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

**BAB II**

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.

2. 1. **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%. Terdapat penelitian yang dilakukan oleh (Liao, Wang, Yu, Sato, & Cheng, 2016) yang membahas tentang sentiment analysis untuk klasifikasi latar situasi pada data Twitter yang membuahkan akurasi pengembangan sebesar 74.5%. Selanjutnya, (Abdelgwad, Soliman, Taloba, & Farghaly, 2021) melakukan penelitian tentang sentiment analysis berbasis aspek Bahasa Arab menggunakan model Bidirectional GRU yang mendapatkan akurasi optimal sebesar 83.98%

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 |
| Mohammed M.Abdelgwad, Taysir Hassan A Soliman, Ahmed I.Taloba,  Mohamed Fawzy Farghaly  (2021) | Arabic Aspect Based Sentiment Analysis Using Bidirectional GRU Based Models | Bidirectional GRU | Mendapatkan akurasi optimal sebesar 83,98% |
| 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% |
| Muhammad Ali Fauzi (2018) | Random Forest Approach for Sentiment Analysis in Indonesia Language | Random Forest | Melakukan nilai OOB/akurasi sebesar 82,9% |
| 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 |
| Munir Ahmad, Shabib Aftab, Iftikhar Ali (2017) | Sentiment Analysis of Tweets using SVM | Support Vector Machine (SVM) | Mendapatkan akurasi sebesar 85% |
| Shiyang Liaoa, Junbo Wang, Ruiyun Yua  ,Koichi Satob  ,Zixue Cheng | CNN for situations understanding based on  sentiment analysis of twitter data | CNN | Mendapatkan akurasi tahap pengembangan sebesar 74,5% |



Tabel 2.1 Tinjauan Pustaka Penelitian.

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.

* 1. **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 2.1 Alur Kerja Sentiment Analysis (Medhat, Hassan, & Korashy, 2014) Jaringan Syaraf Tiruan(Artificial Neural Network)

* + 1. **Jaringan Syaraf Tiruan**

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 seperti sistem saraf manusia. 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. **Dense Layer**

Dense layer merupakan layer yang simple dimana setiap neuron menerima input dari layer sebelumnya ataupun bekerja sebagai input layer (Dumane). Layer ini akan melakukan operasi perkalian vector. Hasil dari setiap neuron dari lapisan sebelumnya menuju ke setiap neuron tunggal dari lapisan padat. Dapat dikatakan bahwa jika lapisan sebelumnya mengeluarkan matriks (M x N) dengan menggabungkan hasil dari setiap neuron, keluaran ini melewati lapisan padat di mana jumlah neuron pada lapisan padat harus N.

Diagram

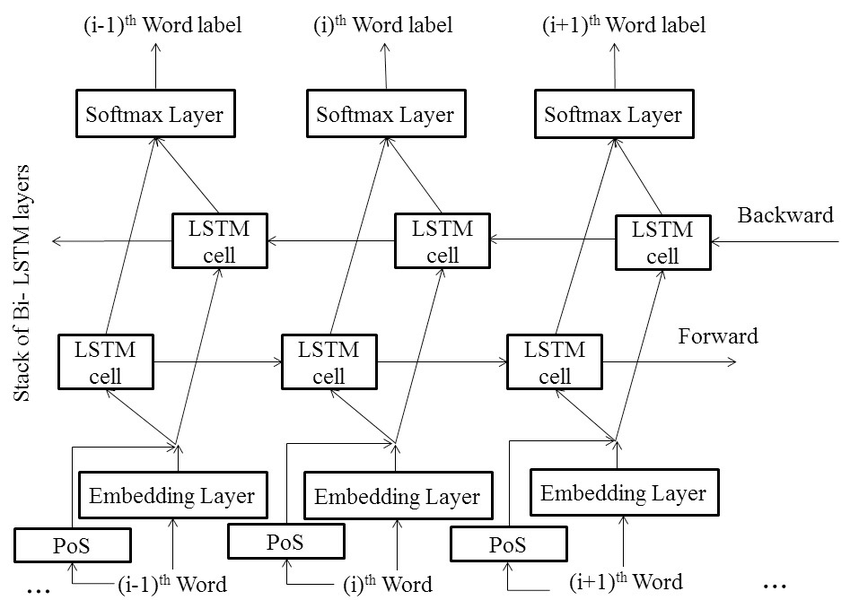
Description automatically generated

Gambar dense layer (Verma, 2021)

