

Technical Report Pytorch

By : Fahmi Nanda Saputro

- 00 Pytorch Fundamentals

Pada materi 00 ini membahas pengenalan Pytorch, pytorch ini biasa digunakan di META, tesla, dan Microsoft. Di materi ini mempelajari bagaimana membuat tensor, random tensor, serta zero dan ones, selain itu, membuat range dan tensor datatype. Melakukan penghitungan basic menggunakan matrix dan mengubah posisi min / max dan index. Dan menghubungkan Pytorch dengan Numpy.

- 01 PyTorch Workflow Fundamentals

Dasar-Dasar Alur Kerja PyTorch mencakup langkah-langkah utama yang penting untuk mengembangkan dan menerapkan model pembelajaran mesin. Langkah pertama melibatkan persiapan dan pemuatan data, di mana kumpulan data diatur dan diubah untuk memfasilitasi pelatihan model. PyTorch menyediakan alat yang mudah digunakan untuk memanipulasi dan memuat data dengan kelas DataLoader dan Dataset, yang menyederhanakan penyisipan data ke dalam model.

Setelah data siap, langkah selanjutnya adalah membuat arsitektur model. Grafik perhitungan dinamis PyTorch memungkinkan desain model yang fleksibel dan intuitif. Pemrogram dapat mendefinisikan arsitektur jaringan saraf dengan membuat kelas yang mewarisi apa yang disebut PyTorch modul. Pada langkah ini, level, aktivasi, dan koneksi yang membentuk model ditentukan. Setelah membangun model, pelatihan menjadi langkah penting. PyTorch menyederhanakan proses ini dengan menggunakan diferensiasi otomatis dengan grafik komputer dinamis, yang memungkinkan pengoptimalan yang efisien menggunakan algoritme pengoptimalan yang berbeda. Setelah pelatihan, model atau inferensi PyTorch yang dilatih menjadi mudah. Model yang dilatih dapat diterapkan pada data baru untuk menghasilkan prediksi dengan meneruskan tensor masukan melalui model dan fungsi feedforward.

Menyimpan dan memuat model PyTorch diperlukan untuk menyimpan model terlatih dan membaginya dengan orang lain. PyTorch menyediakan fungsi untuk menyimpan dan memuat parameter model, memungkinkan integrasi yang lancar di berbagai proyek atau lingkungan penerapan. Hal ini memastikan bahwa model yang dilatih dapat dengan mudah digunakan dalam tugas-tugas mendatang atau dibagikan dengan kolaborator, sehingga meningkatkan kegunaan dan pengulangan alur kerja pembelajaran mesin.

- 02 Pytorch Neural Network Classification

PyTorch Neural Network Classification adalah pendekatan dalam pengembangan model machine learning menggunakan framework PyTorch yang fokus pada tugas klasifikasi. Arsitektur model untuk klasifikasi umumnya melibatkan beberapa lapisan (layers) jaringan saraf, di mana setiap lapisan memiliki sejumlah unit atau neuron. PyTorch memfasilitasi pembuatan arsitektur ini dengan menyediakan

modul-modul seperti `nn.Module` dan `nn.Linear` untuk mendefinisikan lapisan-lapisan serta fungsi aktivasi seperti `ReLU` atau `Sigmoid` untuk mengenali pola dan kompleksitas dalam data.

Proses klasifikasi dalam konteks ini berkaitan dengan kemampuan model untuk mengelompokkan data input ke dalam kategori atau kelas yang sudah ditentukan. Dalam konteks klasifikasi, biasanya digunakan fungsi aktivasi `Softmax` pada lapisan output untuk menghasilkan distribusi probabilitas dari setiap kelas. Proses training melibatkan penyesuaian bobot (`weights`) dan bias model berdasarkan perbedaan antara prediksi model dengan label yang seharusnya. PyTorch menyediakan fungsi-fungsi otomatis diferensiasi yang mempermudah proses ini.

Selain itu, PyTorch Neural Network Classification mencakup langkah-langkah penting seperti preprocessing data, pemilihan fungsi loss (kerugian) yang sesuai, dan pemilihan optimizer untuk memperbarui parameter model. Setelah model dilatih, dapat digunakan untuk membuat prediksi pada data baru dengan menjalankan data tersebut melalui model yang telah tersimpan. Keseluruhan proses ini memungkinkan pengguna untuk membangun, melatih, dan menggunakan model klasifikasi dengan mudah menggunakan PyTorch.

