

**Nama : Arfara Yema Samgusdian**

**NIM : 1103202004**

**Kelas : TK-44-04**

## **Zero to Mastery Learn PyTorch for Deep Learning**

### **07. PyTorch Experiment Tracking**

Pengertian Pelacakan Eksperimen

Machine learning dan deep learning bersifat sangat eksperimental. Dalam proses ini, Anda perlu mengenakan "beret seniman/topi koki" untuk menciptakan berbagai model dan "jas ilmiah" untuk melacak hasil dari berbagai kombinasi data, arsitektur model, dan rezim pelatihan. Di sinilah pelacakan eksperimen menjadi penting.

Mengapa Melacak Eksperimen?

Jika Anda menjalankan banyak eksperimen, pelacakan eksperimen membantu Anda memahami apa yang berhasil dan apa yang tidak. Meskipun memadai untuk melacak hasil beberapa model dengan dicetak dan beberapa kamus jika hanya menjalankan beberapa model, pendekatan sederhana ini dapat sulit diakses ketika jumlah eksperimen meningkat.

Berbagai Metode untuk Melacak Eksperimen Machine Learning

Ada berbagai metode untuk melacak eksperimen machine learning, dan beberapa di antaranya disajikan dalam tabel berikut

Metode	Setup	Kelebihan	Kekurangan	Biaya
Python dictionaries, CSV files, print outs	Tidak ada	Mudah diatur, berjalan dalam Python murni	Sulit melacak jumlah eksperimen yang besar	Gratis
TensorBoard	Minimal, instal TensorBoard	Dibangun ke dalam PyTorch, diakui dan digunakan secara luas, mudah diukur	Pengalaman pengguna tidak sebaik opsi lain	Gratis
Weights & Biases Experiment Tracking	Minimal, instal Weights & Biases (wandb), buat akun	Pengalaman pengguna luar biasa, dapat membuat eksperimen publik, melacak hampir semua hal	Memerlukan sumber daya eksternal di luar PyTorch	Gratis untuk penggunaan pribadi
MLFlow	Minimal, instal MLFlow dan mulai pelacakan	Manajemen siklus hidup MLOps sepenuhnya open-source, banyak integrasi	Agak sulit mengatur server pelacakan jarak jauh dibandingkan dengan layanan lain	Gratis

## 0. Persiapan Awal

Dimulai dengan mengunduh semua modul yang diperlukan untuk bagian ini. Gunakan beberapa skrip Python seperti `data_setup.py` dan `engine.py` yang telah dibuat sebelumnya di bagian 05. PyTorch Going Modular. Unduh direktori `going_modular` dari repositori `pytorch-deep-learning` (jika belum ada) dan peroleh paket `torchinfo` jika belum tersedia.

## 1. Memperoleh Data

Sebelum menjalankan eksperimen machine learning, kita perlu memiliki dataset. Lanjutkan dengan mencoba meningkatkan hasil pada FoodVision Mini dengan mendownload dataset `pizza_steak_sushi.zip`. Kode untuk ini sudah difungsikan untuk digunakan kembali.

## 2. Membuat Dataset dan DataLoader

Setelah mendapatkan data, ubah data tersebut menjadi `DataLoader` PyTorch menggunakan fungsi `create_dataloaders()` yang telah dibuat di bagian 05. PyTorch Going Modular bagian 2. Buat transformasi untuk menyiapkan gambar sesuai dengan kebutuhan model kita.

## 3. Memperoleh Model Terlatih dan Membekukan Layer Dasar

Unduh berat model terlatih untuk model `torchvision.models.efficientnet_b0()`. Persiapkan model tersebut untuk digunakan dengan data kita sendiri.

## 4. Melatih Model dan Melacak Hasil

Persiapkan model untuk pelatihan dengan membuat fungsi kerugian dan pengoptimal. Gunakan fungsi `torch.nn.CrossEntropyLoss()` sebagai fungsi kerugian dan `torch.optim.Adam()` sebagai pengoptimal.

