

Digital Talent Scholarship 2022

Natural Language Processing Pt 3

Lead a sprint through the Machine Learning Track

Agenda

- Introduction to Sequence model & Literature
- Next steps

Are your students ML-ready?

Recap

Sequence Model

What is sequence model?

Sequence model adalah pemrosesan pada input yang berurutan, misalnya pemrosesan bahasa alami (NLP), audio, atau data sekuensial lainnya. Dalam konteks deep learning, model yang berkaitan adalah recurrent neural network (RNN)

LSTM

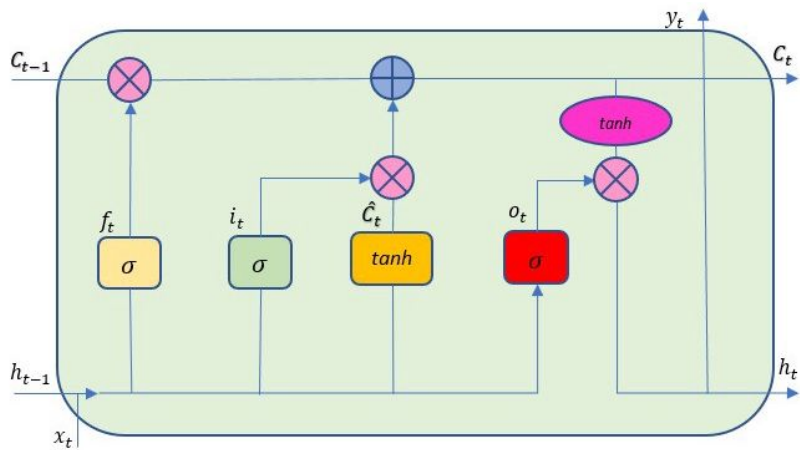
What is LSTM?

Long Short-Term Memory (LSTM) adalah salah satu jenis dari Recurrent Neural Network (RNN) dimana dilakukan modifikasi pada RNN dengan menambahkan memory cell yang dapat menyimpan informasi untuk jangka waktu yang lama. LSTM diusulkan sebagai solusi untuk mengatasi terjadinya vanishing gradient pada RNN saat memproses data sequential yang panjang

Arsitektur LSTM

LSTM

t-1



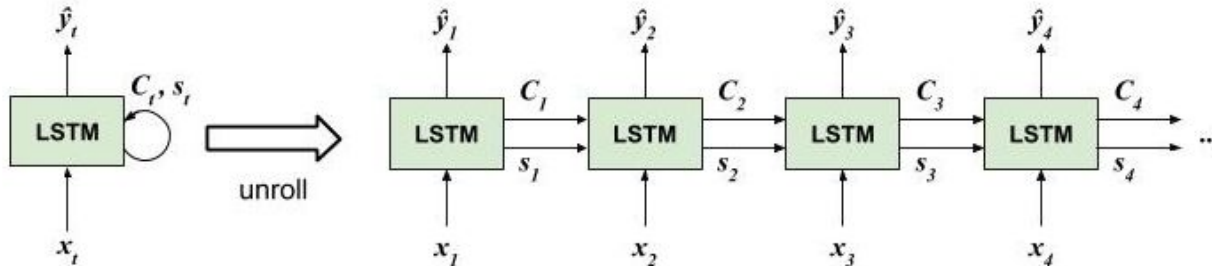
LSTM

t+1

LSTM

LSTM diciptakan oleh [Hochreiter & Schmidhuber \(1997\)](#) dan kemudian dikembangkan dan dipopulerkan oleh banyak periset.

