An Experimental Study of Reccurent Neural Networks

Samuel Olowofila
Department of Computer Science
University of Colorado Colorado Springs
CO, 80918 USA

Abstract

This paper investigates the application of Recurrent Neural Networks (RNNs) in sentiment analysis, using the IMDB movie review dataset to evaluate various RNN architectures including basic RNNs, Long Short-Term Memory (LSTM), and Gated Recurrent Units (GRU). The research involves preprocessing with the Word2Vec technique for word embeddings, followed by experiments with different RNN configurations exploring various activation functions, and advanced LSTM and GRU models. Notably, the Tanh activation function outperforms others in baseline RNNs, challenging established norms. In-depth analysis shows that while LSTM excels in capturing complex patterns, GRU demonstrates greater computational efficiency. The study also explores the impact of bidirectionality in RNNs, assessing its influence on model performance across varying architectures. The findings confirm the effectiveness of RNNs, particularly LSTM and GRU, in sentiment analysis, underscoring their capability to handle complex sequential data and long-range dependencies.

Introduction

In the digital age, the vast proliferation of textual data on the internet has given rise to the need for automated sentiment analysis techniques. Sentiment analysis, often referred to as opinion mining, plays a crucial role in understanding and categorizing the subjective sentiments expressed in textual content. The ability to automatically classify opinions as positive, negative, or neutral has found extensive applications in areas such as marketing, customer feedback analysis, and even political discourse analysis (Tang, Qin, and Liu 2015). This paper presents a study on the application of Recurrent Neural Networks (RNNs) to the task of sentiment analysis using the renowned IMDB movie review dataset. Sentiment analysis, in this context, involves the computational analysis of movie reviews to determine whether the sentiments expressed within them are predominantly positive or negative.

In the dynamic and ever-evolving domain of artificial intelligence, Recurrent Neural Networks (RNNs) stand as a beacon of innovation, particularly in the natural language processing sphere. The exhibited prominence is due to their effectiveness in capturing sequential dependencies within textual data (Yin et al. 2017). RNNs are well-suited for tasks involving sequences, making them a natural choice for analyzing the temporal and contextual nuances inherent in language (Tarwani and Edem 2017). By employing RNNs, we aim to leverage the rich contextual information present in movie reviews to enhance the accuracy of sentiment classification.

Approach

RNNs are characterized by a specific formula that governs their operation, especially in how they process sequences of data over time (Mikolov et al. 2010). The fundamental aspect of RNNs is their ability to maintain a 'memory' of previous inputs by incorporating the output of a neuron back into itself.

$$h_t = f(W_{hh}h_{t-1} + W_{xh}x_t + b_h) \tag{1}$$

Given a sequence of inputs $x_1, x_2, ..., x_t$ where x_t is the input at time step t, Equation 1 shows how the hidden state h_t is computed, where W_{hh} is the weight matrix for the hidden-to-hidden connections, W_{xh} is the weight matrix for the input-to-hidden connections, x_t is the current input vector, b_h is the bias vector for the hidden state, and the activation function of choice is represented as f.

This study begins with the IMDb movie review dataset pre-processing. The data comprises 50,000 reviews, each accompanied by a label indicating the sentiment class as expressed in the review; positive or negative. Firstly, the review texts were stripped of non-alpha-numeric characters, tokenized into sequences of words, and the sequences are padded to have a consistent length of 25 words for input into the model. The Word2Vec technique is then used to derive 300-dimenional embeddings for the words in the dataset (Rong 2014). These word vectors capture semantic relationships between words based on their context in the reviews. For this experiment, we practically explore the nuances of the recurrence dynamics across several RNN architecture adaptations, and analyzed in the Results section. Throughout the series of experiments, the dataset is randomized and split so that 80% is used for training and 20% for validation.

Baseline RNN

Earlier on in this experiment, we implement a single layer Recurrent Neural Network. This architecture features an embedding layer, responsible for mapping each input work token into a 300-dimensional vector space, followed by a Simple RNN layer obtained from the Tensorflow Keras library. This layer is configured to exhibit 32 recurrent units, uses the binary crossentropy loss function, and applies the adaptive moment estimation (Adam) optimizer. This phase of the experiment particularly examines the performances of 3 variants of the simple RNN model, pivoting on the Sigmoid, Rectified Linear Unit (ReLU), and the Hyperbolic Tangent (Tanh) activation functions, which are applied at the output end of the simple RNN layer per experiment instance, while the dense layer is closed with the softmax activation for the classification task. Noteworthy is the fact that all training instances performed in the course of this study were limited to 10 epochs. The corresponding performances are recorded in Table 1.

