Natural Language Understanding: Assignment 2 Report

Kapil Pathak
CSA, IISc Bangalore
kapilpathak@iisc.ac.in

Abstract

This documents contains a results and possible explainations for the second assignment given in the course Natural Language Understanding. The assignment was related to Nueral Machine Translation with LSTM cells including various attention models.

1 Introduction

Before deep learning era, the area of machine translation was largely dominated by statistical machine translation(Koehn, 2017) methods which leverage the statistical information of given corpora to understand one-to-one correspondence and alignment among the words and sentences. But once various deep learning models including LSTM cells and GRU cells gave the better accuracy with minimum efforts, the statistical methods started to fall behind. In this assignment, a 3-layered LSTM as well as GRU models have been studied and used for the translation of English sentences to German and Hindi sentences. The performance of the translation is evalueted by sentence level BLEU score (Papineni et al., 2002)

2 Neural Machine Translation

The neural machine translation system (NMT) consists of an encoder-decoder architecture with one or more layers of RNN cells such as LSTM or GRU units. The sentences are encoded into the embedding of fixed dimension. Each word in the sentence is given as an input to the each RNN cell. Recurrent Neural Network (RNN) is used for various NLP tasks as they encode sequential information such as time series, sentences, speech etc. While training, there can be a problem of vanishing or exploding gradients while backpropagating the error. But these problems can be dealt separately.

2.1 Long Short Term Memory

Long Short Term Memory (LSTM) cells consists of three gates such as input gate, output gate and forget gate.Apart from these gates, LSTM cell takes an input from previous cell, retains a hidden state. The diagram of LSTM cells is given in figure 1.

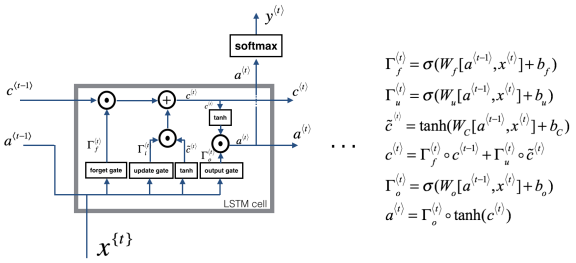

Figure 1: A schematic diagram of LSTM cells taken from http://wiki.hacksmeta.com/machinelearning/deeplearning/rnn-basics.html

2.2 Encoder Architecture

In NMT model, an encoder consists of a series of RNN cells in which each cell takes an input from each word embedding in the sentence as well has previous cell's hidden state. For the first cell, the hidden state is initialized with xavier uniform initialization available in pytorch. In this way, each RNN cell's output is dependent on the current input word embedding as well as the hidden state of previous cell (In case of unidirectional). For bidirectional RNN, the output of the each cell is also dependent on the future words in the sentence.

Here from the implementation point of view, each sentence won't be of the same length. Hence the corpus is filtered out such that left over sentences have lengths less than or equal to some maximum length which is an hyperparameter. In

the current implementation, maximum length is kept 25. For smaller sentences, the sentences are padded with symbol '/PAD'. A batch of sentences are passed into the encoder and then decoded with decoder. Total number of layers are kept 4 as indicated in the paper (Britz et al., 2017). In this paper, the authors have suggested to keep the number of layers in between 2 to 4 for better results and speed.

2.3 Decoder Architecture

Similar to an encoder architecture. a decoder architecture also consists of some series of RNN cells with specified number of layers. While decoding, an attention mechanism is used to get the words on which the decoder needs to give more weight to predict the current the word given the previous words. For each time steps while decoding, the attention weights are calculated with methods such as additive attention (Bahdanau et al., 2015), multiplicative attention (Luong et al., 2015), Key-value attention (Liu and Lapata, 2017) etc. These mechanisms have been discussed in the next section.

2.4 Attention Mechanism

Basic attention mechanism as well as its motive have been discussed earlier. But in various attention schemes, the formulae in which the attention weights are calculated are different. For brevity, various attention weights formulae are given as follows. Here s_j and h_i are encoder state and current hidden state respectively.

