Language Engineering, DD2418

Project report

Machine Translation with Deep Learning

Evi Gogoulou, Adrianna Janik

December 16, 2018

Contents

1 Introduction

1.1 Main objectives and scope of the project

Our major goals in this project were:

- to re-implement the Encoder-Decoder model for the task of Neural Machine Translation (NMT), as stated in the article [10] and adjust it to available resources
- to apply the reference model to the Machine Translation task from English to German
- to identify key limitations of the model
- to configure and monitor the behaviour of learning algorithms for the model
- to recognize risks associated with complex NMT models and minimize them for robust learning

1.2 Background

The task of Machine Translation (MT) can be defined as building a system for automatic translation from a source language to a target language. The traditional approach to this problem is Statistical Machine Translation (SMT), which tries to build a probabilistic model of correct translations $P(y|x)$ where x is the source sentence and y the target sentence. One way of identifying this conditional distribution, widely used in SMT, is using the Expectation Maximization Algorithm [4]. Recent advances in Deep Learning have lead to the emerge of a new approach to the MT task, which is called Neural Machine Translation (MT) [7][10]. Unlike the SMT methods, the goal of NMT is to fit a neural network model to maximize the conditional distribution learned directly from a set of bilingual sentence pairs. Once the conditional distribution has been learned, the correct translation corresponds to searching for the target sentence that maximizes the conditional probability given an source sentence.

A typical neural network architecture for the NMT task is Encoder-Decoder. The Encoder maps the source word sequence to a hidden representation, which is subsequently mapped to a target word sequence by the Decoder. One family of Neural Networks which is able learn a conditional distribution from a set of input sequences is Recurrent Neural Networks (RNNs). An RNN consists of a hidden state $\mathbf{h}$ and an optional output $\mathbf{y}$ which is computed upon an input sequence $\mathbf{x}$ of variable length. At each timestep, the hidden state $\mathbf{h}$ gets updated according to the formula:

$$\mathbf{h}_t = f(\mathbf{h}_{t-1}, x_t) \tag{1}$$

where f is a non-linear activation function. By training an RNN to predict the next input symbol in a sequence, the RNN output corresponds to the conditional distribution over the input sequence. The main disadvantage of RNN units is that they cannot handle long term dependencies in the input sequence. An alternative unit that overcomes this problem is Long Short-Term Memory (LSTM) [6].

According to the basic RNN Encoder-Decoder, proposed in [5], the Encoder maps the input sequence $x = x_1, ..., x_T$ to a context vector $\mathbf{c}$ which corresponds to the sequence of generated hidden states. That is:

$$\mathbf{c} = q(\mathbf{h}_1, ..., \mathbf{h}_T) \tag{2}$$

where q is a nonlinear function and $\mathbf{h_i}$ the hidden state generated by the RNN unit at timestep i according to 1. The context vector $\mathbf{c}$ can be seen as a summary of the input sequence x. Unlike the RNN-Encoder, both the output $\mathbf{y_t}$ and hidden state $\mathbf{h_t}$ of the RNN-Decoder are conditioned both on $\mathbf{y_{t-1}}$ and the context vector $\mathbf{c}$. So now the hidden state $\mathbf{h_t}$ is given by the formula:

$$\mathbf{h}_t = f(\mathbf{h}_{t-1}, y_{t-1}, \mathbf{c}) \tag{3}$$

Similarly, the conditional distribution, generated in the output $\mathbf{y}$ of the RNN-Decoder is given by the formula:

$$P(y_t|y_{t-1}, .., y_1, \mathbf{c}) = g(\mathbf{h_t}, y_{t-1}, \mathbf{c}) \tag{4}$$

for given activation function f, g. Regarding the training objective, both Encoder and Decoder are jointly trained to maximize the conditional likelihood:

$$max_\theta \frac{1}{N} \sum_{n=1}^{N} \log p_\theta(y_n|x_n) \tag{5}$$

where N is the number of pairs of bilingual sentences and θ is the model parameters.

