

Zero to Mastery Learn PyTorch for Deep Learning Course Summary

01. PyTorch Fundamentals

Nama : Ihsan Ridho Asmoro

NIM : 1103204184

Modul ini adalah awal yang bagus buat ngertiin dasar-dasar machine learning dan deep learning, pokoknya yang rumit-rumit gitu. Ide utamanya adalah pake data dari masa lalu buat bikin algoritma, kayak yang namanya jaringan saraf atau neural network. Nah, algoritma ini bisa nemuin pola-pola dalam data, dan setelah ketemu, bisa dipake buat nebak-nebakin kejadian di masa depan.

Di dalam modul ini, kita bakal mulai dari yang simpel dulu, yaitu garis lurus. Terus, kita bakal belajar bikin model PyTorch yang bisa belajar dari pola garis lurus itu dan ngikutin polanya.

Ini dia langkah-langkahnya yang bakal kita bahas:

1. Siapin Data: Data bisa dari mana aja sih, mulai dari angka-angka, gambar, video, suara, sampe struktur protein dan tulisan. Nah, kita bakal mulai dari bikin data yang simpel aja dulu, kayak garis lurus.
2. Bikin Model: Nah, di sini kita bakal bikin model buat belajar dari data. Terus kita juga milih fungsi kerugian, optimizer, dan bikin loop buat training modelnya.
3. Setel Modelnya sama Data (Pelatihan): Udah punya data sama model, sekarang kita biarin modelnya nyari pola di data pelatihan.
4. Bikin Prediksi dan Nilai Modelnya (Inferensi): Model kita udah nemu pola di data, sekarang kita bandingin hasilnya sama data uji yang sesungguhnya.
5. Simpen dan Muat Model: Kalo misalnya nanti mau pake model ini lagi atau di tempat lain, kita bakal bahas cara nyimpen dan ngeload modelnya.
6. Gabungin Semua Langkah: Yuk, kita rangkum dan gabungin semua langkah di atas jadi satu kodingan. Pastiin kodingannya bisa dijalankan di GPU kalo ada, kalo enggak ya ke CPU aja.

Persiapan dan Muat Data:

Jadi, dalam machine learning, data itu bermacam-macam, mulai dari angka, gambar, video, sampe suara atau tulisan. Prosesnya melibatkan dua hal utama, yakni ngubah data jadi kumpulan angka yang bisa dimengerti dan pilih model buat belajar dari angka-angka itu. Nah, di modul ini, kita bakal mulai dari bikin data berbentuk garis lurus sebagai contoh. Caranya:

1. Atur Parameter buat Data Garis Lurus: Tentuin parameter buat data garis lurus.
2. Buat Data pake Linear Regression di PyTorch: Bikin data pake regresi linear di PyTorch.
3. Bagi Data buat Pelatihan dan Uji: Pisahin data jadi dua bagian, 80% buat pelatihan, 20% buat uji.
4. Visualisasi Data buat Paham Pola dan Hubungannya: Liat pola dan hubungan antara fitur (X) sama label (y).

Data yang udah disiapin dan digambarin bakal jadi dasar buat bangun model di langkah selanjutnya.

Bikin Model:

Di langkah ini, kita bakal bikin model regresi linear pake PyTorch. Modelnya bakal jadi kelas Python yang ngelakuin fungsi regresi linear simpel. Caranya:

Buat Model Linear pake PyTorch: Bikin model linear pake subclass `nn.Module` dan `nn.Linear()`.

Modelnya cuma punya satu lapisan linear dengan satu input dan satu output.

Latih Modelnya: Pake loss function `nn.L1Loss()` dan optimizer `torch.optim.SGD()`. Latih modelnya buat beberapa epoch dengan naro datanya di perangkat yang ada (GPU kalo ada, kalo enggak ya default ke CPU).

