

# Recurrent Neural Network

Isjhar Kautsar  
(23519009)

Recurrent Neural Network digunakan untuk network yang jumlah input dan output tidak dapat ditentukan dari awal. RNN akan dicoba diimplementasikan pada tiga studi kasus, yaitu RNN dan LSTM basic, Pasar modal, dan nama dinosaurus generator.

**Keywords:** Machine Learning, Neural Network, Recurrent Neural Network

## 1 Latar Belakang

Pada neural network biasa jumlah input dan output harus terdefinisi dari awal pembuatan network-nya. Hal ini sulit untuk menyelesaikan masalah yang jumlah input dan outputnya tidak dapat didefinisikan dari awal seperti penerjemahan suatu bahasa, panjang input dan output tidak dapat ditentukan dari awal. Oleh karena itu, untuk menangani permasalahan ini digunakanlah recurrent neural network (RNN).

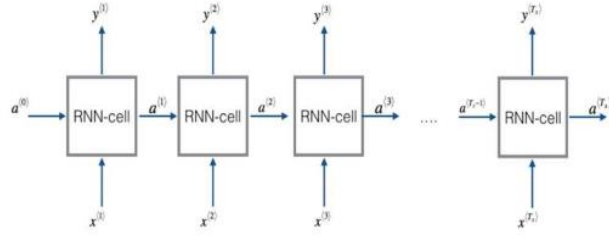
Pada laporan ini memuat hasil implementasi 4 tugas terkait RNN, antara lain, basic RNN dan LSTM, Pasar Modal Tata Global, Pasar Modal Nikkei 225, dan Dinosaur. Tugas basic RNN terkait konsep dasar RNN, Tugas Pasar Modal terkait prediksi harga saham menggunakan dataset Tata dan Nikkei 225, dan Dinosaur terkait membuat nama dinosaur secara otomatis.

## 2 Deskripsi

Bagian ini akan berisi penjelasan hasil pengerjaan keempat tugas tersebut.

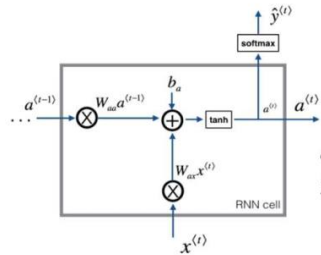
### 2.1 Basic RNN dan LSTM

RNN terdiri atas banyak cell yang saling terhubung (Fig 1). Satu cell terhubung ke satu cell berikutnya. Satu cell merepresentasikan satu time-step. Satu cell terdiri dari dua input yaitu satu input dari data, satu input bobot hidden state sebelumnya, dan dua output yaitu satu output hasil prediksi, dan satu output hidden state. Output hidden state dari cell sebelumnya diteruskan ke cell selanjutnya. Untuk cell pertama nilai hidden state yang menjadi input ditentukan secara acak. Jumlah cell disesuaikan dengan jumlah input .



**Fig. 1.** Struktur RNN

Tugas ini diawali dengan membuat fungsi forward propagation untuk satu cell pada RNN.



**Fig. 2.** Forwrad propagation satu cell RNN

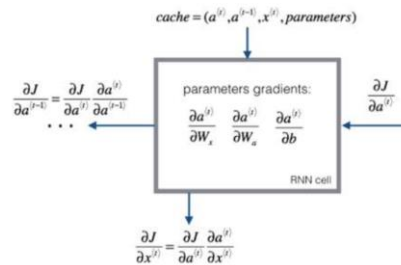
Fungsi perhitungan sebagai berikut:

$$a^{(t)} = \tanh(W_{ax}x^{(t)} + W_{aa}a^{(t-1)} + b_a) \quad (1)$$

$$\hat{y}^{(t)} = \text{softmax}(W_{ya}a^{(t)} + b_y) \quad (2)$$

Selanjutnya dibuat fungsi perhitungan forward propagation untuk seluruh cell dengan menelusuri cell awal sampai cell akhir dimana output hidden state cell sebelumnya dijadikan input hidden state cell berikutnya.

Selanjutnya dibuat fungsi backpropagation untuk satu cell.