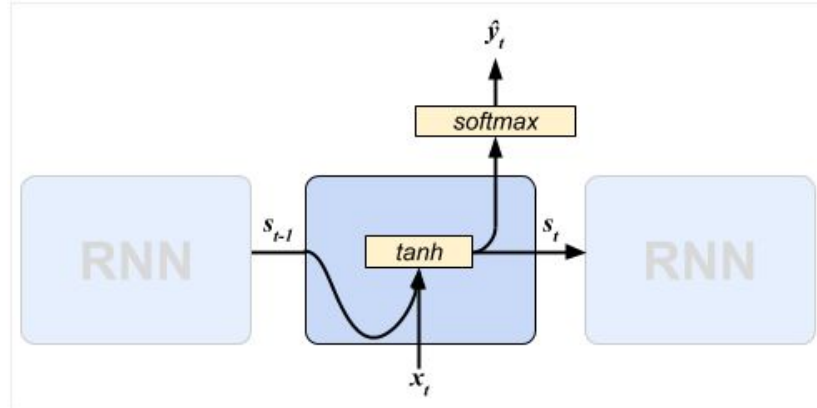
Modul LSTM (satu kotak hijau) mempunyai pemrosesan yang berbeda dengan modul RNN biasa. Perbedaan lain adalah adanya tambahan sinyal yang diberikan dari satu langkah waktu ke langkah waktu berikutnya, yaitu konteks, direpresentasikan dengan simbol C_t



Jaringan LSTM terdiri dari modul LSTM yang dipanggil secara berulang

Pemrosesan pada LSTM

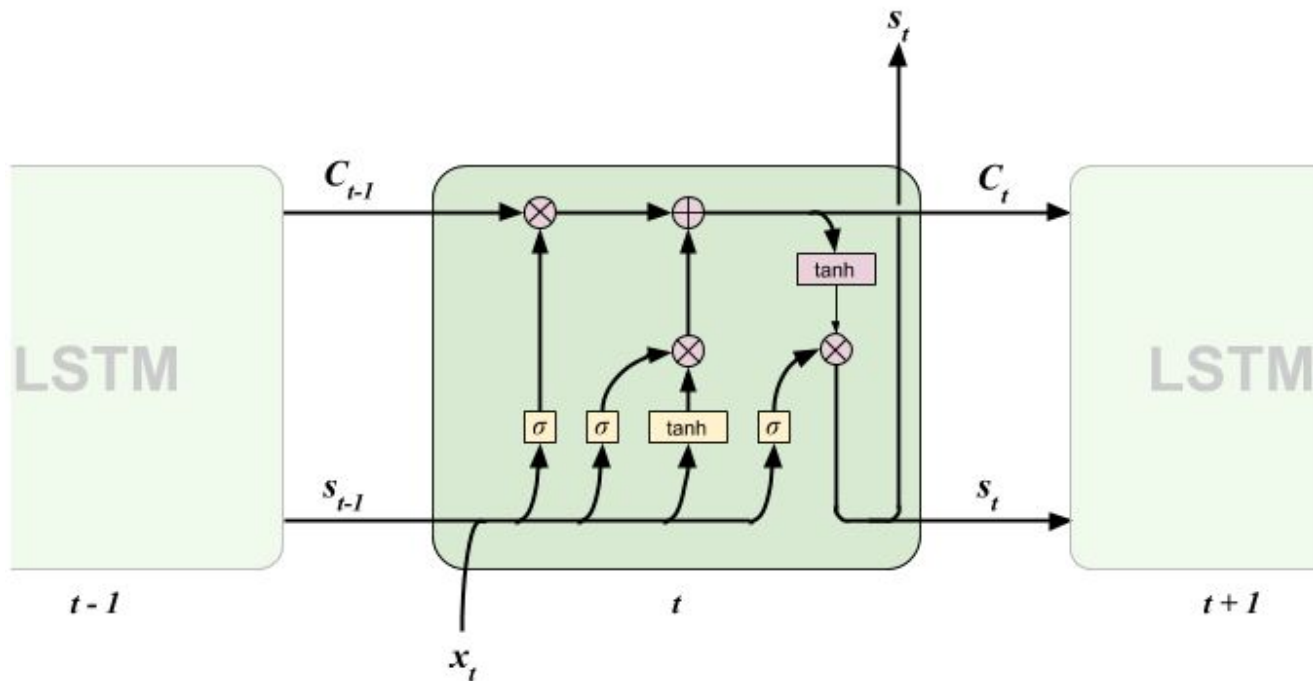
pemrosesan dalam satu modul RNN cukup sederhana, hanya ada satu lapis tanh, dan kalau dikehendaki output dari modul itu maka state internal akan dilewatkan pada fungsi aktivasi seperti softmax untuk mendapatkan output \hat{y}_t :



$$s_t = \tanh(U \cdot x_t + W \cdot s_{t-1})$$

$$\hat{y}_t = \text{softmax}(V \cdot s_t)$$

Sedangkan modul LSTM mengandung lebih banyak komputasi:



