**“SENTIMENT ANALYSIS UJAR KEBENCIAN TWEETS PENGGUNA TWITTER DI INDONESIA MENGGUNAKAN DEEP LEARNING BIDIRECTIONAL LSTM”**

**BAB II**

**TINJAUAN PUSTAKA DAN LANDASAN TEORI**

Pada Bab II ini, akan diuraikan tinjauan Pustaka dari beberapa referensi yang terkait dengan penelitian ini serta teori-teori yang mendukung dan relevan dengan penelitian ini, seperti sentiment analysis, jaringan syaraf tiruan, deep learning, LSTM, Bidirectional LSTM.

## **Tinjauan Pustaka**

Mengklasifikasi ujaran kebencian di Twitter merupakan hal yang penting agar tidak mudah tergiring oleh opini kebencian oleh oknum-oknum yang menyebarkan ujaran tersebut demi sebuah kepentingan. Oleh karena itu, dibutuhkan sebuah keahlian khusus untuk melakukan sentiment analysis/klasifikasi ujaran kebencian. Beberapa penelitian tentang sentiment analysis telah dilakukan terkait sentiment analysis twitter. Penelitian oleh (Ahmad, Ali, & Ahtab, 2017) yang membahas tentang sentiment analysis tweets menggunakan support vector machine (SVM) mendapatkan akurasi 85%. Selanjutnya terdapat juga penelitian oleh (Fauzi, 2018) yang membahas tentang penggunaan random forest untuk sentiment analysis Bahasa Indonesia mendapatkan skor OOB/akurasi sebesar 82,9%. Kemudian, terdapat penelitian oleh (Fitri, Andreswari, & Hasibuan, 2019) yang membahas tentang sentiment analysis twitter menggunakan algoritma Naïve Bayes, Decision Tree, dan Random Forest yang memberikan masing-masing akurasi 83.43%, 82.91%, dan 82.91%. Terdapat juga penelitian oleh (Jianqiang, Xiaolin, & Xuejun, 2018) membahas tentang sentiment analysis menggunakan algoritma Deep Convolutional Neural Network yang mendapatkan akurasi terbaik sebesar 87.62%. Penelitian yang dilakukan oleh (Huq, Ali, & Rahman, 2017) membahas tentang sentiment analysis menggunakan KNN dan support vector machine dengan menggunakan 5-fold cross validation mendapatkan akurasi terbaik 84,32% dan 77,97%.

Untuk mempermudah dalam melihat dan membandingkan penelitian yang sudah diuraikan di atas, dapat melihat ke table 2.1. Dari referensi di atas, dapat dilihat bahwa sentiment analysis untuk Twitter menggunakan berbagai macam algoritma mendapatkan akurasi paling minimum adalah 82,91% dan hal ini bisa dikatakan sudah cukup baik.

|  |  |  |  |
| --- | --- | --- | --- |
| Pengarang dan Tahun | Judul | Metode | Keterangan |
| Munir Ahmad, Shabib Aftab, Iftikhar Ali (2017) | Sentiment Analysis of Tweets using SVM | Support Vector Machine (SVM) | Mendapatkan akurasi sebesar 85% |
| Muhammad Ali Fauzi (2018) | Random Forest Approach for Sentiment Analysis in Indonesia Language | Random Forest | Melakukan nilai OOB/akurasi sebesar 82,9% |
| Veny Amilia Fitri, Rachmadita Andreswari, Muhammad Azani Hasibuan (2019) | Sentiment Analysis of Social Media Twitter with Case of Anti-LGBT Campaign in Indonesia using Naïve Bayes, Decision Tree, and Random Forest Algorithm | Naïve Bayes | Mendapatkan akurasi sebesar 83,43% |
| Decision Tree | Mendapatkan akurasi sebesar 82,91% |
| Random Forest | Mendapatkan akurasi sebesar 82,91% |
| Zhao Jianqiang, Gui Xiaolin, Zhang Xuejun (2018) | Deep Convolution Neural Networks for Twitter Sentiment Analysis | Deep Learning CNN (Convolutional Neural Network) | Mendapatkan akurasi sebesar 87,62% |
| Mohammad Rezwanul Huq, Ahmad Ali, Anika Rahman (2017) | Sentiment Analysis on Twitter Data using KNN and SVM | KNN Classifier | Mendapatkan akurasi terbaik 84,32% dengan 5-fold cross validation |
| Support Vector Machine (SVM) | Mendapatkan akurasi terbaik 77,97% dengan 5-fold cross validation |

Menurut (Singhal & Bhattacharyya, 2016) deep learning mempunyai potensi untuk mengatasi kekurangan pada machine learning, khususnya support vector machine dalam melakukan sentiment analysis dengan memberikan beberapa potensi keuntungan diantaranya adalah kemampuan untuk beradaptasi dengan variasi tugas dengan perubahan yang sangat kecil dalam sistem itu sendiri , memungkinkan pembelajaran representasi yang baik dan tidak memerlukan *feature extraction* tetapi menggunakan *words embedding* sebagai input yang menyimpan informasi.

