

TECHNICAL REPORT UAS
MACHINE LEARNING
(Deep Learning with PyTorch)



Rifqi Fadhila Shandi

1103202042

TK-44-04

PROGRAM STUDI TEKNIK KOMPUTER

FAKULTAS TEKNIK ELEKTRO

UNIVERSITAS TELKOM

2022/2023

Deep Learning with PyTorch

A. Pendahuluan

Deep Learning adalah cabang dari machine learning yang bertujuan untuk mengembangkan algoritma yang dapat belajar dan mengambil representasi yang mendalam (hierarkis) dari data. PyTorch adalah kerangka kerja deep learning yang populer dan kuat yang menyediakan alat yang kuat dan fleksibel untuk membangun dan melatih model deep learning. Laporan ini akan menjelaskan konsep dan teknik dasar dalam deep learning dengan menggunakan PyTorch.

B. Struktur Dasar PyTorch

PyTorch memiliki struktur dasar yang terdiri dari tensor, modul, fungsi, dan optimisasi. Tensor adalah struktur data inti dalam PyTorch yang merepresentasikan array multi-dimensi dan menyediakan operasi numerik yang efisien. Modul adalah unit dasar dalam membangun jaringan saraf, dan berfungsi sebagai wadah untuk parameter-parameter dan operasi-operasi yang terkait. Fungsi-fungsi PyTorch menyediakan operasi matematika dan manipulasi tensor yang lebih rendah. Optimisasi digunakan untuk mengoptimalkan parameter-parameter model dengan menggunakan algoritma seperti stochastic gradient descent (SGD).

C. Langkah - Langkah Dasar Dalam Mempelajari PyTorch

1. Autograd:

Autograd, singkatan dari automatic differentiation (diferensiasi otomatis), adalah komponen penting dalam kerangka kerja deep learning yang memungkinkan perhitungan gradien secara otomatis. Autograd memungkinkan perhitungan gradien yang efisien untuk tensor, yang penting dalam pelatihan jaringan saraf menggunakan teknik seperti backpropagation.

2. Backpropagation:

Backpropagation adalah teknik yang digunakan untuk menghitung gradien parameter-parameter jaringan saraf terhadap fungsi loss. Teknik ini melibatkan propagasi kesalahan dari lapisan output ke lapisan input jaringan. Dengan menghitung gradien menggunakan aturan rantai dari turunan, backpropagation memungkinkan optimisasi yang efisien dari jaringan saraf selama proses pelatihan.

3. Gradient Descent dengan Autograd dan Backpropagation:

Gradient descent adalah algoritma optimisasi yang digunakan untuk meminimalkan fungsi loss dari jaringan saraf. Dengan bantuan autograd dan backpropagation, gradient descent dapat diimplementasikan dengan efisien. Selama pelatihan, gradien parameter-parameter jaringan dihitung menggunakan backpropagation, kemudian digunakan untuk memperbarui

parameter-parameter tersebut ke arah yang berlawanan dari gradien, secara bertahap mengurangi loss.

4. Rantai Pelatihan: Model, Loss, dan Optimizer:

Rantai pelatihan terdiri dari tiga komponen utama: model, fungsi loss, dan optimizer. Model mewakili arsitektur jaringan saraf, termasuk lapisan-lapisan dan parameter-parameter yang dimilikinya. Fungsi loss mengukur perbedaan antara output yang diprediksi oleh model dan output yang diinginkan. Optimizer menentukan bagaimana parameter-parameter model diperbarui berdasarkan gradien yang dihitung selama backpropagation.

5. Regresi Linear:

Regresi linear adalah teknik pemodelan statistik yang digunakan untuk mencari hubungan antara variabel dependen dan satu atau lebih variabel independen. Dalam konteks deep learning, regresi linear mengacu pada jaringan saraf dengan satu lapisan linear tunggal dan sering digunakan untuk memprediksi nilai-nilai kontinu.

6. Regresi Logistik:

Regresi logistik adalah algoritma klasifikasi yang digunakan untuk memprediksi probabilitas terjadinya suatu peristiwa. Biasanya digunakan untuk masalah klasifikasi biner. Dalam deep learning, regresi logistik biasanya diimplementasikan sebagai jaringan saraf dengan fungsi aktivasi sigmoid di lapisan output.

