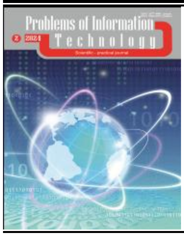


Available online at www.jpit.az

15 (2)
2024

Comparison of deep learning techniques for textual sentiment analysis

Leyla Mammadova

Institute of Information Technology, B. Vahabzade str., 9A, AZ1141 Baku, Azerbaijan

mammedova-1998@mail.ru

https://orcid.org/0009-0008-1600-9911

ARTICLE INFO

Keywords:

RNN
LSTM
GRU
Bidirectional RNN
Bidirectional LSTM
Bidirectional GRU

ABSTRACT

Examining the opinions of the people might provide us with valuable knowledge. Sentiment analysis is a method for analyzing textual data that helps find subjective information, such as opinions and feelings that individuals or groups have expressed. It improves our understanding of human language using deep learning and natural language techniques. Several deep learning models, including RNNs, LSTMs, GRUs, and their bidirectional variants, are compared in this work. Three publicly available datasets - the imdb_reviews, Twitter Sentiment Dataset, and Emotions dataset were used in the investigation. Accuracy performance is evaluated for six deep learning models. According to experimental studies, bidirectional structures outperform their unidirectional counterparts in most cases. Across several datasets, the bidirectional models continuously produced the best accuracy.

1. Introduction

The study of sentiments and views represented in textual data is the primary objective of the broad area of sentiment analysis. Sentiment analysis is the process of interpreting a text's intended context based just on its content (Alguliyev, Aliguliyev & Niftaliyeva, 2019). It involves classification and analysis, to determine if a text expresses a positive, negative, or neutral sentiment. Opinion mining is another term for sentiment analysis (Wankhade, Rao, & Kulkarni, 2022). In general, by examining vast amounts of text data, sentiment analysis assists companies and organizations in learning more about client satisfaction and society's views.

Deep learning models are frequently used and perform very effectively in tasks, including sentiment analysis (Alguliyev, Aliguliyev, & Abdullayeva, 2019). They have been trained on vast amounts of data to acquire instances of language, thus these models can identify complicated patterns and relations in textual data and can properly categorize text into many sentiment categories.

The primary goal of this study is to compare some deep learning models for sentiment analysis and assess each model's performance. In contrast to several other research works that concentrate on a single dataset or datasets that are similar in nature, our study includes three datasets that have significantly different characteristics. This study aims to analyze these models' performances, to determine the correctness of several deep learning models about their ability to accurately identify sentiment from textual input and offer valuable guidance to researchers in the field.

The remaining sections of the paper are structured as follows: The description of deep learning models is presented in Sections 3, this gives the overview of the deep learning models

Received 23 February 2024, Received in revised form 26 April 2024, Accepted 10 May 2024
http://doi.org/10.25045/jpit.v15.i2.05

used for sentiment analysis. Section 4 contains the analysis of deep learning models. Section 4 presents the results of our comparison study on deep learning models, as well as accuracy and loss graphs to analyze these models' performances. Section 5 provides the conclusion.

2. Related works

As a result the enormous volume of textual data created online, sentiment analysis has become as a key component of natural language processing (NLP). In the recent past, rule-based systems or lexicon-based techniques were used in sentiment analysis. But as large data has grown and online content has become more complicated, machine learning methods—especially deep learning models—have become a crucial tool for sentiment analysis. Recurrent neural networks, long short-term memory networks, and gated recurrent units are three popular deep learning models that are used widely because of their ability to identify context and sequential dependencies in text data. Because of their recurrent connections, which enables them to retain a memory of the previous inputs in the sequence, recurrent neural networks have proven essential in processing textual data. Unfortunately, long-term dependencies in the data are difficult to identify with RNNs because to the vanishing gradient problem (Bengio et al., 1994). Long Short-Term Memory networks (Hochreiter & Schmidhuber, 1997) were proposed in order to address these constraints. In sentiment analysis tasks, LSTMs have demonstrated impressive performance, especially for texts such as reviews or articles (Hassan & Mahmood, 2017). The LSTM architecture become simpler with Gated Recurrent Units (Cho et al., 2014) but still perform well with long sequences. The efficacy of RNNs, GRUs and LSTMs for sentiment analysis has been compared in research. For instance, Tang et al. (2015) examined four large-scale review datasets, they found that LSTMs and GRUs performed better than traditional RNNs in terms of accuracy as a result of their more intricate memory cell structure. Models like Bi-RNNs, Bi-LSTMs, and Bi-GRUs improve upon these structures by analyzing sequences in both forward and backward directions. These models perform better in tasks requiring a deep comprehension of context, such sentiment analysis, because of their ability to collect context from both past and future states within the sequence due to their bidirectional processing (Schuster & Paliwal, 1997). Studies has demonstrated that Bi-LSTMs and Bi-GRUs perform well in sentiment analysis tasks. In the case of sentiment classification within the self-driving car dataset, Pandya, and Thakkar, (2024) discovered that Bi-LSTMs and Bi-GRUs exhibit superior performance. Their research highlights the effectiveness of bidirectional models in handling the contextual dependencies present in short and noisy text data, which is characteristic of self-driving car communication.

