

# EE-559 – Deep learning

## 13.1. Attention Mechanisms

François Fleuret  
<https://fleuret.org/ee559/>  
Feb 10, 2020



Attention mechanisms aggregate features with an importance score that

- depends on the feature themselves, not only on their position in the tensor,
- relax locality constraints.

Given a 1D convolutional map

$$x \in \mathbb{R}^{T \times D},$$

a standard approach to build an aggregated information is average pooling

$$y_j = \sum_{i=1}^T \frac{\mathbf{1}_{\{|j-i| \leq \Delta\}}}{\sum_k \mathbf{1}_{\{|j-k| \leq \Delta\}}} x_i.$$

Here the contribution of  $x_i$  to  $y_j$  is entirely driven by their locations [in the tensor]. This is a form of **spatial attention**.

For **context attention**, the importance of  $x_i$  in computing  $y_j$  is not entirely determined by  $i$  and  $j$  but by  $x_i$  and  $j$  through an attention function  $a(x_i; \theta_j)$

$$y_j = \sum_{i=1}^T \frac{e^{a(x_i; \theta_j)}}{\sum_k e^{a(x_k; \theta_j)}} x_i = \sum_{i=1}^T \text{softmax}_i(a(x_i; \theta_j)) x_i,$$

where the  $\theta_j$  are model parameters.

We differentiate this from **self-attention**, where the importance of  $x_i$  in computing  $y_j$  depends on  $x_i$  and  $x_j$

$$y_j = \sum_{i=1}^T \text{softmax}_i (a(x_i, x_j; \theta)) x_i.$$

The most standard approaches implement attention as a dot-product, e.g.

$$a(x; V) = x^T V$$

or

$$a(x, x'; W, W') = (Wx)^T (W'x')$$

for self-attention.

However, the quantity  $a$  can take any form, e.g. an MLP.

In what follows there is often an implicit “channel dimension” and the matrix product operates on the channels at every “location” of every sample.

With

$$W \in \mathbb{R}^{c \times d}$$

and  $x$  a  $d$ -channel tensor

$$x \in \mathbb{R}^{\mathcal{A} \times d \times \mathcal{B}}$$

the resulting tensor has same shape with  $c$  channels

$$Wx \in \mathbb{R}^{\mathcal{A} \times c \times \mathcal{B}}.$$

## Attention for seq2seq

Attention mechanisms were re-introduced for deep learning to provide long-term dependency for sequence-to-sequence translation.

For such a task, given an input sequence  $x_1, \dots, x_T$ , the standard approach (Sutskever et al., 2014) is to use a recurrent model