Gambar 2.2. Dense layer

* + - 1. **Embedding Layer**

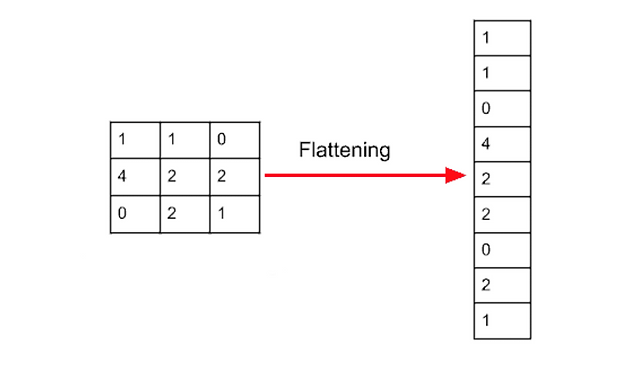
Embedding layer merupakan layer dalam deep learning yang merepresentasikan kosa kata dalam sebuah dokumen yang paling popular yang direpresentasikan dalam sebuah vector (Karani, 2018). Setelah itu, word embedding dapat menangkap konteks, kesamaan semantic dan sintaksis dalam suatu dokumen.



Gambar 2.3 Embedding layer di dalam model BiLSTM (ReaserchGate)

* + - 1. **Flatten Layer**

Flatten layer berfungsi untuk melakukan transformasi neurons dari multi dimensi menjadi dimensi tunggal. Dalam penelitian ini, flatten layer digunakan untuk melakukan transformasi output dari layer Bidirectional/Embedding layer yang tergolong multi dimensi (3 dimensi) menjadi dimensi tunggal agar bisa diterima oleh hidden layer dense.



Gambar 2.4. Flatten layer

* + 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.

Selain 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 2.5 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.

* + - * 1. **RMSProp**

RMSprop menggunakan sinyal gradien yang menyesuaikan ukuran langkah untuk setiap bobot secara terpisah (Musstafa, 2021). 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

Gambar 2.6. persamaan optimizer RMSProp (Musstafa, 2021)

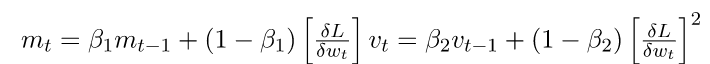
di mana gamma adalah faktor pelupa. Bobot diperbarui dengan rumus di bawah ini

RMS Prop(Root Mean Square) Deep Learning Optimizer

Gambar 2.7 persamaan optimizer RMSProp (Musstafa, 2021)

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 (Musstafa, 2021). 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.

Gambar 2.8 persamaan optimizer adam (Musstafa, 2021)

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) mengatakan 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.

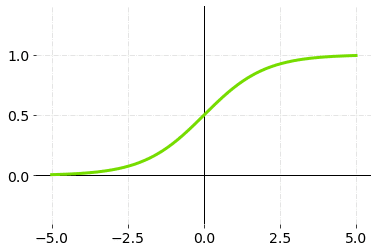
* + - * 1. **Sigmoid**

Activation function sigmoid menerima nilai input dan mengembalikan 0 atau 1 (Marimuthu, 2022). 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 2.9 persamaan activation function sigmoid (Marimuthu, 2022)



Gambar 2.10 grafik activation function sigmoid (Marimuthu, 2022)

