

**Tugas 2 Convolutional Neural Network
Backward Propagation
IF4074 Pembelajaran Mesin Lanjut**



Kelompok V

Faiz Muhammad Muflich (13517093)
Muhammad Hanif Adzkiya (13517120)
Muhammad Ivan Rahmansyah Maulana (13517143)

**PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2020**

A. Source Code

1. class Model

Deskripsi : Kelas untuk memodelkan arsitektur Convolutional Neural Network

method :

Fungsi	Dekripsi
add(Layer)	Menambahkan layer baru ke model
forward(input)	Melakukan fungsi forward propagation dengan memanggil method forward untuk setiap layer dari layer pertama hingga akhir. Hasil akhir berupa array of float
backward	Parameter : target, lrate, momentum_rate melakukan backward propagation untuk menghitung gradient descent dari masing-masing layer yang ada dalam model
fit	Melakukan training model dengan data latih dengan batch size tertentu, epoch, learning rate, dan momentum rate yang ditentukan. Proses update weight dilakukan di akhir setiap batch.

variable :

Variable	Dekripsi
layers	Menyimpan array of object (layer)

2. class Layer

Deskripsi : Kelas abstract untuk memodelkan layer pada Convolutional Neural Network

method :

Fungsi	Dekripsi
forward(input)	Memproses input pada forward step untuk

	menjadi output yang sesuai dengan jenis layer
backward	Melakukan backward di setiap layer Fungsi backward ini akan diimplementasi di setiap layer cnn, aktivasi, flattening, pooling, dan dense
update_weight	Melakukan update_weight setelah 1 batch. Diimplementasikan di class cnn dan dense.
reset_delta	Melakukan reset_delta setiap memasuki batch baru. Diimplementasikan di cnn dan dense.

3. class ConvolutionLayer

Deskripsi : Kelas untuk memodelkan arsitektur Convolutional Neural Network

method :

Fungsi	Dekripsi
init_filter()	menginisialisasi filter sebanyak n_filter, dan berukuran filter_width*filter_height
forward(input)	melakukan proses convolution, dengan input yaitu berupa matrix (width x height x depth).dan melakukan perhitungan terhadap matrix dengan filter yang sudah dinisiasi sebelumnya sesuai dengan parameter yang ada pada kelas convolutionLayer. hasilnya adalah matrix berukuran output_width x output height x n_filter.
backward	Mengimplementasikan fungsi backward dari interface layer
update_weight	Mengimplementasikan fungsi update_weight dari interface layer
reset_delta	Mengimplementasikan fungsi reset_delta dari interface layer

variable :

Variable	Dekripsi
inputs_size	berfungsi untuk menyimpan nilai dari ukuran matrix yang menjadi input pada layer

padding	nilai yang menjadi patokan untuk melakukan padding pada image
n_filter	jumlah filter yang akan digunakan
filter_size	ukuran kernel dari filter
filter	matrix yang menyimpan nilai dari filter
n_stride	jumlah pergeseran yang dilakukan pada input untuk filter
bias_filter	menyimpan nilai bias pada layer

4. class Activation

Deskripsi : Kelas untuk melakukan proses pelatihan model CNN dengan fungsi aktivasi ReLU. Fungsi ReLU akan mengembalikan nilai maksimal dari 0 dengan nilai setiap cell nya. Pada implementasinya menggunakan `np.maximum(input, 0)`

method :

Fungsi	Dekripsi
forward(input)	Memproses input pada forward step untuk menjadi output yang sesuai dengan fungsi aktivase ReLU.
backward	Mengimplementasikan fungsi backward dari interface layer

5. class Pooling

Deskripsi : Kelas untuk memodelkan Layer dengan jenis Pooling. Hal ini dilakukan karena terdapat 2 jenis pooling yaitu average pooling dan max pooling. Fungsi dari pooling ini adalah untuk mereduksi input secara spasial (mengurangi jumlah parameter) dengan operasi down-sampling.

method :

Fungsi	Dekripsi
forward(input)	Memproses input pada forward step untuk menjadi output yang sesuai dengan layer Pooling.
find_pooling(matrix)	Mencari nilai pooling pada sebuah sub bagian matriks input. Metode pencarian nilai pooling kemudian di kostumisasi sesuai dengan metode pooling yang digunakan. Terdapat 2 jenis metode pooling, Max Pooling dan Average Pooling
backward	Mengimplementasikan fungsi backward dari interface layer

variable :

Variable	Dekripsi
pool_size	Menyimpan ukuran pooling. Penyimpan dilakukan dalam bentuk tuple (x,y) yang merepresentasikan ukuran tinggi x lebar dari pooling yang dilakukan
n_stride	Menyimpan jumlah stride yang digunakan dalam proses pooling