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

and to consider that the final hidden state

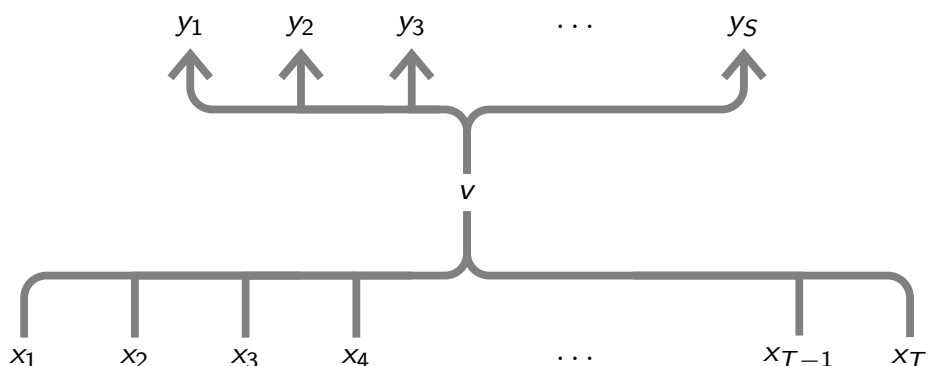
$$v = h_T$$

carries enough information to drive an auto-regressive generative model

$$y_t \sim p(y_1, \dots, y_{t-1}, v),$$

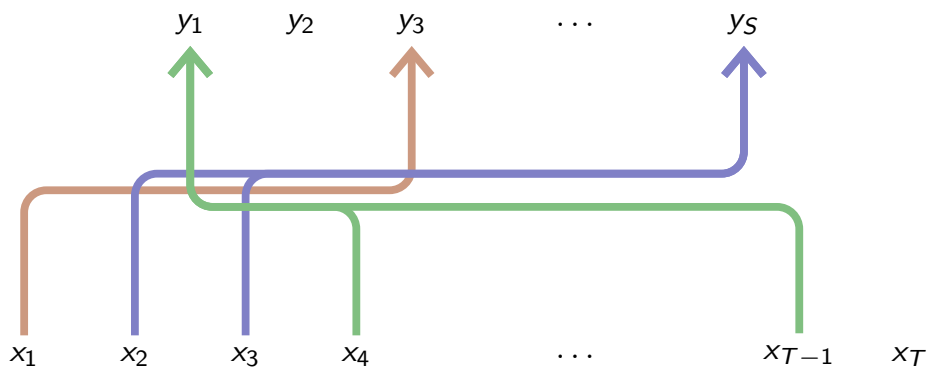
itself implemented with another RNN.

The main weakness of such an approach is that all the information has to flow through a single state  $v$ , whose capacity has to accommodate any situation.



There are no direct “channels” to transport local information from the input sequence to the place where it is useful in the resulting sequence.

Attention mechanisms (Graves et al., 2014; Bahdanau et al., 2014) transport information from parts of the signal to other parts **specified dynamically**.



Bahdanau et al. (2014) propose to extend a standard recurrent model with such a mechanism. They first run a bi-directional RNN to get a hidden state

$$h_i = (h_i^{\rightarrow}, h_i^{\leftarrow}), \quad i = 1, \dots, T.$$

From this, they compute a new process  $s_i, i = 1, \dots, T$  which looks at weighted averages of the  $h_j$ , where **the weight are functions of the signal**.

Given  $y_1, \dots, y_{i-1}$  and  $s_1, \dots, s_{i-1}$  first compute an attention

$$\forall j, e_{i,j} = a(s_{i-1}, h_j)$$

where  $a$  is a one hidden layer tanh MLP (this is “additive attention”, or “concatenation”).

Normalize it into a distribution with a standard softmax

$$\forall j, \alpha_{i,j} = \frac{\exp(e_{i,j})}{\sum_{k=1}^T \exp(e_{i,k})},$$

and compute the **context vector** from the  $h$ s to predict  $y_i$

$$c_i = \sum_{j=1}^T \alpha_{i,j} h_j.$$

The model can now make the prediction

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

$$y_i \sim g(y_{i-1}, s_i, c_i)$$

where  $f$  is a GRU (Cho et al., 2014).

