

TECHNICAL REPORT

MACHINE LEARNING FINAL EXAM

DEEP LEARNING WITH PYTHORCH

1. DEEP LEARNING

Deep learning adalah sebuah cabang dari machine learning yang berfokus pada pelatihan jaringan saraf tiruan (neural networks) untuk belajar dan membuat prediksi atau keputusan. Konsep deep learning terinspirasi dari struktur dan fungsi otak manusia, khususnya jaringan saraf yang saling terhubung.

Dalam deep learning, kita melatih jaringan saraf tiruan yang memiliki beberapa lapisan. Jaringan ini terdiri dari node-node yang saling terhubung, yang disebut juga neuron atau unit buatan, yang tersusun dalam lapisan-lapisan. Lapisan input menerima data masukan, sedangkan lapisan output menghasilkan keluaran atau prediksi yang diinginkan. Lapisan-lapisan di antara keduanya, yang sering disebut sebagai lapisan tersembunyi, bertugas untuk mengekstraksi dan mentransformasi data masukan melalui serangkaian operasi matematika.

Salah satu hal penting dalam deep learning adalah kemampuannya untuk secara otomatis belajar representasi atau fitur-fitur dari data mentah, tanpa perlu melakukan rekayasa fitur secara eksplisit. Jaringan saraf dalam deep learning belajar representasi-representasi ini dengan cara menyesuaikan bobot-bobot dan bias-bias jaringan berdasarkan data pelatihan. Proses ini, yang dikenal dengan istilah backpropagation, menggunakan algoritma optimisasi untuk meminimalkan perbedaan antara keluaran yang diprediksi dan keluaran sebenarnya.

Deep learning telah menjadi sangat populer dan sukses dalam berbagai bidang, seperti pengenalan gambar, pemrosesan bahasa alami, pengenalan suara, dan sistem rekomendasi. Dalam tugas-tugas seperti deteksi objek, klasifikasi gambar, terjemahan mesin, dan sebagainya, deep learning sering kali memberikan hasil yang sangat baik.

Namun, deep learning juga memiliki tantangan dan kompleksitas tersendiri. Melatih jaringan saraf yang dalam membutuhkan jumlah data yang besar dan komputasi yang intensif. Overfitting, di mana model memiliki performa yang baik pada data pelatihan tetapi gagal dalam menggeneralisasi pada data yang belum pernah dilihat sebelumnya, juga perlu diwaspadai dan diatasi dengan menggunakan teknik seperti regularisasi dan dropout.

Secara keseluruhan, deep learning telah membawa kemajuan signifikan dalam bidang kecerdasan buatan (artificial intelligence) dan telah membuka pintu bagi pengembangan sistem yang lebih cerdas.

2. PYTORCH

PyTorch adalah sebuah framework atau perpustakaan (library) dalam bahasa pemrograman Python yang digunakan untuk komputasi ilmiah dan pengembangan model deep learning. Dalam konteks deep learning, PyTorch memberikan alat dan fungsionalitas yang kuat untuk membuat, melatih, dan menganalisis jaringan saraf tiruan.

Salah satu fitur utama PyTorch adalah tensor, yang merupakan struktur data mirip array multidimensi. Tensor dalam PyTorch digunakan untuk merepresentasikan data dalam bentuk numerik, seperti gambar, teks, dan suara. PyTorch menyediakan operasi matematika dan algoritma yang efisien untuk memanipulasi tensor ini, termasuk dukungan untuk komputasi GPU yang cepat.

Selain itu, PyTorch memiliki fitur autograd yang kuat. Autograd memungkinkan perhitungan otomatis dari gradien (turunan) dari suatu fungsi terhadap parameter yang dapat diubah. Ini memudahkan dalam proses backpropagation, di mana gradien dihitung secara otomatis untuk mengoptimasi model deep learning.

PyTorch juga menyediakan modul `torch.nn` yang memudahkan dalam mendefinisikan arsitektur jaringan saraf tiruan (neural network). Modul ini memiliki berbagai jenis lapisan (layer) seperti lapisan linier (linear layer), lapisan konvolusi (convolutional layer), dan lapisan rekurensi (recurrent layer) yang digunakan dalam pembangunan model deep learning.

