

LAPORAN TUGAS BESAR II
Implementasi Forward pada Recurrent Neural Network
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

I. Penjelasan Kode Program

Berikut akan dijelaskan kelas RNN yang telah dibuat dimulai dari atribut-atributnya sebagai berikut:

Nama Kelas	Atribut	Keterangan
RNN	units	<i>Unit RNN layer</i> yang dipakai. Dapat juga dikatakan sebagai ukuran <i>hidden layer</i> ,
	weight_initializer	Metode inisialisasi <i>weight</i> ,
	input_shape	Ukuran dari vektor input,
	return_sequences	Pilihan untuk mengembalikan <i>sequence</i> setiap <i>time series</i> atau tidak,
	activation_name	Nama dari fungsi aktivasi yang digunakan.
	output_shape	Ukuran dari vektor output.
	h	Array penyimpanan output sementara selama berjalannya algoritma.
	activation	Fungsi aktivasi yang digunakan.
	activation_deriv	Turunan dari fungsi aktivasi yang digunakan.
	bias_xh	Bias dari input menuju <i>hidden layer</i> .
	U	Matrix <i>weight</i> dari input menuju <i>hidden layer</i> .
	W	Matrix <i>weight</i> dari <i>hidden layer</i> menuju <i>hidden layer</i> pada <i>timestep</i> berikutnya.

Berikut adalah metode-metode yang diimplementasikan pada kelas RNN:

Nama Fungsi	Parameter	Keterangan
<code>__init__</code>	units weight_initializer input_shape activation return_sequences	Fungsi inisialisasi layer RNN. Inisialisasi dilakukan dengan ukuran <i>hidden layer</i> sebesar <u>unit</u> , metode inisialisasi <i>weight</i> menggunakan <u>weight_initializer</u> , ukuran vektor input sebesar <u>input_shape</u> , menggunakan fungsi aktivasi <u>activation</u> , dan menandakan apakah <i>layer</i> akan

		<p>mengembalikan <i>sequence</i> berdasarkan <u>return_sequences</u></p> <p>Pada fungsi init juga terjadi pemanggilan fungsi updateWBO untuk melakukan inialisasi <i>weight</i> dan penentuan ukuran dari vektor output</p>
initWeight	size	Menginisialisasi weight berdasarkan tipe dari <i>weight_initializer</i> yang digunakan dengan ukuran sebesar <u>size</u> .
updateInputShape	input_shape	Memperbarui ukuran dari vektor input menjadi <u>input_shape</u> .
updateWBO		<p>Melakukan update untuk ukuran dari vektor output, dan juga melakukan inialisasi dan juga update terhadap matrix U dan W.</p> <p>Pada fungsi updateWBO juga terjadi pemanggilan fungsi initWeight untuk melakukan inialisasi matrix U dan W</p>
getSaveData		Membuat <i>dictionary</i> untuk menyimpan identitas layer.
loadData	data	Menginisialisasi weight berdasarkan <u>data</u> berupa <i>dictionary</i> penyimpanan identitas layer.
forward	x_data	Melakukan fungsi inferensi berdasarkan data <u>x_data</u> yang digunakan.

Selain kelas RNN, dibuat juga fungsi-fungsi aktivasi dengan spesifikasi sebagai berikut:

Nama Fungsi	Parameter	Keterangan
tanh	x: data	<p>Mengaplikasikan fungsi tanh ke data.</p> $f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
tanh_deriv	x: data	<p>Mengaplikasikan turunan fungsi tanh ke data.</p> $f'(x) = 1 - \tanh^2(x)$

softmax	x: data	Mengaplikasikan fungsi softmax ke data.
softmax_deriv	x: data	Mengaplikasikan turunan fungsi softmax ke data.
softmax_time_distributed	x: data	Mengaplikasikan fungsi softmax untuk time distributed data.
softmax_time_distributed_deriv	x: data	Mengaplikasikan turunan fungsi softmax untuk time distributed data.

Pada layer Dense yang telah dibuat dari tugas besar sebelumnya, dibuat suatu atribut baru yaitu sebagai berikut:

Atribut	Keterangan
time_distributed	Menandakan apakah Dense dilakukan setiap time series atau tidak. Default = False

II. Hasil Pengujian

No	Inisialisasi Bobot	Hasil Pengujian
1	Nol	<pre> Eksperimen 1 Hidden size: 3 Output size: 4 Sequence Length: 3 Input size: (3, 4) ====Hidden==== Timestep ke 1 [0. 0. 0.] Timestep ke 2 [0. 0. 0.] Timestep ke 3 [0. 0. 0.] ====Output==== Timestep ke 1 [0.25 0.25 0.25 0.25] Timestep ke 2 [0.25 0.25 0.25 0.25] Timestep ke 3 [0.25 0.25 0.25 0.25] </pre>

2	Satu	<pre> Eksperimen 2 Hidden size: 5 Output size: 5 Sequence Length: 5 Input size: (5, 3) Input data: [[1 0 0] [0 1 0] [0 0 1] [0 0 1] [0 0 1]] ====Hidden==== Timestep ke 1 [0.96402758 0.96402758 0.96402758 0.96402758 0.96402758] Timestep ke 2 [0.99999762 0.99999762 0.99999762 0.99999762 0.99999762] Timestep ke 3 [0.99999834 0.99999834 0.99999834 0.99999834 0.99999834] Timestep ke 4 [0.99999834 0.99999834 0.99999834 0.99999834 0.99999834] Timestep ke 5 [0.99999834 0.99999834 0.99999834 0.99999834 0.99999834] ====Output==== Timestep ke 1 [0.2 0.2 0.2 0.2 0.2] Timestep ke 2 [0.2 0.2 0.2 0.2 0.2] Timestep ke 3 [0.2 0.2 0.2 0.2 0.2] Timestep ke 4 [0.2 0.2 0.2 0.2 0.2] Timestep ke 5 [0.2 0.2 0.2 0.2 0.2] </pre>
3	Random	<pre> Eksperimen 3 Hidden size: 4 Output size: 2 Sequence Length: 5 Input size: (5, 4) Input data: [[1. 0. 0. 0.5] [0. 1. 0. 0.3] [0. 0. 1. 0.1] [0. 0. 1. 0.3] [0. 0. 1. 0.4]] ====Hidden==== Timestep ke 1 [0.91491321 0.93251212 0.61325105 0.87634099] Timestep ke 2 [0.99790047 0.99638426 0.99656569 0.99885663] Timestep ke 3 [0.99737184 0.99857703 0.99967926 0.9986858] Timestep ke 4 [0.99804553 0.99878604 0.99975277 0.99900444] Timestep ke 5 [0.99831565 0.99887807 0.99978264 0.99913347] ====Output==== Timestep ke 1 [0.72792131 0.27207869] Timestep ke 2 [0.73832063 0.26167937] Timestep ke 3 [0.73821131 0.26178869] Timestep ke 4 [0.73825969 0.26174031] Timestep ke 5 [0.73827897 0.26172103] </pre>

III. Pembagian Tugas dalam Kelompok

NIM / Nama	Tugas
13517013 / Aditya Putra Santosa	Implementasi time distributed, debugging,

	laporan
13517048 / Leonardo	Implementasi __init__, Implementasi tanh dan turunannya, laporan
13517054 / Vinsen Marselino Andreas	Implementasi Forward, implementasi time distributed, debugging, eksperimen, laporan