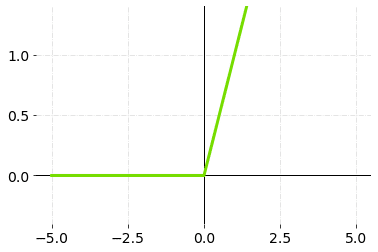
* + - * 1. **Relu**

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 (Marimuthu, 2022). 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 2.11 persamaan activation function ReLu (Marimuthu, 2022)



Gambar 2.12 grafik activation function ReLu (Marimuthu, 2022)

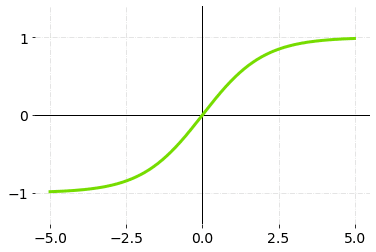
* + - * 1. **Softmax**

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 (Marimuthu, 2022). 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

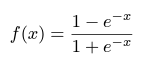
Gambar 2.13 persamaan activation function softmax (Marimuthu, 2022)



Gambar 2.13. grafik activation function softmax (Marimuthu, 2022)

* + - * 1. **TanH**

Fungsi hiperbolik tangen atau tanh cocok digunakan untuk jaringan syaraf tiruan multi-layer (Datta, 2020). Fungsi tanh didefinisikan sebagai:



Gambar 2.14. persamaan activation function tanh (Datta, 2020)

Activation function tanh merupakan modifikasi dari activation function sigmoid karena persamaan tanh didefinisikan sebagai berikut

Gambar 2.15. persamaan activation function tanh

Tanh merupakan fungsi kontinu dimana outputnya berkisar dari negatif satu sampai dengan satu. Dari hal itu, outputnya bisa dikatakan negatif, nol, ataupun positif. Oleh karena itu, fungsi ini terpusat nol dan menyelesaikan masalah 'fungsi aktivasi bukan berpusat nol' dari fungsi sigmoid.

* + - 1. **Regularization**

Regularisasi adalah serangkaian teknik yang mengurangi kompleksitas model jaringan saraf selama pelatihan untuk menghindari overfitting. Salah satu metode regularisasi yang umum digunakan adalah 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 2.16. 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) (Yathish, 2022).

1. Mean Squared Error (MSE)

Text

Description automatically generated

Gambar 2.17 Persamaan MSE (Yathish, 2022)

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

Gambar 2.18 Persamaan MAE (Yathish, 2022)

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 (Yathish, 2022). Sebagai contoh, 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 2.19 Persamaan loss function binary crossentropy

1. Categorical crossentropy

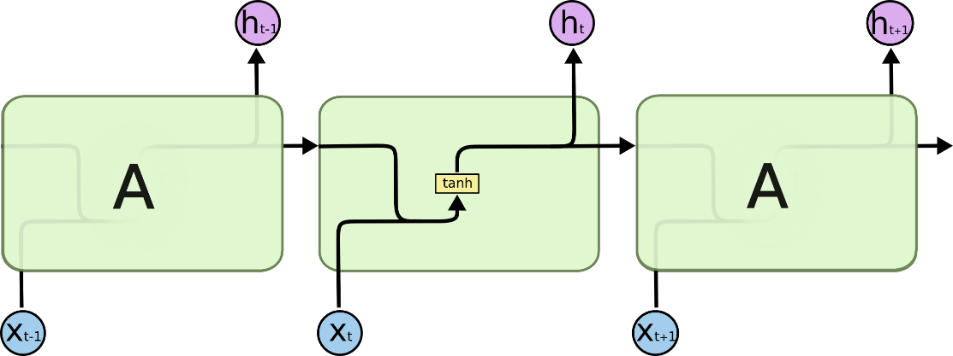
Dalam kasus di mana jumlah kategori lebih dari dua, menggunakan categorical crossentropy. Categorical crossentropy mengikuti proses yang sangat mirip dengan binary crossentropy, satu-satunya perbedaan adalah jumlah kategori yang tersedia.

