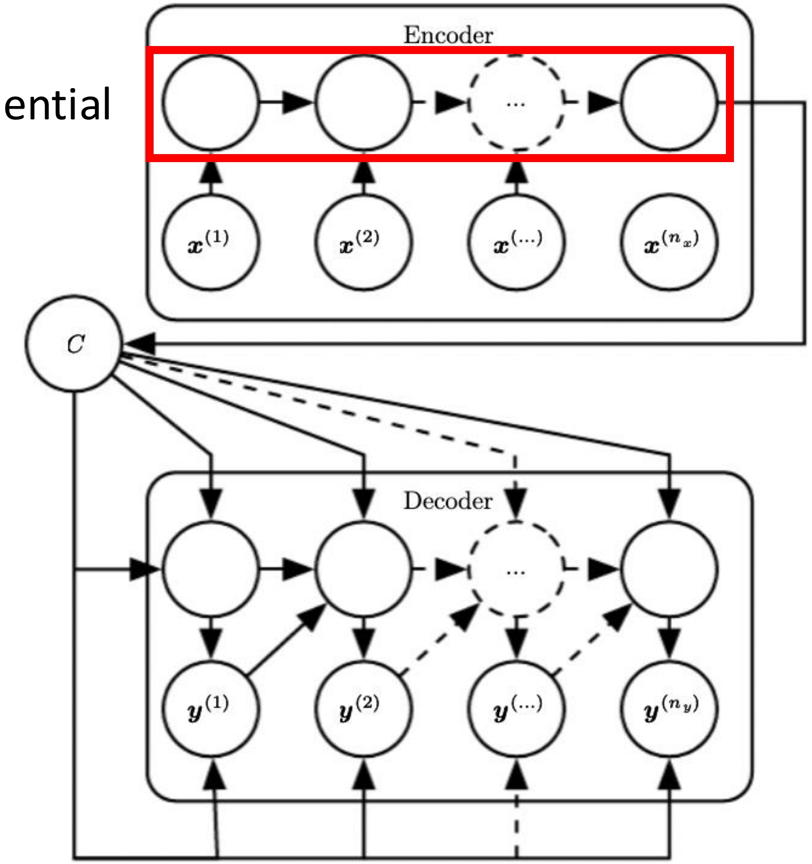
Attention and Transformers

Neural Networks Design And Application

Encoder-decoder

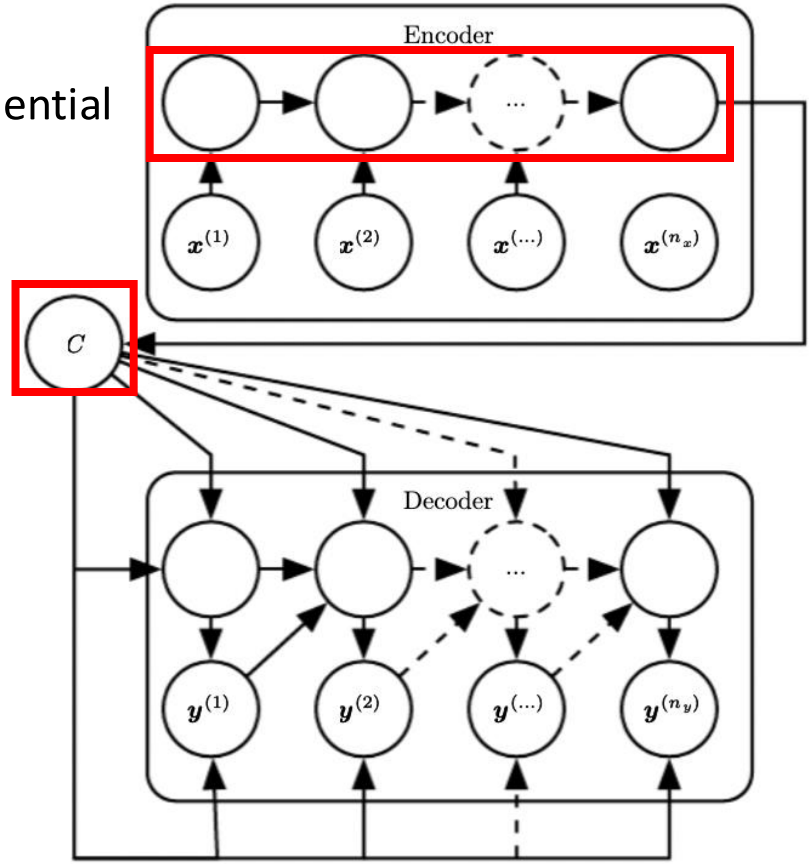
sequential



Encoder-decoder

sequential

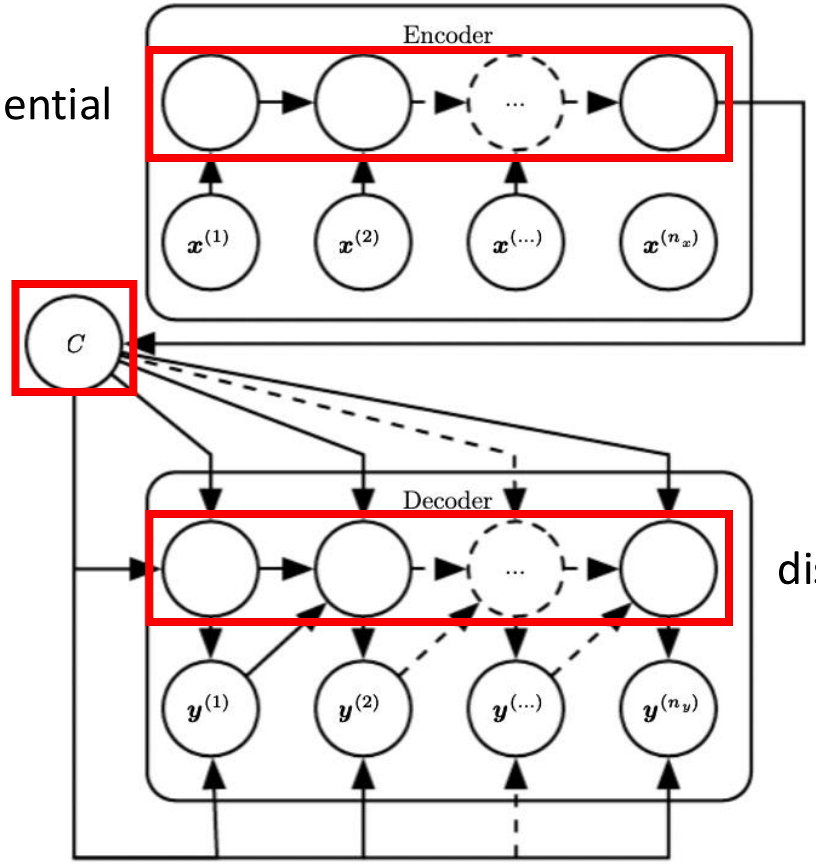
summary: context feature



Encoder-decoder

sequential

summary: context feature

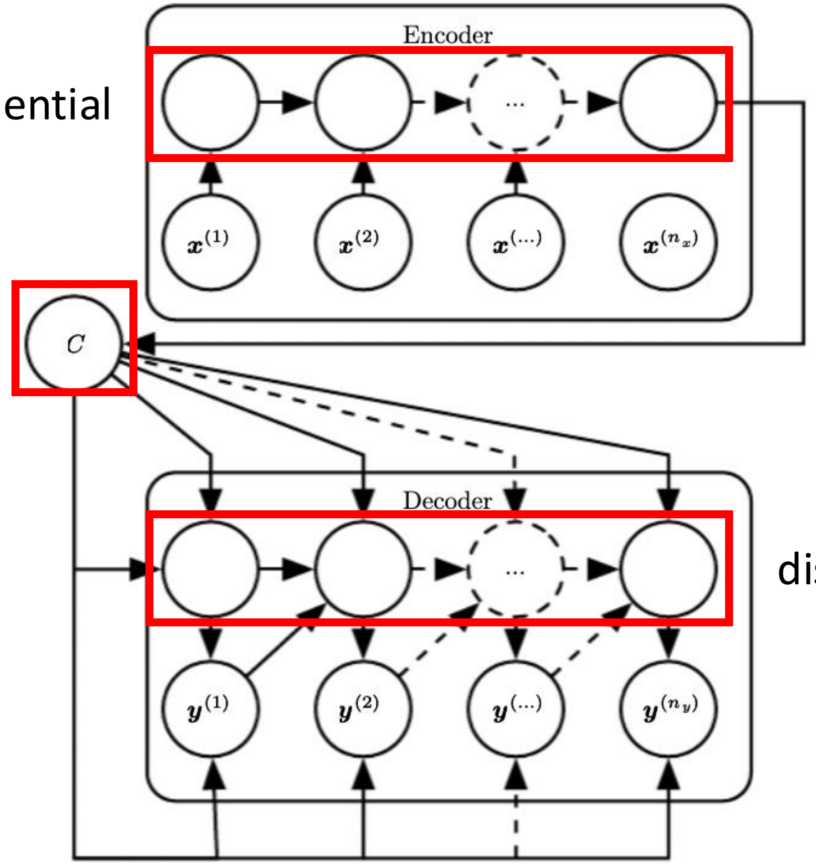


distribute information

Encoder-decoder

sequential

summary: context feature



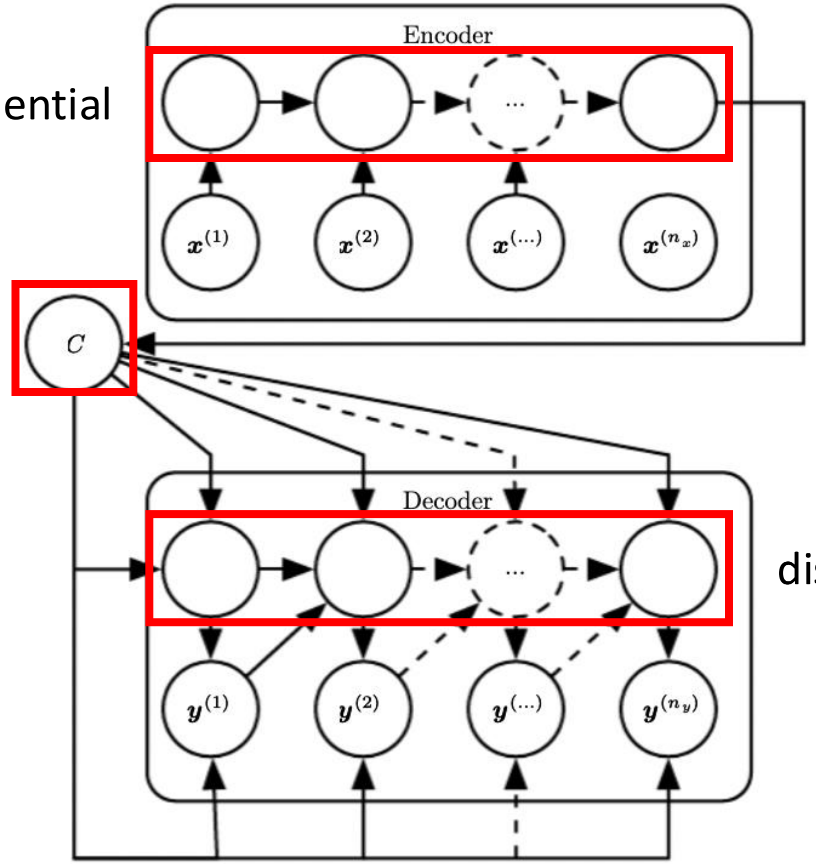
distribute information

Q: can we use bi-directional RNN for input?

Encoder-decoder

sequential

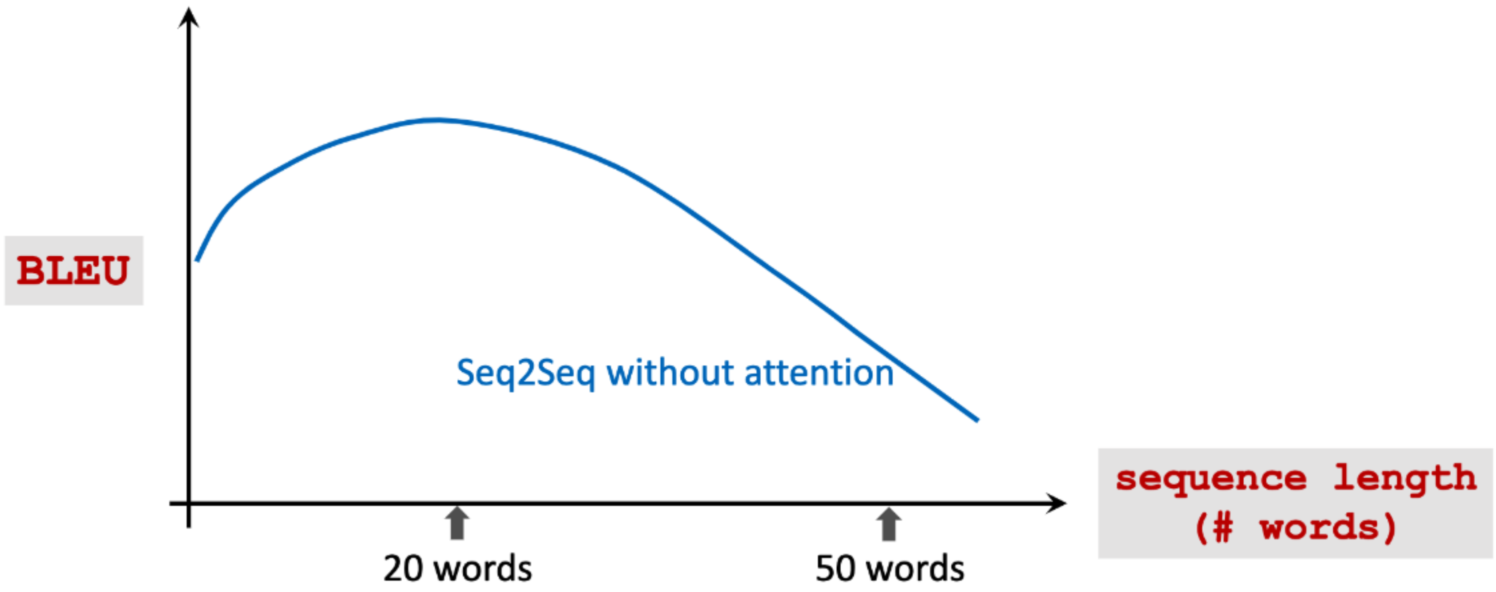
summary: context feature



distribute information

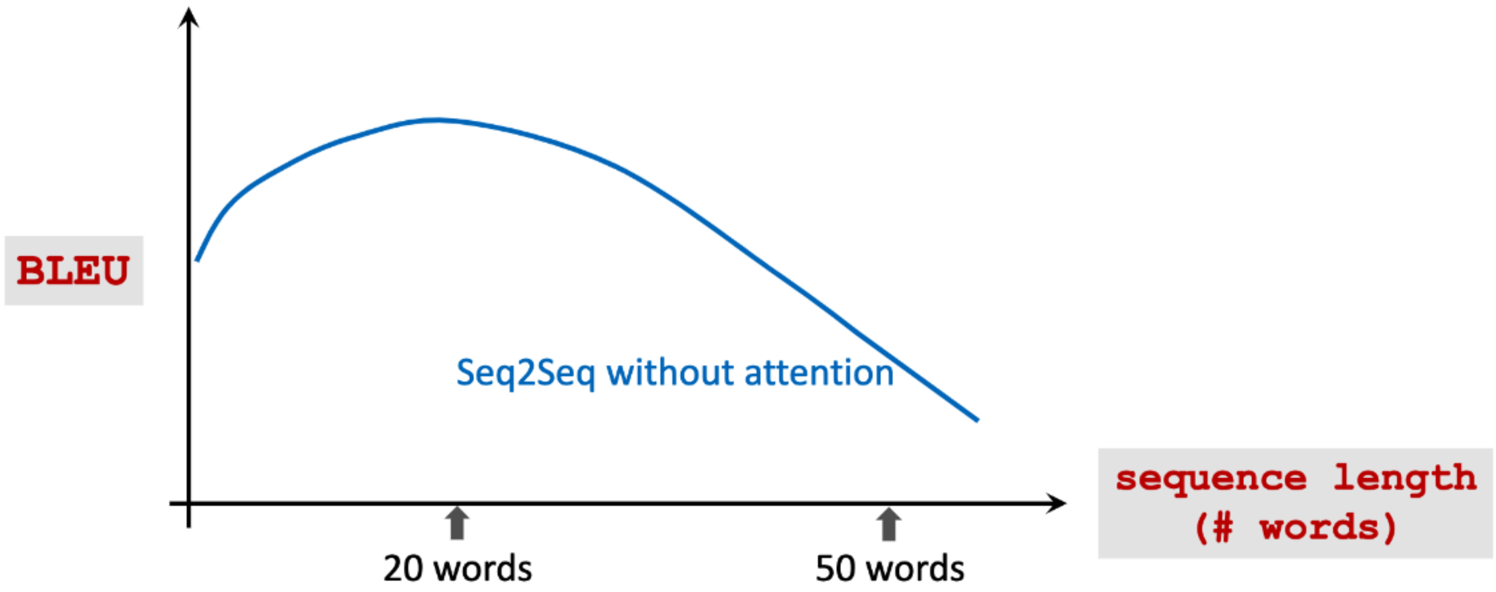
Q: can we use bi-directional RNN for input?

Seq2seq model performance



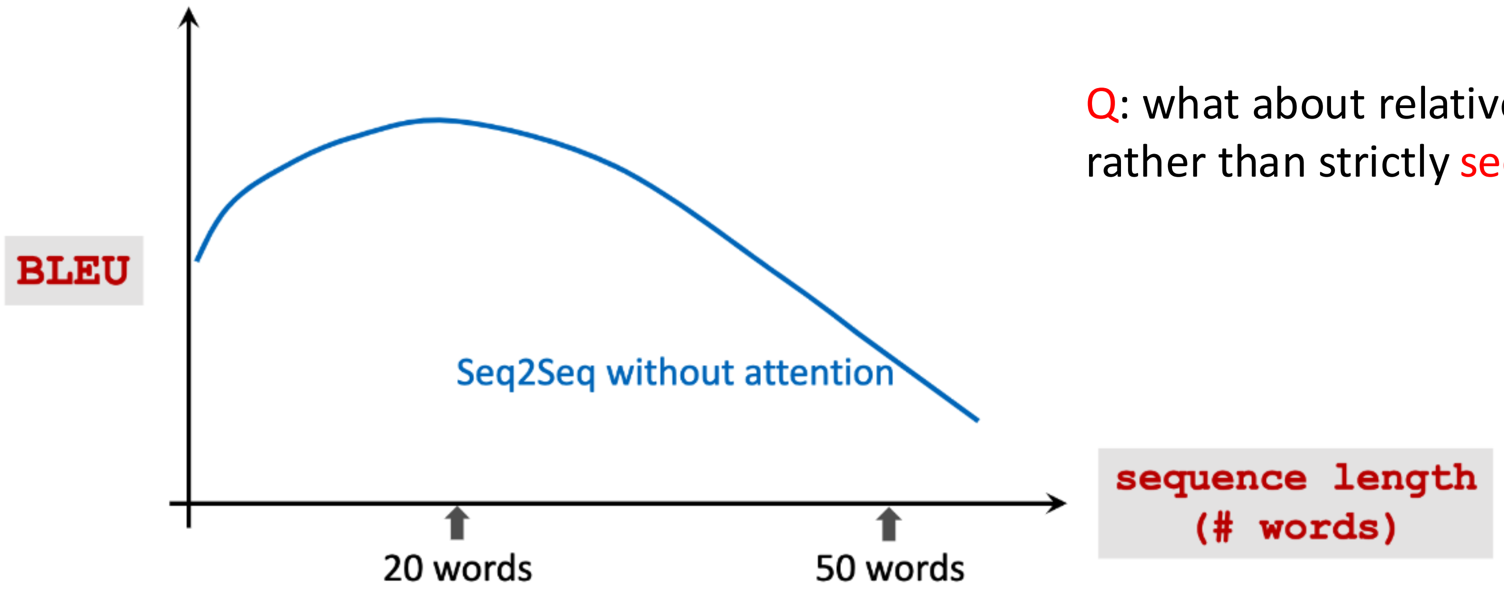
the clouds are in the sky

Seq2seq model performance



I like this town very much. I started my undergraduate study in 2020 and my major is computer science. I like programming and reading. I usually get up at 7AM and do some exercise. I also go fishing at weekend. I grew up in France. I spent my childhood outdoors. Whether it was riding my bicycle around my neighborhood pretending it was a motorcycle, making mud cakes, going on treasure hunts, making and selling perfume out of strong smelling flowers, or simply laying on the grass underneath the sun with a soccer ball waiting for someone to come out and play with me, the outdoors was where I spent my childhood and I cannot be more appreciative of it. I speak fluent *French*.

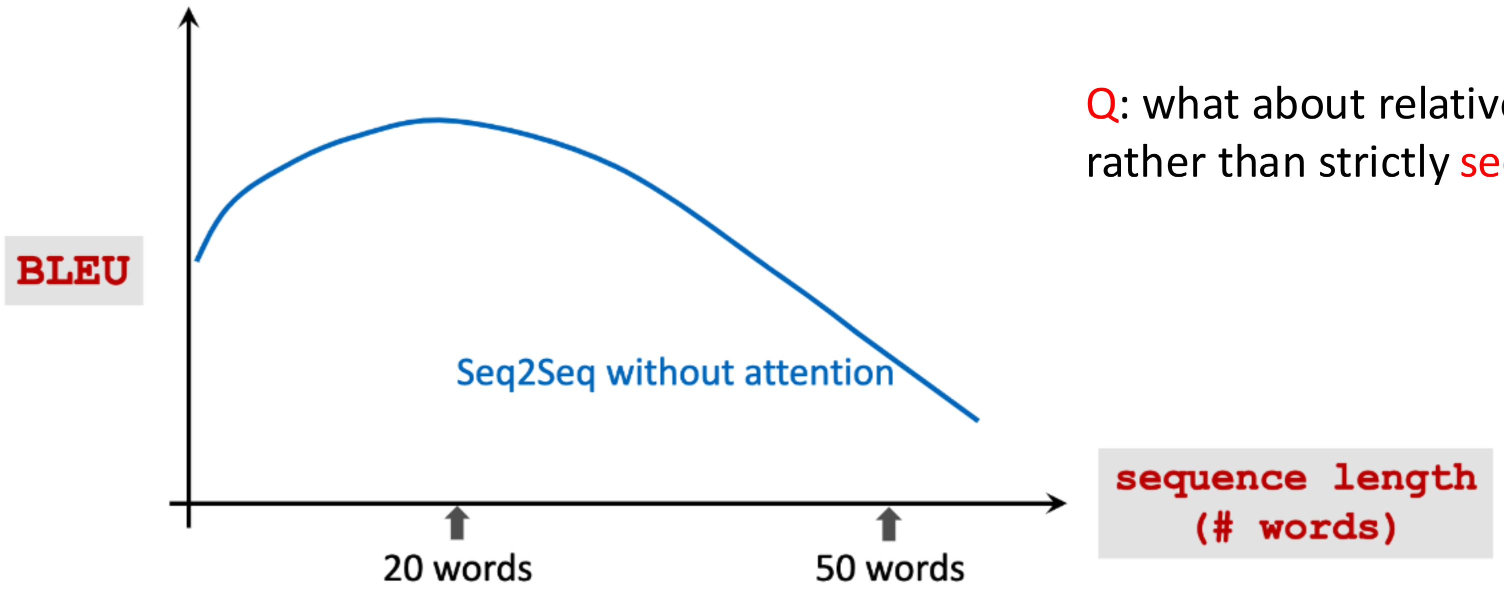
Seq2seq model performance



Q: what about relatively parallel correlation rather than strictly sequential correlation?

I like this town very much. I started my undergraduate study in 2020 and my major is computer science. I like programming and reading. I usually get up at 7AM and do some exercise. I also go fishing at weekend. I grew up in France. I spent my childhood outdoors. Whether it was riding my bicycle around my neighborhood pretending it was a motorcycle, making mud cakes, going on treasure hunts, making and selling perfume out of strong smelling flowers, or simply laying on the grass underneath the sun with a soccer ball waiting for someone to come out and play with me, the outdoors was where I spent my childhood and I cannot be more appreciative of it. I speak fluent *French*.

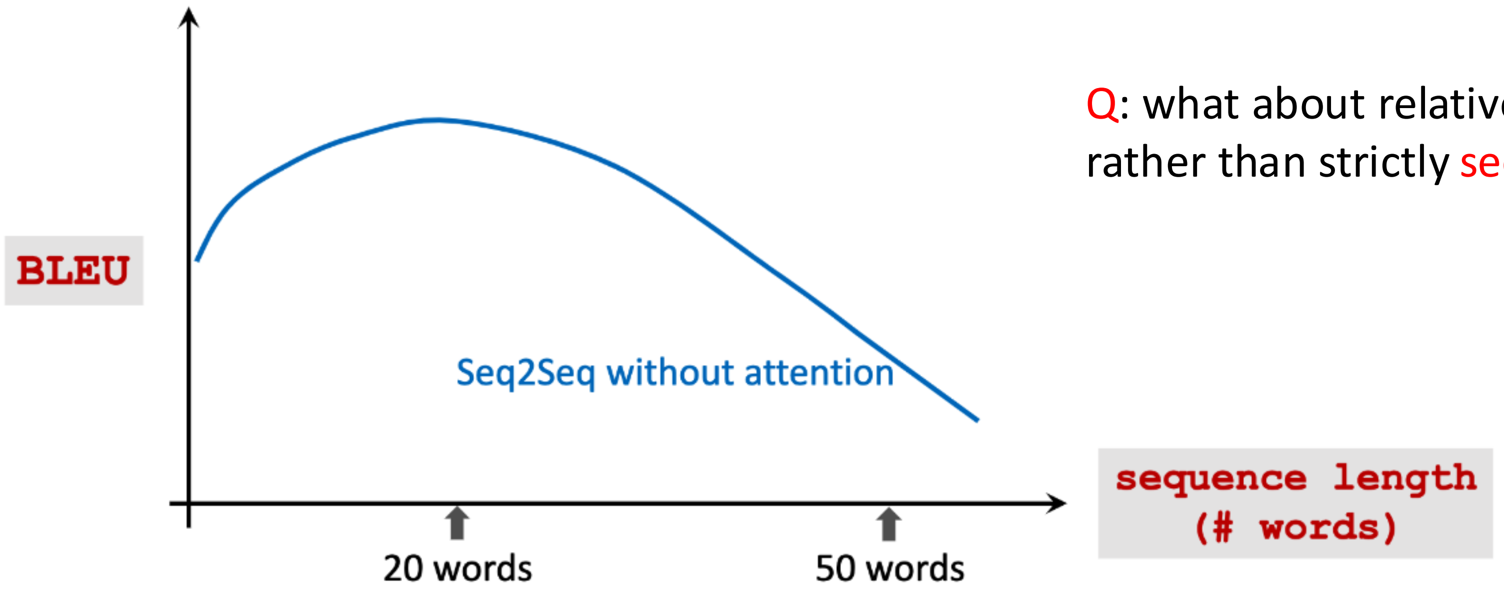
Seq2seq model performance



Q: what about relatively parallel correlation rather than strictly sequential correlation?

I like this town very much. I started my undergraduate study in 2020 and my major is computer science. I like programming and reading. I usually get up at 7AM and do some exercise. I also go fishing at weekend. I grew up in France. I spent my childhood outdoors. Whether it was riding my bicycle around my neighborhood pretending it was a motorcycle, making mud cakes, going on treasure hunts, making and selling perfume out of strong smelling flowers, or simply laying on the grass underneath the sun with a soccer ball waiting for someone to come out and play with me, the outdoors was where I spent my childhood and I cannot be more appreciative of it. I speak fluent *French*.