Keunggulan PyTorch juga terletak pada kejelasan dan kemudahan pemahaman kode. Kode dalam PyTorch biasanya lebih ekspresif dan mudah dibaca, sehingga memudahkan pengembangan, pemecahan masalah, dan eksperimen dengan model deep learning.

PyTorch memiliki komunitas yang aktif dengan banyak sumber daya, tutorial, dan dokumentasi yang tersedia secara online. Hal ini membuat PyTorch menjadi pilihan populer bagi para peneliti, praktisi, dan penggemar deep learning untuk mengembangkan dan melatih model deep learning dengan lebih efektif.

3. ANALISA KODE

a. AutoGrad

PyTorch's autograd adalah package yang menyediakan diferensiasi otomatis untuk semua operasi pada tensor. Ketika kita mendefinisikan tensor dengan `requires_grad=True`, PyTorch akan melacak semua operasi yang dilakukan pada tensor tersebut. Ini memungkinkan kita untuk menghitung gradien tensor terhadap suatu fungsi dengan mudah.

Setiap tensor yang dihasilkan dari operasi memiliki atribut `grad_fn`, yang merujuk pada fungsi yang digunakan untuk menghasilkan tensor tersebut.

Gradien dihitung secara otomatis dengan menggunakan backpropagation saat kita memanggil `backward()` pada tensor hasil yang ingin kita hitung gradiennya.

Gradien dari tensor diakumulasikan dalam atribut `.grad` dari tensor tersebut. Dalam kode tersebut, kita dapat melihat contoh penggunaan `backward()` dan mencetak gradien tensor menggunakan `.grad`.

b. Back Propagation

Backpropagation adalah algoritma pembelajaran dalam jaringan saraf yang digunakan untuk menghitung gradien fungsi kerugian terhadap parameter-model. Algoritma ini bekerja dengan langkah-langkah sebagai berikut:

- Feedforward: Data input diteruskan melalui jaringan saraf untuk menghasilkan prediksi.
- Perhitungan Kerugian: Menghitung tingkat kesalahan antara prediksi dengan target menggunakan fungsi kerugian.
- Backward Pass: Menghitung gradien fungsi kerugian terhadap parameter-model dengan menerapkan aturan rantai. Gradien diteruskan mundur melalui layer-layer jaringan dengan mengalikan gradien pada layer berikutnya dengan bobot yang terkait.
- Update Parameter: Bobot dan bias diperbarui menggunakan algoritma optimisasi berdasarkan gradien yang dihitung.

Backpropagation memungkinkan jaringan saraf untuk belajar dari data dengan memperbarui parameter-model berdasarkan gradien. Algoritma ini merupakan inti dari pelatihan dan optimisasi model jaringan saraf dalam deep learning

c. Gradient Descent

Gradient Descent adalah algoritma optimisasi yang mencari nilai minimum dari suatu fungsi dengan mengikuti arah gradien yang menurun. Langkah-langkahnya adalah menghitung gradien, memperbarui parameter-model, dan mengulanginya hingga konvergensi. Algoritma ini digunakan dalam machine learning untuk menemukan parameter-model yang optimal.

d. Training Pipeline

Training pipeline adalah serangkaian langkah dalam machine learning untuk melatih model. Langkah-langkahnya termasuk preprocessing data, membuat model, melatih model dengan data pelatihan, evaluasi kinerja model dengan data validasi, fine-tuning jika diperlukan, dan menyimpan model untuk penggunaan selanjutnya.

e. Linear Regression

Linear regression adalah metode untuk memodelkan hubungan linier antara variabel input dan output. Tujuannya adalah menemukan garis lurus terbaik yang mewakili data. Langkah-langkahnya meliputi persiapan data, memilih model, melatih model, dan melakukan prediksi pada data baru.

f. Logistic Regression

Logistic regression adalah metode untuk memodelkan hubungan antara variabel input dan probabilitas kategori. Output yang diprediksi adalah probabilitas menggunakan fungsi logistik (sigmoid). Langkah-langkahnya meliputi persiapan data, memilih model, melatih model, dan melakukan prediksi.