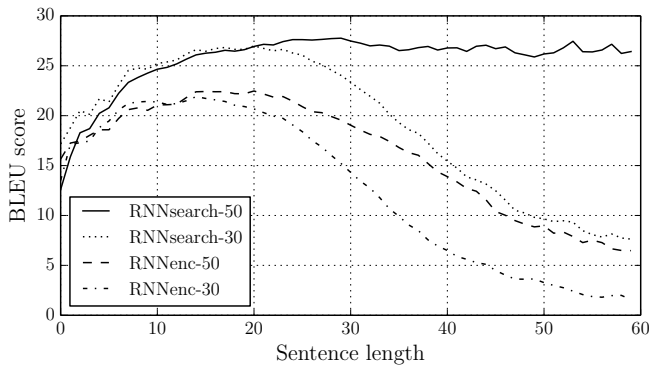
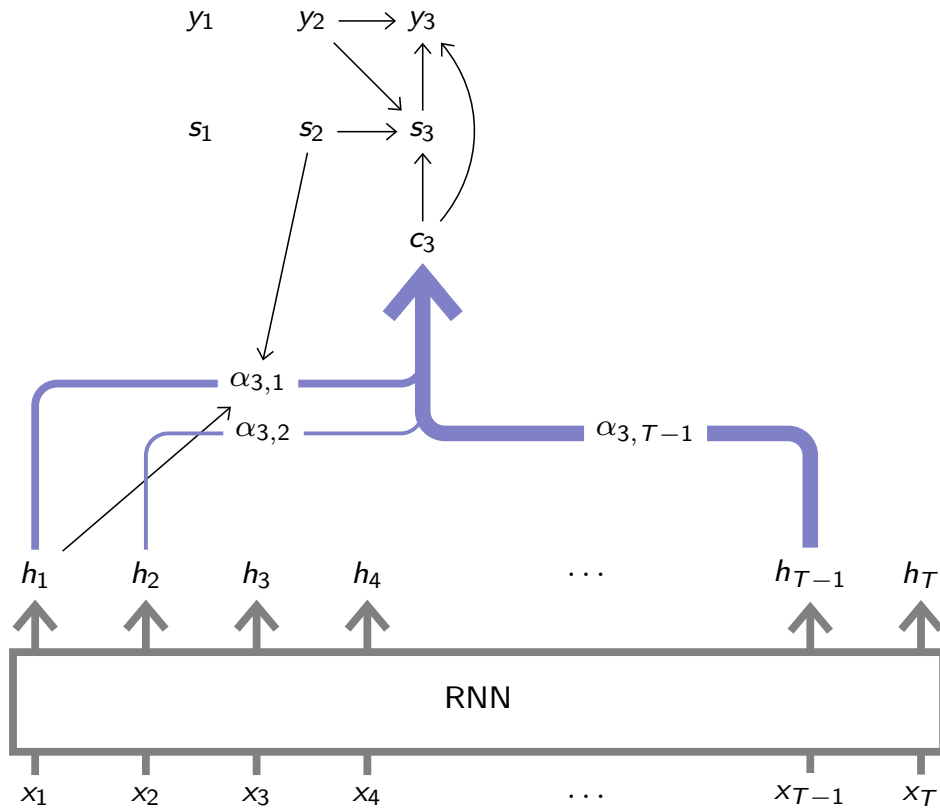
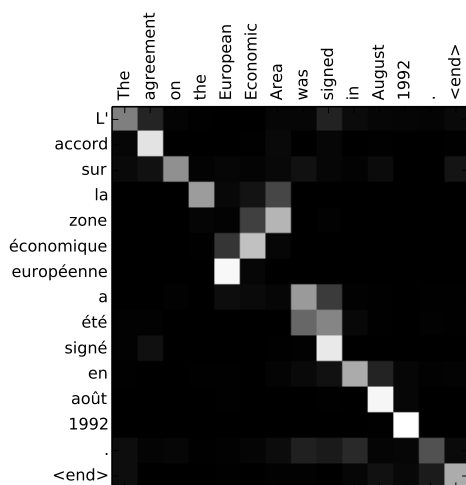


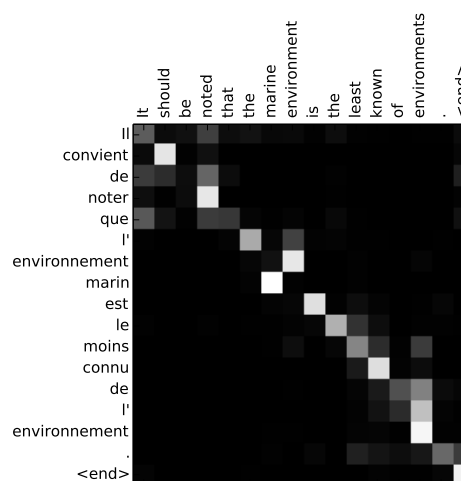
Figure 2: The BLEU scores of the generated translations on the test set with respect to the lengths of the sentences. The results are on the full test set which includes sentences having unknown words to the models.

(Bahdanau et al., 2014)





(a)



(b)

(Bahdanau et al., 2014)

## References

- D. Bahdanau, K. Cho, and Y. Bengio. **Neural machine translation by jointly learning to align and translate.** *CoRR*, abs/1409.0473, 2014.
- K. Cho, B. van Merriënboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio. **Learning phrase representations using RNN encoder-decoder for statistical machine translation.** *CoRR*, abs/1406.1078, 2014.
- A. Graves, G. Wayne, and I. Danihelka. **Neural Turing machines.** *CoRR*, abs/1410.5401, 2014.
- I. Sutskever, O. Vinyals, and Q. V. Le. **Sequence to sequence learning with neural networks.** In *Neural Information Processing Systems (NIPS)*, pages 3104–3112, 2014.