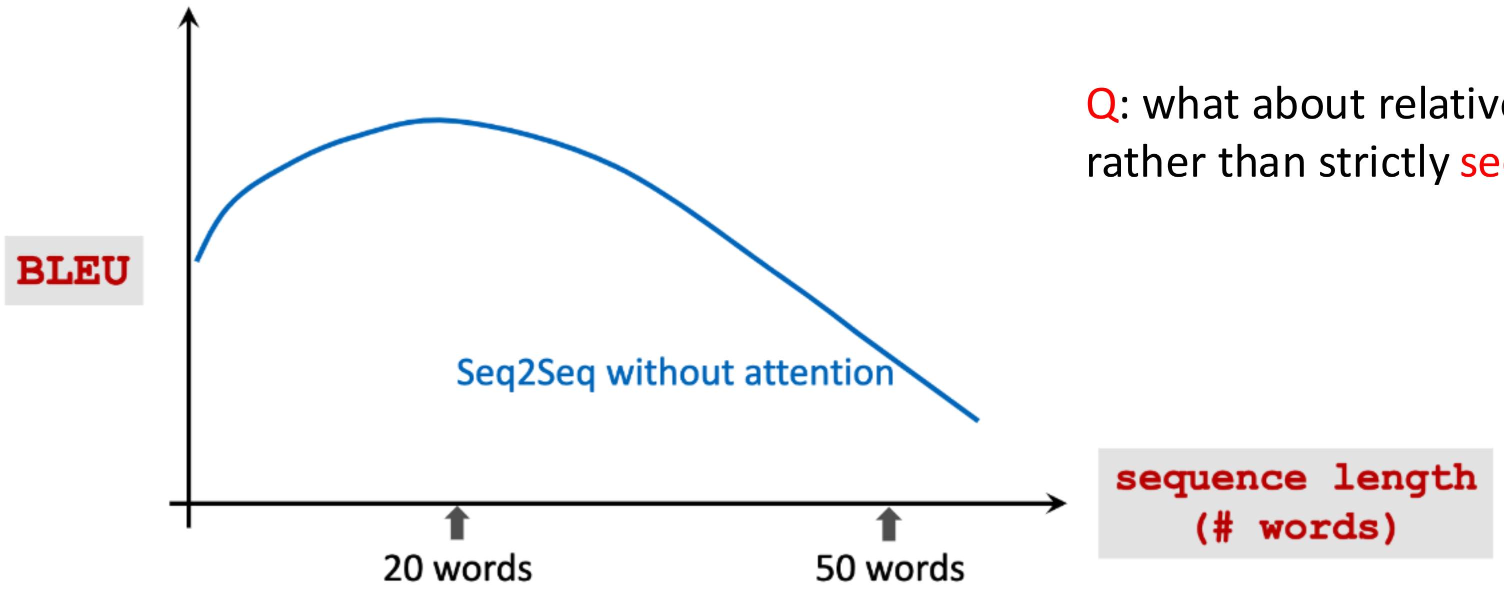
Seq2seq model performance



Q: what about relatively parallel correlation rather than strictly sequential correlation?

I like this town very much. I started my undergraduate study in 2020 and my major is computer science. I like programming and reading. I usually get up at 7AM and do some exercise. I also go fishing at weekend. I grew up in France. I spent my childhood outdoors. Whether it was riding my bicycle around my neighborhood pretending it was a motorcycle, making mud cakes, going on treasure hunts, making and selling perfume out of strong smelling flowers, or simply laying on the grass underneath the sun with a soccer ball waiting for someone to come out and play with me, the outdoors was where I spent my childhood and I cannot be more appreciative of it. I speak fluent *French*.

Seq2seq model performance

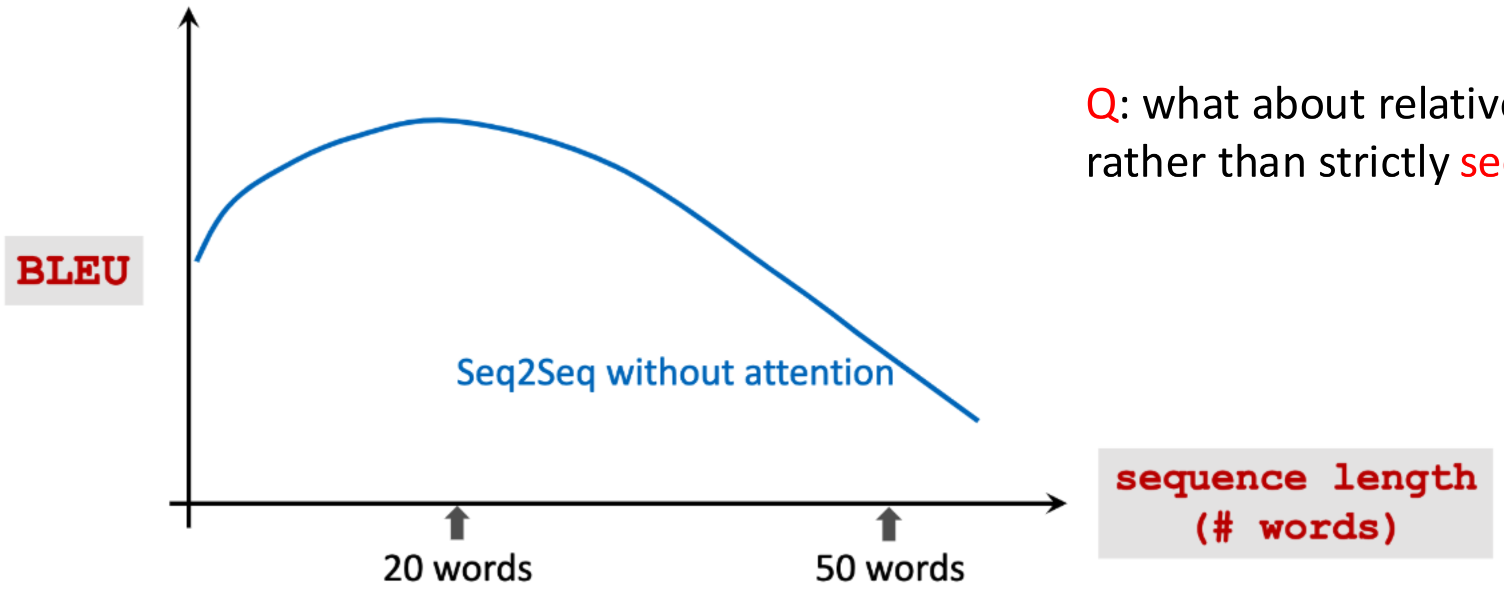


Q: what about relatively parallel correlation rather than strictly sequential correlation?

We need to summarize all context

I like this town very much. I started my undergraduate study in 2020 and my major is computer science. I like programming and reading. I usually get up at 7AM and do some exercise. I also go fishing at weekend. I grew up in France. I spent my childhood outdoors. Whether it was riding my bicycle around my neighborhood pretending it was a motorcycle, making mud cakes, going on treasure hunts, making and selling perfume out of strong smelling flowers, or simply laying on the grass underneath the sun with a soccer ball waiting for someone to come out and play with me, the outdoors was where I spent my childhood and I cannot be more appreciative of it. I speak fluent *French*.

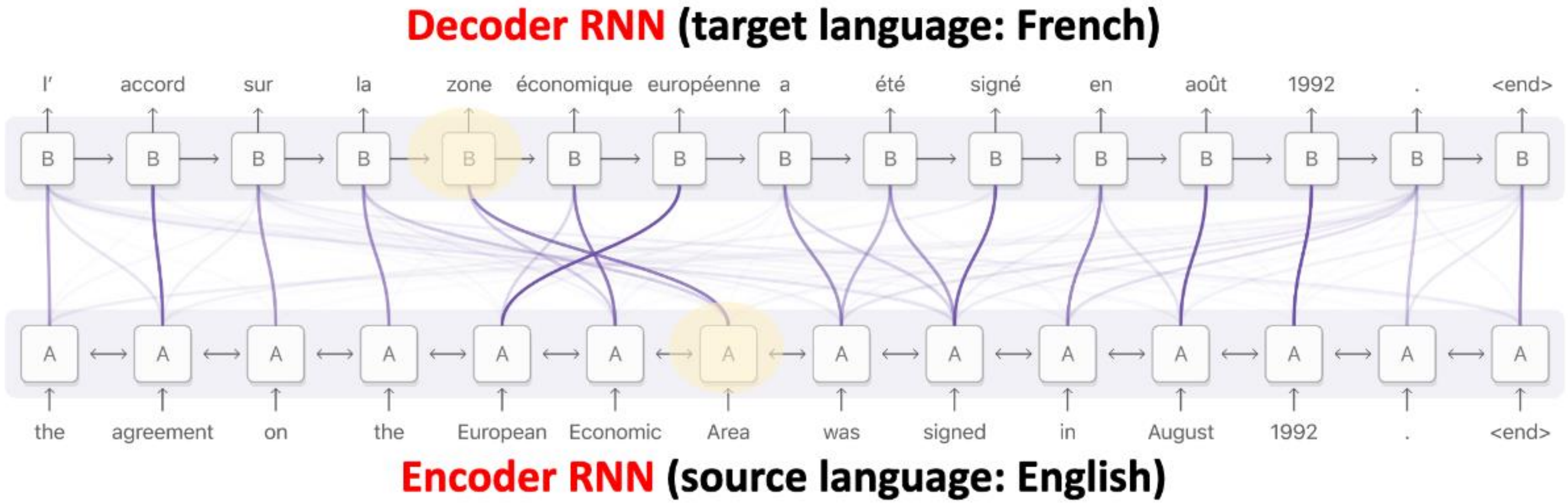
Seq2seq model performance



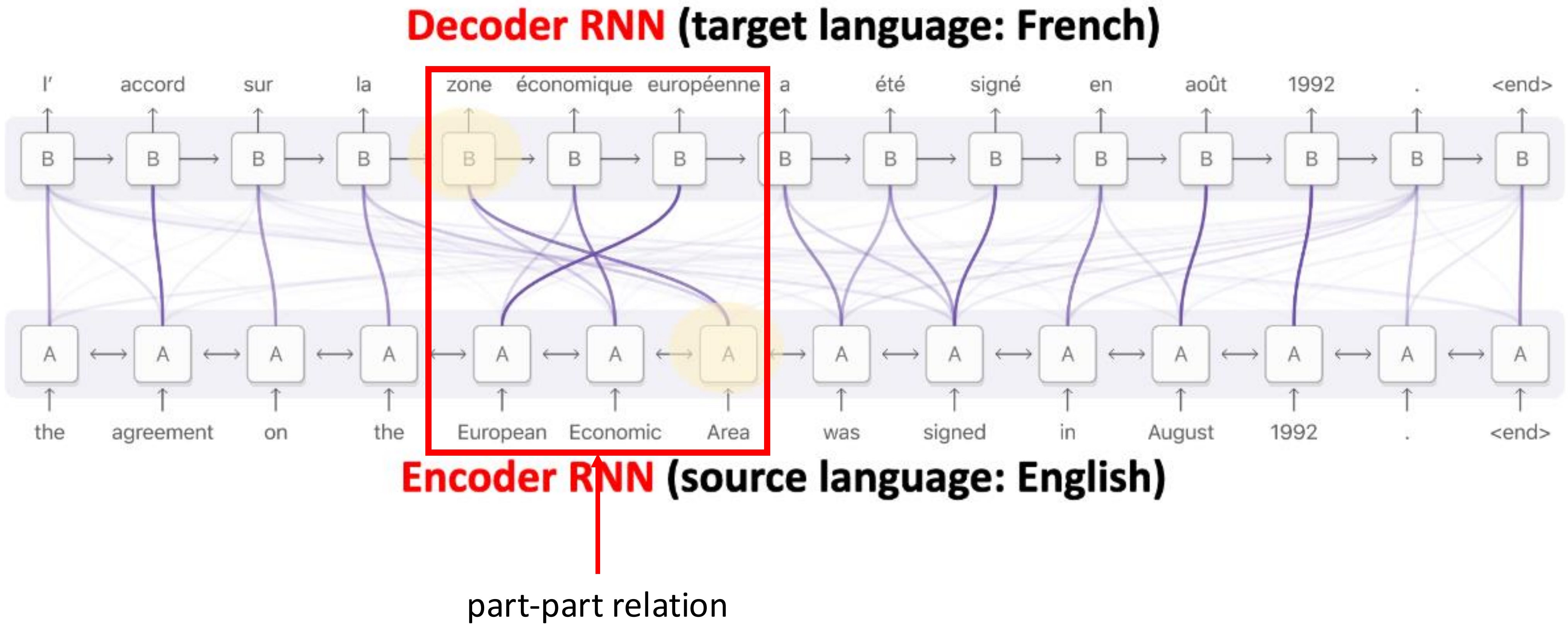
Q: what about relatively parallel correlation rather than strictly sequential correlation?

I like this town very much. I started my undergraduate study in 2020 and my major is computer science. I like programming and reading. I usually get up at 7AM and do some exercise. I also go fishing at weekend. I grew up in France. I spent my childhood outdoors. Whether it was riding my bicycle around my neighborhood pretending it was a motorcycle, making mud cakes, going on treasure hunts, making and selling perfume out of strong smelling flowers, or simply laying on the grass underneath the sun with a soccer ball waiting for someone to come out and play with me, the outdoors was where I spent my childhood and I cannot be more appreciative of it. I speak fluent *French*.

Input-output correlation

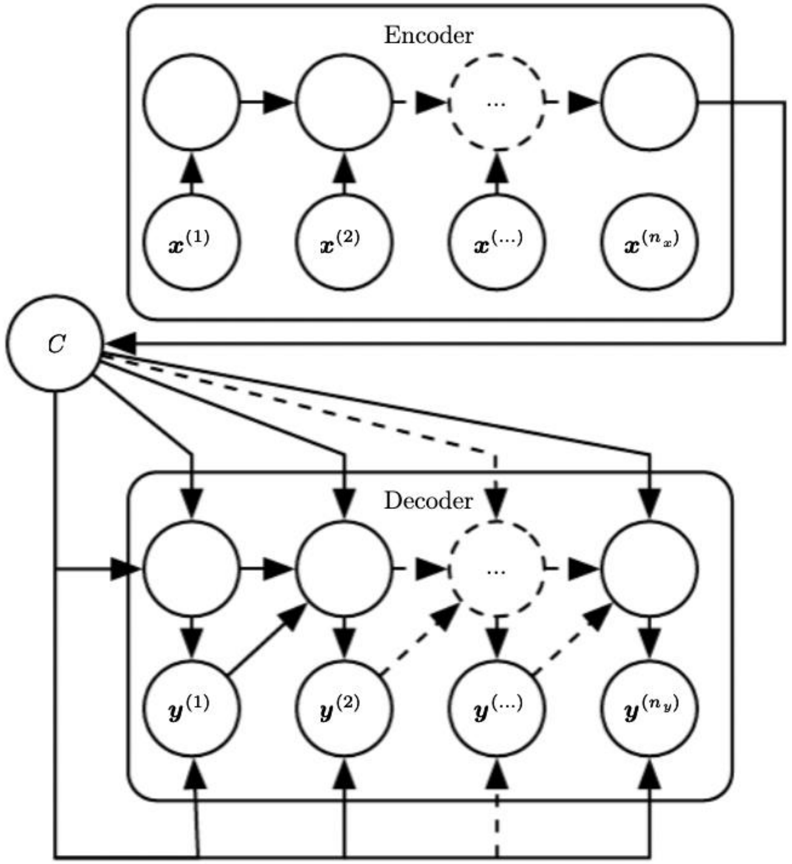


Input-output correlation



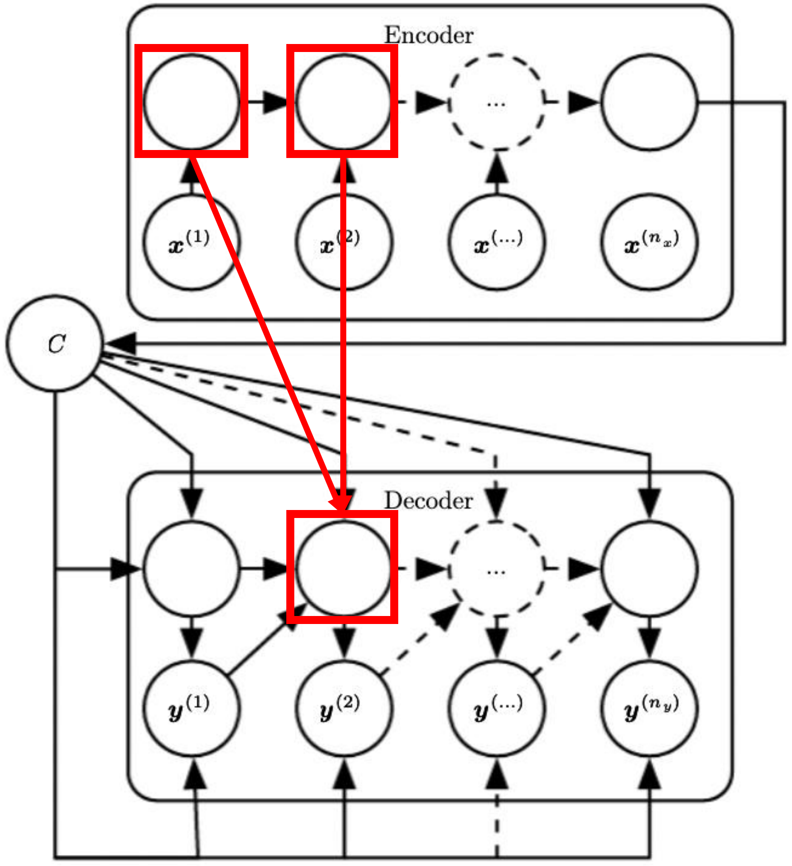
Encoder-decoder

Q: can we create information flow between encoder and decoder nodes?



Encoder-decoder

Q: can we create information flow between encoder and decoder nodes?



Attention mechanism

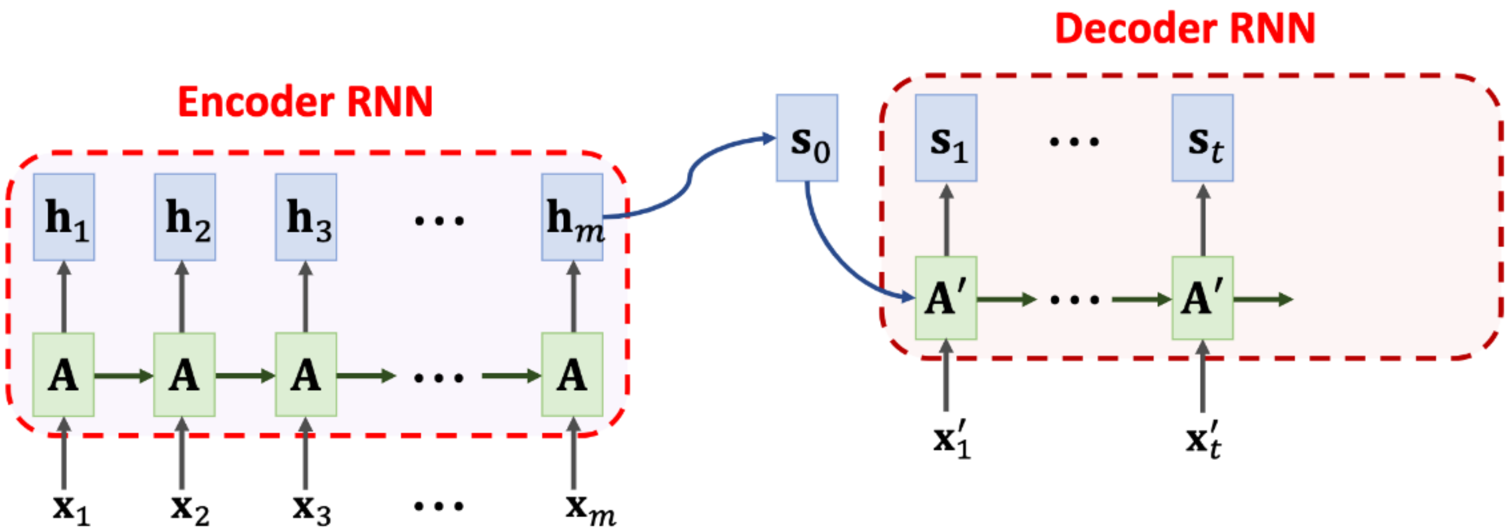


Image from https://github.com/wangshusen/DeepLearning/blob/master/Slides/9_RNN_8.pdf

Attention mechanism

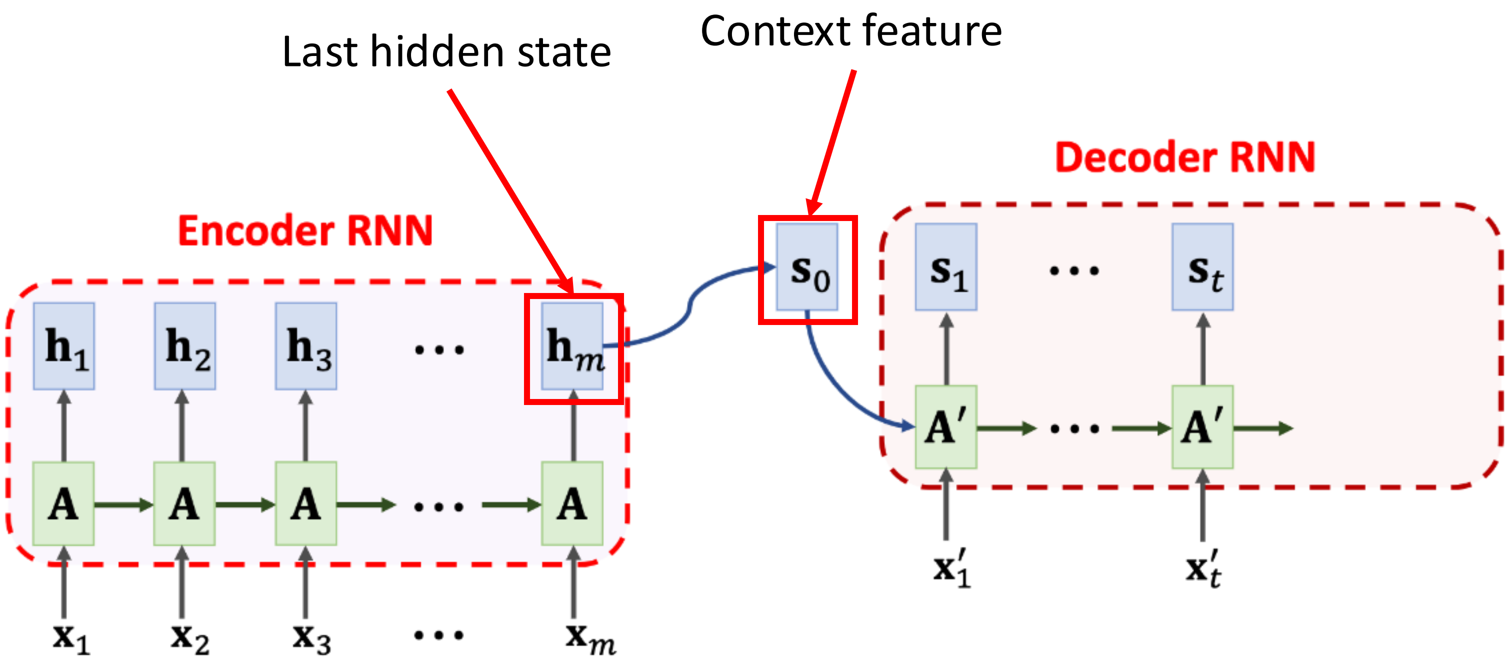


Image from https://github.com/wangshusen/DeepLearning/blob/master/Slides/9_RNN_8.pdf

Attention mechanism

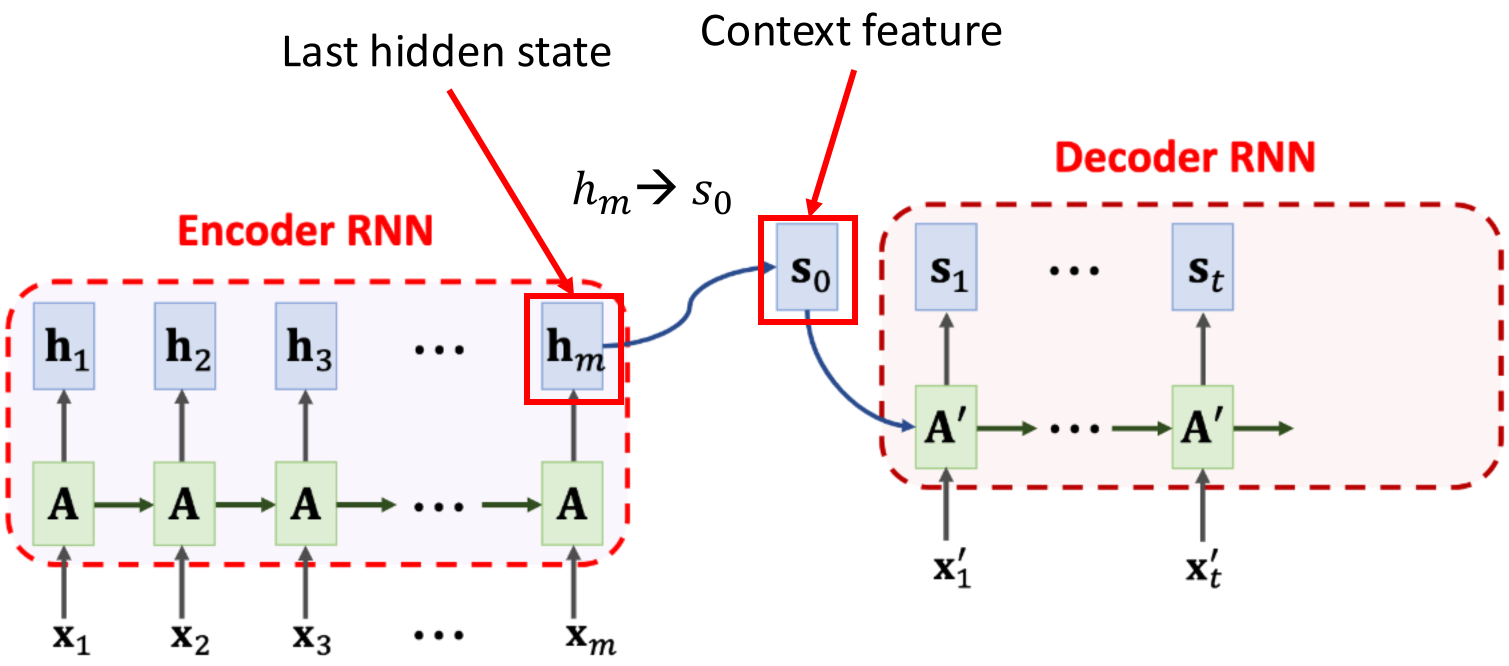
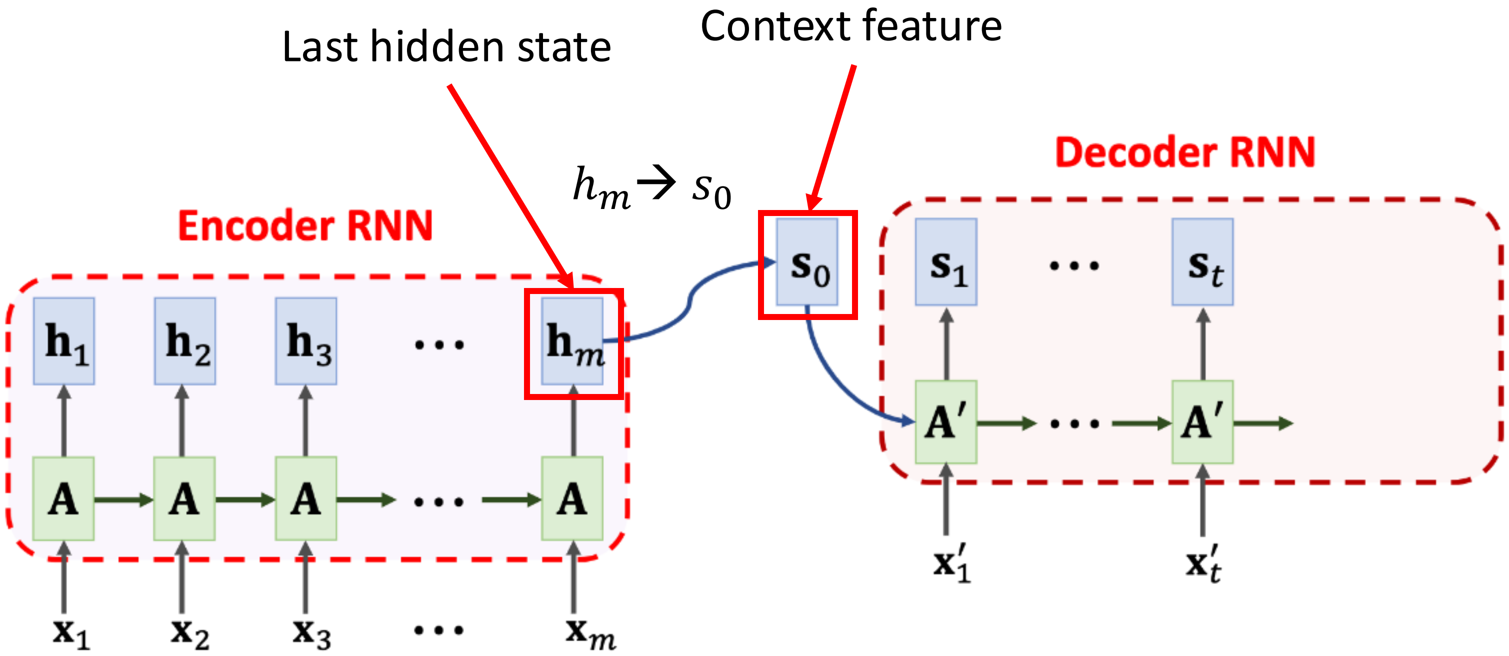