g. Data Loader

Data loader adalah komponen dalam machine learning yang memungkinkan memuat data dalam batch ke dalam model. Ini membantu pengolahan data besar, melibatkan pembuatan data loader, iterasi melalui data loader, dan pemrosesan data dalam batch.

h. Dataset Transform

Dataset transform adalah proses mengubah atau memodifikasi dataset dengan tujuan persiapan sebelum digunakan dalam pelatihan atau pengujian model. Transformasi ini dapat berupa perubahan ukuran, normalisasi, augmentasi, atau operasi lainnya tergantung kebutuhan. Dataset transform memungkinkan data yang lebih siap dan sesuai dengan model yang akan digunakan.

i. Softmax Cross-Entropy

Softmax Cross-Entropy adalah fungsi loss yang digunakan dalam klasifikasi multikelas. Fungsi ini mengukur perbedaan antara distribusi probabilitas prediksi dan distribusi probabilitas target yang sebenarnya. Tujuannya adalah untuk mengoptimalkan model agar prediksi mendekati target yang sebenarnya.

j. Activation Function

Activation function adalah fungsi matematika yang digunakan dalam jaringan saraf buatan untuk memperkenalkan non-linearitas. Fungsi ini membantu model untuk mempelajari hubungan yang kompleks antara input dan output. Beberapa contoh activation function umum adalah ReLU, sigmoid, tanh, dan softmax.

k. Feed Forward Net

Feedforward neural network adalah jenis jaringan saraf buatan yang sinyalnya hanya mengalir maju dari input ke output. Jaringan ini terdiri dari lapisan-lapisan neuron yang terhubung secara sekuensial. Data masukan diberikan ke lapisan masukan, kemudian melalui lapisan-lapisan tersembunyi, dan menghasilkan output di lapisan keluaran. Setiap neuron di lapisan tersembunyi menggunakan fungsi aktivasi untuk memperkenalkan non-linearitas. Feedforward neural network digunakan dalam berbagai tugas seperti klasifikasi dan regresi.

l. Convolutional Neural Network

CNN (Convolutional Neural Network) adalah jenis jaringan saraf buatan yang paling umum digunakan dalam pengolahan citra dan pengenalan gambar. CNN menggunakan lapisan konvolusi untuk mengekstraksi fitur-fitur penting dari input, diikuti oleh lapisan pooling untuk mengurangi ukuran spasial. Kemudian, fitur-fitur tersebut dihubungkan dengan lapisan terhubung penuh untuk klasifikasi atau regresi. CNN menggunakan fungsi aktivasi non-linear untuk memperkenalkan non-linearitas.

CNN efektif dalam mengolah data spasial dan telah memberikan hasil yang baik dalam tugas-tugas pengenalan gambar.

m. Transfer Learning

Transfer learning adalah teknik dalam pembelajaran mesin di mana kita menggunakan model yang sudah dilatih pada tugas lain sebagai titik awal untuk tugas yang baru. Dengan transfer learning, kita dapat memanfaatkan pengetahuan yang sudah ada dari model yang telah dilatih sebelumnya untuk mempercepat dan meningkatkan kinerja model pada tugas baru. Ini sangat berguna ketika kita memiliki sedikit data pelatihan untuk tugas baru atau ketika tugas baru memiliki kesamaan dengan tugas sebelumnya. Dengan menggunakan transfer learning, kita dapat menghemat waktu dan sumber daya yang dibutuhkan untuk melatih model dari awal dan mencapai hasil yang baik lebih cepat.

n. Tensor Board

TensorBoard adalah alat visualisasi yang digunakan dalam TensorFlow untuk memantau dan menganalisis model. Ini membantu pengembang dalam memahami dan mengoptimalkan model dengan melihat grafik arsitektur, metrik pelatihan, distribusi bobot, dan visualisasi data. TensorBoard juga mendukung perbandingan eksperimen dan analisis yang mendalam. Dengan TensorBoard, pengembang dapat mengoptimalkan model mereka dengan lebih baik.