Berangkat dari hal ini, saya ingin melakukan penelitian sentiment analysis ujaran kebencian Twitter di Indonesia menggunakan *deep learning* dengan Teknik LSTM dan Bidirectional LSTM.

## **Landasan Teori**

* + 1. **Sentiment Analysis**

Sentiment analysis adalah studi berbasis komputer tentang pendapat, keyakinan, dan emosi orang tentang entitas tertentu. Entitas memiliki kemampuan untuk mempengaruhi orang, gerakan, atau objek (Medhat, Hassan, & Korashy, 2014) . Analisis sentimen dengan kata lain digunakan untuk mengetahui keadaan suatu data, baik yang mengandung sentimen positif, netral, maupun negatif. Dimungkinkan juga untuk menggunakan sentiment analysis sebagai klasifikasi. Sentiment analysis bekerja menggunakan parameter berikut:

Diagram

Description automatically generated

Gambar Alur Kerja Sentiment Analysis (Medhat, Hassan, & Korashy, 2014)

* + 1. **Jaringan Syaraf Tiruan (Artificial Neural Network)**

Jaringan saraf tiruan terinspirasi oleh model awal pemrosesan sensorik oleh otak. Jaringan saraf tiruan dapat dibuat dengan mensimulasikan jaringan memodelkan neuron di komputer. Dengan menerapkan algoritma yang meniru proses nyata neuron, dapat membuat jaringan 'belajar' untuk memecahkan banyak jenis masalah (A)

Jaringan saraf tiruan (JST) merupakan jaringan yang terdiri dari unit pemrosesan kecil yang dimodelkan setelah sistem saraf manusia (Wikipedia). JST adalah sistem adaptif yang dapat mengubah strukturnya untuk memecahkan masalah berdasarkan informasi eksternal dan internal yang mengalir melalui jaringan. Karena sifatnya yang adaptif, JST sering disebut sebagai jaringan adaptif. Sederhananya, JST adalah alat pemodelan data statistik non-linear. JST dapat digunakan untuk memodelkan hubungan yang kompleks antara input dan output untuk menemukan pola dalam data. Menurut teorema yang disebut "teorema estimasi universal", JST dengan setidaknya satu lapisan tersembunyi dengan fungsi aktivasi nonlinier dapat memodelkan setiap fungsi terukur boreal dari satu dimensi ke dimensi lain.

Model pada JST pada dasarnya merupakan fungsi model matematika yang mendefinisikan fungsi {\displaystyle f:X\rightarrow Y}dari X ke Y. Istilah "jaringan" pada JST merujuk pada interkoneksi dari beberapa *neuron* yang diletakkan pada lapisan yang berbeda. Secara umum, lapisan pada JST dibagi menjadi tiga bagian:

* Lapis masukan (*input layer*) terdiri dari *neuron* yang menerima data masukan dari variabel X. Semua *neuron* pada lapis ini dapat terhubung ke *neuron* pada lapisan tersembunyi atau langsung ke lapisan luaran jika jaringan tidak menggunakan lapisan tersembunyi.
* Lapisan tersembunyi (*hidden layer*) terdiri dari *neuron* yang menerima data dari lapisan masukan.
* Lapisan luaran (*output layer*) terdiri dari *neuron* yang menerima data dari lapisan tersembunyi atau langsung dari lapisan masukan yang nilai luarannya melambangkan hasil kalkulasi dari X menjadi nilai Y.
  + 1. **Deep Learning**

Menurut (LeCun, Bengio, & Hinton, 2015), deep learning memungkinkan pemodelan komputasi yang memungkinkan beberapa lapisan/layer pemrosesan untuk mempelajari representasi data dengan berbagai tingkat abstraksi. Metode ini dapat menemukan struktur rumit dalam kumpulan data besar dengan menggunakan algoritma backpropagation untuk menunjukkan bagaimana mesin harus mengubah parameter internalnya yang digunakan untuk menghitung representasi di setiap lapisan dari representasi di lapisan sebelumnya.

