Attention mechanism

2020.01.0
3

Dongguk University AI LAB
Hyojin Ki

Contents

- Introduction
- machine translation
- attention step
- attention mechanism
- image captioning
- Few-shot learning

Introduction

What is attention mechanism?

ATTENTION

Machine Translation

- Bahdanau,Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." (2014)

NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE

Dzmitry Bahdanau
Jacobs University Bremen, Germany

KyungHyun Cho **Yoshua Bengio**[*]
Université de Montréal

ABSTRACT

Neural machine translation is a recently proposed approach to machine translation. Unlike the traditional statistical machine translation, the neural machine translation aims at building a single neural network that can be jointly tuned to maximize the translation performance. The models proposed recently for neural machine translation often belong to a family of encoder–decoders and encode a source sentence into a fixed-length vector from which a decoder generates a translation. In this paper, we conjecture that the use of a fixed-length vector is a bottleneck in improving the performance of this basic encoder–decoder architecture, and propose to extend this by allowing a model to automatically (soft-)search for parts of a source sentence that are relevant to predicting a target word, without having to form these parts as a hard segment explicitly. With this new approach, we achieve a translation performance comparable to the existing state-of-the-art phrase-based system on the task of English-to-French translation. Furthermore, qualitative analysis reveals that the (soft-)alignments found by the model agree well with our intuition.

1 INTRODUCTION

Neural machine translation is a newly emerging approach to machine translation, recently proposed by Kalchbrenner and Blunsom (2013), Sutskever *et al.* (2014) and Cho *et al.* (2014b). Unlike the traditional phrase-based translation system (see, e.g., Koehn *et al.*, 2003) which consists of many small sub-components that are tuned separately, neural machine translation attempts to build and train a single, large neural network that reads a sentence and outputs a correct translation.

Most of the proposed neural machine translation models belong to a family of *encoder–decoders* (Sutskever *et al.*, 2014; Cho *et al.*, 2014a), with an encoder and a decoder for each language, or involve a language-specific encoder applied to each sentence whose outputs are then compared (Hermann and Blunsom, 2014). An encoder neural network reads and encodes a source sentence into a fixed-length vector. A decoder then outputs a translation from the encoded vector. The whole encoder–decoder system, which consists of the encoder and the decoder for a language pair, is jointly trained to maximize the probability of a correct translation given a source sentence.

A potential issue with this encoder–decoder approach is that a neural network needs to be able to compress all the necessary information of a source sentence into a fixed-length vector. This may make it difficult for the neural network to cope with long sentences, especially those that are longer than the sentences in the training corpus. Cho *et al.* (2014b) showed that indeed the performance of a basic encoder–decoder deteriorates rapidly as the length of an input sentence increases.

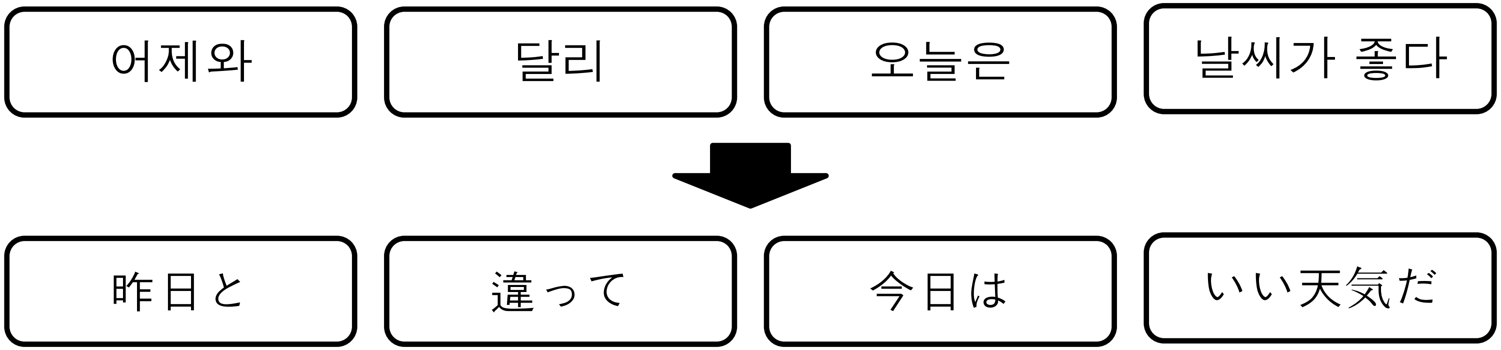
In order to address this issue, we introduce an extension to the encoder–decoder model which learns to align and translate jointly. Each time the proposed model generates a word in a translation, it (soft-)searches for a set of positions in a source sentence where the most relevant information is concentrated. The model then predicts a target word based on the context vectors associated with these source positions and all the previous generated target words.

[*] CIFAR Senior Fellow

Machine Translation

What is Machine translation?

- the process of changing text from one language into another language using a computer.