Image from https://github.com/wangshusen/DeepLearning/blob/master/Slides/9_RNN_8.pdf

Attention mechanism



No direct connection between hidden states of encoder and decoder

Image from https://github.com/wangshusen/DeepLearning/blob/master/Slides/9_RNN_8.pdf

Attention mechanism

weights

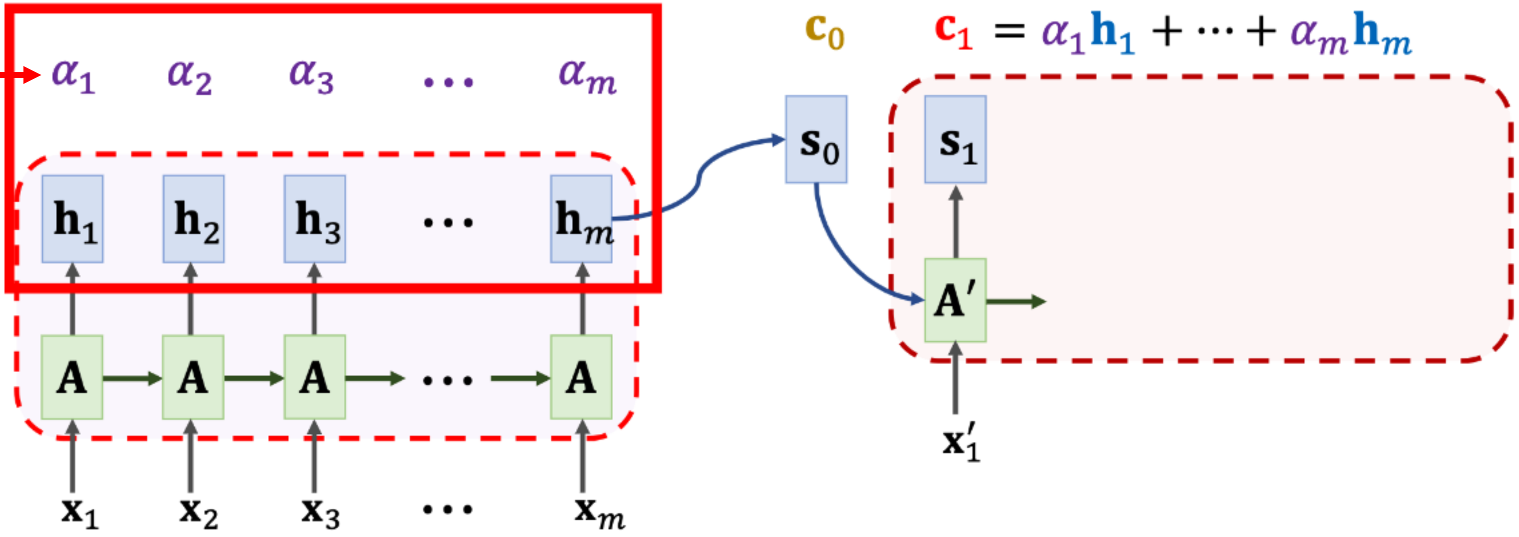


Image from https://github.com/wangshusen/DeepLearning/blob/master/Slides/9_RNN_8.pdf

Attention mechanism

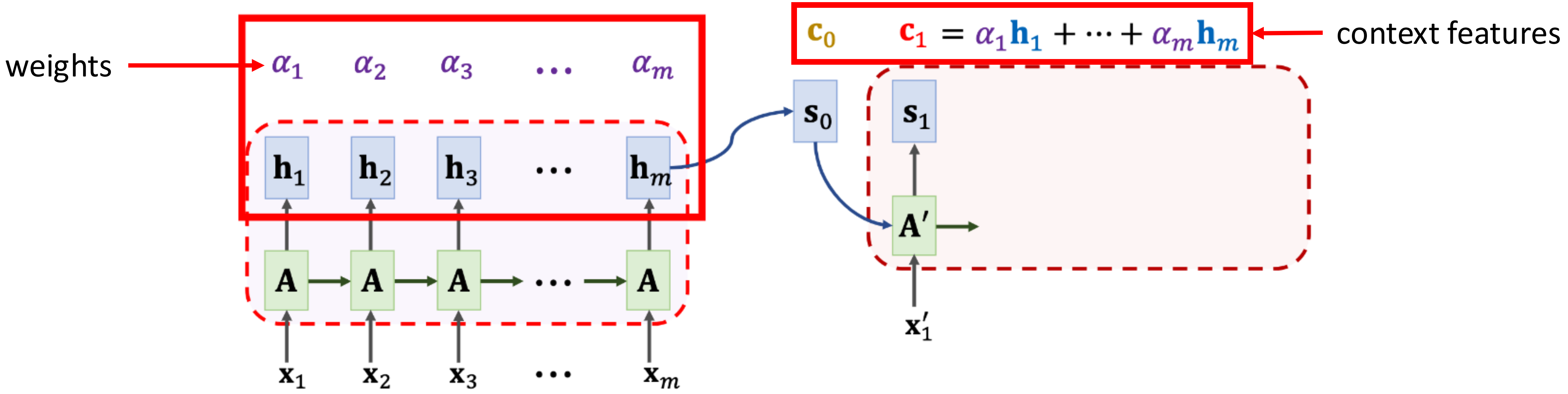


Image from https://github.com/wangshusen/DeepLearning/blob/master/Slides/9_RNN_8.pdf

Attention mechanism

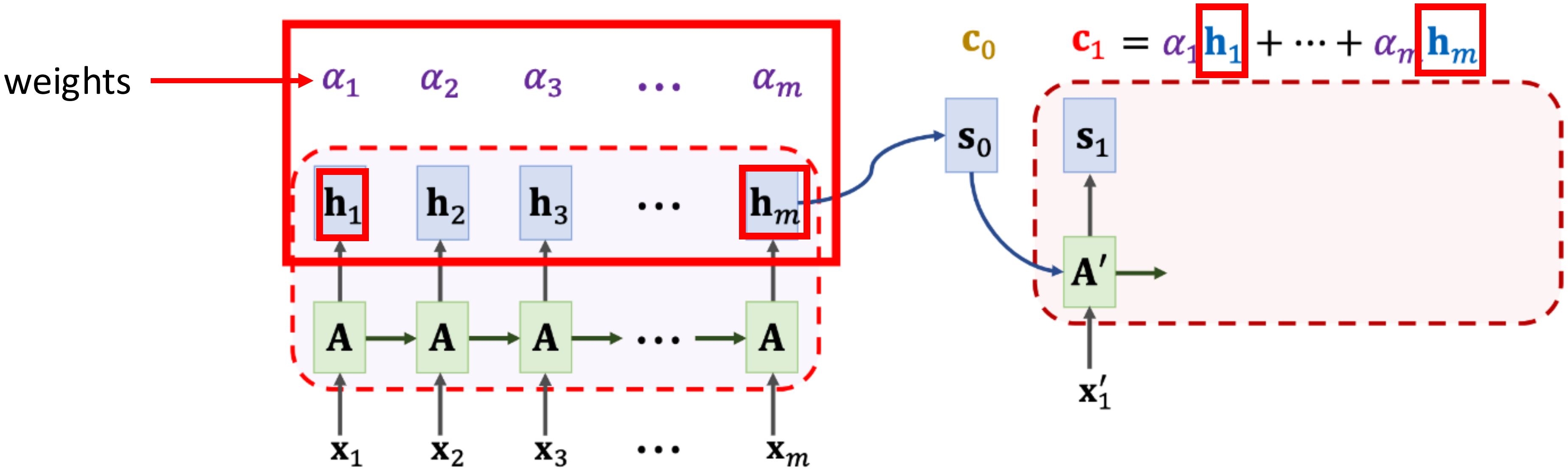


Image from https://github.com/wangshusen/DeepLearning/blob/master/Slides/9_RNN_8.pdf

Attention mechanism

weights

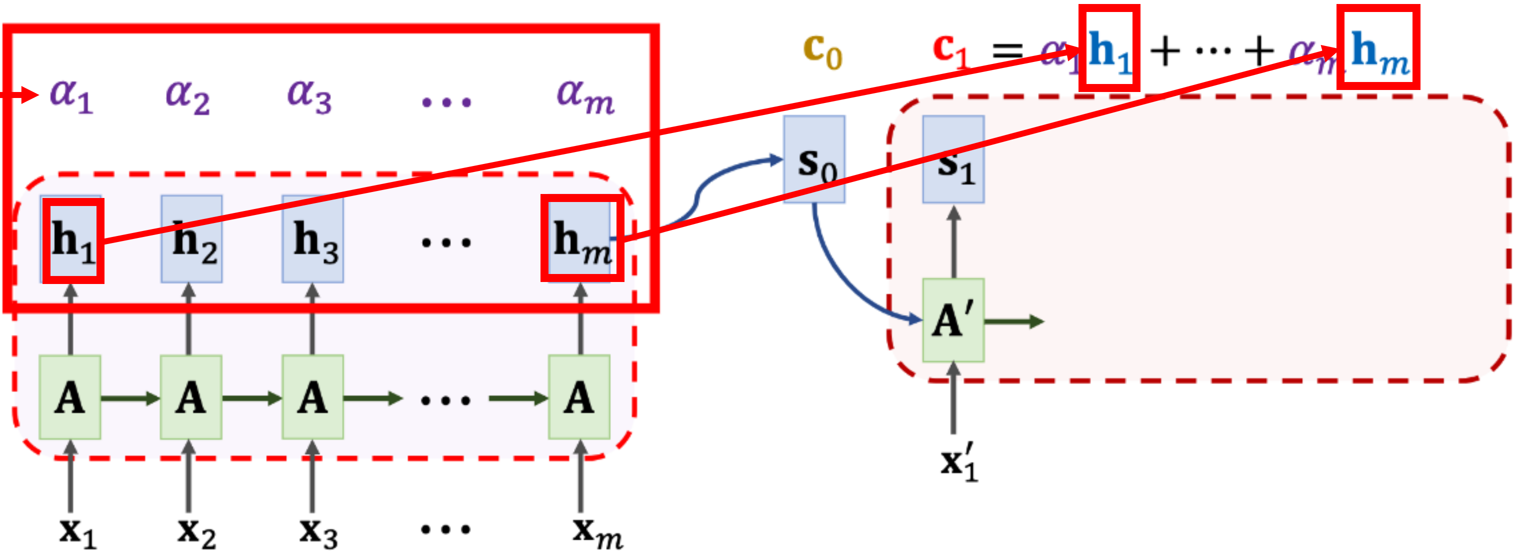


Image from https://github.com/wangshusen/DeepLearning/blob/master/Slides/9_RNN_8.pdf

Attention mechanism

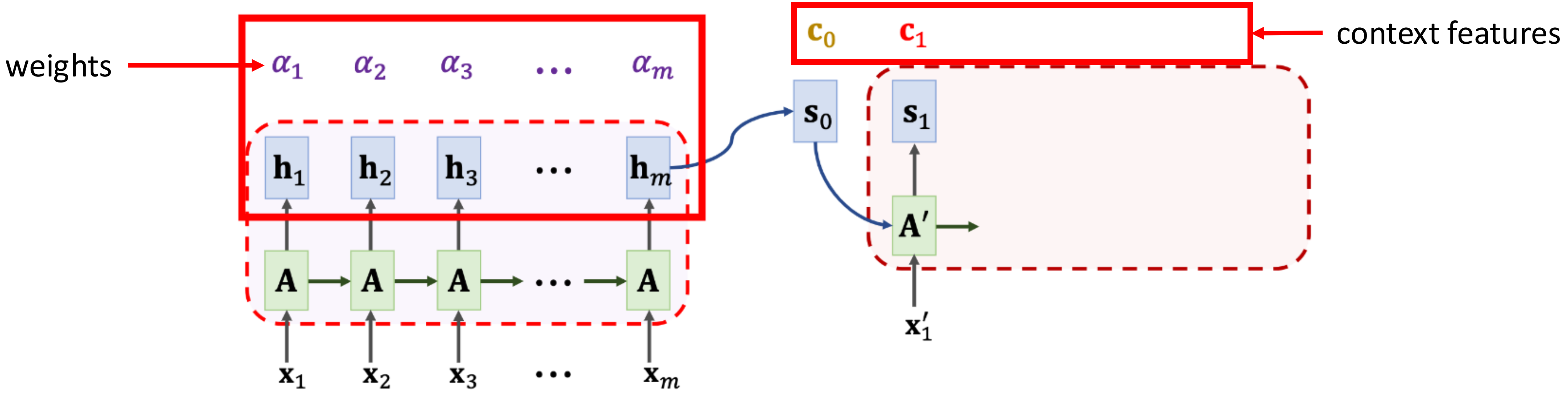


Image from https://github.com/wangshusen/DeepLearning/blob/master/Slides/9_RNN_8.pdf

Attention mechanism

How to use the two variables to build information flow?

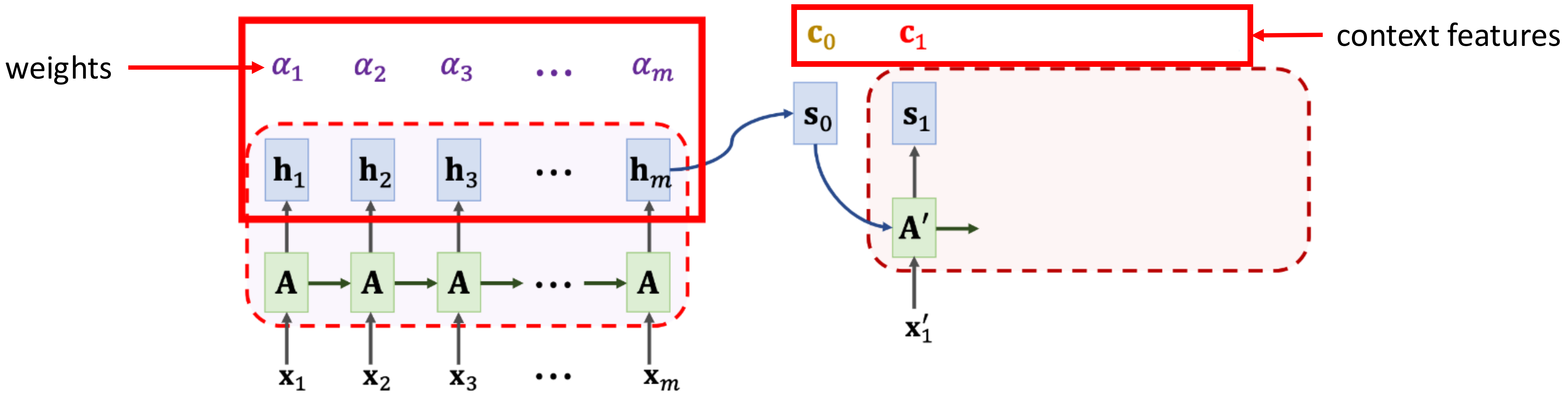


Image from https://github.com/wangshusen/DeepLearning/blob/master/Slides/9_RNN_8.pdf

Attention mechanism

Weight: $\alpha_i = \text{align}(\mathbf{h}_i, \mathbf{s}_0)$.

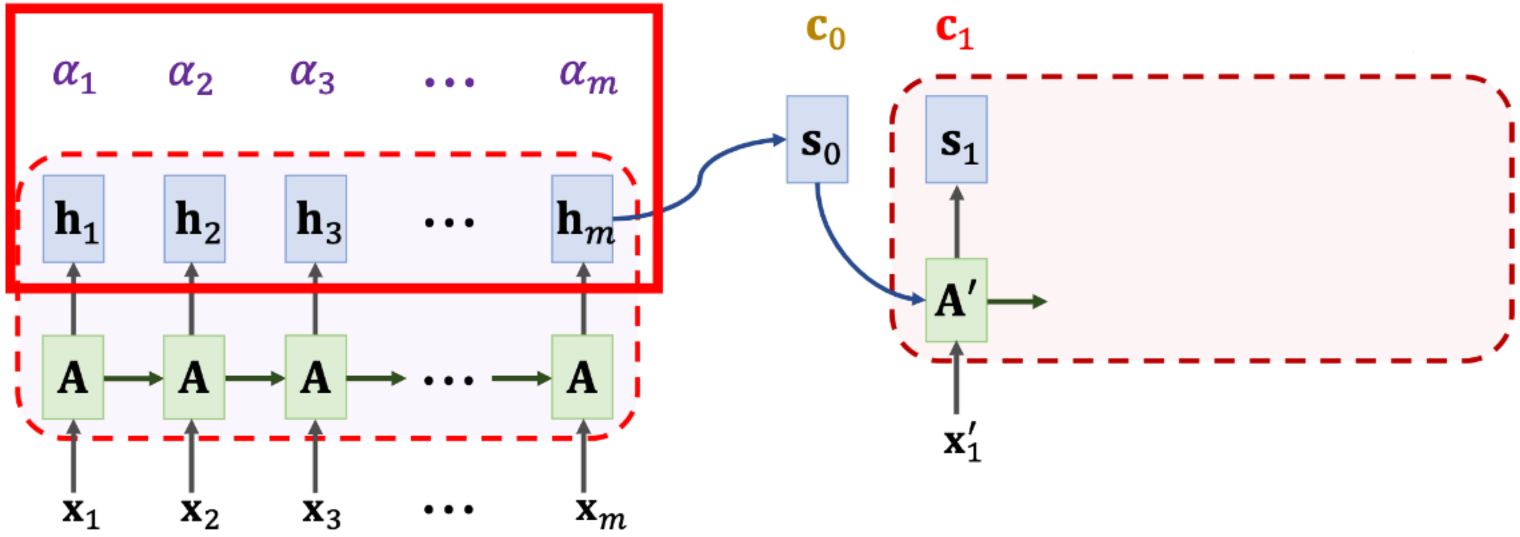


Image from https://github.com/wangshusen/DeepLearning/blob/master/Slides/9_RNN_8.pdf

Attention mechanism

Weight: $\alpha_i = \boxed{\text{align}}(\mathbf{h}_i, \mathbf{s}_0)$.

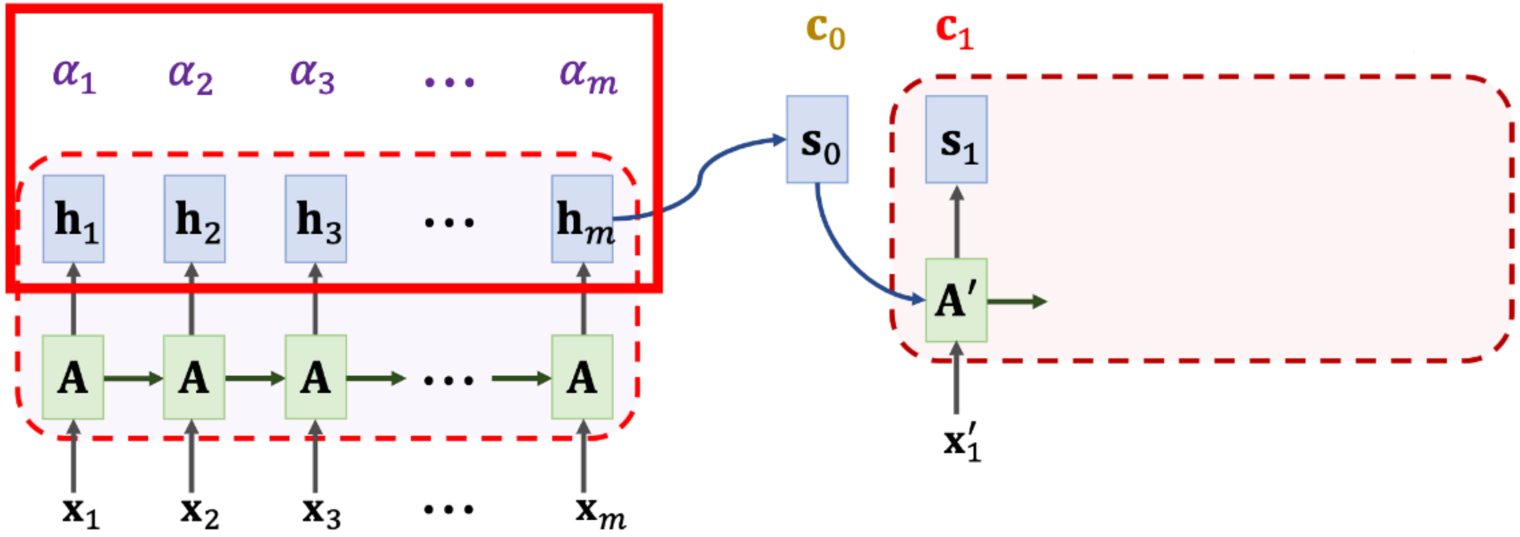


Image from https://github.com/wangshusen/DeepLearning/blob/master/Slides/9_RNN_8.pdf

Attention mechanism

Weight: $\alpha_i = \boxed{\text{align}}(\mathbf{h}_i, \mathbf{s}_0).$

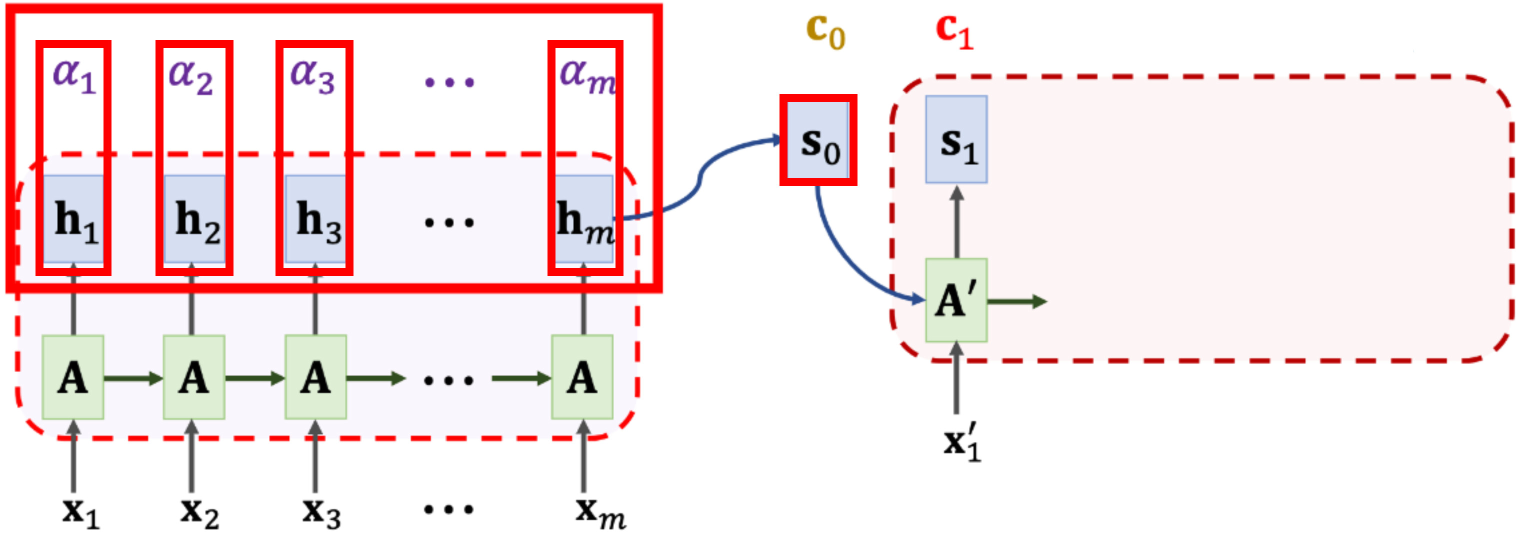
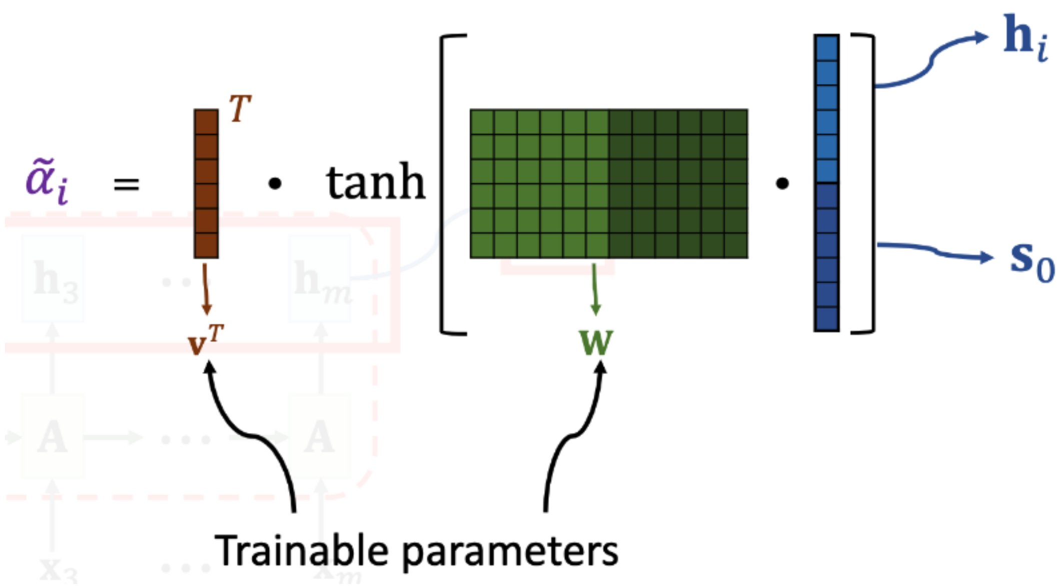


Image from https://github.com/wangshusen/DeepLearning/blob/master/Slides/9_RNN_8.pdf

Attention mechanism

Weight: $\alpha_i = \text{align}(\mathbf{h}_i, \mathbf{s}_0).$



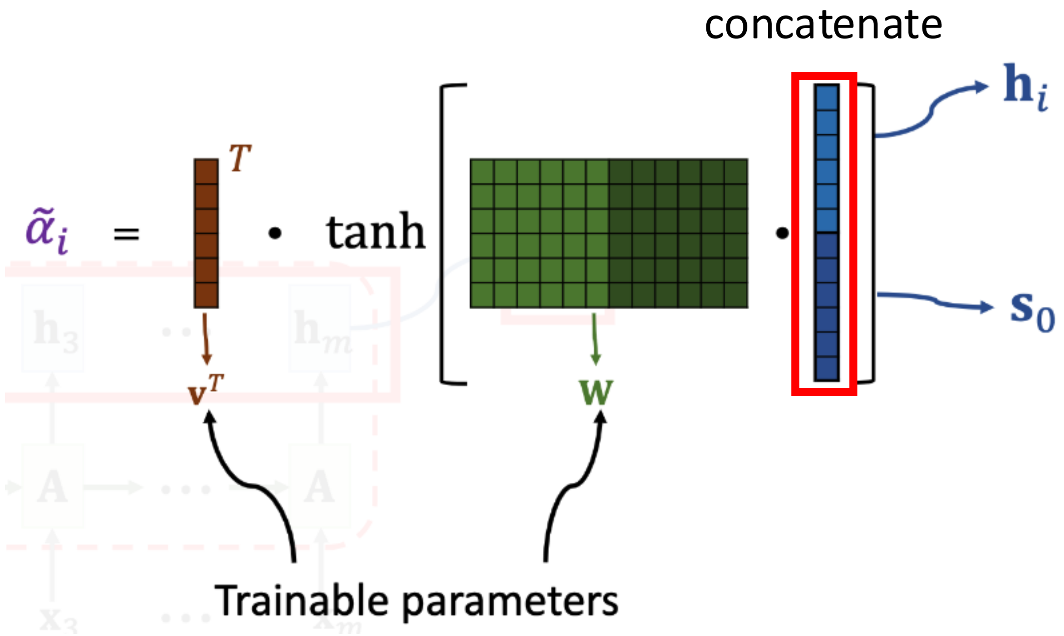
Then **normalize** $\tilde{\alpha}_1, \cdots, \tilde{\alpha}_m$ (so that they sum to 1):

$$[\alpha_1, \cdots, \alpha_m] = \text{Softmax}\left([\tilde{\alpha}_1, \cdots, \tilde{\alpha}_m]\right).$$

Image from https://github.com/wangshusen/DeepLearning/blob/master/Slides/9_RNN_8.pdf

Attention mechanism

Weight: $\alpha_i = \text{align}(\mathbf{h}_i, \mathbf{s}_0)$.