Selai itu, menurut (MathWorks, 2022), deep learning adalah cara menggunakan komputer untuk melakukan tugas-tugas yang biasanya dilakukan oleh manusia atau dapat digambarkan seperti yang dilakukan dengan metode tertentu. Model komputer digunakan dalam deep learning untuk melakukan tes klasifikasi video, audio atau teks. Tujuan model pembelajaran adalah yang terbaru, meskipun mengurangi aktivitas manusia. Model ini didasarkan pada sejumlah besar data berlabel dan struktur jaringan multilayer. Istilah "deep" biasanya digunakan untuk sisa jaringan yang bersangkutan. Jaringan syaraf tradisional terdiri dari dua sampai tiga lapisan, sedangkan jaringan syaraf dengan 150 lapisan dapat ditemukan di deep learning. Di sisi lain, model deep learning menggunakan sejumlah besar data untuk menamainya dan juga menggunakan analisis jaringan saraf yang berasal dari data untuk membuat fungsi yang berbeda. Berikut adalah contoh deep learning dalam jaringan syaraf tiruan:Diagram

Description automatically generated

Gambar contoh jaringan syaraf tiruan pada *deep learning* (MathWorks, 2022)

Di dalam *deep learning* sendiri, terdapat beberapa komponen pendukung seperti optimizer, activation function, regularization dan loss function.

* + - * 1. **Optimizer**

Optimizer adalah algoritma atau metode yang digunakan untuk mengubah atribut jaringan saraf, seperti bobot tingkat pembelajaran, untuk mengurangi loss (Musstafa, 2021) . Algoritma atau strategi optimizer bertanggung jawab untuk mengurangi loss dan mencapai hasil yang paling akurat. Berikut ini adalah beberapa jenis pengoptimalan yang paling umum digunakan dalam deep learning (Musstafa, 2021):

* 1. RMSProp

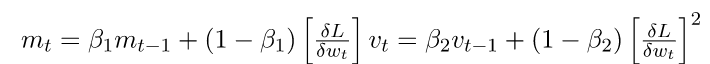
RMSprop menggunakan sinyal gradien yang menyesuaikan ukuran langkah untuk setiap bobot secara terpisah. Dalam algoritma ini, dua gradien pertama dibandingkan untuk tanda. Jika mereka memiliki tanda yang sama, pergi ke arah yang benar dan karena itu tingkatkan ukuran langkah dengan sebagian kecil. Di sisi lain, jika tandanya dibalik, maka harus mengurangi ukuran langkah dan kemudian membatasi ukuran langkah dan melakukan pembaruan bobot. Masalah dengan RMSProp adalah tidak bekerja dengan baik dengan kumpulan data besar atau selama pembaruan mini-lokal. Propagator RMS juga dapat dianggap sebagai peningkatan dari pengoptimal AdaGrad karena meminimalkan tingkat pembelajaran yang menurun secara monoton. Algoritma ini terutama berfokus pada mempercepat proses optimasi dengan mengurangi jumlah evaluasi fungsi untuk mencapai minimum lokal. Algoritme mengkuadratkan rata-rata pergerakan gradien untuk setiap bobot dan membagi gradien dengan akar kuadrat dari kuadrat rata-rata.RMS Prop(Root Mean Square) Deep Learning Optimizer

di mana gamma adalah faktor pelupa. Bobot diperbarui dengan rumus di bawah ini

RMS Prop(Root Mean Square) Deep Learning Optimizer

Sederhananya, jika ada parameter yang menyebabkan fungsi cost banyak berosilasi, itu akan menghukum memperbarui parameter itu. Algoritma ini memiliki beberapa keunggulan dibandingkan versi sebelumnya dari algoritma gradient descent. Algoritme konvergen dengan cepat dan membutuhkan lebih sedikit penyetelan daripada algoritme penurunan gradien dan variannya. Masalah dengan RMS Prop adalah kecepatan pembelajaran harus diatur secara manual dan nilai yang disarankan tidak berfungsi untuk setiap aplikasi.

* 1. Adam

Algoritma optimizer ini merupakan perpanjangan lebih lanjut dari penurunan gradien stokastik untuk memperbarui bobot jaringan selama pelatihan. Pengoptimal Adam memperbarui pembelajaran setiap bobot jaringan secara terpisah. Alih-alih menyesuaikan learning rate berdasarkan momen pertama (mean) seperti RMS Prop, Adamis juga menggunakan momen gradien kedua. Algoritme ini mudah diimplementasikan, memiliki runtime yang lebih cepat, kebutuhan memori yang lebih rendah, dan memerlukan konfigurasi yang lebih sedikit daripada algoritme pengoptimalan lainnya.

