

Nama: Dilara Kynta Putri Raflita
NIM: 1103204059
Kelas: TK44G4

00 Pytorch Fundamentals (Dasar Dasar PyTorch)

PyTorch merupakan suatu kerangka kerja (framework) sumber terbuka untuk machine learning dan deep learning. PyTorch juga digunakan untuk memanipulasi dan memproses data serta menulis algoritma pembelajaran mesin menggunakan kode Python.

Pada chapter 00, kita mempelajari :

1. Pengantar tensor: blok bangunan dasar dari semua machine learning
2. Bagaimana membuat tensor: tensor mewakili hampir semua jenis data
3. Bagaimana mendapatkan informasi dari tensor: memasukkan informasi ke dalam tensor
4. Mempelajari operasi tensor seperti menambah, mengalikan, dan menggabungkan
5. Berurusan dengan bentuk tensor: adanya ketidakcocokan bentuk tensor
6. Pengindeksan pada tensor: memberi indeks pada daftar python atau array NumPy
7. Mencampur tensor PyTorch dan NumPy
8. Menjalankan tensor pada GPU

01 Pytorch Workflow Fundamentals (Dasar-dasar Alur Kerja PyTorch)

Pada chapter 01, kita membahas langkah-langkah standar dalam menggunakan PyTorch untuk melakukan prediksi sederhana. Keseluruhan langkah-langkah proses dari persiapan data hingga penggunaan model untuk membuat prediksi. Hal yang dilakukan yaitu:

1. Persiapan data: data bisa berupa apa saja, tetapi untuk memulai, menggunakan contoh sederhana dengan membuat prediksi garis lurus
2. Membangun model: disini kita akan membuat suatu model yang dapat mempelajari pola dalam data. Selain itu, kita juga akan memilih fungsi kerugian (loss function), pengoptimal (optimizer), dan membangun loop pelatihan

3. Pelatihan model: selanjutnya kita akan menggunakan model yang telah dilatih untuk membuat prediksi dan mengevaluasi hasilnya. Pada tahap ini disebut tahap inferensi, dimana kita membandingkan hasil prediksi dengan data aktual untuk melihat seberapa baik model bekerja.
4. Menyimpan dan memuat model: Model akan disimpan dan dimuat sehingga dapat digunakan kembali nanti

02 Neural Network Classification Pytorch

Pada chapter 02, kita akan mempelajari klasifikasi neural network pada PyTorch dimana kita mencoba untuk menebak atau memprediksi sebuah objek sehingga kita dapat mengetahui apakah sesuatu itu termasuk dalam satu kategori yang lain. Untuk klasifikasi ada beberapa jenis:

1. Klasifikasi biner: seperti menebak ya atau tidak. Contoh ketika ingin menebak apakah seseorang memiliki penyakit jantung berdasarkan data kesehatan mereka, mereka hanya bisa menjawab ya atau tidak.
2. Klasifikasi multi-kelas: memilih lebih dari dua pilihan. Misalnya ingin memutuskan apakah suatu gambar adalah makanan, orang, atau anjing.
3. Klasifikasi multi-label: seperti memberikan label pada beberapa opsi sekaligus. Contoh ketika ingin memberi kategori apa yang sesuai untuk suatu artikel. Artikel bisa masuk ke dalam kategori matematika, sains, dan filsafat sekaligus.

Secara khusus, apa yang dibahas pada chapter ini yaitu:

1. Arsitektur jaringan saraf klasifikasi
2. Menyiapkan data klasifikasi biner
3. Membangun model klasifikasi PyTorch
4. Menyesuaikan model dengan data (pelatihan)
5. Membuat prediksi dan mengevaluasi model (inferensi)
6. Meningkatkan model (dari perspektif model)
7. Non-linearitas
8. Mereplikasi fungsi non-linear

9. Menyatukan semuanya dengan klasifikasi multi-kelas

03 PyTorch Computer Vision

Dalam pembelajaran visi komputer menggunakan PyTorch, kita akan mempelajari mengenai pemahaman dasar tentang apa itu visi komputer dan bagaimana mengaplikasikannya dalam membaca dan memahami informasi dari sebuah gambar.

