Sequence to Sequence Learning with Neural Networks

Murat Tokgoz
2021242
murat.tokgoz@bahcesehir.edu.tr

Abstract

Deep Neural Networks (DNNs) are powerful learning models that have done well on hard learning tasks. DNNs work well when there are a lot of labelled training sets, but they can't be used to map one sequence to another. In this paper, we show a general, end-to-end method for learning sequences that makes few assumptions about how the sequences are put together. Our method uses a multilayered Gated Recurrent Unit (GRU) to map the input sequence to a vector with a fixed number of dimensions, and then decodes the target sequence from the vector using another deep GRU. Our main result is that on an English to German translation task from the Multi30k dataset, the translations produced by the GRU achieve a BLEU score 21.34 on validation data and 21.61 on test data with beam size equal to 1.

1 Introduction

Deep Neural Networks (DNNs) are exceptionally strong machine learning models that exhibit excellent performance on challenging problems such as speech recognition [1, 2] and visual object recognition [3, 4, 5, 6]. Deep Neural Networks (DNNs) are also known as convolutional neural networks. DNNs are extremely effective due to their ability to carry out arbitrary parallel computation with only a limited number of steps. The capacity of DNNs to sort N N-bit values with only two hidden layers having a size that is quadratic is a surprising illustration of the strength that DNNs possess. So, despite the fact that neural networks are related to traditional statistical models, they are capable of learning complex computations. In addition, supervised backpropagation may be used to train large DNNs if the labelled training set contains enough information to precisely determine the network's parameters. This is true even for very large networks. Hence, supervised backpropagation will find these parameters and solve the problem if there is a parameter setting of a big DNN that delivers good results.

Even though DNNs are flexible and powerful, they can only be used to solve problems whose inputs and outputs can be encoded with vectors of a fixed number of dimensions. It's a big problem because sequences whose lengths aren't known ahead of time are often the best way to describe important problems. For instance, speech recognition and machine translation are both problems that come in a certain order. Since this is the case, it is clear that a method that can learn to map sequences to sequences would be helpful.

Sequences present a challenge for deep neural networks (DNNs) because the input and output dimensionality must be fixed, which is not always possible for sequences with varying lengths. In this paper, we demonstrate that the Gated Recurrent Unit (GRU) [2] architecture can effectively solve general sequence-to-sequence problems. The basic idea is to use one GRU to read the input sequence one timestep at a time, producing a large fixed-dimensional vector representation, which is then used as input to another GRU that generates the output sequence Figure 1. The second GRU is essentially a recurrent neural network language model conditioned on the input sequence. The GRU's ability to learn on data with long-range temporal dependencies makes it a suitable choice for sequence-to-sequence problems where there is a considerable time lag between the input and output.

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

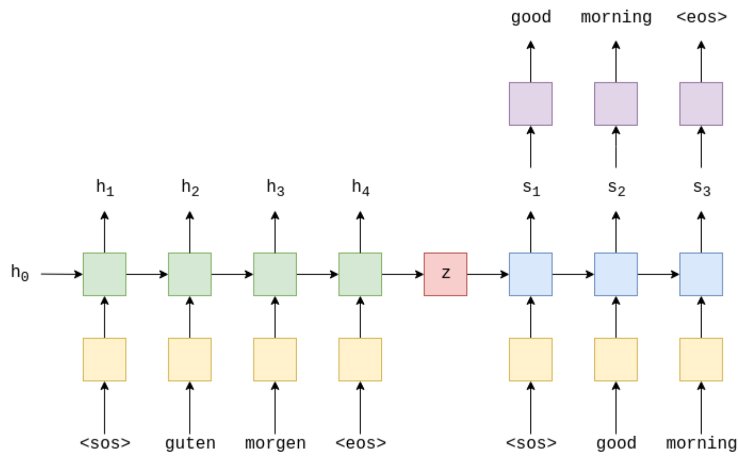

Figure 1: Our model reads an input sentence: The input sentence, "guten morgen", is passed through the embedding layer (yellow) and then input into the encoder (green).

2 Related work

A lot of work has been done on how neural networks can be used to help with machine translation. So far, the easiest and most effective way to use an RNN-Language Model (RNNLM) [11] or a Feedforward Neural Network Language Model (NNLM) [10] on an MT task is to rescore the n-best lists of a strong MT baseline, which always improves the quality of the translation.