**Fig. 3.** Backpropagation satu cell RNN

Fungsi perhitungan backpropagation sebagai berikut:

$$\frac{\partial \tanh(x)}{\partial x} = 1 - \tanh(x)^2 \quad (3)$$

$$\frac{\partial a^{(t)}}{\partial W_{ax}} = \left(1 - \tanh(W_{ax}x^{(t)} + W_{aa}a^{(t-1)} + b^2)\right) x^{(t)T} \quad (4)$$

$$\frac{\partial a^{(t)}}{\partial W_{aa}} = \left(1 - \tanh(W_{ax}x^{(t)} + W_{aa}a^{(t-1)} + b^2)\right) a^{(t-1)T} \quad (5)$$

$$\frac{\partial a^{(t)}}{\partial b} = \sum_{batch} \left(1 - \tanh(W_{ax}x^{(t)} + W_{aa}a^{(t-1)} + b^2)\right) \quad (6)$$

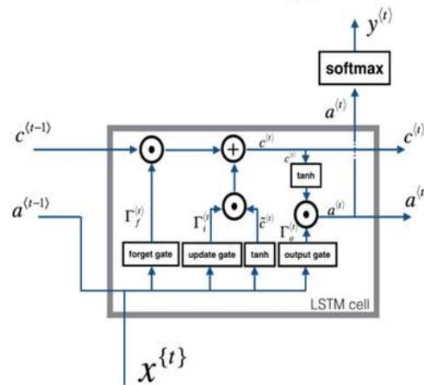
$$\frac{\partial a^{(t)}}{\partial x^{(t)}} = W_{ax}^T \left(1 - \tanh(W_{ax}x^{(t)} + W_{aa}a^{(t-1)} + b^2)\right) \quad (7)$$

$$\frac{\partial a^{(t)}}{\partial a^{(t-1)}} = W_{aa}^T \left(1 - \tanh(W_{ax}x^{(t)} + W_{aa}a^{(t-1)} + b^2)\right) \quad (8)$$

Proses backpropagation dilakukan dari cell paling terakhir hingga cell paling awal. Setelah proses backpropagation selesai, nilai pada bobot-botonyanya akan direvisi.

RNN memiliki kekurangan pada nilai gradient descent semakin mendekati input pertama maka semakin tidak memiliki pengaruh dalam mempengaruhi perubahan bobot, sehingga proses learningnya jelek, hal ini dinamakan dengan vanishing gradient, solusi terhadap permasalahan ini adalah dengan Long Short Term Memory (LSTM).

Proses LSTM secara umum sama dengan dengan RNN, bedanya di perhitung forward dan backward di setiap cell. Jumlah parameter yang digunakan saat perhitungan juga lebih banyak dibanding RNN. LSTM memiliki tiga output pada setiap cellnya, yaitu output hasil prediksi, output hidden state, dan output cell state. Output cell state menjadi input pada cell selanjutnya.



**Fig. 4.** Fordward propagation satu cell LSTM

Fungsi perhitungan forward propagationnya sebagai berikut :

$$\Gamma_f^{(t)} = \sigma(W_f[a^{(t-1)}, x^{(t)}] + b_f) \quad (9)$$

$$\Gamma_u^{(t)} = \sigma(W_u[a^{(t-1)}, x^{(t)}] + b_u) \quad (10)$$

$$\tilde{c}^{(t)} = \tanh(W_c[a^{(t-1)}, x^{(t)}] + b_c) \quad (11)$$

$$c^{(t)} = \Gamma_f^{(t)} \circ c^{(t-1)} + \Gamma_u^{(t)} \circ \tilde{c}^{(t)} \quad (12)$$

$$\Gamma_o^{(t)} = \sigma(W_o[a^{(t-1)}, x^{(t)}] + b_o) \quad (13)$$

$$a^{(t)} = \Gamma_o^{(t)} \circ \tanh(c^{(t)}) \quad (14)$$

Fungsi perhitungan backpropagationnya sebagai berikut :

$$d\Gamma_o^{(t)} = da_{next} * \tanh(c_{next}) * \Gamma_o^{(t)} * (1 - \Gamma_o^{(t)}) \quad (15)$$

$$d\tilde{c}^{(t)} = dc_{next} * \Gamma_u^{(t)} + \Gamma_o^{(t)}(1 - \tanh(c_{next})^2) * i_t * da_{next} * \tilde{c}^{(t)} * (1 - \tanh(\tilde{c})^2) \quad (16)$$

