

LAPORAN TUGAS BESAR
CONVOLUTIONAL NEURAL NETWORK
MILESTONE B - BACKPROPAGATION
IF4074 Pembelajaran Mesin Lanjut



Disusun Oleh:
Ronggur Mahendra Widya Putra / 13519008
Raden Haryosatyo Wisjnunandono / 13520070
Averrous Saloom / 13520100

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung 2023

Bab I

Penjelasan Kode Program

1. Kelas CNN

Merupakan inisialisasi dari kelas CNN yang merupakan kelas yang berperan sebagai pemanggil kelas-kelas layer dan mengoperasikan forward propagation. Kelas ini memiliki atribut layers, is_compiled, dan input_size & feeding_size yang bisa diinisialisasikan.

Nama Fungsi	Penjelasan
add	Menambahkan layer baru pada model yang merupakan objek dari kelas CNN
predict	Melakukan prediksi untuk input yang diberikan, memanggil method feed forward dari tiap-tiap layer yang ada pada model
compile	Mengcompile model dengan memanggil method compile yang ada pada tiap layer yang ada pada model
load_model	Membaca file json dan me load semua layer dan parameter-parameternya kedalam model
save_model	Menyimpan semua layer dan parameter - parameternya ke suatu file json
fit	Melakukan pembelajaran yang meliputi fase feed forward dan fase backpropagation untuk menyesuaikan weights dan biases

2. Kelas Convolution Layer

Merupakan representasi dari layer konvolusi yang ada pada CNN. Kelas ini memiliki atribut filter_num, filter_size, padding, stride, input_shape, dan filter yang dapat diinisialisasikan. Selain itu, juga terdapat atribut output, next_layer, input_matrix, dan dWn yang akan diupdate seiring berjalannya pembelajaran.

Nama Fungsi	Penjelasan
feedForward	Fungsi feedForward ini menerima input berupa Image yang merupakan

	matriks input/gambar RGB. Fungsi ini berfungsi untuk mengkalkulasi konvolusi matriks dengan mengambil sejumlah local area pada input layer sesuai dengan ukuran filter_size yang kemudian dioperasi sum dengan matriks filter/kernel yang di-generate secara random sebagai weight dan ditambahkan bias untuk mendapatkan feature map.
backprop	Melakukan valid convolution antara turunan error terhadap data konvolusi untuk mendapatkan turunan error terhadap weight yang merupakan gradien. Setelah mendapatkan gradien tersebut, akan dilakukan update terhadap weight. Selain itu juga menghitung turunan error terhadap bias yang merupakan jumlah dari elemen yang ada pada masing-masing filter. Update pada bias juga dilakukan.

3. Kelas Detector Layer

Merupakan representasi dari layer detektor yang ada pada CNN untuk menjalankan detector stage yang menambahkan nonlinearitas pada NN menggunakan fungsi aktivasi relu dan sigmoid

Nama Fungsi	Penjelasan
feedForward	Fungsi feedForward ini menerima input berupa Image yang merupakan matriks input/gambar RGB. Fungsi ini berfungsi untuk melakukan kalkulasi antara matriks hasil konvolusi dengan fungsi aktivasi terhadap setiap channel
reLu	Implementasi dari fungsi aktivasi ReLu
sigmoid	Implementasi dari fungsi aktivasi sigmoid
softmax	Implementasi dari fungsi aktivasi softmax
backprop	Karena pada detector layer tidak terdapat perhitungan yang melibatkan weights dan biases, hanya dilakukan update pada dWn (turunan Error terhadap Weight) untuk disamakan dengan nilai dWn dari layer setelahnya. Hal tersebut dilakukan karena pada layer sebelumnya akan dibutuhkan data tersebut.
derived_softmax	Implementasi turunan dari fungsi aktivasi softmax
relu_derivative	Implementasi turunan dari fungsi aktivasi relu
sigmoid_dervative	Implementasi turunan dari fungsi aktivasi sigmoid

4. Kelas Pooling Layer

Merupakan representasi dari layer pooling yang ada pada CNN untuk menjalankan pooling stage yang berguna untuk mengatasi oversampling. Kelas ini memiliki atribut `kernel_size`, `stride`, dan `mode` yang dapat diinisialisasikan. Selain itu, juga terdapat atribut `dOutput`, `prev_output`, `max_pool_index`, dan `dWn` yang akan terus diupdate pada saat pembelajaran berlangsung.