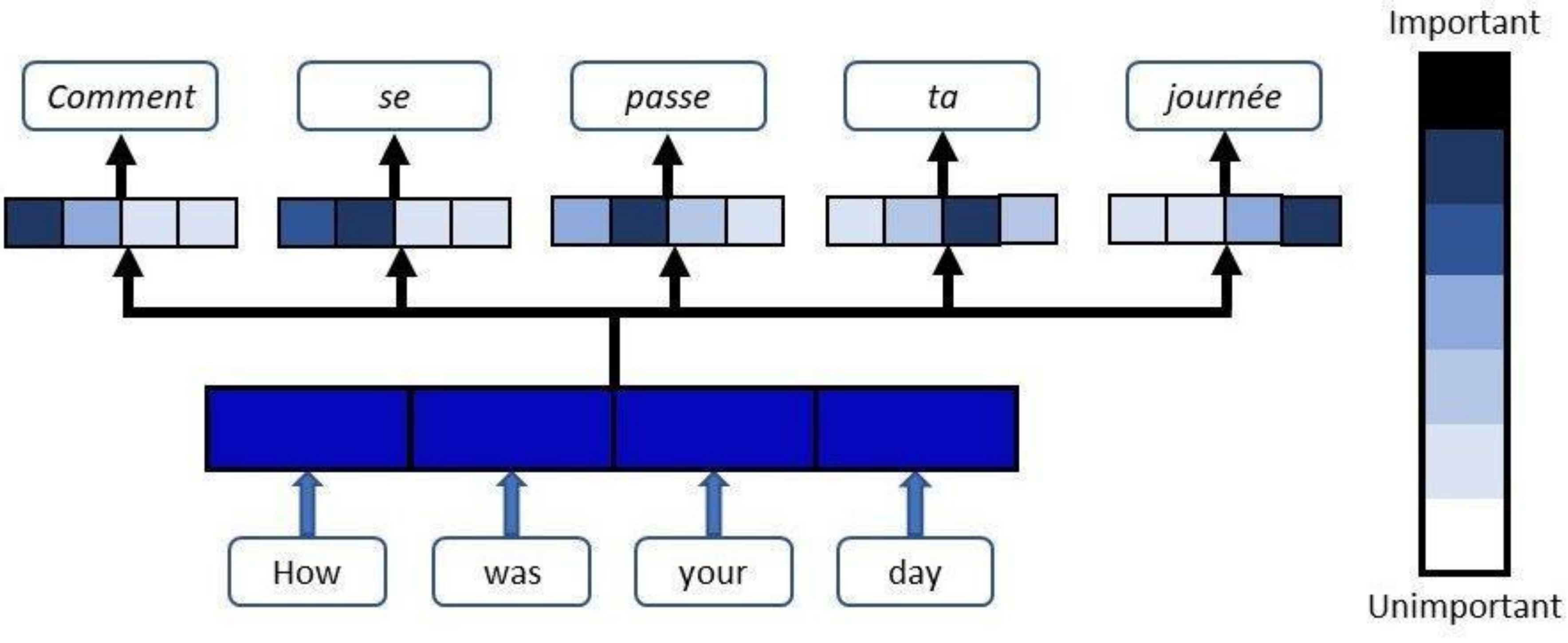


어제와 달리 오늘은 날씨가 좋다

昨日と 違って 今日は いい天気だ

Machine Translation

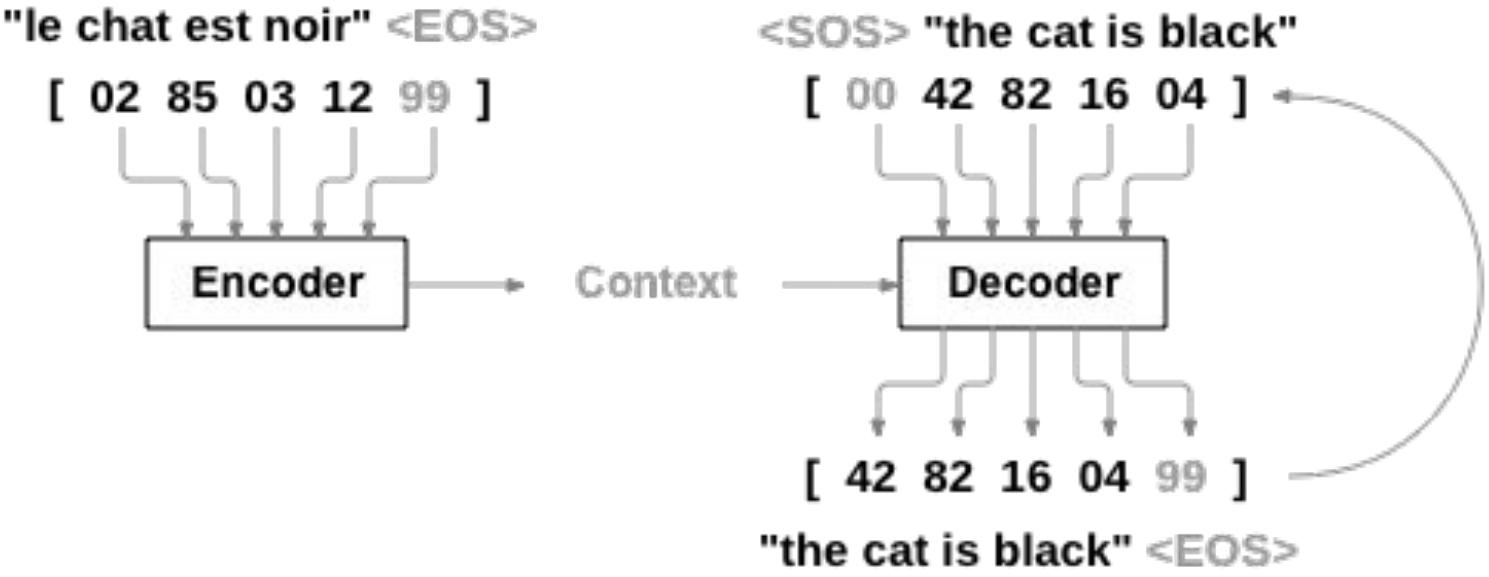
Attention based Machine translation



Machine Translation

What is Seq2Seq(Sequence to Sequence)?

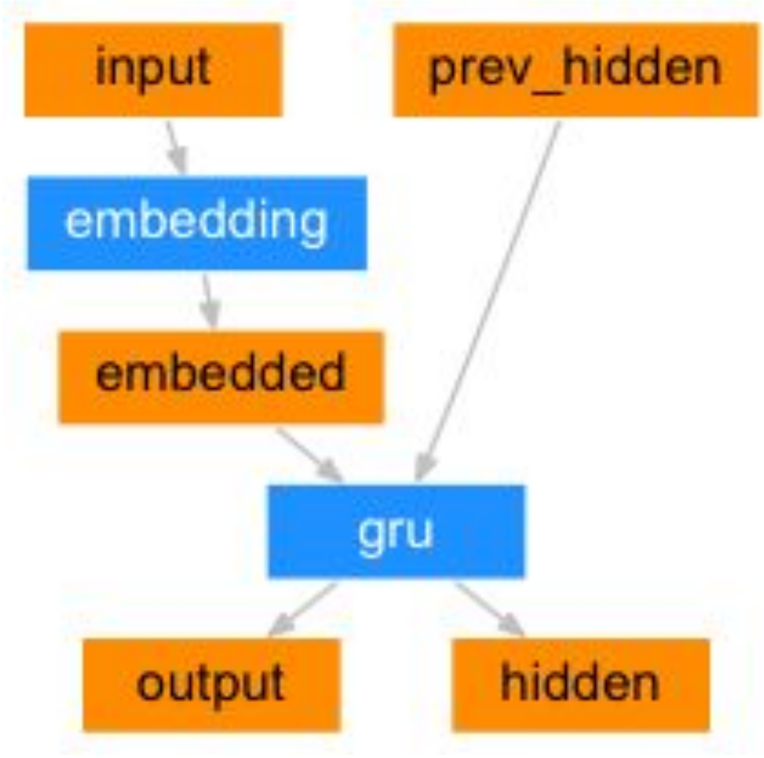
- It consists two recurrent neural network(RNN)s. And two RNNS work together to transform one sequence to another. An encoder network condenses an input sequence into a vector, and a decoder network unfolds that vector into a new sequence.



Machine Translation

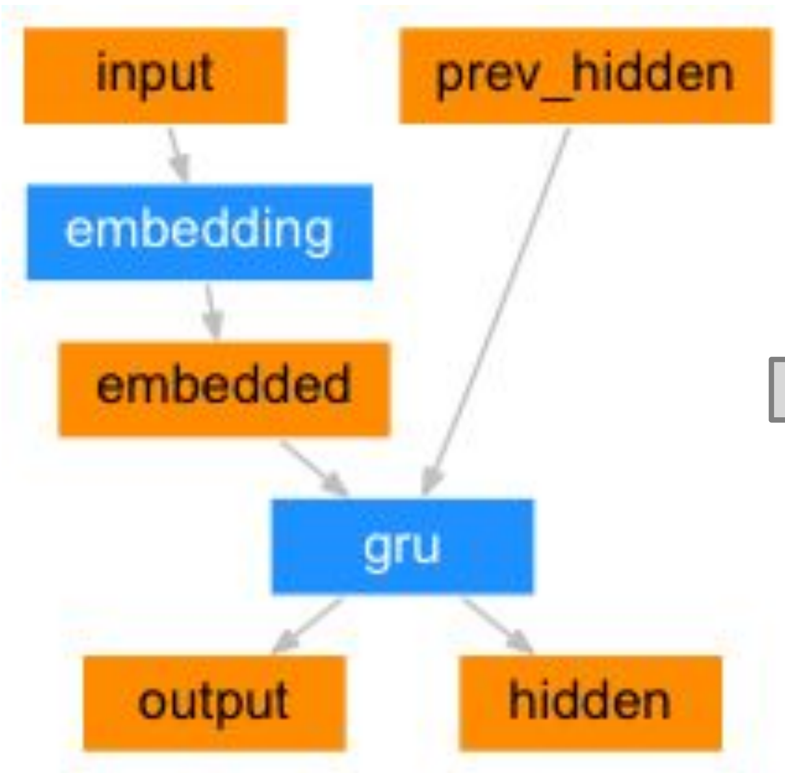
Encoder in Seq2Seq

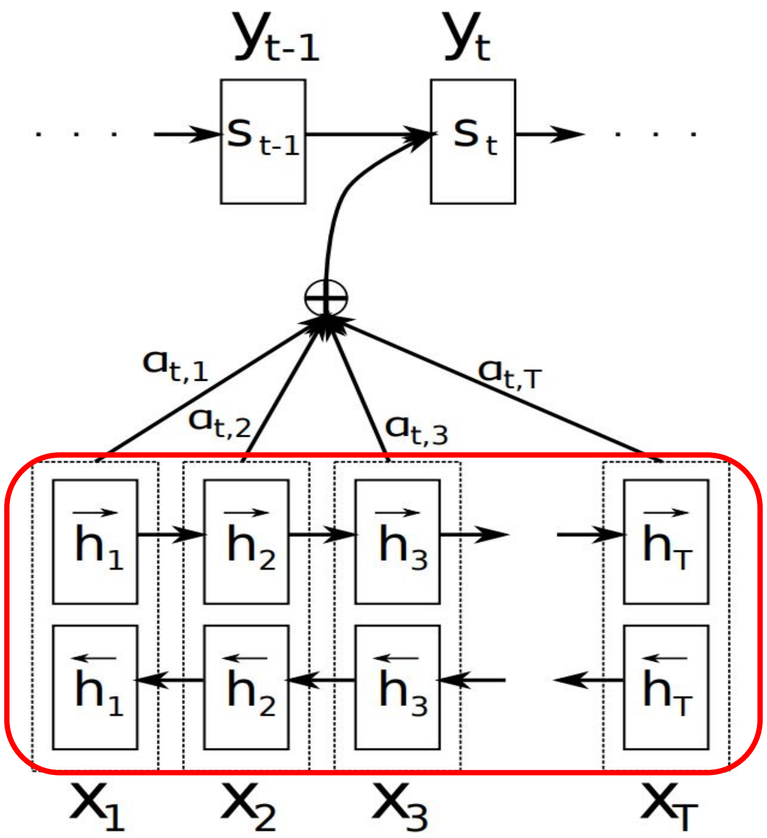
- Encoder RNN outputs some value for every word from the input sentence. For every input word the encoder outputs a vector and a hidden state, and uses the hidden state for the next input word.



Machine Translation

Encoder in Seq2Seq

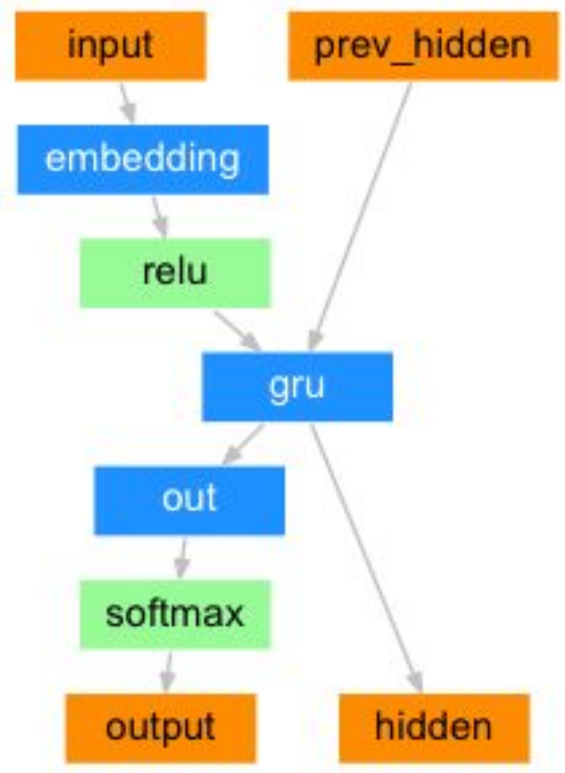




Machine Translation

Decoder in Seq2Seq

- The decoder is another RNN that takes the encoder output vector(s) and outputs a sequence of words to create the translation.

