

TECHNICAL REPORT UAS MACHINE LEARNING

“DEEP LEARNING WITH PYTORCH”



Disusun oleh:

Fariz Rahman Ramadhan
1103204046

PROGRAM STUDI TEKNIK KOMPUTER
FAKULTAS TEKNIK ELEKTRO
UNIVERSITAS TELKOM
2023

PENDAHULUAN

Deep learning merupakan salah satu cabang dari machine learning yang menggunakan arsitektur jaringan saraf tiruan (neural networks) yang dalam beberapa tahun terakhir telah mengalami perkembangan pesat dan menjadi salah satu metode yang paling populer dalam memecahkan berbagai permasalahan di bidang pengenalan pola, pengolahan bahasa alami, visi komputer, dan lain sebagainya. Salah satu framework deep learning yang sangat populer adalah PyTorch.

PyTorch adalah sebuah library open-source yang dikembangkan oleh Facebook's AI Research lab. PyTorch menyediakan kemudahan dalam membangun dan melatih jaringan saraf tiruan dengan menggunakan tensor yang efisien dan fleksibel. Dalam laporan technical report ini, kami akan membahas penggunaan PyTorch dalam implementasi beberapa konsep penting dalam deep learning.

TENSOR BASIC

Pada bagian ini, kami menggunakan konsep tensor dalam PyTorch. Tensor adalah struktur data fundamental dalam PyTorch yang mirip dengan array multidimensional. Kami menjelaskan penggunaan tensor dalam menyimpan dan memanipulasi data numerik dalam konteks deep learning. Kode menggunakan NumPy untuk membuat tensor numpy array. Kemudian, softmax function diterapkan pada array untuk menghasilkan probabilitas. Selanjutnya kode menggunakan PyTorch untuk membuat tensor dari array. Kemudian, softmax function diterapkan pada tensor menggunakan `torch.softmax`.

AUTOGRAD

Autograd adalah modul dalam PyTorch yang mengimplementasikan algoritma diferensiasi otomatis. Dalam deep learning, kita perlu menghitung gradien (turunan) dari fungsi loss terhadap parameter-parameter model. Autograd memungkinkan kita untuk menghitung gradien secara otomatis dengan membangun graph komputasi dan melacak setiap operasi pada tensor. Kode menunjukkan penggunaan autograd di PyTorch. Pada kode, perhitungan gradien dilakukan secara manual menggunakan NumPy. Selanjutnya kode, autograd di PyTorch digunakan untuk secara otomatis menghitung dan melacak gradien pada tensor.

BACK PROPAGATION

Back propagation adalah algoritma yang digunakan untuk menghitung gradien dari fungsi loss terhadap setiap parameter dalam jaringan saraf. Dalam bagian ini, kami menjelaskan konsep dasar back propagation dan bagaimana PyTorch secara otomatis melakukan perhitungan gradien menggunakan autograd. Kode menunjukkan contoh sederhana back propagation secara manual menggunakan NumPy. Selanjutnya kode menggunakan PyTorch dan autograd untuk melaksanakan back propagation secara otomatis dengan menghitung gradien dari fungsi loss terhadap parameter-parameter model.

GRADIENT DESCENT

Gradient descent adalah algoritma optimisasi yang digunakan untuk memperbarui parameter-parameter model berdasarkan gradien dari fungsi loss. Pada bagian ini, kami menunjukkan implementasi gradient descent secara manual menggunakan PyTorch. Kami menghitung gradien secara eksplisit dan memperbarui parameter-parameter model. Selain melakukan gradient descent secara manual, PyTorch juga menyediakan optimizers yang dapat secara otomatis menghitung dan memperbarui gradien. Pada bagian ini, kami menggunakan optimizers yang disediakan oleh PyTorch, seperti stochastic gradient descent (SGD), untuk melakukan optimisasi model secara efisien.

LOSS AND OPTIMIZER

Loss function adalah fungsi yang digunakan untuk mengukur kesalahan prediksi model. Pada bagian ini, kami menjelaskan penggunaan loss function dalam deep learning, seperti mean squared error (MSE) loss dan binary cross entropy (BCE) loss. Kami juga menjelaskan penggunaan optimizer, seperti stochastic gradient descent (SGD), untuk mengoptimasi model berdasarkan nilai loss. Kode menunjukkan penggunaan loss function MSE dan optimizers SGD secara manual menggunakan NumPy. Selanjutnya kode menggunakan PyTorch dan menyediakan loss function seperti MSE dan optimizers seperti SGD, yang mempermudah penggunaan dalam pelatihan model.

LINEAR PROPAGATION

Linear regression adalah metode statistik yang digunakan untuk memodelkan hubungan linier antara variabel input dan variabel target. Pada bagian ini, kami menjelaskan penggunaan PyTorch untuk implementasi linear regression. Kami menggunakan model jaringan saraf dengan satu layer linear untuk memprediksi nilai target berdasarkan variabel input. Kode menggunakan NumPy untuk melakukan linear regression secara manual. Selanjutnya kode menggunakan PyTorch untuk mengimplementasikan linear regression menggunakan model jaringan saraf dengan satu layer linear.

LOGISTIC REGRESSION

Logistic regression adalah metode statistik yang digunakan untuk masalah klasifikasi biner. Pada bagian ini, kami menjelaskan penggunaan PyTorch untuk implementasi logistic regression. Kami menggunakan model jaringan saraf dengan satu layer linear dan fungsi aktivasi sigmoid untuk memprediksi probabilitas kelas positif. Kode menunjukkan penggunaan PyTorch untuk implementasi logistic regression. Model jaringan saraf dengan satu layer linear dan fungsi aktivasi sigmoid digunakan untuk memprediksi probabilitas kelas positif.

