



KECERDASAN BUATAN

PERTEMUAN KE - 13

ENDANG SRI RAHAYU

Teknik Elektro – Fakultas Teknologi Industri – Universitas Jayabaya

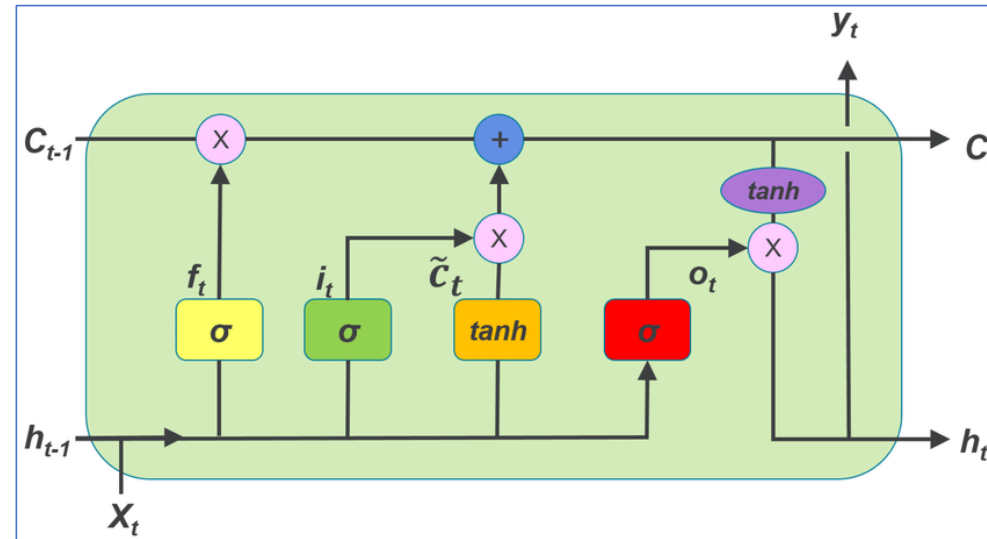
Outlines:

Kasus:

- Analisis Data Time Series menggunakan Neural Network
- Presentasi Tugas Implementasi LSTM (Long Short Term Memory)

Long Short Term Memory (LSTM)