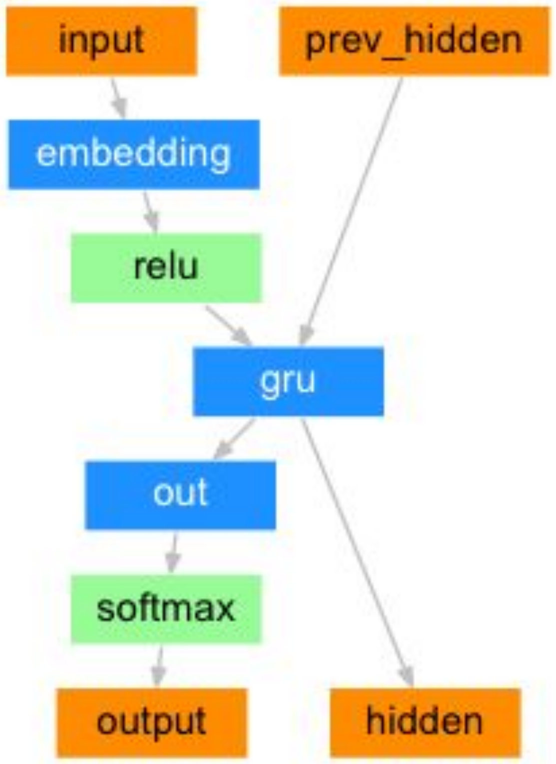


Machine Translation

Decoder in Seq2Seq

- The decoder is another RNN that takes the encoder output vector(s) and outputs a sequence of words to create the translation.

Initial input
<SOS>
Start Of
Sentence

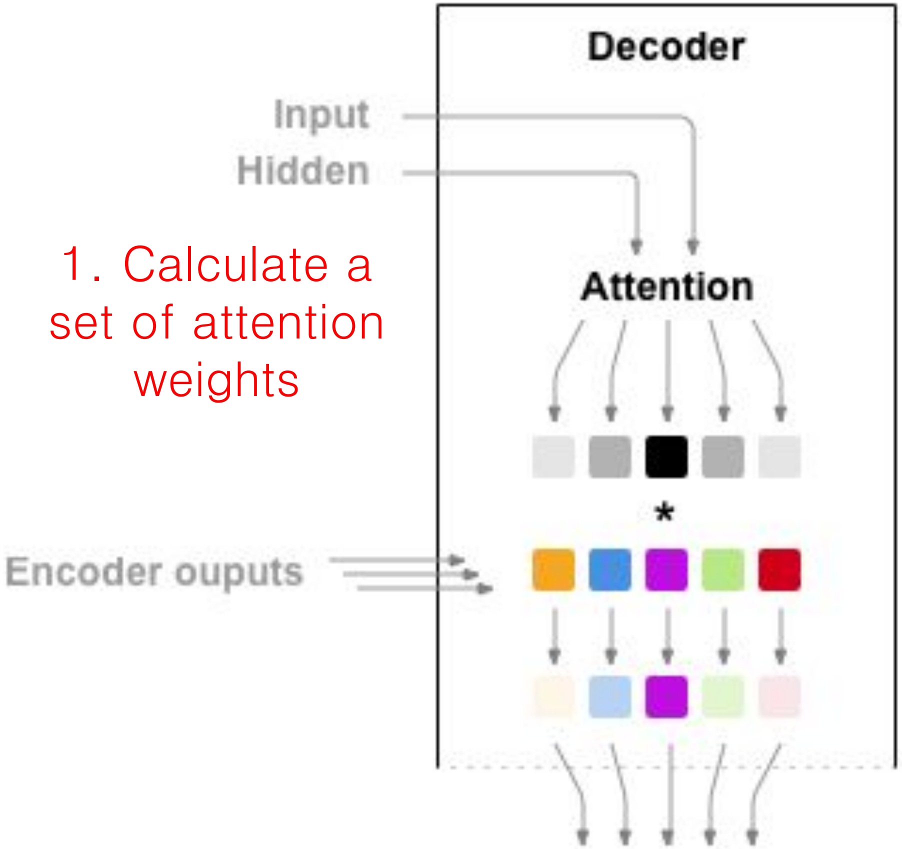


Initial
hidden state
Context
vector
(encoder
output)

Machine Translation

Attention Decoder in Seq2Seq

- Attention allows the decoder network to "focus" on a different part of the encoder's outputs for every step of the decoder's own outputs.