$$d\Gamma_u^{(t)} = dc_{next} * \tilde{c}^{(t)} + \Gamma_o^{(t)}(1 - \tanh(c_{next})^2) * \tilde{c}^{(t)} * da_{next} * \Gamma_u^{(t)} * (1 - \Gamma_u^{(t)}) \quad (17)$$

$$d\Gamma_f^{(t)} = dc_{next} * \tilde{c}_{prev} + \Gamma_o^{(t)}(1 - \tanh(c_{next})^2) * \tilde{c}_{prev} * da_{next} * \Gamma_f^{(t)} * (1 - \Gamma_f^{(t)}) \quad (18)$$

$$dW_f = d\Gamma_f^{(t)} * \left(\frac{a_{prev}}{x_t}\right)^T \quad (19)$$

$$dW_u = d\Gamma_u^{(t)} * \left(\frac{a_{prev}}{x_t}\right)^T \quad (20)$$

$$dW_c = d\tilde{c}^{(t)} * \left(\frac{a_{prev}}{x_t}\right)^T \quad (21)$$

$$dW_o = d\Gamma_o^{(t)} * \left(\frac{a_{prev}}{x_t}\right)^T \quad (22)$$

$$da_{prev} = W_f^T * d\Gamma_f^{(t)} + W_u^T * d\Gamma_u^{(t)} + W_c^T * d\tilde{c}^{(t)} + W_o^T * d\Gamma_o^{(t)} \quad (23)$$

$$dc_{prev} = dc_{next} \Gamma_f^{(t)} + \Gamma_o^{(t)} * (1 - \tanh(c_{next})^2) * \Gamma_f^{(t)} * da_{next} \quad (24)$$

$$dx^{(t)} = W_f^T * d\Gamma_f^{(t)} + W_u^T * d\Gamma_u^{(t)} + W_c^T * d\tilde{c}_t + W_o^T * d\Gamma_o^{(t)} \quad (25)$$

## 2.1 Pasar Modal

Pada tugas ini, LSTM digunakan untuk memprediksi harga saham pada waktu tertentu. Pembuatan LSTM menggunakan library Keras. Tugas ini terbagi atas dua, yaitu menggunakan data set Tata Globe dan Nikkei 225.

**Tata Globe.** Proses pembangunan model prediksi diawali dengan meload dataset. Kemudian data yang diambil hanya bagian atribut “Open”. Selanjutnya dilakukan proses normalisasi rentang data menjadi rentang 0 – 1 untuk mengoptimasi performa pada saat pembuatan model. setelah itu, dibuat data dengan format satu intance terdiri atas 60 timestamp, dimana satu timestamp dijadikan sebagai satu fitur.

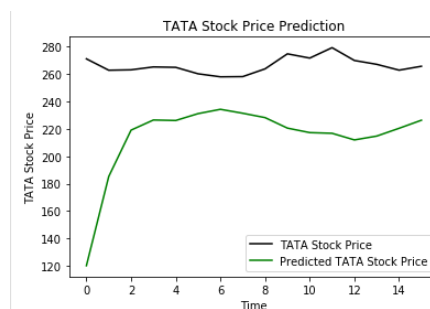
T1	-->	T1	T2	T3	...	T60
T2		T2	T3	T4	...	T61
T3		T3	T4	T5	...	T62
...		T4	T5	T6	...	T63
Tn		T5	T6	T7	...	T64

**Fig. 5.** Format data 60 timestamp untuk satu instance

Selanjutnya model dibangun dengan konfigurasi sebagai berikut:

- Hidden Layer 1 – 3 berupa LSTM dengan jumlah node = 50 dimana seluruh node jadi ouput, dan dropout layer dengan peluang 20%.
- Hidden Layer 4 berupa LSTM dengan jumlah node = 50 dimana node yang paling ujung yang hanya jadi output, dan dropout layer dengan peluang 20%.
- Hidden Layer 5 berupa layer biasa yang terdiri dari 1 node.
- Loss function = mean squared error
- Optimizer = adam
- Epoch = 100
- Batch size = 32

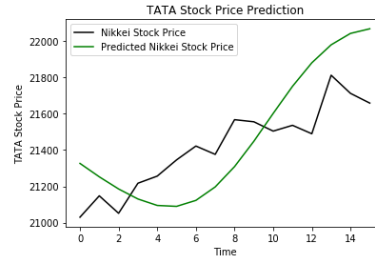
Setelah model ditrain dengan data train, model kemudian diuji dengan data test dan hasil tes divisualisasikan dalam bentuk line chart yang menggambarkan hasil prediksi dan aktual.