Ilustrasi 1 Cell

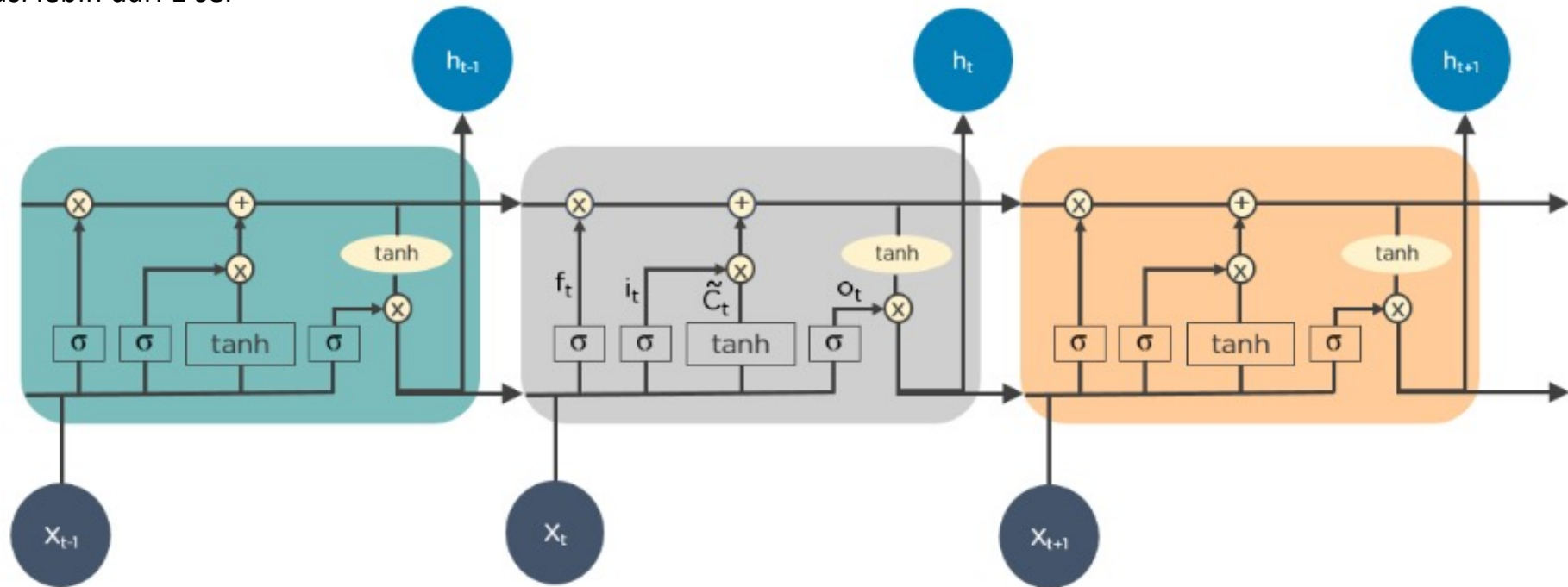


Gate:

- Memory
- Input / Last Output
- Forget
- New Knowledge Creation
- Output



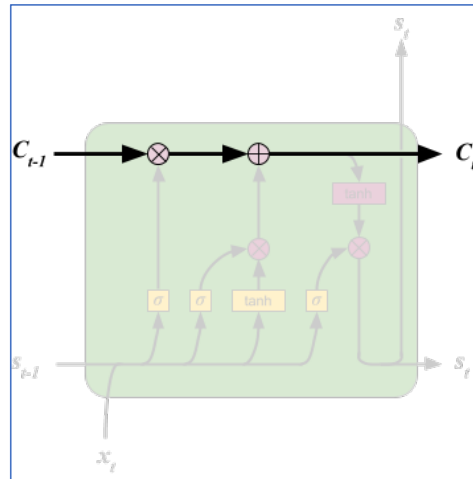
Ilustrasi lebih dari 1 sel



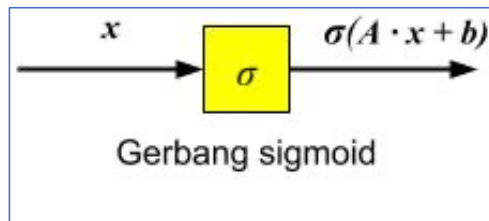
LSTM diciptakan oleh [Hochreiter & Schmidhuber \(1997\)](#)

Varian dari RNN

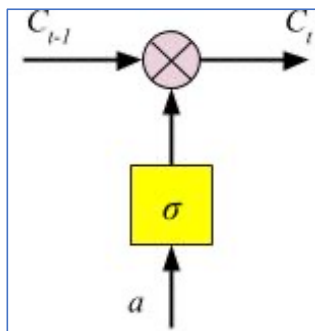
Perbedaan: adanya tambahan sinyal yang diberikan dari satu langkah waktu ke langkah waktu berikutnya, yaitu Memory Cell / Cell state, direpresentasikan dengan simbol C_t



Konteks C_t disebut juga *cell state* atau *memory cell* dalam beberapa artikel. Dengan adanya jalur di atas, suatu nilai di konteks yang lama akan dengan mudah diteruskan ke konteks yang baru dengan sedikit sekali modifikasi, kalau diperlukan

