

**LAPORAN TUGAS BESAR
CONVOLUTIONAL NEURAL NETWORK
MILESTONE A - FORWARD PROPAGATION**

IF4074 Pembelajaran Mesin Lanjutan



Oleh :

Syarifah Aisha Geubrina Yasmin - 13519089

Harith Fakhiri Setiawan - 13519161

Ignatius David - 13518014

**PROGRAM STUDI TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2022**

BAB I

Penjelasan

1. Kelas

Nama Kelas	Penjelasan
ConvolutionalStage	Kelas ini merupakan kelas yang merepresentasikan stage konvolusi pada convolution layer
DetectionStage	Kelas ini merupakan kelas yang merepresentasikan stage detection dan berfungsi untuk aktivasi fungsi RELU pada matriks
PoolingStage	Kelas ini merupakan kelas yang merepresentasikan stage pooling dan berfungsi untuk melakukan downsampling pada matriks input
ConvolutionLayer	Kelas ini berfungsi sebagai pemanggil kelas-kelas diatas dan berguna untuk simplifikasi penyusunan layer pada main.py
DenseLayer	Kelas ini merepresentasikan dense layer, dimana input matriks akan dikalikan dengan sebuah beban float yang di-generate secara random seukuran dengan matriks input dan ditambah dengan bias, kemudian akan dikalkulasi dengan aktivasi fungsi relu atau sigmoid.
CNN	Kelas ini berfungsi sebagai pemanggil kelas-kelas layer dan mengoperasikan forward propagation secara keseluruhan untuk setiap layer yang diinisiasi
Utils	Kelas ini berfungsi untuk bantuan dalam load dan konversi gambar dari bentuk jpg menjadi matriks input RGB yang bisa dioperasikan

2. Fungsi dan Variabel

Kelas ConvolutionalStage

Nama Fungsi	Penjelasan
__init__	<p>Fungsi ini berfungsi sebagai konstruktor tiap kelas yang sudah didefinisikan dan berisikan parameter yang diperlukan pada kelas tersebut</p> <pre>__init__(self, filter_size, num_filter, num_channel, stride=1, padding=0)</pre>

	<p>Filter size merupakan ukuran matriks filter, num_filter merupakan jumlah matriks filter untuk setiap channel matriks input, stride merupakan pergerakan receptive field pada matriks input, dan padding merupakan ukuran padding yang ingin ditambahkan pada matriks input.</p>
zero_padding	<p>Fungsi ini berfungsi untuk memberikan 0 padding pada matriks gambar sesuai dengan masukan pada constructor.</p> <p>zero_padding(self, image, new_width, new_height)</p> <p>Image merupakan matriks input/gambar RGB, dan new_width dan new_height merupakan ukuran baru matriks input. Fungsi ini mengembalikan matriks input baru yang sudah diberikan padding.</p>
getOutputSize	<p>Fungsi ini berfungsi untuk mendapatkan ukuran spasial dari output yang dihasilkan oleh konvolusi layer dengan rumus sesuai yang tertera pada gambar. Fungsi ini mengembalikan ukuran width dan height dari output matrix.</p> <p style="text-align: center;">The spatial size of output: $V \times V \times K$</p> $V = \frac{W - F + 2P}{S} + 1$ <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="background-color: #92d050; padding: 5px; border: 1px solid black;">input size w*w*d</div> <div style="background-color: #4682b4; padding: 5px; border: 1px solid black;">Number (K) and size of kernel (F*F*d)</div> </div> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="background-color: #4682b4; padding: 5px; border: 1px solid black;">Stride value (S)</div> <div style="background-color: #8b4513; padding: 5px; border: 1px solid black;">Number of padding applied (P)</div> </div> <p>getOutputSize(self, width, height)</p> <p>Masukan yang diterima merupakan ukuran matriks input</p>
forward	<p>Fungsi ini berfungsi untuk melakukan kalkulasi konvolusi matriks dengan cara mengambil area pada input layer sesuai dengan ukuran filter_size (receptive field) yang kemudian dioperasikan sum dengan matriks kernel (di-generate secara random) dan ditambahkan bias untuk mendapatkan feature map.</p> <p>forward(self, inputs)</p> <p>Image merupakan matriks input/gambar RGB</p>

Kelas DetectionStage

Nama Fungsi	Penjelasan Fungsi dan Variable
__init__	Fungsi ini berfungsi sebagai konstruktor tiap kelas yang sudah

