What is Teacher Forcing?

A common technique in training Recurrent Neural Networks

Wanshun Wong Oct 15, 2019 · 4 min read ★

Photo by Jerry Wang on Unsplash

A lot of *Recurrent Neural Networks* in *Natural Language Processing* (e.g. in image captioning, machine translation) use *Teacher Forcing* in the training process. Despite the prevalence of *Teacher Forcing*, most articles only briefly describe how it works. For example, the TensorFlow tutorial on Neural machine translation with attention only says "*Teacher forcing* is the technique where the *target word* is passed as the *next input* to the decoder." In this article, we will go over the details of *Teacher Forcing* and answer some of the common questions.

How does Teacher Forcing work?

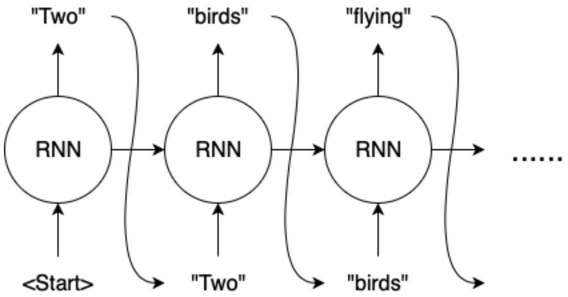
Have you ever had math exam questions that consist of multiple parts, where the answer for part (a) is needed for the calculation in part (b), answer for part (b) is needed for part (c), and so on? I always pay extra

WRITTEN BY
Wanshun Wong
Data Scientist
https://www.linkedin.com/in/wanshunwong/
Follow
421
1

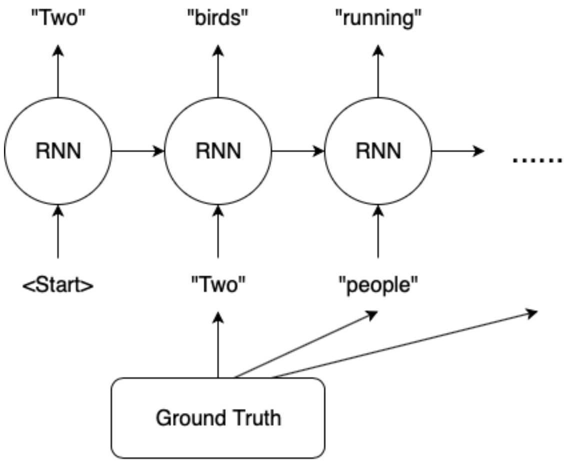
attention to these questions because if we get part (a) wrong, then all the subsequent parts will most likely be wrong as well, even though the formulas and the calculations are correct. *Teacher Forcing* remedies this as follows: After we obtain an answer for part (a), a teacher will compare our answer with the correct one, record the score for part (a), and tell us the correct answer so that we can use it for part (b).

The situation for *Recurrent Neural Networks* that output sequences is very similar. Let us assume we want to train an image captioning model, and the ground truth caption for the above image is "Two people reading a book". Our model makes a mistake in predicting the 2nd word and we have "Two" and "birds" for the 1st and 2nd prediction respectively.

- Without *Teacher Forcing*, we would feed "birds" back to our RNN to predict the 3rd word. Let's say the 3rd prediction is "flying". Even though it makes sense for our model to predict "flying" given the input is "birds", it is different from the ground truth.
- On the other hand, if we use *Teacher Forcing*, we would feed "people" to our RNN for the 3rd prediction, after computing and recording the loss for the 2nd prediction.



Without Teacher Forcing



With Teacher Forcing

Pros and Cons of Teacher Forcing

Pros:

Training with *Teacher Forcing* converges faster. At the early stages of training, the predictions of the model are very bad. If we do not use *Teacher Forcing*, the hidden states of the model will be updated by a sequence of wrong predictions, errors will accumulate, and it is difficult for the model to learn from that.

Cons:

During inference, since there is usually no ground truth available, the RNN model will need to feed its own previous prediction back to itself for the next prediction. Therefore there is a discrepancy between training and inference, and this might lead to poor model performance and instability. This is known as *Exposure Bias* in literature.

Top highlight

Implementation Example

- TensorFlow: See the "Training" session of Neural machine translation with attention
- PyTorch: See the "Training the Model" session of NLP From Scratch: Translation with a Sequence to Sequence Network and Attention

Frequently Asked Questions

Q: Since we pass the whole ground truth sequence through the RNN model, is it possible for the model to “cheat” by simply memorizing the ground truth?

A: No. At timestep *t*, the input of the model is the ground truth at timestep *t - 1*, and the hidden states of the model have been updated by ground truths from timestep *1* to *t - 2*. The model can never peek into the future.

Q: Is it necessary to update the loss at each timestep?

A: No. An alternative approach is to store the predictions at all timesteps in, say, a Python list, and then compute all the losses in one go.

Q: Is *Teacher Forcing* used outside *Natural Language Processing*?

A: Yes. It can be used in any model that output sequences, e.g. in time series forecasting.

Q: Is *Teacher Forcing* used outside *Recurrent Neural Networks*?

A: Yes. It is used in other autoregressive models such as Transformer.

Further Reading

1. Many algorithms have been invented to mitigate *Exposure Bias*, e.g. *Scheduled Sampling* [1] and *Parallel Scheduled Sampling* [3], *Professor Forcing* [5], and *Beam Search* [2], [6].
2. Result of [4] suggests that *Exposure Bias* may not be as significant as it is presumed to be.

References

1. S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks (2015), NeurIPS 2015.
2. R. Collobert, A. Hannun, and G. Synnaeve. A Fully Differentiable Beam Search Decoder (2019), ICML 2019.
3. D. Duckworth, A. Neelakantan, B. Goodrich, L. Kaiser, and S. Bengio. Parallel Scheduled Sampling (2019), arXiv.
4. T. He, J. Zhang, Z. Zhou, and J. Glass. Quantifying Exposure Bias for Neural Language Generation (2019), arXiv.
5. A. Lamb, A. Goyal, Y. Zhang, S. Zhang, A. Courville, and Y. Bengio. Professor Forcing: A New Algorithm for Training Recurrent Networks (2016), NeurIPS 2016.
6. S. Wiseman, and A. Rush. Sequence-to-Sequence Learning as Beam-Search Optimization (2016), EMNLP 2016.

Deep Learning Recurrent Neural Network Teacher Forcing Machine Learning