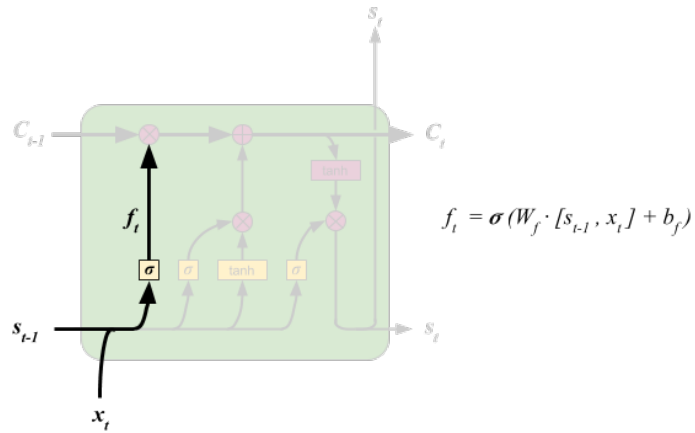


Untuk suatu input x , output dari gerbang sigmoid adalah $\sigma(A \cdot x + b)$, dimana A adalah parameter, b adalah bias, keduanya dipelajari dalam proses latihan, dan σ adalah fungsi sigmoid. Keluaran gerbang adalah angka antara nol dan satu; nol artinya informasinya diblok total, sedangkan satu artinya ikutkan keseluruhan informasi.

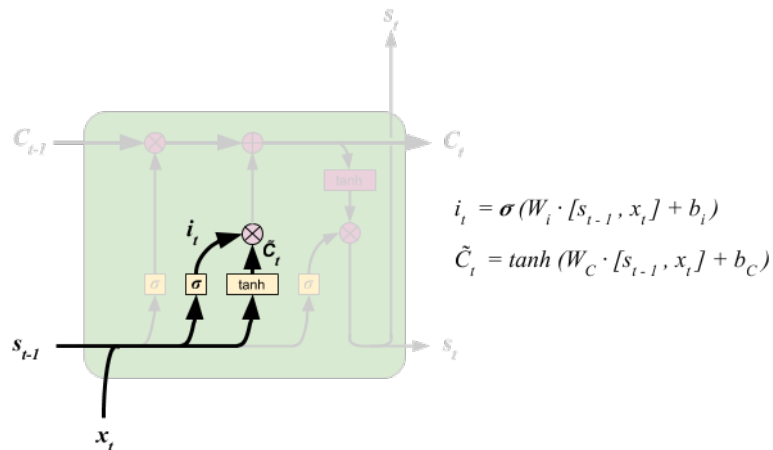


Dengan gerbang sigmoid di atas, LSTM bisa mengatur seberapa banyak informasi dari C_{t-1} yang diikutkan menjadi C_t

