Recurrent Neural Networks

INSTRUCTOR: HONGJIE CHEN

JUNE 23RD 2022

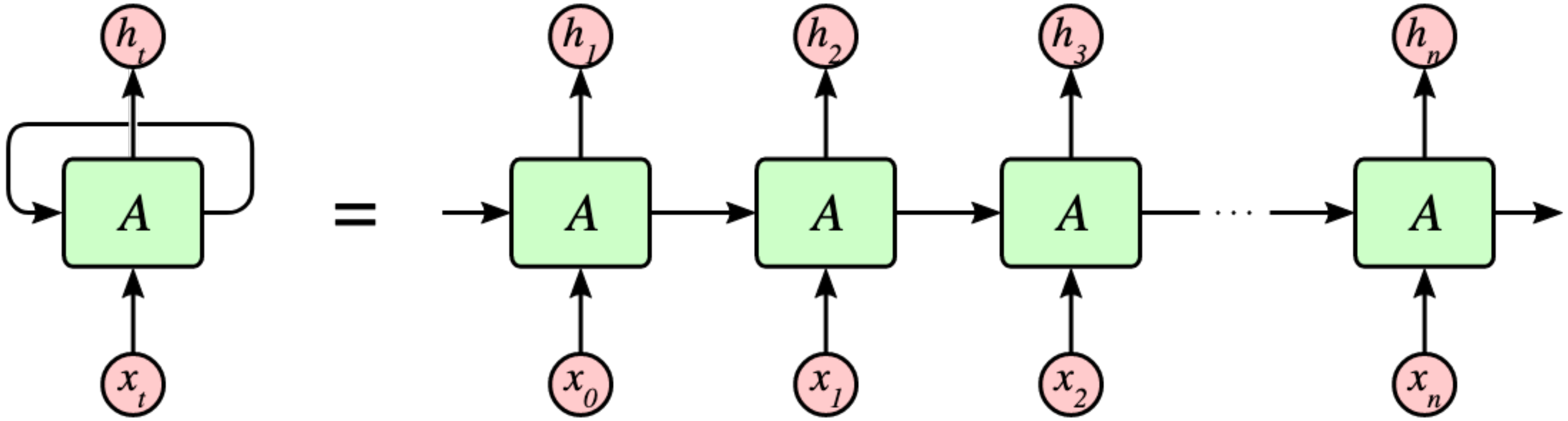
Variable Length Data

- Fixed length data

- Variable length data
  - Time-series, for example?
  - Sequential data

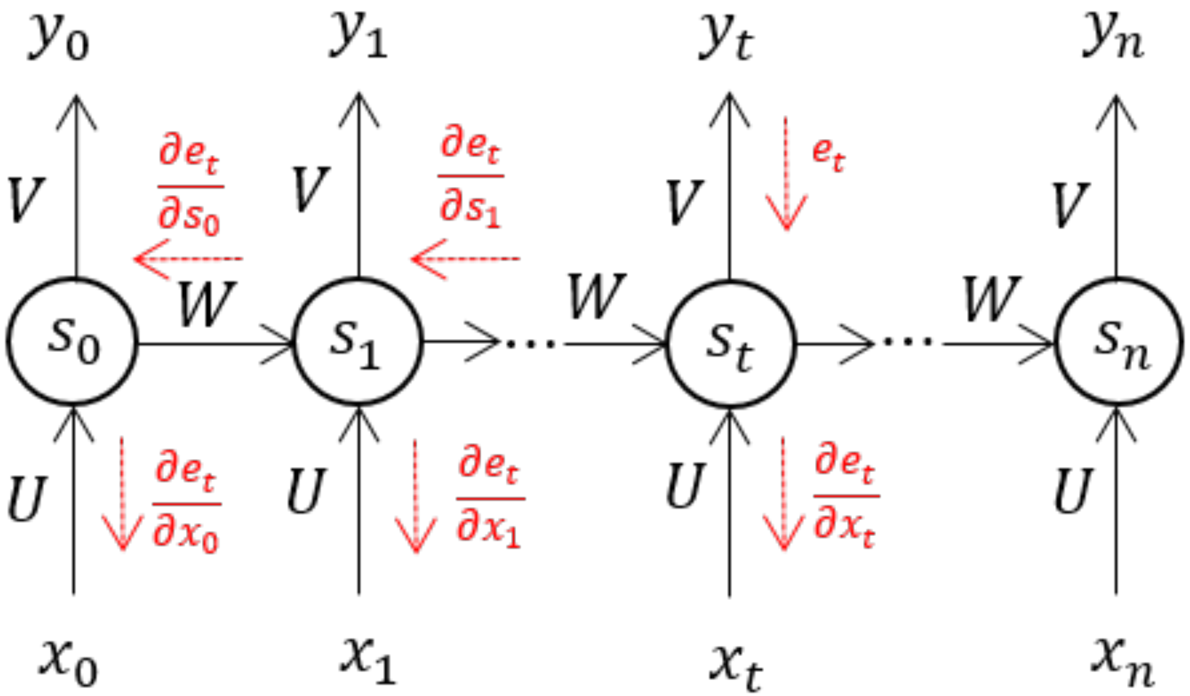
RNN

- Recurrent Neural Networks (RNN) consumes sequential data by encoding data into a hidden state