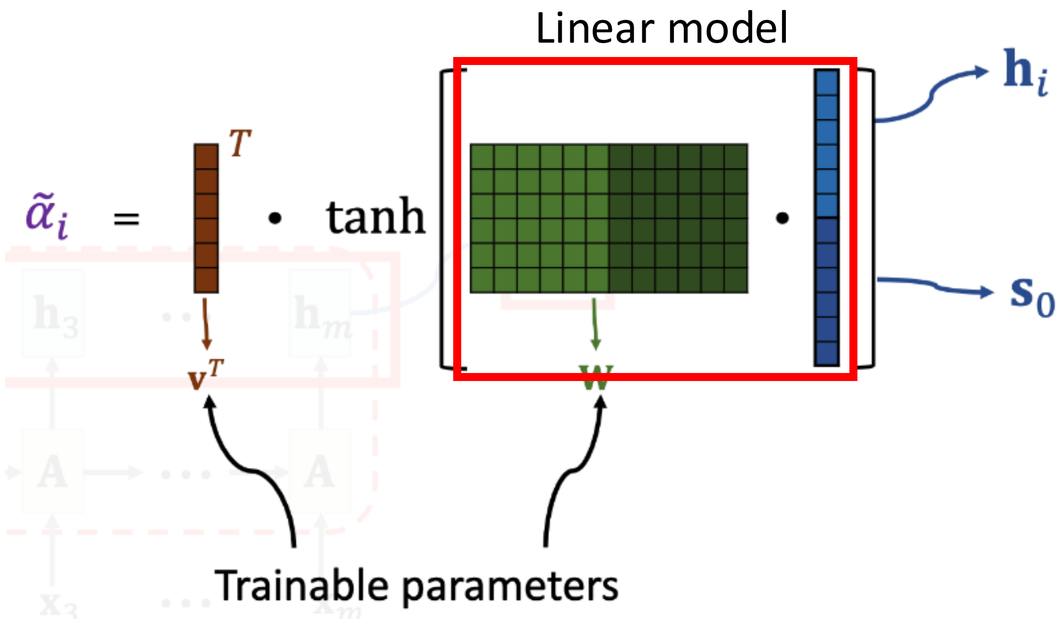
Then **normalize** $\tilde{\alpha}_1, \cdots, \tilde{\alpha}_m$ (so that they sum to 1):

$$[\alpha_1, \cdots, \alpha_m] = \text{Softmax}\left([\tilde{\alpha}_1, \cdots, \tilde{\alpha}_m]\right).$$

Image from https://github.com/wangshusen/DeepLearning/blob/master/Slides/9_RNN_8.pdf

Attention mechanism

Weight: $\alpha_i = \mathrm{align}(\mathbf{h}_i, \mathbf{s}_0).$



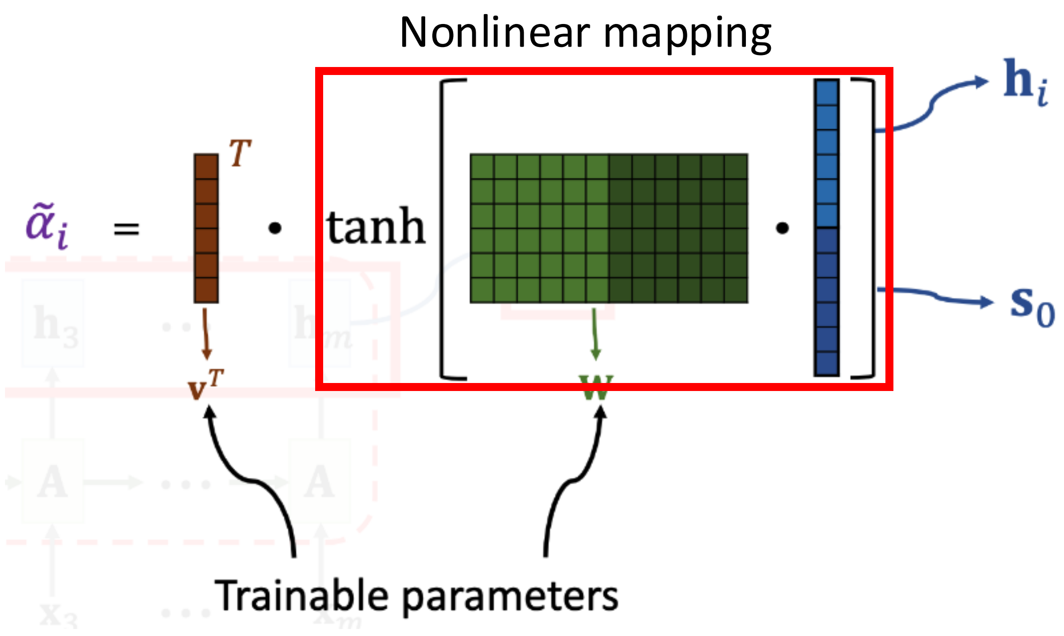
Then **normalize** $\tilde{\alpha}_1, \cdots, \tilde{\alpha}_m$ (so that they sum to 1):

$$[\alpha_1, \cdots, \alpha_m] = \mathrm{Softmax}\left([\tilde{\alpha}_1, \cdots, \tilde{\alpha}_m]\right).$$

Image from https://github.com/wangshusen/DeepLearning/blob/master/Slides/9_RNN_8.pdf

Attention mechanism

Weight: $\alpha_i = \text{align}(\mathbf{h}_i, \mathbf{s}_0)$.



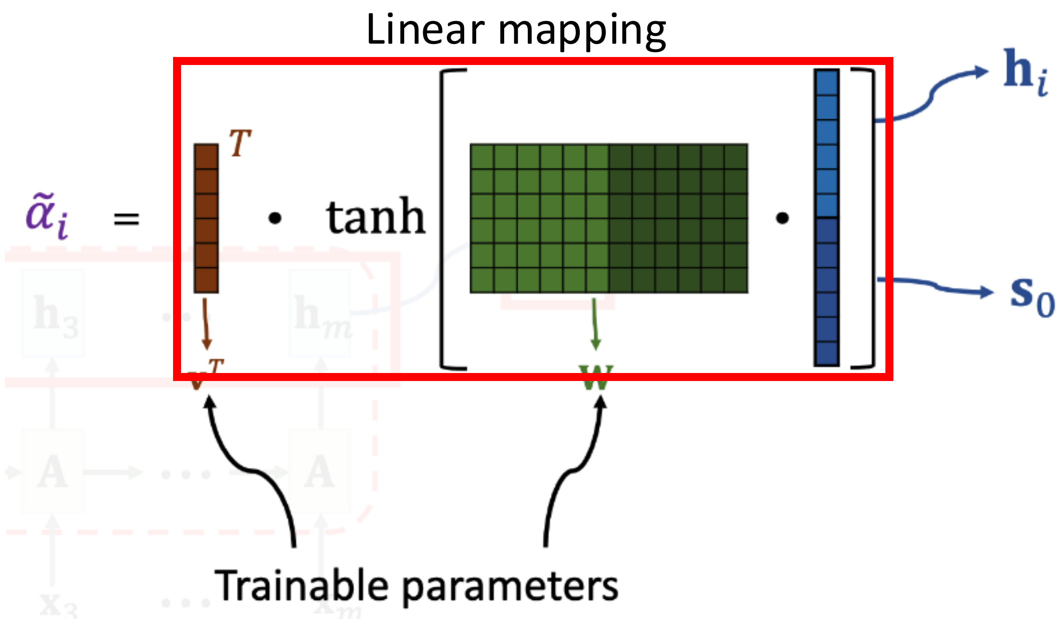
Then **normalize** $\tilde{\alpha}_1, \cdots, \tilde{\alpha}_m$ (so that they sum to 1):

$$[\alpha_1, \cdots, \alpha_m] = \text{Softmax}\left([\tilde{\alpha}_1, \cdots, \tilde{\alpha}_m]\right).$$

Image from https://github.com/wangshusen/DeepLearning/blob/master/Slides/9_RNN_8.pdf

Attention mechanism

Weight: $\alpha_i = \text{align}(\mathbf{h}_i, \mathbf{s}_0)$.



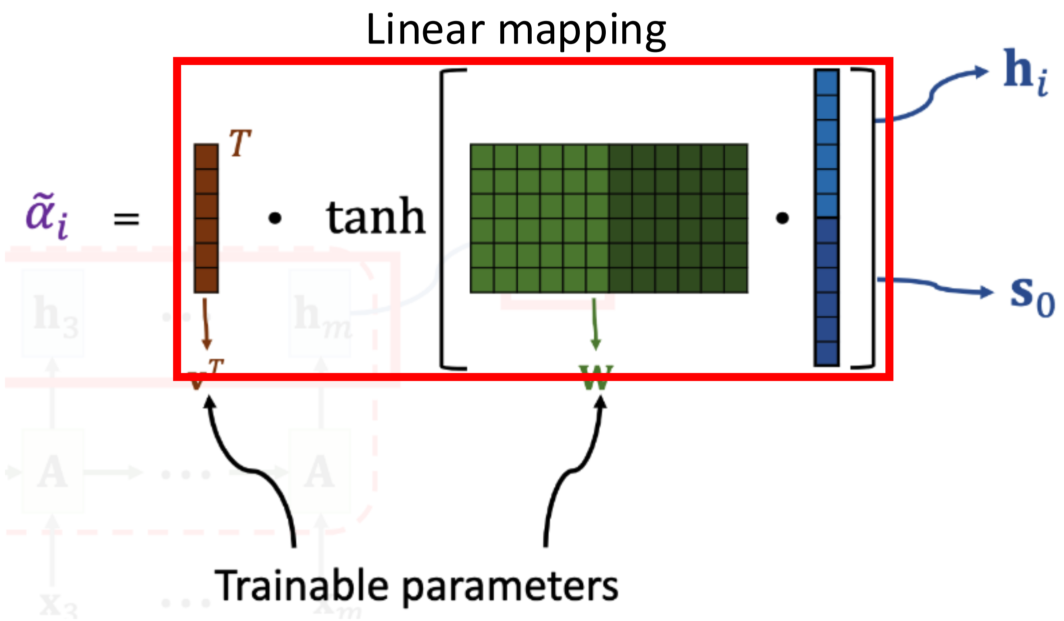
Then **normalize** $\tilde{\alpha}_1, \cdots, \tilde{\alpha}_m$ (so that they sum to 1):

$$[\alpha_1, \cdots, \alpha_m] = \text{Softmax}\left([\tilde{\alpha}_1, \cdots, \tilde{\alpha}_m]\right).$$

Image from https://github.com/wangshusen/DeepLearning/blob/master/Slides/9_RNN_8.pdf

Attention mechanism

Weight: $\alpha_i = \text{align}(\mathbf{h}_i, \mathbf{s}_0).$



Then **normalize** $\tilde{\alpha}_1, \cdots, \tilde{\alpha}_m$ (so that they sum to 1):

$$\boxed{[\alpha_1, \cdots, \alpha_m] = \text{Softmax}\left([\tilde{\alpha}_1, \cdots, \tilde{\alpha}_m]\right).}$$

Image from https://github.com/wangshusen/DeepLearning/blob/master/Slides/9_RNN_8.pdf

Attention mechanism

Weight: $\alpha_i = \text{align}(\mathbf{h}_i, \mathbf{s}_0)$.

Context vector: $\boxed{\mathbf{c}_0} = \alpha_1 \mathbf{h}_1 + \cdots + \alpha_m \mathbf{h}_m$.

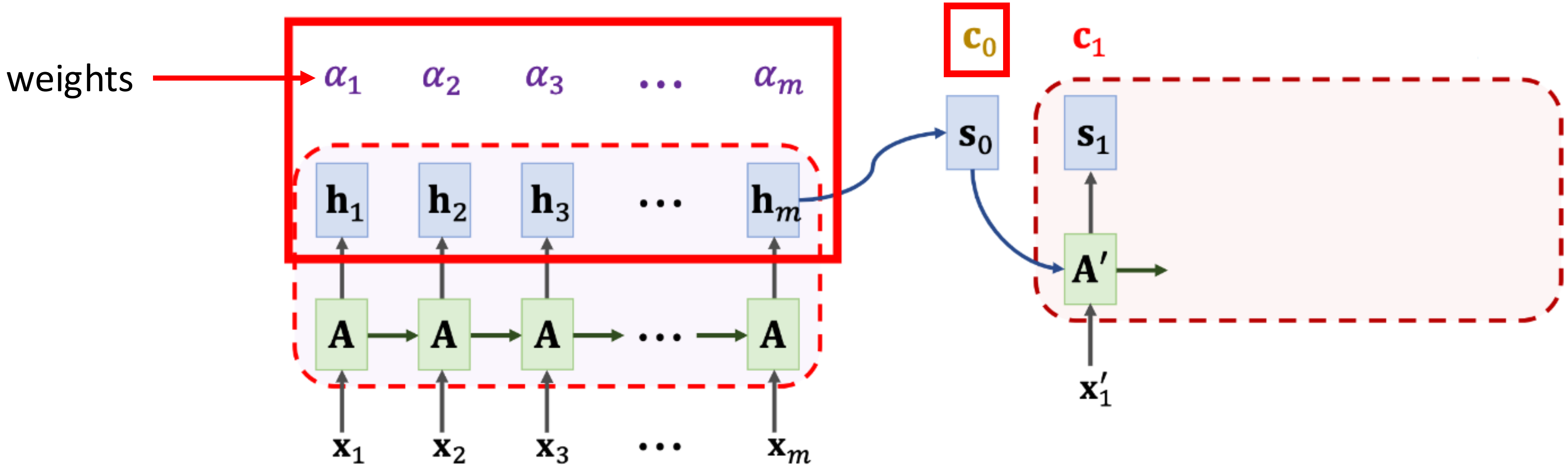


Image from https://github.com/wangshusen/DeepLearning/blob/master/Slides/9_RNN_8.pdf

Attention mechanism

Weight: $\alpha_i = \text{align}(\mathbf{h}_i, \mathbf{s}_0)$.

Context vector: $\boxed{\mathbf{c}_0} = \alpha_1 \mathbf{h}_1 + \cdots + \alpha_m \mathbf{h}_m$.

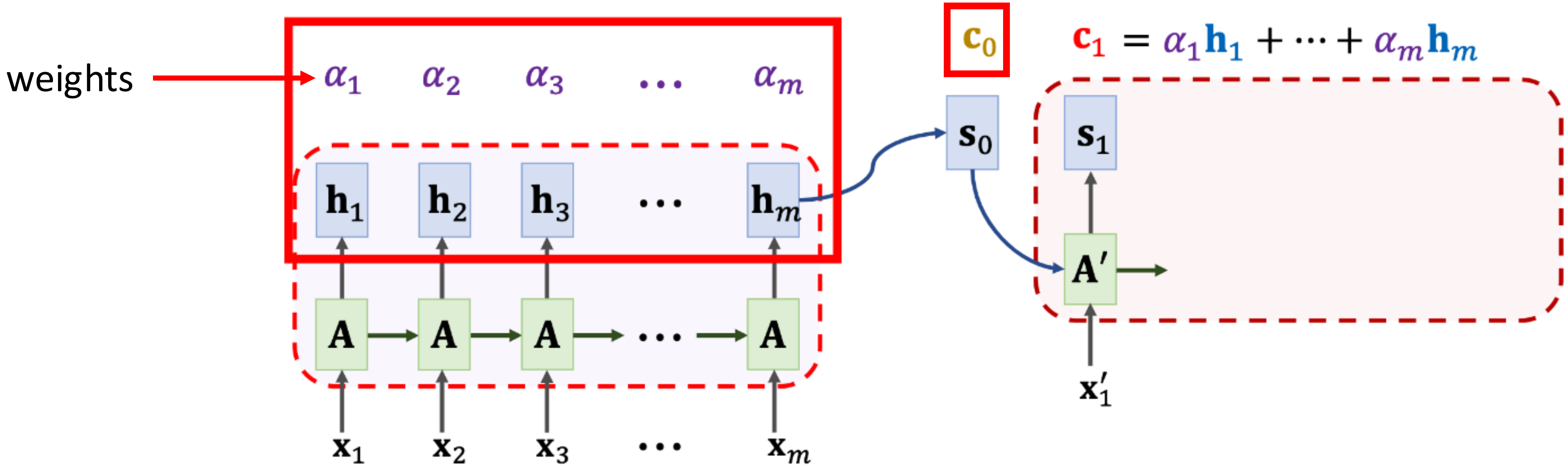


Image from https://github.com/wangshusen/DeepLearning/blob/master/Slides/9_RNN_8.pdf

Attention mechanism

Weight: $\alpha_i = \mathrm{align}(\mathbf{h}_i, \mathbf{s}_0)$.

Context vector: $\boxed{\mathbf{c}_0} = \alpha_1 \mathbf{h}_1 + \cdots + \alpha_m \mathbf{h}_m$.

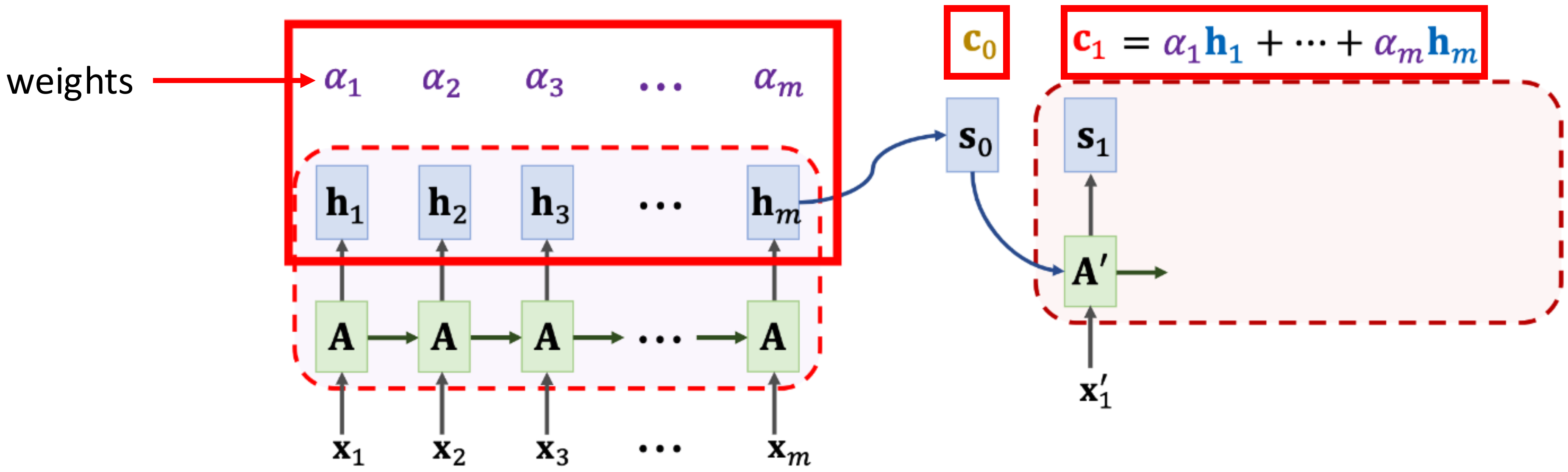


Image from https://github.com/wangshusen/DeepLearning/blob/master/Slides/9_RNN_8.pdf

Attention mechanism

Weight: $\alpha_i = \text{align}(\mathbf{h}_i, \mathbf{s}_0).$

Context vector: $\boxed{\mathbf{c}_0} = \alpha_1 \mathbf{h}_1 + \cdots + \alpha_m \mathbf{h}_m.$

Q: should we use different $\alpha' s$ for different c_t?

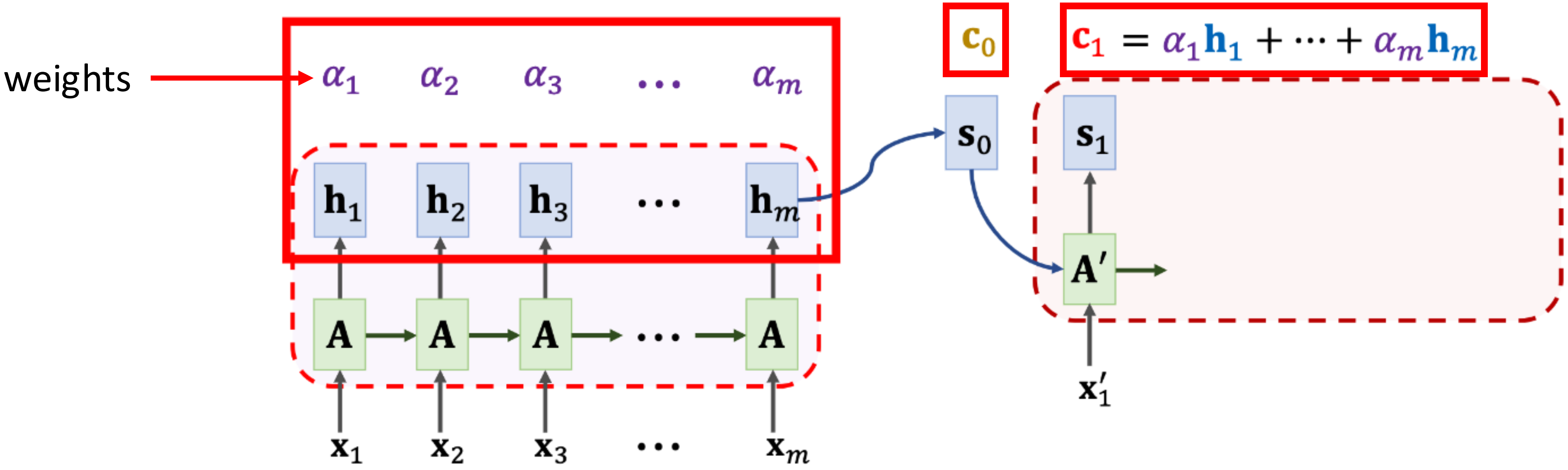


Image from https://github.com/wangshusen/DeepLearning/blob/master/Slides/9_RNN_8.pdf

Attention mechanism

Weight: $\alpha_i = \mathrm{align}(\mathbf{h}_i, \mathbf{s}_0)$.

Context vector: $\boxed{\mathbf{c}_0} = \alpha_1 \mathbf{h}_1 + \cdots + \alpha_m \mathbf{h}_m$.

Q: should we use different $\alpha' s$ for different c_t?

Different $\alpha' s$. Why?

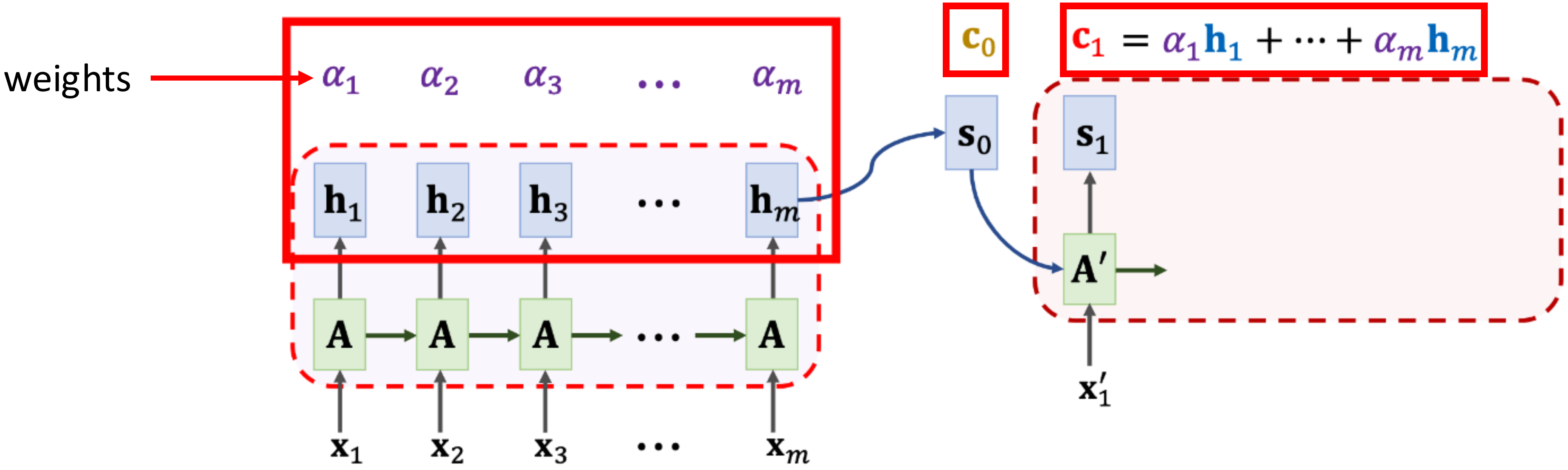


Image from https://github.com/wangshusen/DeepLearning/blob/master/Slides/9_RNN_8.pdf

Attention mechanism

$$c_i = \sum_{j=1}^{T_x} \boxed{\alpha_{ij}} h_j.$$

In the original paper

Weight: $\alpha_i = \mathrm{align}(\mathbf{h}_i, \mathbf{s}_0).$

Context vector: $\boxed{\mathbf{c}_0} = \alpha_1 \mathbf{h}_1 + \cdots + \alpha_m \mathbf{h}_m.$

Q: should we use different $\alpha' s$ for different c_t?

Different $\alpha' s$. Why?

