

Nama: Rayyan Ramanda
NIM: 1103200202

00. PyTorch Fundamentals

Pendahuluan: Apa itu PyTorch?

PyTorch merupakan sebuah kerangka kerja pembelajaran mesin dan pembelajaran mendalam yang bersifat open source. Dikembangkan oleh Facebook's AI Research lab, PyTorch menyediakan keleluasaan bagi pengguna untuk memanipulasi data dan menulis algoritma pembelajaran mesin menggunakan Python. Banyak perusahaan teknologi besar seperti Meta (Facebook), Tesla, Microsoft, serta lembaga penelitian kecerdasan buatan seperti OpenAI menggunakan PyTorch untuk mendukung penelitian dan mengimplementasikan pembelajaran mesin dalam produk mereka. Kelebihan utama PyTorch adalah sintaksis yang dinamis, memungkinkan pengguna untuk melakukan komputasi tensor dengan cara yang mirip dengan Python biasa.

PyTorch bisa digunakan untuk apa?

PyTorch dapat digunakan untuk berbagai tugas dalam machine learning seperti pengenalan gambar, pemrosesan bahasa alami, prediksi, klasifikasi, dan sebagainya. PyTorch juga mendukung pembuatan model neural network dengan arsitektur yang kompleks dan canggih, membuatnya sangat berguna dalam riset dan pengembangan kecerdasan buatan.

Alasan Menggunakan PyTorch

PyTorch populer di kalangan peneliti karena kemudahannya dalam eksperimen dan prototyping cepat. Sejak Februari 2022, PyTorch menjadi kerangka kerja pembelajaran mendalam yang paling banyak digunakan di platform Papers With Code.

Instalasi dan Pengaturan PyTorch

Untuk menggunakan PyTorch, pengguna perlu mengikuti langkah-langkah pengaturan yang tersedia di situs resmi PyTorch. Pengguna Google Colab dapat langsung menggunakan PyTorch tanpa instalasi tambahan.

Konsep Dasar PyTorch

- Tensor: Struktur data utama di PyTorch, serupa dengan array di NumPy, tetapi dengan dukungan GPU.
- Operasi Tensor: PyTorch menyediakan berbagai operasi untuk manipulasi tensor, termasuk penjumlahan, pengurangan, dan operasi matriks.
- Penggunaan GPU: PyTorch mempermudah penggunaan GPU untuk mempercepat perhitungan.

Praktik Penggunaan PyTorch

- a. Mengimpor PyTorch: `import torch`
- b. Mengecek Versi: `torch.__version__`
- c. Membuat Tensor: `tensor = torch.Tensor([[1, 2], [3, 4]])`

Apa itu Tensor?

Tensor adalah struktur data utama dalam PyTorch. Secara sederhana, tensor adalah array multidimensional yang mirip dengan matriks atau vektor, tetapi dapat memiliki dimensi yang lebih tinggi. Tensor digunakan untuk merepresentasikan data dalam bentuk yang dapat dioperasikan oleh model machine learning.

Apa tugas tensor?

Tensor memiliki tugas utama untuk menyimpan dan mengelola data numerik yang digunakan sebagai input dan output dalam model machine learning. Mereka juga berperan penting dalam operasi matematika yang diperlukan selama pelatihan dan inferensi model.

Membuat Tensor

Membuat tensor dapat dilakukan dengan berbagai cara. Contohnya, dengan menggunakan konstruktor `torch.tensor` untuk mengonversi list Python menjadi tensor. Ini memungkinkan pembuatan tensor dengan nilai yang sudah ditentukan. Berikut adalah contohnya

```
import torch

# Membuat tensor dari list
tensor_list = torch.tensor([1, 2, 3])

# Membuat tensor berukuran 2x3 dengan nilai acak
tensor_random = torch.rand(2, 3)
```

Sumber Daya untuk Pembelajaran

PyTorch memiliki komunitas yang aktif dengan berbagai sumber daya pembelajaran, termasuk dokumentasi resmi, forum pengembang, dan kursus online.

Kesimpulan dan Sumber Daya Tambahan

PyTorch menawarkan fleksibilitas dan kekuatan untuk berbagai aplikasi pembelajaran mesin dan pembelajaran mendalam. Dengan komunitas yang berkembang dan sumber daya yang melimpah, PyTorch menjadi pilihan populer di kalangan peneliti dan pengembang.

Sumber Daya Tambahan

- GitHub PyTorch: <https://github.com/pytorch/pytorch>
- Forum Diskusi PyTorch: <https://discuss.pytorch.org/>
- Kursus dan Tutorial Online

PyTorch terus berkembang, dengan inovasi dan peningkatan yang dilakukan secara terus-menerus, menjadikannya sebagai salah satu pilar penting dalam dunia kecerdasan buatan saat ini.

01. PyTorch Workflow Fundamentals

Pendahuluan: Pengenalan Alur Kerja PyTorch

PyTorch, sebagai kerangka kerja pembelajaran mesin dan pembelajaran mendalam, menawarkan pendekatan sistematis untuk pengembangan model. Alur kerja ini dimulai dengan pemahaman data dan berakhir dengan penyimpanan serta penerapan model. Laporan ini akan membahas langkah-langkah penting dalam alur kerja PyTorch, menggunakan contoh sederhana yaitu prediksi garis lurus.

