

Nama : Arfara Yema Samgusdian

NIM : 1103202004

Kelas : TK-44-04

Zero to Mastery Learn PyTorch for Deep Learning

01. PyTorch Workflow Fundamentals

Membuat Data

- Data sintetis dibuat dengan menggunakan parameter yang diketahui (weight dan bias).
- Melibatkan penggunaan PyTorch untuk membuat deret bilangan sebagai fitur (X) dan menghitung nilai label (y) berdasarkan hubungan linear yang ditentukan.

Membagi Data

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

Membangun Model

- Model regresi linear dibangun menggunakan PyTorch.
- Model ini memiliki dua parameter yang akan disesuaikan selama pelatihan: weight dan bias.
- Dalam model ini, weight dan bias diinisialisasi dengan nilai acak.

Memeriksa Parameter Model

- Parameter awal model, yaitu weight dan bias, diperiksa menggunakan PyTorch.

Melakukan Prediksi

- Model yang belum dilatih digunakan untuk membuat prediksi terhadap data pengujian.
- Hasil prediksi ini belum akurat karena model masih menggunakan parameter acak.

Visualisasi Prediksi

- Visualisasi dilakukan menggunakan fungsi `plot_predictions`, yang menampilkan data pelatihan (biru), data pengujian (hijau), dan prediksi model (merah).
- Pada tahap ini, prediksi model tampak buruk karena parameter model masih acak.

Pelatihan Model

- Loss function (fungsi kerugian) dan optimizer dibuat untuk melatih model.
- Loss function digunakan untuk mengukur seberapa salah prediksi model dibandingkan dengan label sebenarnya.
- Optimizer digunakan untuk memperbarui parameter internal model agar dapat mengurangi nilai loss.

- Training loop diimplementasikan untuk iteratif melakukan forward pass, perhitungan loss, backpropagation, dan pembaharuan parameter model.
- Model dilatih selama 100 epoch dan diuji setiap 10 epoch.

Menyimpan dan Memuat Model

- Model yang telah dilatih disimpan dengan menyimpan `state_dict()` menggunakan `torch.save`.
- Model baru dibuat, dan `state_dict()` dari model yang disimpan dimuat ke model baru menggunakan `load_state_dict`.

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.
-

Dalam praktiknya, kita membuat kode yang bersifat agnostik perangkat, sehingga dapat menggunakan GPU jika tersedia, dan jika tidak, akan kembali ke CPU.

Kita mulai dengan mengimpor library standar yang diperlukan, seperti PyTorch dan matplotlib. Selanjutnya, kita membuat kode yang dapat berjalan pada perangkat apa pun dengan menentukan perangkat "cuda" jika GPU tersedia, atau "cpu" jika tidak.

Berikutnya, kita membuat data untuk pelatihan dan pengujian seperti sebelumnya. Setelah itu, kita membagi data menjadi set pelatihan dan pengujian dengan rasio 80/20.

Kemudian, kita membuat model regresi linear menggunakan `nn.Linear` untuk mendefinisikan parameter berat dan bias secara otomatis. Model tersebut ditempatkan pada perangkat yang sesuai (GPU atau CPU).

Langkah selanjutnya adalah menentukan fungsi kerugian dan optimizer, seperti sebelumnya. Kita melakukan pelatihan model dengan menggunakan data pada perangkat yang sesuai, baik GPU atau CPU.

Setelah pelatihan selesai, kita mengevaluasi hasilnya dan memeriksa parameter yang telah dipelajari oleh model. Hasilnya cukup baik, dan parameter-model mendekati nilai yang kita tetapkan secara manual.

Terakhir, kita membuat prediksi menggunakan model yang telah dilatih dan menyimpan model tersebut untuk penggunaan di masa mendatang. Proses menyimpan dan memuat model berhasil, dan model yang dimuat memberikan prediksi yang konsisten dengan model asli.