Bikin Prediksi dan Nilai Modelnya (Inferensi):

Setelah melatih model, langkah selanjutnya adalah bikin prediksi dengan model itu. Ini langkah-langkahnya:

1. Setel Modelnya jadi Mode Evaluasi: Pake fungsi `model.eval()` buat pastiin kalo model lagi di mode evaluasi, di mana beberapa fungsi yang enggak perlu pas pelatihan dimatikan.
2. Pake Konteks Inference Mode: Pake with `torch.inference_mode()`: buat bikin prediksi. Ini biar kalkulasi dilakuin dengan cepat dan efisien.
3. Pastiin Kalkulasi di Perangkat yang Sama: Pastiin model sama data ada di perangkat yang sama, entah itu CPU atau GPU.

Setelah langkah-langkah itu, kita bisa bikin prediksi pake model yang udah dilatih di data uji. Hasil prediksinya bisa dibandingin sama prediksi sebelumnya buat pastiin kalo modelnya bener-bener nyimpen dan ngebaca data dengan baik.

Simpan dan Muat Model PyTorch:

Setelah latih model, langkah selanjutnya adalah nyimpen modelnya buat dipake nanti. PyTorch punya beberapa cara buat nyimpen dan ngeload model, dan di bawah ini tiga metodenya:

1. `torch.save`: Simpen objek yang udah di-serialize ke disk pake utilitas pickle Python. Model, tensor, dan berbagai objek Python lainnya, kayak kamus, bisa disimpen pake `torch.save`.
2. `torch.load`: Pake fitur deserialisasi pickle buat ngeload file objek Python yang udah di-serialize (kayak model, tensor, atau kamus) ke dalam memori. Kita juga bisa pilih perangkat mana objeknya mau dimuat (CPU, GPU, dll).
3. `torch.nn.Module.load_state_dict`: Ngeload kamus parameter model (`model.state_dict()`) pake objek `state_dict()` yang udah disimpen.

Disarankan buat nyimpen dan ngeload model cuma pake `state_dict()`, soalnya lebih fleksibel dan nggak tergantung sama struktur kelas atau direktori yang spesifik.

1. menyimpan Model:
 - Buat Direktori buat Nyimpen Model (contohnya, "models").
 - Bikin Path buat Nyimpen Model.
 - Pake `torch.save(obj, f)` di mana `obj` itu `state_dict()` dari model target dan `f` itu nama file buat nyimpen model.

2. Ngeload Model:
 - Buat Instance baru dari Model buat Dimuat.
 - Ngeload `state_dict()` model yang udah disimpan pake `torch.load(f)` dan masukin ke instance model baru pake `loaded_model.load_state_dict()`.
3. Bikin Prediksi dengan Model yang Udah Dimuat:
 - Setel model yang udah dimuat ke mode evaluasi.
 - Pake konteks `inference_mode` buat bikin prediksi.

Setelah ngeload model, prediksinya yang dihasilkan bisa dibandingin sama prediksi sebelumnya buat pastiin kalo modelnya nyimpen dan ngeload dengan benar.

Gabungin Semua Langkah:

1. Siapin Data
 - a) Buat data dengan nentuin weight dan bias.
 - b) Buat rentang nilai antara 0 dan 1 buat X.
 - c) Pake X, weight, dan bias buat bikin label (y) pake rumus regresi linear ($y = \text{weight} * X + \text{bias}$).
 - d) Bagi data jadi set pelatihan dan pengujian (80/20 split).
2. Bikin Model Linear PyTorch
 - Bikin model linear pake subclass `nn.Module` dan `nn.Linear()`.
 - Modelnya cuma punya satu lapisan linear dengan satu input dan satu output.
3. Latih Model
 - Pake loss function `nn.L1Loss()` dan optimizer `torch.optim.SGD()`.
 - Latih model selama serangkaian epoch dengan naro data di perangkat yang tersedia (GPU kalo ada, kalo enggak default ke CPU).
4. Bikin Prediksi
 - Ubah model ke mode evaluasi.
 - Bikin prediksi di data pengujian.
5. Nyimpan dan Ngeload Model
 - Simpen model ke file pake `torch.save()` dengan ngikutin `state_dict()` model.
 - Ngeload model pake `torch.load()` dan `load_state_dict()`.
 - Evaluasi model yang udah di-load buat pastiin prediksinya sesuai dengan model sebelumnya.
6. Keseluruhan

Gabungin semua langkah di atas dalam satu kumpulan kode yang bisa dijalankan di GPU kalo ada, kalo enggak di CPU aja.