1. Persiapan Data

Langkah pertama dalam alur kerja PyTorch adalah persiapan data. Dalam konteks ini, kita akan menciptakan dataset garis lurus. Data ini akan digunakan untuk melatih model dan menguji kemampuannya dalam mempelajari pola.

2. Pembuatan Model

Setelah data siap, langkah selanjutnya adalah pembuatan model. Proses ini melibatkan pemilihan fungsi kerugian, pengoptimal, dan pembangunan loop pelatihan. Model ini akan dirancang untuk mengidentifikasi dan belajar dari pola dalam data.

3. Melatih Model

Dengan data dan model yang siap, kita akan melatih model untuk mencari pola dalam data pelatihan. Proses pelatihan ini krusial untuk efektivitas model dalam melakukan prediksi yang akurat.

4. Prediksi dan Evaluasi Model

Setelah pelatihan, langkah selanjutnya adalah menggunakan model untuk membuat prediksi dan mengevaluasi kinerjanya terhadap data uji. Ini akan mengukur seberapa baik model mempelajari pola dari data pelatihan dan menerapkannya pada situasi baru.

5. Menyimpan dan Memuat Model

Setelah model terbukti efektif, kita dapat menyimpannya untuk digunakan di masa mendatang. Proses ini memungkinkan pemulihan model yang sama tanpa perlu melatih ulang dari awal.

6. Mengintegrasikan Semua Langkah

Langkah terakhir adalah menggabungkan semua elemen - dari persiapan data hingga penyimpanan model - menjadi sebuah alur kerja yang terpadu. Integrasi ini penting untuk memastikan efisiensi dan efektivitas dalam pengembangan model pembelajaran mesin.

Kesimpulan

Alur kerja PyTorch menawarkan panduan yang jelas dan sistematis dalam pengembangan model pembelajaran mesin. Dengan mengikuti langkah-langkah ini, pengembang dapat memastikan bahwa model yang mereka kembangkan siap untuk menghadapi tantangan nyata dalam aplikasi dunia nyata.

02. PyTorch Neural Network Classification

Pengantar: Apa Itu Masalah Klasifikasi?

Klasifikasi dalam pembelajaran mesin adalah proses memprediksi kategori atau kelas dari suatu objek. Ini termasuk klasifikasi biner (dua kelas), multi-klasifikasi (lebih dari dua kelas), dan multi-label (beberapa kelas secara bersamaan). Contohnya meliputi prediksi penyakit jantung (ya atau tidak) atau mengidentifikasi gambar (makanan, orang, atau anjing).

1. Masalah Klasifikasi dalam Pembelajaran Mesin

- **Klasifikasi Biner:** Memilih antara dua pilihan, misalnya, prediksi penyakit jantung.
- **Klasifikasi Multi-Kelas:** Memilih antara lebih dari dua pilihan, seperti klasifikasi gambar.
- **Klasifikasi Multi-Label:** Menetapkan lebih dari satu label ke objek, misalnya, kategorisasi artikel Wikipedia.

2. Alur Kerja PyTorch untuk Klasifikasi

Laporan ini mengulang alur kerja PyTorch yang telah dibahas sebelumnya, dengan penekanan pada aplikasi untuk masalah klasifikasi. Termasuk dalam ini adalah langkah-langkah dalam membangun, melatih, dan mengevaluasi model jaringan neural untuk berbagai jenis masalah klasifikasi.

3. Membangun Model Klasifikasi

Dalam membangun model klasifikasi dengan PyTorch, ada beberapa komponen penting:

- **Lapisan Input:** Sesuai dengan jumlah fitur masukan.
- **Lapisan Tersembunyi:** Tergantung pada spesifik masalah, dengan jumlah neuron yang bervariasi.
- **Lapisan Keluaran:** Mencerminkan jumlah kelas dalam masalah klasifikasi.

- Fungsi Aktivasi: Seperti ReLU untuk lapisan tersembunyi dan sigmoid atau softmax untuk lapisan keluaran.

4. Pelatihan dan Evaluasi Model

Model dilatih dengan data pelatihan untuk mempelajari pola dan diuji dengan data uji untuk mengevaluasi keefektifannya. Ini melibatkan penggunaan fungsi kerugian yang sesuai (misalnya, binary crossentropy atau cross entropy) dan pengoptimal seperti SGD atau Adam.

5. Implementasi Praktis dengan PyTorch

Notebook ini menyediakan contoh praktis langkah demi langkah dalam mengimplementasikan model klasifikasi menggunakan PyTorch. Ini meliputi persiapan data, konstruksi model, proses pelatihan, dan evaluasi.

Kesimpulan

Klasifikasi jaringan neural adalah komponen kunci dalam banyak aplikasi pembelajaran mesin. Dengan memahami berbagai jenis masalah klasifikasi dan menggunakan alur kerja PyTorch, pengembang dapat secara efektif membangun dan menerapkan model ini dalam berbagai skenario.

