

Praktikum Kecerdasan Buatan

Klasifikasi Image menggunakan CNN

1. Buat project baru pada google colaboratory. Beri nama `image_classification_cifar10.ipynb`. Kemudian ikuti langkah-langkah pada tutorial klasifikasi image dengan CNN menggunakan pytorch pada link berikut: https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html

2. Lakukan split data pada train set sehingga train set yang awalnya sebanyak 50000, terbagi menjadi train set 90% dan validation set 10%.

Hint:

- a. Gunakan seed yang sama, misalnya `seed = 43`

```
torch.manual_seed(43)
```

Ikuti semua langkah-langkah untuk menjaga reproducibility dari link berikut.

<https://discuss.pytorch.org/t/reproducibility-with-all-the-bells-and-whistles/81097>

Modifikasi dataloader, pastikan menggunakan seed yang sama, misalnya `seed=43`, seperti yang dijelaskan pada link berikut.

<https://pytorch.org/docs/stable/notes/randomness.html>

- b. Gunakan fungsi `random_split` berikut.

```
trainset, valset = random_split(trainset, [train_size, val_size])
```

- c. Jangan lupa untuk membuat variable `valloader` untuk memuat data dari validation set.

3. Modifikasi kode tersebut agar dijalankan menggunakan GPU.

Hint:

- a. Gunakan variable `device` untuk mengecek device yang available untuk digunakan saat ini (cek apakah GPU available?)

- b. Jangan lupa mengubah setting runtime pada google colaboratory agar kode program dijalankan menggunakan GPU.

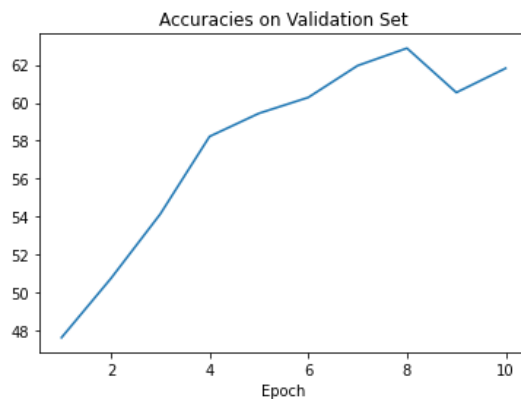
- c. Buat agar model, input dan target dijalankan pada device tersebut.

- d. Gunakan `.detach().cpu()` pada saat testing untuk mengkonversi kembali tensor ke array dari GPU ke CPU.

4. Perhatikan kode yang digunakan untuk membuat layer-layer pada CNN. Gambarkan arsitekturnya, beserta size dari setiap input dan outputnya.
5. Tambahkan kode untuk mengecek akurasi pada validation set untuk setiap epoch ketika proses training. Lalu buat line chart untuk menggambarkan akurasi pada validation set untuk setiap epoch menggunakan `matplotlib` library.

Hint:

- a. Buat variable `n_epoch` untuk menyimpan jumlah epoch yang dilakukan pada saat training. Kemudian sesuaikan proses training menggunakan variable tersebut.
- b. Gunakan kode seperti yang digunakan untuk menghitung akurasi pada testing set, tapi kali ini menggunakan validation set. Letakkan kode tersebut setelah selesai melakukan training setiap epoch.
- c. Berikut contoh output line chart yang diharapkan.



6. Perhatikan hasil akurasi pada validation set untuk setiap epoch. Kemudian gunakan model dengan epoch yang memberikan akurasi terbaik pada validation set, untuk kemudian digunakan untuk mengukur akurasi pada training set. Jangan lupa menyimpan output dari proses training dan testing sebagai laporan praktikum.
7. Modifikasi convolution (Conv2d) dengan ketentuan berikut:
 - a. Ubah parameter pada `conv1`, `out_channel = 24`. Jangan lupa sesuaikan dimensi pada `conv2` dan `fc1`. Restart runtime pada google colab, lalu jalankan kembali kode program dari awal. Lalu catat akurasi dari hasil prediksinya. Bandingkan dengan akurasi pada model original.
 - b. Ubah kembali parameter pada `conv1`, `out_channel = 24`, `stride = 2`, `padding = 1`. Jangan lupa sesuaikan dimensi pada `conv2` dan `fc1`. Restart runtime pada google colab, lalu jalankan kembali kode program dari awal. Lalu catat akurasi dari hasil prediksinya. Bandingkan dengan akurasi pada model original.
 - c. Ambil kesimpulan berdasarkan hasil perbandingan akurasi dari ketiga model tersebut (model original, model_3a, dan model_3b)

Responsi

Buat laporan praktikum yang terdiri dari arsitektur CNN, output proses training pada setiap epoch (loss training, akurasi validasi, line chart akurasi validasi) serta akurasi pada testing set untuk setiap model CNN (model original, model_3a, dan model_3b). Jangan lupa tuliskan model pada epoch berapa yang digunakan sebagai model terbaik.