LAPORAN TUGAS BESAR I
Milestone A : Forward Propagation
IF4074 Pembelajaran Mesin Lanjut

Disusun Oleh

Aditya Putra Santosa / 13517013
Leonardo / 13517048
Vinsen Marselino / 13517054

Institut Teknologi Bandung
Sekolah Teknik Elektro dan Informatika
Teknik Informatika

2020

1. Penjelasan Kode Program

Pada implementasi CNN kali ini, kami membuat beberapa kelas. Sebuah kelas, dapat merepresentasikan sebuah layer, ataupun menampung seluruh layer yang ada. Setiap kelas layer harus mengimplementasikan fungsi updateInputShape dan forward yang kemudian akan dipanggil dari kelas sequential. Berikut adalah kelas yang diimplementasikan dan penjelasannya.

No	Nama Kelas	Deskripsi	Fungsi & Variabel
1	Sequential	Berfungsi untuk menyimpan model CNN / mendefinisikan struktur model. Kelas ini dapat menampung layer-layer yang akan digunakan dalam model. Pada layer pertama, perlu didefinisikan input_shape. Untuk layer-layer selanjutnya, akan disesuaikan dengan output_shape dari layer sebelumnya.	Fungsi: ● Add : Berguna untuk menambahkan layer kedalam model. ● Forward : Menerima matrix (n, ...input_shape), dimana n adalah banyak data yang akan di forward. Variabel: ● layers : list dari layer yang ada pada sebuah model CNN. ● output_shape : ukuran keluaran dari model
2	Conv2D	Berfungsi sebagai layer konvolusi dan detector pada CNN. Kelas ini dapat ditambahkan pada kelas sequential untuk melakukan konvolusi.	Fungsi: ● forward : mengembalikan hasil penjumlahan konvolusi per channel untuk setiap feature map yang dihasilkan Variabel: ● input_shape : ukuran masukan untuk sebuah layer ● output_shape : ukuran keluaran dari sebuah layer ● num_filter : menyatakan banyaknya filter yang digunakan / banyak feature map yang dihasilkan ● kernel_shape : menyatakan ukuran dari kernel (w_kernel, h_kernel, c) ● pad : besar padding ● stride : besar stride ● input_shape : menyatakan ukuran input (w, h, c)

			● activation : jenis fungsi aktivasi yang dipakai
3	Pooling2D	Berfungsi sebagai layer pooling pada CNN. Kelas ini digunakan untuk memproses feature map hasil dari konvolusi. Seperti kelas layer lainnya, kelas ini juga dapat ditambahkan pada model sequential.	Fungsi: ● forward : menerima sejumlah feature map hasil konvolusi dan melakukan downsampling dengan mengambil value paling representatif dari window yang didefinisikan ukuran pool (pool_shape) untuk setiap dimensi Variabel: ● input_shape : ukuran masukan untuk sebuah layer ● output_shape : ukuran keluaran dari sebuah layer ● pool_shape : bentuk window dari matrix pooling ● padding : besar padding ● stride : besar stride ● pool_mode : jenis dari pooling yang digunakan (max, avg)
4	Flatten	Kelas ini digunakan untuk melakukan flattening, atau mengubah bentuk matrix hasil dari konvolusi menjadi bentuk array.	Fungsi: ● forward : Menerima hasil dari konvolusi, dan mengembalikan array hasil flattening dari hasil konvolusi tersebut. Variabel: ● input_shape : ukuran masukan untuk sebuah layer ● output_shape : ukuran keluaran dari sebuah layer
5	Dense	Digunakan sebagai representasi FCNN layer. Kelas ini akan menerima input_shape dan jumlah unit sehingga output yang dihasilkan adalah (n, unit) dengan n adalah banyaknya data yang masuk.	Fungsi: ● forward : mengalikan matrix input (n, input_shape) dengan matrix weight (input_shape, unit) kemudian menjumlahkannya dengan matrix bias. Variabel:

			● input_shape : ukuran masukan untuk sebuah layer ● output_shape : ukuran keluaran dari sebuah layer ● units : banyaknya unit yang akan dibuat ● activation : jenis fungsi aktivasi yang akan dipakai

2. Contoh Hasil Prediksi

Berikut ini adalah hasil dari CNN yang kami implementasikan.

```
In [22]: model = Sequential()

         model.add(Conv2D(1, (2, 2), pad=1, stride=10, input_shape=(data.shape[1], data.shape[2], data.shape[3]), activation='relu'))
         model.add(Pooling2D((2, 2), stride=1))
         model.add(Flatten())
         model.add(Dense(4, activation='relu'))
         model.add(Dense(2, activation='relu'))
         model.add(Dense(1, activation='sigmoid'))

In [23]: pred = model.forward(data)

In [24]: pred

Out[24]: array([[0.59485088],
                [0.91398501],
                [0.59485088],
                [0.59485088],
                [0.87698655],
                [0.90038842],
                [0.59485088],
                [0.59485088],
                [0.59485088],
                [0.59485088],
                [0.59881751],
                [0.59485088],
                [0.59485088],
                [0.59485088],
                [0.87667973],
                [0.86558711],
                [0.59485088],
                [0.59485088],
                [0.59485088],

In [25]: pred.shape

Out[25]: (40, 1)
```

Pada gambar tersebut, dapat dilihat bahwa untuk setiap input gambar, kami membuat sebuah model sequential yang terdiri dari layer konvolusional, pooling, flatten, dan dua hidden layer FCNN. Untuk outputnya, terdiri dari sebuah nilai (sebuah unit). Pada layer konvolusional, konfigurasi yang dipakai adalah satu buah filter dengan ukuran 2 x 2, padding 1, stride 10, 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.

3. Pembagian Tugas

NIM Anggota	Nama Anggota	Tugas
13517013	Aditya Putra Santosa	● Mengimplementasikan kelas Conv2D ● Memodifikasi kelas Dense dan Sequential dari tugas sebelumnya (Mata kuliah Pembelajaran Mesin) ● Debugging
13517048	Leonardo	● Membantu implementasi kelas Conv2D ● Mengimplementasikan kelas Pooling2D ● Menggabungkan seluruh implementasi ● Debugging
13517054	Vinsen Marselino	● Mengimplementasikan kelas Flatten ● Menggabungkan seluruh implementasi ● Membuat read image dari folder ● Debugging ● Membuat Laporan