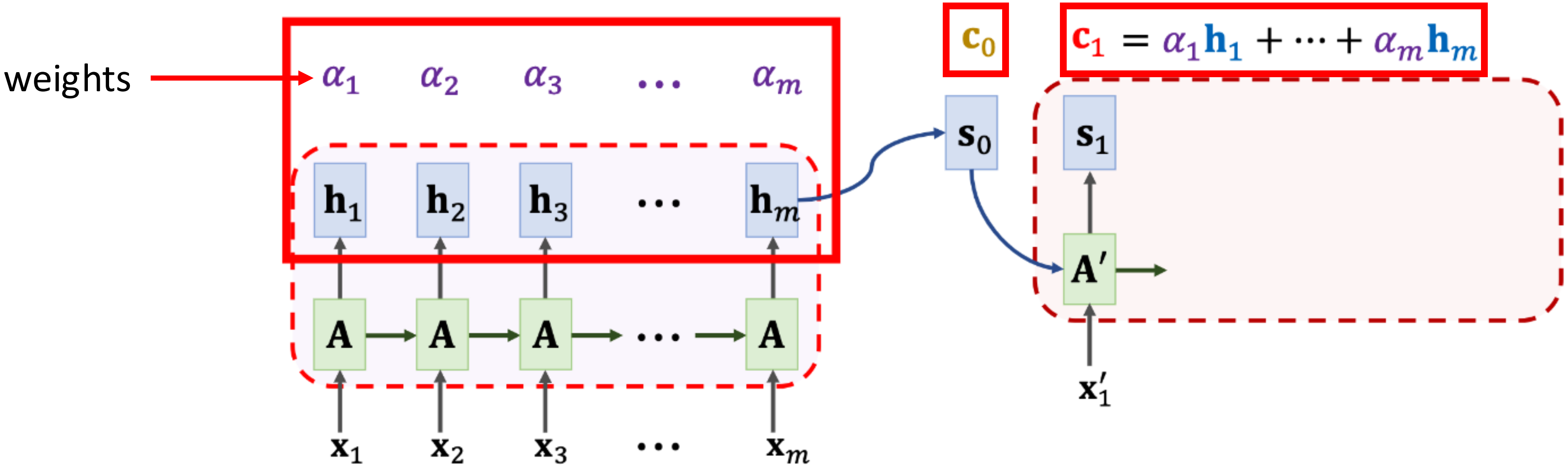


Image from https://github.com/wangshusen/DeepLearning/blob/master/Slides/9_RNN_8.pdf

Attention mechanism

$$c_i = \sum_{j=1}^{T_x} \boxed{\alpha_{ij}} h_j.$$

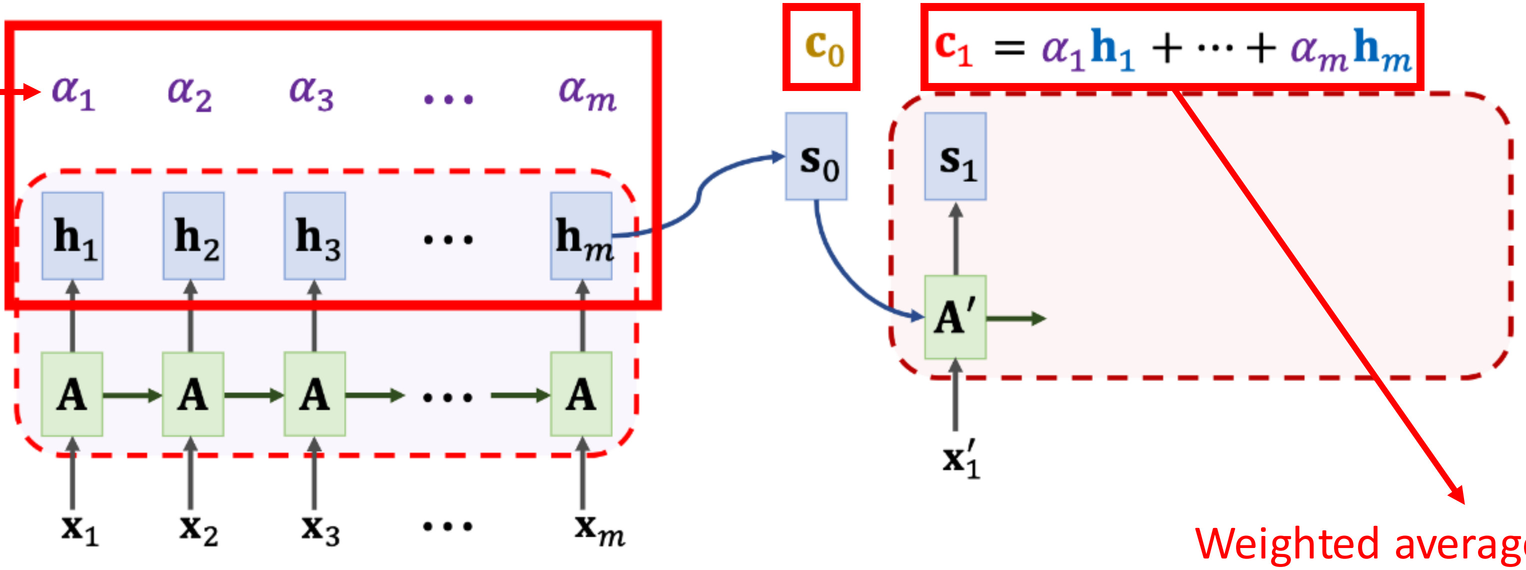
In the original paper

Weight: $\alpha_i = \text{align}(\mathbf{h}_i, \mathbf{s}_0).$

Context vector: $\boxed{\mathbf{c}_0} = \alpha_1 \mathbf{h}_1 + \cdots + \alpha_m \mathbf{h}_m.$

Q: should we use different $\alpha's$ for different c_t?
Different $\alpha's$. Why?

weights



Weighted average of
previous hidden features

Image from https://github.com/wangshusen/DeepLearning/blob/master/Slides/9_RNN_8.pdf

Attention mechanism

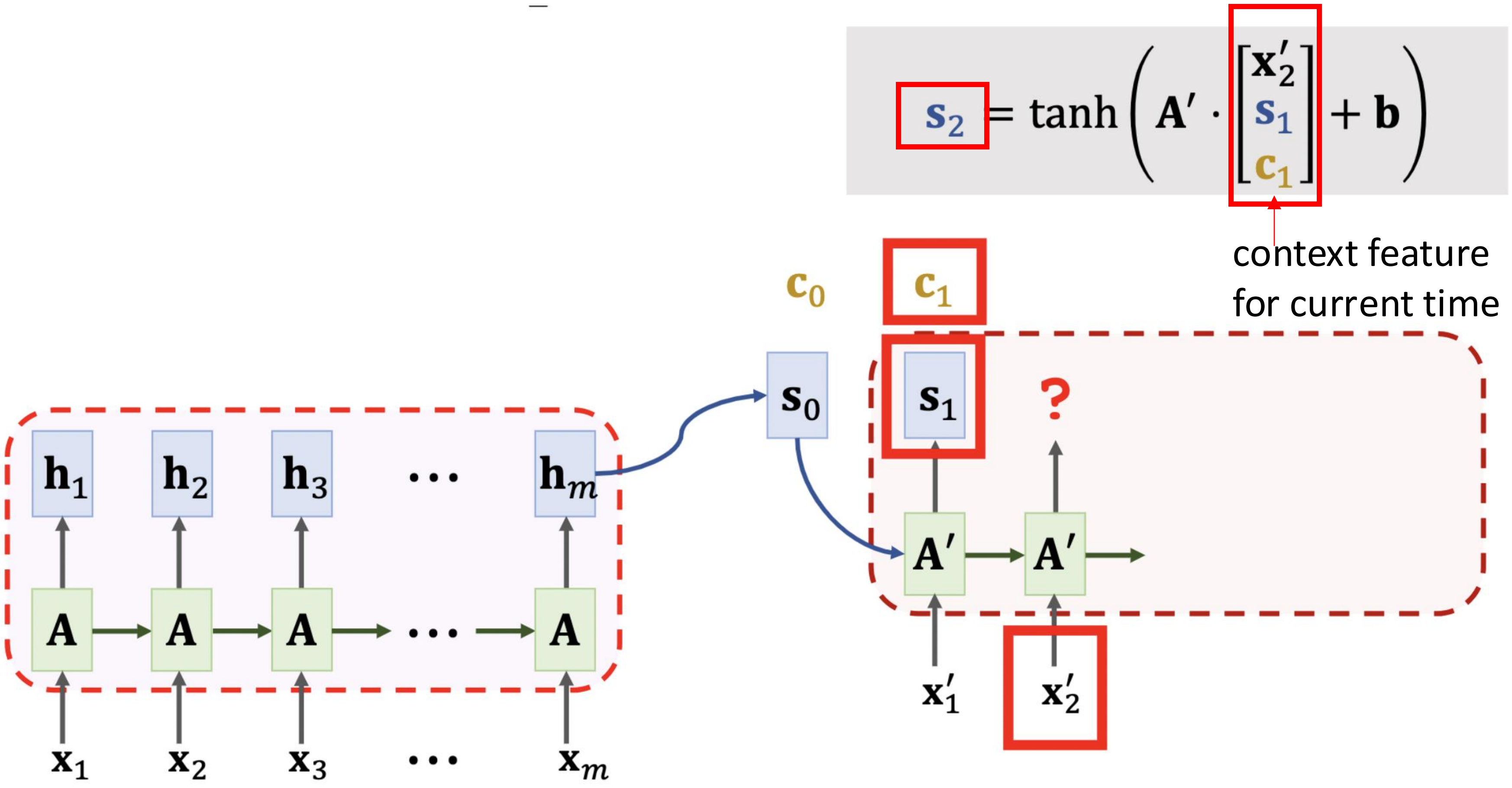


Image from https://github.com/wangshusen/DeepLearning/blob/master/Slides/9_RNN_8.pdf

Attention mechanism

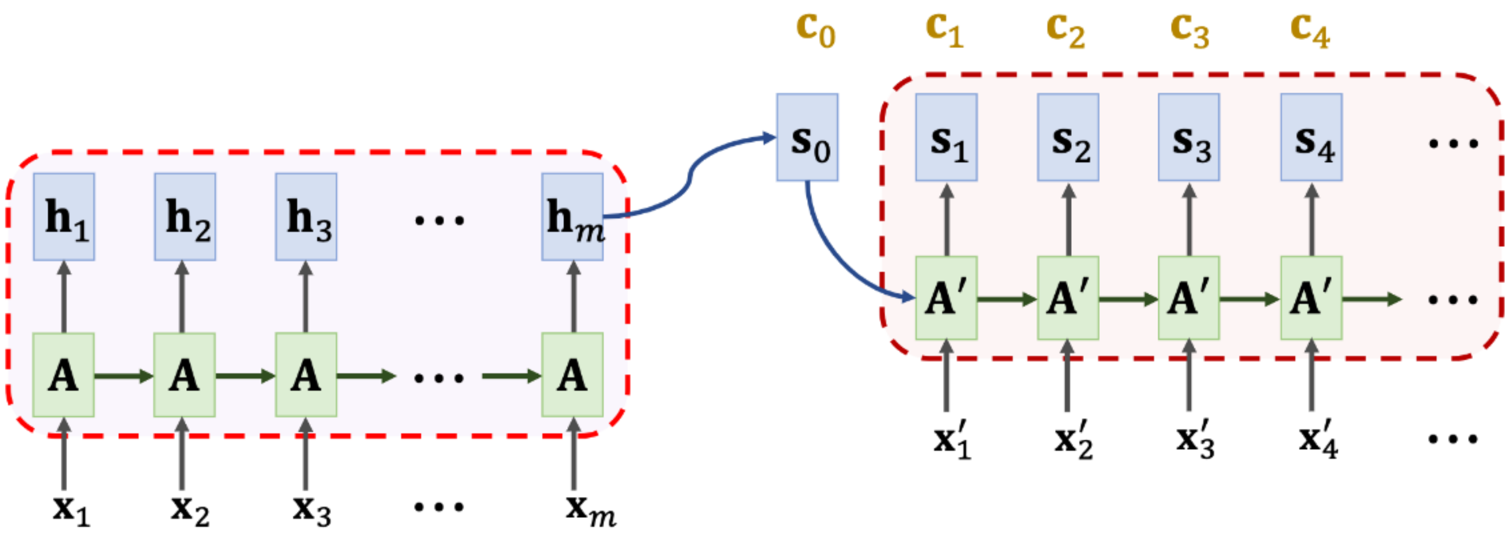


Image from https://github.com/wangshusen/DeepLearning/blob/master/Slides/9_RNN_8.pdf

Attention mechanism

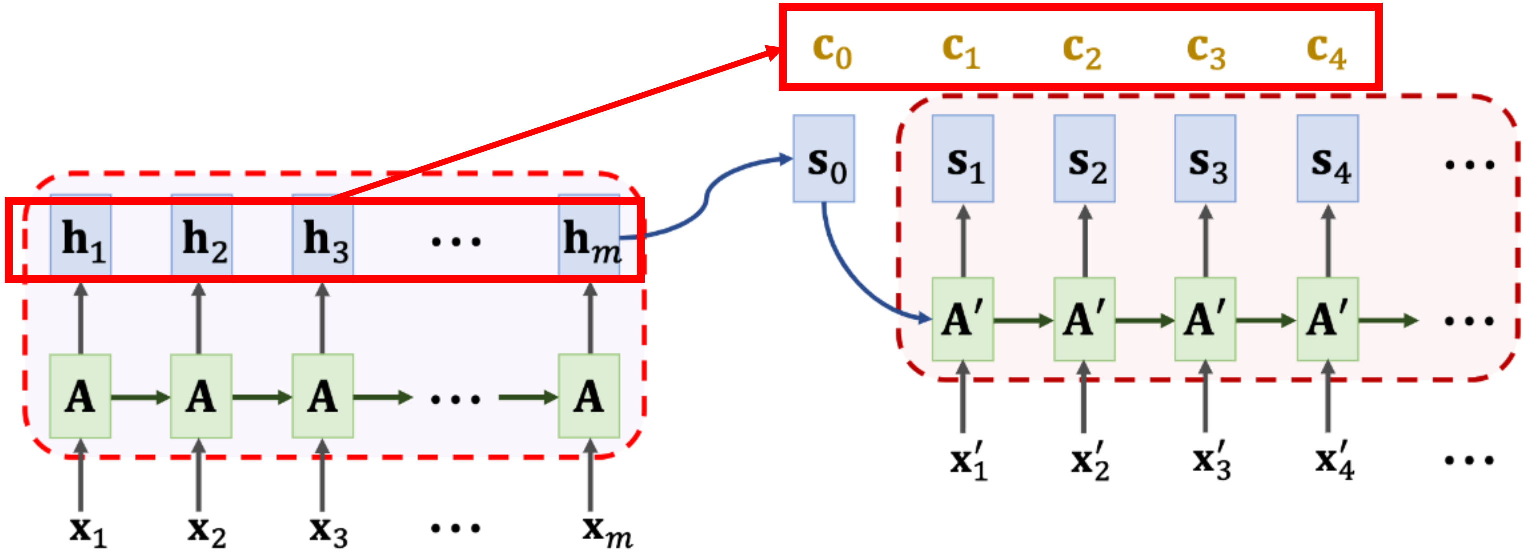
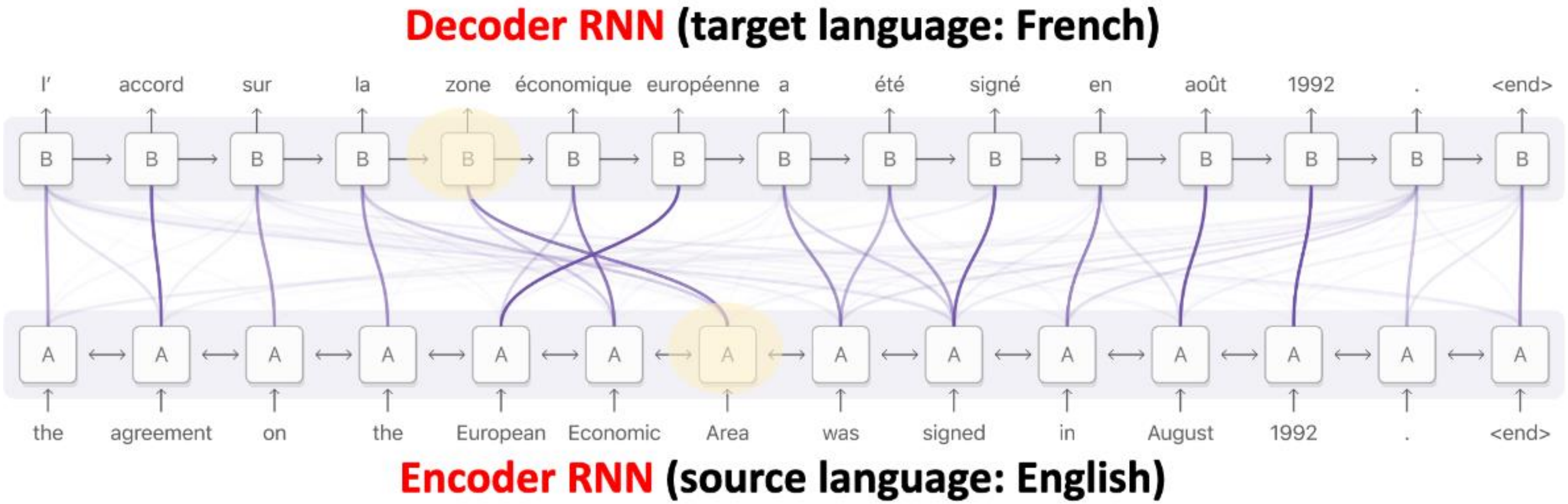
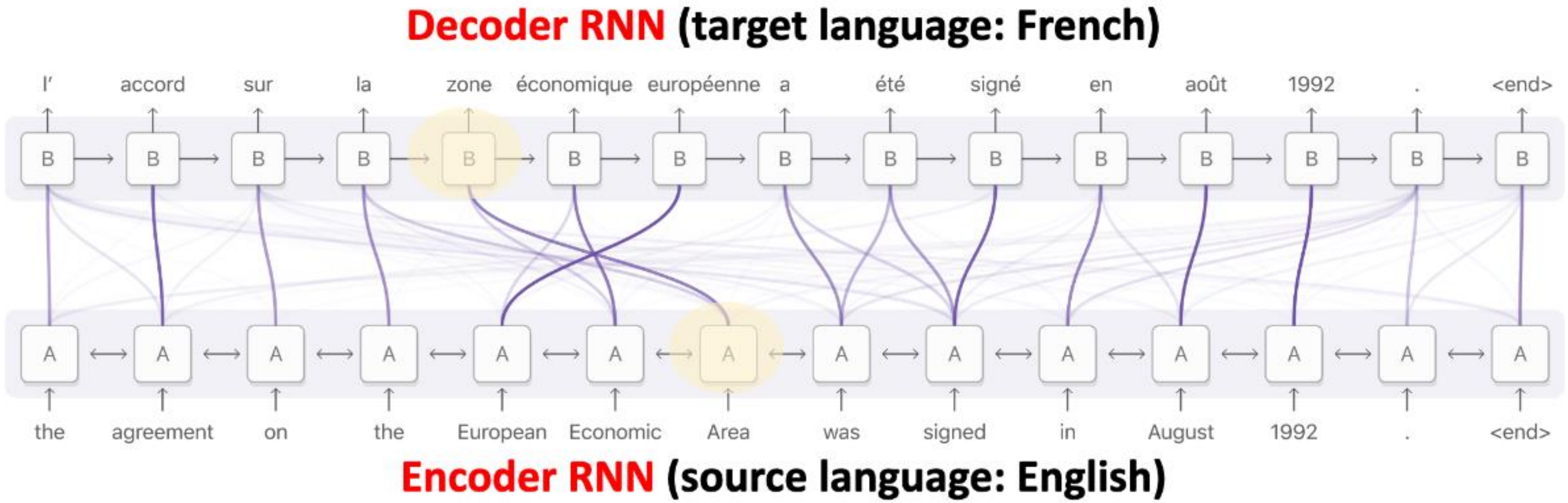


Image from https://github.com/wangshusen/DeepLearning/blob/master/Slides/9_RNN_8.pdf

Input-output correlation



Input-output correlation



Q: can we build attention mechanism in a single RNN (e.g., the encoder)?

Self-attention (intra-attention)

SimpleRNN:

$$\mathbf{h}_1 = \tanh\left(\mathbf{A} \cdot \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{h}_0 \end{bmatrix} + \mathbf{b}\right)$$

Image from https://github.com/wangshusen/DeepLearning/blob/master/Slides/9_RNN_9.pdf

Self-attention (intra-attention)

SimpleRNN:

$$\mathbf{h}_1 = \tanh\left(\mathbf{A} \cdot \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{h}_0 \end{bmatrix} + \mathbf{b}\right)$$

SimpleRNN + Self-Attention:

$$\mathbf{h}_1 = \tanh\left(\mathbf{A} \cdot \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{c}_0 \end{bmatrix} + \mathbf{b}\right)$$

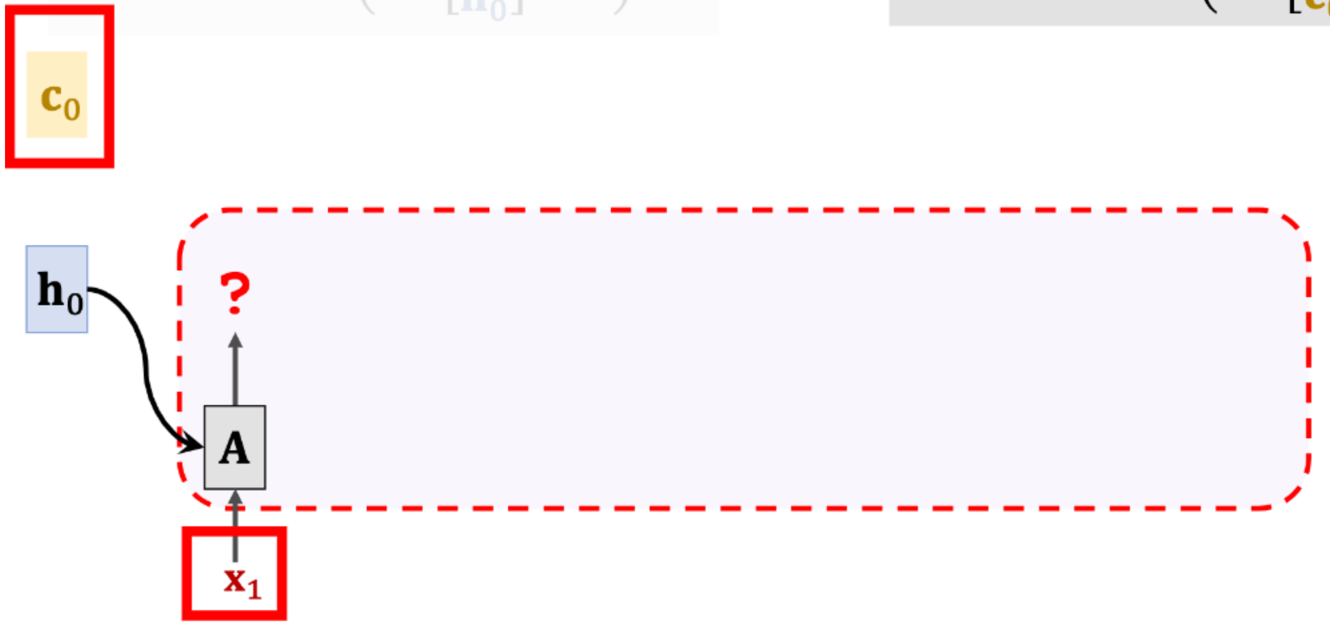


Image from https://github.com/wangshusen/DeepLearning/blob/master/Slides/9_RNN_9.pdf

Self-attention (intra-attention)

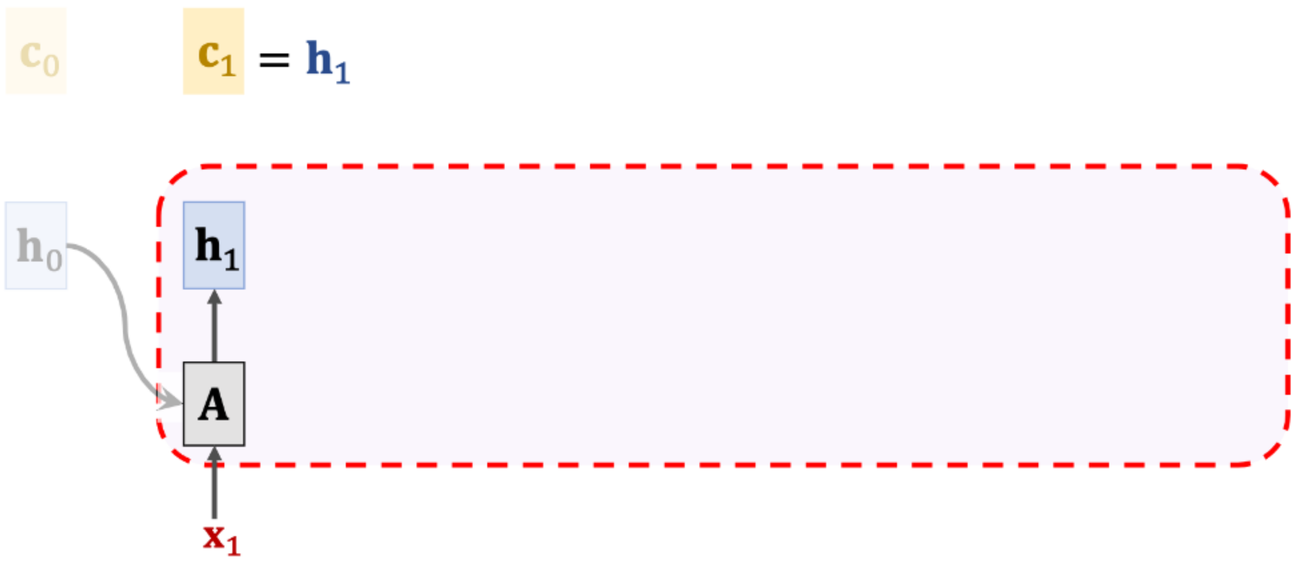


Image from https://github.com/wangshusen/DeepLearning/blob/master/Slides/9_RNN_9.pdf

Self-attention (intra-attention)

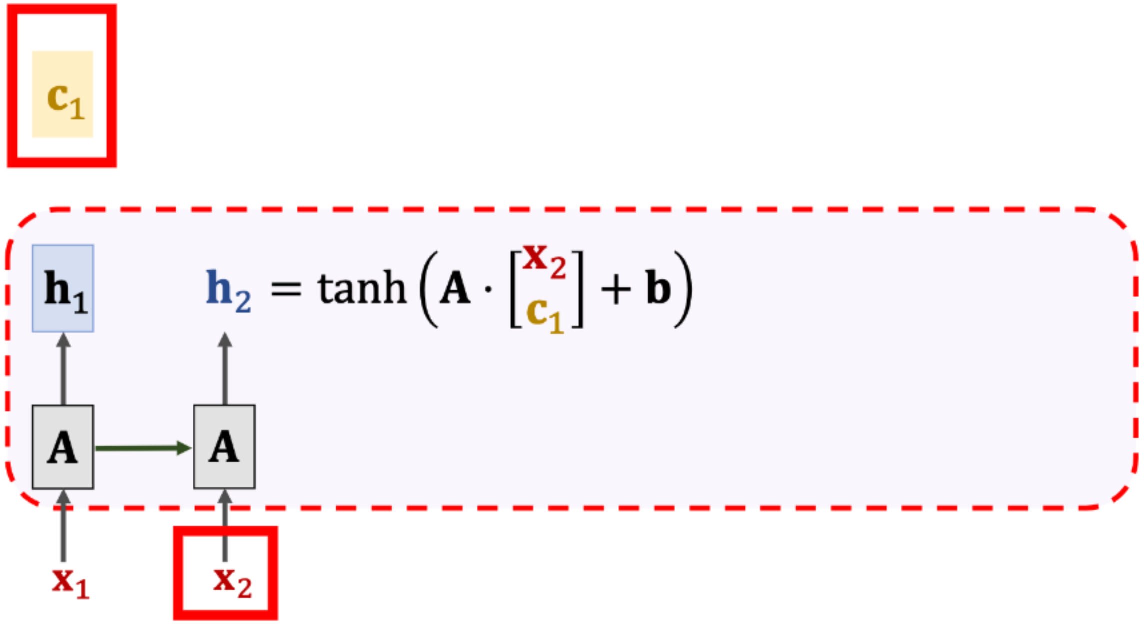


Image from https://github.com/wangshusen/DeepLearning/blob/master/Slides/9_RNN_9.pdf

