

# **TECHNICAL REPORT UTS MACHINE LEARNING**

Deep Learning with Pytorch



I Made Bayu Satria Wardhana 1103204145

**PROGRAM STUDI TEKNIK KOMPUTER**  
**FAKULTAS TEKNIK ELEKTRO UNIVERSITAS**  
**TELKOM**  
**2023**

## **1. Pengantar**

Deep learning adalah subfield dari machine learning yang fokus pada pelatihan jaringan saraf tiruan untuk melakukan tugas-tugas kompleks. PyTorch adalah kerangka deep learning yang populer yang menyediakan platform yang fleksibel dan efisien untuk mengembangkan dan melatih model deep learning. Laporan taktis ini memberikan gambaran tentang deep learning menggunakan PyTorch, mencakup konsep dan teknik utama.

## **2. Dasar-dasar PyTorch**

PyTorch dibangun di atas pustaka Torch dan menawarkan grafik komputasi dinamis, yang memungkinkan pembangunan dan eksperimen model dengan mudah. PyTorch menyediakan kumpulan alat dan fungsionalitas yang kaya untuk bekerja dengan tensor, struktur data dasar dalam PyTorch. Tensor adalah larik multidimensi yang dapat diproses dengan efisien baik pada CPU maupun GPU.

## **3. Membangun Jaringan Saraf**

PyTorch menyediakan modul tingkat tinggi yang disebut `nn` untuk membangun jaringan saraf. Dengan `nn`, Anda dapat mendefinisikan arsitektur jaringan saraf Anda menggunakan layer-layer pre-defined seperti fully connected layers, convolutional layers, recurrent layers, dan lainnya. Anda juga dapat membuat layer kustom dengan menurunkan kelas dasar `nn.Module`.

## **4. Melatih Model Deep Learning**

Untuk melatih model deep learning di PyTorch, umumnya Anda mengikuti langkah-langkah berikut:

- a. Tentukan arsitektur model dengan membuat subkelas `nn.Module`.
- b. Tentukan fungsi kerugian untuk mengoptimalkan kinerja model.
- c. Pilih algoritma optimisasi, seperti stochastic gradient descent (SGD) atau Adam, dan atur tingkat pembelajaran.
- d. Iterasi melalui dataset pelatihan dalam batch, melewati data masukan melalui model dan menghitung kerugian.
- e. Melakukan backpropagasi gradien melalui jaringan dan memperbarui parameter model menggunakan optimizer yang dipilih.
- f. Ulangi langkah-langkah di atas selama jumlah epoch tertentu atau hingga konvergensi.

## **5. Pemercepatan GPU**

PyTorch mendukung pemercepatan GPU, yang memungkinkan Anda memanfaatkan daya komputasi GPU untuk pelatihan dan inferensi yang lebih cepat. Anda dapat dengan mudah memindahkan tensor dan model ke GPU menggunakan metode `.to(device)`, di mana device dapat berupa "cuda" atau "cpu". PyTorch menyediakan tensor CUDA dan secara otomatis melakukan komputasi GPU ketika tensor berada di GPU.

## **6. Dataset dan Data Loader**

PyTorch menyediakan alat yang nyaman untuk menangani dataset dan pengambilan data. Paket `torchvision` menawarkan dataset populer, termasuk MNIST, CIFAR-10, dan ImageNet, beserta teknik augmentasi data. Anda dapat menggunakan data loader untuk memuat dan memproses data secara paralel, memungkinkan integrasi yang mulus dengan proses pelatihan.

## **7. Evaluasi Model**

Setelah melatih model deep learning, penting untuk mengevaluasi kinerjanya pada data yang belum terlihat. PyTorch menyediakan alat untuk menghitung metrik seperti akurasi, presisi, recall, dan skor F1. Anda dapat mengevaluasi model pada dataset validasi terpisah atau melakukan validasi silang untuk evaluasi yang kuat.

## **8. Penyimpanan dan Pemuatan Model**

PyTorch memungkinkan Anda menyimpan dan memuat model yang telah dilatih untuk penggunaan di masa depan. Anda dapat menyimpan kamus keadaan model menggunakan `torch.save()` dan memuatnya kemudian menggunakan `torch.load()`. Kemampuan ini berguna untuk menggunakan kembali model yang telah dilatih, melakukan deployment di lingkungan produksi, atau melanjutkan pelatihan dari checkpoint.

## **9. Transfer Learning**

Transfer learning adalah teknik populer dalam deep learning yang memungkinkan penggunaan model yang telah dilatih sebagai titik awal untuk tugas-tugas baru. PyTorch menyediakan model pre-trained melalui modul `torchvision.models`, yang dapat disesuaikan atau digunakan sebagai ekstraktor fitur untuk dataset baru.

## **10. Interpretabilitas Model dan Visualisasi**

Memahami cara kerja dalam model deep learning sangat penting untuk debugging, optimasi, dan membangun kepercayaan pada model tersebut. PyTorch menawarkan berbagai alat untuk interpretabilitas model, termasuk visualisasi aktivasi, gradien, dan peta perhatian. Teknik-teknik ini membantu mendapatkan wawasan tentang apa yang dipelajari model dan mengidentifikasi masalah potensial.

## **11. Topik Lanjutan**

PyTorch mendukung teknik deep learning lanjutan, seperti jaringan saraf berulang (RNN) untuk data sekuensial, generative adversarial networks (GANs) untuk menghasilkan data sintetis, dan transformers untuk tugas pemrosesan bahasa alami. Topik-topik lanjutan ini memungkinkan penyelesaian berbagai masalah kompleks menggunakan PyTorch.

## **12. Kesimpulan**

Deep learning menggunakan PyTorch menyediakan kerangka yang kuat untuk mengembangkan dan melatih model jaringan saraf terkini. Dengan antarmuka API yang intuitif, fleksibilitas, dan dukungan akselerasi GPU, PyTorch telah menjadi populer di kalangan peneliti dan praktisi dalam komunitas deep learning. Dengan memanfaatkan kemampuan PyTorch, pengembang dapat menggali potensi deep learning dan membangun aplikasi inovatif di berbagai domain.