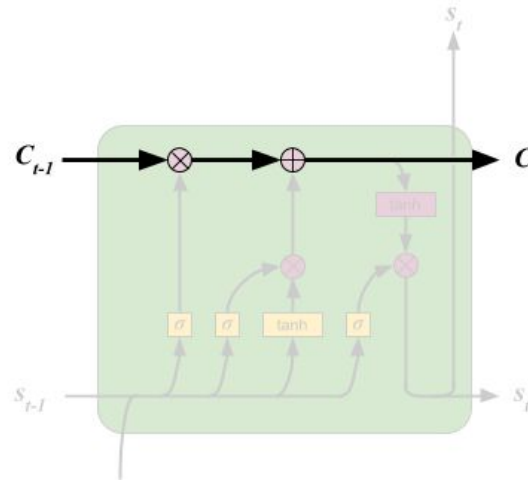
Terlihat rumit bukan? Mari kita bahas satu per satu

Notasi



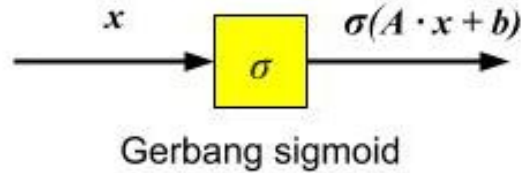
Mekanisme Kunci LSTM

Ide kunci dari LSTM adalah jalur yang menghubungkan konteks lama (C_{t-1}) ke konteks baru (C_t) di bagian atas modul LSTM

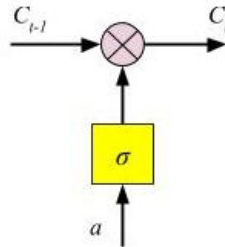


Mekanisme Kunci LSTM

Ide kunci yang lain adalah adanya gerbang sigmoid (sigmoid gate) yang mengatur seberapa banyak informasi bisa lewat.

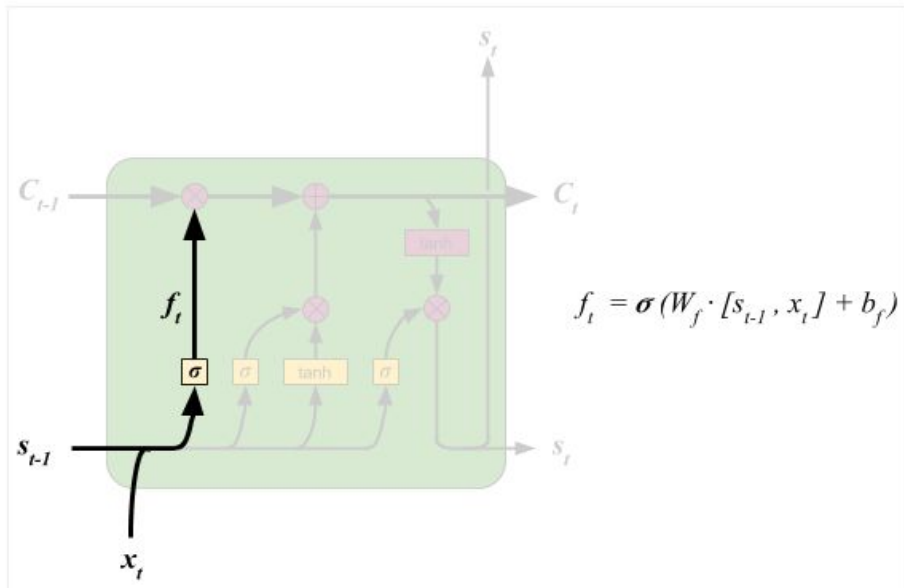


Output dari gerbang sigmoid akan dikalikan dengan suatu nilai lain untuk mengontrol seberapa banyak nilai tersebut dipakai. Dengan gerbang sigmoid, LSTM bisa mengatur seberapa banyak informasi dari C_{t-1} yang diikutkan menjadi C_t .