Self-attention (intra-attention)

Weights: $\alpha_i = \text{align}(\mathbf{h}_i, \ \mathbf{h}_2).$

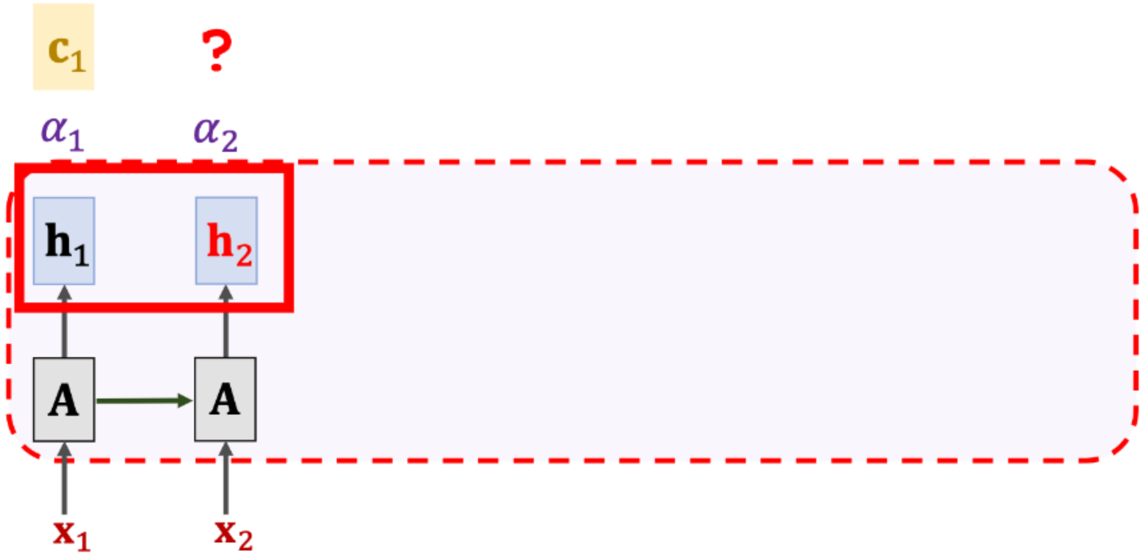


Image from https://github.com/wangshusen/DeepLearning/blob/master/Slides/9_RNN_9.pdf

Self-attention (intra-attention)

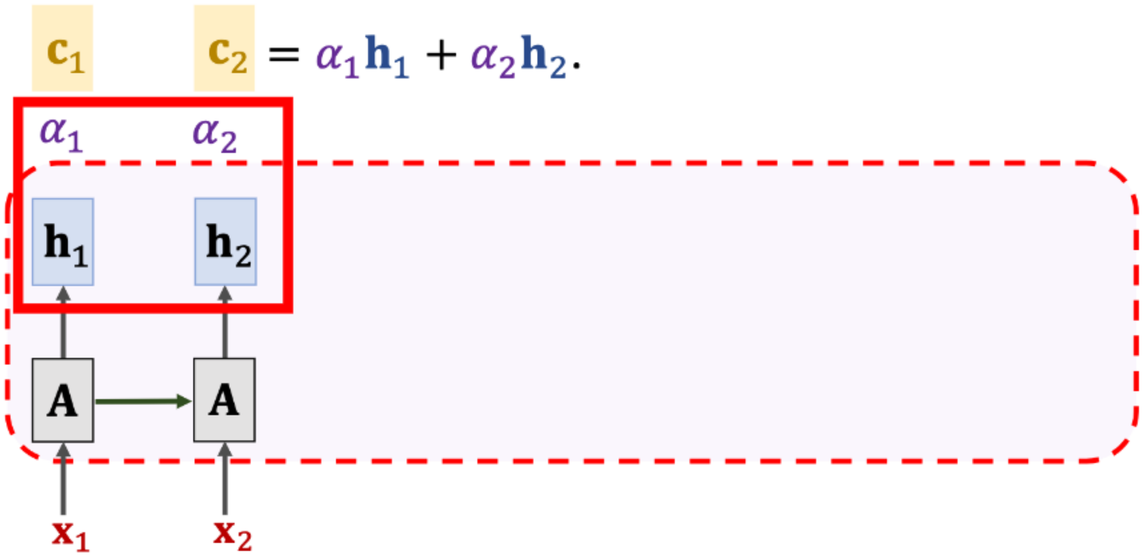


Image from https://github.com/wangshusen/DeepLearning/blob/master/Slides/9_RNN_9.pdf

Self-attention (intra-attention)

Weighted average

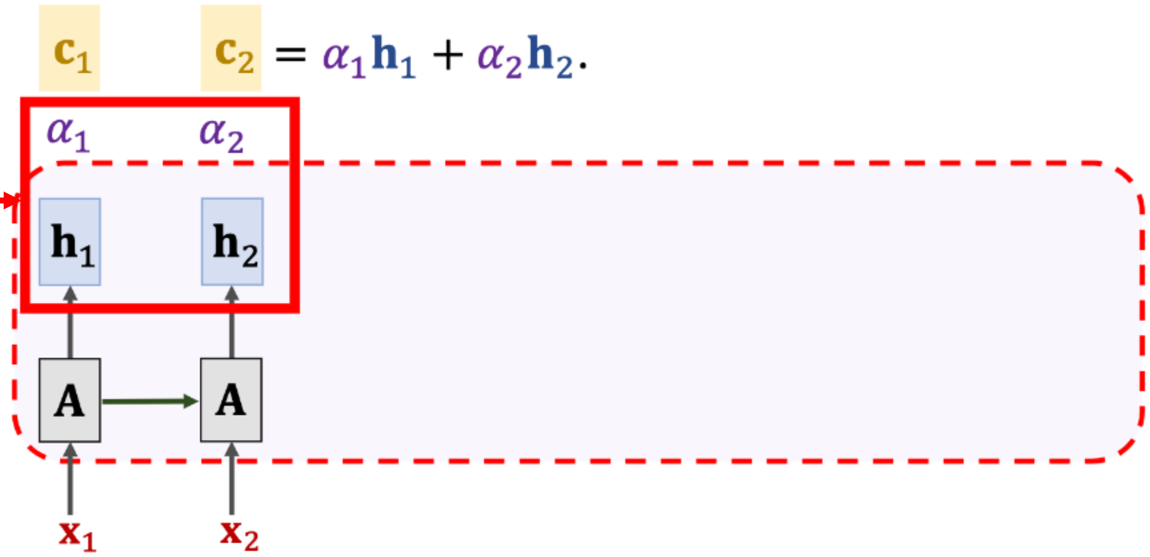


Image from https://github.com/wangshusen/DeepLearning/blob/master/Slides/9_RNN_9.pdf

Self-attention (intra-attention)

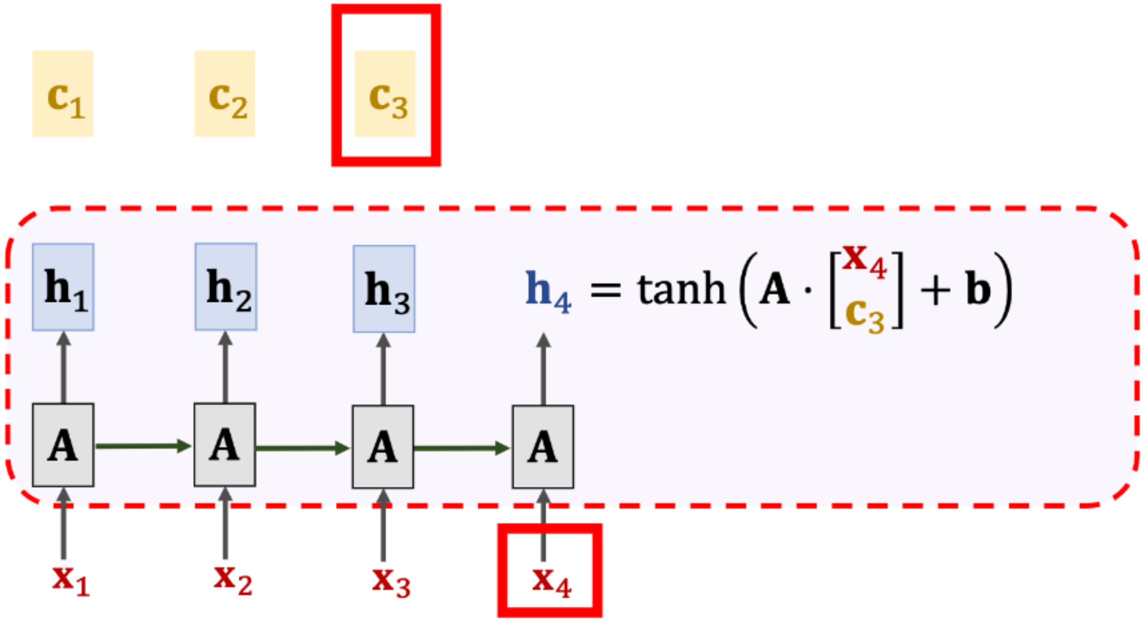


Image from https://github.com/wangshusen/DeepLearning/blob/master/Slides/9_RNN_9.pdf

Self-attention (intra-attention)

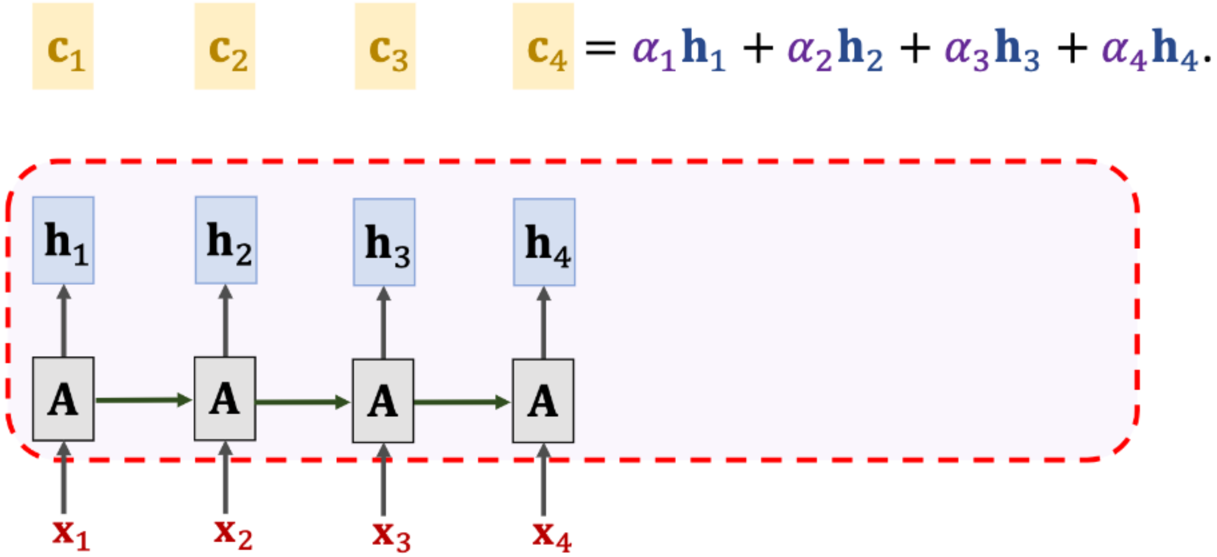


Image from https://github.com/wangshusen/DeepLearning/blob/master/Slides/9_RNN_9.pdf

Self-attention (intra-attention)

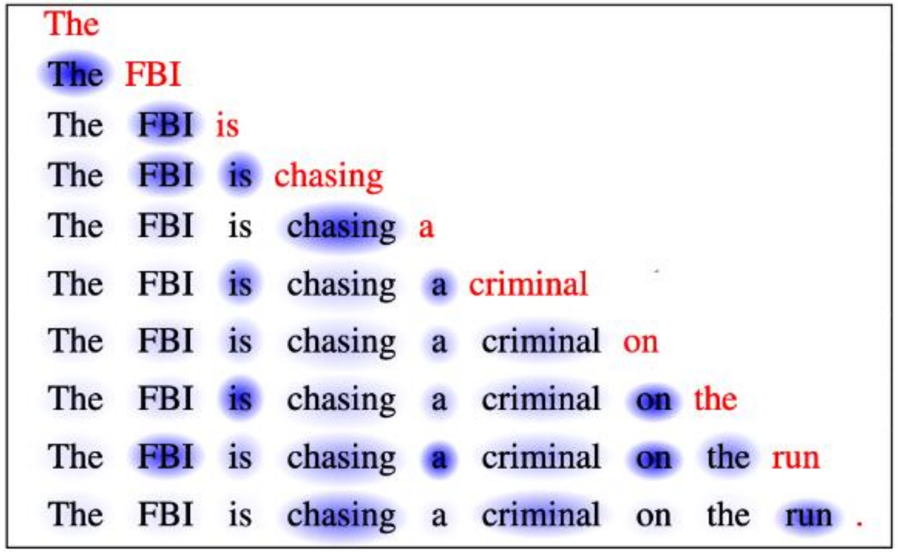
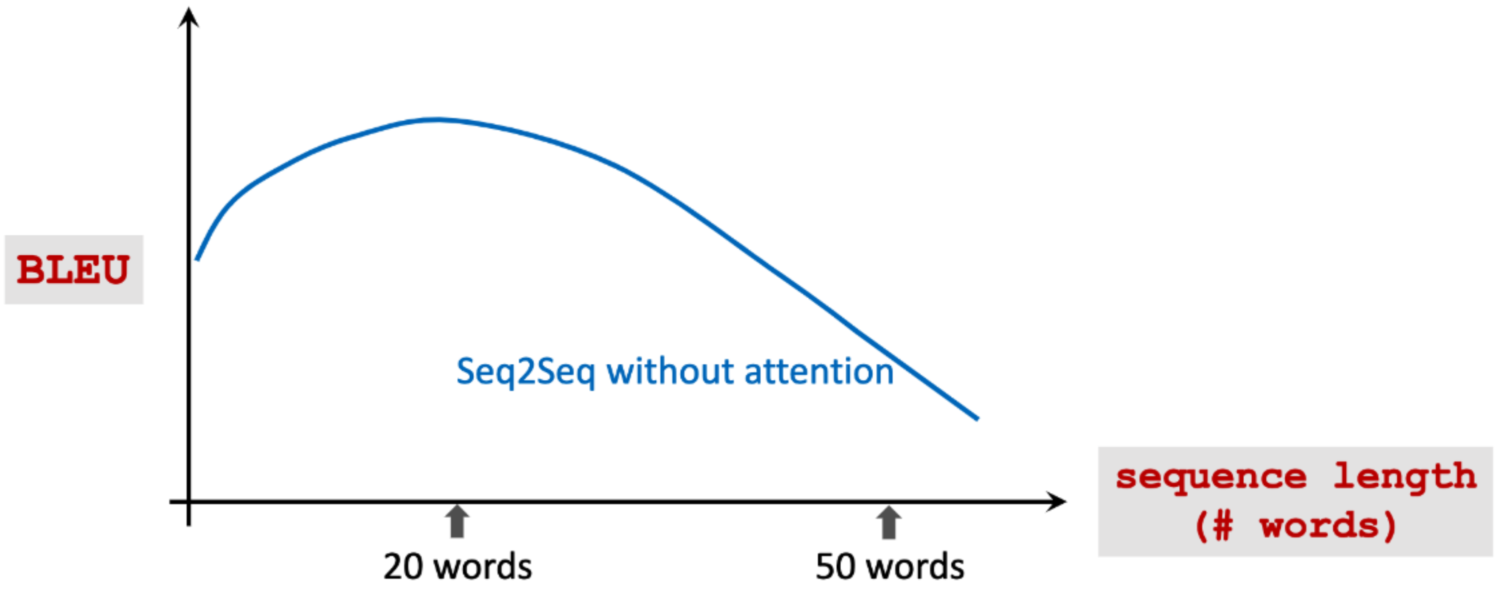


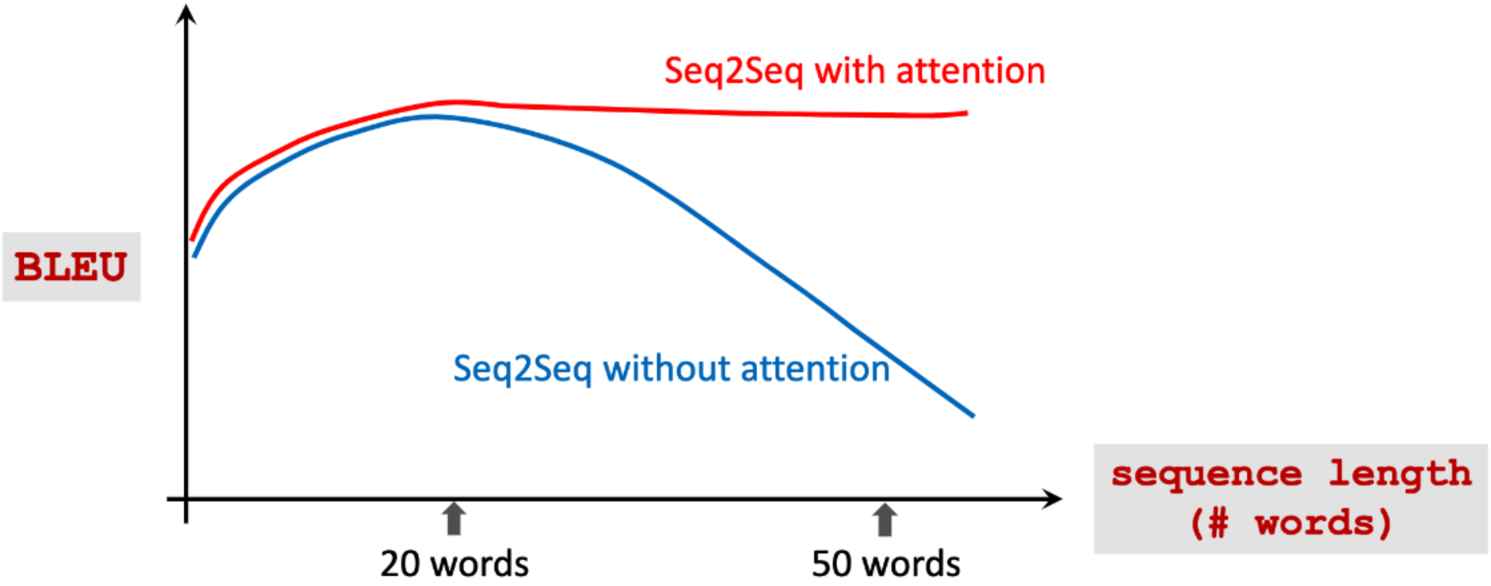
Figure is from the paper " Long Short-Term Memory-Networks for Machine Reading."

Pay attention to the context relevant to the new input

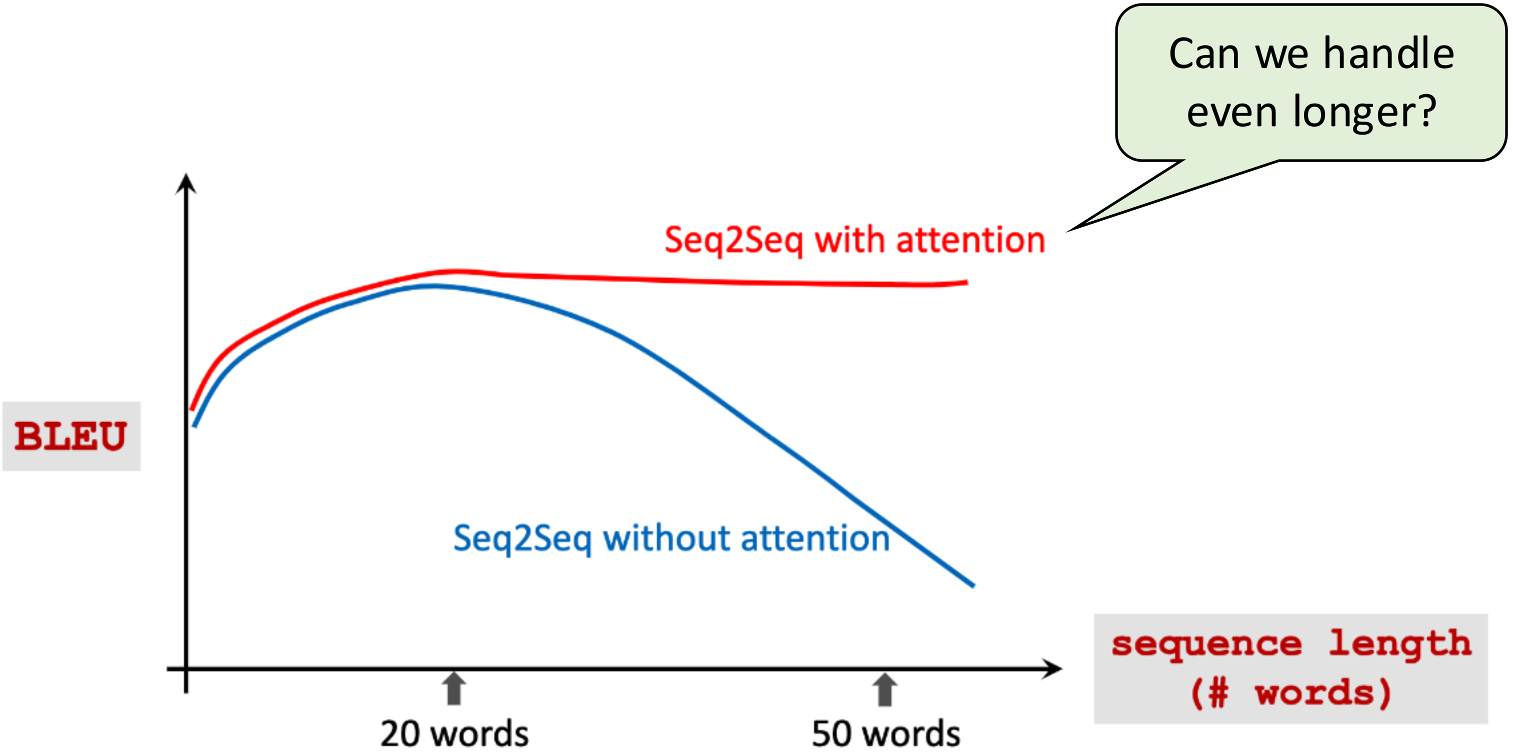
Seq2seq model performance



Seq2seq model performance



Seq2seq model performance



Transformer networks

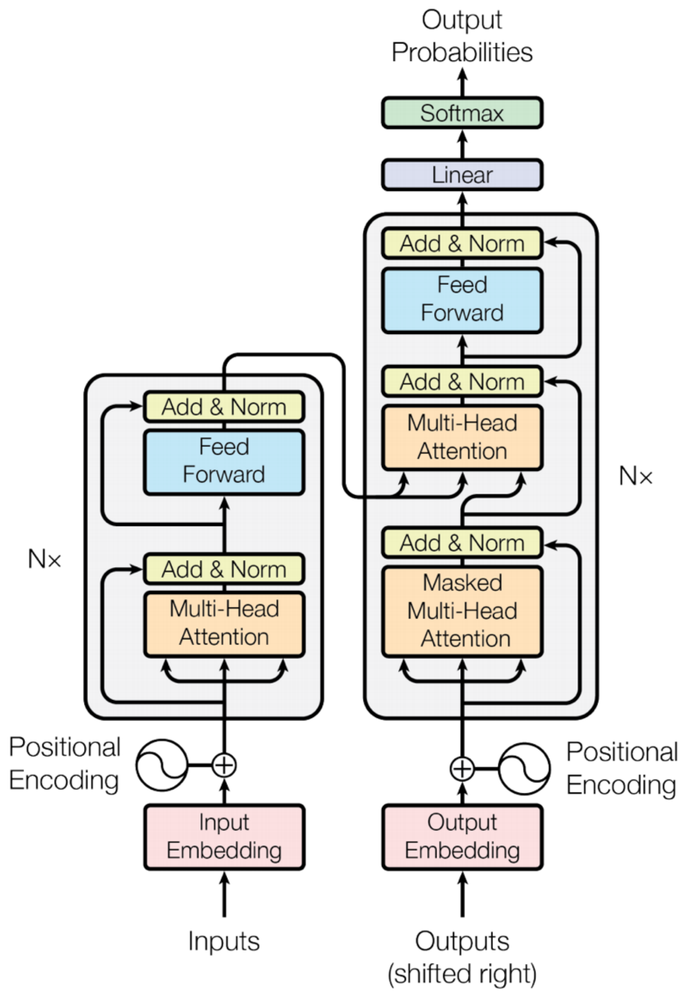


Figure 1: The Transformer - model architecture.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *arXiv preprint arXiv:1706.03762* (2017).

Transformer networks

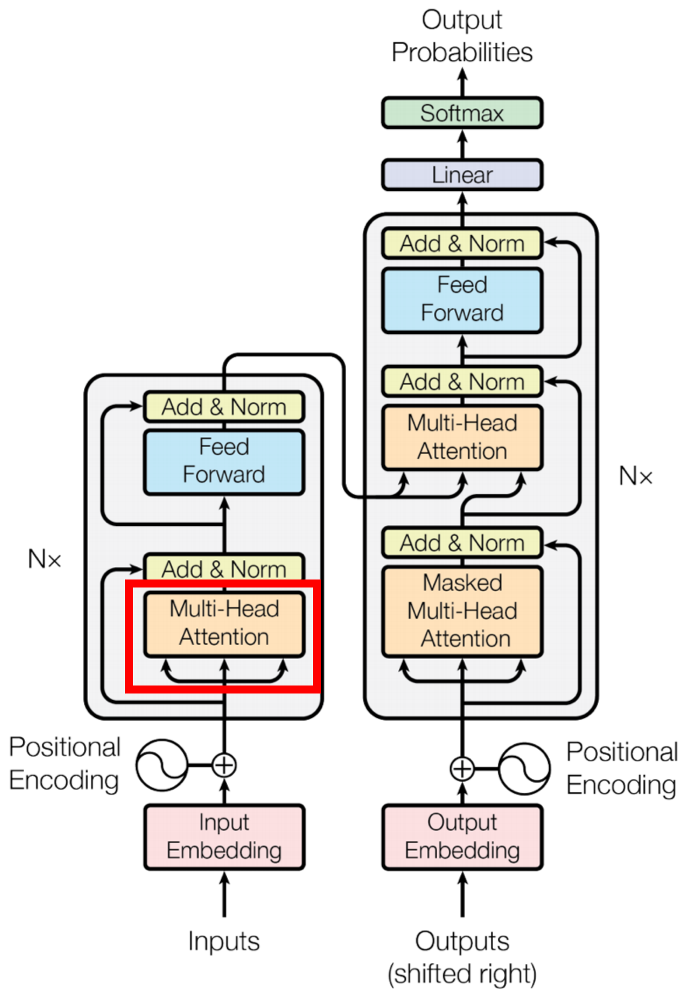


Figure 1: The Transformer - model architecture.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *arXiv preprint arXiv:1706.03762* (2017).