Even with these contributions, there is a clear lack in the research when it comes to a thorough analysis of deep learning models on various diverse datasets. Many studies typically concentrate on just one kind of dataset or a limited selection of datasets, which hinders the applicability of their conclusions.

3. Description of deep learning models

Deep learning models are developed using neural network architectures, to process massive amounts of data and perform highly complex calculations that are inspired by the human brain.

In recent years, numerous deep learning models have been examined and analyzed. Typically, the term "deep" refers to the number of hidden layers present in a neural network. Deep learning models processes input through several layers, each of which generally extracts features and passes information to the following layer and may contain hundreds, thousands of hidden layers (LeCun et al., 2015).

In this section, a brief description is provided for a few of the most well-known models, such as Recurrent Neural Network (RNN), Long Short-Term Memory networks (LSTMs), Gated Recurrent Unit (GRU) networks, and their bidirectional variants.

3.1. Recurrent neural networks (RNNs)

Recurrent Neural Networks (RNNs) are a type of artificial neural network made for processing data sequences (Park et al., 2020). RNNs are named because with the results based on previous computations, they properly execute the same task for each element in the sequence. As a data text, audio, video, and other data types, including time series, are all acceptable.

The network computes a weighted sum of the current input and the previous hidden state at each time step by using shared weights to analyze input data in a single time step. By passing the weighted sum through an activation function, the current hidden state is obtained.

Recurrent Neural Networks use backpropagation through time (BPTT) during training (Ahmad et al., 2004). This involves computing gradients to update the network's parameters, and minimizing a loss function allows the model to learn from data.

$$h^{<t>} = g_1(W_h h^{<t-1>} + W_x x^{<t>} + b_h) \quad (1)$$

$$\hat{y}^{<t>} = g_2(W_y h^{<t>} + b_y) \quad (2)$$

In Eqs. (1)-(2), $x^{<t>}$ and $\hat{y}^{<t>}$ are the input and output at time step t, $h^{<t-1>}$ represent the hidden state from the previous time step, and W_x, W_h, and W_y are weight matrices for input, hidden state, and output. b_h and b_y are bias terms for the hidden state, and g_1 and g_2 represent activation functions (Li et al., 2018).

A basic RNN is defined by these equations, in which the input and the previous hidden state are used to update the hidden state at each time step.

Based on input and output quantities, there are four main types of RNN architectures: One-to-one: are the most basic type of RNNs. Image classification is an example of one-to-one RNNs. One-to-many: generates a sequence of outputs from a single input. This architecture can be used for in music generation and image captioning. Many-to-one: uses a given set of inputs to produce a single output. This type of RNNs is frequently observed in tasks such as sentiment analysis (Dadoun & Troncy, 2020). Many-to-many: generates a set of outputs after receiving a set of inputs. Machine translation is an example of this type of RNN.

However, RNNs have a difficulty with vanishing gradients and are unable to retain long-term information. There are several approaches to solving this issue. Two widely used and effective variants of advanced RNN like Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) are among the popular approaches.

3.2. Long short-term memory

In contrast to a typical RNN, which has a straightforward structure of input, hidden state, and output, an LSTM has a more complicated structure with additional memory cells and gates that allow it to selectively remember or forget information from past time steps (Smagulova & James, 2019). The LSTM has become very popular for a variety of problems requiring sequence modeling and natural language processing. This architecture presents a memory cell, which gives the network the capacity to store and retrieve data over long sequences, making it easier to gather important contextual data (Alguliyev, Aliguliyev, & Abdullayeva, 2019).

The four essential components of an LSTM network are the forget gate, input gate, output gate, and cell state (Gers et al., 2000).

The input gate ($i^{<t>}$) - allows us to update the cell state. This gate determines which input values should be added to the cell state.

Forget gate ($f^{<t>}$) - which data from the previous cell state should be kept and which should be forgotten is determined by the forget gate. The output values range from 0 to 1. If the value close to zero, it represents forgetting, and if it close to one, it represents retaining.

The cell state ($c^{<t>}$) - is important for passing information through different time steps. With every new input, the cell state is updated, allowing the network to remember important information and forget irrelevant details.