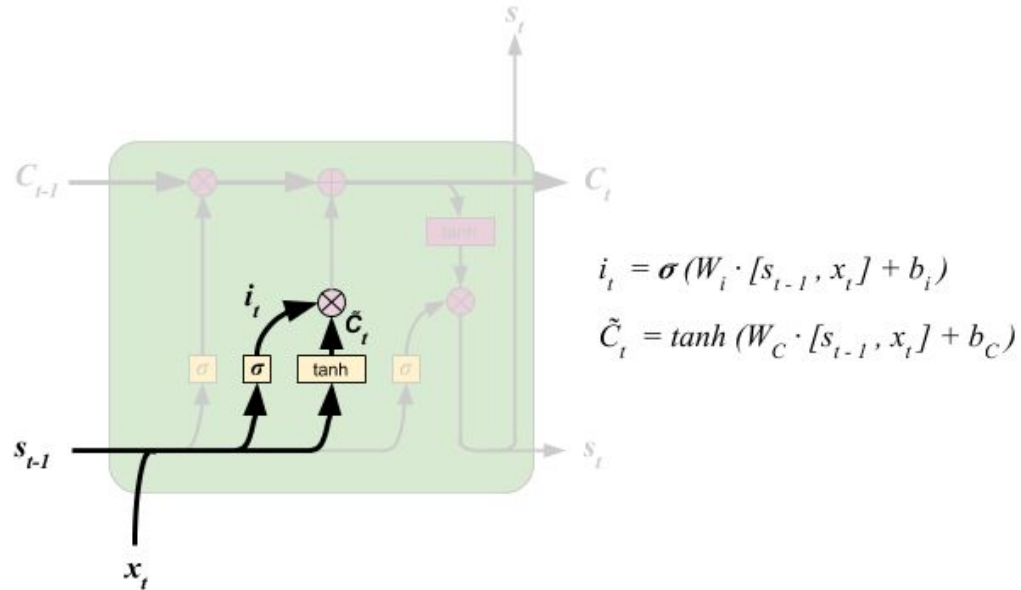
Langkah 1

Memutuskan informasi apa yang akan dibuang dari konteks C_{t-1} , dengan menggunakan gerbang sigmoid yang kita sebut “gerbang lupa” (forget gate, f_t).



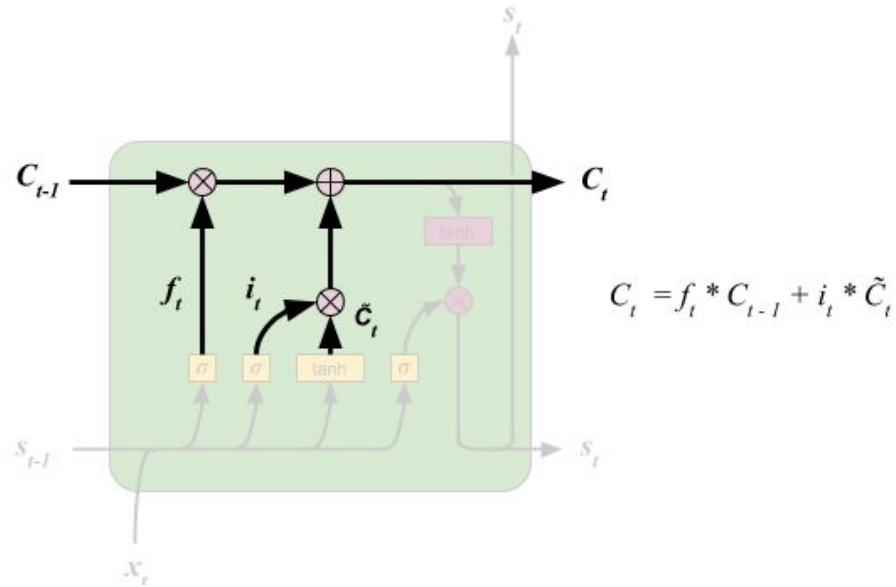
Langkah 2

Memutuskan informasi baru apa yang akan kita gunakan di C_t . Proses ini memiliki dua bagian