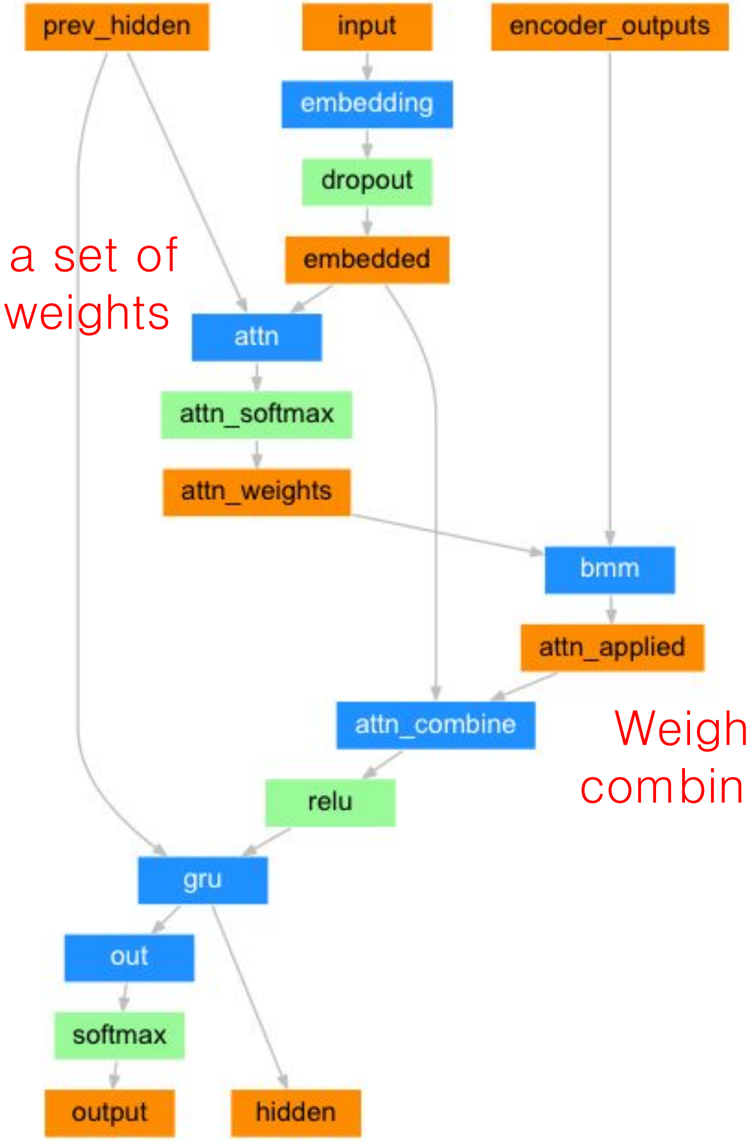


1. Calculate a set of attention weights

2. Multiply Attention weights and encoder outputs

Machine Translation

Attention Decoder in Seq2Seq

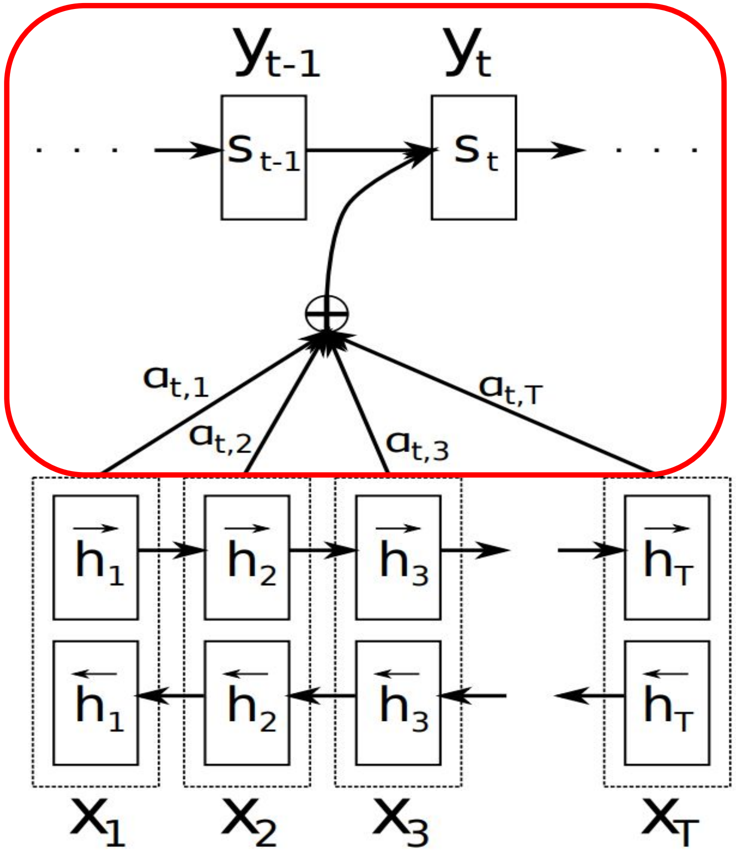


Calculate a set of attention weights

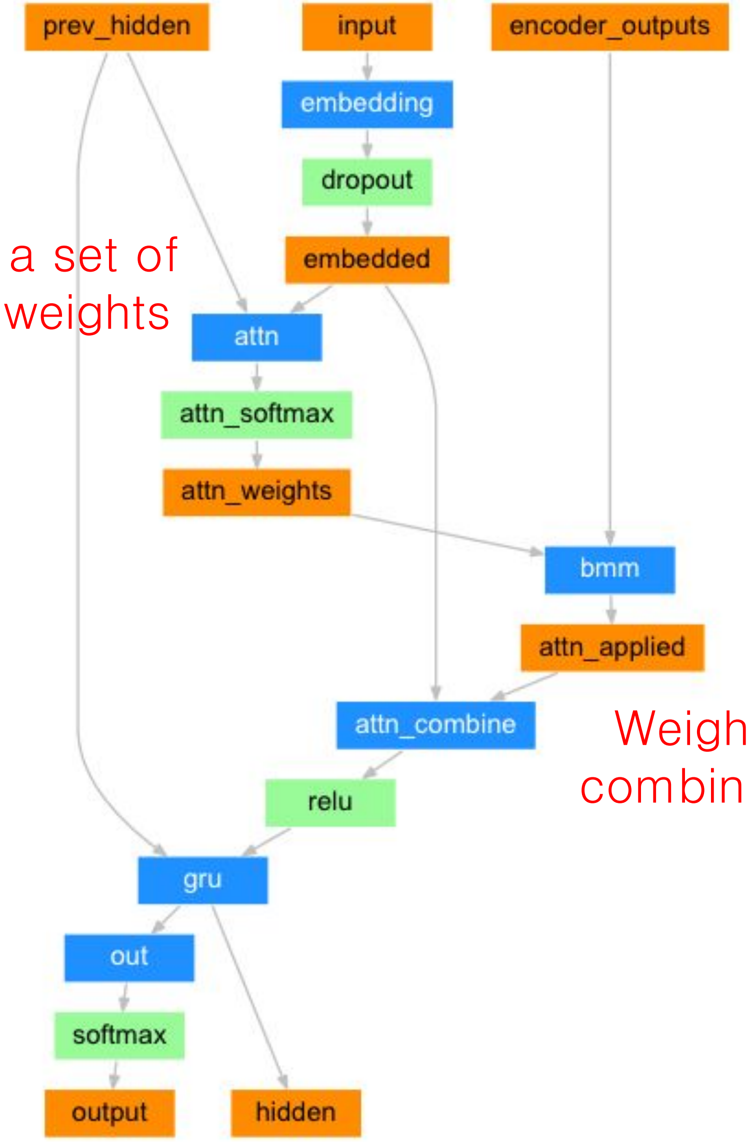
Weighted combination

Machine Translation

Attention Decoder in Seq2Seq



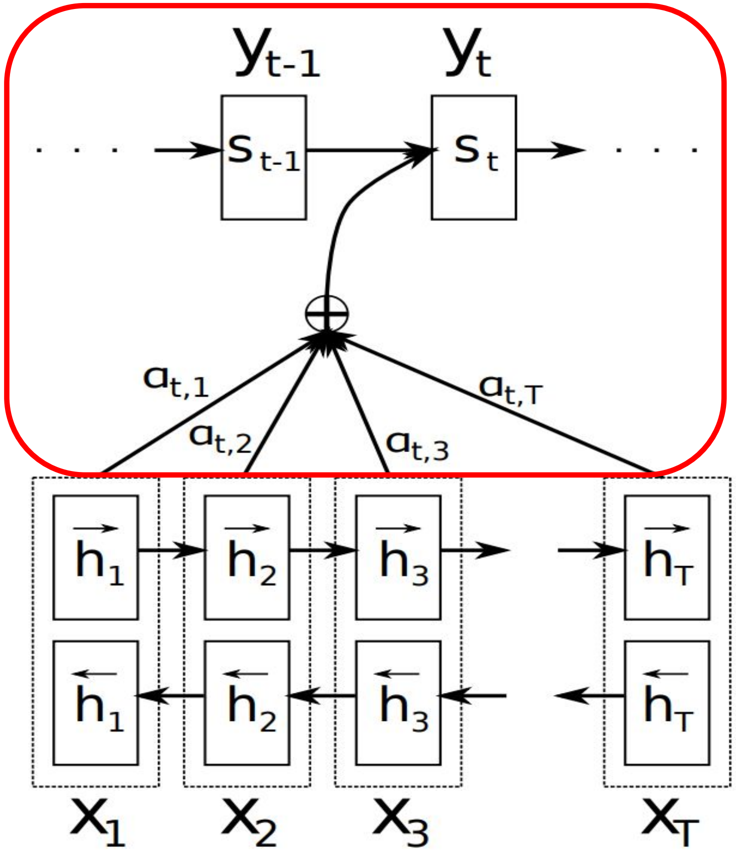
Calculate a set of attention weights



Weighted combination

Machine Translation

Attention Decoder in Seq2Seq



Targe word Source word RNN hidden state for time i

$$p(y_i|y_1,..,y_{i-1},x) = g(y_{i-1},s_i,c_i)$$

Conditional probabiliy

$$s_i = f(s_{i-1},y_{i-1},c_i)$$

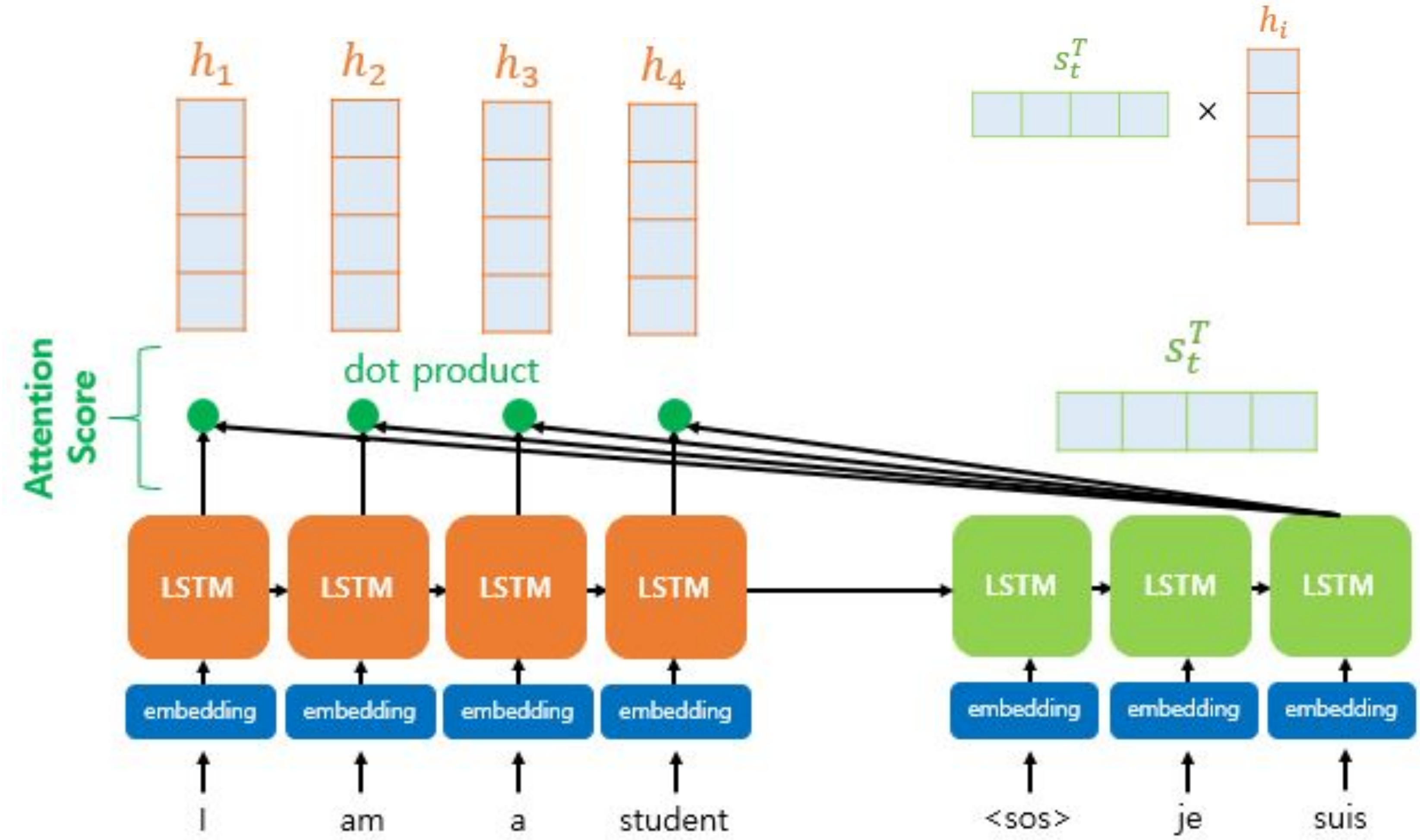
Context vector $c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}$$