	<p>didefinisikan dan berisikan parameter yang diperlukan pada kelas tersebut</p> <p><code>__init__(self, activation_function)</code></p> <p>Activation_function merupakan parameter string nama fungsi aktivasi yang ingin digunakan</p>
calc_activation_func	<p>Fungsi ini berfungsi untuk menjalankan operasi activation function yang diminta, untuk kasus ini terdapat RELU dan sigmoid.</p> <p><code>calc_activation_func(self, inputs)</code></p>
forward	<p>Fungsi ini berfungsi untuk melakukan kalkulasi aktivasi fungsi terhadap setiap channel matriks input sesuai dengan memanggil fungsi calc_activation_func</p> <p><code>forward(self, inputs)</code></p> <p>Inputs merupakan matriks input/gambar RGB dan akan mengembalikan matriks hasil operasi.</p>

Kelas PoolingStage

Nama Fungsi	Penjelasan
<code>__init__</code>	<p>Fungsi ini berfungsi sebagai konstruktor tiap kelas yang sudah didefinisikan dan berisikan parameter yang diperlukan pada kelas tersebut</p> <p><code>__init__(self, filter_size, stride, isMax)</code></p> <p>Filter size merupakan parameter untuk ukuran filter yang digunakan, stride adalah ukuran stride, dan isMax merupakan boolean dimana apabila True, maka digunakan metode maxpooling, apabila False maka dilakukan avgpooling</p>
forward	<p>Fungsi ini berfungsi untuk melakukan kalkulasi terhadap matriks dengan cara mengambil area dengan ukuran filter yang digunakan, kemudian akan dilakukan aktivasi fungsi menggunakan maxpooling, atau averagepooling. Fungsi dari pooling adalah downsampling terhadap ukuran matriks input.</p> <p><code>forward(self, inputs)</code></p> <p>Inputs merupakan matriks input yang berisi matriks output dari proses detection stage.</p>

findMax	<p>Fungsi ini berfungsi untuk mencari nilai maksimum dan dijalankan ketika fungsi yang diaktivasi adalah maxpooling stage yang berguna untuk mencari nilai elemen maksimum yang terdapat pada matrix.</p> <p>findMax(self, inputs)</p> <p>Inputs merupakan matriks yang telah diambil seukuran filter size.</p>
findAvg	<p>Fungsi ini berfungsi untuk mencari nilai rata-rata dan dijalankan ketika fungsi yang diaktivasi adalah avgpooling stage yang berguna untuk mencari nilai rata-rata dari elemen yang terdapat pada matrix.</p> <p>findAvg(self,inputs)</p> <p>Inputs merupakan matriks yang telah diambil seukuran filter size.</p>

Kelas ConvolutionLayer

Nama Fungsi	Penjelasan
__init__	<p>Fungsi ini berfungsi sebagai konstruktor tiap kelas yang sudah didefinisikan dan berisikan parameter yang diperlukan pada kelas tersebut dan menginisialisasi kelas ConvolutionStage, PoolingStage, dan DetectionStage sebagai atribut kelas.</p> <p>__init__(self, filter_size, num_filter, num_channel, isMax, act_func_detection, stride=1, padding=0):</p> <p>Parameter yang digunakan merupakan parameter-parameter yang terdapat pada inisialisasi kelas ConvolutionStage, PoolingStage, dan DetectionStage</p>
forward	<p>Fungsi ini berfungsi untuk memanggil fungsi forward pada kelas ConvolutionStage, PoolingStage, dan DetectionStage secara berurutan</p> <p>forward(self, inputs)</p> <p>Inputs merupakan matriks input/gambar RGB dan akan mengembalikan matriks hasil operasi poolingstage.</p>

Kelas DenseLayer

Nama Fungsi	Penjelasan
__init__	<p>Fungsi ini berfungsi sebagai konstruktor tiap kelas yang sudah didefinisikan dan berisikan parameter yang diperlukan pada kelas</p>

	<p>tersebut.</p> <pre>__init__(self, n_units, activation_function, n_inputs=None)</pre> <p>N_units merupakan jumlah neuron, activation_function merupakan parameter string berisi nama fungsi yang akan digunakan, dan n_inputs merupakan jumlah input yang masuk.</p>
forward	<p>Fungsi ini berfungsi untuk melakukan operasi dot seluruh input yang terhubung dengan dense layer dengan matriks bobot bertipe float dan ditambahkan dengan bias. Untuk mendapatkan hasil prediksi akhir, fungsi aktivasi yang digunakan adalah sigmoid.</p> <pre>forward(self, inputs)</pre> <p>Inputs merupakan matriks input/gambar RGB dan akan result dari satu input</p>
calc_activation_func	<p>Fungsi ini berfungsi untuk menjalankan operasi activation function yang diminta, untuk kasus ini terdapat RELU dan sigmoid.</p> <pre>calc_activation_func(self, inputs)</pre>