The output gate ($o^{<t>}$) - chooses which part of the updated cell state will be used as the output for the current time step.

$$i^{<t>} = \sigma(W_i h^{<t-1>} + W_i x^{<t>} + b_i), \quad (3)$$

$$f^{<t>} = \sigma(W_f h^{<t-1>} + W_f x^{<t>} + b_f), \quad (4)$$

$$o^{<t>} = \sigma(W_o h^{<t-1>} + W_o x^{<t>} + b_o), \quad (5)$$

$$g^{<t>} = \tanh(W_g h^{<t-1>} + W_g x^{<t>} + b_g), \quad (6)$$

$$c^{<t>} = f^{<t>} \odot c^{<t-1>} + i^{<t>} \odot g^{<t>}, \quad (7)$$

$$h^{<t>} = o^{<t>} \odot \tanh(c^{<t>}), \quad (8)$$

In Eqs. (3)-(8), σ represents the sigmoid activation function, $x^{<t>}$ is the input at time t, $h^{<t-1>}$ is the previous hidden state, g_t represents candidate for cell state at timestamp(t), W_i, W_f, W_o, W_g are weights for the respective gate neurons, b_i, b_f, b_0, b_g are biases for the respective gates, $\odot$ denotes element-wise multiplication (Shiri et al., 2023a).

3.3. Gated recurrent unit

One special development in the field of recurrent neural network architectures is the GRU, aiming to overcome essential limitations associated with traditional RNNs. However, compared to LSTM, GRU has a simpler architecture.

GRU were proposed for two main reasons:

1) to solve the problem of vanishing gradients

2) to determine long-range dependencies in data.

The GRU architecture consists of the following components (Dey & Salem, 2017):

Update gate ($u^{<t>}$) - This gate determines how much of the previous information should be carried forward into the future.

Reset gate ($r^{<t>}$) - The model uses a reset gate to determine what amount of the previous information should be discarded.

$$u^{<t>} = \sigma(W_u h^{<t-1>} + W_u x^{<t>} + b_u) \quad (9)$$

$$r^{<t>} = \sigma(W_r h^{<t-1>} + W_r x^{<t>} + b_r) \quad (10)$$

$$h'^{<t>} = tanh(W_h(r^{<t>} \odot h^{<t-1>}) + W_h x^{<t>} + b_h) \quad (11)$$

$$h^{<t>} = (1 - u^{<t>}) \odot h^{<t-1>} + u^{<t>} \odot h'^{<t>} \quad (12)$$

$$\hat{y}^{<t>} = g(W_y h^{<t>} + b_y) \quad (13)$$

In Eqs. (9)-(13), σ represents the sigmoid activation function, $x^{<t>}$ is current input at time step t, $h^{<t-1>}$ is a previous hidden state at time step t-1, $h'^{<t>}$ is a candidate hidden state, $h^{<t>}$ is an updated hidden state at time step t, W_u, W_r, W_h, W_y are weights for the update gate, reset gate, candidate hidden state, and output prediction, b_u, b_r, b_h, b_y are bias terms for the update gate, reset gate, candidate hidden state, and output prediction. g is an activation function for the output prediction, $\hat{y}^{<t>}$ is an output prediction at time step t (Yin et al., 2017).

3.4. Bidirectional RNNs

In contrast to traditional RNNs, which analyze sequences from past to future, Bidirectional RNNs process input information in two directions at the same time: forward and backward order (Schuster & Paliwal, 1997). One for processing the sequence from beginning to end and another for processing the sequence from end to beginning (Berglund et al., 2015). The network's capability to capture dependencies in both directions results in a more thorough comprehension of data. This represents the fundamental characteristic of a Bidirectional RNN architecture. For tasks like speech recognition, natural language processing, and other data analytic applications, the Bidirectional RNN is especially helpful.

In the context of bidirectional RNNs, assuming x^t as the input at time step t, hidden state as h_t at time step t and the output at time step t as $\hat{y_t}$.

The equations for Bidirectional RNN can be represented as follows (Berglund et al., 2015).

$$h_t^f = tanh(W_h^f h_{t-1}^f + W_x^f x^t + b_h^f), \quad (14)$$

$$h_t^b = tanh(W_h^b h_{t+1}^b + W_x^b x^t + b_h^b) \quad (15)$$

$$\hat{y_t} = \varphi(W_y^f h_t^f + W_y^b h_t^b + b_y), \quad (16)$$

φ is the softmax activation function, b_h^f, b_h^b and b_y are the hidden layer and output bias vectors. W_y^f, W_y^b, W_h^f, W_h^b, W_x^f, W_x^b are the weight matrices of output layer, hidden layer and input. The forward and backward directions have separate non-tied weights and hidden activations, and are denoted by the superscript f and b for forward and backward.

3.5. Bidirectional LSTM

Bidirectional Long Short-Term Memory is a type of extension of the LSTM architecture. It's widely used in natural language processing (NLP) tasks. In a bidirectional LSTM, the input sequence is processed forward and backward directions, while a standard LSTM processes input sequences only in the forward direction. Bidirectional LSTMs gather deeper insights of the sequence's flow by simultaneously accessing information from the past and the future, the ability to detect long-range dependencies may be greatly improved by this increased context awareness, which could result in better performance on a variety of tasks (Imrana et al., 2021).

