

Tugas Besar 1
IF4074 Advanced Machine Learning
CNN - Forward Propagation



oleh :

Syihabuddin Yahya Muhammad	/ 13519149
Imam Nurul Hukmi	/ 13519150
Muhammad Furqon	/ 13519184

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2022

1. Penjelasan Kode

1.1. Kelas CNN

Kelas CNN merupakan kelas utama dari implementasi arsitektur CNN. untuk menggunakan model CNN, cukup kelas ini saja yang digunakan. Kelas ini baru perisikan 2 fungsi yaitu *initialization* dan *predict*.

1.2. Kelas Convolution Layer

Kelas Convolution Layer berfungsi untuk melakukan konvolusi terhadap *matriks input*. Pada tahapan ini didefinisikan sejumlah input yang akan dioperasikan dengan matriks kernel *filter*. *Weight* dan bias diinisialisasi secara acak saat dilakukan inisialisasi. Output yang dihasilkan sebanyak jumlah *filter*.

1.3. Kelas Detector Layer

Kelas Detector Layer memiliki fungsi yang sederhana, yaitu menerapkan fungsi RELU pada setiap elemen pada matriks input. Sehingga, kelas ini tidak membutuhkan argumen apapun ketika diinisialisasi.

1.4. Kelas Pooling Layer

Kelas Pooling Layer berfungsi untuk memperkecil ukuran matriks input sehingga layer selanjutnya dapat memproses matriks dengan lebih efisien. Kelas Pooling Layer bekerja dengan cara memetakan elemen-elemen yang saling bertetangga dan menerapkan suatu fungsi, seperti mencari maksimal atau menghitung rata-rata, kepada kumpulan elemen tersebut sehingga menghasilkan suatu nilai baru. Hal ini dilakukan agar matriks dapat diperkecil dengan tetap mempertahankan sebagian fitur penting yang ada pada matriks input.

1.5. Kelas Dense & FFNN

Kelas FFNN merupakan sebuah implementasi arsitektur *Feed Forward Neural Network*. Kelas ini diambil dari hasil tubes mata kuliah *Machine Learning* tanpa modifikasi sama sekali. Sedangkan kelas Dense merupakan wrap class dari FFNN sebagai interface untuk digunakan oleh CNN.

1.6. Fungsi Utility

Pada Program ini, terdapat 2 fungsi utilitas yang diletakkan pada *util.py*. Fungsi utilitas pertama adalah fungsi *preprocessing*. Fungsi ini digunakan untuk membaca gambar, dan mengubahnya menjadi array of matriks dengan ukuran tertentu(gambar di *resize*). Fungsional ini diimplementasikan dengan menggunakan library *cv2*. Fungsi kedua adalah fungsi *flatten* yang berfungsi

untuk mengubah array 3 dimensi menjadi array 1 dimensi. Fungsi ini digunakan untuk proses passing data menuju kelas Dense/FFNN.

1.7. Lain-Lain

Pada kode juga terdapat beberapa file yang tidak dituliskan disini. File tersebut merupakan file yang digunakan oleh FFNN. karena keterbatasan desain(tidak mengubah FFNN dan komponennya) dan bahasa(python) seluruh kode harus disimpan di file yang sama.

2. Hasil

Kode program dapat diakses di <https://github.com/LastAeon/tubes-ML-lanjutan-CNN>

2.1. Input



```
# # testing CNN
image_src = "test\cats\cat.0.jpg"
cnn_test = CNN("CNN_architecture.txt")
# cnn_test.forwardPropagation(image_src)
print(cnn_test.forwardPropagation(image_src))
```

Digunakan gambar input yang diambil dari dataset. Gambar di-load sesuai dengan path pada program. Pada gambar yang telah di-load didapatkan matriks

input yang akan dilakukan preproses untuk mendapatkan kumpulan matriks input untuk CNN.

Untuk pengaturan dari arsitektur CNN digunakan *file* yang berisi parameter arsitektur. *File* yang di-load menentukan parameter untuk inisialisasi dari CNN yang digunakan.

2.2. Architecture

Dikarenakan implementasi FFNN untuk dense layer menggunakan proyek sebelumnya yang sudah ada, maka file arsitektur dibagi menjadi 2 yaitu arsitektur untuk CNN dan arsitektur untuk Dense/FFNN. Berikut arsitektur yang digunakan untuk melakukan testing.

```
≡ CNN_architecture.txt
1  3 7
2  4
3  Convolution
4  [] 1 1 3 2
5  Detector
6  []
7  Pooling
8  [] 2 2 1 max
9  Dense
10 Dense_Architecture.txt

≡ Dense_Architecture.txt
1  2
2  RELU 2
3  1 1 1 1 1 1 1 1 1 1
4  1 1 1 1 1 1 1 1 1 1
5  linear 1
6  1 1 1
```

Pada arsitektur CNN, baris pertama merupakan ukuran resize gambar. Untuk arsitektur di atas, gambar yang diterima akan diubah menjadi array of matrix RGB yang kemudian di resize menjadi ukuran 3*7. Kemudian baris berikutnya adalah jumlah layer. Untuk dense layer dianggap 1 layer saja karena penjelasan detailnya dilakukan di file berbeda. Baris berikutnya merupakan 2 baris berupa jenis layer dan parameternya yang diulang sebanyak jumlah layer kali. Untuk konfigurasi arsitektur dense layer dapat dilihat [disini](#). Ukuran dari hasil CNN sebelum dense layer harus sesuai dengan jumlah output dari dense layer.

2.3. Output

```
D:\Users\Asus\Documents\GitHub\tubes-ML-lanjutan-CNN>C:/Users/syiha/AppData/Local/Programs/Python/Python39/python.exe d:/Users/Asus/Documents/GitHub/tubes-ML-lanjutan-CNN/test.py
[32787.0]
```

```
D:\Users\Asus\Documents\GitHub\tubes-ML-lanjutan-CNN>C:/Users/syiha/AppData/Local/Programs/Python/Python39/python.exe d:/Users/Asus/Documents/GitHub/tubes-ML-lanjutan-CNN/test.py
[17563.0]
```

```
D:\Users\Asus\Documents\GitHub\tubes-ML-lanjutan-CNN>C:/Users/syiha/AppData/Local/Programs/Python/Python39/python.exe d:/Users/Asus/Documents/GitHub/tubes-ML-lanjutan-CNN/test.py
[11439.0]
```

```
D:\Users\Asus\Documents\GitHub\tubes-ML-lanjutan-CNN>C:/Users/syiha/AppData/Local/Programs/Python/Python39/python.exe d:/Users/Asus/Documents/GitHub/tubes-ML-lanjutan-CNN/test.py
[30029.0]
```

Berdasarkan hasil pemrosesan gambar dengan CNN, didapatkan hasil yang selalu berubah-ubah. Hal ini terjadi karena bobot dan bias pada Convolution Layer dan Dense Layer selalu diacak pada awal inisialisasi kelas, sehingga sudah sepatutnya hasil prediksi selalu berubah-ubah dalam setiap eksekusi program.

Selain itu, berbeda dari ekspektasi sebuah CNN klasifikasi, keluaran CNN tidak bernilai 0 atau 1 karena fungsi aktivasi yang digunakan pada pengetesan adalah "linear" yang tidak mengubah nilai input sama sekali.

3. Pembagian Tugas

NIM	Nama	Tugas yang Dikerjakan
13519149	Syihabuddin Yahya Muhammad	Preproses gambar dan Dense layer
13519150	Imam Nurul Hukmi	Detector dan Pooling Layer
13519184	Muhammad Furqon	Convolution Layer dan Kelas CNN