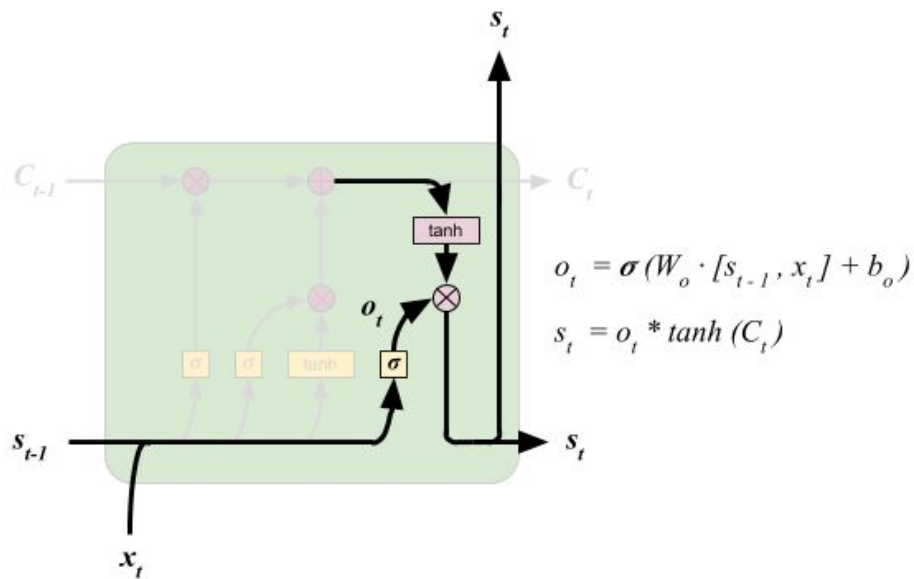
Langkah 3

Memperbarui konteks lama C_{t-1} ke konteks baru C_t

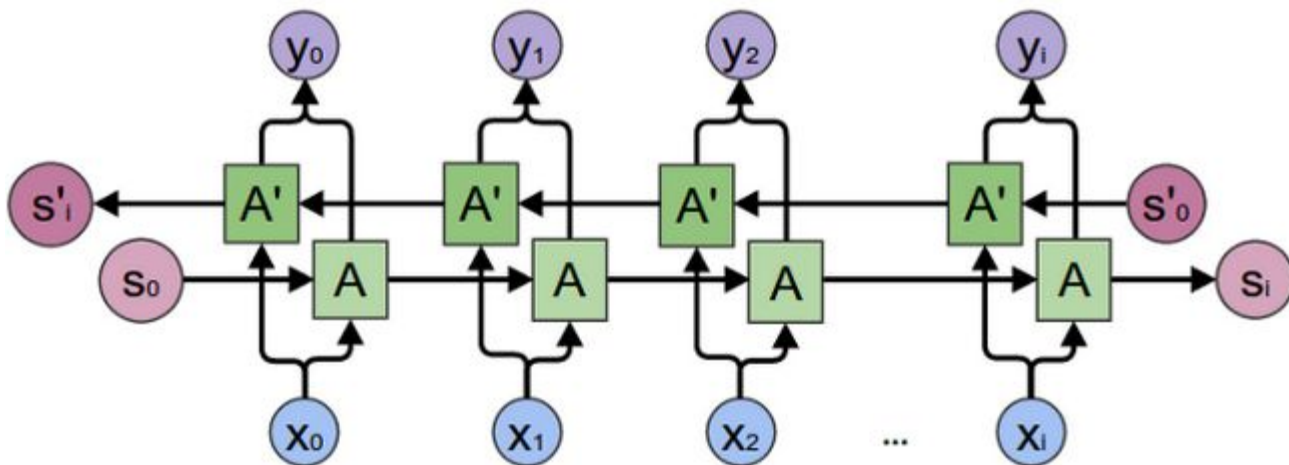


Langkah 4

Memutuskan apa yang akan kita hasilkan



Bidirectional LSTM menjalankan input dalam dua cara, forward dan backward.