Essential parts of a Bidirectional LSTM are two sets of hidden states: While one captures information from the past, the other captures it from the future. The output sequence $\hat{y_t}$ for each time step is produced by combining these two sets of hidden states, thus (Graves et al., 2013; Mousa & Schuller, 2017):

$$h_t^f = \mathcal{H}(W_h^f h_{t-1}^f + W_x^f x^t + b_h^f), \quad (17)$$

$$h_t^b = \mathcal{H}(W_h^b h_{t+1}^b + W_x^b x^t + b_h^b) \quad (18)$$

$$\hat{y_t} = W_y^f h_t^f + W_y^b h_t^b + b_y, \quad (19)$$

$\mathcal{H}$ is an element-wise application of the sigmoid function.

Bidirectional LSTM is a popular option, for many NLP tasks, including sentiment analysis, text classification, machine translation, and speech recognition.

3.6. Bidirectional GRU

Like Bidirectional RNNs and Bidirectional LSTM, Bidirectional GRU processes information in forward and backward order. As a result, Bidirectional GRU is able to extract contextual information from both directions of the input sequence (Lynn et al., 2019). This leads to a deeper comprehension of the information. If update gate z_t^f, reset gate r_t^f, $\tilde{h}_t^f$ candidate hidden state, the hidden state h_t^f at time step t, so forward GRU equations can be described as follows (Liu et al., 2020):

$$z_t^f = \sigma(W_z^f x_t + W_z^f h_{t-1}^f + b_z^f), \quad (20)$$

$$r_t^f = \sigma(W_r^f x_t + W_r^f h_{t-1}^f + b_r^f), \quad (21)$$

$$\tilde{h}_t^f = \tanh(W_h^f x_t + W_h^f (r_t^f \odot h_{t-1}^f) + b_h^f), \quad (22)$$

$$h_t^f = (1 - z_t^f) \odot \tilde{h}_t^f + z_t^f \odot h_{t-1}^f \quad (23)$$

σ represents the sigmoid activation function, x_t is the input at time step t, W_z^f, W_r^f and W_h^f are weight matrices. b_z^f, b_r^f, b_h^f are bias terms.

The reversed input is sent into the backward GRU. It executes a sequence of calculations at time step t, much like the forward GRU, and generates a hidden state h_t^b, which is combined with h_t^f to create the final output.

4. Experimental results

Three datasets were used in our experimental research: imdb reviews, Twitter Sentiment Dataset, and Emotions dataset for NLP. The aim of the study was to compare different deep learning models. Six various models were analyzed: RNN, LSTM, GRU, Bidirectional RNN, Bidirectional LSTM, and Bidirectional GRU. Our goal was to evaluate these models' performance.

The imdb_reviews is a large dataset offers a set of movie reviews grouped into positive or negative sentiments. The dataset is split into training and testing sets, each containing 25,000 reviews (Maas et al., 2011). Training samples contains 12,500 positive and 12,500 negative samples and testing sets likely follows the same distribution (positive: 12,500, negative: 12,500) together constitute the total of 50,000 reviews in the dataset.

Twitter Sentiment dataset provides tweets categorized into three sentiments (positive: 72250, negative: 35510, and neutral: 55213) for sentiment classification purposes. 162,980 unique tweets, providing a rich resource for sentiment classification (Hussein, 2021).

Emotions dataset for NLP is a collection of documents with their associated emotions, sourced from Kaggle. Anger (2159), fear (1937), joy (5362), love (1304), sadness (4666), and surprise (572) are the six fundamental emotions represented in this collection. Dataset comprises 15,999 samples (Saravia et al, 2018).

The models were trained for a range of epochs (5-20) and a unit size of 64 for each dataset. Based on this evaluation, 8 epochs emerged as the optimal point where the models achieved good performance (high validation accuracy with minimal overfitting) for all three datasets.

To analyze these models' performance, cross-entropy loss and accuracy assessment metrics were used. The Cross-Entropy Loss is a popular loss function for sentiment analysis applications. This loss function is very useful for classification tasks, like sentiment analysis, in which the objective is to divide text into predetermined groups. Cross-entropy loss is given by (Connor et al., 2024):

$$H(p, q) = - \sum_{k \in classes} p(k) \log q(k) \quad (24)$$

Where $p(k)$ is the true probability distribution (one-hot) and $q(k)$ is the predicted probability distribution.

Represented accuracy assessment metric in Eq. (25) (Gaafar et al., 2022; Shiri et al., 2023b):

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (25)$$

here, TP – True Positives, FN – False Negatives, FP – False Positives, TN – True Negatives.

Through conducting a comparative analysis using accuracy metric on deep learning models, valuable insights can be obtained concerning the performance and effectiveness of these models in sentiment analysis.

In this study, six various deep learning models were applied on datasets for sentiment categorization. The results of the comparison study are presented on deep learning models, including RNN, LSTM and GRU, as well as their bidirectional variants using datasets specifically designed for sentiment analysis purposes. Six deep learning models were evaluated using three different datasets.

