PyTorch Technical Report for Machine Learning Final Examination

Oleh :

Muthie Armalia Soeriamaritsa (1103200178)

PRODI S1 TEKNIK KOMPUTER
FAKULTAS TEKNIK ELEKTRO
UNIVERSITAS TELKOM
BANDUNG
2023

Bab 00: PyTorch Fundamental

00.1 Pengantar PyTorch

- Apa itu PyTorch?

  Pytorch adalah framework machine learning dan deep learning open source. PyTorch dapat digunakan untuk memanipulasi dan memproses data dan meembuat algoritma machine learning menggunakan Python. PyTorch sendiri sudah banyak digunakan di banyak Perusahaan teknologi besar seperti Meta (Facebook), Tesla, dan Mirosoft juga Perusahaan riset artificial intelligence seperti OpenAI menggunakan PyTorch untuk riset mereka dan menggunakan machine learning untuk produk mereka.

00.2 Tensor

- Apa itu Tensor?

  Tensor sendiri adalah elemen dasar dari machine learning. Tensor berguna untuk menyatakan data dalam bentuk numerik.

- Skalar

  Skalar adalah bilangan tunggal, dalam istilah tensor scalar adalah tensor berdimensi nol. Namun, scalar sendiri masih dapat diisi oleh sebuah nilai. Jika scalar di print pun akan mengeluarkan output sesuai isi dari nilai scalar tersebut. Dimensi tensor dapat dicel menggunakan atribut ‘ndim’. Jika ingin mengubah menjadi integer, gunakan metode ‘item()’.

- Vektor

  Vektor adalah tensor satu dimensi yang dapat menampung banyak angka. Misalnya, kita bisa membuat vektor [3, 2] untuk mendeskripsikan [kamar tidur, kamar mandi] di rumah. Atau kita bisa menggunakan [3, 2, 2] untuk mendeskripsikan [kamar tidur, kamar mandi, tempat parkir mobil] di rumah. Namun intinya, vector ini flekseibel sehingga dapat mewakilkan atau menyataka apa saja.

  Perlu diingat bahwa untuk mengetahui berapa banyak dimensi, bukan dari berapa banyak angka, namun dari berapa banyak isi elemen kurung siku dalam suatu tensor. Namun, untuk menyatakan ada berapa format angka dalam satu tensor menggunakan istilah shape.

- Nol dan Satu

  Kita dapat membuat tensor berisi hanya nilai nol dan tensor berisi hanya nilai satu dengan menggunakan t’torch.zeros()’ dan ‘torch.ones()’.

- Membuat range dalam tensor

  Untuk membuat range kita dapat menggunakan ‘torch.range(start, end, step)’

- Tipe Data

  Tensor memiliki banyak tipe data. Banyak nya tipe data ini bertujuan untuk komputasi yang presisi. Presisi yang dimaksud adalah seberapa detail digunakan untuk mendeskripsikan angka. Hal ini penting pada deep learning dan komputasi angka karena kita akan menggunakan banyak operasi. Semakin banyak detail yang harus dihitung, semakin banyak komputasi yang harus digunakan. Jadi, tipe data dengan presisi yange lebih rendah biasanya komputasi perhitungannya akan lebih cepat, tapi akan lebih kurang akurat.

- Mendapatkan informasi dari tensor

  shape - apa bentuk tensornya? (beberapa operasi memerlukan aturan bentuk tertentu)
  dtype - tipe data apa yang menyimpan elemen dalam tensor?
  device - di perangkat apa tensor disimpan? (biasanya GPU atau CPU)

- Memanipulasi tensor (oeprasi tensor)

  Pada deep learning, data apapun bentuknya (gambar, teks, video, audio, struktur protein, apapun itu) di representasikan menggunakan tensor. Jadi, nanti modelnya belajar dari menyelidiki tensor dengan melakukan serangkaian operasi untuk membuat

representasi dari pola input data. Operasi tersebut dapat berupa pertambahan, pengurangan, perkalian, pembagian, ataupun perkalian matriks.

Perlu diingat bahwa nilai tensor tidak akan berubah setelah operasi tersebut jika tensor tidak dideklarasikan ulang.

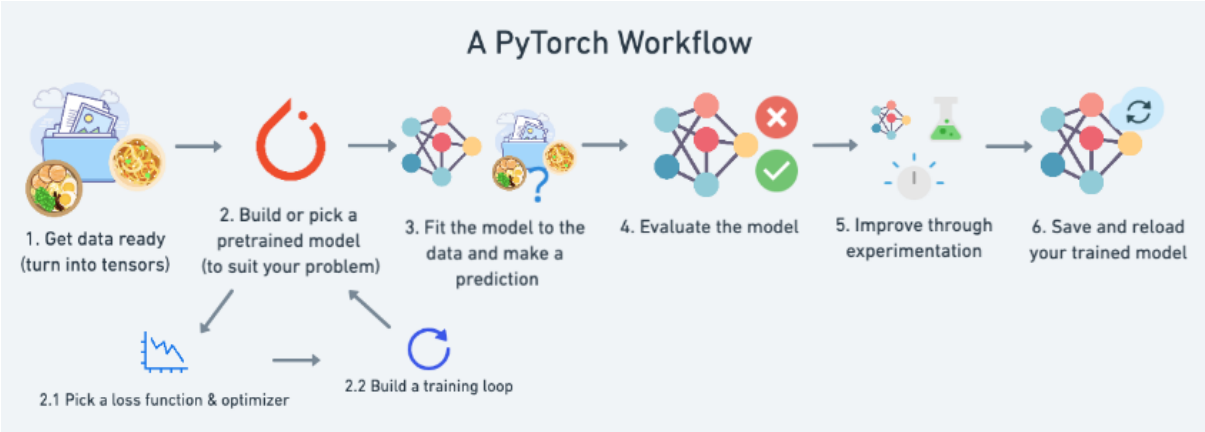
00.3 Error yang biasa ditemukan dalam deep learning