Rumus di atas merupakan cara kerja pengoptimal adam. Di sini B1 dan B2 mewakili laju peluruhan rata-rata gradien.

Kelemahan dari optimizer ini biasanya berfokus pada waktu komputasi yang lebih cepat, sedangkan algoritma seperti penurunan gradien stokastik fokus pada titik data. Oleh karena itu, algoritme seperti SGD menggeneralisasi data dengan lebih baik dengan mengorbankan kecepatan komputer yang rendah. Oleh karena itu, algoritma optimasi dapat dipilih sesuai dengan kebutuhan dan tipe data.

* + - * 1. **Activation Function**

(Marimuthu, 2022) berhipotesis bahwa activation function bertanggung jawab atas apakah neuron dalam jaringan saraf deep learning harus diaktifkan atau tidak. Artinya, ia menggunakan beberapa operasi matematika sederhana untuk menentukan apakah neuron input jaringan relevan atau tidak relevan dengan proses prediksi. Tujuan dari fungsi aktivasi adalah kemampuan untuk memasukkan nonlinier ke dalam jaringan saraf dan menghasilkan output dari sekumpulan nilai input yang dimasukkan ke dalam lapisan.

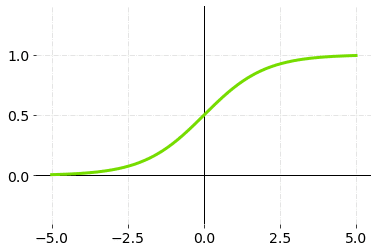
Berikut adalah contoh fungsi aktivasi (Marimuthu, 2022):

1. Sigmoid

Activation function sigmoid menerima nilai input dan mengembalikan 0 atau 1. Sigmoid sering digunakan untuk pemrosesan klasifikasi biner. Sigmoid memberikan probabilitas bahwa kelas tertentu ada. Ketika sigmoid diwakili dalam matematika, persamaan itu adalah:Diagram, schematic

Description automatically generated

Gambar persamaan activation function sigmoid (Marimuthu, 2022)



Gambar grafik activation function sigmoid (Marimuthu, 2022)

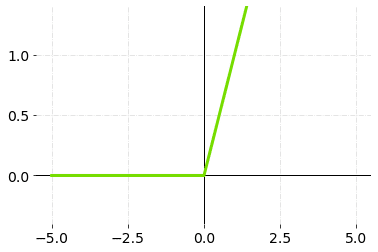
1. ReLu (Rectified Linear Unit)

ReLU adalah singkatan dari Rectified Linear Unit dan merupakan salah satu activation function yang paling umum digunakan dalam aplikasi. Re-Lu memecahkan masalah gradien yang hilang karena fungsi re-lu memiliki gradien maksimum 1. Ini juga memecahkan masalah saturasi neuron karena kemiringan fungsi ReLU tidak pernah nol. ReLU berkisar dari 0 hingga tak terhingga. Ketika direpresentasikan dalam matematika, Re-Lu memiliki persamaan ini:

Text

Description automatically generated

Gambar persamaan activation function ReLu



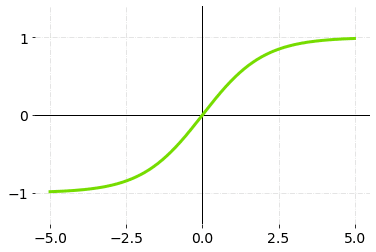
Gambar grafik activation function ReLu

1. Softmax

Kombinasi dari banyak sigmoid disebut fungsi Softmax. Ini menentukan probabilitas relatif. Serupa dengan activation function sigmoid, fungsi Softmax mengembalikan probabilitas setiap kelas/label. Dalam klasifikasi multi-kelas, activation function softmax paling sering digunakan untuk lapisan terakhir dari jaringan saraf. Fungsi softmax memberikan probabilitas kelas saat ini relatif terhadap yang lain. Ini berarti bahwa dia juga mempertimbangkan kemungkinan kelas lain. Ketika direpresentasikan dalam matematika, softmax adalah persamaan seperti ini:Text

Description automatically generated

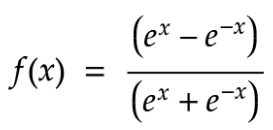
Gambar persamaan activation function softmax



