ADL 2020 PROJECT-I REPORT

Mayank Gupta and Bazil Ahmed

{mayankg, bazilahmed}@iisc.ac.in

ABSTRACT

The standard Attention mechanism is used in a lot of tasks such as machine translation, text generation, etc. to obtain better results than the basic seq-to-seq LSTM based models. But the standard attention mechanism requires the decoder to attend to the complete input sequence for each time stamp, which makes it computationally expensive especially for longer sequences. In this work, we propose a multi-level attention mechanism that is, in theory, computationally efficient than the standard attention mechanism without much loss in performance. We have conducted the experiments on two tasks, which have also been experimented in [1], to analyze the trade off between the time and the performance measure of our proposed mechanism.

Index Terms— Attention, Sequence to Sequence, Hierarchy, LSTM, ACT

1. INTRODUCTION

Attention mechanism ([2],[3]) is used to compute a context vector for each decoder step, which requires looking at all encoder hidden states. This makes attention computationally expensive for long sequences.

Humans do not look at the complete source sequence while translating from one language to another ([1]). Inspired by this we propose a new attention mechanism which builds hierarchical representation of the inputs, required for capturing localized context efficiently.

2. TECHNICAL DETAILS

In this method, we build a hierarchical representation of input as shown in Figure 1. The main idea is to divide the encoder hidden states into blocks, and compute a representative hidden vector for each block at the next level and this goes on for some levels, say L, which is a hyper-parameter. Representative hidden vectors at each level are convex combination of the block it represents. Once we have this representation ready, while decoding we find attention on the top most level representative hidden vectors, and go deeper along the branches whose representative hidden vector gets the most attention score by the decoder. This goes on till we get to the last level, where decoder is attending to the block of original hidden states of the encoder.

We propose three ways to achieve this hierarchical representation which have been discussed further.

2.1. RNN Based Hierarchical Attention

Consider a basic Encoder-Decoder model of bidirectional LSTM cells, with $\mathbf{x} = (x_1, x_2, ..., x_T)$ as input embeddings and $\mathbf{h} = (h_1, h_2, ..., h_T)$ being the corresponding hidden states produced by the Encoder, where T represents the sequence length. Now consider an RNN cell placed over the LSTM cells of the Encoder, which takes $\mathbf{h}$ as input to produce $\mathbf{s^2} = ({s^2}_1, {s^2}_2, ..., {s^2}_T)$, which is then used to produce the 2^{nd} level representation of the inputs, also called $\mathbf{p^2}$. See the equations below for $l \in \{2, 3, ..., L\}$ and $t \in \{1, 2, ..., T\}$.

$$p_t^1 = h_t = LSTM(x_t, h_{t-1}) \tag{1}$$

$$s_t^l = RNN(p_t^{l-1}, s_{t-1}^l) \tag{2}$$

$$p_i^l = \sum_{j=k(i-1)+1}^{ki} \alpha_{ij} p_j^{l-1} \tag{3}$$

$$\alpha_{ij} = score(s_{ki}^l, h_j) \tag{4}$$

Where k is the fixed size of the block which is a hyperparameter and $\mathbf{p^l} = ({p^l}_1, {p^l}_2, ..., {p^l}_M)$ where $M = T/k^l$, denotes the number of representative hidden vectors at level l.

2.2. CNN Based Hierarchical Attention

In this approach, unlike RNN we apply 1D-CNN over $\mathbf{p^l}$ with the following configurations: kernel size = k, stride = k, and number of filters = dimension of hidden states of the encoder. This gives a replacement for eqn 2 and eqn 4 as following:

$$\mathbf{q^l} = (q_1, q_2, ..., q_{M/k}) = CNN(p_1^{l-1}, p_2^{l-1}, ..., p_M^{l-1}) \tag{5}$$

$$\alpha_{ij} = score(q_i^l, h_j) \tag{6}$$

2.3. ACT Based Hierarchical Attention

In the previous two approaches, we have used fixed size of blocks which is a hyperparameter. This hard constraint, independent of input sequence, is too strict to capture the localized context. In this approach, we are using dynamic block sizes

Thanks to Google for colab platform.

which are a function of the input sequence, based on the ideas of [4].
At level l, we do linear transformation of $p_t^{l-1}, \forall t$ to compute probability z_t as shown below.

$$z_t = \sigma(W^T p_t^{l-1} + b) \text{ where } W \in R^{d \times 1}, b \in R \quad (7)$$

To compute next level representative hidden vectors, we do the following:

$$k_0 = 0 \quad (8)$$

$$k_i = min\{n : \sum_{j=k_{i-1}+1}^{n} z_j \geq 1 - \epsilon\} \quad (9)$$

$$r_{k_i} = 1 - \sum_{j=k_{i-1}+1}^{k_i - 1} z_j \quad (10)$$

$$p_i^l = \sum_{j=k_{i-1}+1}^{k_i - 1} z_j p_j^{l-1} + r_{k_i} p_{k_i}^{l-1} \quad (11)$$

Also to encourage reduction of representative hidden vectors at consecutive higher levels, we penalise the model for smaller block sizes on the same lines of idea that of [4].