Transformer networks

3 inputs

1 output

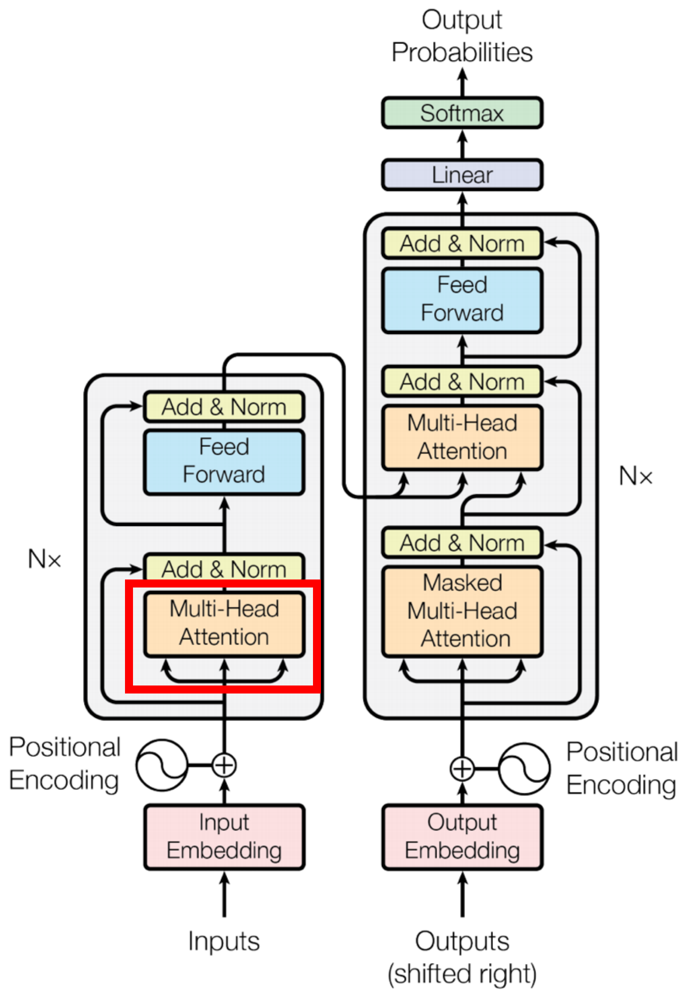
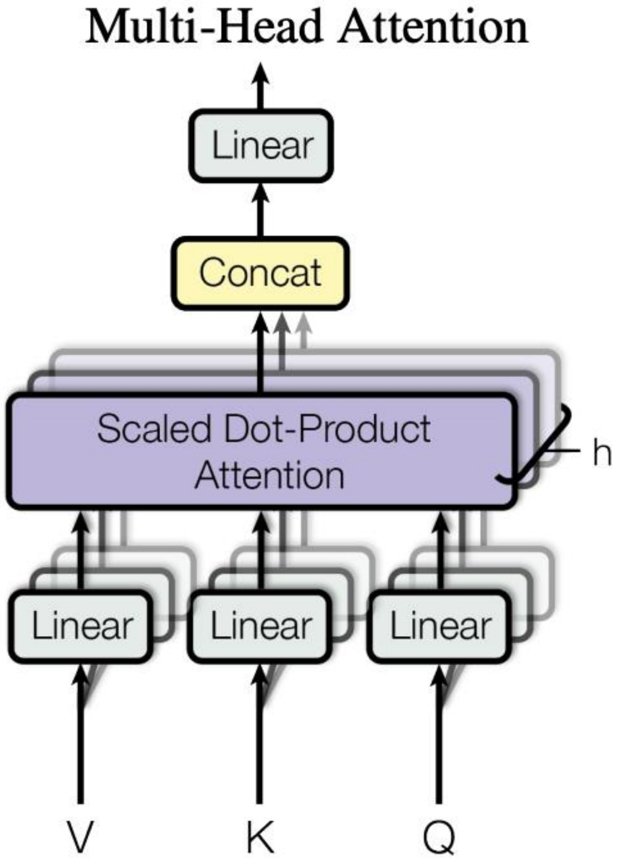


Figure 1: The Transformer - model architecture.

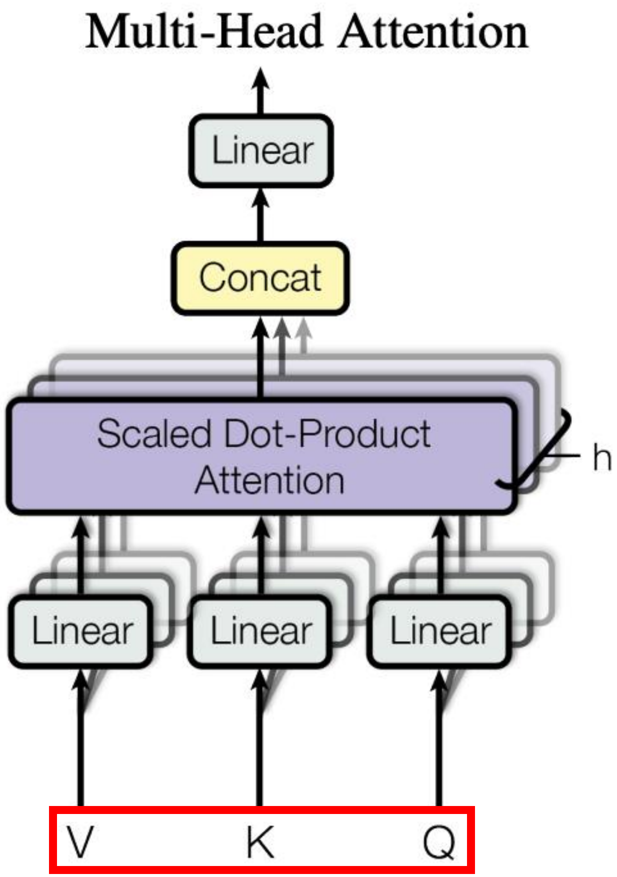
Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *arXiv preprint arXiv:1706.03762* (2017).

Transformer networks



Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *arXiv preprint arXiv:1706.03762* (2017).

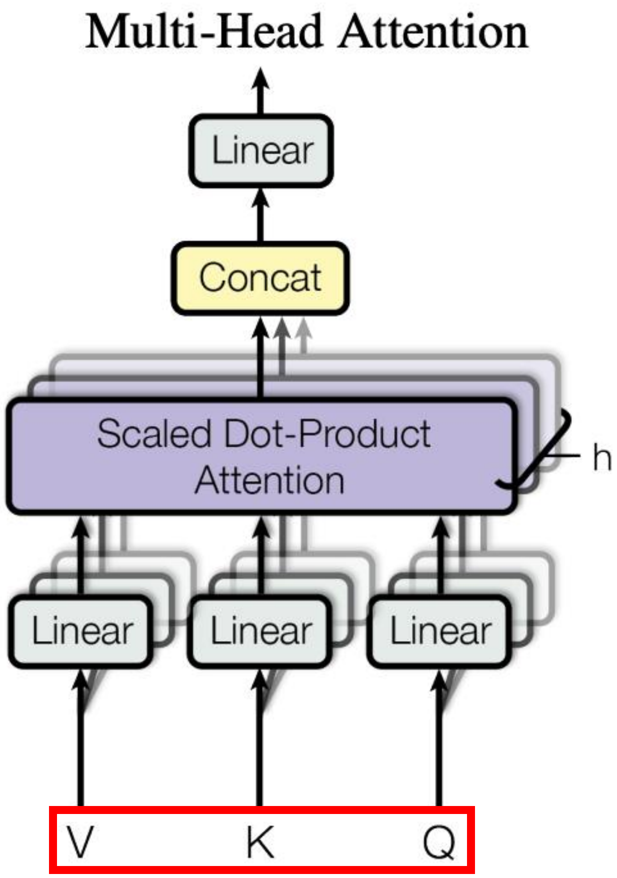
Transformer networks



V: value
K: key
Q: query

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *arXiv preprint arXiv:1706.03762* (2017).

Transformer networks

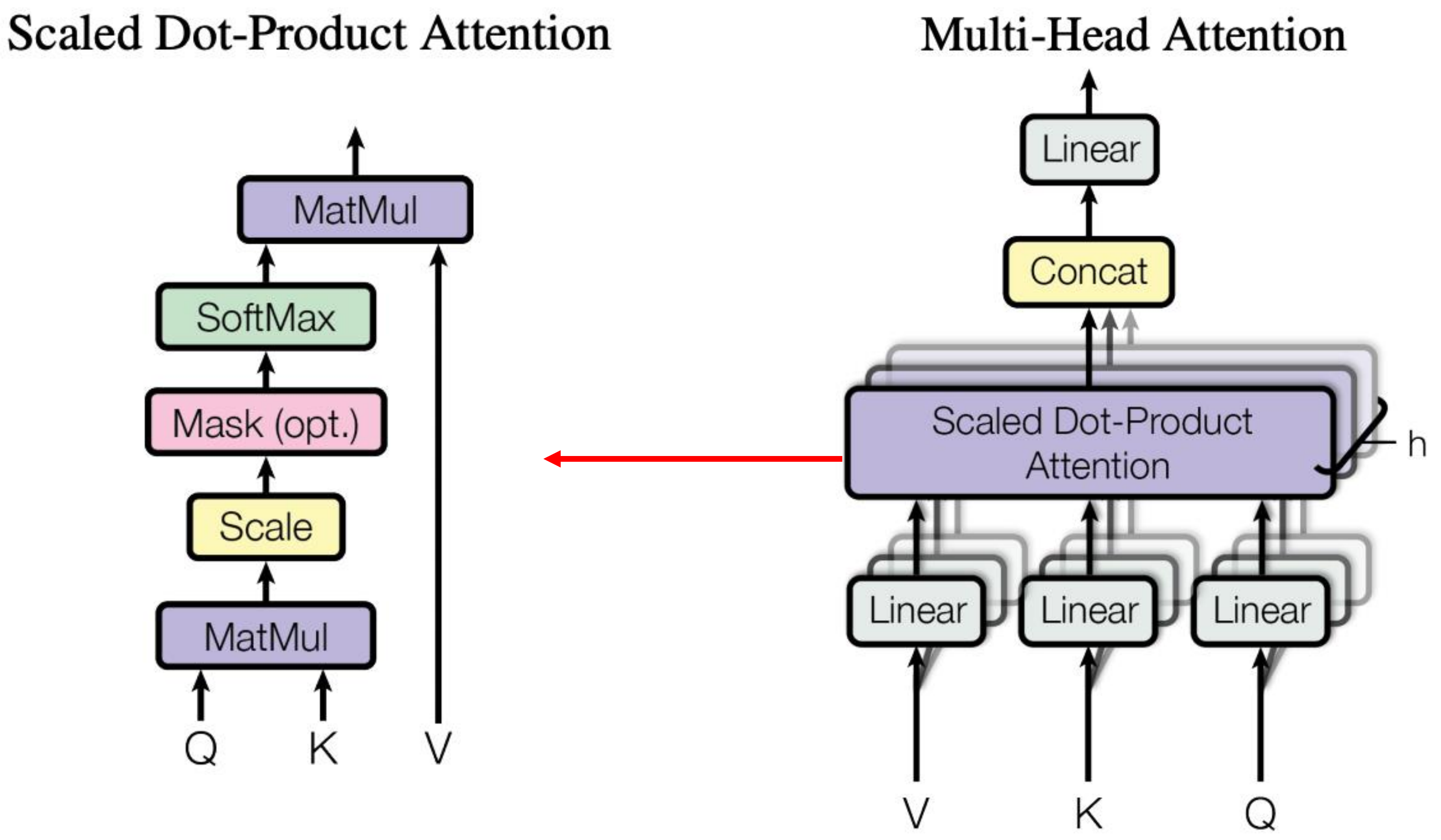


V: value
K: key
Q: query

Interpreted from information retrieval systems: https://stats.stackexchange.com/questions/421935/what-exactly-are-keys-queries-and-values-in-attention-mechanisms

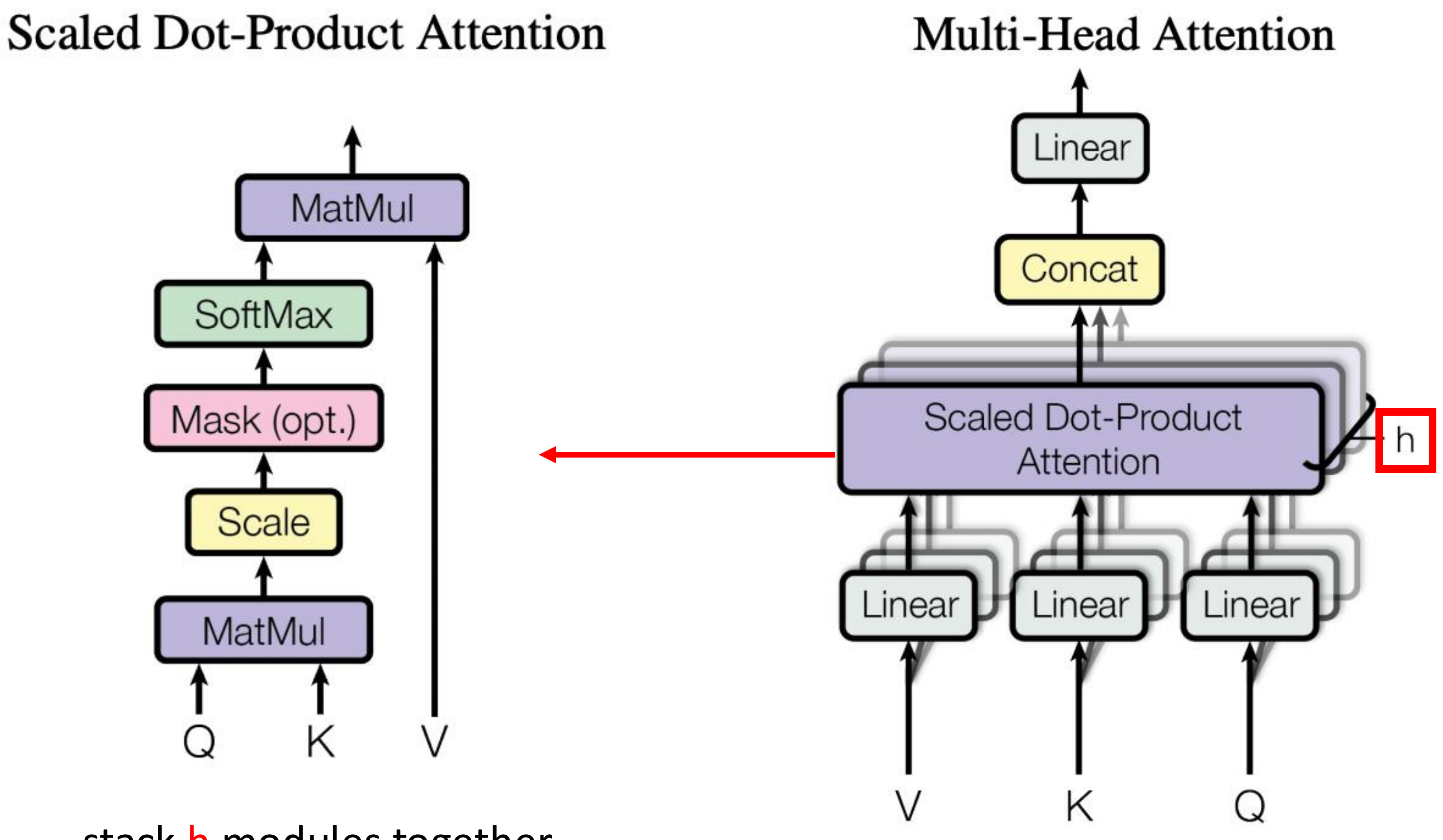
Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *arXiv preprint arXiv:1706.03762* (2017).

Transformer networks



Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *arXiv preprint arXiv:1706.03762* (2017).

Transformer networks

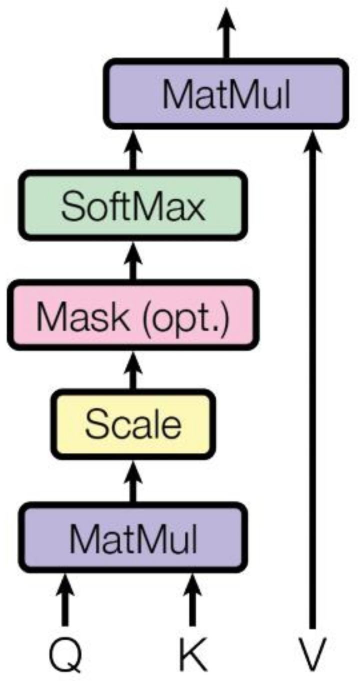


stack h modules together

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *arXiv preprint arXiv:1706.03762* (2017).

Transformer networks

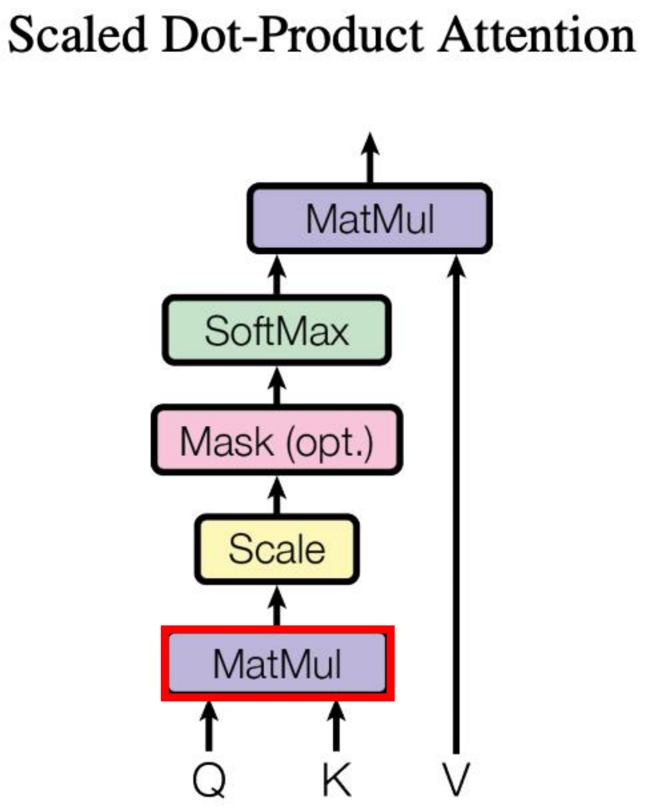
Scaled Dot-Product Attention



$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *arXiv preprint arXiv:1706.03762* (2017).

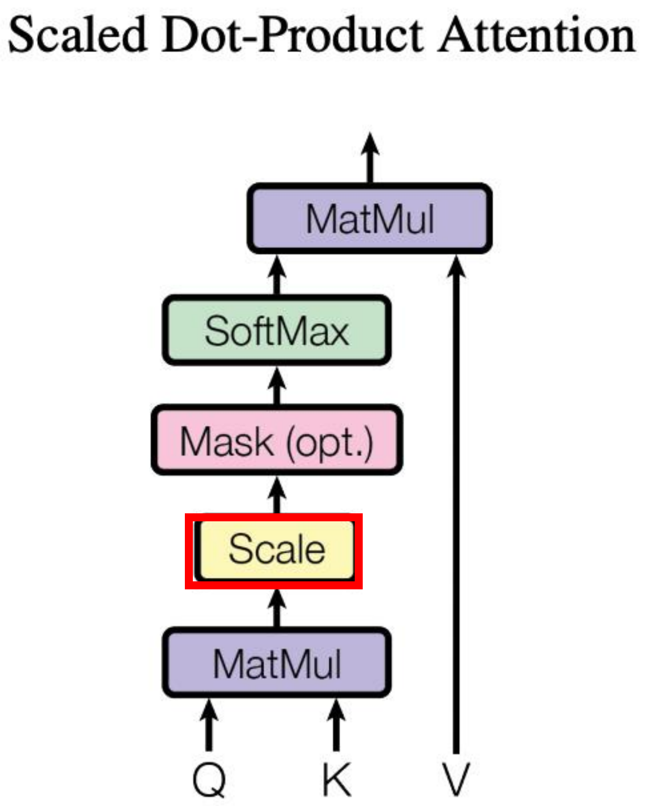
Transformer networks



$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{\boxed{QK^T}}{\sqrt{d_k}}\right)V$$

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *arXiv preprint arXiv:1706.03762* (2017).

Transformer networks

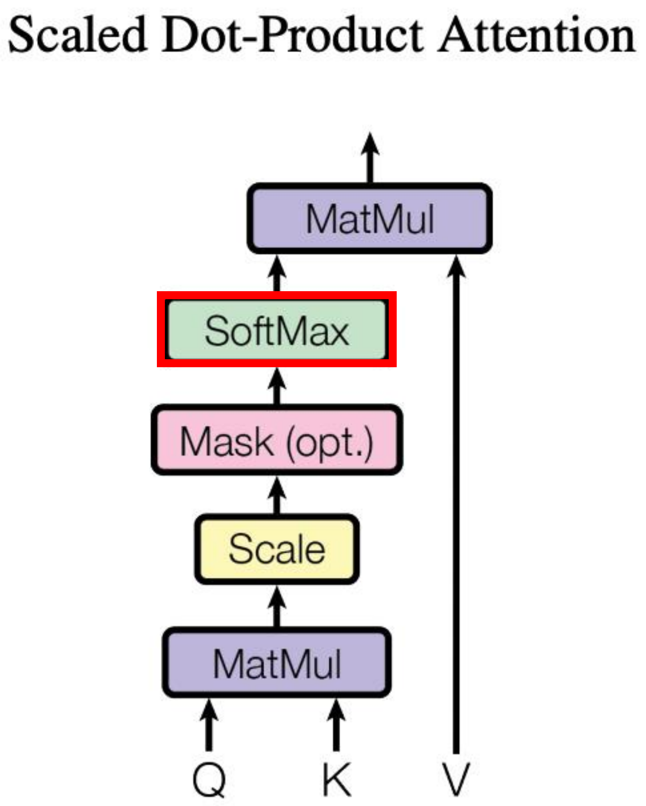


$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\boxed{\sqrt{d_k}}}\right)V$$

Dimension of keys

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *arXiv preprint arXiv:1706.03762* (2017).

Transformer networks

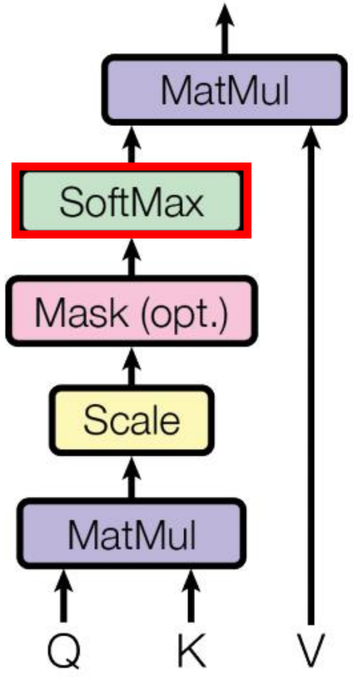


$$\text{Attention}(Q, K, V) = \boxed{\text{softmax}}(\frac{QK^T}{\sqrt{d_k}})V$$

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *arXiv preprint arXiv:1706.03762* (2017).

Transformer networks

Scaled Dot-Product Attention



weights for selection (keys in a retrieval system)

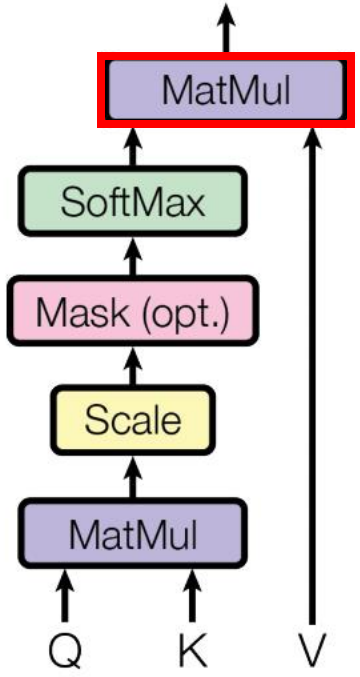
$$\mathrm{Attention}(Q,K,V) = \boxed{\mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})}V$$

An example: one-hot format

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *arXiv preprint arXiv:1706.03762* (2017).

Transformer networks

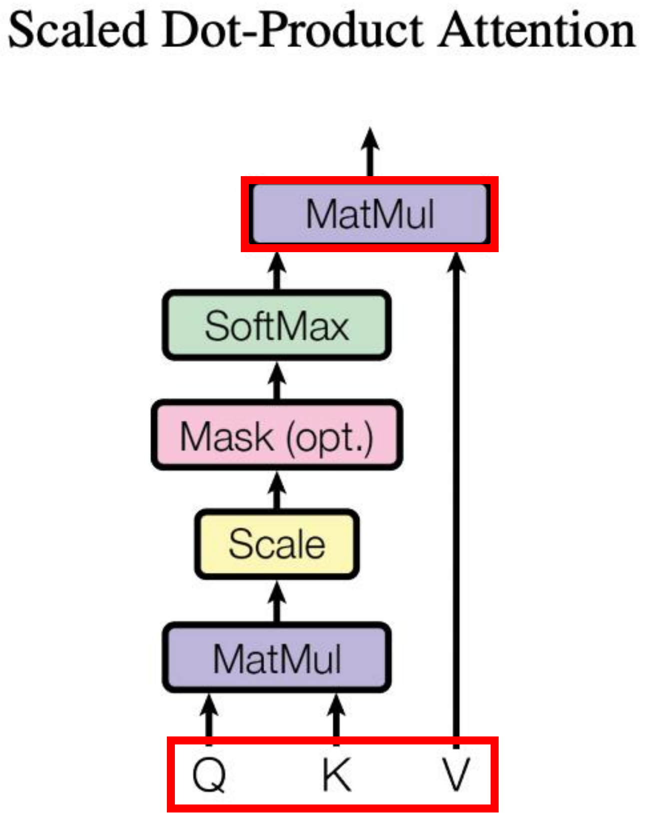
Scaled Dot-Product Attention



$$\mathrm{Attention}(Q,K,V) = \boxed{\mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V}$$

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *arXiv preprint arXiv:1706.03762* (2017).

Transformer networks



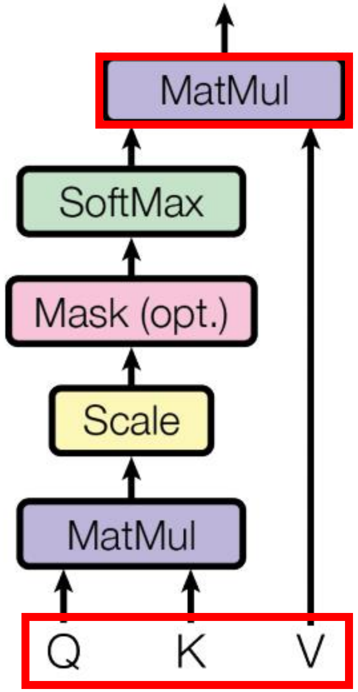
How to generate

$$\mathrm{Attention}(Q,K,V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *arXiv preprint arXiv:1706.03762* (2017).

Transformer networks

Scaled Dot-Product Attention



In addition to attention sub-layers, each of the layers in our encoder and decoder contains a fully connected feed-forward network, which is applied to each position separately and identically. This consists of two linear transformations with a ReLU activation in between.

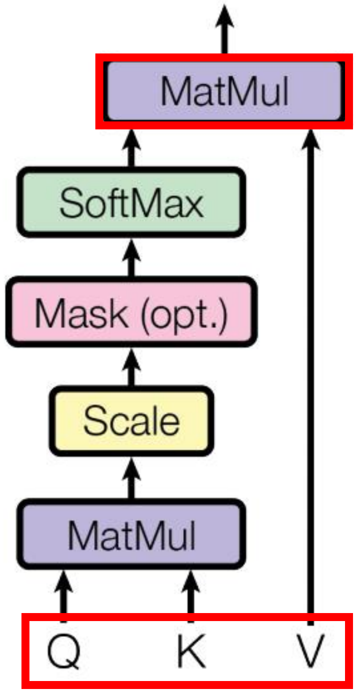
$$\boxed{\mathrm{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2} \tag{2}$$

While the linear transformations are the same across different positions, they use different parameters from layer to layer. Another way of describing this is as two convolutions with kernel size 1. The dimensionality of input and output is $d_{\text{model}} = 512$, and the inner-layer has dimensionality $d_{ff} = 2048$.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *arXiv preprint arXiv:1706.03762* (2017).

