

TECHNICAL REPORT UAS MACHINE LEARNING

Implementasi dan Analisis Penggunaan PyTorch



Disusun Oleh :

Zaidan Luthfi

1103203238

PROGRAM STUDI S1 TEKNIK KOMPUTER

FAKULTAS TEKNIK ELEKTRO

UNIVERSITAS TELKOM

2023

1. Pendahuluan

Laporan ini menjelaskan penggunaan PyTorch, termasuk manajemen tensor, perhitungan gradien, dan implementasi model regresi linear, regresi logistik, dan neural network. Tujuannya adalah untuk memberikan pemahaman yang mendalam tentang cara mengimplementasikan dan menganalisis model pembelajaran mendalam dengan PyTorch. PyTorch adalah kerangka kerja pembelajaran mendalam yang populer dan kuat yang memungkinkan pengembang untuk membuat dan melatih model neural network dengan mudah. Ini juga menyediakan fitur-fitur yang memudahkan manajemen data, perhitungan gradien, dan implementasi model yang kompleks. Laporan ini akan mengeksplorasi implementasi dan analisis dari beberapa konsep dan teknik penting dalam PyTorch.

PyTorch adalah salah satu kerangka kerja pembelajaran mendalam yang paling banyak digunakan saat ini. Ini menawarkan kemudahan penggunaan dan fleksibilitas yang memungkinkan pengembang untuk membuat model pembelajaran mendalam yang canggih dengan cepat. Dengan fitur-fitur seperti manajemen tensor yang efisien, perhitungan gradien otomatis, dan dukungan untuk berbagai jenis model pembelajaran mendalam, PyTorch telah menjadi pilihan populer bagi banyak peneliti dan pengembang di bidang pembelajaran mesin. Laporan ini akan memberikan wawasan tentang cara menggunakan PyTorch untuk mengimplementasikan dan menganalisis model pembelajaran mendalam yang canggih.

2. Pengelolaan Tensor dalam PyTorch

Tensor adalah struktur data utama dalam PyTorch yang digunakan untuk menyimpan dan memanipulasi data dalam operasi numerik. Ada beberapa konsep penting dalam manajemen tensor yang perlu diketahui, termasuk:

- 1) Inisialisasi Tensor Kosong: Ini memungkinkan untuk membuat tensor kosong dan menginisialisasi nilainya.
- 2) Inisialisasi Tensor dengan Angka Acak: Ini memungkinkan untuk membuat tensor dengan angka acak sesuai dengan distribusi tertentu.
- 3) Inisialisasi Tensor dengan Nol: Ini memungkinkan untuk membuat tensor dengan nilai nol.

- 4) Mengubah Ukuran Tensor: Ini memungkinkan untuk mengubah ukuran tensor dengan menggabungkan atau memisahkan dimensinya.
- 5) Konversi antara Torch Tensor dan NumPy Array: Ini memungkinkan untuk mengkonversi tensor PyTorch menjadi array NumPy, dan sebaliknya.

Manajemen tensor yang efisien sangat penting dalam deep learning, karena ini memungkinkan untuk menyimpan dan memproses data dengan cepat dan efisien. Dengan memahami konsep-konsep ini, PyTorch dapat digunakan secara lebih efektif untuk mengimplementasikan model deep learning yang canggih.

3. Komputasi Gradien dengan PyTorch

Perhitungan gradien adalah elemen penting dalam pelatihan model deep learning. Dalam bagian ini, akan dijelaskan konsep perhitungan gradien menggunakan PyTorch, termasuk:

- 1) Perhitungan Gradien dengan Tensor: Ini memungkinkan untuk menghitung gradien dari tensor menggunakan metode autograd di PyTorch.
- 2) Gradien dan Fungsi Autograd: Ini menjelaskan hubungan antara gradien dan fungsi autograd di PyTorch.
- 3) Perhitungan Gradien dan Propagasi Mundur: Ini memungkinkan untuk melakukan perhitungan gradien dan menjalankan propagasi mundur untuk memperbarui parameter model.

Dengan memahami konsep-konsep ini, PyTorch dapat digunakan secara lebih efektif untuk melatih model deep learning yang canggih. Perhitungan gradien yang akurat sangat penting untuk memastikan bahwa model dapat belajar dari data dengan baik dan mencapai hasil yang optimal.

