From RNNs to BERT: A Review of Neural Models for Sequence Learning

Yuxuan Zhao

Department of Mathematics, University of Toronto, Toronto, Canada
zhaoyuxuan.zhao@mail.utoronto.ca

Abstract. Learning sequence data is important in machine learning fields, including speech recognition, natural language processing, and time series prediction. Various approaches have been put out in recent years to manage these jobs. Early models like the Recurrent Neural Network (RNN) were able to process sequential information but encountered vanishing and exploding gradients problems. These issues were eventually addressed with the introduction of the Long Short-Term Memory (LSTM) and the Gated Recurrent Unit (GRU), which enhanced the capacity to learn long-term dependencies. The proposal of the attention mechanisms further enhanced the GRU's performance and led the Transformer model to replace recurrence with attention, making training faster and more effective for large-scale data. Furthermore, BERT used pre-training and fine-tuning methods that brought a remarkable improvement in many NLP tasks. This paper reviews the development of these models, introduces the mechanisms of each model, compares their strengths and weaknesses, and finally discusses the challenges that still remain.

Keywords: Recurrent Neural Network, Long Short-Term Memory, Attention Mechanism, Transformer, Bidirectional Encoder Representations from Transformers (BERT).

1. Introduction

Many domains, including language, time series, etc., frequently use sequence data. In natural language processing (NLP), understanding and generating sentences require models that can deal with sequences. Although early approaches like n-gram models and Hidden Markov Models were useful, they had drawbacks including the requirement for large amounts of data and their inability to adequately incorporate long-term dependencies [1].

With the rise of deep learning, new models were created to solve these problems. In 1986, inspired by the coarticulation in speech research, Jordan created one of the first deep learning models designed for sequence data, the Recurrent Neural Network (RNN) [2]. Later, the Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) were developed to solve vanishing gradients problem in RNN [3,4]. These models became popular for tasks such as speech recognition and language modeling. However, these models still had limits in dealing with very long sequences. better the development of the Attention mechanism and then the Transformer model, which allowed much better performance and faster training [5,6]. Ultimately, pre-trained models like BERT, which is based on the bidirectional Transformer model, emerged as a new NLP standard [7].

The goal of this paper is to review the development of six important models: RNN, LSTM, GRU, Attention mechanism, transformer, and BERT. We will first introduce the architecture of each network, and then describe their strengths and weaknesses, and finally show how they connect to each other in the history of sequence modeling.

2. Development

2.1. Recurrent Neural Networks (RNN)

Recurrent Neural Networks (RNN) are proposed by Michael I. Jordan in 1986. They are a class of artificial neural networks that are used to analyze sequential input. These networks are inspired by sequential behaviors of humans, where humans' speech system predicts the future actions and changes the shape of lip before pronouncing, which allows people to speak fast [2]. The RNN introduces recurrence (i.e., one or more cycles) to the neural network, which enables the networks' hidden state to retain contextual information from earlier time steps.

In Elman network (1990), the hidden state at time step t, h_t, is

$$h_t = \sigma\left(W_{xh}x_t + W_{hh}h_{t-1} + b_h\right), \tag{1}$$

where σ is a non-linear activation function, usually tanh; W_{hh} and W_{xh} are weight matrices for the recurrent and input neuron, respectively; x_t is the input vector; and b_h is the bias for the hidden state. Moreover, the output at time step t, y_t, is

$$y_t = W_{hy}h_t + b_y, \tag{2}$$

where W_{hy} is the output weight matrix, and b_y is the bias for the output [8].

There are several uses for RNNs, such as character-level and word-level language modeling [9], speech recognition [10], etc. In the language modeling, RNNs successfully generate predictive texts and probabilistic modeling of sequences; and in speech recognition, RNNs can effectively model the temporal patterns in audio signals. However, the RNNs encounter problems, such as they can only use the previous context [10], and the gradient vanishes or explodes when processing long sequences, because repeated multiplication of weight matrices causes gradients to decay or grow exponentially [11]. Hence, later networks like Long Short-Term Memory (LSTM), are created to address these limitations of RNNs.

2.2. Long Short-Term Memory (LSTM)

Hochreiter and Schmidhuber proposed the LSTM model in 1997. It was aimed at fixing the exponentially decaying gradient error in RNNs. LSTM achieves this by introducing memory cells and gated units (input, output, and forget gates). The input gate limits the data being added to the memory cells, the forget gate removes unimportant sequences from the memory cells, and the output gate decides the outcome from memory cells. To prevent conflicts between input and output weights, each cell is constructed around a central linear unit with a constant error carousel (CEC) [3].