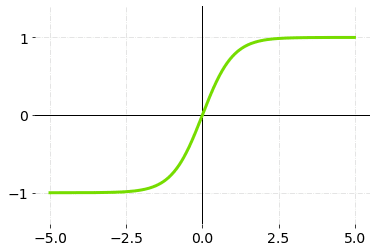
Gambar grafik activation function softmax

1. Tan H

Tan H merupakan activation function non-linear yang outputnya berpusat pada nol. Tan H mempunyai keuntungan dimana input negatif akan dipetakan dengan kuat ke input negative dan nol akan dipetakan kuat ke “hampir nol”. Kelebihan dari tan h adalah gradien yang dimiliki lebih kuat dibandingkan dengan sigmoid, atau berarti turunan dari grafik tan h lebih curam. Ketika direpresentasikan ke dalam persamaan, maka akan didapatkan persamaan sebagai berikut:



Gambar persamaan Tan H activation function



Gambar grafik activation function tan H

* + - * 1. **Regularization**

Regularisasi adalah serangkaian teknik yang mengurangi kompleksitas model jaringan saraf selama pelatihan untuk menghindari overfitting. Salah satu metode legalisasi yang umum digunakan adalah Dropout.

1. Dropout

Istilah "dropout" mengacu pada putusnya node (input dan lapisan tersembunyi) dalam jaringan saraf. Semua koneksi maju dan mundur dengan node yang dijatuhkan dihapus sementara, sehingga menciptakan arsitektur jaringan baru dari jaringan induk (Yadav).

Dropout berarti selama proses training dengan beberapa probabilitas P neuron dari jaringan syaraf dimatikan.

A picture containing watch

Description automatically generated

Gambar contoh implementasi dari Dropout

Contoh dapat diambil dari gambar di atas, di mana di sebelah kiri adalah jaringan saraf feedforward tanpa Dropout. Menggunakan penurunan dengan probabilitas P = 0,5 selama proses pelatihan, secara acak mematikan neuron dan membentuk jaringan saraf seperti yang ditunjukkan di sebelah kanan. Ini berarti bahwa setengah dari neuron mati dan tidak dihitung, yang membuat jaringan saraf lebih sederhana dan tidak terlalu kompleks, yang mengurangi terjadinya over-tuning. Penonaktifan neuron dengan probabilitas tertentu P diterapkan pada setiap langkah maju dan pembaruan bobot.

* + - * 1. **Loss Function**

Loss function adalah fungsi yang membandingkan target dan nilai keluaran yang diprediksi; mengukur seberapa baik jaringan saraf memodelkan data pelatihan. Selama pelatihan, tujuannya adalah untuk meminimalkan loss antara kekuatan yang diprediksi dan target (Yathish, 2022). Ada dua jenis loss function dalam jaringan syaraf tiruan, yaitu fungsi regresi dan fungsi kerugian klasifikasi (Yathish, 2022)*.*

1. Regression Loss Function

Saat digunakan dalam jaringan saraf regresi, model memprediksi nilai keluaran yang benar dengan nilai masukan (bukan judul yang dipilih sebelumnya, misalnya, mean squared error (MSE) dan mean absolute error (MAE)).

1. Mean Squared Error (MSE)

Text

Description automatically generated

Fungsi ini memiliki banyak properti yang membuatnya sangat cocok untuk menghitung loss. Selisihnya kuadrat, artinya tidak masalah apakah nilai prediksi lebih besar atau lebih kecil dari nilai target; Namun, skor dengan kesalahan besar dihukum. MSE juga merupakan fungsi cembung dengan minimum global yang terdefinisi dengan baik. Ini memudahkan penggunaan pengoptimal penurunan gradien untuk menentukan nilai bobot. Namun, salah satu kelemahan dari fungsi kerugian ini adalah sangat sensitif terhadap penyimpangan, artinya jika nilai prediksi secara signifikan lebih tinggi atau lebih rendah dari nilai target, maka akan meningkatkan kerugian secara signifikan.

1. Mean Absolute Error (MAE)

Text

Description automatically generated with medium confidence

Loss function ini digunakan dalam beberapa kasus sebagai alternatif untuk MSE. MAE digunakan dalam kasus di mana ada banyak outlier dalam data pelatihan. Fungsi kerugian ini juga memiliki beberapa kelemahan; karena jarak rata-rata mendekati nol, optimasi penurunan gradien tidak akan bekerja karena turunan dari fungsi pada 0 tidak terdefinisi (yang menyebabkan kesalahan karena tidak mungkin untuk membagi dengan 0).

