



LANGUAGE TRANSLATION-USING SEQUENCE TO SEQUENCE LEARNING

Arghajyoti Singha¹, A NithyaKalyani²,
Mainak Pal Chowdhury³,

^{1,2,3}Department of Computer Science and Engineering,
SRM Institute of Science and Technology, Chennai

July 24, 2018

Abstract

The paper deals with the implementation on the recent studies made on using DNN (Deep Neural Networks) to map sequences to sequences and implementing the same to automate Language translation process without the requirement of expert attention or statistical model. The paper deals with the demonstration of production ready end-to-end NMT (Neural Machine Translation) model using multi-layered Bidirectional RNN (Recurrent Neural Network), Long Short-Term Memory (LSTM) to encode the source statement to a sequence of numbers and the train another such RNN to decode the same to the target language statement. Our initial attempt is to build a vanilla NMT model and then further build the best possible NMT model using TensorFlow best practices. We reach this position by: i) Using the recent decoder / attention wrapper API, TensorFlow 1.2 data iterator, ii) Incorporating Bidirectional LSTM, iii) Making use of TensorFlow methods such as batching, bucketing, beam search as well as deploying the model to multiple GPUs using GNMT attention. The main test bed is on 3 language conversions, English-Vietnamese, German-English and English-French.

1 Introduction

In a human involved translation, it is common practice to go through the sentence and produce a translation as a whole. While automating the process through a machine, we commonly employ maxim-dependent demarcation of a sentence and take various lumps of these and translate maxim by maxim[8][9]. This model has shown a higher level of accuracy for an automated procedure but mostly results in a wrongly translated sentence. Thus to properly imitate the human-like translation process into a machine automated one, a Neural Machine Translation^{[10][18][19]} model is employed. The exact process in which an NMT models works is by creating a median layer called a 'thought' vector which depicts the sentence provided to it in a collection of numbers. This process is typically called encoding. This vector is then subjected to another process, i.e., decoding, where the sequence of numbers is again mapped back to a sentence of another language, thus providing a translation. This is usually called an "encoder-decoder architecture" ^{[1][5]} (Figure 1).

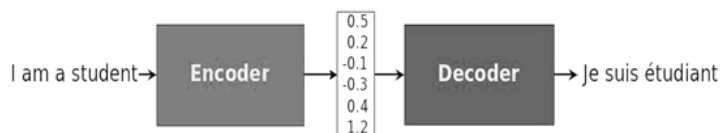


Figure 1. Encoder Decoder Architecture

These models can actually remember a long range of data thus giving correct translations like humans. They are evidently better than maxim-based models. The NMT model may employ various architectures but most common one is to use an RNN (Recurrent Neural Network)^[7] to build the encoder and decoder.

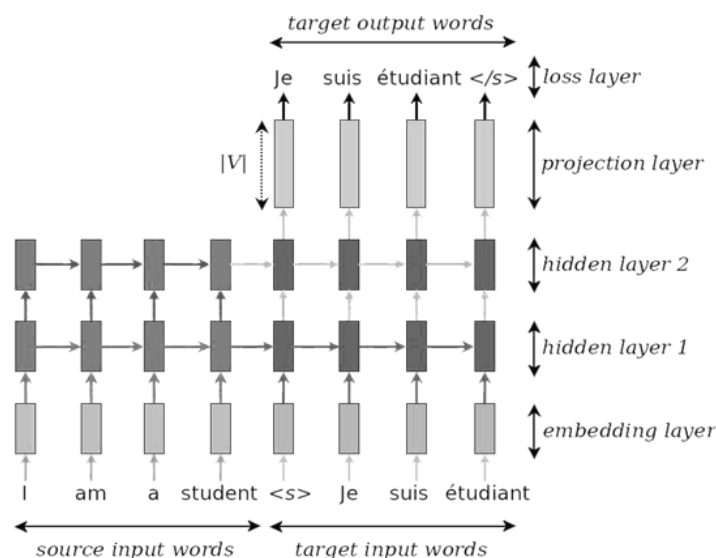


Figure 2. Sequence to Sequence Architecture

Here, we consider an example where we translate an English statement to French. The internal architecture block diagram (Figure 2) explains the same. There are various types of RNN that can be used for the purpose. In this model, a RNN called Long Short-term Memory (LSTM)^[16] has been employed. These RNNs specialize in gathering long range patterns of data thus making them perfect for Language Translation purposes.