DATALOADER

Dalam deep learning, dataset yang besar seringkali dibagi menjadi batch-batch kecil untuk mempercepat proses training. Dataloader adalah komponen dalam PyTorch yang digunakan untuk memuat dan mengelola data dalam batch. Pada bagian ini, kami menjelaskan penggunaan dataloader untuk memuat dan membagi dataset menjadi batch-batch dalam proses pelatihan model. Kode

menunjukkan penggunaan dataloader di PyTorch untuk memuat dan membagi dataset menjadi batch-batch dalam pelatihan model. Dataset dibagi menjadi batch-batch kecil untuk mempercepat proses training.

TRANSFORMERS

Transformers adalah arsitektur jaringan saraf yang sangat populer dalam pemrosesan bahasa alami. Pada bagian ini, kami memberikan penjelasan singkat tentang transformers dan bagaimana PyTorch menyediakan modul dan fungsionalitas untuk mengimplementasikan transformers. Kami juga menjelaskan kemampuan transformers dalam memodelkan hubungan konteks dalam teks.

SOFTMAX AND CROSSENTROPY

Softmax function adalah fungsi yang mengonversi nilai numerik menjadi distribusi probabilitas. Fungsi ini umumnya digunakan pada output layer dalam permasalahan klasifikasi multikelas. Pada contoh kode, softmax function diimplementasikan secara manual menggunakan numpy untuk menghasilkan distribusi probabilitas dari nilai numerik pada array input. Crossentropy loss function digunakan untuk mengukur kesalahan antara probabilitas prediksi yang dihasilkan oleh softmax function dan label yang sebenarnya. Pada contoh kode, crossentropy loss dihitung secara manual menggunakan numpy berdasarkan probabilitas prediksi dan label yang diberikan.

ACTIVATION FUNCTIONS

Activation function adalah fungsi matematika yang diterapkan pada output neuron untuk memperkenalkan non-linearitas dalam model jaringan saraf. Pada contoh kode, ReLU (Rectified Linear Unit) digunakan sebagai activation function pada layer tersembunyi. Fungsi ini menghasilkan nilai nol untuk input negatif dan mempertahankan nilai input positif. Sigmoid function, yang juga dikenal sebagai logistic function, digunakan sebagai activation function pada output layer dalam permasalahan logistik regression. Fungsi ini menghasilkan output antara 0 dan 1, yang dapat diinterpretasikan sebagai probabilitas kelas positif.

PLOT ACTIVATIONS

Pada contoh kode, scatter plot digunakan untuk memvisualisasikan data input (`X_numpy`) dan aktivasi output pada model linear regression. Dengan menggunakan matplotlib, data input ditampilkan sebagai titik-titik merah, sedangkan aktivasi output (predicted) ditampilkan sebagai garis biru.

CONVOLUTIONAL NEURAL NETWORK (CNN)

Convolutional Neural Network (CNN) adalah jenis arsitektur jaringan saraf yang secara khusus dirancang untuk pengolahan citra dan tugas-tugas vision. CNN menggunakan layer-layer konvolusi untuk mengekstraksi fitur dari citra secara hierarkis, diikuti oleh layer-layer pooling untuk mengurangi dimensi dan mempertahankan fitur-fitur penting. PyTorch menyediakan modul dan fungsionalitas

untuk membangun CNN dengan mudah, termasuk konvolusi 2D, max pooling, dan fully connected layer.

TRANSFER LEARNING

Transfer learning adalah pendekatan di mana model yang telah dilatih sebelumnya pada tugas yang serupa digunakan sebagai dasar untuk mempelajari tugas baru. Dalam transfer learning, pengetahuan yang telah terakumulasi dalam model yang telah dilatih sebelumnya dapat dimanfaatkan untuk meningkatkan performa dan efisiensi model pada tugas yang baru. PyTorch mendukung transfer learning dengan menggunakan model-model yang telah dilatih sebelumnya, seperti yang tersedia dalam torchvision. Model-model ini dapat diunduh dan digunakan sebagai dasar untuk membangun model baru atau untuk mengekstraksi fitur-fitur yang relevan.

TENSORBOARD

Tensorboard adalah alat visualisasi yang disediakan oleh TensorFlow untuk memantau dan menganalisis performa model jaringan saraf. Dalam contoh kode yang diberikan, tidak terdapat penggunaan Tensorboard. Namun, PyTorch juga memiliki alat visualisasi sendiri seperti TorchMetrics, TensorboardX, atau PyTorch Lightning yang dapat digunakan untuk tujuan yang serupa.

KESIMPULAN

Dalam laporan ini, kita telah mempelajari dasar-dasar deep learning dengan menggunakan PyTorch. Mulai dari konsep tensor dan autograd, hingga pelatihan model menggunakan backpropagation dan optimasi dengan gradient descent. Kita juga menjelaskan penggunaan loss function, optimizer, serta contoh aplikasi linear regression dan logistik regression. Selain itu, kita membahas konsep penting lainnya seperti softmax, crossentropy, activation functions, plotting activations, Convolutional Neural Networks (CNN), Transfer Learning, Tensorboard, dan Save/Load model. Semua ini memberikan pemahaman yang komprehensif tentang pengembangan dan pemahaman deep learning. Dengan PyTorch, kita dapat dengan mudah membangun, melatih, dan menganalisis model jaringan saraf untuk berbagai tugas dan domain. Laporan ini diharapkan dapat menjadi panduan yang bermanfaat bagi praktisi dan peneliti dalam mempelajari dan menerapkan deep learning menggunakan PyTorch.