model.

2.12 Perbandingan Hasil dan Waktu Pelatihan:

Hasil pelatihan dari beberapa model (model_0, model_1, model_2) dibandingkan, termasuk akurasi dan waktu pelatihan.

2.13Visualisasi Hasil Model

Hasil model, seperti akurasi, direpresentasikan dalam bentuk diagram batang.

2.14 Prediksi Acak dan Evaluasi

Model terbaik (model_2) digunakan untuk membuat prediksi acak pada beberapa sampel uji.

Prediksi dan label sebenarnya ditampilkan dalam plot.

2.15 Confusion Matrix

Confusion matrix dibuat untuk evaluasi lebih lanjut terhadap prediksi model terbaik.

2.16 Penyimpanan dan Pemulihan Model

Model terbaik (model_2) disimpan sebagai file .pth menggunakan torch.save().

Model kemudian diinisialisasi kembali menggunakan FashionMNISTModelV2 dan torch.load().

Model yang dimuat dievaluasi untuk memastikan konsistensi hasil dengan model asli.

2.17 Perbandingan Hasil Model Asli dan Model yang Dimuat

Hasil dari model asli (model_2_results) dibandingkan dengan hasil dari model yang dimuat (loaded_model_2_results) untuk memastikan kesamaan.

2.18 Evaluasi Kesamaan Hasil Model

Kesamaan hasil model dievaluasi menggunakan torch.isclose() dengan toleransi tertentu.

2.19 Visualisasi dan Pembandingan Hasil Model Asli dan Model yang Dimuat

Hasil akhir dari model asli dan model yang dimuat disajikan dan dibandingkan.