- 03 PyTorch Computer Vision

PyTorch dalam bidang Computer Vision menyediakan kerangka kerja yang kuat untuk mengatasi tugas-tugas seperti klasifikasi gambar menggunakan Convolutional Neural Networks (CNNs). Proses dimulai dengan persiapan dataset, yang melibatkan pengumpulan dan pengorganisasian kumpulan data gambar yang mencakup berbagai kategori atau kelas. PyTorch menyediakan berbagai utilitas untuk memuat dan mempersiapkan dataset, termasuk kelas `DataLoader` yang mempermudah pembagian data latih, validasi, dan uji secara efisien ke dalam model.

Selanjutnya, untuk membuat prediksi dalam konteks Computer Vision, CNNs menjadi algoritma yang sangat populer. CNNs dirancang untuk menangani data gambar dengan memanfaatkan operasi konvolusi untuk mengekstrak fitur-fitur lokal dari gambar. PyTorch menyediakan modul-modul khusus seperti `nn.Conv2d` untuk lapisan konvolusi dan `nn.MaxPool2d` untuk lapisan pooling, memungkinkan pengguna untuk dengan mudah membangun arsitektur CNN. Setelah model CNN dibangun, proses training melibatkan menggunakan dataset yang telah disiapkan sebelumnya untuk mengoptimalkan bobot model menggunakan metode-metode seperti stochastic gradient descent (SGD). Setelah model dilatih, dapat digunakan untuk melakukan prediksi pada data gambar baru, memberikan klasifikasi yang akurat tergantung pada apa yang telah dipelajari oleh model selama proses training. Inilah dasar dari PyTorch Computer Vision, menggabungkan dataset, `DataLoader`, algoritma CNN, dan proses prediksi untuk menangani tugas-tugas penglihatan komputer secara efektif.

- 04 PyTorch Custom Datasets

PyTorch Custom Datasets memungkinkan pengguna untuk membuat dataset yang sesuai dengan kebutuhan khusus mereka. Transformasi data adalah komponen penting dalam PyTorch Custom Datasets, memungkinkan pengguna untuk secara dinamis mengubah atau memanipulasi data input sebelum dimasukkan ke dalam model. Contohnya, operasi transformasi seperti cropping, resizing, dan normalisasi dapat diterapkan pada gambar dalam dataset untuk meningkatkan keberlanjutan dan kinerja model. PyTorch menyediakan modul `torchvision.transforms` untuk mempermudah penerapan transformasi data ini, memungkinkan pengguna untuk dengan mudah menyesuaikan data masukan sesuai dengan persyaratan model atau tugas klasifikasi tertentu.

Salah satu arsitektur Convolutional Neural Network (CNN) yang sering digunakan dalam konteks PyTorch Custom Datasets adalah VGGnet (Visual Geometry Group network). VGGnet dikenal karena memiliki struktur yang mendalam dan homogen, terdiri dari lapisan-lapisan konvolusi dan pooling berturut-turut. Model ini telah terbukti efektif dalam tugas-tugas klasifikasi gambar. Dengan menggunakan PyTorch, pengguna dapat dengan mudah membuat dan melatih model VGGnet untuk mengenali pola dan fitur pada dataset yang telah dipersiapkan dengan transformasi data. Inilah bagaimana PyTorch Custom Datasets dan model VGGnet dapat bekerja bersama untuk menangani tugas-tugas penglihatan komputer yang spesifik.

- 05 PyTorch Going Modular

PyTorch Going Modular mengacu pada pendekatan modular dalam pembangunan program machine learning dengan menggunakan PyTorch. Pertama, program tersebut mencari atau mengumpulkan data yang dibutuhkan untuk melatih dan menguji model. Pengguna dapat memanfaatkan berbagai sumber data seperti gambar, teks, atau data numerik, dan kemudian menggunakan modul PyTorch `DataLoader` untuk mengelola dan memuat data tersebut secara efisien ke dalam model.