2.4. Decoding Over The Hierarchical Representation

Let **d** = $(d_1, d_2, ..., d_S)$ be the decoder hidden states, **c** = $(c_1^l, c_2^l, ..., c_S^l)$ be the context vectors at level l then,

$$\alpha_{it}^l = score(d_t, p_i^l) \quad (12)$$

$$z = \max_i \{\alpha_{it}^l\} \quad (13)$$

$$c_t^l = \sum_{i \neq z} \alpha_{it} p_i^l + \alpha_{zt} c_t^{l-1} \quad (14)$$

c_t^{l-1} is computed over the children of p_z^l and d_t. For $l = 1$, it will be a simple attention as a function of d_t and children of p_z^2.

2.5. Theoretical Time Complexity

D = hidden state dimension, T = Seq length of input, S = Seq length of output, and k = Block size.

Model	Complexity
Seq2Seq	$\mathcal{O}(D^2(S+T))$
Bahdanau	$\mathcal{O}(D^2(ST))$
2-Levels	$\mathcal{O}(D^2(ST)/k)$
$\log_k^T$ Levels	$\mathcal{O}(D^2(Sk\log_k^T))$

3. RESULTS

All models share the same hyperparameters and has been trained for 10 epochs, unless explicitly mentioned. Though theoretically our mechanism is computationally efficient, in our implementation we were not able to achieve better speed than standard mechanism. This happened due to the expensive decoding operations which could not be reduced to matrix operations, despite being parallelizable.

Model	k	levels	loss	PPI	Bleu
Bahdanau	-	-	3.214	24.872	29.33
CNN	3	3	3.411	30.293	23.82
RNN	6	2	3.358	28.727	24.02
ACT (3 epochs)	3	3	3.764	43.105	19.39

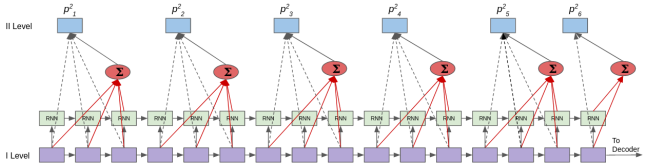
Table 1. Results of NMT Task on Multi30k Dataset

Model	k	levels	loss	PPI
Bahdanau	-	-	3.214	24.872
CNN	5	2	1.842	6.312
RNN	5	3	2.056	7.811

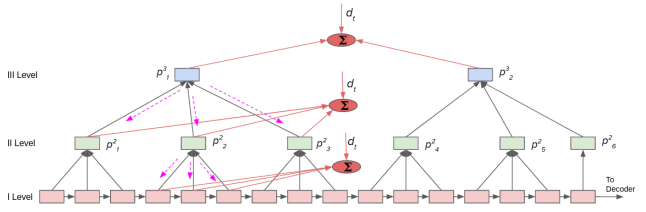
Table 2. Results of Sequence Copy Task for sequence length in [10, 100]

4. CONTRIBUTIONS

We have proposed a new attention mechanism which works on the hierarchical representation of inputs. We have also implemented three approaches to compute the hierarchical representation required for our attention mechanism (RNN, CNN, ACT).



(a) Encoding: with L=2, k=3



(b) Decoding: Context vector computation for one step

Fig. 1. Encoding-Decoding for RNN based Hierarchy.

5. RESOURCES

Codebase: https://github.com/bentrevett/pytorch-seq2seq

6. REFERENCES

[1] Denny Britz, Melody Y Guan, and Minh-Thang Luong, "Efficient attention using a fixed-size memory representation," *arXiv preprint arXiv:1707.00110*, 2017.

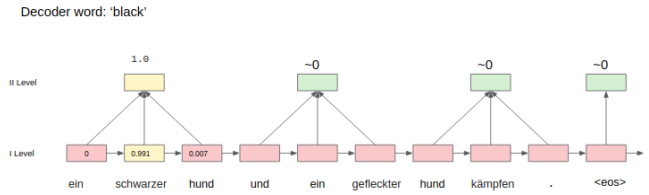
[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

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

[4] Alex Graves, "Adaptive computation time for recurrent neural networks," *arXiv preprint arXiv:1603.08983*, 2016.

7. APPENDIX

Attention flow is shown for word 'black', where source sequence is 'ein schwarzer hund und ein gefleckter hund kämpfen.' and the target sequence is 'a black dog and a spotted dog are fighting'



(a) Attention Visualization: with L=2, k=3