Propagation in RNN

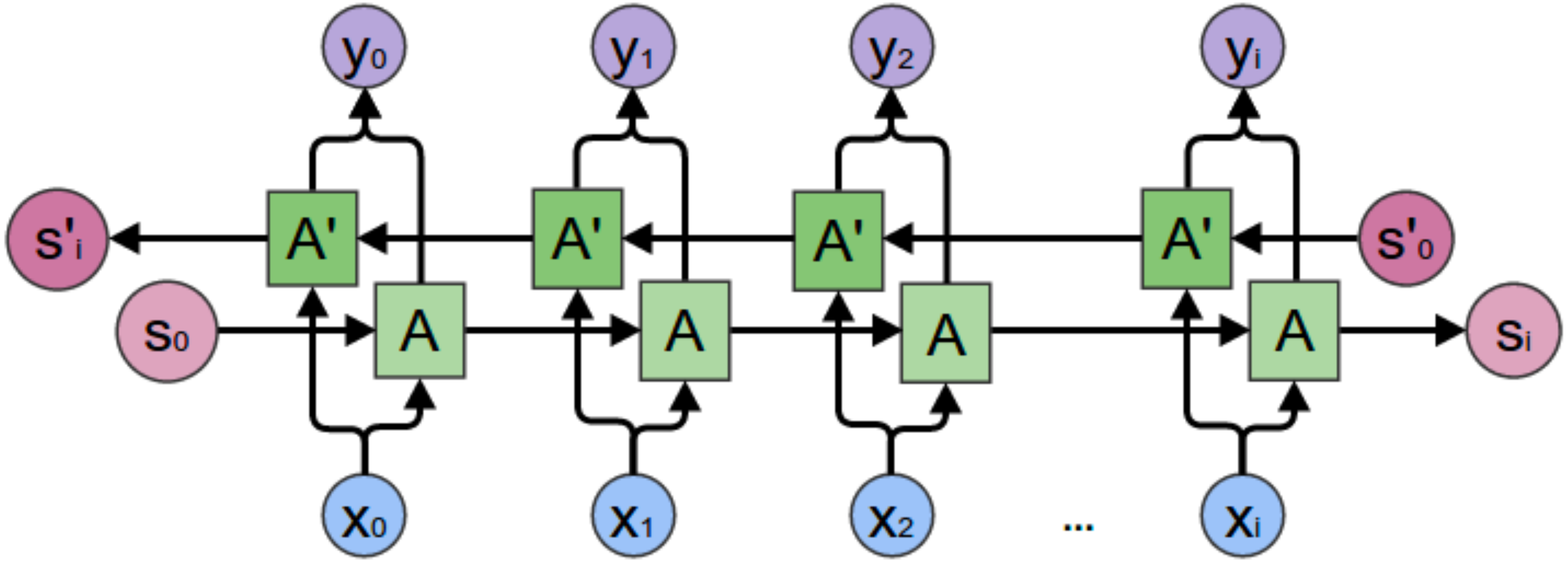
- Computation in an unrolled RNN



VIRGINIA TECH

Imputation v.s. Prediction

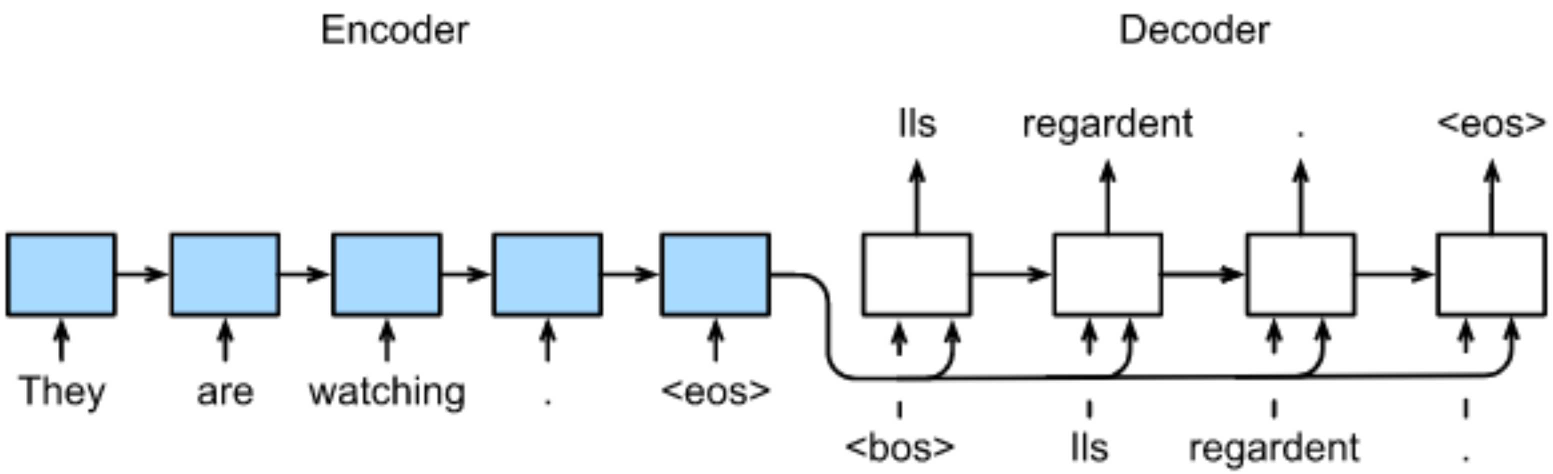
- In some application, we have observations from the past and future, and we want to imputate a state in between
- Bi-directional RNN



Encoder-decoder Architecture