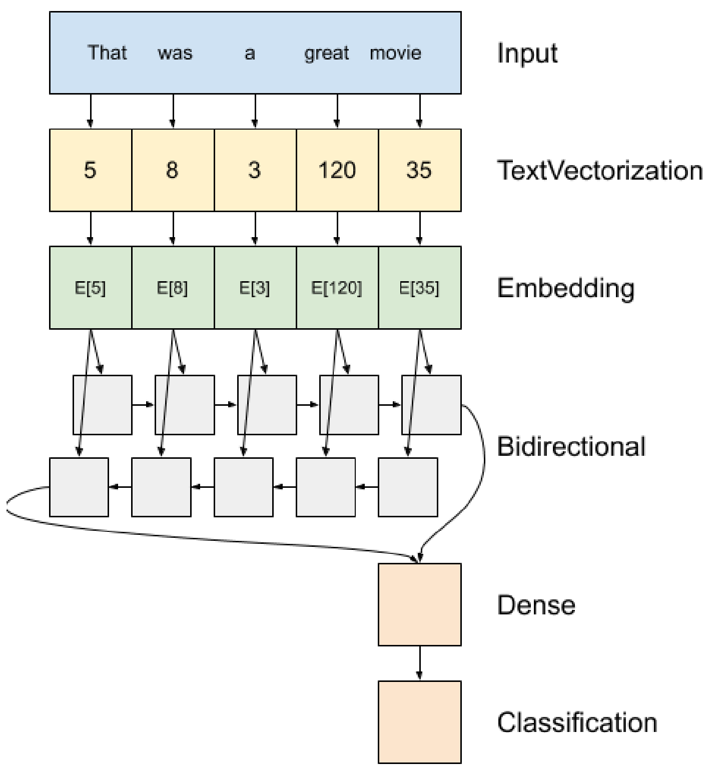

Figure 1: The Bidirectional RNN Architecture

Dual-layer RNN

The dual-layer RNN model is here design to investigate the effect of further depth in the RNN architecture. This model is technically a duplicate of the baseline RNN in the first phase of this study. It is a sequentially oriented model with an embedding layer, stacked with two similar simple RNN layers. However, the first RNN layer in this instance returns the full sequence of outputs for each sample, which is a necessity for stacking another RNN layer on top of it. The second simple RNN layer is then added. This layer, however, returns the output of the last time step only, making it suitable for feeding into the dense layer for the required classification. We conduct a variant of the experiment that emphasizes the depth with respect to layers, along with some other variations of select RNNs.

Long Short-Term Memory

The Long Short-Term Memory (LSTM) architecture is a type of RNN, and it was introduced to overcome the vanishing gradient problem common in standard RNNs. This issue occurs during backpropagation in deep networks when gradients are computed and propagated back through the network; if the gradient is very small, it diminishes as it propagates through each layer, making it difficult to learn long-range dependencies (Sherstinsky 2020). The key to LSTM's effectiveness lies in its ability to regulate the flow of information through the use of gates: the forget gate, input gate, and output gate. These gates control the addition and removal of information in the cell state, enabling the LSTM to mitigate the vanishing gradient problem and capture long-term dependencies in data.

Gated Recurrent Unit

The Gated Recurrent Unit is a more recent innovation, and is also regarded as a variation of LSTM. GRUs simplify the LSTM architecture and are designed to address the vanishing gradient problem of standard RNNs while being computationally more efficient than LSTMs. They are often used in tasks like language modeling, machine translation, and speech recognition. Their simpler structure allows for faster training times compared to LSTMs, making them a popular choice in scenarios where computational efficiency is a priority (Dey and Salem 2017).

Bi-directionality in RNNs

Bidirectionality in Recurrent Neural Networks (RNNs) is a design feature that allows the network to consider not only the past context (information from previous time steps) but also the future context (information from future time steps) when making predictions or processing sequences. This bidirectional information flow is achieved by using two separate hidden states, one for processing the sequence in the forward direction and another for processing it in the backward direction (Liu, Joty, and Meng 2015). In this phase of the study, we perform a multifacated experiment, implementing bidirectionality on the Baseline RNN, a pre-trained LSTM, and GRU models, and with varying depths ranging one to three layers per architecture.