- Additive attention :

$$\alpha_{i,j} = v_p tanh(W_p[s_j; h_i])$$

- Multiplicative attention:

$$\alpha_{i,j} = h_i^T W_p s_j$$

- Key value pair attention:

$$v_i = tanh(W_1 s_i), v_j = tanh(W_1 h_j)$$

$$\alpha_{i,j} = v_j^T W_p v_i$$

- Scaled dot-product attention(Vaswani et al., 2017)

$$\alpha = softmax(QK^T/\sqrt{d_k})V$$

3 Implementation Details

3.1 English to German Translation

3.1.1 Dataset

Due to lack of enough computational power, an NMT model is trained on the dataset given at a site http://www.manythings.org/anki/deu-eng.zip. Here two separate files are created for both English as well as German sentences. Out of these, maximum length of sentences is limited to 25 and minimum 3. Total sentences taken for training are 1,00,000.

3.1.2 Architecture Details

Here both encoder and decoder layers use 4 layers of bidirectional LSTM. The training is done batchwise with a batchsize of 256. Embedding size is kept as 128 as given in (Britz et al., 2017) embedding dimension doesn't add much for BLEU score though 2048 achieved best results according to the paper. Hidden size dimension is kept as 512, learning rate as 0.001. The training is via ADAM optimizer with 20 epochs of 1 Lakhs sentences. The gradients have been clipped once they go above 50.

3.1.3 Results

Here from the dataset given above, 1,00,000 sentences are filtered out according to their lengths. Out of these sentences, 85,000 were put into training corpus and 15,000 were put into testing corpus. The loss graphs for each of attention scheme is given in following diagrams.

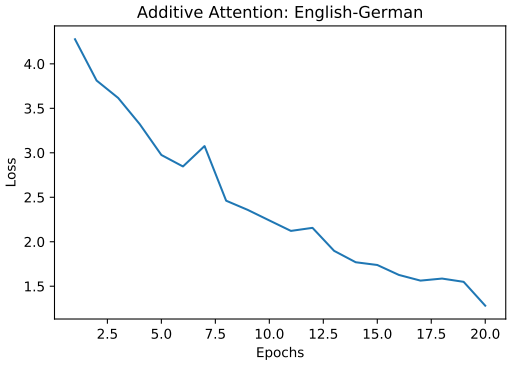

Figure 2: Loss in training for Additive attention for English-German translation

Some of the translations have been recorded from the training corpus as follows:

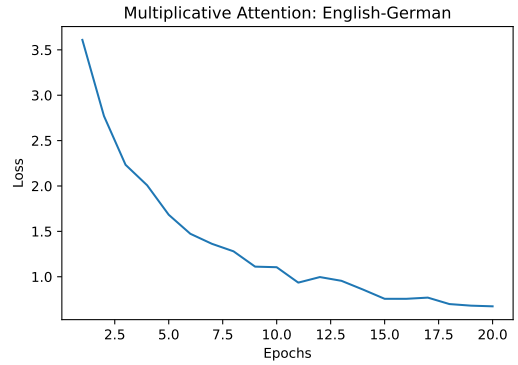

Figure 3: Loss in training for Multiplicative attention for English-German translation

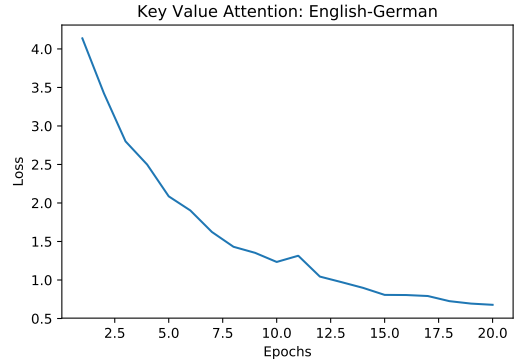

Figure 4: Loss in training for Key Value attention for English-German translation

Source :take this book back to him
Truth : bringen sie ihm das buch zuruck
Prediction : bringen das dieses buch

Source :she looked after her baby
Truth : sie kummerte sich um ihr baby
Prediction : sie hat sich um ihrem

One of exactly matched sentence

Source :he went to some place or other
Truth : er ging irgendwohin
Prediction : er ging irgendwohin

As length of the sentence increases, the prediction gets deteriorated. While shorter sentences are translated more and less correctly.