Pada chapter ini, beberapa hal yang dibahas yaitu:

1. Library computer vision di PyTorch: mengimport semua pustaka library visi komputer yang digunakan
2. Manipulasi dan representasi gambar: mengolah dan mempresentasikan gambar menggunakan tensor, yang merupakan struktur data dasar dalam PyTorch
3. Pembuatan dataset dan dataloader: menyiapkan dataset gambar untuk pelatihan model, serta menggunakan data loader untuk mempermudah proses pengambilan data
4. Arsitektur jaringan saraf untuk visi komputer: pemahaman tentang jenis-jenis arsitektur jaringan saraf yang umum digunakan dalam tugas visi komputer, seperti Convolutional Neural Network (CNNs)
5. Pelatihan model: proses melatih model menggunakan dataset gambar, dengan mengoptimalkan parameter agar model dapat memahami pola dan fitur dalam gambar
6. Evaluasi dan validasi model: mengukur kinerja model pada data yang tidak digunakan selama pelatihan, untuk memastikan model mampu melakukan prediksi yang baik pada data baru.
7. Transfer learning: menggunakan pengetahuan yang telah dipelajari oleh model dari tugas sebelumnya untuk meningkatkan kinerja model pada tugas visi komputer yang baru
8. Deteksi objek: memahami cara mendeteksi dan mengenali objek dalam gambar, termasuk penggunaan bounding boxes untuk menandai lokasi objek.
9. Segmentasi gambar: pemahaman tentang teknik segmentasi gambar untuk memisahkan dan mengidentifikasi area tertentu dalam gambar

10. Generasi gambar: bagaimana model menghasilkan gambar baru, seperti menghasilkan gambar dengan gaya seni tertentu atau memanipulasi gambar
11. Penggunaan GPU untuk visi komputer: memahami cara menggunakan GPU untuk mempercepat pelatihan dan inferensi pada model visi komputer

04 Pytorch Custom Datasets

Pada chapter ini, penggunaan custom datasets sangat penting dan diperlukan untuk mengatasi situasi dimana dataset bawaan tidak cukup dan perlu menyesuaikan struktur dataset sesuai dengan kebutuhan spesifik tugas atau proyek.

Langkah-langkah yang dipelajari pada chapter 04 yaitu:

1. Membuat custom dataset: pembuatan kelas dataset khusus yang disusun dan dapat diakses oleh PyTorch
2. Membaca data: implementasi metode untuk membaca data dari dataset khusus. Data akan melibatkan membaca gambar, teks, atau jenis data lainnya yang ditangani.
3. Transformasi data: penggunaan transformasi untuk memproses dan mengubah data saat dibaca. Transformasi mencakup normalisasi, pembalikan, data atau augmentasi data untuk meningkatkan variasi.
4. Menggunakan DataLoader: integrasi dataset khusus dengan PyTorch 'DataLoader' untuk membantu mengatur data dan memungkinkan anda untuk menggunakannya dengan mudah dalam pelatihan model
5. Membuat custom collate function: beberapa kasus yaitu perlu membuat fungsi collate khusus untuk mengelola batch data dalam cara yang sesuai
6. Penggunaan dataset dalam pelatihan: penggunaan dataset khusus dalam proses pelatihan model yang melibatkan iterasi melalui batch data, memberikan input model, dan mengoptimalkan model
7. Handling labels dan anotasi: dataset yang melibatkan tugas klasifikasi atau deteksi objek perlu memahami bagaimana mengelola label dan anotasi yang terkait dengan dataset

8. Visualisasi data: memvisualisasikan data dari dataset untuk membantu memahami dan memeriksa apakah data telah dimuat dan diproses dengan benar
9. Penggunaan dataset pada validasi dan pengujian: integrasi dataset khusus dalam proses validasi dan pengujian model untuk mengukur kinerja dan generalisasi

05 PyTorch Going Modular

Pada chapter ini, hal yang dipelajari yaitu pendekatan modular dalam pengembangan perangkat lunak atau pustaka di dalam ekosistem PyTorch. Disini modular melibatkan kode notebook menjadi serangkaian skrip Python.

Pengubahan kode notebook dari serangkaian sel menjadi file python:

1. Data_setup.py : file untuk menyiapkan dan mengunduh data jika diperlukan
2. Engine.py: file yang berisi berbagai fungsi pelatihan
3. Model_builder.py : file untuk memanfaatkan semua file lain dan melatih model PyTorch target
4. Utils.py: file yang didedikasikan untuk fungsi utilitas yang bermanfaat.