Results

From an experimental standpoint, parts of the results obtained were somewhat expected while some others tend to challenge theoretical norms. Firstly in the early stages of the experiments, a simple RNN architecture is used to contrast the performances of three prominent activation functions; ReLU, Sigmoid, and Tanh.

Despite proven postulations claiming that sigmoid kernels are tailored and best suited for classification tasks, (Raeside 1988), our outcomes as reported in Table 1 stand to prove

	ReLU	Sigmoid	Tanh
Train Accuracy	0.9949	0.9159	**0.9978**
Validation Accuracy	0.7909	0.7600	**0.7954**
Training Time	**48.769**	53.467	49.675

Table 1: Comparison of Activation Functions performance in RNN Sentiment Analysis

otherwise, as tanh recorded superior accuracy over the others. Hence, tanh became the activation function of choice for the rest of the experiments. Moreso, GRU recorded the best runtime while ReLU was the worst in this regard. As depicted in Figure 2, a multifacated investigation was conducted to determine the best performing model when we have up to three layers of Baseline RNN, GRU, or LSTM, all exhibiting bi-directionality. Here, the Baseline RNN recorded the worst accuracies in all three instances of the examination. However, LSTM edged the others when two and three layers of the model was stacked, while GRU performed the best in the single layer instance of the experiment.

LAYER 1

	BASELINE RNN	GRU	LSTM
ACCURACY	0.7652	**0.8021**	0.802
LOSS	0.652	0.5387	**0.4331**
RUNTIME	**61.437**	94.336	100.96

LAYER 2

	BASELINE RNN	GRU	LSTM
ACCURACY	0.7735	0.7945	**0.8055**
LOSS	0.6415	0.5482	**0.4548**
RUNTIME	**81.158**	143.04	153.78

LAYER 3

	BASELINE RNN	GRU	LSTM
ACCURACY	0.7496	0.7906	**0.8095**
LOSS	0.6427	0.5098	**0.4332**
RUNTIME	**100.75**	188.84	205.92

Figure 2: Performance Comparison of Baseline RNN, GRU, and LSTM

Conclusion

Based on results obtained from this study, recurrent neural networks have proven to be significantly effective at sentiment analysis. Furthermore, it can be inferred that LSTM capture complex patterns more effiectively, despite being more time consuming compared to GRU which is computationally more efficient due to its simpler structure, making them faster to train. In summary, LSTM and GRU represent significant advancements in the field of recurrent neural networks, providing more powerful tools for modeling sequential data, especially where understanding long-range dependencies is critical.

References

Dey, R., and Salem, F. M. 2017. Gate-variants of gated recurrent unit (gru) neural networks. In *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)*, 1597–1600. IEEE.

Liu, P.; Joty, S.; and Meng, H. 2015. Fine-grained opinion mining with recurrent neural networks and word embeddings. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, 1433–1443.

Mikolov, T.; Karafiát, M.; Burget, L.; Cernockỳ, J.; and Khudanpur, S. 2010. Recurrent neural network based language model. In *Interspeech*, volume 2, 1045–1048. Makuhari.

Raeside, R. 1988. The use of sigmoids in modelling and forecasting human populations. *Journal of the Royal Statistical Society Series A: Statistics in Society* 151(3):499–513.

Rong, X. 2014. word2vec parameter learning explained. *arXiv preprint arXiv:1411.2738*.

Sherstinsky, A. 2020. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena* 404:132306.

Tang, D.; Qin, B.; and Liu, T. 2015. Deep learning for sentiment analysis: successful approaches and future challenges. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 5(6):292–303.

Tarwani, K. M., and Edem, S. 2017. Survey on recurrent neural network in natural language processing. *Int. J. Eng. Trends Technol* 48(6):301–304.

Yin, W.; Kann, K.; Yu, M.; and Schütze, H. 2017. Comparative study of cnn and rnn for natural language processing. *arXiv preprint arXiv:1702.01923*.