One of the main design choices in the RNN Encoder-Decoder architecture concerns the choice the RNN network used, which is encoded with f in the formulation above. The LSTM choice is quite popular,[10], comparing to the basic RNN unit, since it has better performance in longer input sequences. The choice of q function is also critical. In the simple case, presented in [5].

$$\mathbf{c} = q(\mathbf{h}_1, ..., \mathbf{h}_T) = \mathbf{h}_T \tag{6}$$

This choice of q leads to context vectors c of fixed size, which is problematic for long input sequences. Alternative approaches presented in [3] and [8] consider context vectors of the form:

$$\mathbf{c} = \sum_{j=1}^{T} a_{ij} h_j \tag{7}$$

In this case, a_{ij} corresponds to the alignment score between source word j and target word i, which acts as a weight on the hidden state $\mathbf{h_j}$. The set of alignment scores a is modelled as a feedforward neural network layer, jointly trained together with the Encoder and Decoder. Intuitively, this allows the decoder to focus on specific words in the source sentence, which are aligned to the current target word. This strategy leads to improved translation performance in long sentences.

2 Methods

2.1 Architecture

Our method follows the work presented by Sutskever et al [10]. The reference architecture consists of an LSTM Encoder-Decoder model with 4 layers in each LSTM. A softmax layer of 80000 nodes has been added in the output of the decoder, in order to generate the conditional probabilities of correct translation. Regarding the data representation, word embeddings with 1000 dimensions are used for all the words in the source and target vocabulary. An additional characteristic of the reference architecture is that the source sentences are given as input to the encoder in reverse order, with the goal of reducing the distance between source and target words which are aligned but the sentences are long. This trick can be seen as an alternative approach to the alignment scores described in section 1.2.

The main limitation identified in the reference method is the large number of parameters, which increases the demand for hours of training the model. As stated in [10] (Section 3.5), the model was trained for 10 days on a 8-GPU machine. Since our project had serious restrictions both in time and computational resources, we decided to use a restricted version of the reference architecture in terms of number of parameters. The detailed description of the parameters used is listed in Section **??**. Our architecture is illustrated in Figure 1:

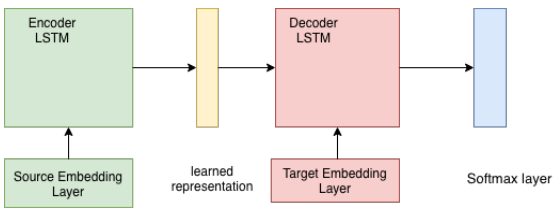

Figure 1: Network Architecture

2.2 Dataset

The reference architecture presented in [10] was tested on a subset of 12M pairs of sentences from the WMT'14 English-French dataset. The goal of our project was to evaluate the reference architecture on the WMT'15 English-German dataset [1], because of our wrong assumption that the source of our data should be correct, we fell into a trap of trying to train model on corrupted dataset. This discovery that unfortunately got revealed after presentation made us change the dataset to a News Commentary data set, that contained pairs of headlines in English and German, properly aligned with each other. It was also coming from Workshop on Statistical Machine Translation 2014 and it can be found here [2]. Example data from the dataset can be seen in Table 1

	English	German
1.	$10,000 Gold?	Steigt Gold auf 10.000 Dollar?
2.	SAN FRANCISCO - It has never been easy to have a rational conversation about the value of gold.	SAN FRANCISCO - Es war noch nie leicht, ein rationales Gespräch über den Wert von Gold zu führen.
3.	Lately, with gold prices up more than 300% over the last decade, it is harder than ever.	In letzter Zeit allerdings ist dies schwieriger denn je, ist doch der Goldpreis im letzten Jahrzehnt um über 300 Prozent angestiegen.

Table 1: Pairs of sentences

We also had a look at the most frequent terms occurring among the headlines from news dataset. To do that we plotted frequency distribution plot as in Figure 2 and from that plot we learned that most of the articles that the headlines were taken from was about economy and politics, the most frequent terms were: European, countries, political, global etc...

