

Nama: Pratyenggo Damar I.P.

NIM: 1103194141

## **Technical Report: PyTorch Workflow Fundamentals**

### **1. Membuat Data**

Membuat data sintetis dilakukan dengan menggunakan parameter yang diketahui, seperti weight dan bias. PyTorch digunakan untuk menciptakan deret bilangan sebagai fitur (X) dan menghitung nilai label (y) berdasarkan hubungan linear yang ditentukan.

### **2. Membagi Data**

Data dibagi menjadi dua set: pelatihan (training) dan pengujian (testing) dengan rasio 80:20. Pemisahan ini krusial agar model dapat mempelajari pola dari data pelatihan dan diuji pada data yang belum pernah dilihat sebelumnya.

### **3. Membangun Model**

Model regresi linear dikonstruksi menggunakan PyTorch. Dalam konteks ini, model memiliki dua parameter, yaitu weight dan bias, yang akan disesuaikan selama pelatihan. Kedua parameter ini diinisialisasi dengan nilai acak.

### **4. Memeriksa Parameter Model**

Parameter awal model, seperti weight dan bias, diperiksa menggunakan fasilitas PyTorch.

### **5. Melakukan Prediksi**

Model yang belum dilatih digunakan untuk membuat prediksi terhadap data pengujian. Namun, hasil prediksi belum akurat karena model masih menggunakan parameter acak.

### **6. Visualisasi Prediksi**

Visualisasi dilakukan dengan menggunakan fungsi `plot_predictions`. Grafik menampilkan data pelatihan (biru), data pengujian (hijau), dan prediksi model (merah). Pada tahap ini, prediksi model terlihat suboptimal karena parameter model masih bersifat acak.

### **7. Pelatihan Model**

Fungsi kerugian dan optimizer dibuat untuk melatih model. Fungsi kerugian mengukur sejauh mana prediksi model berkorelasi dengan label sebenarnya, sementara optimizer bertanggung jawab untuk memperbarui parameter model agar nilai kerugian berkurang. Proses pelatihan dilakukan melalui iterasi forward pass, perhitungan loss, backpropagation, dan pembaharuan parameter model. Model dilatih selama 100 epoch dan diuji setiap 10 epoch.

## 8. Menyimpan dan Memuat Model

Model yang telah dilatih disimpan menggunakan **torch.save** dan `state_dict()`. Model baru dibuat, dan `state_dict()` dari model yang disimpan dimuat ke model baru menggunakan **load\_state\_dict**.

## 9. Melakukan Inference

Model yang telah dilatih dan dimuat digunakan untuk membuat prediksi terhadap data pengujian. Prediksi dari model awal dan model yang dimuat dibandingkan untuk memastikan konsistensi.

## 10. Kode Agnostik Perangkat

Dalam praktiknya, kode bersifat agnostik perangkat, dapat menggunakan GPU jika tersedia, dan jika tidak, akan kembali ke CPU. Import library standar seperti PyTorch dan matplotlib dilakukan, diikuti dengan penentuan perangkat "cuda" jika GPU tersedia, atau "cpu" jika tidak.

## Kesimpulan

Melalui laporan ini, kita mengikuti langkah-langkah dasar dalam alur kerja PyTorch untuk membuat, melatih, menyimpan, dan memuat model regresi linear. Dengan pendekatan yang agnostik perangkat, kode dapat berjalan efisien di berbagai platform. Proses ini menunjukkan kemampuan PyTorch dalam mempermudah implementasi dan eksperimen pada model machine learning.