Table 1.Accuracy results of deep learning models

Model	IMDB	Twitter	Emotions
RNN	0.6131	0.9279	0.9901
LSTM	0.9750	0.9942	0.9945
GRU	0.9898	0.9890	0.9968
Bi RNN	0.6955	0.9282	**0.9970**
Bi LSTM	0.9861	**0.9954**	0.9966
Bi GRU	**0.9894**	0.9889	0.9965

Dataset-based analysis: Imdb review Dataset: RNN has the lowest accuracy (0.6131), indicating poor performance for sentiment analysis. Bidirectional RNN model performs better than RNN model but still relatively low at 0.6955. They perform similarly. GRU performs better than RNN and LSTM. With an accuracy of 0.9894, Bi GRU performs much better than all other models. Bi

LSTM with an accuracy of 0.9861, demonstrate high result.

Twitter Sentiment Dataset: LSTM performs noticeably better than RNN and GRU. The highest performing models are LSTM and Bi LSTM, with accuracies of 0.9942 and 0.9954. With accuracies of 0.9379 and 0.9282, RNN and Bi RNN both show lowest results.

Emotions Dataset: A relatively small number separates RNN from other models. While LSTM performs very well, GRU is much better, with a 0.9968 accuracy. All bidirectional models perform excellently. Bi RNN has the highest accuracy result.

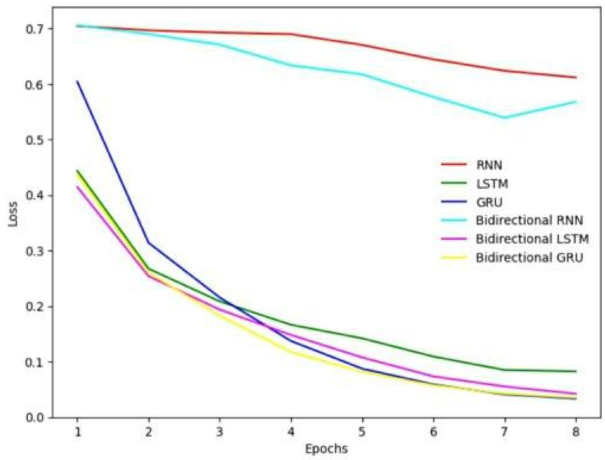
Model-based analysis: RNN: shows the lowest performance across all datasets, indicating that simple RNNs are less effective for these tasks.

LSTM and GRU has consistent results; typically performs better than RNN, demonstrating increased effectiveness and capacity for learning.
Bi RNN: while it performs moderately on other datasets, Bi RNN has best results on the Emotions dataset.

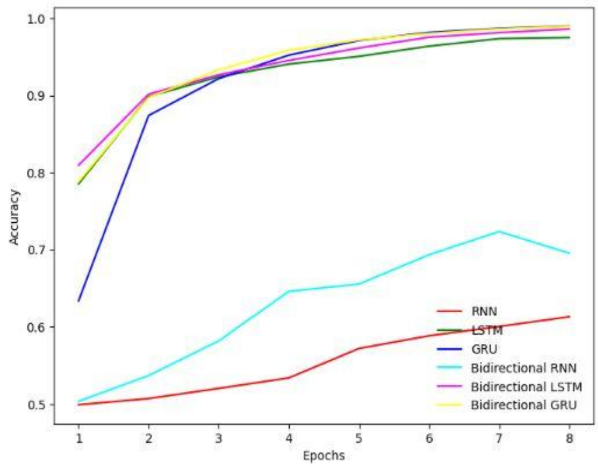
Bi LSTM and Bi GRU: are the most reliable models for a range of applications since they consistently perform the best across all datasets.

Bidirectional models perform better on all datasets than their unidirectional equivalents. This demonstrates the benefit of obtaining context from sequences that are both past and future. Unidirectional models typically perform worse; GRU and LSTM outperforms RNN in most cases.

Loss and accuracy graphs were used to analyze these models' performances. These graphs offer insights into the models' learning process from the data. The x-axis is labeled "Epochs" and refers to the number of times the training data is passed through the neural network. The y-axis is labeled 'Loss' and 'Accuracy' which refers to how well the model is performing on a specific task. Figures 1 and 2 visually illustrate the changes in loss and accuracy values during the training of deep learning models on ***imdb_reviews*** dataset. Every graph presents a plot with a count of epochs on the x-axis and losses or accuracy for each epoch on the y-axis.