Selanjutnya, dalam konteks modular, pembuatan dataset menjadi langkah yang terpisah dan mudah diatur. PyTorch memungkinkan pengguna untuk membuat dataset khusus dengan mendefinisikan kelas dataset yang sesuai dengan struktur data yang dimiliki. Modul PyTorch `Dataset` dapat diimplementasikan untuk mengelola data dan memberikan akses yang nyaman ke sampel-sampel dalam proses pelatihan. Setelah dataset dibuat, program modular dapat melibatkan langkah-langkah seperti pembuatan model, definisi fungsi pelatihan (train), dan evaluasi model. Pengguna dapat merancang arsitektur model sesuai kebutuhan dan mengembangkan fungsi pelatihan yang menyesuaikan bobot model berdasarkan hasil dari fungsi kerugian yang diterapkan pada data pelatihan. Evaluasi model kemudian dilakukan dengan menggunakan dataset uji terpisah untuk mengukur kinerja dan akurasi model secara objektif. Dengan pendekatan modular ini, pengguna dapat dengan

mudah menyesuaikan dan mengembangkan setiap komponen dalam alur kerja machine learning sesuai kebutuhan spesifik mereka.

- 06 PyTorch Transfer Learning

Transfer learning dalam PyTorch merujuk pada teknik memanfaatkan model jaringan saraf yang sudah dilatih pada dataset besar dan menyesuakannya untuk melakukan tugas yang berbeda atau pada dataset yang lebih kecil. Pendekatan ini memungkinkan model untuk memanfaatkan pengetahuan yang diperoleh selama pelatihan pada tugas sumber dan mempercepat proses pembelajaran untuk tugas target. PyTorch menyediakan lingkungan yang ramah pengguna untuk mengimplementasikan transfer learning, memungkinkan pengembang dengan mudah memuat model yang sudah dilatih, mengubah lapisan terakhir, dan menyesuaikan jaringan untuk aplikasi tertentu.

- 07 Pelacakan Eksperimen PyTorch

Pelacakan eksperimen PyTorch melibatkan organisasi sistematis dan pencatatan berbagai eksperimen yang dilakukan selama pengembangan dan pelatihan model pembelajaran mesin menggunakan PyTorch. Alat seperti TensorBoard dan pustaka pihak ketiga seperti MLflow atau Weights & Biases dapat digunakan untuk mencatat dan memvisualisasikan metrik kunci, hiperparameter, dan artefak model. Praktik ini membantu peneliti dan pengembang dalam mempertahankan catatan yang jelas dari proses eksperimen mereka, memfasilitasi perbandingan model, reproduktibilitas, dan pemahaman keseluruhan kinerja model di berbagai konfigurasi.

- 08 Replikasi Makalah PyTorch

Replikasi makalah PyTorch melibatkan proses mengimplementasikan dan mereproduksi hasil dari makalah penelitian yang menyajikan algoritma atau model baru menggunakan kerangka kerja PyTorch. Praktik ini sangat penting untuk memvalidasi dan membangun penelitian yang sudah ada, memastikan bahwa komunitas ilmiah dapat mempercayai dan memperluas metodologi yang diusulkan. Replikasi makalah di PyTorch memerlukan pemahaman yang mendalam terhadap konten makalah, implementasi hati-hati dari model yang diusulkan, dan validasi terhadap hasil yang dilaporkan untuk mengonfirmasi kebenaran dan efektivitas metode yang dijelaskan dalam makalah.

- 09 Penyebaran Model PyTorch

Penyebaran Model PyTorch merujuk pada proses mengambil model PyTorch yang sudah dilatih dan membuatnya tersedia untuk digunakan dalam lingkungan produksi. Ini melibatkan mengonversi model PyTorch menjadi format yang cocok untuk penyebaran, mengintegrasikannya dengan infrastruktur penyebaran, dan memastikan bahwa model dapat menangani data dunia nyata dengan efisien dan dapat diandalkan. Opsi penyebaran populer termasuk menyebar model sebagai API REST, mengintegrasikannya ke dalam aplikasi web, atau menyebar model pada perangkat edge. Alat seperti TorchServe dan kerangka kerja seperti Flask atau FastAPI umumnya digunakan untuk memfasilitasi penyebaran model PyTorch di berbagai pengaturan produksi.