Source :i already told tom everything
Truth : ich habe tom schon alles gesagt
Prediction : ich habe gesehen alles gesagt hat hat hat hat hat hat hat hat hat hat hat hat hat hat gesagt hat hat hat gesagt

Source :she acted like a real baby
Truth : sie hat sich wie ein riesenbaby aufgefuhrt
Prediction : sie hat sich ein schulmadchen aufzufuhren

Attention Mechanism	Training	Testing
Additive	0.69719	0.45870
Multiplicative	0.67133	0.39753
Key Value Pair	0.60638	0.31837

Table 1: BLEU Score Comparisons

For reserved testing corpus, some of the predictions are as follows:

Source :do you know how to do that
Truth : wei t du wie das gemacht wird
Prediction : kannst du das wie tun

Source :that computer might not work
Truth : der rechner funktioniert moglicherweise nicht
Prediction : dieses boot funktioniert nicht

Source :i ll sleep in my room
Truth : ich werde in meinem zimmer schlafen
Prediction : ich schlafe als zimmer zimmer

Source :many promises had been made
Truth : es waren viele versprechungen gemacht worden
Prediction : viele versprechungen waren zu worden

3.1.4 Comparison of different attention mechanisms

Here, three types of attentions have been implemented. There BLEU-4 score on both training as well testing corpus can be seen in table 1. Here BLEU score is calculated sentence wise and then averaged. Here additive attention(Bahdanau et al., 2015) is performing better than other attention mechanisms on testing corpus. But BLEU score is treated as data dependent as well. The corpus on

which the model is trained, has sentences sorted according to their lengths. Here around 10,000 training set predictions as well as 500 testing set predictions have been uploaded on github repository.

Here the attention maps are shown which have been plotted at the end training. But they are little inconclusive.

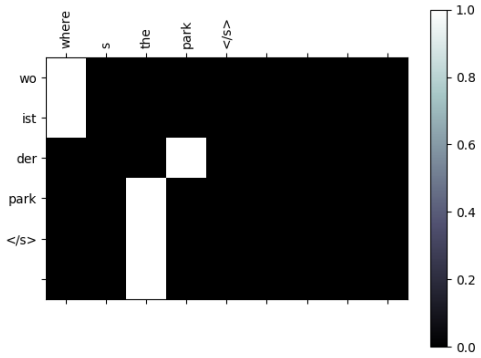

Figure 5: Attention map for Additive attention for English-German translation

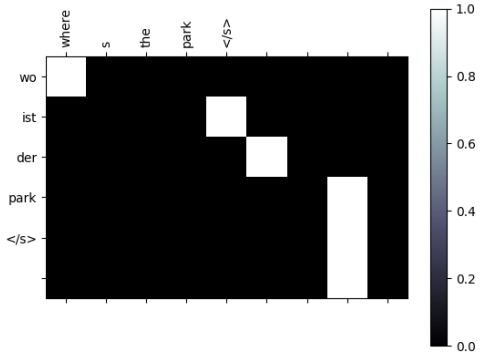

Figure 6: Attention map for Multiplicative attention for English-German translation

3.2 English to Hindi Translation

3.2.1 Dataset

An NMT model is trained on the dataset given in the assignment. Here two separate files are created for both English as well as German sentences. Out of these, maximum length of sentences is limited

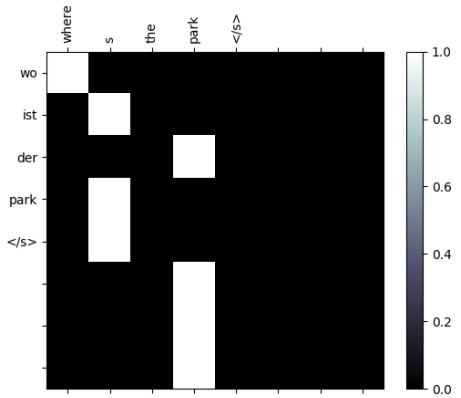

Figure 7: Attention map for Key-Value attention for English-German translation

to 25 and minimum 3. Total sentences taken for training are 10,000. Here due to tokenization issue, the GPU memory usage was exceeding GPU capacity.