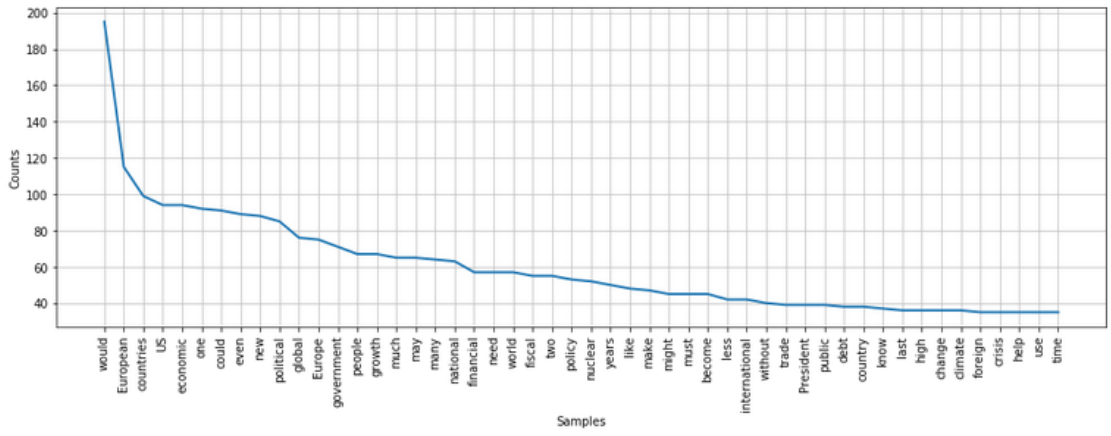

Figure 2: Frequency distribution plot

2.3 Evaluation Metrics

The most widely used [10][8][3] metric for evaluation of the translation quality is the Bilingual Evaluation Understanding (BLEU) score [9]. Given the candidate translation and the reference translation, the BLUE score counts the number of matches between all possible n-grams in the candidate translation and the reference translation. High value of BLEU score is an indicator of good trnanslation quality. Following the reference paper [10], we used the BLEU score to evaluate the quality of sentence-level translation.

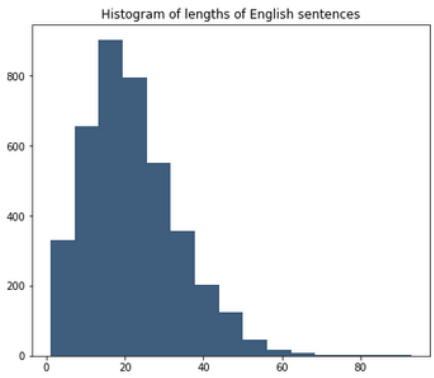

Figure 3: English sentences

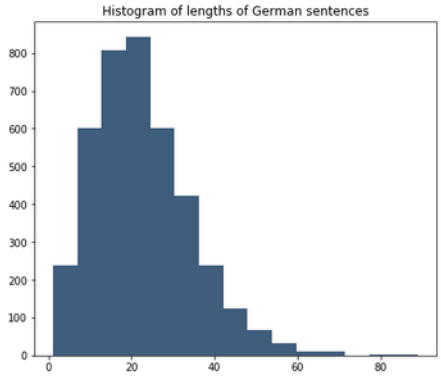

Figure 4: German sentences

Figure 5: Length distribution

3 Implementation

3.1 Training Details

In this project we set a goal to re-implement the model presented in [10]. To do that we identified parameters as well as architecture and the learning structure that the authors used in the article and then we tried to iteratively adjust the model to the resources that were available during our project.

parameter	original model	our model
size of embeddings	1000	300
# of cells in LSTM	1000	300
# of LSTM layers	4	1
optimizer	SGD	RMSPROP
epochs	7,5	260
sentence pairs	12M	2000
training time	10 days	1 night
# of GPUs	8	1

Table 2: Comparison of the reference model and our model

One improvement that we propose regarding the selection of training sentences is choosing only sentences that their length falls between 5th and 95th percentile. In other words, we decided to exclude outliers in terms of sentence length. Our assumption was that this improvement would lead to a more robust model, considering that fact that our context vector $\mathbf{c}$ has fixed size (section 1.2). To visualize this we can have a look at the box plots of both distributions in Figure 8.

