Curriculum's Minor Project on

Attention-Based Neural Machine Translation Model

Submitted By -

Vineet Yadav (1706002)
Aniket Sharma (1706003)
Vikas Kumar Chaurasia (1706008)
Deepu Gupta (1706025)

Submitted To -

Dr. Somaraju Suvvari

Introduction

- The Program trains on a Sequence to Sequence model for Spanish to English translation.

- Attention is proposed as a solution to the limitation of the Encoder-Decoder model. Encoding the input sequence to one fixed length vector from which to decode each output time step.

- This issue is believed to be more of a problem when decoding long sequences.

- The Input will be a Spanish sentence, such as "¿todavia estan en casa?", and it return Output as the English translation: "are you still at home?"

Datasets

- Used a language dataset provided by http://www.manythings.org/anki/.
- The Dataset contains language translation pairs in the format:

May I borrow this book? ¿Puedo tomar prestado este libro?

- There are a variety of languages available, but we‘ve used the English-Spanish dataset.

- The steps we‘ve taken to prepare the data:

1. Added a *start* and *end* token to each sentence.
2. Cleaned the sentences by removing special characters.
3. Created a word index and reversed word index (dictionaries mapping from word → id and id → word).
4. Padded each sentence to a maximum length.

Neural Machine Translation

A neural machine translation system is a neural network that directly models the conditional probability *p(y|x)* of translating a source sentence,
x1,…., xn to a target sentence y1, …, ym.

- A basic form of NMT consist of two components :

1. **E*ncoder*** which computes a representation for each source sentences.
2. **Decoder** which generates one target word at a time and hence de-composes the conditional probability as :

$$\log p(y|x) = \sum_{j=1}^{m} \log p\left(y_j|y_{<j}, \boldsymbol{s}\right)$$

- A natural choice to model such a decomposition in the decoder is to use a **Recurrent neural network** architecture, which most of the recent NMT work.

- Various models used many other forms such as stacked multiple layers of an RNN with a Long Short-Term Memory (LSTM) hidden unit for both the encoder and the decoder.

- Some adopted different version of the RNN with an LSTM-inspired hidden unit, the gated recurrent unit (GRU) for both the components.

- In more detail, we can parameterize the probability of decoding each word as :

$$p\left(y_j|y_{<j}, \boldsymbol{s}\right) = \text{softmax}\left(g\left(\boldsymbol{h}_j\right)\right)$$

with g being the transformation function that outputs a vocabulary-sized vector. Here, h*j is the* RNN hidden unit, abstractly computed as:

$$\boldsymbol{h}_j = f(\boldsymbol{h}_{j-1}, \boldsymbol{s}),$$

Our training object is formulated as :

$$J_t = \sum_{(x,y)\in\mathbb{D}} -\log p(y|x)$$

Attention

- Attention is proposed as a method to both align and translate.
- Alignment is the problem in machine translation that identifies which parts of the input sequence are relevant to each word in the output.
- whereas Translation is the process of using the relevant information to select the appropriate output.

Attention-Based Model

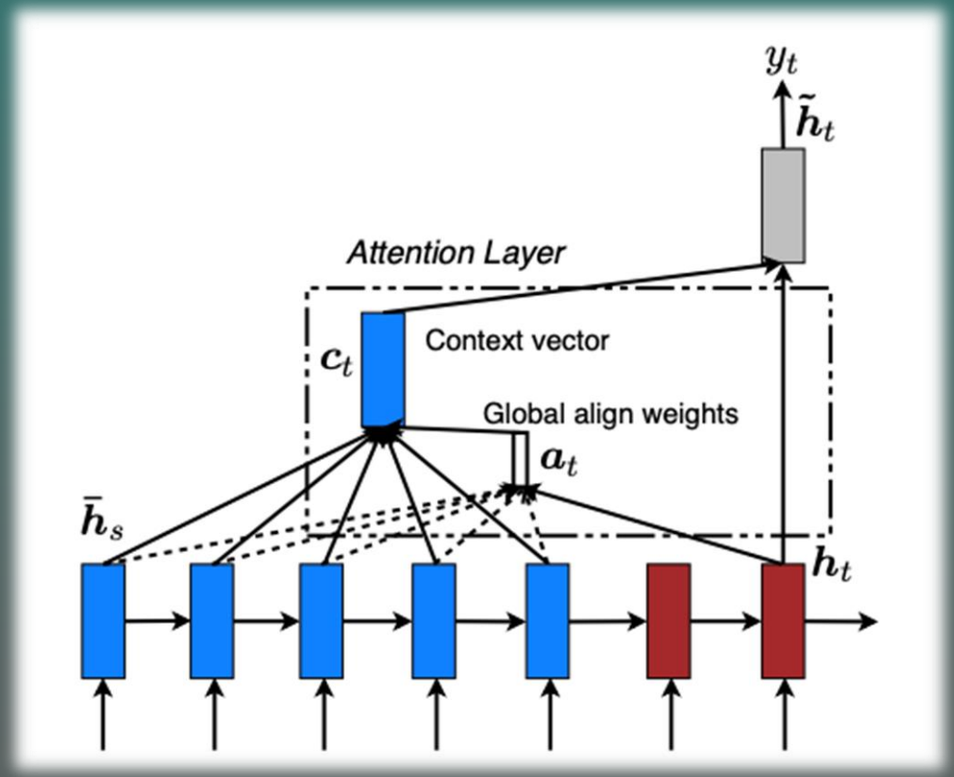
- Attention-based models are classified into two broad categories – Global and Local.