3.2.2 Architecture Details

Here both encoder and decoder layers use 4 layers of bidirectional LSTM. The training is done batch-wise with a batchsize of 128. Embedding size is kept as 128 as given in (Britz et al., 2017) . Hidden size dimension is kept as 512, learning rate as 0.001. The training is via ADAM optimizer with 50 epochs of 10,000 sentences. The gradients have been clipped once they go above 50. In the assignment, the focus has been kept on being able to translate few sentences but as much as good quality as possible. Hence in both language translations, few sentences has been picked than mentioned and more epochs has been run to ensure we get good translation quality at least on training data.

3.2.3 Results

Here from the dataset given above, 10,000 sentences are filtered out according to their lengths. Out of these sentences, 9,000 were put into training corpus and 1000 were put into testing corpus. The loss graphs for each of attention scheme is given in following diagrams.

Some of the translations have been recorded from the training corpus as follows: Predictions in in testing dataset are not good as compared to training set.

BLEU Scores can be compared as follows:

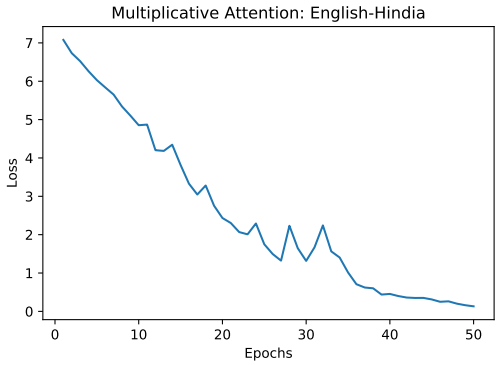

Figure 8: Loss in training for Multiplicative attention for English-Hindi translation

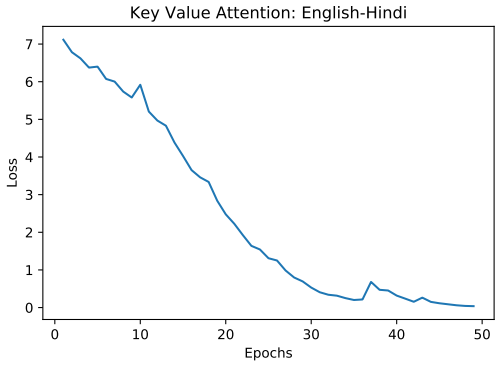

Figure 9: Loss in training for Key Value attention for English-Hindi translation

Attention Mechanism	Training	Testing
Additive	0.50719	0.07278
Multiplicative	0.44616	0.06974
Key Value Pair	0.64616	0.05975

Table 2: BLEU Score Comparisons

Here, in case of English to Hindi translation, the vocabulary size is increasing of the order 35,000 if we take 100000 sentences for training. So it gives GPU CUDA memory error. But if we reduce the size of the training set and run more epochs (50 in the current implementation), translation on the training set is good, but it quickly deteriorates on the testing dataset. Hence English to Hindi translation is found to be more challenging.

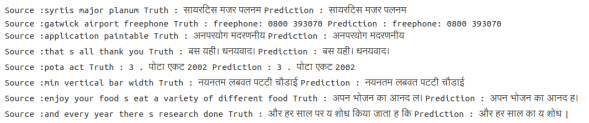

Figure 10: Translation in training for English-Hindi translation

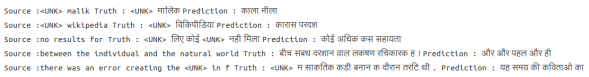

Figure 11: Translation in testing for English-Hindi translation

Acknowledgments

Here I acknowledge following NMT tutorials.

- https://pytorch.org/tutorials/intermediate/seq2seq_translation_tutorial.html
- https://github.com/spro/practical-pytorch/blob/master/seq2seq-translation/seq2seq-translation-batched.ipynb
- https://nbviewer.jupyter.org/github/DSKSD/DeepNLP-models-Pytorch/tree/master/notebooks/
- https://github.com/bentrevett/pytorch-seq2seq

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Denny Britz, Anna Goldie, Minh-Thang Luong, and Quoc Le. 2017. Massive exploration of neural machine translation architectures. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1442–1451. Association for Computational Linguistics.

Philipp Koehn. 2017. Neural machine translation. *CoRR*, abs/1709.07809.

Yang Liu and Mirella Lapata. 2017. Learning structured text representations. *CoRR*, abs/1705.09207.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. *CoRR*, abs/1508.04025.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.