Nama Fungsi	Penjelasan
feedForward	Fungsi <code>feedForward</code> ini menerima input berupa Image yang merupakan matriks input/gambar RGB. Fungsi ini berfungsi untuk melakukan downsampling terhadap ukuran matriks input dengan cara mengkalkulasi matriks input dengan cara mengambil area dengan ukuran filter yang digunakan, kemudian akan dilakukan aktivasi fungsi menggunakan max pooling, atau average pooling.
pooling	Fungsi <code>pooling</code> ini menerima input berupa matriks yang berukuran kernel sebanyak besarnya <code>stride</code> . Pada fungsi ini akan dilakukan pooling sesuai mode poolingnya. Apabila dilakukan maximum pooling maka akan di- <i>return</i> nilai maksimum dari matriks input. Apabila dilakukan average pooling maka akan di- <i>return</i> nilai rata-rata dari matriks input.
backprop	Menghitung gradien (turunan Error terhadap weights) dengan memasukkan nilai dari <code>dWn</code> pada layer selanjutnya (dalam bentuk flat) kedalam posisi data maximum yang dipooling

5. Kelas Flatten Layer

Merupakan representasi dari layer flatten yang ada pada CNN untuk menjalankan flattening pada input

Nama Fungsi	Penjelasan
feedForward	Fungsi <code>feedForward</code> ini menerima input berupa Image yang merupakan matriks input/gambar RGB. Fungsi ini berfungsi untuk flattening pada input yang merupakan matriks dua dimensi menjadi satu saja.
backprop	Karena pada flatten layer tidak terdapat perhitungan yang melibatkan weights dan biases, hanya dilakukan update pada <code>dWn</code> (turunan Error terhadap Weight) untuk disamakan dengan nilai <code>dWn</code> dari layer setelahnya.

	Hal tersebut dilakukan karena pada layer sebelumnya akan dibutuhkan data tersebut.
--	------------------------------------------------------------------------------------

6. Kelas Dense Layer

Merupakan representasi dari dense layer, dimana input matriks akan dikalikan dengan sebuah weight yang di-generate secara random seukuran dengan matriks input dan ditambah dengan bias, kemudian akan dikalkulasi dengan aktivasi fungsi relu, softmax, atau sigmoid.

Nama Fungsi	Penjelasan
feedForward	Memanggil fungsi ff dan mengubahnya menjadi bentuk numpy array
ff	Fungsi ff ini berfungsi untuk melakukan operasi dot seluruh input dan bias yang terhubung dengan dense layer dengan matriks weight. Untuk mendapatkan hasil prediksi akhir hasil dari dot product akan dikalikan dengan fungsi aktivasi yang bisa berupa relu atau sigmoid
dO_dW	Menghitung turunan net output (sebelum dimasukkan dalam fungsi aktivasi) terhadap weight
dE_dO	Menghitung turunan antara Error dengan net output
dReLU	Menghitung turunan antara output dengan relu activation function
backprop	Mengimplementasikan aturan rantai untuk mencari turunan error dengan weight dengan mengalikan hasil dari fungsi dE_dO dengan dO_dW. Melakukan update weight dan bias

7. Fungsi Utilitas

Merupakan sekumpulan fungsi yang digunakan dalam preprocessing gambar dan mengubah dataset yang berupa image menjadi matriks yang dapat digunakan dalam model CNN

Nama Fungsi	Penjelasan
createPath	Membuat path ke dataset yang ingin digunakan
readDatasets	Membaca file yang ada dalam sebuah folder dan mengembalikan array of files
createMatrix	Mengubah gambar menjadi matriks yang dapat digunakan dalam

	pembelajaran CNN dengan pertama-tama menyesuaikan ukuran dengan menggunakan padding lalu memisahkannya menjadi 3 channel (RGB)
loadData	Mengubah sejumlah file image menjadi matriks dengan memanggil fungsi createMatrix menggunakan for loop
separateChannel	Memisahkan gambar RGB menjadi masing-masing channelnya sendiri (R, G, B)
squaredPadding	Melakukan penyesuaian ukuran matriks dari input gambar dengan menggunakan padding

8. Eksperimen

Merupakan aplikasi dari model CNN yang telah kami buat

Nama Fungsi	Penjelasan
load_images_from_folder	Membaca kumpulan gambar dari folder serta labelnya dari nama foldernya
n_fold_cross_validation	Menerima data serta label untuk dibagi menjadi n_folds fold untuk didapatkan model dengan performa terbaik
evaluate	Membandingkan hasil prediksi dan ground truth dan menghitung akurasi, precision, recall, dan f1

Bab II

Hasil Pengujian

```
Model 0
(64, 64, 3)
training 0

Epoch 1
Accuracy: 0.48

Epoch 2
Accuracy: 0.48
Model 1
(64, 64, 3)
training 1
c:\Users\rongg\kuliah\Semester 9\MLL\TugasBesar_IF4074PembelajaranMesinLanjut\Src\utils.py:80: RuntimeWarning: invalid value encountered in double_scalars
  result[i][j] = arr1[i] * arr2[j]

Epoch 1
Accuracy: 0.6

Epoch 2
Accuracy: 0.6
Model 2
(64, 64, 3)
training 2

Epoch 1
Accuracy: 0.6
```

Karena keterbatasan waktu pengujian dilakukan hanya untuk beberapa model pertama pada n cross_validation untuk pengujian tersebut didapatkan akurasi terbaik 0.6

Bab III

Pembagian Tugas

NIM	Tugas
13519008	K-fold cross validation, preprocessing data for inference, save_mode, load_model, evaluasi model
13520070	Menghitung gradien, update weight, update bias
13520100	Menghitung gradien, update weight, update bias