- These classes differ in terms of whether the “attention” is placed on all source positions or on only a few source positions.

Global Attention

- It consider all the hidden states of the encoder when deriving the context vector c_t. In this model type, a variable-length alignment vector a_t, whose size equals the number of time steps on the source side, is derived by comparing the current target hidden state $h_t$with each source hidden state h_s:

$$\begin{aligned} \boldsymbol{a}_t(s) &= \text{align}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) \\ &= \frac{\exp\left(\text{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s)\right)}{\sum_{s'} \exp\left(\text{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_{s'})\right)} \end{aligned}$$



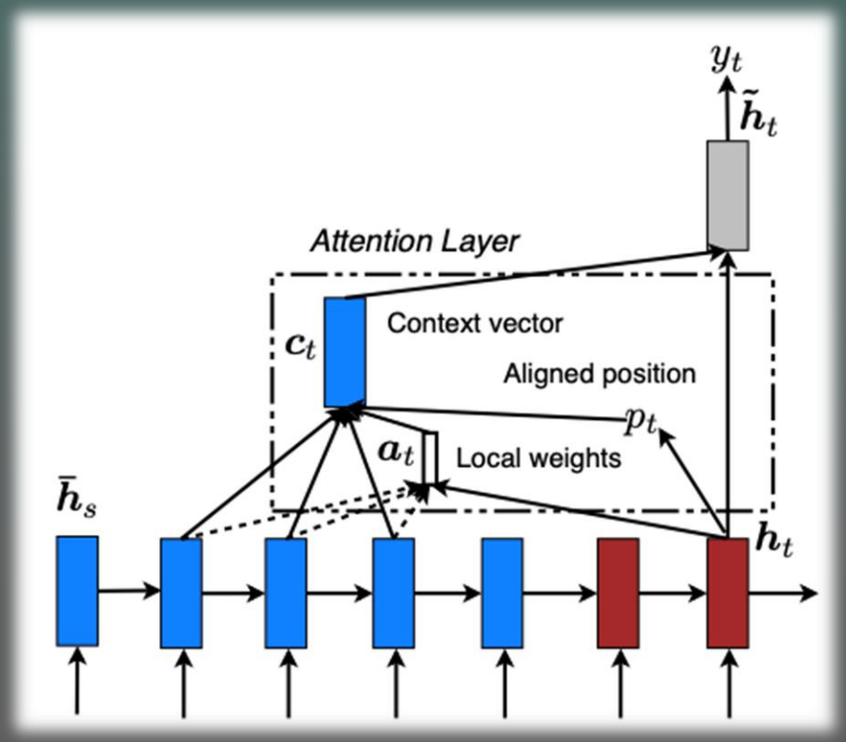
Drawback : It has to attend to all words on the source side for each target word, which is expensive and can potentially render it impractical to translate longer sequences.

Local Attention

- To address this drawback, we propose a *Local* Attention mechanism that chooses to focus only on a small subset of the source positions per target word.
- Local attention mechanism selectively focuses on a small window of context and is differentiable.
- This approach has an advantage of avoiding the expensive computation incurred in the soft attention and at the same time, is easier to train than the hard attention approach.

Our alignment weights are defined as:

$$\boldsymbol{a}_t(s) = \text{align}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) \exp\left(-\frac{(s - p_t)^2}{2\sigma^2}\right)$$



Optimizer

- **Adam** optimization is a stochastic gradient descent method that is based on adaptive estimation of first-order and second-order moments.

- The method is "computationally efficient, has little memory requirement, invariant to diagonal rescaling of gradients, and is well suited for problems that are large in terms of data/parameters”.

Advantages of using Adam optimizer :

- Easy to implement.
- Quite computationally efficient.
- Requires little memory space.
- Works well on problems with noisy or sparse gradients.
- Works well with large data sets and large parameters.

- **ADAM** updates any parameter with an individual **learning rate**.

- This means that every parameter in the network have a specific **learning rate** associated.

