

BAB II

TINJAUAN PUSTAKA

2.1 *Machine Learning*

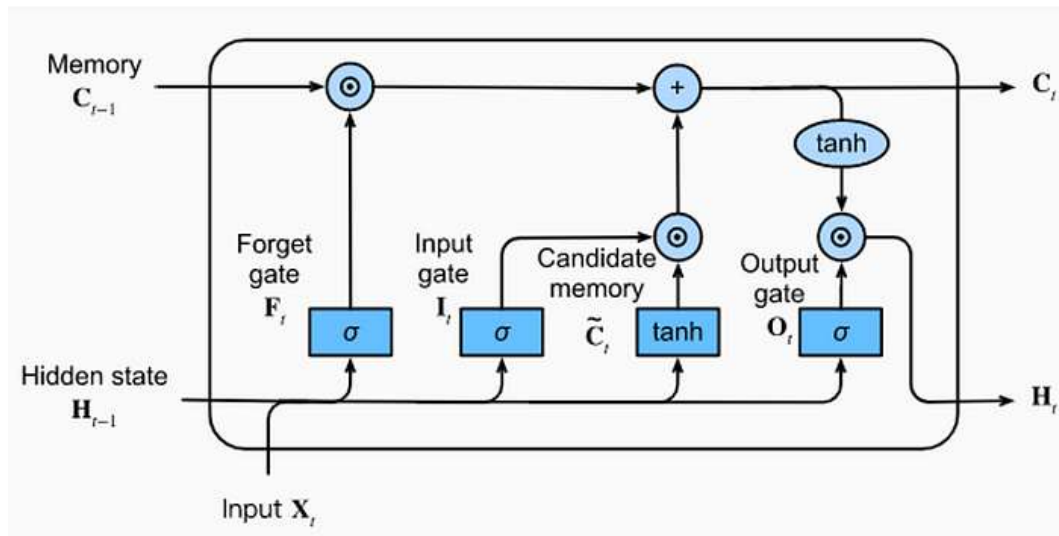
Machine learning merupakan sebuah algoritma yang bertujuan untuk menemukan dan mengaplikasikan pola-pola di dalam data. Algoritma di *machine learning* menggunakan teknik-teknik statistik untuk menemukan pola-pola tersebut. Data yang bisa diolah tidak hanya berupa teks, tetapi bisa juga berupa gambar, audio, maupun video [12].

2.2 *Recurrent Neural Network (RNN)*

Recurrent neural network atau yang biasa disebut dengan RNN merupakan jenis jaringan saraf tiruan yang biasa digunakan untuk memproses data sekuensial seperti *time series* dengan mempertahankan informasi historis. Namun, RNN konvensional mengalami masalah *vanishing gradient* sehingga sulit mengingat pola jangka panjang. Oleh karena itu, dikembangkan LSTM (*long short-term memory*) sebagai salah satu solusinya.

2.3 *Long Short-Term Memory (LSTM)*

Long short-term memory atau yang biasa disebut dengan LSTM merupakan jenis khusus dari RNN yang dirancang untuk mengatasi kelemahan dalam mengingat informasi jangka panjang. LSTM memiliki struktur sel yang kompleks yang mencakup tiga gerbang utama yaitu *input gate*, *forget gate*, dan *output gate*. Ketiga komponen ini memungkinkan LSTM mempertahankan, membuang, atau memperbarui informasi dari waktu ke waktu [4].



Gambar 2.1 Struktur Diagram LSTM

Pada gambar 2.1 di atas menggambarkan arsitektur internal yang digunakan dalam jaringan saraf tiruan untuk memproses data sekuensial.

Data *input* saat ini (X_t) dan *hidden state* sebelumnya (H_{t-1}) serta *memory cell* sebelumnya (C_{t-1}) digunakan sebagai masukan. Kemudian *Forget Gate* (F_t) Menggunakan fungsi sigmoid (σ) untuk memutuskan informasi mana dari *memory cell* sebelumnya (C_{t-1}) yang akan dilupakan.

Pada *Input Gate* (I_t) menggunakan fungsi sigmoid (σ) untuk memutuskan informasi baru mana yang akan disimpan dari *input* saat ini. Setelah itu, *Candidate Memory* (\tilde{C}_t) menggunakan tanh untuk membuat kandidat nilai baru yang akan ditambahkan ke *memory cell*. Pada *Output Gate* (O_t) menggunakan fungsi sigmoid (σ) untuk memutuskan informasi mana dari *memory cell* saat ini yang akan dikeluarkan sebagai *hidden state*. Setelah itu, *memory cell* (C_t) diperbarui dengan menggabungkan informasi dari *forget gate*, *input gate*, dan *candidate memory*. Rumusnya: $C_t = F_t \cdot C_{t-1} + I_t \cdot \tilde{C}_t$, di mana informasi lama dipertahankan atau dilupakannya berdasarkan F_t , dan informasi baru ditambahkan berdasarkan I_t dan \tilde{C}_t .

Hidden State (H_t) Dihasilkan dari *output gate* yang mengatur informasi dari *memory cell* (C_t) yang telah diproses melalui fungsi tanh. Rumusnya: $H_t = O_t \cdot \tanh(C_t)$. Alur Informasi mengalir dari C_{t-1} dan H_{t-1} , diproses melalui *gate-gate*, menghasilkan C_t dan H_t untuk langkah waktu berikutnya. Dengan arsitektur ini,

LSTM mampu menjaga konteks jangka panjang, dan sangat efektif dalam aplikasi *time series* seperti prediksi harga saham.

2.4 Bahasa Pemrograman *Python*

Python merupakan salah satu bahasa pemrograman yang populer dalam pengembangan aplikasi *machine learning* dan analisis data karena sintaksnya yang sederhana serta dukungan komunitas yang luas. *Python* mendukung paradigma pemrograman berorientasi objek dan fungsional menjadikannya fleksibel untuk berbagai kebutuhan.

Beberapa pustaka (*library*) *python* yang umum digunakan dalam pengolahan data dan *machine learning* antara lain:

- *NumPy*: untuk perhitungan numerik dan manipulasi *array*
- *Pandas*: untuk manipulasi dan analisis data dalam bentuk tabel (*data frame*)
- *Matplotlib*: untuk visualisasi data
- *Scikit-learn*: untuk model *machine learning* klasik seperti regresi, klasifikasi, dan klusterisasi
- *TensorFlow* dan *Keras*: untuk pembangunan model *deep learning* termasuk LSTM

Library ini mempermudah proses *preprocessing*, pelatihan model, evaluasi, hingga visualisasi hasil prediksi.

2.5 *Google Colab*

Google colab merupakan sebuah *platform* berbasis *cloud* yang memungkinkan pengguna dapat menulis dan menjalankan kode *python* secara langsung dari *browser*, tanpa memerlukan konfigurasi perangkat lokal. *Platform* tersebut terintegrasi dengan *google drive*, yang dapat memudahkan penggunaanya dalam menyimpan dan berbagi *notebook*.