1. Classification Loss Function

Digunakan untuk klasifikasi dalam jaringan saraf, di mana jaringan saraf menghasilkan vektor probabilitas input yang dimiliki oleh beberapa kelas yang telah ditentukan, dan kemudian dapat memilih kelas dengan probabilitas kepemilikan tertinggi; Misalnya, entropi silang biner dan entropi silang kategoris dari fungsi kerugian.

1. Binary Crossentropy

Jaringan saraf klasifikasi menghasilkan vektor probabilitas dari kemungkinan bahwa input yang diberikan cocok dengan setiap kelas yang telah ditentukan, dan kemudian memilih kelas yang paling mungkin sebagai hasilnya. Dalam klasifikasi biner, hanya ada dua nilai yang mungkin, nilai aktual y adalah 0 atau 1. Untuk menentukan secara akurat kerugian antara nilai aktual dan prediksi, perlu membandingkan nilai aktual (0 atau 1) dengan probabilitas bahwa input termasuk dalam kategori itu (p(i ) ) = probabilitas, kelas itu adalah 1; 1 — p(i) = probabilitas kelas itu Text

Description automatically generated with low confidence

Gambar persamaan loss function binary crossentropy

1. Categorical crossentropy

Dalam kasus di mana jumlah kategori lebih dari dua, menggunakan categorical crossentropy. Categorical crossentropy mengikuti proses yang sangat mirip dengan binary crossentropy, satu-satunya perbedaan adalah jumlah kategori yang tersedia.

Diagram, text

Description automatically generated

Gambar persamaan loss function categorical crossentropy

* + - * 1. **Neural Network Layer**

Neural network memiliki banyak jenis dan macam layer. Penggunaan jenisnya tergantung dari model yang akan dirancang dan menyesuaikan dari kasus yang ada. Berikut beberapa contoh layer yang sering digunakan dalam penyelesaian kasus text di dalam deep learning.

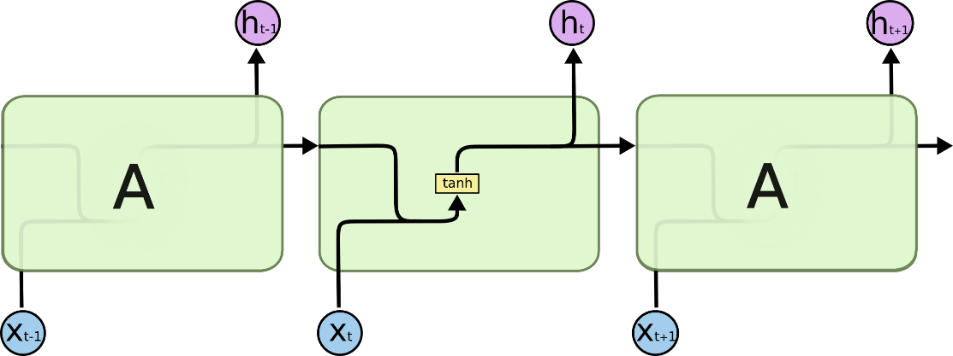
1. Dense Layer

Dense layer merupakan layer yang simple dimana setiap neuron menerima input dari layer sebelumnya ataupun bekerja sebagai input layer (Dumane).

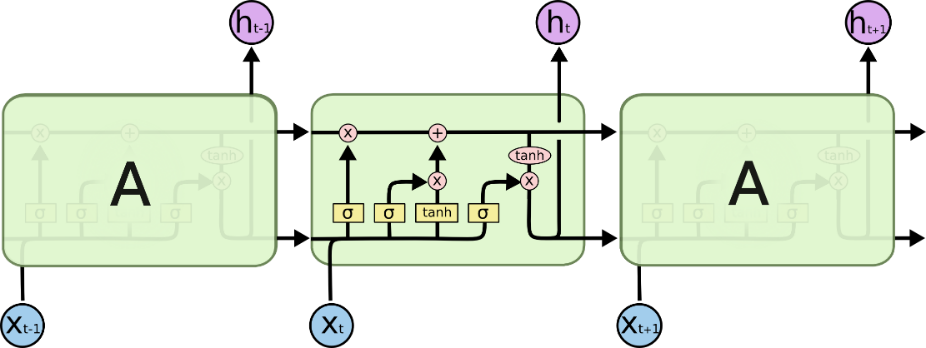
1. Word Embedding

Word embedding layer merupakan layer dalam deep learning yang merepresentasikan kosa kata dalam sebuah dokumen yang paling popular yang direpresentasikan dalam sebuah vector (Karani, 2018). Word embedding dapat menangkap konteks, kesamaan semantic dan sintaksis dalam suatu dokumen. Word embedding akan mengelompokan