- A sequence to sequence model (seq2seq)

- Learn an embedding that serves as the input for the decoder
- Machine translation

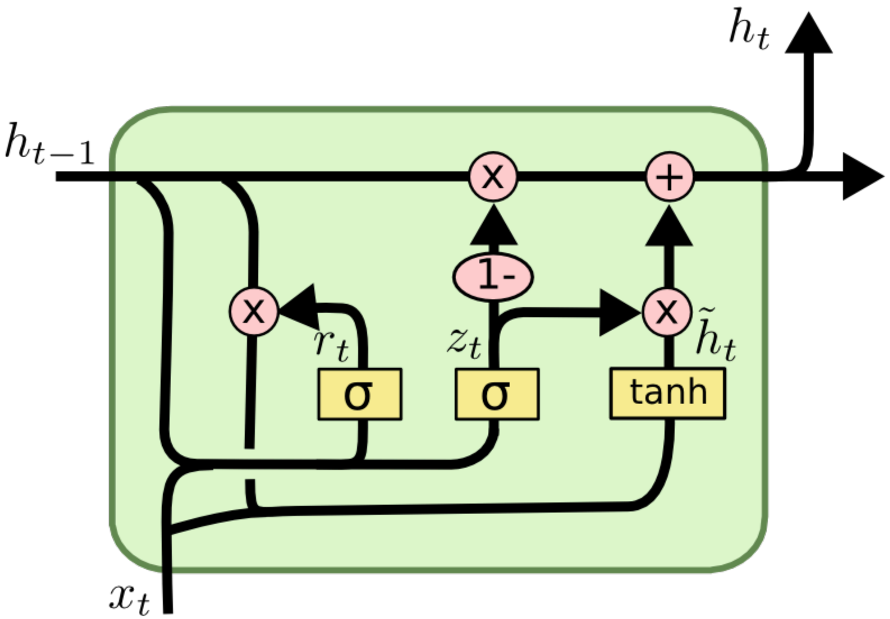


Training RNN

- Backpropagation on the unrolled network
  - All given time steps
- Challenges
  - Vanishing and explosion gradient
  - Long range memory
  - Prediction drift
- Use Demo: link