$$e_{ij} = a(s_{i-1},h_j)$$

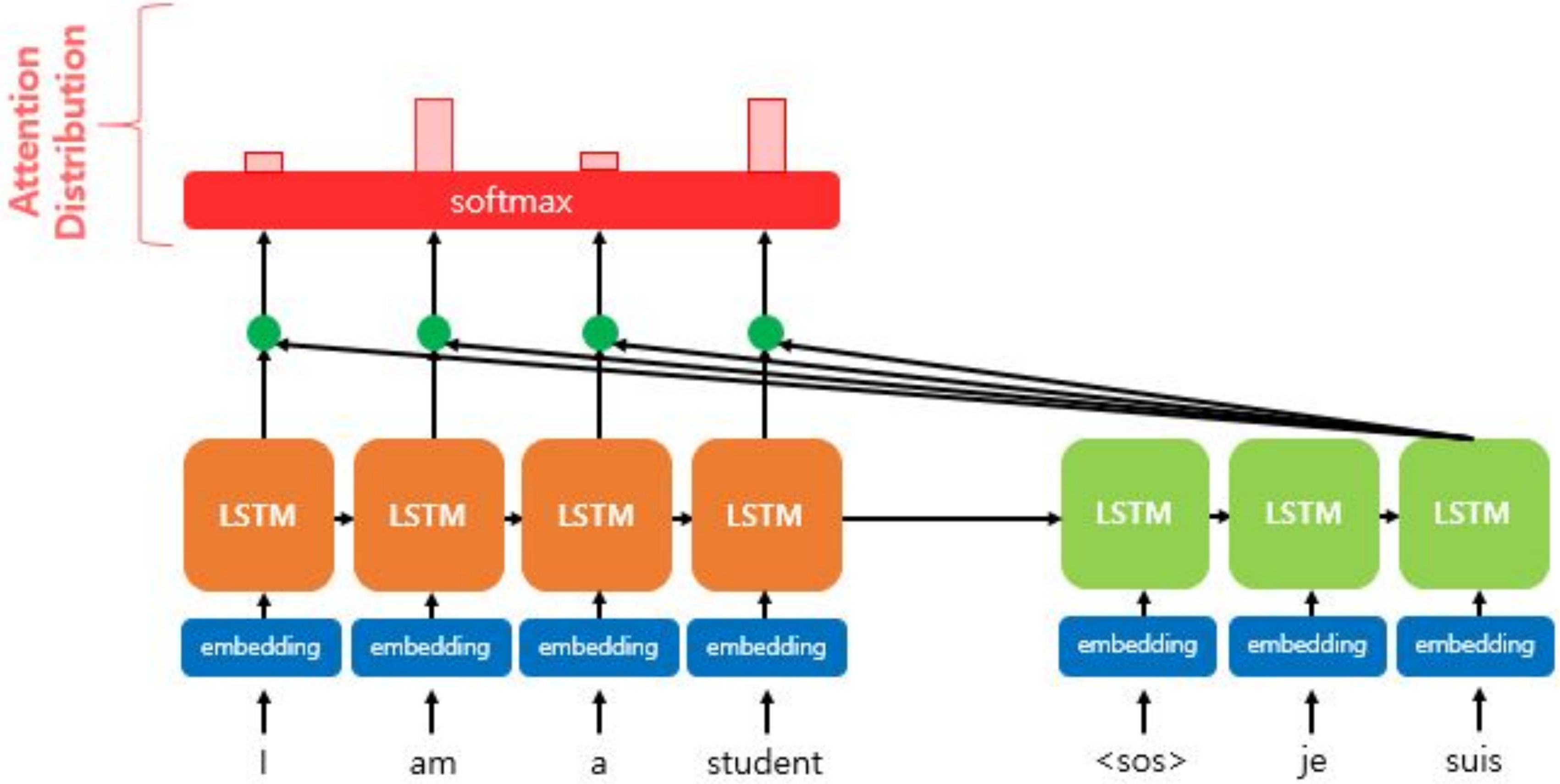
Attention mechanism

1. Calculate Attention Score



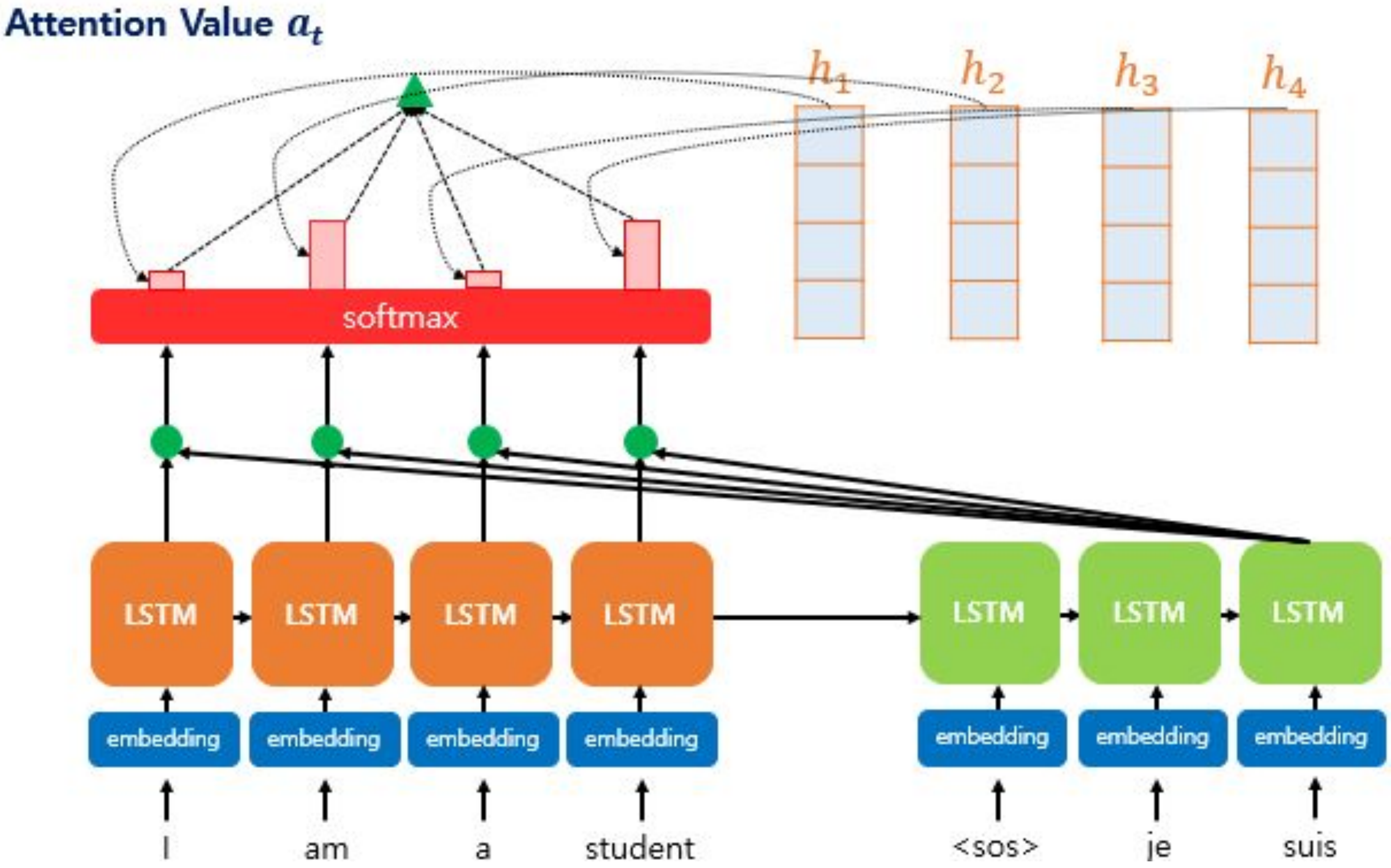
Attention mechanism

2. Calculate Attention Distribution using Softmax function



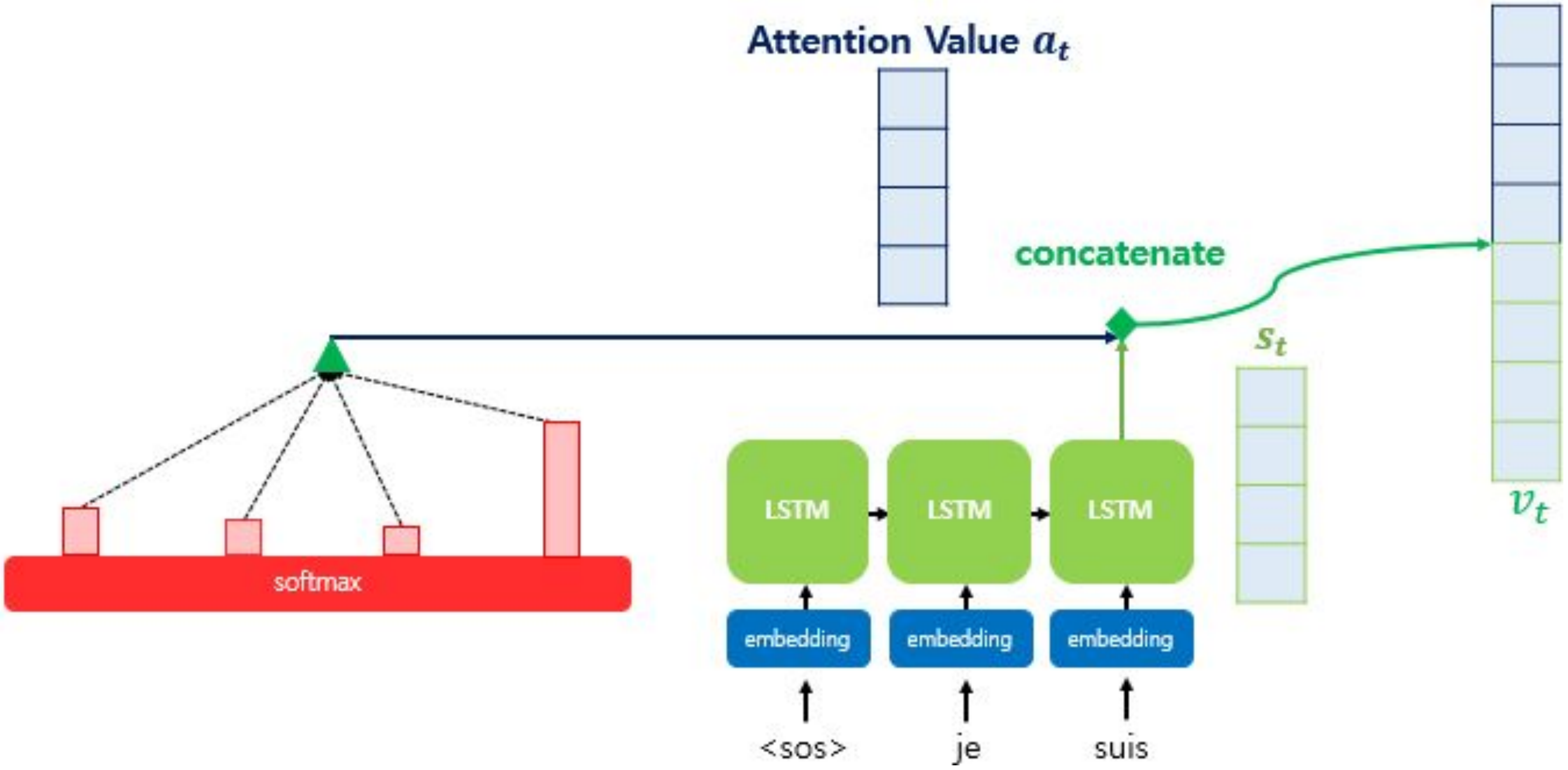
Attention mechanism

3. Calculate Attention value using weighted sum of the attention weight and the hidden state of each encoder.

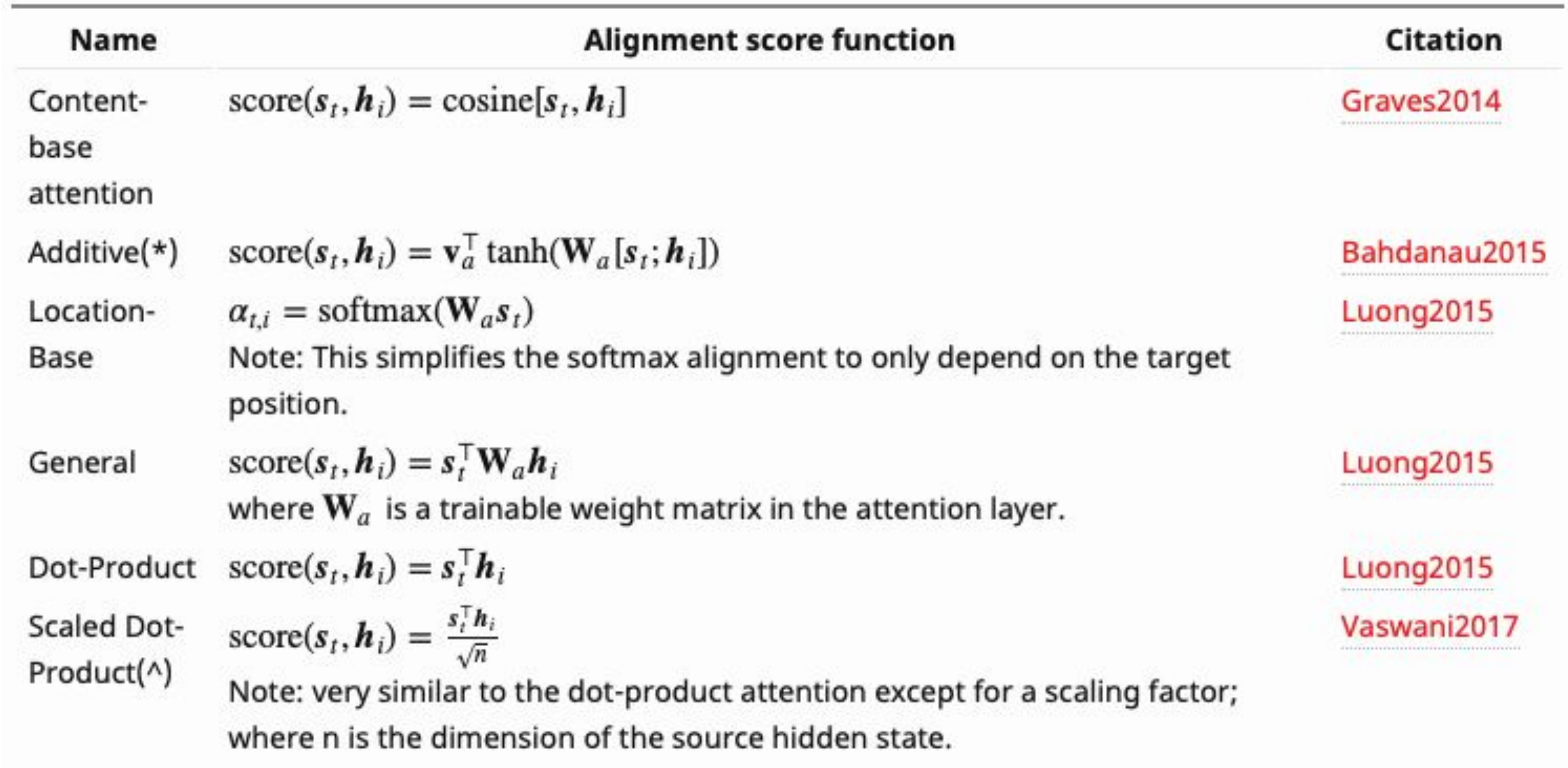


Attention mechanism

4. Concatenate the Attention value and the decoder hidden state at time step t.

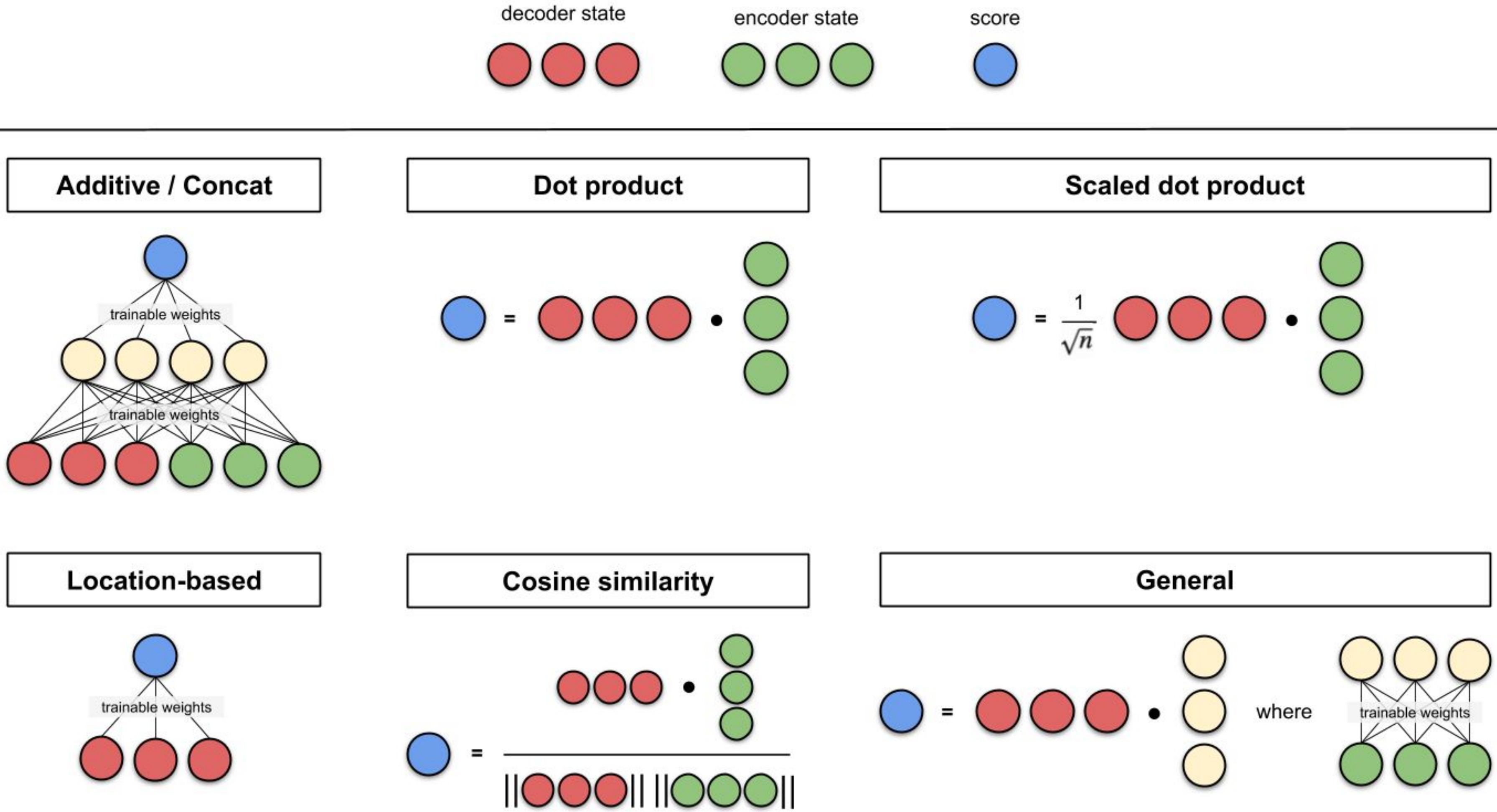


Attention score



Name	Alignment score function	Citation
Content-base attention	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \text{cosine}[\boldsymbol{s}_t, \boldsymbol{h}_i]$	Graves2014
Additive(*)	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \mathbf{v}_a^\top \tanh(\mathbf{W}_a[\boldsymbol{s}_t; \boldsymbol{h}_i])$	Bahdanau2015
Location-Base	$\alpha_{t,i} = \text{softmax}(\mathbf{W}_a \boldsymbol{s}_t)$ Note: This simplifies the softmax alignment to only depend on the target position.	Luong2015
General	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \boldsymbol{s}_t^\top \mathbf{W}_a \boldsymbol{h}_i$ where $\mathbf{W}_a$ is a trainable weight matrix in the attention layer.	Luong2015
Dot-Product	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \boldsymbol{s}_t^\top \boldsymbol{h}_i$	Luong2015
Scaled Dot-Product(^)	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \frac{\boldsymbol{s}_t^\top \boldsymbol{h}_i}{\sqrt{n}}$ Note: very similar to the dot-product attention except for a scaling factor; where n is the dimension of the source hidden state.	Vaswani2017

Attention score



decoder state

encoder state

score

Additive / Concat

Dot product

Scaled dot product

Location-based

Cosine similarity

General

Image captioning

What is image captioning?

- Image Captioning is the process of generating textual description of an image.

