

Nama: Luthfiah Maulidya

NIM: 1103223076

Mata Kuliah: Deep Learning

A. Inisialisasi dan Pembuatan Tensor

Langkah pertama dalam praktikum ini adalah mempelajari berbagai cara membuat tensor:

- **Directly from data:** Mengonversi list atau array menjadi tensor menggunakan `torch.tensor()`.
- **From NumPy:** Menggunakan `torch.from_numpy()` yang memungkinkan berbagi memori antara CPU dan GPU.
- **Random & Constant:** Membuat tensor dengan nilai acak (`torch.rand()`) atau nilai konstan (`torch.ones()`, `torch.zeros()`) untuk inisialisasi bobot (weight) awal dalam model.

B. Atribut Tensor

Setiap tensor memiliki atribut penting yang menentukan kinerjanya dalam komputasi:

- **Shape:** Dimensi dari tensor (misal: 3×4).
- **Datatype:** Tipe data (seperti float32 atau int64).
- **Device:** Menunjukkan apakah tensor berada di **CPU** atau **GPU (CUDA)**. Pemindahan tensor ke GPU sangat krusial untuk mempercepat pelatihan model besar.

C. Operasi Aritmatika dan Matriks

Operasi yang dilakukan mencakup:

- **Slicing & Indexing:** Memilih bagian tertentu dari tensor, serupa dengan teknik pada NumPy.
- **Concatenation:** Menggabungkan beberapa tensor menggunakan `torch.cat()`.
- **Matrix Multiplication:** Menggunakan fungsi `matmul` atau operator `@`. Ini adalah operasi inti dalam perhitungan *Forward Pass* pada Neural Network.
- **In-place Operations:** Operasi yang menggunakan akhiran `_` (seperti `add_()`) untuk menghemat penggunaan memori dengan cara langsung mengubah nilai asli tensor tersebut.

D. Bridge dengan NumPy

Salah satu fitur utama PyTorch adalah kemampuannya untuk berinteraksi dengan ekosistem Python lainnya. Praktikum menunjukkan bahwa perubahan pada tensor di CPU akan tercermin pada array NumPy yang terhubung, karena keduanya berbagi lokasi memori yang sama.

3. Kesimpulan

Berdasarkan praktikum yang telah dilakukan, dapat disimpulkan bahwa:

1. PyTorch menyediakan fleksibilitas tinggi dalam memanipulasi data melalui struktur Tensor.
2. Pemahaman mengenai dimensi (*shape*) sangat penting untuk menghindari error saat melakukan operasi perkalian matriks dalam Deep Learning.
3. Kemampuan PyTorch untuk berpindah dari CPU ke GPU memungkinkan pemrosesan data dalam skala besar secara efisien.