**Fig. 6.** Grafik antara hasil prediksi dan nilai sebenarnya pada data Tata Globe

**Nikkei 225.** Pembuatan model dimulai dengan meload data set. Kemudian membagi data set menjadi data train dan data test dengan komposisi 90% dan 10%. Proses ini tidak dilakukan di data set sebelumnya karena data set nya dari awal sudah terdiri dari data train dan data test. Untuk tahap selanjutnya sama dengan data set sebelumnya mulai dari pemilihan atribut, normalisasi, mengubah format data,

membangun model, melatih model dengan data train, dan menguji model dengan data test. Berikut hasil tesnya:



**Fig. 7.** Grafik antara hasil prediksi dan nilai sebenarnya pada data Nikkei 225

## 2.1 Dinosaur

Tugas ini diminta untuk membuat model yang dapat me-generate nama dinosaurus secara random. Model dibangun menggunakan RNN. Model dilatih menggunakan dataset yang berisi nama-nama dinosaurus.

Sebelum membuat model, preprocessing dilakukan pada dataset. Preprocessing pertama yaitu dengan menghitung jumlah charater dan jumlah jenis character yang berada pada dataset. Kemudian dibuat kamus dimana kamus tersebut berisi setiap jenis character yang dipetakan ke sebuah angka. Selanjutnya dari kamus ini juga nanti digunakan untuk memetakan character ke angka dan sebaliknya pada saat proses pembangunan model.

Character	Angka	Character	Angka
\n	0	n	14
a	1	o	15
b	2	p	16
c	3	q	17
d	4	r	18
e	5	s	19
f	6	t	20
g	7	u	21
h	8	v	22
i	9	w	23
j	10	x	24
k	11	y	25
l	12	z	26
m	13		

**Fig. 8.** Kamus pemetaan karakter ke angka

Proses pembuatan model menggunakan RNN. Tahapannya sama seperti sebelumnya, hanya saja sebelum parameter diupdate, nilai gradient akan di clip agar

nilainya tidak terlalu besar dan terlalu kecil (exploding). Dengan clip, nilai gradient tidak akan melewati batas yang ditentukan, jika nilai gradient melewati batas maka nilai gradient tersebut diubah sama dengan batas yang ditentukan.

Satu fungsi tambahan dibuat pada tugas ini yang berbeda dari tugas sebelumnya yaitu sampling. Sampling ini berfungsi untuk mengubah output dari model yang sudah dilatih menjadi nama. Caranya adalah memilih satu angka yang merepresentasikan character dari rentang angka yang sudah dibuat sebelumnya dimana peluang kemunculan masing-masing angka ditentukan dari output satu cell pada RNN. angka yang terpilih kemudian diubah menjadi karakter sesuai mappingnya. selanjutnya angka yang terpilih kemudian dijadikan input pada cell berikutnya untuk mendapat karakter berikutnya. Proses generate nama berhenti jika panjang karakter telah mencapai 50 atau karakter yang digenerate berupa new line (\n). Untuk input awal nilainya 0.

Model dilatih sebanyak 35000 iterasi menggunakan data latih yang sama dimana setiap 2000 iterasi, model dites dengan fungsi sampling untuk me-generate beberapa nama dinosaurus. Makin lama iterasi, nama dinosaurus yang digenerate makin bagus.

### 3 Kesimpulan

RNN digunakan untuk menangani kasus dimana input dan output tidak dapat ditentukan dari awal. Ada beberapa teknik yang dapat digunakan dalam implementasi RNN, antara lain LSTM yang berguna untuk menghindari gradient vanishing, Dropout Layer untuk menghindari overfitting, dan Clip untuk menghindari exploding.

### References

1. Brownlee, J. (2018, December 3). A Gentle Introduction to Dropout for Regularizing Deep Neural Networks. Machine Learning Mastery: <https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/> adresinden alındı
2. Keras LSTM tutorial – How to easily build a powerful deep learning language model. (2019, February 3). Adventures in Machine Learning: <https://adventuresinmachinelearning.com/keras-lstm-tutorial/> adresinden alındı