4. Implementasi Regresi Linear dengan PyTorch

Regresi linear adalah salah satu teknik dasar dalam machine learning. Dalam bagian ini, akan dijelaskan langkah-langkah untuk mengimplementasikan regresi linear sederhana menggunakan PyTorch, termasuk:

- 1) Persiapan Data dan Parameter: Ini melibatkan memuat data latihan, menginisialisasi parameter model, dan membagi data menjadi data latihan dan data uji.
- 2) Fungsi Forward dan Fungsi Loss: Ini melibatkan mengimplementasikan fungsi forward untuk melakukan prediksi dan fungsi loss untuk mengukur kesalahan prediksi.
- 3) Pelatihan Model: Ini melibatkan pelatihan model dengan melakukan iterasi untuk mengoptimalkan parameter menggunakan metode gradient descent.

Dengan mengikuti langkah-langkah ini, dapat diimplementasikan model regresi linear sederhana menggunakan PyTorch. Regresi linear adalah teknik yang berguna untuk memprediksi nilai kontinu berdasarkan variabel independen, dan dapat digunakan dalam berbagai aplikasi machine learning.

5. Implementasi Regresi Logistic dengan PyTorch

Regresi logistik adalah salah satu teknik dasar dalam machine learning. Dalam bagian ini, akan dijelaskan langkah-langkah untuk mengimplementasikan regresi logistik sederhana menggunakan PyTorch, termasuk:

- 1) Persiapan Data dan Parameter: Ini melibatkan memuat data latihan dari dataset kanker payudara, menginisialisasi parameter model, membagi data menjadi data latihan dan data uji, serta melakukan penskalaan pada data latihan dan data uji.
- 2) Fungsi Forward dan Fungsi Loss: Ini melibatkan mengimplementasikan fungsi forward untuk melakukan prediksi dengan menggunakan fungsi sigmoid pada akhirnya dan fungsi loss untuk mengukur kesalahan prediksi dengan menggunakan `nn.BCELoss`.
- 3) Pelatihan Model: Ini melibatkan pelatihan model dengan melakukan iterasi untuk mengoptimalkan parameter menggunakan metode gradient descent dengan optimizer `torch.optim.SGD` selama 100 epoch.

Dengan mengikuti langkah-langkah ini, telah diimplementasikan model regresi logistik sederhana menggunakan PyTorch. Regresi logistik adalah teknik yang berguna untuk memprediksi probabilitas kejadian suatu peristiwa dengan

menggunakan fungsi sigmoid, dan dapat digunakan dalam berbagai aplikasi machine learning.

6. Implementasi Neural Network dengan PyTorch

neural network adalah salah satu model yang paling kuat dalam deep learning. Dalam bagian ini, akan dijelaskan langkah-langkah untuk mengimplementasikan neural network menggunakan PyTorch, termasuk:

- 1) Persiapan Data dan Parameter: Ini melibatkan memuat data latihan, menginisialisasi parameter model, dan membagi data menjadi data latihan dan data uji.
- 2) Fungsi Forward dan Fungsi Loss: Ini melibatkan mengimplementasikan fungsi forward untuk melakukan prediksi dan fungsi loss untuk mengukur kesalahan prediksi.
- 3) Pelatihan Model: Ini melibatkan pelatihan model dengan melakukan iterasi untuk mengoptimalkan parameter menggunakan metode gradient descent.

Dengan mengikuti langkah-langkah ini, dapat diimplementasikan model neural network menggunakan PyTorch. neural network adalah model yang sangat kuat yang dapat digunakan untuk menyelesaikan berbagai masalah dalam deep learning, termasuk klasifikasi, regresi, dan pembelajaran autonom.

7. Kesimpulan

Melalui laporan ini, telah dilakukan implementasi dan analisis dari berbagai konsep dan teknik penting dalam PyTorch seperti Manajemen tensor, perhitungan gradien, implementasi regresi linear, regresi logistik, dan neural network menggunakan PyTorch. Dengan pemahaman yang diperoleh melalui laporan ini, dapat mengimplementasikan dan menganalisis model deep learning menggunakan PyTorch. PyTorch adalah kerangka kerja yang kuat dan fleksibel yang memungkinkan pengembang untuk mengembangkan dan melatih model neural networks dengan mudah. Ini menawarkan fitur-fitur canggih seperti manajemen tensor yang efisien, perhitungan gradien otomatis, dan dukungan untuk berbagai jenis model deep

learning. Ini menjadikan PyTorch salah satu pilihan utama bagi para praktisi di bidang kecerdasan buatan.

Dengan menggunakan PyTorch, dapat diimplementasikan berbagai jenis model deep learning, mulai dari regresi linear sederhana hingga neural network yang kompleks. Ini memungkinkan untuk menyelesaikan berbagai masalah dalam pembelajaran mesin, termasuk klasifikasi, regresi, dan pembelajaran autonom. Dengan pemahaman yang mendalam tentang konsep-konsep dan teknik-teknik yang dijelaskan dalam laporan ini, dapat mengoptimalkan penggunaan PyTorch untuk mengembangkan model deep learning yang .