Popular RNN variants

- LSTM and **GRU**, specific structure design
- Gated structure to control memorization and forgetting



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

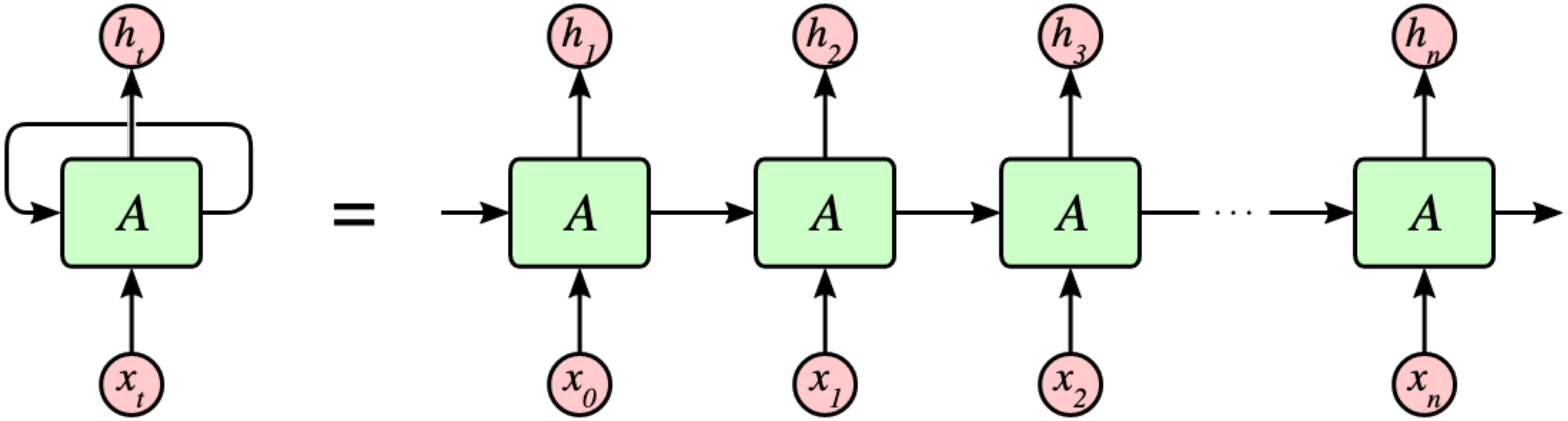
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