2 Embedding

For the reason that the nature of the language is unpredictable, analogous translation of the word chunks are mapped for every word distinctively. For the aforementioned idea we have a vocabulary. Let's reckon a vocabulary of size V . The most habitually used V words are pursued and the other words are placed the equivalent embedding weight. The other words are renewed into tokens. Embedding weights can be cultured for the duration of training or we can pre initialize them with previously trained word representations.^{[2][3]}

3 Encoding

Firstly the word segments are extracted which are afterwards given as input into two RNN with multiple layer. Previous one is used for encoding^[5], which is amending the source language and the later one is used decoding, working for translating the required to the target language. Encoder RNN^[1] will initiate from a NULL vector. Both the RNNs can own idiosyncratic constraints, but typically rake-off equivalent weights^{[11][12]}.

4 Decoding

The decoder^{[5][6]} has a requirement to contact the source information in sequence. So it is being fed, the most modern branded state of the encoder. In figure 2 we are feeding the concealed condition of the encoder at source word "student". The chief element of the decoder is the Basic Decoder entity which receives the decoder unit and the encoder condition of the previous iteration.

5 Gradient Computation and Optimization

A significant procedure in RNN training is gradient cutting. For this we make use of the international rule. Then we decide on the optimizer. Adam Optimizer^[4] is a well-liked preference. A learning tempo is required to be selected. We lay down this denomination in the assortment of 0.0001 to 0.001. This denomination is prepared to diminish as the training progresses. For the purpose of this project we have utilized a gradient optimizer with a learning curve that steadily diminishes. Taking $L(t)$ as Loss, $w(t)$ as gradient is obtained by:

$$\begin{aligned}
m_w^{(t+1)} &\leftarrow \beta_1 m_w^{(t)} + (1 - \beta_1) \nabla_w L^{(t)} \\
v_w^{(t+1)} &\leftarrow \beta_2 v_w^{(t)} + (1 - \beta_2) (\nabla_w L^{(t)})^2 \\
\hat{m}_w &= \frac{m_w^{(t+1)}}{1 - \beta_1^t} \\
\hat{v}_w &= \frac{v_w^{(t+1)}}{1 - \beta_2^t} \\
w^{(t+1)} &\leftarrow w^{(t)} - \eta \frac{\hat{m}_w}{\sqrt{\hat{v}_w} + \epsilon}
\end{aligned}$$

6 Loss

The technique to gauge the loss is by taking the negative log probability and the remaining sum of squares of regression and classification[21]. Our point is to curtail the loss function's quantity with regard to the parameters like the attention weights. We can diminish the loss function by optimization. We can presume that the loss will lessen with each iteration. Loss is computed when the model is not being trained. Loss is the metric which advises us how suitably or deficiently the model performs.

$$\begin{aligned}
h_t &= \sigma(W^{hh} h_{t-1} + W^{hx} x_t) \\
y_t &= \text{softmax}(W^S h_t) \\
\text{Loss Function at a step } t &= J^t(\theta) = - \sum_{j=1}^K t_{t,j} \log y_{t,j}
\end{aligned}$$

7 Attention Mechanism

The NMT model persevere in duplicating human-like translation process, thus making it more and more difficult in fine tuning for grater nuances of the language. One such action that model requires is the ability to provide attention to the encoding process

for further lucid decoding process. Humans, while translating a statement do not read the root statement once and keep translating. Rather they keep going back and forward to the root sentence. Thus this behavior explains that the encoding and the decoding process cannot be kept atomic and alienated of the others presence. Attention mechanism (Figure 3) makes use of various tools and criteria to provide a machine automated translation model the ability to bring together the encoding and decoding process. The main algorithm of this functionality is to calculate two more sequences of numbers called Context and Attention vectors during the encoding process. This results to a more informed decoding process which makes use of these vectors and Attention weights to clear volatility of language in sheer nuances.