## 5. Melihat Hasil Model di TensorBoard

Gunakan `SummaryWriter()` untuk menyimpan hasil model dalam format TensorBoard. TensorBoard adalah program visualisasi yang dapat digunakan untuk melihat dan memeriksa informasi tentang model dan data. Visualisasikan hasil model dengan membuka TensorBoard.

## 6. Membuat Fungsi Pembantu untuk Membangun Instance SummaryWriter()

Buat fungsi pembantu bernama `create_writer()` untuk membuat instance `SummaryWriter()` yang dapat melacak hasil ke direktori kustom untuk setiap eksperimen. Direktori ini mencakup tanggal/waktu eksperimen, nama eksperimen, nama model, dan informasi tambahan.

## 7. Menyiapkan Serangkaian Eksperimen Pemodelan

Mari tingkatkan sedikit.

Sebelumnya, kita menjalankan berbagai eksperimen dan memeriksa hasilnya satu per satu.

Tetapi bagaimana jika kita dapat menjalankan beberapa eksperimen sekaligus dan kemudian memeriksa hasilnya secara bersamaan?

Mari kita lakukan.

### 7.1 Jenis Eksperimen Apa yang Harus Anda Jalankan?

Itu adalah pertanyaan sejuta dolar dalam machine learning.

Karena sebenarnya tidak ada batasan untuk eksperimen yang dapat Anda jalankan.

Kebebasan ini adalah mengapa machine learning begitu menarik dan menakutkan pada saat yang bersamaan.

Di sinilah Anda harus mengenakan mantel ilmuwan dan mengingat moto praktisi machine learning: eksperimen, eksperimen, eksperimen!

Setiap hiperparameter merupakan titik awal untuk eksperimen yang berbeda:

Mengubah jumlah epoch.

Mengubah jumlah lapisan/unit tersembunyi.

Mengubah jumlah data.

Mengubah tingkat pembelajaran.

Mencoba berbagai jenis augmentasi data.

Memilih arsitektur model yang berbeda.

Dengan berlatih dan menjalankan berbagai eksperimen, Anda akan mulai membangun intuisi tentang apa yang mungkin membantu model Anda.

## 9. Memuat Model Terbaik dan Melakukan Prediksi

Melihat log TensorBoard untuk delapan eksperimen kita, tampaknya eksperimen nomor delapan mencapai hasil terbaik secara keseluruhan (akurasi uji tertinggi, kerugian uji kedua terendah).

Ini adalah eksperimen yang menggunakan:

EffNetB2 (dua kali lipat parameter EffNetB0)

Data pelatihan 20% pizza, steak, sushi (dua kali lipat data pelatihan asli)

10 epoch (dua kali lipat waktu pelatihan asli)

Pada dasarnya, model terbesar kita mencapai hasil terbaik.

Namun, tidak seperti hasil ini jauh lebih baik dari model lain.

Model yang sama pada data yang sama mencapai hasil serupa dalam setengah waktu pelatihan (eksperimen nomor 6).

Ini menunjukkan bahwa, potensialnya, bagian-bagian yang paling berpengaruh dari eksperimen kita adalah jumlah parameter dan jumlah data.

Menginspeksi hasil lebih lanjut, tampaknya secara umum model dengan lebih banyak parameter (EffNetB2) dan lebih banyak data (20% data pelatihan pizza, steak, sushi) memberikan performa lebih baik (kerugian uji lebih rendah dan akurasi uji lebih tinggi).

Lebih banyak eksperimen dapat dilakukan untuk menguji ini lebih lanjut, tetapi untuk sekarang, mari impor model terbaik dari eksperimen delapan (disimpan di: `models/07_effnetb2_data_20_percent_10_epochs.pth`, Anda dapat mengunduh model ini dari GitHub kursus) dan melakukan evaluasi kualitatif