

LAPORAN TUGAS BESAR I
Milestone B : Backward 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.

	Nama Kelas	Deskripsi	Fungsi & Variabel
	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 <code>input_shape</code> . Untuk layer-layer selanjutnya, akan disesuaikan dengan <code>output_shape</code> dari layer sebelumnya.	Fungsi: <ul style="list-style-type: none">• Add : Berguna untuk menambahkan layer kedalam model.• Forward : Menerima matrix (n, ...input_shape), dimana n adalah banyak data yang akan di forward. Variabel: <ul style="list-style-type: none">• layers : list dari layer yang ada pada sebuah model CNN.• output_shape : ukuran keluaran dari model
	Conv 2D	Berfungsi sebagai layer konvolusi dan detector pada CNN. Kelas ini dapat ditambahkan pada	Fungsi: <ul style="list-style-type: none">• forward : mengembalikan hasil penjumlahan konvolusi per channel untuk setiap

		<p>kelas sequential untuk melakukan konvolusi.</p>	<p>feature map yang dihasilkan</p> <p>Variabel:</p> <ul style="list-style-type: none"> • 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
	Pooling2D	<p>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.</p>	<p>Fungsi:</p> <ul style="list-style-type: none"> • 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 <p>Variabel:</p>

			<ul style="list-style-type: none"> • 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)
	Flatte n	Kelas ini digunakan untuk melakukan flattening, atau mengubah bentuk matrix hasil dari konvolusi menjadi bentuk array.	<p>Fungsi:</p> <ul style="list-style-type: none"> • forward : Menerima hasil dari konvolusi, dan mengembalikan array hasil flattening dari hasil konvolusi tersebut. <p>Variabel:</p> <p>input_shape : ukuran masukan untuk sebuah layer</p> <p>output_shape : ukuran keluaran dari sebuah layer</p>
	Dens e	Digunakan sebagai representasi FCNN layer. Kelas ini akan menerima input_shape dan jumlah unit sehingga output yang dihasilkan adalah (n, unit) dengan n adalah	<p>Fungsi:</p> <ul style="list-style-type: none"> • forward : mengalikan matrix input (n, input_shape) dengan matrix weight (input_shape, unit) kemudian menjumlahkannya dengan matrix bias.

		banyaknya data yang masuk.	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
--	--	----------------------------	---

Selain kelas-kelas untuk *layers* dan Sequential model, terdapat juga beberapa file yang berisikan fungsi-fungsi pembantu yang dipakai dalam beberapa kelas yang disebutkan di atas.

1. util.py

No.	Nama Fungsi	Parameter Fungsi	Penjelasan Fungsi
1.	conv2d_batch	<ul style="list-style-type: none"> • <u>mat</u>: matrix data x • <u>kernel</u>: kernel untuk konvolusi • <u>pad</u>: padding untuk matrix data • <u>stride</u>: lompatan data yang dilakukan saat konvolusi 	Fungsi conv2d melakukan konvolusi untuk matrix <u>mat</u> yang telah dipadding sebanyak <u>pad</u> menggunakan <u>kernel</u> dengan pergeseran sebanyak <u>stride</u> . Konvolusi dilakukan untuk setiap batch data masukan.
2.	conv2d_batch_kernel	<ul style="list-style-type: none"> • <u>mat</u>: matrix data x • <u>kernel</u>: kernel untuk konvolusi • <u>pad</u>: padding untuk matrix data 	Melakukan konvolusi batch namun menggunakan kernel batch yang seukuran dengan matrix <u>mat</u> yang telah diberi <i>padding</i> <u>pad</u> .

		<ul style="list-style-type: none"> • <u>stride</u>: lompatan data yang dilakukan saat konvolusi 	
3.	get_pooling_region	<ul style="list-style-type: none"> • <u>x</u>: matrix data x • <u>pool_shape</u>: ukuran dari pool untuk mendapatkan region pool • <u>stride</u>: lompatan data yang dilakukan saat pooling • <u>output_shape</u>: ukuran output yang diharapkan 	Fungsi mengembalikan satu-per-satu <i>region</i> untuk melakukan pooling.
4.	one_channel_pooling	<ul style="list-style-type: none"> • <u>x</u>: matrix data x • <u>pool_shape</u>: ukuran dari pool untuk mendapatkan region pool • <u>stride</u>: lompatan data yang dilakukan saat pooling • <u>padding</u>: padding untuk matrix data • <u>pool_mode</u>: mode dari pooling yang dilakukan (maximum atau average) (default: max) 	Melakukan pooling untuk suatu gambar 1 channel. Pooling dilakukan terhadap data <u>x</u> yang telah dipadding sebanyak <u>padding</u> . Pooling dilakukan dengan ukuran pool <u>pool_shape</u> dengan pergeseran <u>stride</u> dan mode pooling <u>pool_mode</u> .
5.	pooling2d	<ul style="list-style-type: none"> • <u>x</u>: matrix data x • <u>pool_shape</u>: ukuran dari 	Melakukan multi-channel pooling dengan parameter-parameter yang

		<p>pool untuk mendapatkan region pool</p> <ul style="list-style-type: none"> • <u>stride</u>: lompatan data yang dilakukan saat pooling • <u>padding</u>: padding untuk matrix data • <u>pool_mode</u>: mode dari pooling yang dilakukan (maximum atau average) (default: max) 	diberikan.
6.	readImage	<ul style="list-style-type: none"> • <u>path</u>: lokasi direktori dari gambar yang ingin dibaca • <u>image_size</u>: ukuran yang diinginkan dari gambar. (berbentuk tuple of width height). 	Melakukan pembacaan gambar dari direktori <u>path</u> menggunakan OpenCV. Gambar lalu akan di- <i>resize</i> menjadi ukuran <u>image_size</u> sebelum di- <i>return</i> .