3.2 Tools

A list of the tools and libraries used for this project is the following:

- Jupyter Python Notebook with Python 3.7.
- Numpy,
- Pandas,
- Scikit-learn,
- Keras,
- Tensorboard
- Google Cloud Platform, VM with 20Gb disk space, 8Gb RAM, 1GPU

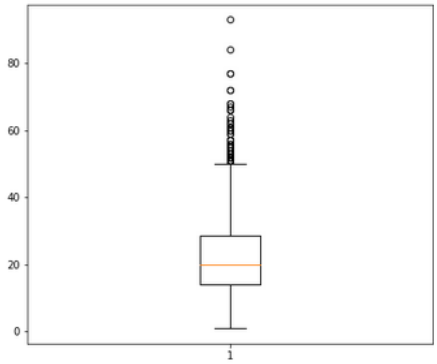

Figure 6: English sentences

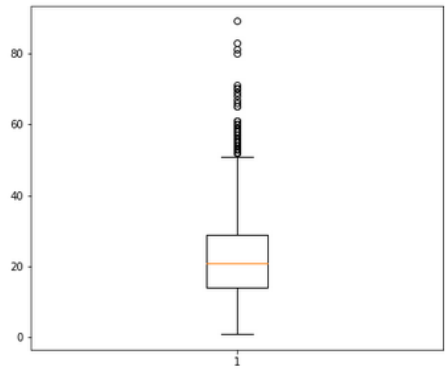

Figure 7: German sentences

Figure 8: Length distribution

4 Results

Our experiments resulted in a very low BLEU score, as observed below, when the BLEU score achieved with the model presented in [10] is 34.5. But despite that we could observe process of learning the language model during the training, more on that in discussion. We computed the BLEU score on the training set of sentence pairs during the training at the end of each epoch. We are aware that our model has almost zero generalization power at the moment, a fact that was expected given the low amount of training data we used. An example of translation achieved with our model is illustrated below:

```
Epoch 250
loss: 1.9015 - acc: 0.1387 - val_loss: 4.0334 - val_acc: 0.0437
----------------------------------------------------------------------
Input sentence: because it will take time to collect revenues from the regional
vat COMMA the eu should provide financial help during the transition
Decoded sentence:  die des bzw des aufsteigen anzukurbeln _END
BLEU score: 1.4147351699132998e-231
----------------------------------------------------------------------
Input sentence: this COMMA in turn COMMA created once again the kind of winwin
situation that is so important for future relations between the eu and russia
Decoded sentence:  die des bzw des aufsteigen anzukurbeln _END
BLEU score: 1.4147351699132998e-231
----------------------------------------------------------------------
Input sentence: many germans today rightly feel that any system of fiscal
transfers will morph into a permanent feeding tube COMMA much the way that
northern italy has been propping up southern italy for the last century
Decoded sentence:  großmachtansprüche aufzustehen _END
BLEU score: 1.4637115948630222e-231

----------------------------------------------------------------------

Epoch 251
loss: 1.8983 - acc: 0.1386 - val_loss: 4.0333 - val_acc: 0.0441
----------------------------------------------------------------------
Input sentence: but it makes far more sense to use the force of markets - the power
of incentives - than to rely on goodwill COMMA especially when it comes to oil
companies that regard their sole objective as maximizing profits COMMA regardless
of the cost to others
Decoded sentence:  die der des bzw des einsatzgüter krieges krieges anzukurbeln
grundlagen gegenwärtig die für die die der der des aneinander des entlang
```

```
gleichwohl auseinandersetzen nationalistischen leistungen erfolgreiches fragte
krieges reihe multilateralismus kostennutzenbasis ray huang die der der der des
aneinander des entlang erm erscheinen übernimmt versprach und die märkte - die
BLEU score: 9.269709363626008e-232
----------------------------------------------------------------------------
Input sentence: regional vat rates might be increased slightly if expenditure
restraint is not sufficient to offset the loss of tariff revenues due to the customs union
Decoded sentence:  die des bzw des aufsteigen anzukurbeln _END
BLEU score: 1.3682868820983658e-231
----------------------------------------------------------------------------
Input sentence: the feds judgment appeared to be that it was largely if not completely
powerless it had done all that it could COMMA and the levers of monetary policy
were no longer strongly connected to determining the level of economic activity
Decoded sentence:  die der des bzw des einsatzgüter krieges _END
BLEU score: 1.3020095722934844e-231
```

The plots of accuracy and loss in training and validation set in 400 epochs of training are illustrated in Figure 9.