Diagram, text

Description automatically generated

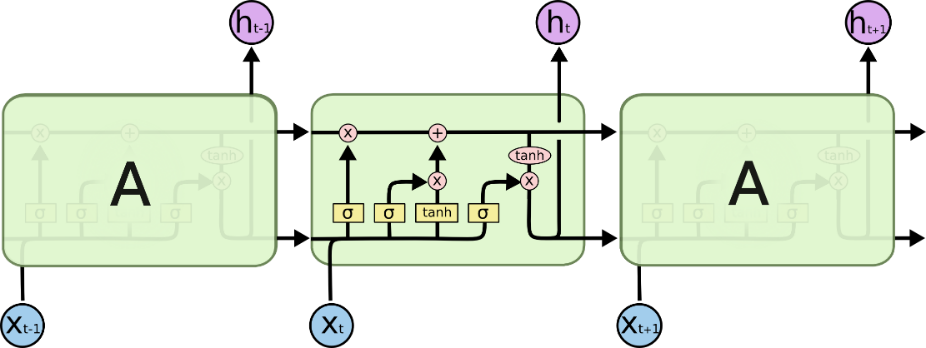
Gambar 2.20 Persamaan loss function categorical crossentropy

* + 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.

Gambar 2.21 Contoh RNN (Colah, 2015)

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.



Gambar 2.22 Struktur LSTM (Colah, 2015)

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:

Diagram

Description automatically generated

Gambar 2.23 Cell state melalui forget gate (Colah, 2015)

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

Diagram

Description automatically generated

Gambar 2.24 Output gerbang sigmoid (Colah, 2015)

Menentukan informasi baru yang akan disimpan dalam cell state.

Diagram

Description automatically generated with medium confidence

Gambar 2.25 Mengupdate cell state lama (Colah, 2015)

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 2.26 Menentukan output terbaru dari cell state. (Colah, 2015)

* + 1. **Bidirectional Long Sshort-Term Memory (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 2.27 Bidirectional LSTM (Mungalpara, 2021)

* + 1. **Preprocessing**

Preprocessing merupakan sebuah proses untuk membuat input untuk analisis yang diberikan kurang kompleks dengan cara yang tidak mempengaruhi interpretabilitas atau kesimpulan substantif dari model selanjutnya (Denny & Spirling, 2017). Preprocessing pada dasarnya bertujuan untuk melakukan transformasi data tanpa mengubah isi dari data tersebut agar data dapat lebih mudah digunakan. Preprocessing yang biasanya ada dalam melakukan klasifikasi teks ialah lowercasing, stemming, stopword removal, pembersihan karakter khusus, pembersihan karakter non-alphanumeric dan masih banyak lagi.

* + - 1. **Lowercasing**

Lowercasing merupakan salah satu tahap preprocessing yang sering diaplikasikan jika menghadapi data teks. Hal ini bertujuan untuk mengatasi masalah suatu karakter yang sebenernya sama, akan tetapi dapat terdeteksi beda oleh computer karena perbedaan huruf kapital dan kecil (misal Gajah dengan gajah) (Denny & Spirling, 2017).

* + - 1. **Stemming**

Stemming merupakan tahapan dalam teks preprocessing yang akan mereduksi sebuah kata menjadi bentuk dasar dari kata tersebut (Denny & Spirling, 2017). Stemming sendiri sering diartikan sebagai Teknik pengurangan kosakata. Stemming dapat dimisalkan dengan reduksi kata “menyapu” menjadi “sapu”.

* + - 1. **Stopword Removal**

Terdapat beberapa kata yang memiliki makna yang berarti, dan terdapat beberapa kata yang tidak memiliki kata berarti. Kata-kata yang tidak memberikan informasi/makna berarti sering disebut dengan “stop word”. Stopword removal berfungsi untuk menghilangkan karakter yang kurang memiliki makna (Denny & Spirling, 2017). Kata-kata yang kurang memiliki makna biasanya merupakan kata konjungsi, kata-kata fungsi seperti “itu”, “dan”, dan “dia” sebagai contoh.