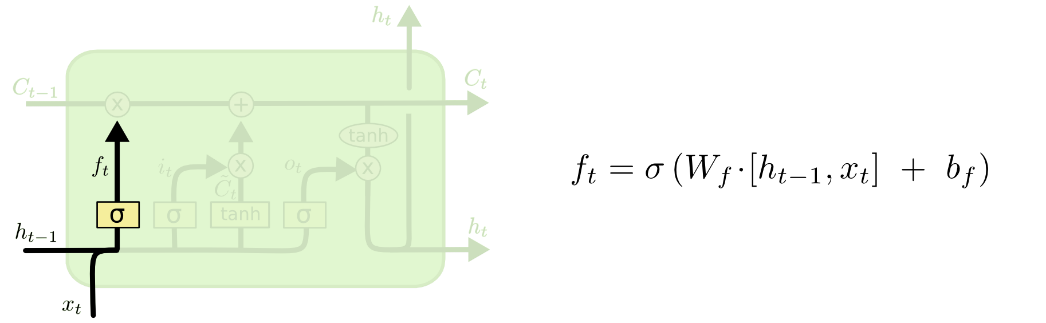
1. Flatten Layer
   * 1. **Long Short-Term Memory (LSTM)**

Long Short-Term Memory atau biasa dikenal dengan LSTM merupakan sebuah RNN (jaringan syaraf berulang) yang dapat mengatasi permasalahan ketergantungan jangka Panjang (Yu, Si, Hu, & Zhang, 2019) . LSTM meningkatkan kapasitas mengingat sel rekuren standar (RNN) dengan mengenalkan “gate” atau gerbang ke dalam cell state. LSTM dapat mengingat informasi untuk waktu yang lama (Colah, 2015). Semua jenis RNN mengambil bentuk rantai modul jaringan saraf berulang.

LSTM juga memiliki struktur seperti rantai ini, tetapi modul yang diulang memiliki struktur yang berbeda. Alih-alih satu lapisan jaringan saraf, ada empat yang berkomunikasi dengan cara yang sangat spesifik.



Kunci LSTM adalah cell state, garis horizontal yang melintasi bagian atas grafik. Keadaan sel itu bergerak langsung di sepanjang rantai dengan hanya beberapa interaksi linier kecil. Sangat mudah untuk informasi mengalir melalui tidak berubah. LSTM dapat menghapus atau menambahkan data ke keadaan sel yang dikendalikan oleh struktur yang disebut Gate. Gate adalah cara opsional untuk mentransfer informasi. Mereka terdiri dari lapisan jaringan saraf sigmoid dan operasi perkalian titik. Alur kerja dari LSTM sendiri akan berjalan seperti berikut:



Gambar cell state melalui forget gate

Langkah pertama adalah menentukan informasi apa yang akan dibuang dari cell state yang dilakukan oleh forget gate.

Diagram

Description automatically generated

Gambar dari output gerbang sigmoid

Menentukan informasi baru yang akan disimpan dalam cell state.

Diagram

Description automatically generated with medium confidence

Gambar mengupdate cell state lama

Melakukan pembaharuan informasi terhadap cell state lama dengan cell state yang baru, atau bisa dikatakan melakukan kolaborasi informasi antara informasi di cell state lama dengan informasi yang ada di cell state yang terbaru

A picture containing text, clock

Description automatically generated

Gambar menentukan output terbaru dari cell state.

* + 1. **Bidirectional LSTM**

Bidirectional LSTM atau LSTM dua arah adalah perpanjangan dari LSTM tradisional yang dapat meningkatkan kinerja model untuk masalah klasifikasi urutan. Untuk masalah di mana semua langkah waktu dari urutan input tersedia, LSTM dua arah melatih 2 urutan input. Yang pertama dalam urutan input apa adanya dan yang kedua dalam salinan terbalik dari urutan input. Ini dapat memberikan konteks tambahan ke jaringan dan mengarah pada pembelajaran masalah yang lebih cepat dan lebih lengkap (Mungalpara, 2021).