(a)



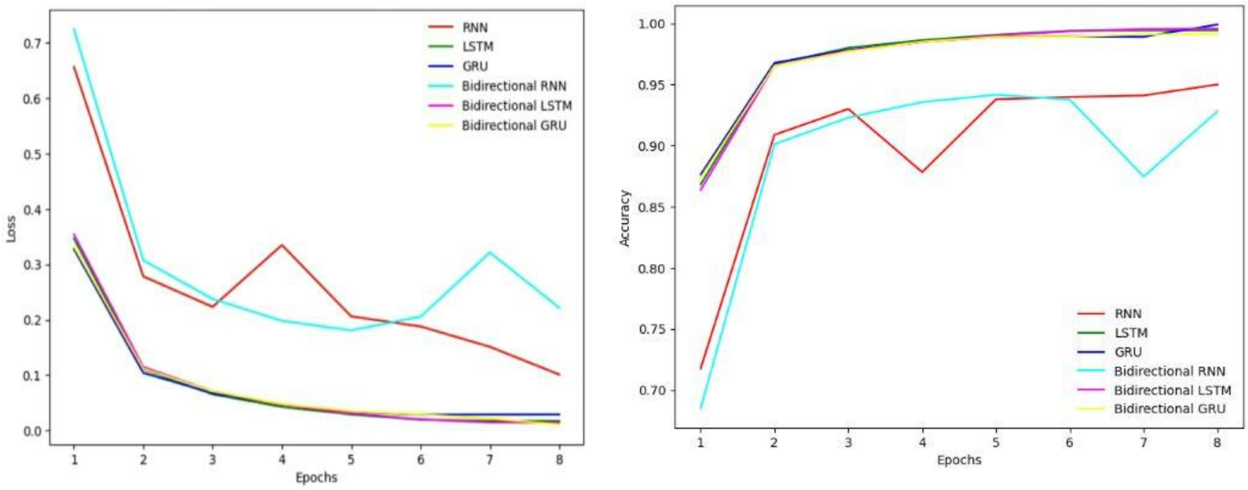
(b)

Fig. 1. Performance evaluation of deep learning models on the ***imdb_reviews*** dataset: (a) Loss and (b) Accuracy

It is possible to make a comprehensive analysis of the models' performance on the imdb_reviews dataset based on the provided graphs.

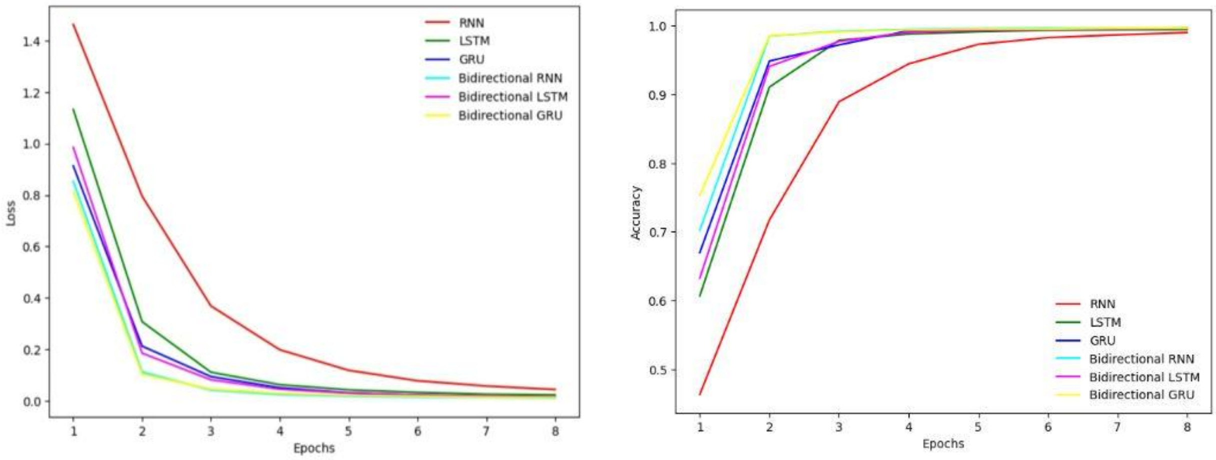
Lower loss and highest accuracy indicates better performance. Overall, the performance of all the models improves as the number of epochs increases, which means the models, are learning from the training data. RNN and bidirectional RNN are the least effective models, with accuracy and loss improving slowly. Compared to RNN and LSTM, GRU is more efficient, with GRU showing a faster learning rate. Bidirectional LSTM and GRU models perform better in both loss reduction and accuracy increase, making them the best options for the IMDB dataset. While unidirectional GRU is a better choice than RNN and LSTM, it is still not as effective as bidirectional models. For this assignment, RNN model performs the worst, which struggles to significantly decrease loss and increase accuracy. Compared to their unidirectional counterparts, bidirectional model (particularly LSTM and GRU) converge more quickly, achieve lower loss, and greater accuracy faster.

Figures 3 and 4 visually illustrate the changes in loss and accuracy values during the training of deep learning models on *Twitter Sentiment* dataset. RNN is the least stable model, with noticeable differences in accuracy and loss. Bidirectional RNN exhibits instability even if it performs better than RNN. LSTM, GRU, Bidirectional LSTM, and Bidirectional GRU models are more stable. The Bidirectional variants exhibit the highest level of stability. Bidirectional LSTM and Bidirectional GRU outperform all other models, achieving the lowest loss and highest accuracy. While still performing well, LSTM and GRU lag behind their bidirectional counterparts. RNN and Bidirectional RNN perform worse than one another, with greater loss and less accuracy.