2. activation.py

	Nama Fungsi	Parameter Fungsi	Penjelasan Fungsi
	sigmoid	<ul style="list-style-type: none"> • \underline{x}: data yang akan diaktivasi 	Pengimplementasian fungsi sigmoid $f(x) = \frac{1}{(1 + e^{-x})}$
	sigmoid_deriv	<ul style="list-style-type: none"> • \underline{x}: data yang akan diaktivasi 	Pengimplementasian turunan fungsi sigmoid $f'(x) = \sigma(x) - (1 - \sigma(x))$
	relu	<ul style="list-style-type: none"> • \underline{x}: data yang akan diaktivasi 	Pengimplementasian fungsi relu $f(x) = \max(0, 1)$
	relu_deriv	<ul style="list-style-type: none"> • \underline{x}: data yang akan diaktivasi 	Pengimplementasian turunan fungsi relu $f'(x) = \begin{cases} 0 & x \leq 0 \\ 1 & x > 0 \end{cases}$
	leaky_relu	<ul style="list-style-type: none"> • \underline{x}: data yang akan diaktivasi 	Pengimplementasian fungsi leaky relu dengan parameter leak 1/100 $f(x) = \max(\frac{x}{100}, x)$
	leaky_relu_deriv	<ul style="list-style-type: none"> • \underline{x}: data yang akan diaktivasi 	Pengimplementasian turunan fungsi leaky relu $f'(x) = \begin{cases} 0.01 & x \leq 0 \\ 1 & x > 0 \end{cases}$

2. Hasil Eksperimen

Pada bagian ini akan dipaparkan hasil dari beberapa kali eksperimen yang dilakukan, baik menggunakan split train maupun k-fold.

2.1. Split Train 90% 10%

ID	Arsitektur	Parameter	Hasil Eksperimen
1	Conv2D (8 filter, kernel_size = (2,2), pad = 0, stride = 2, relu) Pooling2D (pool_size = (2,2), stride = 2, padding = 0, average) Flatten Dense (8 unit, relu) Dense (4 unit, relu) Dense (1 unit, sigmoid)	Image_size = (30,30) Epoch = 10 Learning rate = 0.04 Momentum = 0.3 Batch-size = 10	<pre> precision recall f1-score support 0 0.67 1.00 0.80 2 1 1.00 0.50 0.67 2 accuracy 0.83 macro avg 0.83 0.75 0.73 4 weighted avg 0.83 0.75 0.73 4 [[2 0] [1 1]] </pre>
2	Conv2D (4 filter, Kernel_size = (3,3), Pad = 0, Stride = 2, relu) Conv2D (4 filter, kernel_size = (2,2), pad = 0, stride = 2, relu) Pooling2D (pool_size = (2,2), stride = 2, padding = 1, average) Flatten Dense (4 unit, relu) Dense (1 unit, sigmoid)	Image_size = (30,30) Epoch = 5 Learning rate = 0.04 Momentum = 0.3 Batch-size = 10	<pre> precision recall f1-score support 0 1.00 0.50 0.67 2 1 0.67 1.00 0.80 2 accuracy 0.83 macro avg 0.83 0.75 0.73 4 weighted avg 0.83 0.75 0.73 4 [[1 1] [0 2]] </pre>

2.2. 10-fold cross validation

ID	Arsitektur	Parameter	Hasil Eksperimen
1	Conv2D (6 filter, kernel_size = (2,2), pad = 0, stride = 2, relu) Pooling2D (pool_size = (2,2), stride = 2, padding = 0, average) Flatten Dense (4 unit, relu) Dense (1 unit, sigmoid)	Image_size = (30,30) Epoch = 10 Learning rate = 0.04 Momentum = 0.3 Batch-size = 10	Test Data Mean Accuracy = 53% Max Accuracy = 75% (fold 3, 5, 7, 10) All Data Mean Accuracy = 56.75% Max Accuracy = 65% (fold 4) Max Accuracy 2 = 62.5% (fold 3, 6) Max Accuracy 3 = 60% (fold 5)

3. Pembagian Tugas

Anggota	Tugas
13517013 / Aditya Putra Santosa	<ul style="list-style-type: none"> Implementasi <i>Backpropagation</i> Convolutional Layer Implementasi Momentum Refaktor <i>Backpropagation</i> Dense Layer Refaktor Sequential Model Eksperimen model
13517048 / Leonardo	<ul style="list-style-type: none"> Implementasi <i>Backpropagation</i> Pooling Layer Refaktor fungsi-fungsi utilitas Eksperimen model Pembuatan Laporan
13517054 / Vinsen Marselino	<ul style="list-style-type: none"> Implementasi <i>Backpropagation</i> Flatten Layer Eksperimen model Implementasi <i>Save</i> dan <i>Load</i> model

	<ul style="list-style-type: none">• Pembuatan Laporan
--	---