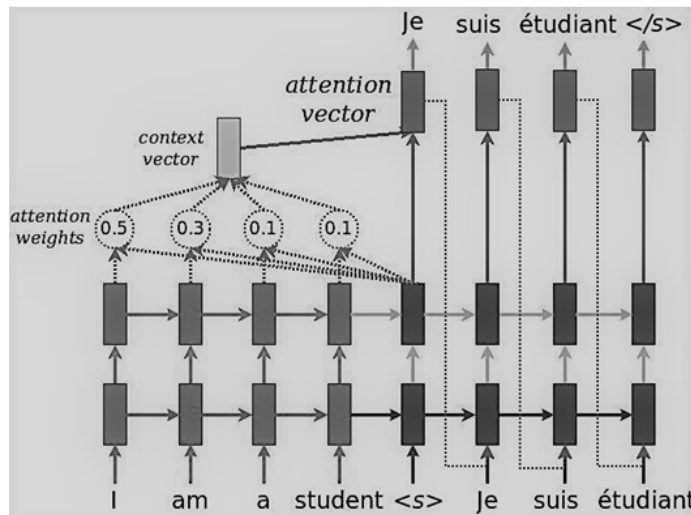


Figure 3. Architecture for Attention Mechanism

$$\alpha_{ts} = \frac{\exp(\text{score}(\mathbf{h}_t, \bar{\mathbf{h}}_s))}{\sum_{s'=1}^S \exp(\text{score}(\mathbf{h}_t, \bar{\mathbf{h}}_{s'}))} \quad [\text{Attention weights}] \quad (1)$$

$$\mathbf{c}_t = \sum_s \alpha_{ts} \bar{\mathbf{h}}_s \quad [\text{Context vector}] \quad (2)$$

$$\mathbf{a}_t = f(\mathbf{c}_t, \mathbf{h}_t) = \tanh(\mathbf{W}_c[\mathbf{c}_t; \mathbf{h}_t]) \quad [\text{Attention vector}] \quad (3)$$

Figure. depicts the visualisation of the attention model

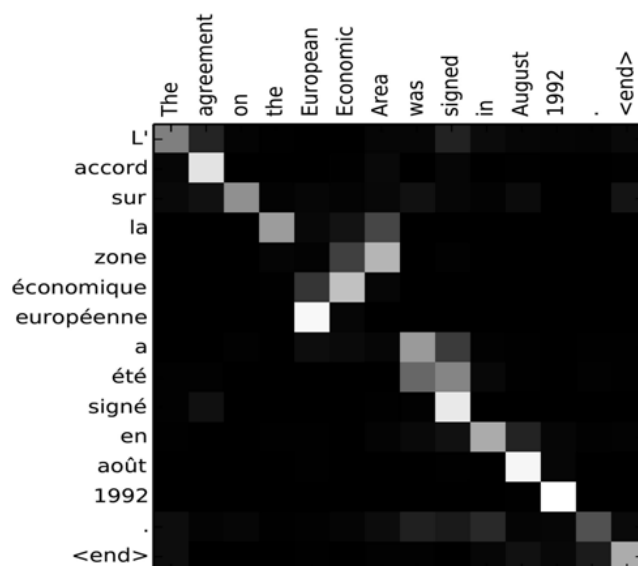


Figure 4. Visualization of attention mechanism

8 Beam Search

In a more traditional NMT approach, we make use of Greedy Search[28] process to find relevant vocabulary maps in the decoding process. This process looks for heuristics to find the best possible solution in the local scope at each step. This does not necessarily produce an optimized solution. Thus to counter this and obtain a much preferred solution Beam Search[31] is employed. In Beam Search, best possible candidates providing partial solutions are stored in each stage. This reduces memory constraints and develops better and optimized solutions. It also reduces step time of decoding.

9 Results and Discussion

As mentioned before, the test beds for the above NMT model are for the following translation:

1. English to Vietnamese:

This translation model is the smallest dataset (IWSLT English Vietnamese); 133KB. 2. German to English: This translation model is the moderate dataset (WMT German-English); 4.5MB. 3. English to French:

This translation model is the complex dataset (TEDx Parallel Corpus French-English); 6.2MB.

All such model had been trained on the following parameters:

- Number of training steps: 12000
- Steps per statistics: 100
- Number of LSTM Layers: 2
- Number of LSTM units: 128
- Dropout: 0.2.