- Karena aturan tipe data yang ketat sering kali error disebabkan oleh bentuk yang tidak sesuai. Cara agar matriks sesuai salah satunya dapat dilakukan transpose.

Bab 01: PyTorch Workflow Fundamentals

01.1 PyTorch Workflow

Secara garis besar berikut workflow PyTorch:



Menyiapkan dan memuat data

- Data yang dimaksud dapat berupa apa saja (gambar, teks, numerik, apapun bentuk datanya)
- Machine learning terdiri dari dua hal utama, yaitu: merubah data apapun bentuknya menjadi bentuk angka, lalu yang kedua adalah membuat atau memilih model untuk mempelajari data tersebut.

Membuat model.

Melatih model

Menyimpan dan memuat model PyTorch

Bab 02: PyTorch Neural Network Classification

02.1 Pendahuluan

- Apa itu masalah klasifikasi?
  Masalah klasifikasi meliputi memprediksi apakah suatu benda adalah benda yang sama dengan yang lainnya. Jadi, masalah klasifikasi adalah bagaimana mengelompokan atau mengetahui apakah suatu benda adalah benda yang sama dengan benda lainnya atau tidak.

- Klasifikasi arsitektur jaringan saraf
  Berikut adalah arsitektur umum jaringan saraf:

Hiperparameter	Klasifikasi Biner	Klasifikasi multikelas
Bentuk lapisan masukan (`in_features`)	Sama dengan jumlah fitur (misalnya 5 untuk usia, jenis kelamin, tinggi badan, berat badan, status merokok dalam prediksi penyakit jantung)	Sama seperti klasifikasi biner
Lapisan tersembunyi	Khusus masalah, minimum = 1, maksimum = tidak terbatas	Sama seperti klasifikasi biner
Neuron per lapisan tersembunyi	Masalah spesifik, umumnya 10 hingga 512	Sama seperti klasifikasi biner
Bentuk lapisan keluaran (`out_features`)	1 (satu kelas atau lainnya)	1 per kelas (misalnya 3 untuk makanan, foto orang atau anjing)
Aktivasi lapisan tersembunyi	Biasanya ReLU (rectified linear unit) namun bisa juga banyak lainnya	Sama seperti klasifikasi biner
Aktivasi keluaran	Sigmoid (`torch.sigmoid` di PyTorch)	Softmax (`torch.softmax` di PyTorch)
Fungsi kerugian	Crossentropi biner (`torch.nn.BCELoss` di PyTorch)	Entropi silang (`torch.nn.CrossEntropyLoss` di PyTorch)
Pengoptimal	SGD (penurunan gradien stokastik), Adam (lihat `torch.optim` opsi selengkapnya)	Sama seperti klasifikasi biner

1. Buatlah dan siapkan data klasifikasi
   Kami akan menggunakan make_circles() metode dari Scikit-Learn untuk menghasilkan dua lingkaran dengan titik berwarna berbeda.
   a. Bentuk input dan output
   b. Ubah data menjadi tensor dan buat pemisahan pelatihan dan pengujian
2. Membangun model
   a. Mengatur fungsi kerugian dan pengoptimal
3. Model Kkereta api
   a. Beralih dari output model mentah ke label prediksi (logit -> peluang prediksi -> label prediksi)
   b. Membangun lingkaran pelatihan dan pengujian
4. Membuat prediksi dan mengevaluasi model
5. Memperbaiki model (dari perspektif model)
   a. Mempersiapkan data untuk melihat apakah model kita dapat memodelkan garis lurus
6. Bagian yang hilang: non-linearitas
   a. Membuat ulang data non-linier
   b. Membangun model dengan non-linieritas

    c. Melatih model dengan non-linieritas
    d. Mengevaluasi model yang dilatih dengan fungsi aktivasi non-linier
7. Mereplikasi fungus aktivasi nono-linier
8. Menyatukan berbagai hal dengan membangun model PyTorch multikelas (kelas yang digunakan lebih dari satu atau banyak kelas)
    a. Memnuat klasifikasi multikelas
    b. Membangun model klasifikasi kelas jamak di PyTorch
    c. Membuat fungsi kerugian dan pengoptimal untuk model PyTorch multikelas
    d. Mendapatkan probabilitas prediksi untuk model PTorch kelas jamak
    e. Membuat loop pelatihan dan pengujian untuk model PyTorch multikelas
    f. Membuat dan mengevaluasi prediksi dengan model kelas jamak PyTorch