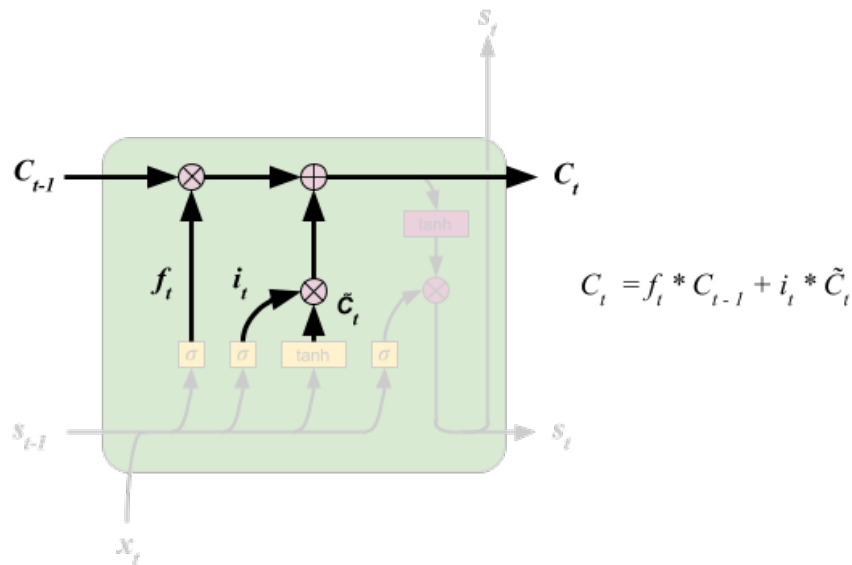
Langkah-langkah LSTM



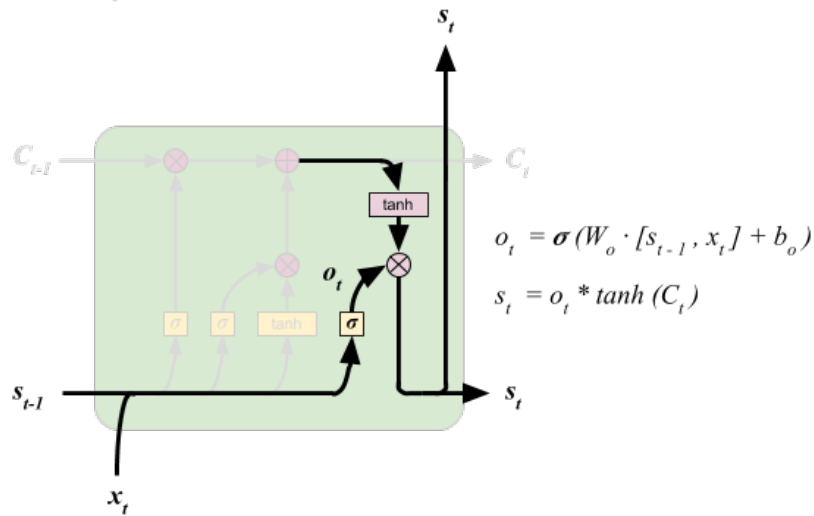
1. memutuskan informasi apa yang akan dibuang dari konteks C_{t-1} , dengan menggunakan gerbang sigmoid yang kita sebut “gerbang lupa” (*forget gate*, f_t). Gerbang ini membaca nilai s_{t-1} dan x_t , dan menghasilkan angka antara 0 dan 1 untuk setiap elemen dalam C_{t-1} . Nilai 1 artinya “diingat” sementara 0 artinya “dilupakan.”



2. memutuskan informasi baru apa yang akan kita gunakan di C_t . Proses ini memiliki dua bagian. Pertama, gerbang sigmoid yang disebut “gerbang masukan” (*input gate*, i_t) memutuskan nilai mana yang akan kita perbarui. Lalu sebuah lapis *tanh* menghasilkan kandidat vektor konteks baru, \tilde{c}_t , (dibaca: C tilde). Lalu kita gabungkan keduanya untuk membuat pembaruan ke konteks nanti.



3. memperbarui konteks lama C_{t-1} ke konteks baru C_t . Kita kalikan konteks lama dengan f_t untuk melupakan hal-hal yang kita putuskan untuk dilupakan. Kita kalikan kandidat konteks baru kita \tilde{C}_t dengan i_t untuk memutuskan seberapa banyak kita akan menyertakan kandidat konteks baru. Lalu kita tambahkan keduanya



4. memutuskan apa yang akan kita hasilkan. Output ini akan didasarkan pada nilai dalam konteks kita dan dilewatkan ke suatu filter. Pertama, kita jalankan gerbang sigmoid yang kita namakan gerbang output (*output gate*, o_t) untuk memutuskan bagian-bagian apa dari konteks yang akan kita hasilkan. Kemudian, kita lewatkan konteks melalui \tanh untuk membuat nilainya menjadi antara -1 dan 1 , dan kita kalikan dengan output gerbang sigmoid tadi sehingga kita hanya menghasilkan bagian yang kita putuskan.

Contoh implementasi LSTM (menggunakan Python)

- Presentasi contoh implementasi LSTM
- Langkah-langkah:
 - Baca data
 - Mengatur data (training 80%, testing 20%)
 - Model LSTM (LSTM, Bi-LSTM, CNN-LSTM, ...)
 - Menampilkan hasil graf (plot)
 - Menampilkan hasil akurasi, loss

Terima kasih