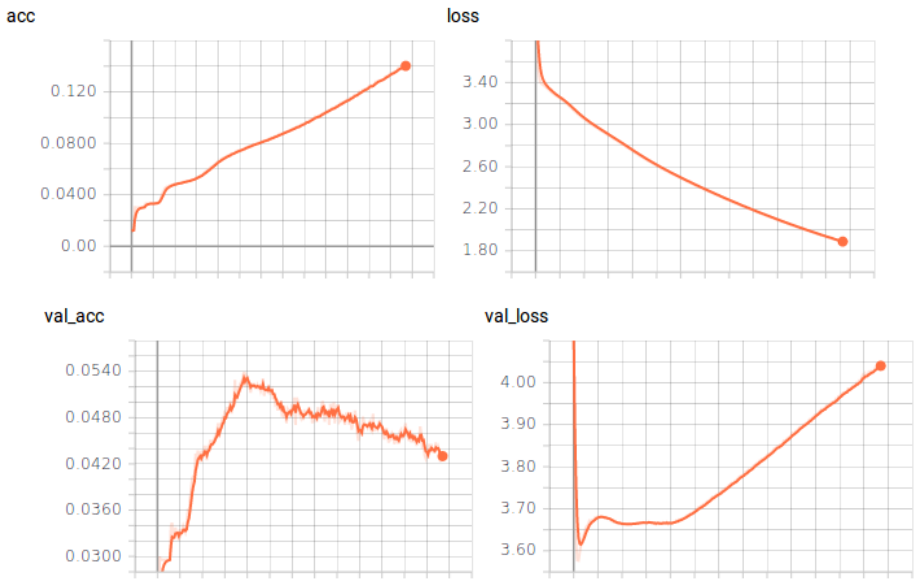

Figure 9: Accuracy and loss on training set (up) and training set (down)

5 Discussion

First thing that we discovered during this project was the difficulty of tuning parameters of the network to get any results. Our first approach was to implement the model exactly like it was presented in the original work. When we tried to train the network we encountered first problem, we could not create such large softmax layer because of memory issue. Which was not a surprise since we did not have the same hardware to train our network on. Our solution to that problem was to limit the data, with 8GB of RAM we could fit around 2000 sentences. For us the bottleneck was the three dimensional output array with the size of target vocabulary as one dimension. If we do a simple math for that, given vocabulary of 10000, maximum sentence size of 50 and input data size of 2000, we get array with 1000000000 cells, if each cell is float32 witch occupies space of 32 bits, we would need 32000000000 bits of memory witch is 4Gb. This size of an array is still within our reach to process but we had to leave it this way as we wanted to restore model after training and another constraint showed up with serialization methods for our particular implementation. Consequently, the number of parameters of our model should be definitely smaller comparing to

the number of parameters of the reference model. From 4 layer LSTM in encoder and decoder we moved to 1 layer in each of them. We also limited the size of embeddings to 50.

To conclude it is not at all trivial to re-implement model from scientific paper and adapt it to available resources. There are many constraints that needs to be taken into account: available time, RAM, disk space, data, processor speed and margin for human errors. During first few epochs predicted sentences looked very similarly, each word was predicted to be "und". After around 20 epochs network started to output COMMA tags instead, then it started to become more interesting when network put certain words together like: "der EU" or "der landes" it was at around 35 epoch. Then we observed that the language model started to learn some other rules between words like " für der eu in den finanzsektor", that was observed at 260 epoch. So although from provided charts 9 the loss and accuracy on training set looks promising when we look at the validation set it looks very unstable. We suspect that this is because of complexity of the problem and small size of data set all together.

References

[1] Neural machine translation. https://nlp.stanford.edu/projects/nmt/. Accessed December 15, 2018.

[2] Wmt' 14 shared task: Machine translation. www.statmt.org/wmt14/translation-task.html. Accessed December 15, 2018.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[4] Peter F Brown, Vincent J Della Pietra, Stephen A Della Pietra, and Robert L Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311, 1993.

[5] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[6] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[7] Nal Kalchbrenner and Phil Blunsom. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, 2013.

[8] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

[9] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

[10] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.