

Pattern Recognition: Recurrent Neural Networks (RNN)

Arrival Dwi Sentosa
(23519035)
Institut Teknologi Bandung
23519035@std.stei.itb.ac.id

Abstract. *Recurrent Neural Networks* (RNN) salah satu teknik dalam machine learning terkhusus deep learning, dengan kriteria jaringan yang input dan output nya belum diketahui. GNU dan LSTM adalah dua variasi dari RNN, dan pada makalah ini akan dibahas implementasi dari tiga studi kasus dengan RNN dan LSTM.

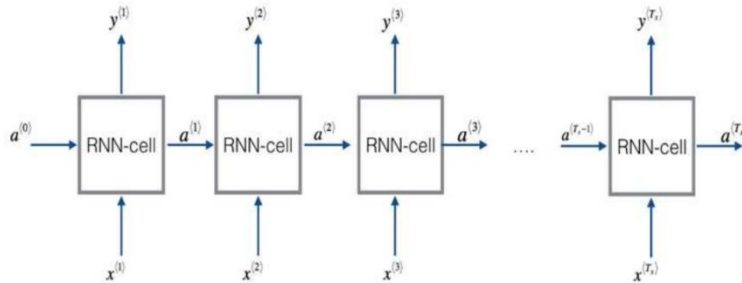
1 Pendahuluan

Recurrent Neural Network (RNN) adalah kelas dari jaringan saraf tiruan dimana koneksi antara nodes membentuk grafik diarahkan sepanjang urutan temporal, ini memungkinkannya perilaku dinamis temporal. Tidak seperti jaringan neural *feed forward*, RNNs dapat menggunakan status internal (memori) untuk memproses urutan input. Ini membuatnya bisa digunakan untuk tugas-tugas seperti pengenalan tulisan tangan yang tidak tersegmentasi, *connected handwriting recognition* [1] atau *speech recognition* [2], [3].

2 Hands-On RNN and LSTM

2.1 RNN dan LSTM Basic

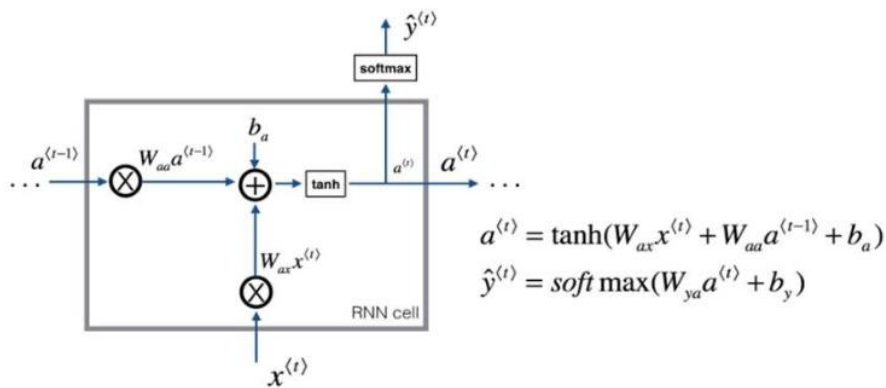
RNN dasar yang akan Anda implementasikan memiliki struktur di bawah ini.



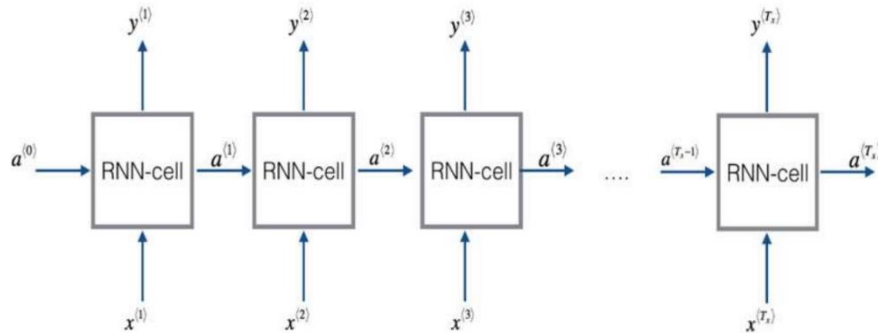
Langkah:

- Melaksanakan perhitungan yang diperlukan untuk satu langkah waktu RNN.
- Terapkan loop over overTx langkah-waktu untuk memproses semua font sizes are also given in Table 1. Kindly refrain from using “0” when numbering your section headings.