6. class AvgPooling

Deskripsi : Kelas untuk melakukan proses pooling dengan menggunakan pendekatan average pooling.

method :

Fungsi	Dekripsi
find_pooling(matrix)	Mencari nilai pooling pada sebuah sub bagian matriks input. Nilai output yang dikeluarkan berupa rata - rata nilai pada matriks masukan
backward	Mengimplementasikan fungsi backward dari interface layer

7. class MaxPooling

Deskripsi : Kelas untuk melakukan proses pooling dengan menggunakan pendekatan max pooling.

method :

Fungsi	Dekripsi
find_pooling(matrix)	Mencari nilai pooling pada sebuah sub bagian matriks input. Nilai output yang dikeluarkan berupa nilai maksimal pada matriks masukan
backward	Mengimplementasikan fungsi backward dari interface layer

8. class DenseLayer

Deskripsi : Kelas untuk memodelkan fully connected neural network layer.

method :

Fungsi	Dekripsi
forward(input)	Melakukan forward propagation untuk dense layer, masukan berupa array of matriks. Output berupa hasil aktivasi dari penjumlahan perkalian setiap elemen input dengan weight ditambah bias.
backward	Mengimplementasikan fungsi backward dari interface layer
update_weight	Mengimplementasikan fungsi update_weight dari interface layer
reset_delta	Mengimplementasikan fungsi reset_delta dari interface layer

variable :

Variable	Dekripsi
units	Menyimpan jumlah neuron.
weights	Menyimpan weight setiap neuron. Disimpan dalam format matriks 2 Dimensi (jumlah input x units).

bias	Menyimpan bias dalam format array 1 dimensi dengan inisiasi 0 semua.
activation	Menyimpan jenis fungsi aktivasi. Dapat berupa sigmoid atau ReLu
outputs	Menyimpan hasil dari aktivasi setelah dilakukan forward propagation untuk setiap neuron. Format array 1 dimensi sebanyak units.

B. Contoh Hasil Prediksi

eksperimen

```
model = Model()
model.add(ConvolutionLayer(inputs_size=(100,100,3), padding=0, n_filter=2,
filter_size=(3,3), n_stride=1))
model.add(Activation())
model.add(MaxPooling((2,2), 2))
model.add(FlattenLayer())
model.add(DenseLayer(units=8, activation='relu'))
model.add(DenseLayer(units=1, activation='sigmoid'))
model.fit(train_list_images,train_list_labels,2,3,0.1,0.1)
```

output

```
[1.15369068e-48] 0
[1.83985204e-125] 0
[3.03661406e-120] 0
[1.79294281e-97] 0
[6.07921703e-211] 0
[2.62532708e-46] 0
[1.25872624e-89] 0
[6.91717008e-82] 0
[2.09879809e-150] 0
[2.46449656e-191] 0
[5.64361016e-135] 0
[9.91625011e-204] 0
[3.01528142e-95] 0
[1.27144458e-190] 0
[8.31003348e-184] 0
[9.63614954e-133] 0
[5.06720889e-124] 0
[3.03252601e-71] 0
```

```

[1.] 0
[4.82123172e-216] 0
[3.16616601e-96] 1
[2.33201035e-166] 1
[1.90268967e-132] 1
[5.55138884e-166] 1
[7.36194631e-133] 1
[2.86419446e-127] 1
[2.81160113e-177] 1
[5.19547161e-112] 1
[1.41564578e-121] 1
[4.33347611e-171] 1
[2.25814314e-134] 1
[4.35406537e-92] 1
[7.2022958e-91] 1
[4.80323749e-86] 1
[1.45738443e-74] 1
[2.62949179e-45] 1
[6.44043569e-116] 1
[1.76309785e-141] 1
[1.38625074e-124] 1
[1.95715266e-164] 1

```

Accuracy : 0.475

C. Pembagian Tugas

Nama	Pembagian Tugas
Faiz Muhammad Muflich (13517093)	Kelas Main, Kelas Dense Layer, Testing dan Prediksi, Laporan, Mengerjakan coding lewat share vs code secara bersama sama
Muhammad Hanif Adzkiya (13517120)	Kelas Main, Kelas Activation dan Pooling, Testing dan Prediksi, Laporan, Mengerjakan coding lewat share vs code secara bersama sama
Muhammad Ivan Rahmansyah Maulana (13517143)	Kelas Main, Kelas ConvolutionLayer, Testing dan prediksi, Laporan, Mengerjakan coding lewat share vs code secara bersama sama Mengerjakan coding lewat

	share vs code secara bersama sama
--	--------------------------------------