These structures make LSTM possible to learn tasks that contain extensive time intervals, while the RNNs are hard to handle. LSTM's ability to capture long-range dependencies led to a variety of applications, including speech recognition, time-series predicting, language modeling, etc. Graves et al in 2013 compared the ability of RNN and LSTM in speech recognition and reached a result that LSTM performs much better than the traditional $\tanh$ model [10], Sundermeyer et al applied LSTM in two language modeling tasks [12], and Ma et al predicted traffic speed in Beijing using LSTM [13].

Due to its advantages, it has been one of the most popular networks after it was proposed. However, LSTM has its own limitations – compared to RNN, LSTM's calculations are much more complex, making it expensive to use. This was not improved until 2014, when further models are put forward.

2.3. Gated Recurrent Unit (GRU)

Gated Recurrent Unit (GRU) was a streamlined variant of LSTM proposed by Cho et al. in 2014, initially called RNN Encoder-Decoder. It creates an update gate by combining input and forget gates:

$$z = \sigma\left(W_z \cdot \left[h_{t-1},\ x\right]\right), \tag{3}$$

and creates a reset gate by combining the cell state and hidden state:

$$r = \sigma\left(W_r \cdot \left[h_{t-1},\ x\right]\right), \tag{4}$$

where σ is the logistic sigmoid function, W_r, W_z are the weight matrices, h_{t-1} is the hidden state at last time step, and x is the input. This leads to fewer parameters, making GRUs quicker to train, and more computationally efficient, particularly in environments with limited resources [4].

Preliminary experiments show that GRU and LSTM have similar performance in the tasks of sequence modeling, while both outperform the traditional tanh unit RNNs [14]. Empirical evaluations show that while LSTMs may outperform GRUs on very long sequences or highly complex strings such as language modeling, GRUs perform better than LSTMs for low-complexity dependencies [15,16].

The compact design of GRUs makes them more reliable when training data is limited, since it reduces the risk of overfitting. Hence, GRUs have been widely adopted in the tasks that both performance and efficiency are critical, including time series forecasting [17] and machine translation, which also inspired the invention of attention mechanism [5].

2.4. Attention mechanism

Presented by Bahdanau et al. in 2015, a novel architecture called the attention mechanism was to enhance the efficiency of traditional encoder–decoder networks such as GRU, which degenerates quickly when the length of sentences increases. This mechanism uses the bidirectional RNN as the encoder, and simulates the searching process when decoding [5]. The encoder maps the input to a sequence of hidden states $(h_1, \ldots, h_{T_x})$, called annotations, and then the decoder computes a weighted sum of the annotations as the context vector c_t :

$$c_t = \sum_{i=1}^{T_x} \alpha_{t,i} h_i, \tag{5}$$

where $\alpha_{t,i}$ are the attention weight, computed via a score function based on the hidden state s_{i-1} of RNN and the j-th annotation h_j :

$$\alpha_{t,i} = \frac{\exp(\text{score}(s_{t-1}, h_i))}{\sum_{j=1}^{T_x} \exp(\text{score}(s_{t-1}, h_j))}. \tag{6}$$

By enabling selective access to different input positions, attention mechanisms significantly improve sequence to sequence (Seq2Seq) performance, especially for long-range dependencies. Today, attention mechanisms are widely applied not only in machine translation but also in image captioning and time series forecasting [17,18]. The attention mechanism’s innovation also laid the foundation for the Transformer architecture, where self-attention mechanisms completely replace the recurrence architecture and enable highly parallelizable computation [6].

2.5. Transformers

Vaswani et al. presented the Transformer architecture in their article “Attention is All You Need.” in 2017. The novel architecture completely removes the recurrent structures in previous networks, and solely relying on self-attention mechanisms. It enables the Transformer to deal with highly parallelizable computations and can capture global dependencies more effectively. The model introduces three matrices: query matrix Q, key matrix K, and value matrix V, which compute the matrix of output as

$$Attention\,(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \tag{7}$$

The self-attention architecture avoids the leftward information flow that previous recurrent mechanisms used. The decoder of the Transformers pay attention to all positions in a sentence, and thus can capture richer contextual information at different semantic levels [6].

