

Laporan Tugas Besar 1
IF4074 Pembelajaran Mesin Lanjut
Convolutional Neural Network
Milestone B: Backpropagation



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

- Backward

Input : Matriks error sebelumnya

Proses : Menghitung dw , db , dan dx berdasarkan $prev_errors$. db dihitung dengan menjumlahkan matriks pada setiap filter. Dw dihitung dengan menjumlahkan semua $prev_error * inputs$, sedangkan dx dengan menjumlahkan semua $prev_error * weight$. Nilai yang dikembalikan adalah dx yang diberi fungsi aktivasi ReLU.

Output : gradien

- Update weight

Input : learning rate

Proses : weight dikurangi dengan learning rate * dw dan bias dikurangi dengan learning rate * db . Setelah itu error dw dan db di-reset.

Output : -

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 feature map hasil pooling berukuran V yang didapatkan

$V = (W - F) / S$. 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
- Backward
Input : Matriks error sebelumnya
Proses : Perhitungan output berupa nilai gradien dengan deklarasi nilai awal matriks nol yang memiliki ukuran yang sama dengan matriks input dan di-update apabila penjumlahan indeksnya kurang dari *weight* dan *height* (*average pool layer*) dengan setiap nilai dari matriks error sebelumnya.
Output : gradien
- Update weight
Input : -
Proses : Pada backward *pooling operations*, tidak perlu dihitung bobot dan turunan bias nya karena tidak ada parameter di *pooling operations* sehingga fungsi update bobot dapat di-*skip* atau dilewatkan.
Output : -

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

Input : -

Proses : terjadi perhitungan perubahan weight dan bias dengan rumus negatif gradien

Output : -

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

- Backward

Input : array dimensi 1

Output : Matriks dimensi 2,

- Update weight

Input : -

Proses : Pada backward *flatten*, tidak perlu dihitung bobot dan turunan bias nya karena tidak ada parameter di *flatten* sehingga fungsi update bobot dapat di-skip atau dilewatkan.

Output : -

BAB II

Hasil Prediksi

Berikut adalah hasil prediksi dengan backpropagation menggunakan dua metode yaitu metode biasa dan metode 10-fold cross validation.

1. Metode Biasa

```
E:\Coding\MLL\IF4074_Tugas1CNN>python ./src/main.py
Epoch: 1
E:\Coding\MLL\IF4074_Tugas1CNN\src\common.py:7: RuntimeWarning: overflow encountered in exp
  return 1/(1+np.exp(-input))
Loss: 0.2
Accuracy: 0.6
Epoch: 2
Loss: 0.12882731140470488
Accuracy: 0.4
Epoch: 3
Loss: 0.12595689878469862
Accuracy: 0.4

Predicted: [1. 1. 1. 1.]

Accuracy: 0.75

Confusion matrix:
[[0 1]
 [0 3]]
```

Berdasarkan hasil prediksi dengan metode biasa seperti pada gambar di atas, diperoleh accuracy sebesar 75% dan data prediksi sudah benar untuk semuanya dengan nilai 1 yang menunjukkan hasil prediksi berupa dog.

2. Metode 10-fold cross validation

```
Epoch: 1
Loss: 0.11794603436040763
Accuracy: 1.0
Epoch: 2
Loss: 0.11794603436040763
Accuracy: 1.0
Epoch: 3
Loss: 0.11794603436040763
Accuracy: 1.0

Accuracy: 1.0
Confusion matrix:
[[4]]

Best Accuracy: 1.0
```

Berdasarkan hasil prediksi dengan metode 10-fold cross validation seperti pada gambar di atas yang menunjukkan fold terakhir dan nilai best accuracy, diperoleh best

accuracy diantara semua fold sebesar 100% dan hal ini menunjukkan bahwa data prediksi yang dihasilkan sudah benar.

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*. Penggunaan metode 10-fold cross validation sangat mempengaruhi nilai accuracy karena data lebih banyak di-fit dan di-train. Perbandingan nilai accuracy yang lebih baik pada 10-fold cross validation dengan metode biasa dapat dilihat pada hasil di atas, dimana dengan metode biasa hanya memiliki nilai accuracy sebesar 75%, sedangkan dengan metode 10-fold cross validation dapat dihasilkan nilai accuracy sebesar 100%.

BAB III

Pembagian Tugas

Berikut adalah pembagian tugas dalam kelompok.

NIM	Tugas
13518021	Dense layer, flatten, save, load
13519171	Convolution, detector, fit function
13519177	Pooling, 10-fold cross validation