Supplementary Material

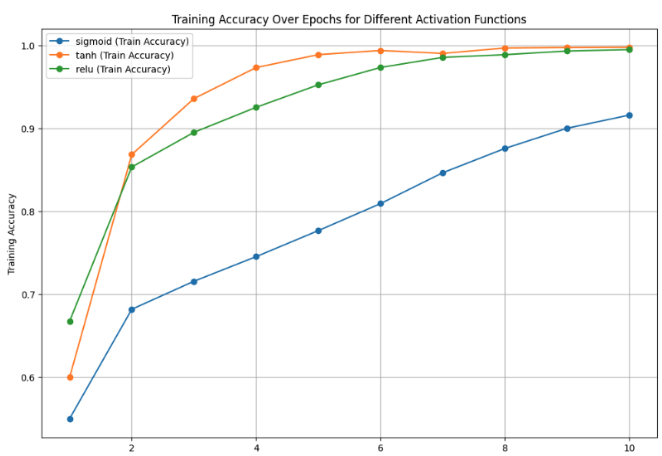

Figure 3: Training accuracies for the sentiment analysis on various activation functions using the Baseline RNN.

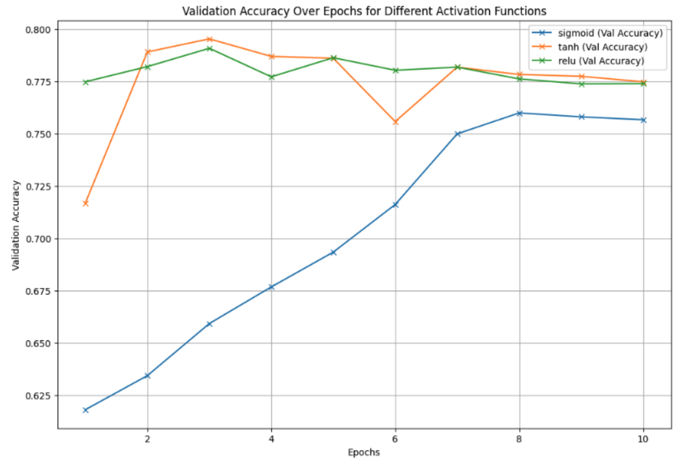

Figure 4: Validation accuracies for the sentiment analysis on various activation functions using the Baseline RNN.

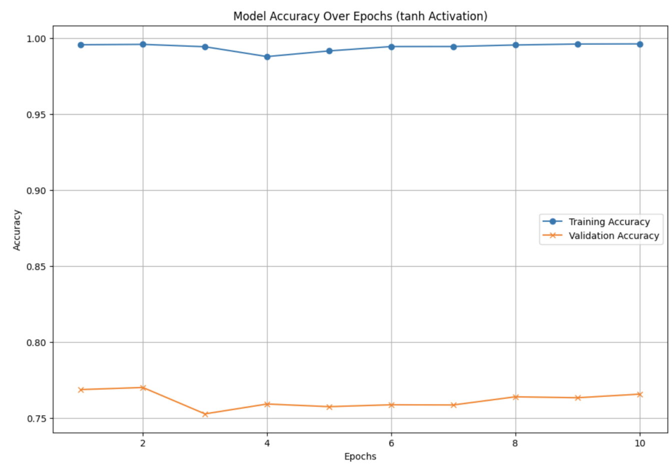

Figure 5: Training against validation accuracies for dual layer simple RNN.

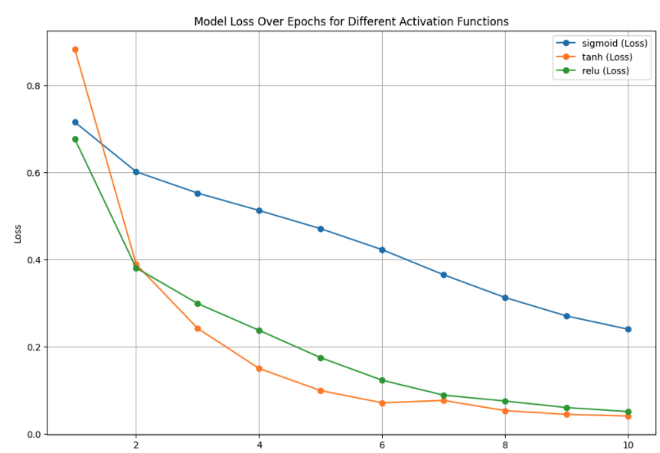

Figure 6: Loss plots for the activation function comparison..