The models for the following dataset has been trained for 40 hours, 235 hours and 210 hours respectively. Each dataset thus trained on this time frame resulted in BLEU[24] scores of 29.9, 24.4 and 18.2 respectively.

10 Related Work

Research is being carried out on Neural Machine Translation in Stanford University. They have released their code base which fabricates state of the art solutions. Their NMT support English-German and English-Czech. Open NMT is a MIT initiative for neural machine studies. Open NMT was launched in December 2016. Currently Open NMT has three implementations,

- Open NMT-lua: This is the original project and is developed with Lua Torch.
- OpenNMT-py: This project is an OpenNMT-lua clone using the more modern PyTorch.
- OpenNMT-tf: This is a TensorFlow alternative to the Open NMGT-lua

A study named Massive Exploration of Neural Machine Translation Architectures ” talks about how NMT can be used for processing data sets and finding long distance relations among the word chunks.

11 Conclusion

The paper chalks a proper procedure to implementing a production ready language translation model based on Neural Machine Translation Model scoring significant BLEU scores of 29.9. The model can be improved further with longer hours of training and providing various datasets. The model is also capable of multi lingual transcription. However, the model is resource extensive, i.e., requires heavy GPU access to reduce training time frame.

References

- [1] M. Auli, M. Galley, C. Quirk, and G. Zweig. Joint language and translation modeling with recurrent neural networks. In EMNLP, 2013.
- [2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473, 2014.
- [3] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A neural probabilistic language model. In Journal of Machine Learning Research, pages 11371155, 2003.
- [4] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks, 5(2):157166, 1994.
- [5] 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.
- [6] D. Ciresan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In CVPR, 2012.
- [7] 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.

- [8] 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.
- [9] Nadir Durrani, Barry Haddow, Philipp Koehn, and Kenneth Heafield. Edinburghs phrase-based machine translation systems for wmt-14. In WMT, 2014.
- [10] A. Graves. Generating sequences with recurrent neural networks. In Arxiv preprint arXiv:1308.0850, 2013.
- [11] A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In ICML, 2006.
- [12] K. M. Hermann and P. Blunsom. Multilingual distributed representations without word alignment. In ICLR, 2014.
- [13] 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.
- [14] S. Hochreiter. Untersuchungen zu dynamischen neuronalen netzen. Masters thesis, Institut für Informatik, Technische Universität, München, 1991.
- [15] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001.
- [16] S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural Computation, 1997.
- [17] S. Hochreiter and J. Schmidhuber. LSTM can solve hard long time lag problems. 1997.
- [18] N. Kalchbrenner and P. Blunsom. Recurrent continuous translation models. In EMNLP, 2013.
- [19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In NIPS, 2012.

- [20] 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.
- [21] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 1998.
- [22] T. Mikolov. Statistical Language Models based on Neural Networks. PhD thesis, Brno University of Technology, 2012.
- [23] T. Mikolov, M. Karafiat, L. Burget, J. Cernocký, and S. Khudanpur. Recurrent neural network based ‘ language model. In INTERSPEECH, pages 10451048, 2010.
- [24] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu. BLEU: a method for automatic evaluation of machine translation. In ACL, 2002.
- [25] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. arXiv preprint arXiv:1211.5063, 2012.
- [26] J. Pouget-Abadie, D. Bahdanau, B. van Merriënboer, K. Cho, and Y. Bengio. Overcoming the curse of sentence length for neural machine translation using automatic segmentation. arXiv preprint arXiv:1409.1257, 2014.
- [27] A. Razborov. On small depth threshold circuits. In Proc. 3rd Scandinavian Workshop on Algorithm Theory, 1992.
- [27] D. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533536, 1986.
- [28] H.Schwenk.University le mans. http://www-lium.univ-le-mans.fr/schwenk/cslm_joint_paper/,2014. [Online; accessed 03-September-2014].
- [29] M. Sundermeyer, R. Schluter, and H. Ney. LSTM neural networks for language modeling. In INTERSPEECH, 2010.
- [30] P. Werbos. Backpropagation through time: what it does and how to do it. Proceedings of IEEE, 1990.