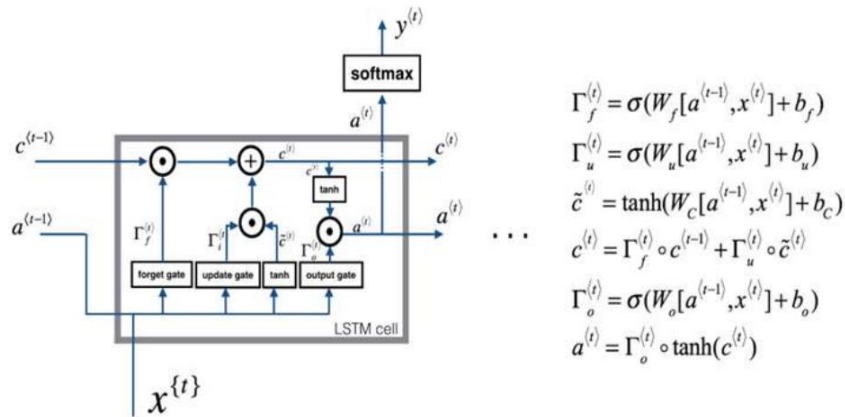
Jaringan saraf berulang dapat dilihat sebagai pengulangan dari sel tunggal. Anda pertama-tama akan mengimplementasikan perhitungan untuk *single time-step*. Gambar berikut menjelaskan operasi untuk *single time-step* sel RNN.



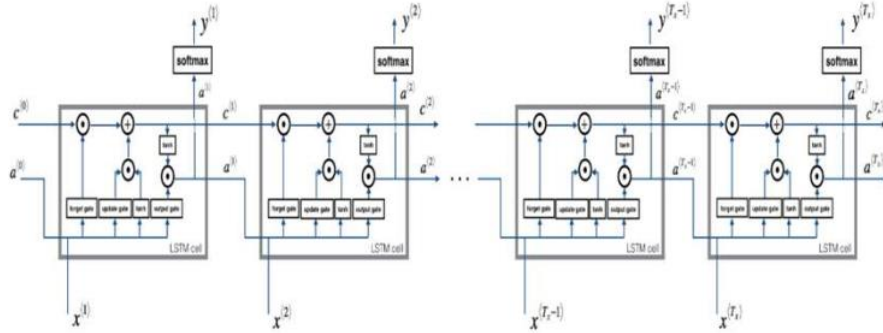
Setiap sel mengambil input keadaan tersembunyi dari sel sebelumnya ($a^{(t-1)}$) dan langkah-waktu saat ini input data ($x^{(t)}$). Ini menghasilkan status tersembunyi ($a^{(t)}$) dan prediksi ($y^{(t)}$) untuk langkah-waktu ini.



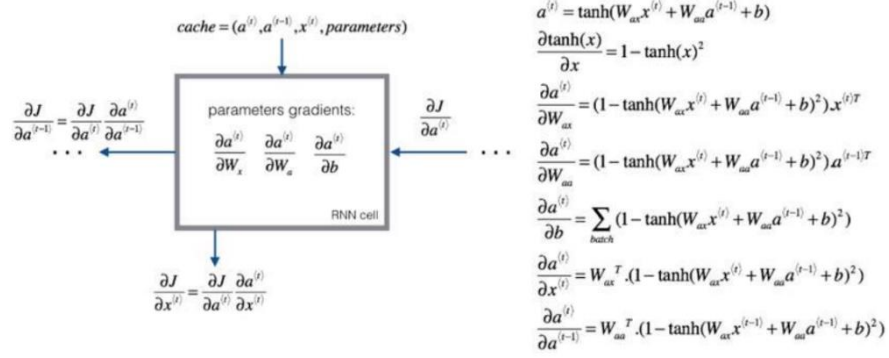
LSTM-cell. Ini melacak dan memperbarui "status sel" atau variabel memori $c(t)$ pada setiap langkah waktu, yang bisa berbeda dari $a(t)$



Sekarang Anda telah menerapkan satu langkah dari LSTM, Anda sekarang dapat mengulanginya di atas ini menggunakan for-loop untuk memproses urutan input $T \times T \times x$.