Algorithm

Task: Machine translation

Data: $\{x_i = source_i, y_i = target_i\}_{i=1}^{N}$

Encoder:

$$h_t = RNN(h_{t-1}, x_t)$$
$$s_t = h_t$$

Decoder:

$$e_{jt} = V_{att}^{T}\, tanh(U_{att} S_{t-1} + W_{att} h_j)$$
$$a_{jt} = softmax(e_{jt})$$
$$c_t = \sum_{j=1}^{T} a_{jt} * h_t$$
$$s_t = RNN(s_{t-1}, [c_t, e(y_{t-1})])$$
$$l_t = softmax(V s_t + b)$$

Training

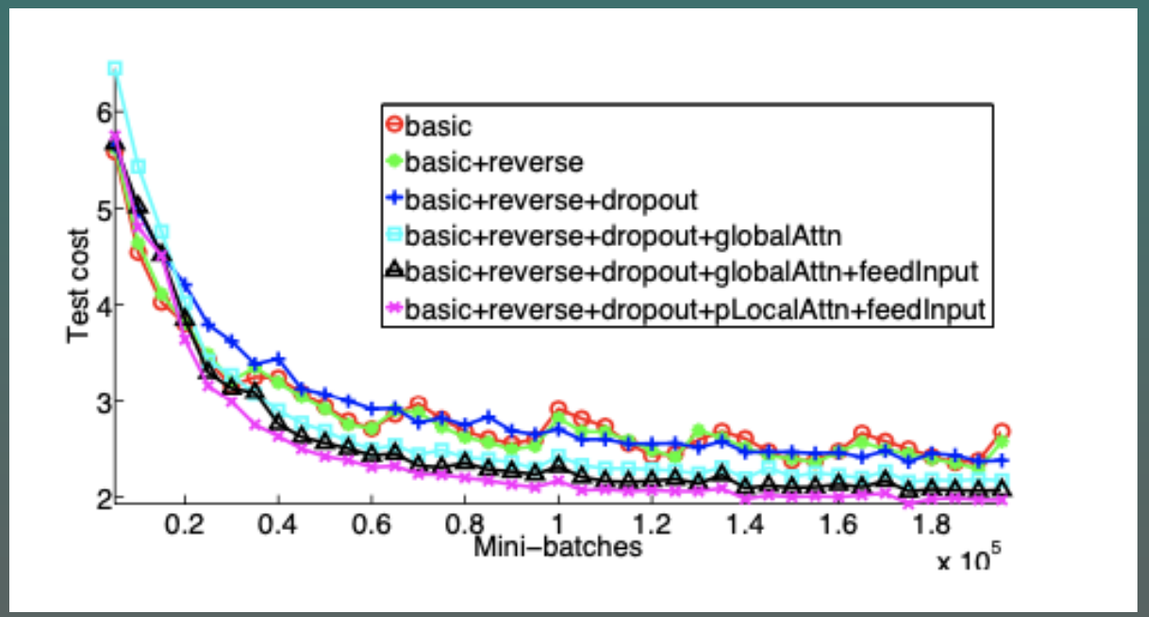
- All our models are trained on the WMT’14 training data consisting of Spanish and English words.
- Our stacking LSTM models have 4 layers, each with 1000 cells, and 1000-dimensional embedding.
- We train for 10 epochs using plain SGD.
- A simple learning rate schedule is employed – we start with a learning rate of 1; after 5 epochs, we begin to halve the learning rate every epoch.
- we also use dropout with probability 0.2 for our LSTMs.
- For dropout models, we train for 12 epochs and start halving the learning rate after 8 epochs.

Analysis

We conduct extensive analysis to better understand our models in terms of learning, the ability to handle long sentences, choices of attention architectures, and alignment quality. All results reported here are on English-German news test.

Learning Curves

- It is pleasant to observe in Figure a clear separation between non-attentional and attention models.



- Between non-attentional and attentional models. The input-feeding approach and the local attention model also demonstrate their abilities in driving the test costs lower.

Conclusion

- we propose two simple and effective attentional mechanisms for neural machine translation:-
    1. The *global* approach which always looks at all source positions and
    2. The *local* one that only attends to a subset of source positions at a time.
- We test the effectiveness of our models in the WMT translation tasks between English and Spanish in both directions.
- Our local attention yields large gains of up to 5.0 BLEU over non-attentional models which already incorporate known techniques such as dropout.
- We have compared various alignment functions and shed light on which functions are best for which attentional models.
- Our analysis shows that attention-based NMT models are superior to non-attentional ones in many cases, for example in translating names and handling long sentences.

References

- **Effective Approaches to Attention-based Neural Machine Translation**

  Minh-Thang Luong Hieu Pham Christopher D. Manning

  Computer Science Department, Stanford University, Stanford, CA 94305

  Link:- https://arxiv.org/abs/1508.04025v5

- **TensorFlow**

Thank You