Gated Recurrent Unit (GRU)

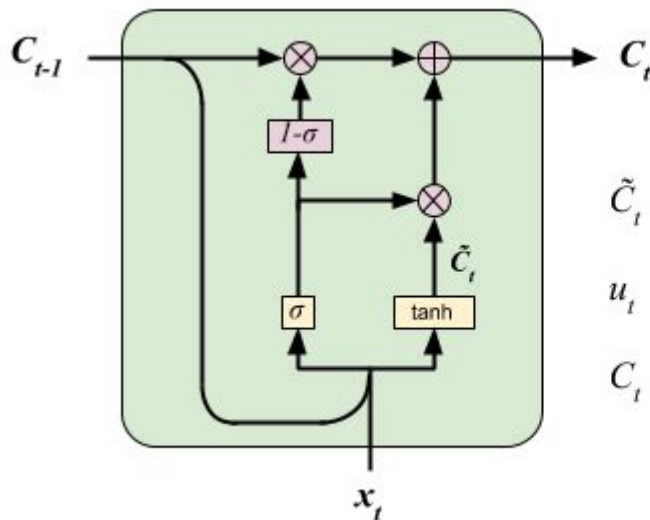
GRU

LSTM mempunyai banyak sekali varian, misalnya LSTM dengan koneksi lubang intip (peephole connection), LSTM yang menggabungkan gerbang input dengan gerbang lupa, dan sebagainya. Salah satu varian yang populer adalah Gated Recurrent Unit atau disingkat GRU. GRU dimunculkan dalam makalah oleh Cho dkk (2014) dan Chung dkk (2014)

Kelebihan Gru

Komputasi pada GRU lebih sederhana dari LSTM, namun mempunyai akurasi yang setara dan masih cukup efektif untuk menghindari permasalahan gradien yang menghilang.

GRU

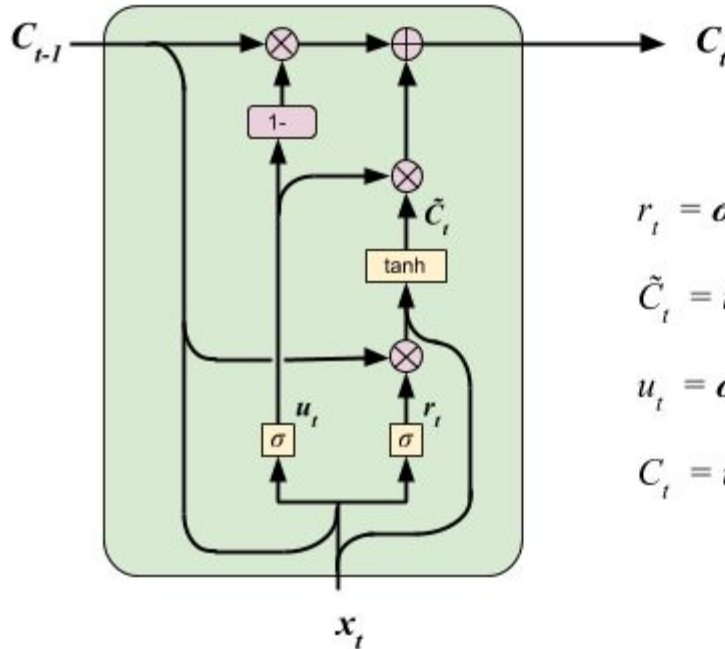


$$\tilde{C}_t = \tanh(W_C \cdot [C_{t-1}, x_t] + b_C)$$

$$u_t = \sigma(W_u \cdot [C_{t-1}, x_t] + b_u)$$

$$C_t = u_t * \tilde{C}_t + (1 - u_t) * C_{t-1}$$

Komputasi Gru yang lengkap



$$r_t = \sigma(W_r \cdot [C_{t-1}, x_t] + b_r)$$

$$\tilde{C}_t = \tanh(W_c \cdot [r_t * C_{t-1}, x_t] + b_c)$$

$$u_t = \sigma(W_u \cdot [C_{t-1}, x_t] + b_u)$$

$$C_t = u_t * \tilde{C}_t + (1 - u_t) * C_{t-1}$$



Demo

Q n A

Natural Language Processing (NLP) Tutorial with Python & NLTK - YouTube

Thank You