

Technical Report Ujian Akhir Semester Machine Learning

Disusun untuk memenuhi tugas Mata Kuliah Machine Learning



Disusun oleh:

Muhammad Faisal Ramadhan (1103203227)

PROGRAM SARJANA TEKNIK KOMPUTER

FAKULTAS TEKNIK ELEKTRO

TELKOM UNIVERSITY

2022/2023

A. Pendahuluan

PyTorch adalah sebuah perpustakaan atau library dalam bahasa pemrograman Python yang digunakan untuk mengembangkan dan melatih model jaringan saraf (neural network). PyTorch dirancang khusus untuk mendukung komputasi tensor dan aliran grafik (graph computation) yang efisien dalam pengolahan data pada jaringan saraf.

Salah satu fitur utama PyTorch adalah kemampuannya dalam mendefinisikan dan melatih model jaringan saraf secara dinamis. Dalam PyTorch, Anda dapat dengan mudah membuat dan mengubah model jaringan saraf secara fleksibel menggunakan sintaksis Python yang intuitif. Hal ini memungkinkan para peneliti dan pengembang untuk bereksperimen dengan berbagai arsitektur jaringan saraf dengan mudah.

Selain itu, PyTorch juga menyediakan alat-alat yang kuat untuk melakukan komputasi numerik, seperti operasi tensor yang efisien, pengoptimalan model, dan utilitas untuk memproses data. PyTorch juga memiliki antarmuka yang baik dengan perangkat keras seperti GPU (Graphics Processing Unit), yang memungkinkan pengguna memanfaatkan kecepatan paralelisme yang tinggi untuk melatih model jaringan saraf secara efisien.

PyTorch juga memiliki komunitas yang aktif dan luas, yang menyediakan sumber daya pembelajaran, tutorial, dan kode sumber terbuka yang dapat digunakan untuk mempelajari dan mengembangkan aplikasi menggunakan PyTorch.

Secara keseluruhan, PyTorch telah menjadi salah satu perpustakaan paling populer untuk pengembangan model jaringan saraf karena kelebihanannya dalam fleksibilitas, kinerja, dan komunitas yang kuat.

B. Tensor pada Sebuah PyTorch

Tensor dalam PyTorch adalah struktur data utama yang digunakan untuk menyimpan dan mengelola data numerik dalam bentuk multi-dimensional. Tensor mirip dengan matriks dalam matematika, tetapi dengan fleksibilitas yang lebih besar. PyTorch menggunakan tensor untuk melakukan komputasi numerik dan memproses data pada model jaringan saraf.

Berikut ini beberapa karakteristik utama dari tensor dalam PyTorch:

Dimensi: Tensor dapat memiliki dimensi apa pun, mulai dari tensor skalar (0 dimensi) hingga tensor multidimensi yang kompleks. Sebagai contoh, tensor 1D merupakan vektor, tensor 2D merupakan matriks, dan tensor 3D atau lebih tinggi dapat merepresentasikan array multidimensi.

Tipe Data: Tensor dalam PyTorch dapat mengandung data dengan tipe numerik seperti float, integer, dan sebagainya. PyTorch menyediakan beberapa tipe data bawaan seperti float32, float64, int8, int16, int32, dan sebagainya.

Operasi: PyTorch menyediakan berbagai operasi yang dapat diterapkan pada tensor, seperti penjumlahan, pengurangan, perkalian, pembagian, transpose, pemotongan, dan banyak lagi. Operasi-operasi ini memungkinkan manipulasi dan transformasi tensor secara efisien.

Komputasi Paralel: PyTorch dirancang untuk berinteraksi dengan perangkat keras seperti GPU. Dengan menggunakan GPU, PyTorch dapat melakukan komputasi paralel pada tensor, yang mempercepat proses pelatihan dan inferensi pada model jaringan saraf.

Autograd: Salah satu fitur penting dari PyTorch adalah kemampuannya dalam melakukan pelacakan otomatis (autograd) terhadap operasi yang dilakukan pada tensor. Autograd memungkinkan perhitungan gradien (gradients) untuk digunakan dalam proses pelatihan model jaringan saraf dengan mudah.

Tensor dalam PyTorch dapat dibuat dari berbagai sumber data, seperti array NumPy, data yang diambil dari file, atau secara langsung melalui fungsi-fungsi pembuatan tensor yang disediakan oleh PyTorch itu sendiri.

Dengan menggunakan tensor, PyTorch memungkinkan para pengguna untuk melakukan berbagai komputasi numerik dan memanipulasi data dengan mudah dan efisien dalam konteks pengembangan model jaringan saraf.

C. PyTorch untuk Komputasi Gradien

Pada penggunaan PyTorch kali ini, dilakukan sebuah pencarian nilai gradien menggunakan pyTorch dimana inisialisasi datanya menggunakan fungsi dari torch.rand yang kemudian akan menghasilkan data random yang nantinya akan dicari gradiennya menggunakan fungsi grad.fn() yang membutuhkan sebuah grad pada inisialisasi datanya requires_grad = True.

Kita melakukan perhitungan ini dengan implementasi autoGrad yang menghitung gradien secara otomatis. Di dalamnya kita menggunakan backward propagation untuk menghitung nilai gradien tersebut.

D. Regresi Linier PyTorch

Regresi linier adalah salah satu jenis model regresi yang digunakan untuk memodelkan hubungan linier antara variabel independen (fitur) dan variabel dependen (target). PyTorch menyediakan kemampuan untuk melatih model regresi linier dengan menggunakan optimasi gradien turun (gradient descent) dan fitur autograd.

Pada percobaannya kita menggunakan dataset.make_regression dari sklearn yang nantinya diubah ke datatype tensor pyTorch menggunakan torch.from_numpy. Kemudian dilakukan pelatihan model menggunakan sebuah loop dengan isi loop menggunakan forward pass, perhitungan loss, backward pass untuk menghitung

gradien, dan optimasi menggunakan `optimizer.step()` yang akhirnya gradien direset ke zero.

E. Kesimpulan

Deep learning with pyTorch, merupakan salah satu pendekatan yang populer dalam bidang kecerdasan buatan (artificial intelligence) yang memungkinkan kita untuk membangun dan melatih model jaringan saraf yang dalam (deep neural networks). PyTorch adalah salah satu framework deep learning yang sangat populer dan kuat yang menggunakan bahasa pemrograman Python. PyTorch menyediakan berbagai alat yang sangat baik untuk membangun, melatih, dan menerapkan model jaringan saraf yang dalam. PyTorch memiliki kelebihan mudah digunakan dan memiliki sintaks yang intuitif. Ini memungkinkan para peneliti dan praktisi untuk dengan cepat membangun prototipe model baru, melakukan eksperimen, dan memecahkan masalah kecerdasan buatan dengan cepat.

Framework ini juga memiliki fitur yang kuat seperti automatic differentiation melalui modul `torch.autograd`, yang memudahkan dalam perhitungan gradien yang diperlukan untuk pelatihan model. Selain itu, PyTorch menyediakan berbagai fungsi loss dan optimizer yang terintegrasi dengan baik, sehingga mempermudah proses pelatihan model.

Secara keseluruhan, PyTorch adalah framework deep learning yang populer, kuat, dan fleksibel yang memungkinkan para peneliti dan praktisi untuk membangun dan melatih model jaringan saraf yang dalam dengan mudah. Dukungan komunitas yang luas dan ekosistem Python yang kuat menjadikan PyTorch pilihan yang menarik dalam pengembangan dan eksplorasi dalam bidang deep learning.