(a) (b)

Fig. 2. Performance evaluation of deep learning models on the ***Twitter Sentiment*** dataset: (a) Loss and (b) Accuracy



(a) (b)

Fig. 3. Performance evaluation of deep learning models on the ***Emotion*** dataset: (a) Loss and (b) Accuracy

RNN is less suitable for this assignment according to these graphs. Exhibits the highest initial loss and the slowest decrease in loss over time. By the end of the 8 epochs, it still has a higher loss compared to other models, indicating less efficient learning and poor performance. While both LSTM and GRU exhibit notable advancements, GRU outperforming LSTM slightly, they are still not as good as their bidirectional counterparts. The results indicate that bidirectional models are the most successful in capturing dependencies in the dataset, as it performs best in terms of both accuracy and loss reduction. Bidirectional GRU demonstrates that it is also quite successful for this kind of task, since it performs similarly to Bidirectional LSTM and bidirectional RNN. Bidirectional models—specifically, Bidirectional RNN and Bidirectional GRU—perform best for the Emotion dataset in terms of accuracy and loss reduction. This suggests that for emotion detection tasks, collecting context in both directions is essential.

5. Conclusion and future work

This paper provides a thorough analysis of deep learning models. Several deep learning models, including RNN, LSTM, GRU, and their bidirectional variants, are covered in the article.
Three publicly available datasets were used in the experiments: imdb_reviews, Twitter Sentiment, and Emotions. The results of our experiments demonstrate that best performing models are bidirectional LSTM and bidirectional GRU for achieving the highest accuracy across most datasets, demonstrating their exceptional capacity capturing and using bidirectional context. They are the best options for tasks that need context knowledge from both directions. Generally, RNN perform poorly, particularly on complicated datasets.

Bidirectional LSTM or Bidirectional GRU models are recommended for practical applications due to their improved accuracy, if computational resources are limited, unidirectional GRU models can be an effective alternative, providing a good performance and efficiency.

The achievements could be very useful many different industries and applications that leverage sentiment analysis tasks, which may include social media analysis, customer feedback analysis and so on. Regarding sentiment analysis tasks, achievements offer insightful information on how various deep learning models perform. Identifying which models produce the best accuracy impacts future study and application development. With the assist of these insights, researchers and professionals may choose and optimize deep learning models for sentiment analysis tasks with more knowledge, which will eventually result in more precise and efficient solutions.

Acknowledgment

This work was supported by the Azerbaijan Science Foundation-**Grant № AEF-MCG-2023-1(43)-13/04/1-M-04**

References

Ahmad, A. M., Ismail, S., & Samaon, D. F. (2004). Recurrent neural network with backpropagation through time for speech recognition. In IEEE International Symposium on Communications and Information Technology (ISCIT), Sapporo, Japan, October 2004 (pp. 98-102). https://doi.org/10.1109/ISCIT.2004.1412458

Alguliyev, R. M., Aliguliyev, R. M., & Abdullayeva, F. J. (2019). Deep learning method for prediction of DDoS attacks on social media. Advances in Data Science and Adaptive Analysis, 11(01n02), 1950002. https://doi.org/10.1142/S2424922X19500025

Alguliyev, R. M., Aliguliyev, R. M., & Abdullayeva, F. J. (2019). The improved LSTM and CNN models for DDoS attacks prediction in social media. International Journal of Cyber Warfare and Terrorism (IJCWT), 9(1), 1-18. https://doi.org/10.4018/IJCWT.2019010101

Alguliyev, R. M., Aliguliyev, R. M., & Niftaliyeva, G. Y. (2019). Extracting social networks from e-government by sentiment analysis of users' comments. Electronic Government, an International Journal, 15(1), 91-106. https://doi.org/10.1504/EG.2019.096576

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. IEEE transactions on neural networks, 5(2), 157-166. https://doi.org/10.1109/72.279181

Berglund, M., Raiko, T., Honkala, M., Kärkkäinen, L., Vetek, A., & Karhunen, J. T. (2015). Bidirectional recurrent neural networks as generative models. Advances in neural information processing systems. arXiv preprint arXiv: 1504.01575.

Cho, K., Van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv: 1409.1259.

Connor, R., Dearle, A., Claydon, B., & Vadicamo, L. (2024). Correlations of cross-entropy loss in machine learning. Entropy, 26(6), 491. https://doi.org/10.3390/e26060491

Dadoun, A., & Troncy, R. (2020). Many-to-one recurrent neural network for session-based recommendation. arXiv preprint arXiv: 2008.11136.