Researchers have recently started to look into ways to add information about the source language to the NNLM. Auli et al. [7] is an example of this kind of work. They combine a NNLM with a topic model of the input sentence, which makes rescoring work better. Devlin et al. [12] used a similar method, but they added their NNLM to the decoder of an MT system and used the alignment information from the decoder to tell the NNLM which words in the input sentence were the most useful. Their plan worked very well, and it led to big improvements over their baseline.

While in [13] convert sentences to vectors using convolutional neural networks, which preserves the ordering of the words, our work is closely connected to theirs because we were the first to map the input sentence into a vector and then back to a sentence. Prior to integrating their neural network into an SMT system, [14] also mapped phrases into vectors and back using an LSTM-like RNN architecture. [9] Neural network with an attention mechanism to attempt direct translations, avoiding the poor performance on long sentences seen by [14].

3 Materials and methods

We're using the term RNN generally here, it could be any recurrent architecture, such as an *LSTM* (Long Short-Term Memory) or a *GRU* (Gated Recurrent Unit).

The EncoderLayer class defines the architecture for the encoder of the seq2seq model. In the constructor, the necessary variables are initialized, and the encoder layers are defined, including an embedding layer and a Gated Recurrent Unit (GRU) layer. The forward method takes the input sequences and their lengths, passes them through the embedding layer, and then through the GRU layer. It returns the final hidden state of the encoder, which is a tensor of size $[n_layers \times 2, batch_size, hidden_size]$, where n_layers is the number of layers in the GRU, $batch_size$ is the size of the input batch, and $hidden_size$ is the size of the hidden state.

The DecoderLayer class defines the architecture for the decoder of the seq2seq model. In the constructor, the necessary variables are initialized, and the decoder layers are defined, including an embedding layer, a GRU layer, and a fully connected (linear) layer. The forward method takes the input word index and the previous hidden state of the decoder, passes them through the embedding layer, and then through the GRU layer. It returns the logits, which are the output of the linear layer

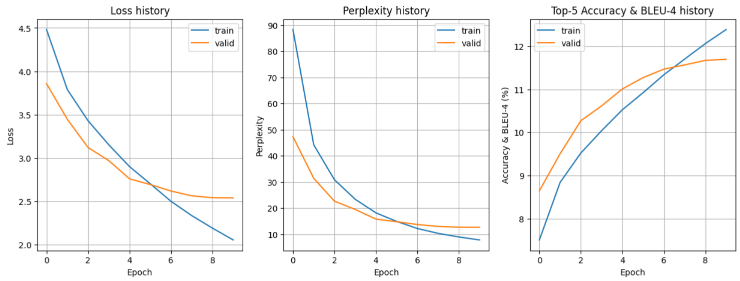

Figure 2: Losses, Perplexity and BLEU plot

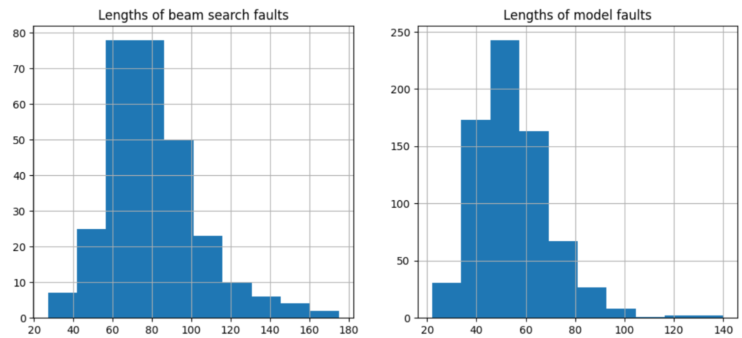

Figure 3: Distribution of lengths of the sentences

applied to the final output of the GRU layer, and the final hidden state of the decoder, which is a tensor of size $[n_layers, batch_size, hidden_size]$

4 Results and Discussion

4.1 Dataset

The dataset Multi30k was used. The dataset with 30,000 parallel English, German and French sentences, each with 12 words per sentence. 29000 number of training examples were used. Build the vocabulary for the source and target languages. The vocabulary is used to associate each unique token with an index (an integer). The vocabularies of the source and target languages are distinct.

