

**Tugas 1**  
**IF4074 Pembelajaran Mesin Lanjutan**

**Milestone A : Forward Propagation pada *Convolutional Neural Network***



Disusun oleh:

**Kelompok 10**

M. Rafli Zamzami	13519067
Aria Bachrul Ulum Berlian	13519115
Safiq Faray	13519145

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**  
**INSTITUT TEKNOLOGI BANDUNG**  
**2023**

## A. Struktur Program

Program yang kami buat adalah implementasi *Convolutional Neural Network* (CNN), yaitu pada proses *forward propagation*-nya saja pada milestone ini. Setiap layer yang kami buat merupakan operasi matriks, dan kami menggunakan library “numpy” untuk memproses hal tersebut. Untuk pembacaan sebuah gambar menjadi matriks (single channel ataupun multi channel), kami menggunakan library “opencv”.

Proses pembuatan model dan operasi lainnya dilakukan di file “main.py”. Untuk implementasi CNN, kami letakkan di folder “neuralnetwork” sebagai modul untuk di-import.

Input pada CNN yang kami buat akan memproses matriks dengan bentuk (height, width, channel) atau (height, width) saja. Artinya, CNN dapat memproses gambar multi channel seperti gambar RGB atau gambar single channel seperti gambar grayscale.

Berikut adalah *class* yang kami buat untuk menyusun CNN.

### a. Convolution Layer

Kami membuat class dengan nama “ConvLayer” yang menerima parameter ukuran input, ukuran padding, jumlah filter, ukuran filter, ukuran stride, dan jenis fungsi aktivasi yang akan digunakan sebagai detector. Pada umumnya, fungsi aktivasi untuk detector adalah fungsi reLu. Berdasarkan parameter yang diberikan, class akan inisiasi bobot secara acak dan menghitung ukuran feature map. Untuk ukuran feature map, kami menggunakan rumus sebagai berikut.

$$V = ((input\_height - filter\_size + 2.padding) / stride) + 1$$
$$feature\ map\ size = V . V . number\_of\_filters$$

Proses forward propagation akan mengekstrak fitur dari input dengan proses konvolusi pada setiap channel gambar. Jika masing-masing channel gambar telah dikonvolusi, maka akan dijumlahkan lalu tiap elemen dari matriks feature map akan dimasukkan ke dalam fungsi detector.

### b. Pooling Layer

Class “Pooling” yang kami buat menerima parameter tipe pooling (“max” atau “avg”), ukuran pool, dan stride. Pooling layer memiliki proses yang mirip dalam hal ekstraksi fitur gambar, tetapi yang menjadi pembedanya adalah tidak ada

bobot (filter), sehingga setelah ekstraksi fitur pada gambar, yang dilakukan selanjutnya adalah mencari nilai maksimal atau rata-rata dari fitur yang telah didapatkan. Jika convolution layer berfungsi untuk mendapatkan fitur tertentu, pooling layer adalah untuk mengurangi jumlah parameter input dengan mengurangi jumlah fitur.

**c. Flatten Layer**

Class “Flatten” yang kami buat menerima input berupa matriks pada proses forward propagationnya yang kemudian akan dijadikan array satu dimensi. Class ini tidak menerima parameter apapun saat instansiasi objek.

**d. Dense Layer**

Class “Dense” yang kami buat menerima parameter jumlah unit, panjang input, dan jenis fungsi aktivasi. Saat instansiasi, objek akan menginisiasi bobot secara acak beserta bias. Saat forward propagation dilakukan, input akan dijadikan array satu dimensi, yang kemudian dilakukan perkalian dot product antara input dan bobot yang ditranspose, kemudian dijumlahkan dengan bias.

**e. Activation**

Untuk fungsi aktivasi, kami membuat Softmax, Sigmoid, dan Relu pada masing-masing class berbeda dengan menggunakan satu interface bernama Activation.

**f. Model**

Model yang kami buat diimplementasikan dengan class bernama “NN”. Class NN menerima parameter bentuk input dan layer-layernya. Objek yang telah diinstansiasi dapat ditambahkan layer baru dengan method add. Model yang dibuat akan melakukan semua proses forward propagate pada masing-masing layer jika dijalankan method forward\_propagate() pada modelnya.

**B. Hasil**

Berikut adalah susunan kode sebagai contoh pembuatan model dan hasil inferensi pada salah satu gambar.

File main.py
--------------

```

import cv2
from neuralnet import NN, ConvLayer
from neuralnet import Pooling
from neuralnet import Flatten
from neuralnet import Dense

image = cv2.imread("251.jpeg")

# Build the model
model = NN(image.shape)
model.add(
    ConvLayer(
        input_shape=image.shape,
        padding=0,
        num_filters=1,
        filter_size=(3, 3),
        stride=2,
        detector_function="relu",
    )
)
model.add(Pooling(mode="max", pool_size=(2, 2), stride=2))
model.add(Flatten())

flat_shape = model.layers[2].feature_map_shape
model.add(Dense(1024, flat_shape, "relu"))

model.add(Dense(64, 1024, "relu"))

flat_shape = model.layers[4].feature_map_shape
model.add(Dense(1, 64, "sigmoid"))

result = model.forward_propagate(image)

if result <= 0.5:
    category = "Bear"
else:
    category = "Panda"

print(f"Hasil prediksi: {category}")

```

Hasil inferensi adalah sebagai berikut.

Input gambar :



```
PS C:\DATA\Smt9\ml1\tubes-cnn\src> & C:/Users/HP/AppData/L  
c/main.py  
Hasil prediksi: Panda
```

Proses forward propagation telah berhasil dilakukan. Namun, hasil inferensi masih salah. Hal ini dikarenakan belum diimplementasi proses training dengan backward propagation.

### C. Pembagian Tugas

Anggota Kelompok	Pembagian Tugas
M, Rafli Zamzami (13519067)	Pembuatan Model, perbaikan Convolution dan Pooling Layer, pembuatan contoh hasil inferensi
Aria Bachrul Ulum Berlian (13519115)	Pembuatan Flatten Layer, pembuatan Dense Layer
Safiq Faray (13519145)	Pembuatan Convolutional Layer, pembuatan Pooling Layer