Dey, R., & Salem, F. M. (2017). Gate-variants of gated recurrent unit (GRU) neural networks. In 60th IEEE International Midwest Symposium on Circuits and Systems (MWSCAS), Boston, USA, August 2017 (pp. 1597-1600). https://doi.org/10.1109/MWSCAS.2017.8053243

Gaafar, A. S., Dahr, J. M., & Hamoud, A. K. (2022). Comparative analysis of performance of deep learning classification approach based on LSTM-RNN for textual and image datasets. Informatica, 46(5), 21-28. https://doi.org/10.31449/inf.v46i5.3872

Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. Neural computation, 12(10), 2451-2471. https://doi.org/10.1049/cp:19991218

Graves, A., Mohamed, A. R., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In IEEE international conference on acoustics, speech and signal processing, Vancouver, Canada, May 2013 (pp. 6645-6649). https://doi.org/10.1109/ICASSP.2013.6638947

Hassan, A., & Mahmood, A. (2017, April). Deep learning approach for sentiment analysis of short texts. In 3rd international conference on control, automation and robotics (ICCAR), Nagoya, Japan, April 2017 (pp. 705-710). https://doi.org/10.1109/ICCAR.2017.7942788

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hussein, S. (2021). Twitter Sentiments Dataset. Mendeley Data. https://doi.org/10.17632/z9zw7nt5h2.1

Imrana, Y., Xiang, Y., Ali, L., & Abdul-Rauf, Z. (2021). A bidirectional LSTM deep learning approach for intrusion detection. Expert Systems with Applications, 185, 115524. https://doi.org/10.1016/j.eswa.2021.115524

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444. https://doi.org/10.1038/nature14539

Li, S., Li, W., Cook, C., Zhu, C., & Gao, Y. (2018). Independently recurrent neural network (indRNN): Building a longer and deeper RNN. In IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, USA, June 2018 (pp. 5457-5466). https://doi.org/10.1109/CVPR.2018.00572

Liu, X., You, J., Wu, Y., Li, T., Li, L., Zhang, Z., & Ge, J. (2020). Attention-based bidirectional GRU networks for efficient HTTPS traffic classification. Information Sciences, 541, 297-315. https://doi.org/10.1016/j.ins.2020.05.035

Lynn, H. M., Pan, S. B., & Kim, P. (2019). A deep bidirectional GRU network model for biometric electrocardiogram classification based on recurrent neural networks. IEEE Access, 7, 145395-145405. https://doi.org/10.1109/ACCESS.2019.2939947

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011). Learning word vectors for sentiment analysis. In 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, USA, June 2011 (pp. 142-150).

Mousa, A., & Schuller, B. (2017). Contextual bidirectional long short-term memory recurrent neural network language models: A generative approach to sentiment analysis. In 15th Conference of the European Chapter of the Association for Computational Linguistics, Valencia, Spain, April 2017 (pp. 1023–1032).

Pandya, D., & Thakkar, A. (2024). Sentiment Analysis of Self Driving Car Dataset: A comparative study of Deep Learning approaches. Procedia Computer Science, 235, 12-21. https://doi.org/10.1016/j.procs.2024.04.002

Park, J., Yi, D., & Ji, S. (2020). Analysis of recurrent neural network and predictions. Symmetry, 12(4), 615. https://doi.org/10.3390/sym12040615

Saravia, E., Liu, H. C. T., Huang, Y. H., Wu, J., & Chen, Y. S. (2018). CARER: Contextualized affect representations for emotion recognition. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October-November 2018 (pp. 3687-3697). https://doi.org/10.18653/v1/D18-1404

Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. IEEE transactions on Signal Processing, 45(11), 2673-2681. https://doi.org/10.1109/78.650093

Shiri, F. M., Perumal, T., Mustapha, N., & Mohamed, R. (2023a). A comprehensive overview and comparative analysis on deep learning models: CNN, RNN, LSTM, GRU. arXiv preprint arXiv:2305.17473.

Shiri, F., Perumal, T., Mustapha, N., Mohamed, R., Ahmadon, M. A. B., & Yamaguchi, S. (2023b). A Survey on Multi-Resident Activity Recognition in Smart Environments, arXiv preprint arXiv: 2304.12304.

Smagulova, K., & James, A. P. (2019). A survey on LSTM memristive neural network architectures and applications. The European Physical Journal Special Topics, 228(10), 2313-2324. https://doi.org/10.1140/epjst/e2019-900046-x

Tang, D., Qin, B., & Liu, T. (2015). Document modeling with gated recurrent neural network for sentiment classification. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, September 2015 (pp. 1422-1432).

Wankhade, M., Rao, A. C. S., & Kulkarni, C. (2022). A survey on sentiment analysis methods, applications, and challenges. Artificial Intelligence Review, 55(7), 5731-5780. https://doi.org/10.1007/s10462-022-10144-1

Yin, W., Kann, K., Yu, M., & Schütze, H. (2017). Comparative study of CNN and RNN for natural language processing. arXiv preprint arXiv:1702.01923.