Kelas FlattenLayer

Nama Fungsi	Penjelasan
forward	<p>Fungsi ini berfungsi untuk meratakan input matriks menjadi hanya 1 dimensi saja agar dapat dioperasikan oleh dense layer dengan mudah.</p> <pre>forward(self, inputs)</pre> <p>Inputs merupakan matriks input/gambar RGB dan akan mengembalikan satu dimensi matriks</p>

Kelas CNN

Nama Fungsi	Penjelasan
__init__	<p>Fungsi ini berfungsi sebagai konstruktor tiap kelas yang sudah didefinisikan dan berisikan parameter yang diperlukan pada kelas tersebut, yaitu pointer to layer-layer yang didefinisikan untuk satu buah CNN</p> <pre>__init__(self, *layers)</pre>

	Layers merupakan pointer yang menunjuk pada kelas-kelas layer yang sebelumnya sudah didefinisikan
forward	<p>Fungsi ini berfungsi untuk mengoperasikan CNN dari layer pertama sampai layer terakhir</p> <p>forward(self, inputs)</p> <p>Inputs merupakan matriks input/gambar RGB dan akan mengembalikan matriks output hasil operasi sebuah matriks RGB/1 gambar</p>
predict	<p>Fungsi ini berfungsi untuk melakukan training pada suatu dataset yang telah dilabel (antara cats atau dogs) dengan cara menjalankan fungsi forward propagation untuk seluruh dataset untuk memprediksi output dari suatu gambar.</p> <p>predict(self, features):</p> <p>Feature merupakan matriks input gambar.</p>

Kelas Utils

Nama Fungsi	Penjelasan
__init__	<p>Fungsi ini merupakan konstruktor yang berfungsi menyimpan root, path_dogs yang merupakan path untuk dataset dogs, dan path_cats untuk dataset cats.</p> <p>__init__(self)</p>
createPath	<p>Fungsi ini merupakan fungsi yang membuat directory ke dalam path dataset yang diperlukan</p> <p>createPath(self, animal)</p> <p>Animal diisi dengan parameter dataset yang diperlukan, dapat berisi “cats” atau “dogs”.</p>
loadTrainDataSet	<p>Fungsi ini merupakan fungsi yang melakukan load terhadap dataset yang ada, yaitu dataset cats dan dogs. Fungsi ini mengembalikan array yang berisi image dari dataset.</p> <p>loadTrainDataSet(self)</p>
createMatrix	<p>Fungsi ini merupakan fungsi yang digunakan untuk mengkonversi gambar menjadi matriks RGB pixel, kemudian mengkonversi lagi</p>

	<p>menjadi matriks red, matriks green, dan matriks blue.</p> <p>createMatrix(self, animal_arr, idx, animal)</p> <p>Animal_arr adalah variabel parameter yang berisi array dari image cats dan dogs yang telah di load, idx adalah posisi gambar cats/dogs pada animal_arr, animal adalah parameter variable yang berisi “cats” atau “dogs”</p>
loadAllData	<p>Fungsi ini merupakan fungsi yang akan melakukan konversi seluruh dataset gambar menjadi sebuah matriks RGB dan juga mengembalikan label kelas tiap gambar/matriksnya.</p> <p>loadAllData(self, animal_arr)</p> <p>Animal_arr adalah variabel parameter yang berisi array dari image cats dan dogs yang telah di load</p>
createOneColor	<p>Fungsi ini merupakan fungsi yang mengambil satu unsur warna pada matriks RGB, hal ini dilakukan untuk menyesuaikan bentuk array yang diperlukan dalam operasi konvolusi sehingga menjadi (3, width, height)</p> <p>createOneColor(self, color_one,idx)</p> <p>color_one adalah variabel parameter yang berisi array RGB dari gambar yang akan di load, dan idx adalah idx warna dimana 0 adalah red, 1 adalah green, dan 2 adalah blue.</p>
squaredPadding	<p>Fungsi ini merupakan fungsi yang menambahkan padding pada bagian vertikal/horizontal matriks gambar agar height dan width berukuran sama.</p> <p>squaredPadding(self, RGB_Matrix)</p> <p>RGB_matriks merupakan matriks hasil konversi gambar menjadi matriks RGB.</p>