Diagram

Description automatically generated

Gambar Bidirectional LSTM (Mungalpara, 2021)

# References

A, K. (n.d.). What are artificial neural network? *2008*. Retrieved from https://www.nature.com/articles/nbt1386

Ahmad, M., Ali, I., & Ahtab, S. (2017). *Sentiment Analysis of Tweets using SVM*. Retrieved from https://www.researchgate.net/profile/Shabib-Aftab-2/publication/321084834\_Sentiment\_Analysis\_of\_Tweets\_using\_SVM/links/5a1497b90f7e9b925cd514b0/Sentiment-Analysis-of-Tweets-using-SVM.pdf

Colah. (2015). Understanding LSTM Networks. Retrieved from https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Dumane, G. (n.d.). Introduction to Convolutional Neural Network (CNN) using Tensorflow. *2020*. Retrieved from https://towardsdatascience.com/introduction-to-convolutional-neural-network-cnn-de73f69c5b83#:~:text=Dense%20Layer%20is%20simple%20layer,multiple%20number%20of%20such%20neurons.

Fauzi, M. A. (2018). *Random Forest Approach for Sentiment Analysis in Indonesian Language*. Retrieved from https://www.researchgate.net/profile/Muhammad-Fauzi-6/publication/327060733\_Random\_Forest\_Approach\_for\_Sentiment\_Analysis\_in\_Indonesian\_Language/links/5d305ff3458515c11c39adfd/Random-Forest-Approach-for-Sentiment-Analysis-in-Indonesian-Language.pdf

Fitri, V. A., Andreswari, R., & Hasibuan, M. A. (2019). *Sentiment Analysis of Social Media Twitter with Case of Anti-LGBT Campaign in Indonesia using Naïve Bayes, Decision Tree, and Random Forest Algorithm*. Retrieved from https://www.researchgate.net/profile/Muhammad-Fauzi-6/publication/327060733\_Random\_Forest\_Approach\_for\_Sentiment\_Analysis\_in\_Indonesian\_Language/links/5d305ff3458515c11c39adfd/Random-Forest-Approach-for-Sentiment-Analysis-in-Indonesian-Language.pdf

Huq, M. R., Ali, A., & Rahman, A. (2017). Sentiment Analysis on Twitter Data using KNN and SVM. Retrieved from https://pdfs.semanticscholar.org/05a8/78000170abcd0c6f8208080470858422e17c.pdf

Jianqiang, Z., Xiaolin, G., & Xuejun, Z. (2018). *Deep Convolution Neural Networks for Twitter Sentiment Analysi*. Retrieved from https://ieeexplore.ieee.org/abstract/document/8244338

Karani, D. (2018). Introduction to Word Embedding and Word2Vec. Retrieved from https://towardsdatascience.com/introduction-to-word-embedding-and-word2vec-652d0c2060fa

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. Retrieved from https://www.nature.com/articles/nature14539

Marimuthu, P. (2022). How Activation Functions Work in Deep Learning. Retrieved from https://www.kdnuggets.com/2022/06/activation-functions-work-deep-learning.html

MathWorks. (2022). What Is Deep Learning? Retrieved from https://www.mathworks.com/discovery/deep-learning.html#:~:text=Deep%20learning%20is%20a%20machine,a%20pedestrian%20from%20a%20lamppost

Medhat, W., Hassan, A., & Korashy, H. (2014). *Sentiment analysis algorithms and applications: A survey*. Retrieved from https://www.sciencedirect.com/science/article/pii/S2090447914000550

Mungalpara, J. (2021). What does it mean by Bidirectional LSTM? Retrieved from https://medium.com/analytics-vidhya/what-does-it-mean-by-bidirectional-lstm-63d6838e34d9

Musstafa. (2021). Optimizers in Deep Learning. Retrieved from https://medium.com/mlearning-ai/optimizers-in-deep-learning-7bf81fed78a0

Singhal, P., & Bhattacharyya, P. (2016). Sentiment Analysis and Deep Learning: A Survey. Retrieved from https://www.cfilt.iitb.ac.in/resources/surveys/sentiment-deeplearning-2016-prerna.pdf

Wikipedia. (n.d.). Jaringan Syaraf Tiruan. *2022*. Retrieved from https://id.wikipedia.org/wiki/Jaringan\_saraf\_tiruan

Yadav, H. (n.d.). Dropout in Neural Networks. *2022*. Retrieved from https://towardsdatascience.com/dropout-in-neural-networks-47a162d621d9

Yathish, V. (2022). Loss Functions and Their Use In Neural Networks. Retrieved from https://towardsdatascience.com/loss-functions-and-their-use-in-neural-networks-a470e703f1e9

Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. Retrieved from https://direct.mit.edu/neco/article-abstract/31/7/1235/8500/A-Review-of-Recurrent-Neural-Networks-LSTM-Cells