4.2 Training the SeqToSeq Model

We define the encoder, decoder and then our SeqToSeq model, which we place on the device. Initialize the weights of model as mentioned before, the input and output dimensions are defined by the size of the vocabulary. The embedding dimesions and dropout for the encoder and decoder can be different, but the number of layers and the size of the hidden/cell states must be the same.

4.3 Experimental Results

We used the cased BLEU [8] score to evaluate the Losses, Perplexity and BLEU history as for validation data with beam size 1 the evaluation score is 21.34 and 21.61 respectively with on test data.

In Figure 3 represent first subplot shows the distribution of lengths of the sentences that were generated by a beam search algorithm and caused errors. The second subplot shows the distribution of lengths of the sentences that were generated by the model directly and caused errors. The x-axis represents the length of the sentence, and the y-axis represents the frequency of sentences with that length. The histograms help to visualize how often errors occur for sentences of different lengths, and whether there are any patterns or trends in the data. A few examples of long source sentence translations produced by the GRU alongside the ground truth and predicted translation in Figure 4.

Source: mann in einem rot-weißen fußballtrikot steht an der seitenlinie mit einem gelb-blauen fußball .
Ground truth translation: man in red and white soccer uniform stands on the field boundary lines with yellow and blue soccer ball .
Predicted translation: man in a red uniform uniform is standing with a soccer ball on the ball .

Source: leute entspannen in einem wald neben kanus .
Ground truth translation: people chilling in at forest next to canoes .
Predicted translation: people are inside a stream in the water .

Source: eine gruppe von männern in roten und schwarzen jacken sitzt auf motorrädern und wartet .
Ground truth translation: a group of men in red and black jackets waits on motorcycles .
Predicted translation: a group of men in red jackets and black jackets are sitting on stools .

Source: ein kleines kind schläft in seinem bett mit einem offenen buch auf der brust .
Ground truth translation: a young child sleeping in her bed with an open book on her chest .
Predicted translation: a young child is sleeping in a bathroom with his lap in his mouth .

Source: ein großer schwarzer pudel läuft auf dem gras mit einem spielzeug im maul .
Ground truth translation: a big black poodle running on the grass with a toy in its mouth .
Predicted translation: a large black poodle runs with a toy in its mouth .

Figure 4: A few examples of long translations produced

5 Conclusion

In this work, we demonstrated that a big deep GRU with a constrained vocabulary and essentially no issue structure assumptions can outperform a traditional SMT-based system with an infinite vocabulary on a large-scale MT challenge. Our simple GRU technique worked well on MT and should work well on other sequence learning problems with appropriate training data.

We conclude that a problem encoding with the most short-term dependencies simplifies learning. We were unable to train a normal RNN on the non-reversed translation problem, but we think it should be easy to train when the source phrases are reversed although we did not verify it experimentally.

References

[1] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition. IEEE Signal Processing Magazine, 2012.

[2] G. E. Dahl, D. Yu, L. Deng, and A. Acero. Context-dependent pre-trained deep neural networks for large vocabulary speech recognition. IEEE Transactions on Audio, Speech, and Language Processing - Special Issue on Deep Learning for Speech and Language Processing, 2012

[3] D. Ciresan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In CVPR, 2012

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In NIPS, 2012.

[5] Q.V. Le, M.A. Ranzato, R. Monga, M. Devin, K. Chen, G.S. Corrado, J. Dean, and A.Y. Ng. Building high-level features using large scale unsupervised learning. In ICML, 2012.

[6] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition.

[7] M. Auli, M. Galley, C. Quirk, and G. Zweig. Joint language and translation modeling with recurrent

[8] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu. BLEU: a method for automatic evaluation of machine translation. In ACL, 2002.

[9] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473, 2014.

[10] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A neural probabilistic language model. In Journal of Machine Learning Research, pages 1137–1155, 2003.

[11] T. Mikolov, M. Karafi ́ at, L. Burget, J. Cernock ̀ y, and S. Khudanpur. Recurrent neural network based language model. In INTERSPEECH, pages 1045–1048, 2010.

[12] J. Devlin, R. Zbib, Z. Huang, T. Lamar, R. Schwartz, and J. Makhoul. Fast and robust neural network joint models for statistical machine translation. In ACL, 2014.

[13] N. Kalchbrenner and P. Blunsom. Recurrent continuous translation models. In EMNLP, 2013.

[14] K. Cho, B. Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In Arxiv preprint arXiv:1406.1078, 2014.