03. PyTorch Computer Vision

Pendahuluan: Pengantar Visi Komputer

Visi komputer adalah seni mengajarkan komputer untuk "melihat" dan menginterpretasikan visual. Ini meliputi pengembangan model yang mampu mengklasifikasikan gambar (misalnya, apakah sebuah foto adalah kucing atau anjing) atau mendeteksi objek dalam video. Dalam konteks ini, PyTorch menjadi alat yang kuat untuk mengimplementasikan solusi visi komputer.

1. Aplikasi Visi Komputer

- Di Ponsel Pintar: Aplikasi kamera yang menggunakan visi komputer untuk meningkatkan dan mengelompokkan gambar.
- Di Mobil Modern: Sistem yang menggunakan visi komputer untuk menghindari tabrakan dan tetap di jalur.
- Di Industri Manufaktur: Mengidentifikasi cacat produk.
- Di Sistem Keamanan: Deteksi potensi ancaman atau aktivitas mencurigakan.

2. Implementasi Tugas Visi Komputer dengan PyTorch

Notebook ini memberikan demonstrasi praktis dari berbagai tugas visi komputer, termasuk:

- Klasifikasi Biner dan Multi-Kelas: Mengklasifikasikan gambar ke dalam dua atau lebih kategori.
- Deteksi Objek: Mengidentifikasi dan menentukan lokasi objek dalam frame.
- Segmentasi: Memisahkan objek yang berbeda dalam satu gambar.

3. Membangun dan Menilai Model

Langkah-langkah untuk menciptakan dan mengevaluasi model untuk tugas visi komputer meliputi:

- Persiapan Data: Menyiapkan dataset yang akan digunakan.
- Pemilihan Fungsi Kerugian dan Pengoptimal: Menentukan parameter untuk melatih model.
- Membuat Loop Pelatihan: Proses iteratif untuk mengajarkan model.
- 4. Membandingkan dan Menilai Model
- Teknik Perbandingan Model: Cara untuk membandingkan kinerja berbagai model.

- **Evaluasi Kinerja Model:** Menggunakan matriks kebingungan dan metrik lain untuk menilai efektivitas model.

Kesimpulan

Visi komputer adalah area penting dalam kecerdasan buatan dan pembelajaran mesin, dengan aplikasi yang luas di berbagai bidang. PyTorch menyediakan alat yang diperlukan untuk mengembangkan solusi visi komputer yang efektif dan efisien.

04. PyTorch Custom Datasets

Pendahuluan: Mengenal Dataset Kustom

PyTorch, sebagai framework pembelajaran mesin, memungkinkan penggunaan dataset kustom untuk berbagai aplikasi. Dataset kustom adalah kumpulan data yang spesifik terhadap suatu masalah atau proyek tertentu. Contohnya bisa berupa gambar makanan untuk aplikasi klasifikasi makanan, ulasan pelanggan untuk analisis sentimen, atau sampel suara untuk aplikasi klasifikasi suara.

1. Apa itu Dataset Kustom?

- **Definisi:** Kumpulan data yang dirancang untuk memecahkan masalah spesifik.
- **Contoh Aplikasi:** Klasifikasi gambar makanan, analisis sentimen dari ulasan, klasifikasi sampel suara.

2. Menggunakan Dataset Kustom dalam PyTorch

Laporan ini akan menjelaskan bagaimana membuat dan menggunakan dataset kustom dalam PyTorch. Langkah-langkahnya meliputi:

- **Membuat Dataset Kustom:** Langkah awal dalam mengolah data khusus untuk digunakan dalam model.
- **Proses Pemuatan Data:** Teknik memuat dan memproses data agar siap digunakan oleh model.
- **Menerapkan Augmentasi Data:** Penggunaan teknik augmentasi data untuk meningkatkan variasi dan kualitas dataset.

3. Membangun dan Menilai Model dengan Dataset Kustom

- **Konstruksi Model:** Membuat model pembelajaran mesin yang disesuaikan dengan data kustom.
- **Evaluasi Model:** Menilai kinerja model menggunakan metrik yang sesuai.

4. Augmentasi Data dan Teknik Lanjutan

- **Penggunaan Augmentasi Data:** Memperluas variasi dataset dengan teknik seperti rotasi gambar, perubahan skala, atau pemotongan.
- **Manfaat Augmentasi Data:** Meningkatkan robustness model dan mencegah overfitting.

5. Aplikasi Praktis dan Prediksi

- **Penerapan Model pada Data Baru:** Demonstrasi penggunaan model yang telah dilatih untuk membuat prediksi pada data di luar set pelatihan.
- **Menghadapi Tantangan dengan Dataset Kustom:** Solusi untuk tantangan umum yang muncul saat bekerja dengan dataset kustom.

Kesimpulan

Penggunaan dataset kustom dalam PyTorch menawarkan fleksibilitas yang besar untuk menyelesaikan berbagai masalah pembelajaran mesin yang spesifik. Dengan pemahaman yang tepat tentang cara membuat dan menggunakan dataset tersebut, pengembang dapat mengoptimalkan performa model mereka.