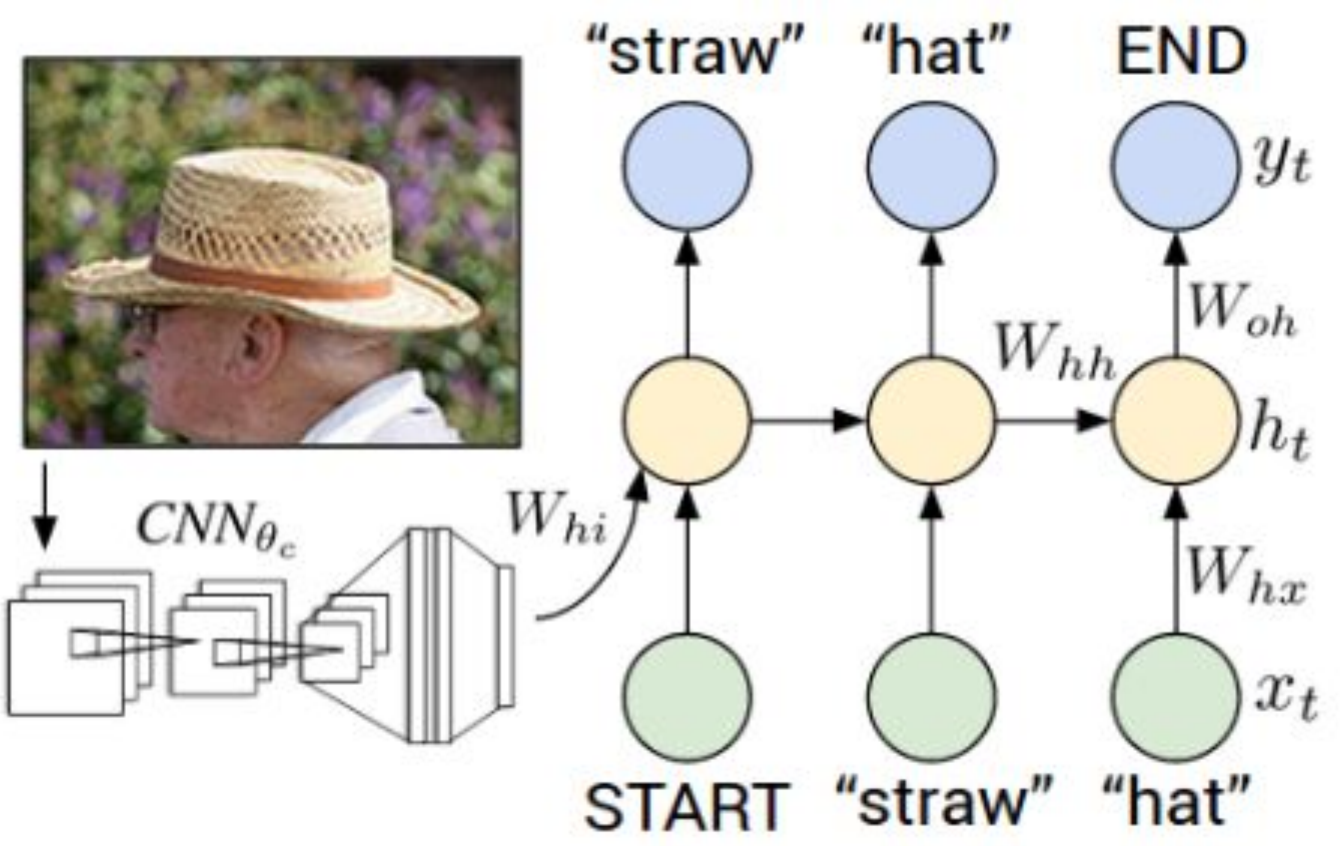


Image captioning

Attention based Image captioning

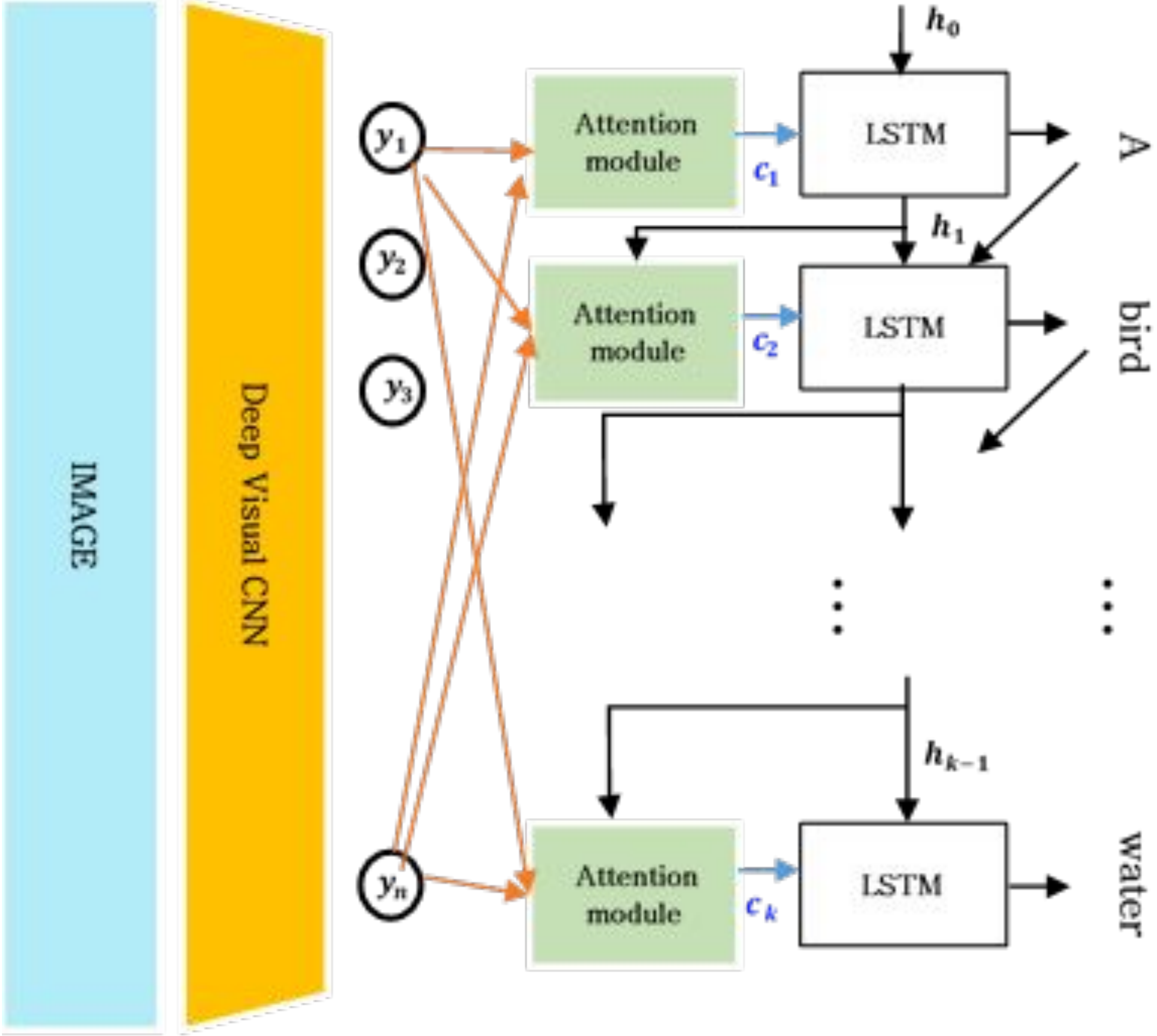


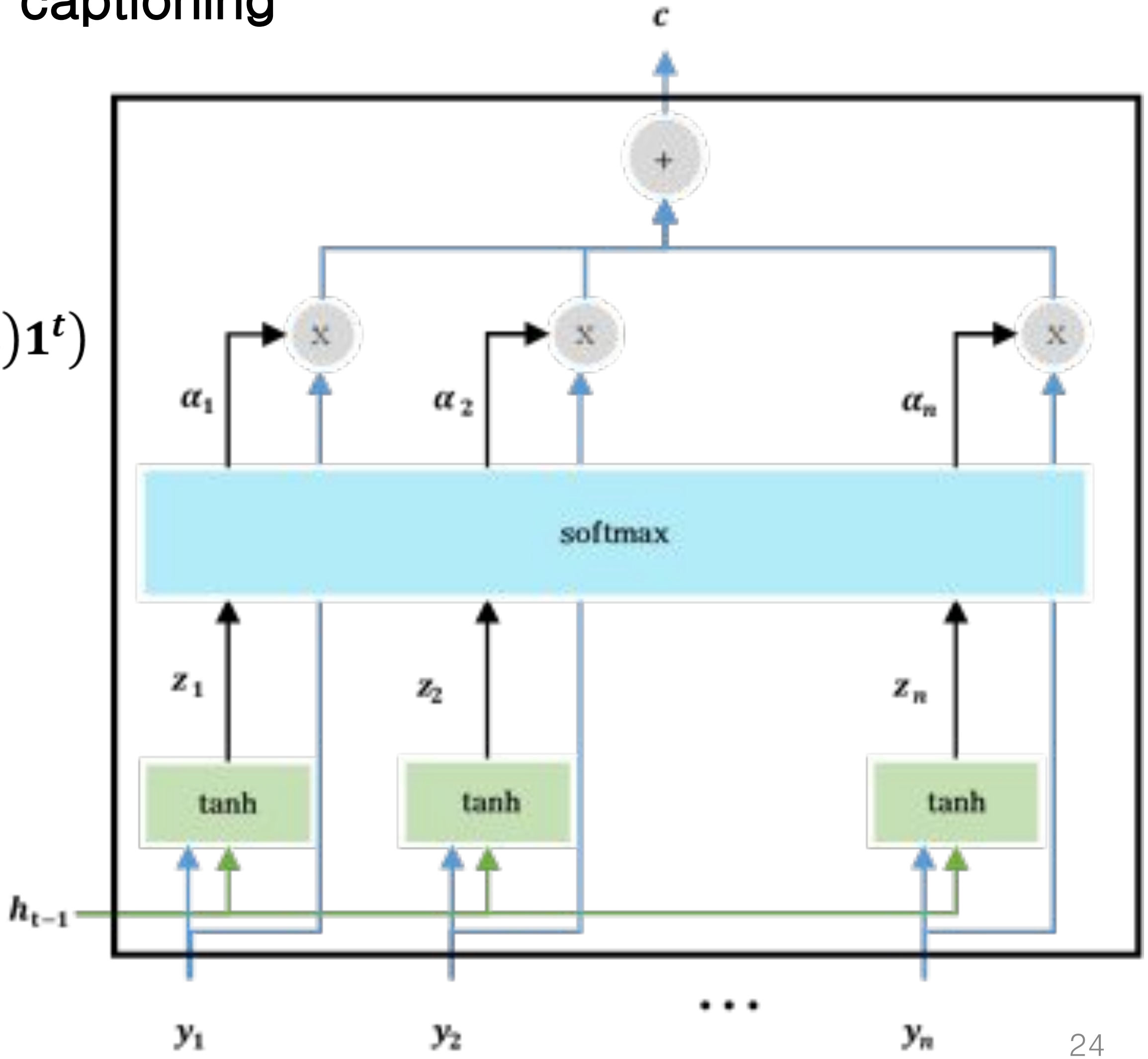
Image captioning

Attention based Image captioning
(soft attention)

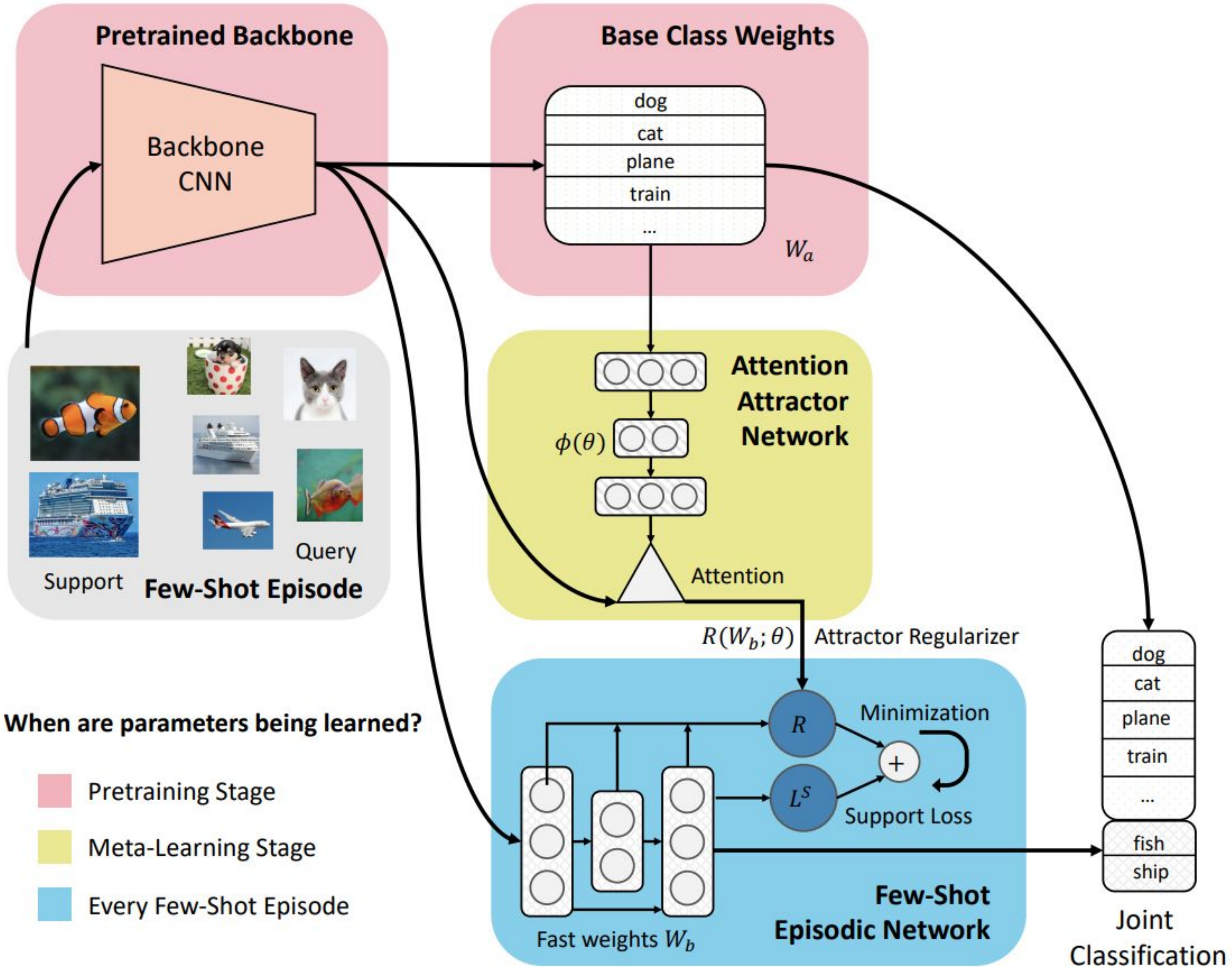
$$z_t = w_h^T \tanh\left(W_v V + (W_g h_t)1^t\right)$$

$$\alpha_t = softmax(z_t)$$

$$c_t = \sum_{i=1}^{k} \alpha_{ti} v_{ti}$$



Few shot learning



[출저]
https://papers.nips.cc/paper/8769-incremental-few-shot-learning-with-attention-attractor-networks.pdf

Reference

- https://blog.floydhub.com/attention-mechanism/
- Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).
- https://pytorch.org/tutorials/intermediate/seq2seq_translation_tutorial.html
- http://docs.likejazz.com/attention/
- https://wikidocs.net/22893
- https://towardsdatascience.com/image-captioning-in-deep-learning-9cd23fb4d8d2
- https://papers.nips.cc/paper/8769-incremental-few-shot-learning-with-attention-attractor-networks.pdf

Thank You!

Do you have any question?