

Laporan Tugas Besar 1
IF4074 Pembelajaran Mesin Lanjut
Convolutional Neural Network
Milestone A: Forward Propagation



Disusun Oleh Kelompok 17:

Zaidan Naufal Sudrajat / 13518021

Fauzan Yubairi Indrayadi / 13519171

M. Ibnu Syah Hafizh / 13519177

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2022

BAB I

Penjelasan Kode Program

Kode program dibagi menjadi empat file yang berbeda berdasarkan keperluannya masing-masing. File tersebut terdiri dari `common.py` yang berisi fungsi aktivasi seperti ReLU dan sigmoid, `generateImage.py` berisi fungsi untuk men-generate matriks dari image, `layers.py` berisi class untuk setiap layer berupa `DenseLayer`, `ConvolutionLayer`, `Pooling`, dan `FlattenLayer`, dan terakhir adalah `main.py` untuk menjalankan kode program.

Berikut adalah penjelasan dari tiap kelas, fungsi, dan variabel yang terdapat pada kode program.

1. Fungsi Read Dataset (`read_dataset`)

Read dataset adalah fungsi yang digunakan untuk membaca tiap file path dari image yang ada di dalam folder. Parameter yang dibutuhkan adalah dataset path, yaitu path folder yang berisi folder dengan nama yang merepresentasikan kelasnya dan berisi foto-foto yang termasuk kelas tersebut. Fungsi ini mengembalikan list class folder path (*folder_path*), label class (*class_label*), dan *class dictionary* (*class_dictionary*).

2. Fungsi List Image to Matrix (`list_img_to_matrix`)

Fungsi ini digunakan untuk mengubah tiap image dalam sebuah folder dataset menjadi matriks RGB dengan ukuran yang diinginkan, ukuran defaultnya 400x400. Ukuran 400x400 dipilih dengan alasan karena 400 merupakan nilai rata-rata dari ukuran seluruh data. Untuk setiap foto dalam setiap folder akan dilakukan pembacaan dan pengkonversian menjadi matriks RGB. Selanjutnya matriks tersebut disesuaikan ukurannya dengan size yang diinginkan (default 400x400) menggunakan *third-party* OpenCV. Kemudian setiap matriks image akan dimasukkan dalam sebuah matriks (*list_of_image_matrix*) dan dibentuk numpy array serta dilakukan transpose agar siap digunakan dan dimanipulasikan pada Forward Propagation Convolutional Neural Network.

3. Convolution Layer

Inisiasi kelas *ConvolutionLayer* membutuhkan input berupa ukuran input *nb_channel*, ukuran filter *nb_filter*, jumlah filter *filter_size*, ukuran *padding*, dan ukuran *stride*. Setelah itu, *weight* untuk masing-masing matriks filter diinisiasi secara random dan *bias* diinisiasi dengan nilai nol.

Kelas ini memiliki beberapa fungsi sebagai berikut.

- `Add_zero_padding`

Input : Matriks

Proses : Menambahkan padding bernilai 0 setebal nilai inisialisasi padding pada sekeliling matriks input

Output : Matriks input yang telah ditambahkan padding nol

- `Forward`

Input : Matriks

Proses : Menginisialisasi matriks Feature Map dengan ukuran $V \times V$, dimana nilai V dapat diperoleh dengan rumus $V = (W - F + 2P) / S + 1$. Setelah itu, nilai pada matriks Feature Map dapat diperoleh dengan melakukan perkalian antara Receptive Field yang dihasilkan dari matriks input dengan matriks weight pada inisialisasi, kemudian dijumlahkan dengan bias. Terakhir, Feature Map akan diberi fungsi aktivasi ReLU.

Output : Feature map yang telah dimasukkan detector ReLU

- Detector

Input : Matriks

Proses : Memberikan fungsi aktivasi ReLU kepada matriks input

Output : Matriks yang telah diaktivasi dengan ReLU

4. Pooling Layer

Inisiasi Pooling Layer dapat ditentukan `filter_size`, `stride_size`, dan `mode` (Max atau Average).

Pooling Layer memiliki atribut:

- `filter_size` = ukuran filter
- `stride_size` = besar pergeseran
- `mode` = jenis pooling yang akan dilakukan (Max atau Average)

Pooling Layer memiliki fungsi kelas:

- Forward

Input : Matriks,

Proses : Pengambilan cuplikan dari matriks yang akan dilakukan pooling sesuai dengan mode yang diinginkan

Output : Matriks eature map hasil pooling berukuran V yang didapatkan

$V = (W - F) / S + 1$. W merupakan dimensi input, F merupakan ukuran filter, dan S merupakan besaran pergeseran

- Average

Input : Matriks,

Proses : Perhitungan output berupa nilai average dari semua nilai matriks

Output : Float

- Max

Input : Matriks,

Proses : Perhitungan output berupa nilai maksimum dari semua nilai matriks

Output : Float

5. Dense Layer

Pada inisiasi Dense Layer dapat ditentukan jumlah unit yang dan fungsi aktivasi. Terdapat 2 fungsi aktivasi yang dapat digunakan yaitu "Relu" dan "Sigmoid".

Dense Layer memiliki atribut:

- `n_unit` = jumlah unit yang dimiliki.
- `activation` = fungsi aktivasi yang digunakan

- bias = bias dalam perhitungan (inisiasi random)
- weight = bobot perceptron (inisiasi random)

Dense Layer memiliki fungsi kelas:

- Forward
Input : Array,
Proses : Multiplikasi matriks input dengan array ditambah bias
Output : Array berupa hasil multiplikasi dimasukkan pada fungsi aktivasi

6. Flatten Layer

Merupakan layer dengan satu fungsi dengan tujuan mengubah array 2D menjadi array berdimensi 1. Digunakan ketika mengubah array hasil output layer convolusi atau pool sebelum dimasukkan ke dense layer.

Dense Layer memiliki fungsi kelas:

- Forward
Input : Matriks dimensi 2,
Output : Array dimensi 1

BAB II

Hasil Prediksi

Berikut ini adalah hasil prediksi dari program yang dibangun dengan data yang digunakan antara lain,

1. Cat Image (cats.X.jpg)

```
E:\Coding\MLL\IF4074_Tugas1CNN>python src/main.py  
Picture: ./test\cats\cat.0.jpg  
Predict Result : cats
```

Dapat dilihat hasil prediksi yang didapatkan dari input image sesuai dengan hasil yang diharapkan yaitu kelas *Cat*

2. Dog Image (dog.X.jpg)

```
E:\Coding\MLL\IF4074_Tugas1CNN>python src/main.py  
Picture: ./test\dogs\dog.0.jpg  
Predict Result : dogs
```

Dapat dilihat hasil prediksi yang didapatkan dari input image sesuai dengan hasil yang diharapkan yaitu kelas *Dog*

Berdasarkan hasil prediksi yang didapat oleh program yang dibangun dapat disimpulkan bahwa program dapat mengklasifikasikan sebuah objek foto apakah objek tersebut merupakan kelas *Cats* atau kelas *Dogs*. Namun, dalam beberapa percobaan program tidak dapat memberikan hasil prediksi seperti yang diharapkan. Hal ini diakibatkan karena nilai bobot (weight) merupakan nilai acak sehingga dapat menyebabkan tidak akuratnya prediksi.

BAB III

Pembagian Tugas

Berikut adalah pembagian tugas dalam kelompok.

NIM	Tugas
13518021	Pooling, dense layer
13519171	Convolution, detector
13519177	Image processing