7. Dataset dan DataLoader:

Dataset mewakili kumpulan data input dan label yang sesuai. Dataset digunakan sebagai data pelatihan atau pengujian bagi jaringan saraf. DataLoader adalah utilitas dalam kerangka kerja deep learning yang membantu dalam pemuatan dan pembagian data dari dataset secara efisien selama pelatihan atau evaluasi.

8. Transformasi Dataset:

Transformasi dataset adalah operasi yang diterapkan pada data input untuk melakukan augmentasi data atau pra-pemrosesan. Transformasi umum termasuk perubahan ukuran, pemangkasan, normalisasi, dan gangguan acak. Transformasi ini membantu meningkatkan generalisasi dan kinerja jaringan saraf.

9. Softmax dan Cross Entropy:

Softmax adalah fungsi aktivasi yang umum digunakan dalam lapisan output jaringan saraf untuk klasifikasi multi-kelas. Ia mengubah output kasar jaringan menjadi probabilitas yang jumlahnya menjadi 1. Cross entropy adalah fungsi loss yang mengukur perbedaan antara probabilitas yang diprediksi dan probabilitas yang sebenarnya dari kelas-kelas tersebut. Biasanya digunakan bersamaan dengan fungsi aktivasi softmax.

10. Fungsi Aktivasi:

Fungsi aktivasi memperkenalkan non-linearitas ke dalam jaringan saraf, memungkinkan jaringan untuk mempelajari hubungan kompleks dalam data. Fungsi aktivasi umum meliputi ReLU (Rectified Linear Unit), sigmoid, tanh (tangent hiperbolik), dan softmax. Setiap fungsi aktivasi memiliki karakteristiknya sendiri dan digunakan pada bagian yang berbeda dalam jaringan saraf.

11. Jaringan Saraf Feed-Forward:

Jaringan saraf feed-forward, juga dikenal sebagai multilayer perceptron (MLP), adalah jenis jaringan saraf di mana informasi mengalir hanya ke satu arah, dari lapisan input ke lapisan output. Ia terdiri dari beberapa lapisan neuron, di mana setiap neuron terhubung dengan neuron-neuron pada lapisan-lapisan yang berdekatan. Jaringan saraf feed-forward digunakan untuk berbagai tugas, termasuk klasifikasi dan regresi.

12. Jaringan Saraf Konvolusi (CNN):

Jaringan saraf konvolusi (CNN) adalah jenis khusus dari jaringan saraf yang umum digunakan untuk menganalisis data visual, seperti gambar. CNN memanfaatkan lapisan konvolusi, yang menerapkan filter atau kernel pada data input, memungkinkan jaringan untuk secara otomatis mempelajari hierarki spasial dan mengekstraksi fitur yang bermakna. CNN telah mencapai kesuksesan luar biasa dalam tugas-tugas seperti klasifikasi gambar dan deteksi objek.

13. Transfer Learning:

Transfer learning adalah teknik dalam deep learning di mana model yang telah dilatih sebelumnya pada dataset besar digunakan sebagai titik awal untuk menyelesaikan tugas atau dataset baru. Dengan memanfaatkan pengetahuan yang dipelajari dari pelatihan sebelumnya, transfer learning

dapat secara signifikan mengurangi waktu pelatihan dan meningkatkan kinerja model pada tugas baru dengan data terbatas yang berlabel.

14. Tensorboard:

TensorBoard adalah toolkit visualisasi yang disediakan oleh TensorFlow yang memungkinkan pengguna untuk memvisualisasikan dan menganalisis berbagai aspek dari model deep learning mereka. Ia menyediakan visualisasi interaktif dari nilai skalar, histogram, arsitektur model, dan bahkan penggolongan. TensorBoard membantu peneliti dan praktisi untuk memperoleh wawasan tentang model mereka, memantau kemajuan pelatihan, dan men-debug masalah potensial.

15. Menyimpan dan Memuat Model:

Setelah melatih model deep learning, penting untuk menyimpan parameter dan arsitektur model untuk penggunaan di masa depan. Menyimpan model memungkinkan untuk kegunaan ulang, implementasi, dan penyetelan lebih lanjut. Kerangka kerja deep learning menyediakan fungsionalitas untuk menyimpan dan memuat model dalam berbagai format, seperti `state_dict` PyTorch atau format `SavedModel` TensorFlow. Model yang disimpan ini dapat dimuat kembali dan digunakan untuk prediksi atau pelatihan lanjutan pada tahap yang kemudian.