The Transformer’s ability to process data in parallel and capture global context makes it much more efficient and powerful than older sequence models. In jobs like machine translation, English constituency parsing, and text generation, it has demonstrated remarkable performance, since it can notice the connections between words in one sentence such as the pronouns’ reference [6]. More importantly, it became the foundation for later pre-trained models like BERT [7].

2.6. Bidirectional Encoder Representations from Transformers (BERT)

BERT, proposed by Devlin et al. in 2019, is one of the most influential models in NLP. While the earlier Transformers included both an encoder and a decoder, the BERT is built on a multi-layer bidirectional Transformer that only has an encoder. The bidirectional refers to that when dealing with sentences, the model will read the context both from left to right and from right to left simultaneously, while the basic models can only process the information in one direction.

The success of BERT introduced the pre-training and fine-tuning paradigm. BERT pre-training uses big text corpus like Wikipedia and BooksCorpus, and requires two tasks: the Masked Language Model (MLM) and the Next Sentence Prediction (NSP). The MLM task randomly hides several words in the input sentence, and the BERT must anticipate these components according to the context. NSP gives BERT two sentences, and the model must decide whether the second sentence really follows the previous one, which improves BERT's understanding of sentence relationships. In fine-tuning, the BERT is adapted to specific tasks like sentiment analysis and question answering.

These approaches led BERT to achieve excellent results on many benchmarks, such as GLUE and SQuAD [7]. It also inspired many later models, such as RoBERTa and ALBERT, which improved efficiency and performance [19,20]. Overall, BERT shows the effectiveness of large-scale pre-trained language models.

3. Conclusion

This paper reviews the development of sequence learning models, in their development order from RNN to LSTM and GRU, and later to the Attention mechanism, Transformer, and BERT. The later models solved important problems of the previous ones. RNNs introduced recurrent structures for sequence data, LSTMs and GRUs improved the learning of long-term dependencies, while Attention mechanisms proposed important architectures that enable the later Transformer model to replace recursion with attention, and BERT brought pre-training methods that changed NLP research.

However, the BERT model still has limits, as it requires huge resources to be pre-trained. In the future, we expect more efficient architectures, lighter models for deployment, and a stronger focus on ethics and fairness. These trends will guide the next stages of sequence modeling research.

References

[1] Jurafsky, D., & Martin, J. H. (2019). Speech and language processing. Stanford University.

[2] Jordan, M. I. (1997). Serial order: A parallel distributed processing approach. Advances in Psychology, 471–495.

[3] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780.

[4] Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP).

[5] Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural Machine Translation by Jointly Learning to Align and Translate. arXiv: 1409.0473.

[6] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention Is All You Need. Advances in Neural Information Processing Systems, 30, 5998–6008.

[7] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (pp. 4171–4186).

[8] Elman, J. (1990). Finding structure in time. Cognitive Science, 14(2), 179–211.

[9] Mikolov, T., Karafiát, M., Burget, L., Černocký, J., & Khudanpur, S. (2010). Recurrent neural network-based language model. Interspeech 2010.

[10] Graves, A., Mohamed, A., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, 6645–6649.

[11] 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.

[12] Sundermeyer, M., Schlüter, R., & Ney, H. (2012). LSTM neural networks for language modeling. Interspeech 2012.

[13] Ma, X., Tao, Z., Wang, Y., Yu, H., & Wang, Y. (2015). Long short-term memory neural network for traffic speed prediction using Remote Microwave Sensor Data. Transportation Research Part C: Emerging Technologies, 54, 187–197.

[14] Chung, Junyoung & Gulcehre, Caglar & Cho, KyungHyun & Bengio, Y.. (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling.

[15] Cahuantzi, R., Chen, X., & Güttel, S. (2023). A comparison of LSTM and GRU networks for learning symbolic sequences. Lecture Notes in Networks and Systems, 771–785.

[16] Jozefowicz, R., Zaremba, W., & Sutskever, I. (2015). An Empirical Exploration of Recurrent Network Architectures. In Proceedings of the 32nd International Conference on Machine Learning (pp. 2342–2350).

[17] Lim, B., & Zohren, S. (2021). Time-series forecasting with Deep Learning: A Survey. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 379(2194), 20200209.

[18] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R., & Bengio, Y. (2015). Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In Proceedings of the 32nd International Conference on Machine Learning (pp. 2048–2057).

[19] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. ArXiv, abs/1907.11692.

[20] Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In International Conference on Learning Representations (ICLR).