Transformer networks

Scaled Dot-Product Attention



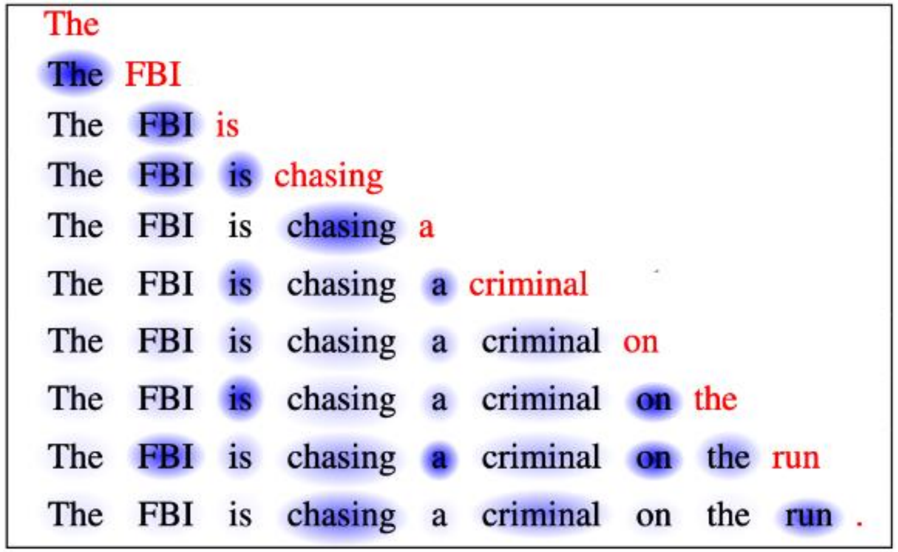
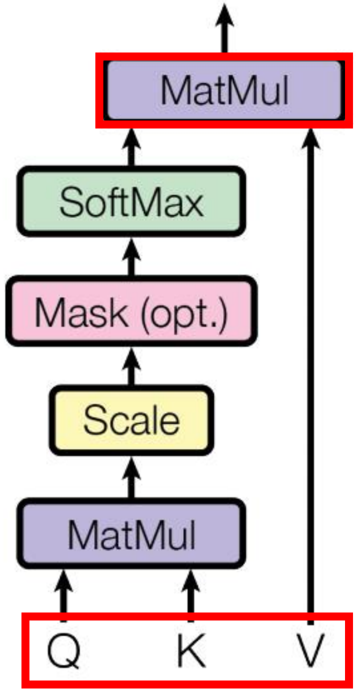


Figure is from the paper " Long Short-Term Memory-Networks for Machine Reading."

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *arXiv preprint arXiv:1706.03762* (2017).

Transformer networks

Scaled Dot-Product Attention



Greater weights (larger Softmax output)

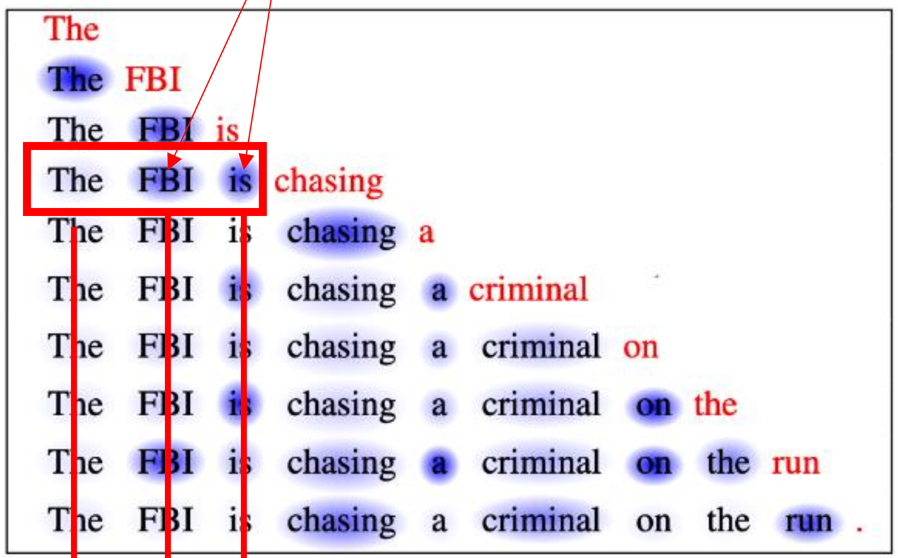


Figure is from the paper " Long Short-Term Memory-Networks for Machine Reading."

Embedding feature vectors
(a matrix)

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *arXiv preprint arXiv:1706.03762* (2017).

Transformer networks

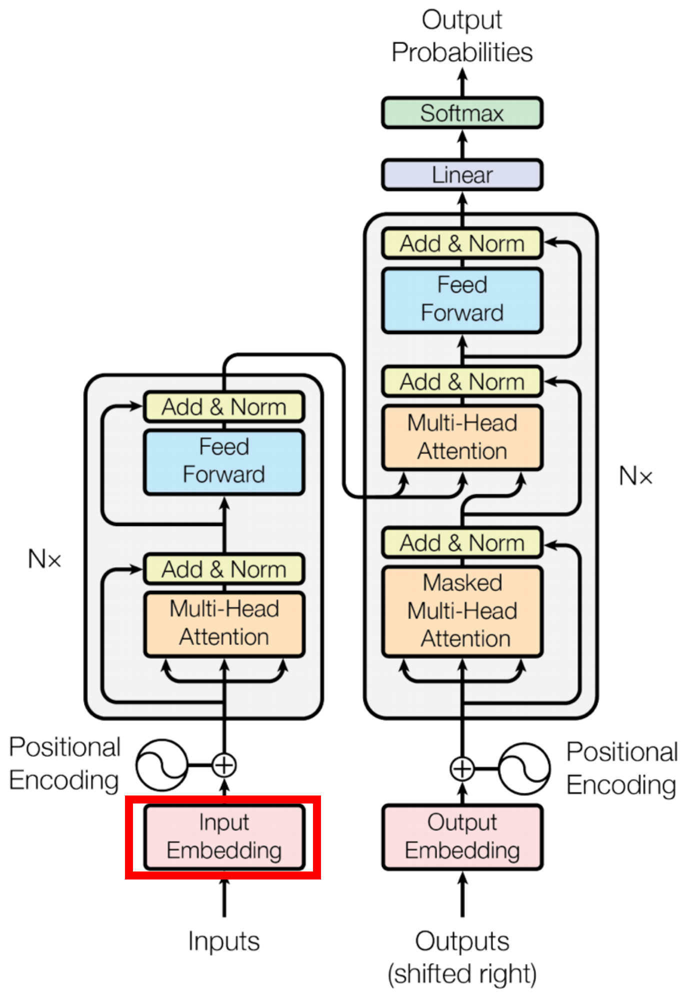


Figure 1: The Transformer - model architecture.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *arXiv preprint arXiv:1706.03762* (2017).

Transformer networks

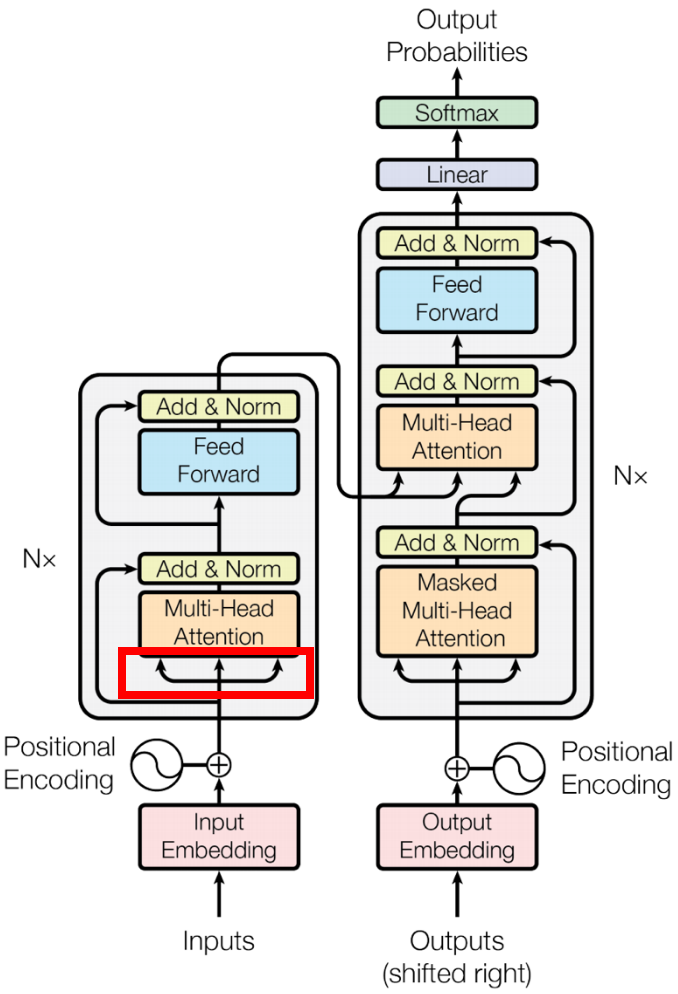


Figure 1: The Transformer - model architecture.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *arXiv preprint arXiv:1706.03762* (2017).

Transformer networks

Shortcut connection

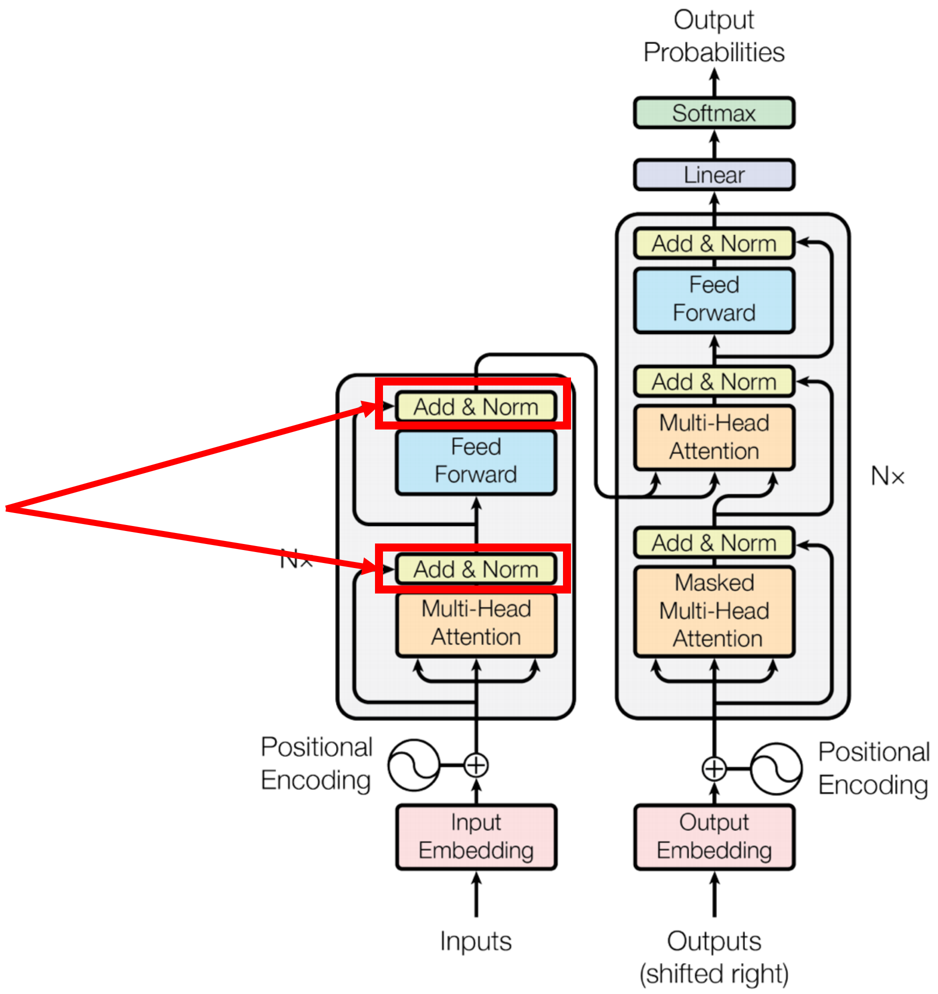


Figure 1: The Transformer - model architecture.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *arXiv preprint arXiv:1706.03762* (2017).

Transformer networks

Shortcut connection
(residual connection)

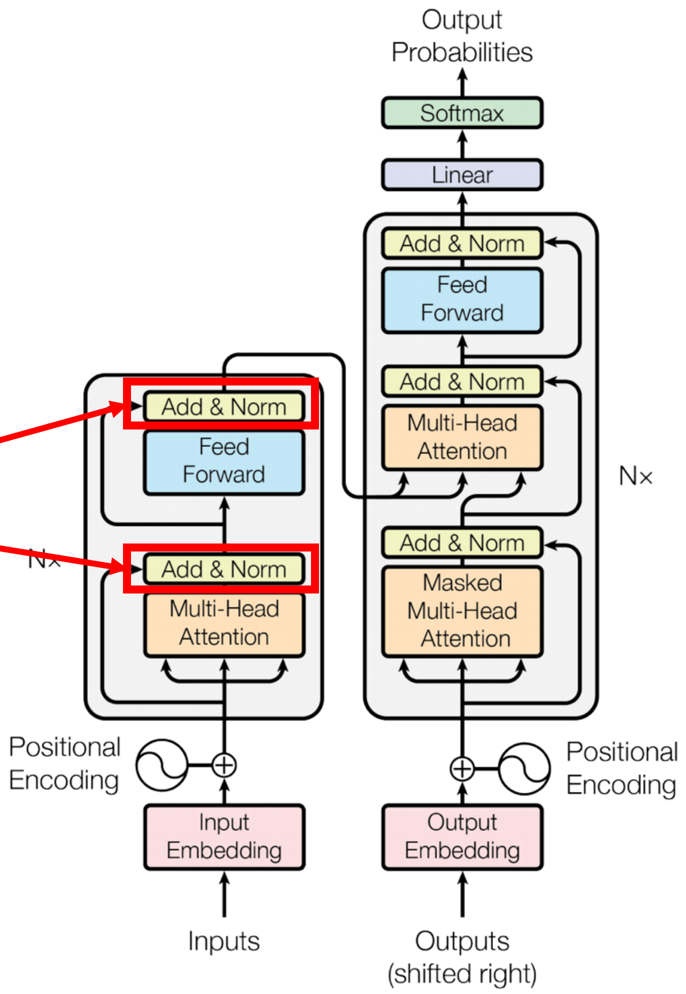


Figure 1: The Transformer - model architecture.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *arXiv preprint arXiv:1706.03762* (2017).

Transformer networks

No recurrent structure → not RNN

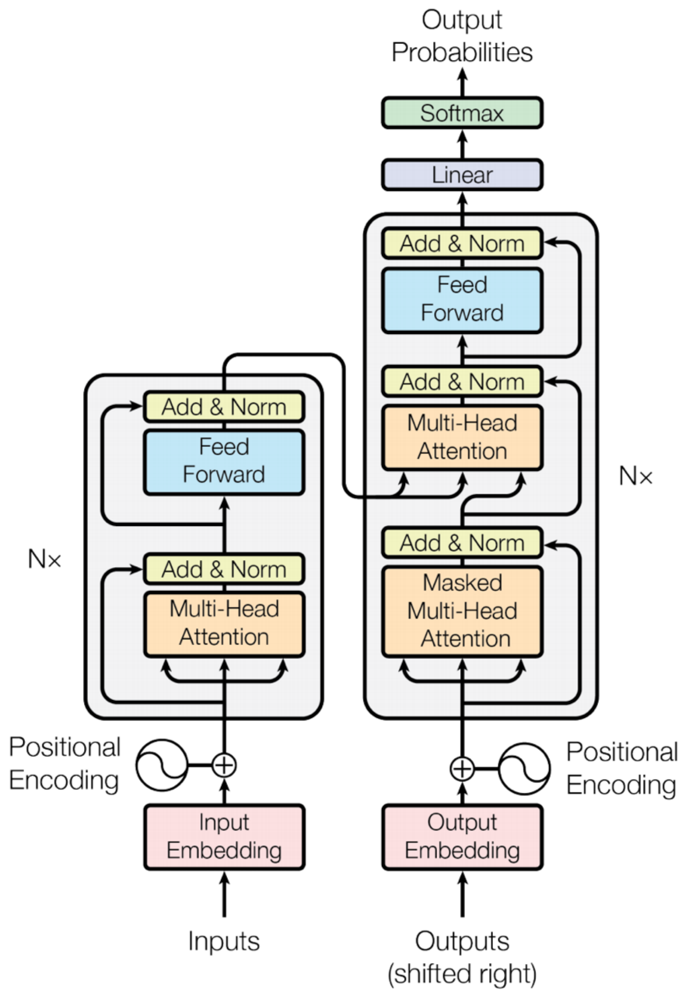


Figure 1: The Transformer - model architecture.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *arXiv preprint arXiv:1706.03762* (2017).

Transformer networks

No recurrent structure → not RNN

Q: any benefit?

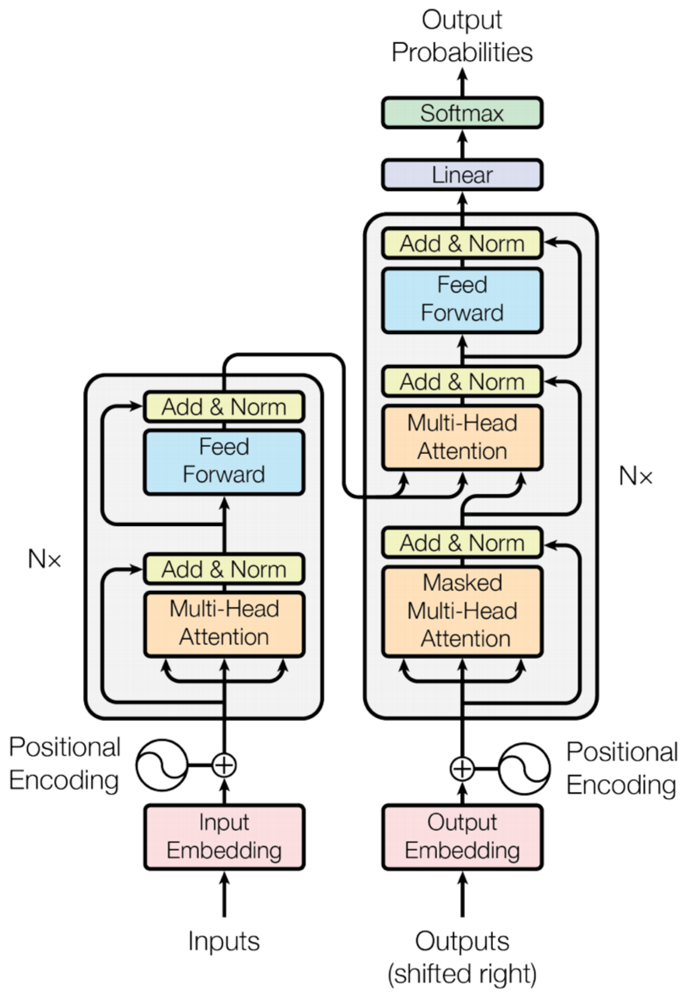


Figure 1: The Transformer - model architecture.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *arXiv preprint arXiv:1706.03762* (2017).

Transformer networks

No recurrent structure → not RNN

Q: any benefit?

1. No sequential dependence

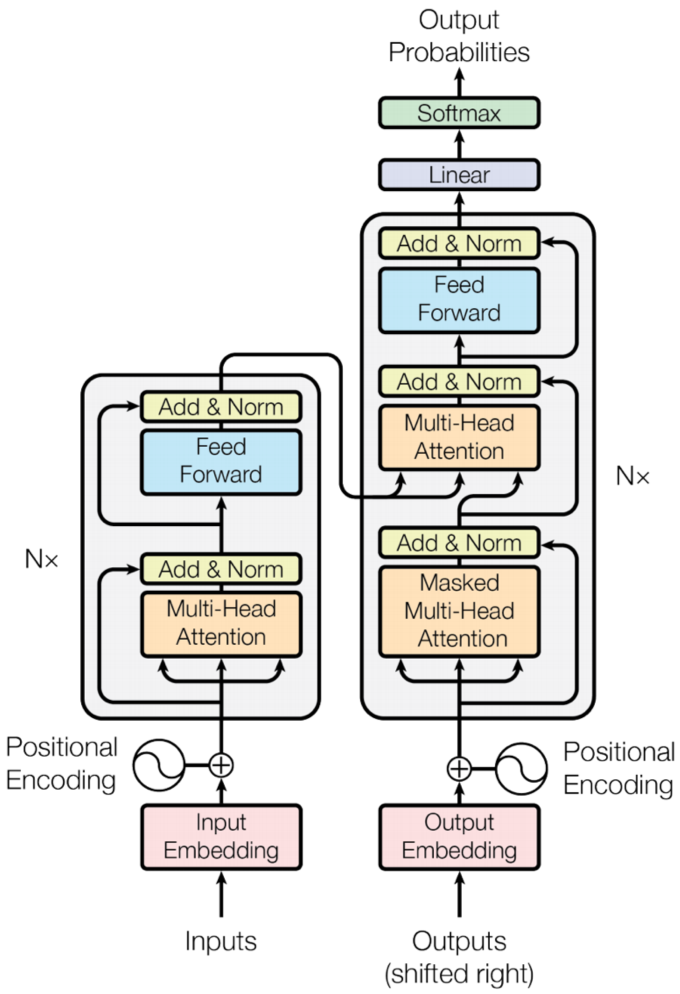


Figure 1: The Transformer - model architecture.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *arXiv preprint arXiv:1706.03762* (2017).

Transformer networks

No recurrent structure → not RNN

Q: any benefit?

1. No sequential dependence
2. Parallel processing

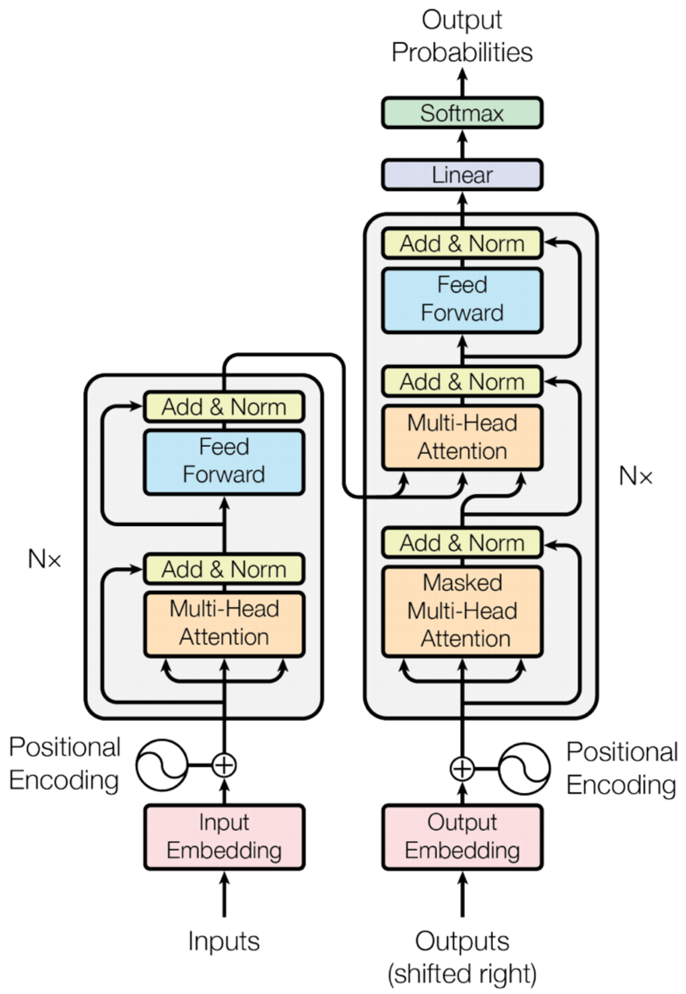


Figure 1: The Transformer - model architecture.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *arXiv preprint arXiv:1706.03762* (2017).

Transformer networks

No recurrent structure → not RNN

Q: any benefit?

1. No sequential dependence
2. Parallel processing
3. …

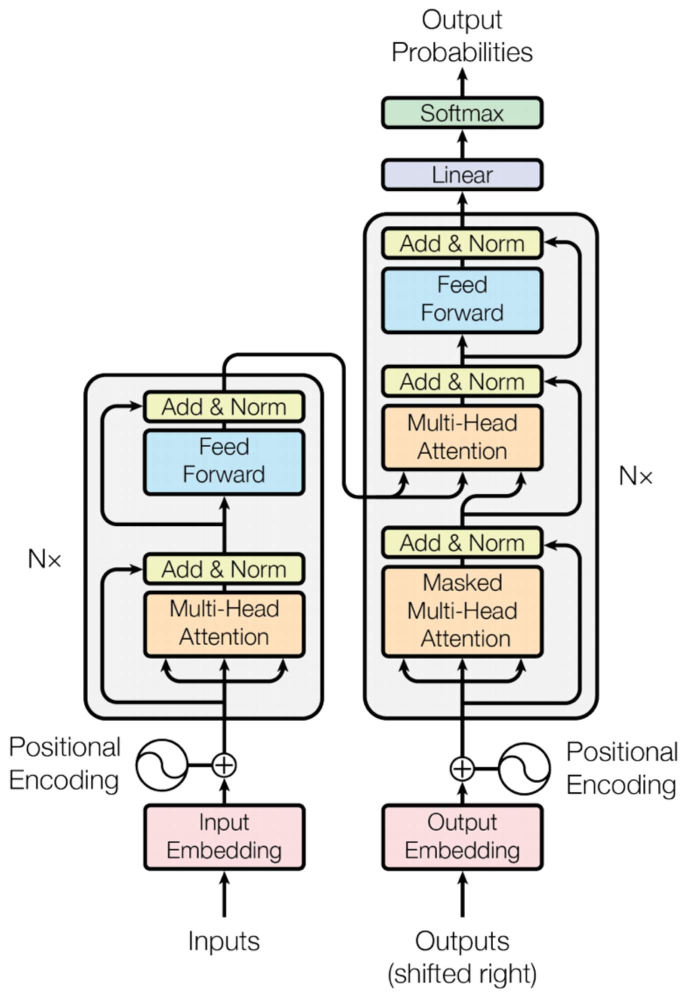
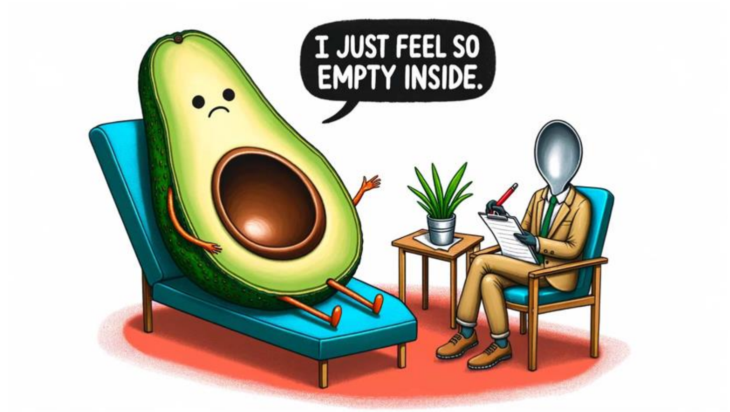


Figure 1: The Transformer - model architecture.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *arXiv preprint arXiv:1706.03762* (2017).

AI products based on Transformers

ChatGPT



AlphaFold