Basic RNN Backward Pass



LSTM Backward Pass

gate derivatives

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

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

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

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

parameter derivatives

$$\begin{aligned} dW_f &= d\Gamma_f^{(t)} * \begin{pmatrix} a_{prev} \\ x_t \end{pmatrix}^T \\ dW_u &= d\Gamma_u^{(t)} * \begin{pmatrix} a_{prev} \\ x_t \end{pmatrix}^T \\ dW_c &= d\tilde{c}^{(t)} * \begin{pmatrix} a_{prev} \\ x_t \end{pmatrix}^T \\ dW_o &= d\Gamma_o^{(t)} * \begin{pmatrix} a_{prev} \\ x_t \end{pmatrix}^T \end{aligned}$$

2.2 Mangosaurus

Pada tugas ini diminta membuat sebuah model untuk melakukan generate nama-nama baru untuk spesies dinosaurus secara acak berdasarkan nama-nama dinosaurus yang telah ada sebelumnya dengan menggunakan RNN. Dan data train berisi nama-nama dinosaurus dataset. Dan diperlukan preprocessing pada dataset tersebut dengan membaca dataset (dinos.txt) setelah itu dilakukan refactoring dengan mengubah semua data menjadi lowercase, menghitung jumlah karakter dan jumlah jenis karakter yang dipetakan tiap karakter antara 0 sampai 26 lalu memetakan kembali nilai indeks ke dictionary.

Proses pembangunan model RNN pada kasus ini adalah melakukan clip pada nilai gradient. Pada bagian ini akan mengimplementasikan fungsi yang melakukan satu langkah penurunan gradien stokastik (dengan terpotong gradien). pelatihan satu per satu, sehingga algoritma pengoptimalan akan menjadi penurunan gradien stokastik. berikut adalah langkah-langkah loop pengoptimalan umum untuk RNN:

- Maju merambat melalui RNN untuk menghitung kerugian
- Merambat mundur melalui waktu untuk menghitung gradien kerugian sehubungan dengan parameter
- Klip gradien jika perlu
- Perbarui parameter Anda menggunakan gradient descent

Pelatihan model mengingat dataset nama dinosaurus, kami menggunakan setiap baris dataset (satu nama) sebagai satu contoh pelatihan. Setiap 100 langkah keturunan gradien stokastik, 10 nama yang dipilih secara acak untuk melihat bagaimana algoritma berjalan. Sebanyak 35000 iterasi menggunakan train data yang sama dimana 2000 iterasi, model menggunakan sampling untuk membuat nama dinosaurus baru.

2.3 Pasar Modal

Tugas melakukan prediksi harga saham dengan menggunakan LSTM, ada dua kasus yaitu prediksi saham menggunakan dataset Tata globe dan saham NASDAQ dengan tambahan library Keras.

Tata Globe

Langkah awal adalah dengan mengambil dataset “NSE-TATAGLOBAL.csv” lalu memilih kolom Open dan High yang akan digunakan sebagai model. Kolom Buka adalah harga awal sedangkan kolom Tutup adalah harga akhir suatu saham pada hari perdagangan tertentu. Tinggi dan Rendah kolom mewakili harga tertinggi dan terendah untuk hari tertentu. Library Keras digunakan untuk membangun Model LSTM. Terdiri dari 4 layer LSTM dengan dropout layers (rate: 0.2) dan 1 layer dense untuk menghasilkan 1 output. Tiap layer LSTM memiliki 50 units

NASDAQ

Melakukan dataset dari NASDAQ.csv lalu pembagian hold-out untuk data train 80% dan data test 20%, untuk proses lainnya sama seperti yang dilakukan pada kasus Tata Globe.

3 Kesimpulan

Pada eksperimen/tugas diatas yaitu Dinosaur name generator (language modeling) dan prediksi saham adalah contoh tepat implementasi dari RNN dan LSTM karena kasus diatas dimana input dan output tidak bisa ditentukan dari awal. Dengan memahami beberapa teknik yang dapat digunakan dalam upaya meningkatkan tingkat akurasi yang dicapai seperti menghindari dropout layer, exploding, dan lainnya.

References

- [1] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, “A novel connectionist system for unconstrained handwriting recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, 2009.
- [2] H. Sak, A. Senior, and F. Beaufays, “Long short-term memory recurrent neural network architectures for large scale acoustic modeling,” in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2014.
- [3] X. Li and X. Wu, “Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition,” in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2015.