BAB II

Contoh Prediksi

```
dog.0.jpg --> label :  dogs and predicted as : cats
```

```
=====
```

```
dog.24.jpg --> label :  dogs and predicted as : cats
```

```
=====
```

```
dog.25.jpg --> label :  dogs and predicted as : dogs
```

```
=====
```

```
dog.3.jpg --> label :  dogs and predicted as : dogs
```

```
=====
```

```
dog.30.jpg --> label :  dogs and predicted as : dogs
```

```
=====
```

```
dog.34.jpg --> label :  dogs and predicted as : dogs
```

```
=====
```

```
dog.36.jpg --> label :  dogs and predicted as : dogs
```

```
=====
```

```
dog.38.jpg --> label :  dogs and predicted as : cats
```

```
=====
```

```
dog.4.jpg --> label :  dogs and predicted as : dogs
```

```
=====
```

```
dog.40.jpg --> label :  dogs and predicted as : dogs
```

```
=====
```

```
dog.42.jpg --> label :  dogs and predicted as : cats
```

```
=====
```

```
dog.5.jpg --> label :  dogs and predicted as : dogs
```

```
=====
```

```
dog.50.jpg --> label :  dogs and predicted as : dogs
```

```
=====
```

```
dog.51.jpg --> label :  dogs and predicted as : dogs
```

```
=====
```

```
dog.58.jpg --> label :  dogs and predicted as : cats
```

```
=====
```

```
dog.60.jpg --> label :  dogs and predicted as : cats
```

```
=====
```

```
dog.61.jpg --> label :  dogs and predicted as : cats
```

```
=====
```

```
dog.63.jpg --> label :  dogs and predicted as : cats
```

```
=====
```

```
dog.8.jpg --> label :  dogs and predicted as : dogs
```

```
=====
```

```
dog.9.jpg --> label :  dogs and predicted as : dogs
```

```
=====
cat.0.jpg : label =  cats and predicted as : dogs
=====
```

```
cat.15.jpg : label =  cats and predicted as : cats
=====
```

```
cat.17.jpg : label =  cats and predicted as : cats
=====
```

```
cat.19.jpg : label =  cats and predicted as : dogs
=====
```

```
cat.2.jpg : label =  cats and predicted as : dogs
=====
```

```
cat.21.jpg : label =  cats and predicted as : dogs
=====
```

```
cat.23.jpg : label =  cats and predicted as : cats
=====
```

```
cat.26.jpg : label =  cats and predicted as : dogs
=====
```

```
cat.36.jpg : label =  cats and predicted as : cats
=====
```

```
cat.38.jpg : label =  cats and predicted as : dogs
=====
```

```
cat.40.jpg : label =  cats and predicted as : cats
=====
```

```
cat.45.jpg : label =  cats and predicted as : dogs
=====
```

```
cat.48.jpg : label =  cats and predicted as : cats
=====
```

```
cat.49.jpg : label =  cats and predicted as : cats
=====
```

```
cat.50.jpg : label =  cats and predicted as : cats
=====
```

```
cat.58.jpg : label =  cats and predicted as : dogs
=====
```

```
cat.60.jpg : label =  cats and predicted as : cats
=====
```

```
cat.61.jpg : label =  cats and predicted as : dogs
=====
```

```
cat.71.jpg : label =  cats and predicted as : cats
=====
```

```
cat.9.jpg : label =  cats and predicted as : cats
=====
```

Accuration score

```
=====
the final accuracy score is 0.575
```

BAB III

Pembagian Tugas

No.	Nama	NIM	Kontribusi
1.	Syarifah Aisha Geubrina Y.	13519089	Load image, preprocessed matrix, convolution stage, detection stage, pooling stage, convolution layer, dense layer, flatten layer, CNN, main.py, laporan (Mengerjakan semuanya bersamaan melalui sharescreen dan live share)
2.	Harith Fakhiri S.	13519161	Load image, preprocessed matrix, convolution stage, detection stage, pooling stage, convolution layer, dense layer, flatten layer, CNN, main.py, laporan (Mengerjakan semuanya bersamaan melalui sharescreen dan live share)
3.	Ignatius David P.	13518014	Memberi referensi awal, flatten layer

Program ini dibuat dengan metode live share dan share screen, sehingga tidak ada pembagian tugas yang spesifik.