

**LAPORAN TUGAS BESAR 1**  
**IF4174 Pembelajaran Mesin Lanjut**  
**Milestone A: Forward Propagation**



**Disiapkan oleh:**

Christian Tobing Alexandro - 13519109

Christian Gunawan - 13519199

**Teknik Informatika**  
**Sekolah Teknik Elektro dan Informatika**  
**Institut Teknologi Bandung**  
**2022**

## 1. Penjelasan Kode Program

Kelas	Fungsi	Variabel
Activation	Membuat fungsi perhitungan dasar dan dapat digunakan pada kelas lainnya untuk aktivasi	<p>sigmoid : <math>1/(1+\exp(-x))</math></p> <p>sigmoid_deriv : turunan sigmoid</p> <p>relu: nilai maximum antara nilai dan 0</p> <p>relu_deriv : turunan relu</p>
Dense	Representasi FCNN layer yang akan menerima masukan dan jumlah unit untuk output banyaknya data yang masuk	Forward: mengalikan matriks input dan matriks weight dan dijumlahkan dengan matriks bias, lalu diaktivasi sesuai dengan parameter aktivasi yang dimasukkan seperti relu, sigmoid, tanh, dsb.
Conv2D	Layer convolution dan detektor pada CNN. Conv2D dapat ditambahkan pada kelas Sequential untuk konvolusi	<p>Forward: Pengembalian hasil penjumlahan konvolusi untuk setiap feature map yang dihasilkan</p> <p>input_shape: ukuran masukan layer</p> <p>output_shape: ukuran keluaran layer</p> <p>num_filter: banyak filter yang dihasilkan</p> <p>kernel_shape: ukuran kernel</p> <p>pad: besar padding</p> <p>stride: besar stride</p>
Sequential	Menyimpan model CNN atau mendefinisikan struktur model. Sequential dapat menampung layer yang digunakan dalam model. Layer pertama adalah masukan dan layer selanjutnya adalah	<p>Add : Menambah layer dalam model</p> <p>Forward: Menerima matriks dengan banyak data yang</p>

	output dari layer sebelumnya	<p>akan di forward</p> <p>layers : list dari layer</p> <p>Output_shape : ukuran keluaran model</p>
Pooling2D	Layer pooling pada CNN yang digunakan untuk memproses feature map hasil dari konvolusi.	<p>Forward: menerima feature map hasil konvolusi dan melakukan downsampling dengan mengambil value paling representatif dari window</p> <p>input_shape: ukuran masukan layer</p> <p>output_shape: ukuran keluaran layer</p> <p>pool_shape: bentuk window</p> <p>pad: besar padding</p> <p>stride: besar stride</p> <p>pool_mode: jenis pooling (max atau avg)</p>
Flatten	Melakukan flattening atau mengubah bentuk matriks hasil dari konvolusi menjadi bentuk array	<p>Forward: menerima hasil konvolusi dan mengembalikan array hasil flattening pada konvolusi</p> <p>input_shape: ukuran masukan layer</p> <p>output_shape: ukuran keluaran layer</p>

## 2. Contoh Hasil Prediksi

Berikut adalah hasil dari CNN

```
model = Sequential()
model.add(Conv2D(1, (2, 2), pad=1, stride=2, input_shape=(xdata.shape[1], xdata.shape[2], xdata.shape[3]), activation='relu'))
model.add(Pooling2D((2, 2), stride=1))
model.add(Flatten())
model.add(Dense(8, activation='sigmoid'))
model.add(Dense(4, activation='sigmoid'))
model.add(Dense(1, activation='sigmoid'))
```

```

1 pred = model.forward(x_train)
2 print(pred)

```

✓ 8.2s

```

[[0.16970184]
 [0.17359118]
 [0.16970184]
 [0.95895826]
 [0.65460354]
 [0.16970184]
 [0.95895826]
 [0.65460354]
 [0.16970184]
 [0.65460354]
 [0.16970184]
 [0.95895826]
 [0.16970184]
 [0.95895826]
 [0.95895826]
 [0.16970184]
 [0.83881471]

```

	precision	recall	f1-score	support
0	1.00	0.06	0.11	18
1	0.51	1.00	0.68	18
accuracy			0.53	36
macro avg	0.76	0.53	0.39	36
weighted avg	0.76	0.53	0.39	36

```

Predicted Array
['cats' 'dogs' 'cats' 'cats' 'cats' 'cats' 'cats' 'cats' 'cats' 'cats'
 'cats' 'cats' 'cats' 'cats' 'cats' 'cats' 'dogs' 'cats' 'cats' 'dogs'
 'cats' 'cats' 'dogs' 'dogs' 'cats' 'cats' 'cats' 'cats' 'cats' 'cats'
 'cats' 'cats' 'cats' 'cats' 'cats' 'cats']
Actual Array
['dogs' 'dogs' 'cats' 'dogs' 'dogs' 'dogs' 'dogs' 'cats' 'cats' 'dogs'
 'cats' 'cats' 'cats' 'cats' 'cats' 'dogs' 'dogs' 'cats' 'dogs' 'dogs'
 'cats' 'cats' 'cats' 'dogs' 'dogs' 'cats' 'cats' 'cats' 'cats' 'cats'
 'dogs' 'dogs' 'dogs' 'dogs' 'cats' 'dogs']

```

Pada gambar diatas, untuk setiap input gambar, akan dibuat sebuah model Sequential yang terdiri dari layer konvolusi, pooling, flatten, dan dua hidden layer FCNN. Untuk output terdiri dari nilai. Pada layer konvolusional, konfigurasi yang dipakai adalah satu buah filter dengan ukuran 2 x 2, padding 1, stride 2,, dan input\_shape sebesar (100, 100, 3). Untuk pooling, pool\_shape yang dipakai berukuran 2 x 2 dan stride 1. Jenis pooling yang dipakai secara default adalah max.

Hasilnya dapat dilihat juga pada gambar diatas, terdapat 40 buah nilai yang setiap nilainya menentukan sebuah gambar diklasifikasikan pada kelas tertentu dengan 0 adalah cats dan 1 adalah dogs.

### 3. Pembagian Kelompok

Nim	Nama	Tugas
13519109	Christian Alexandro Tobing	Sequential, Dense, Convo2D, Flatten, Laporan, Debugging
13519199	Christian Gunawan	Activation, Pooling2D, Main, Laporan, Debugging