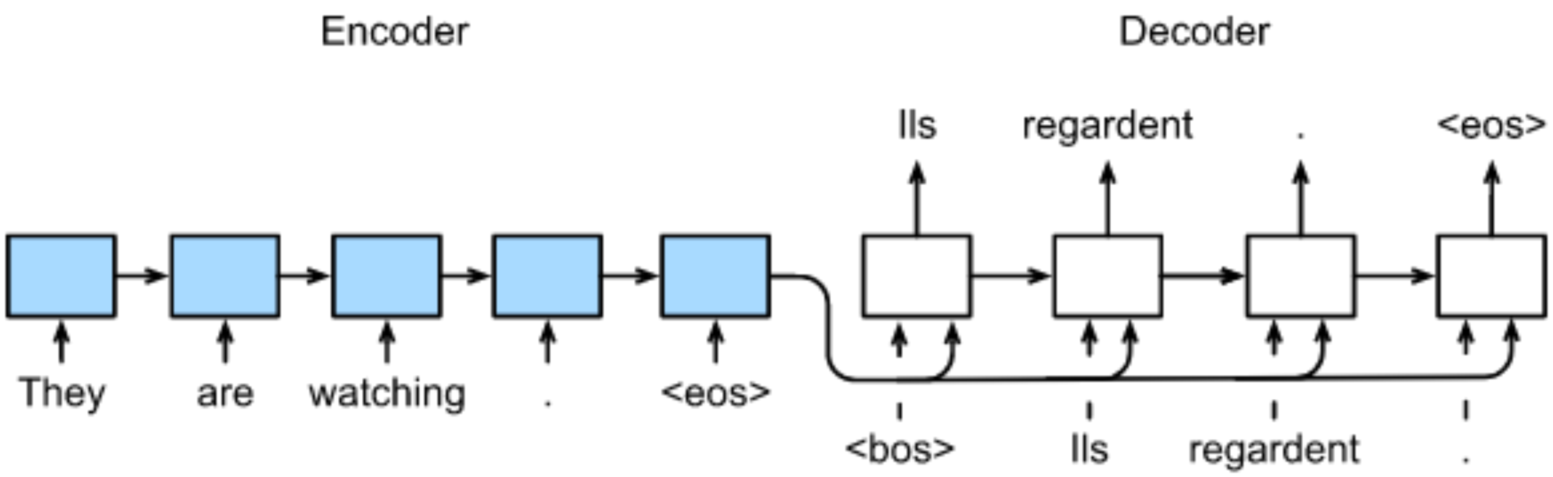
Unrolled with LSTM or GRU

- A is LSTM or GRU



Recall encoder-decoder structure

- The learned embedding is fed to all unrolled steps in decoder.

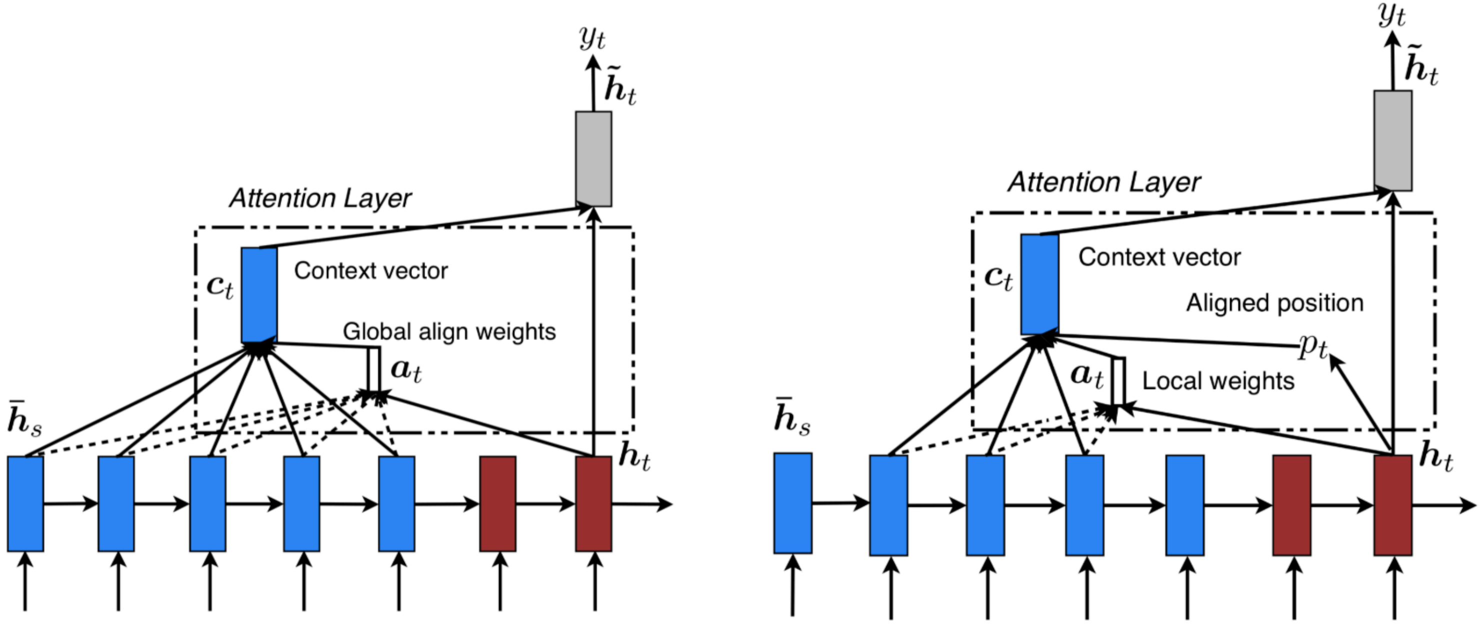


- However, may care less about history from far

VIRGINIA TECH.

Attention

- Parameters that are used to highlight important features



Global Attention Model **Local Attention Model**

- Can also be incoporeated for computer vision tasks

General Attention

- A $(query, key, value)$ attention mechanism
- $$attention(q, \mathbf{k}, \mathbf{v}) = \sum_i similarity(q, k_i) \times v_i$$
- And more

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

- Optional reading: link