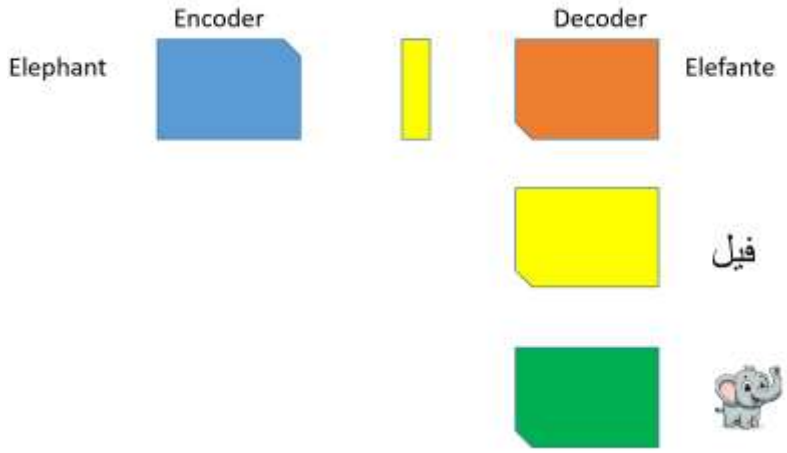
Lecture 9

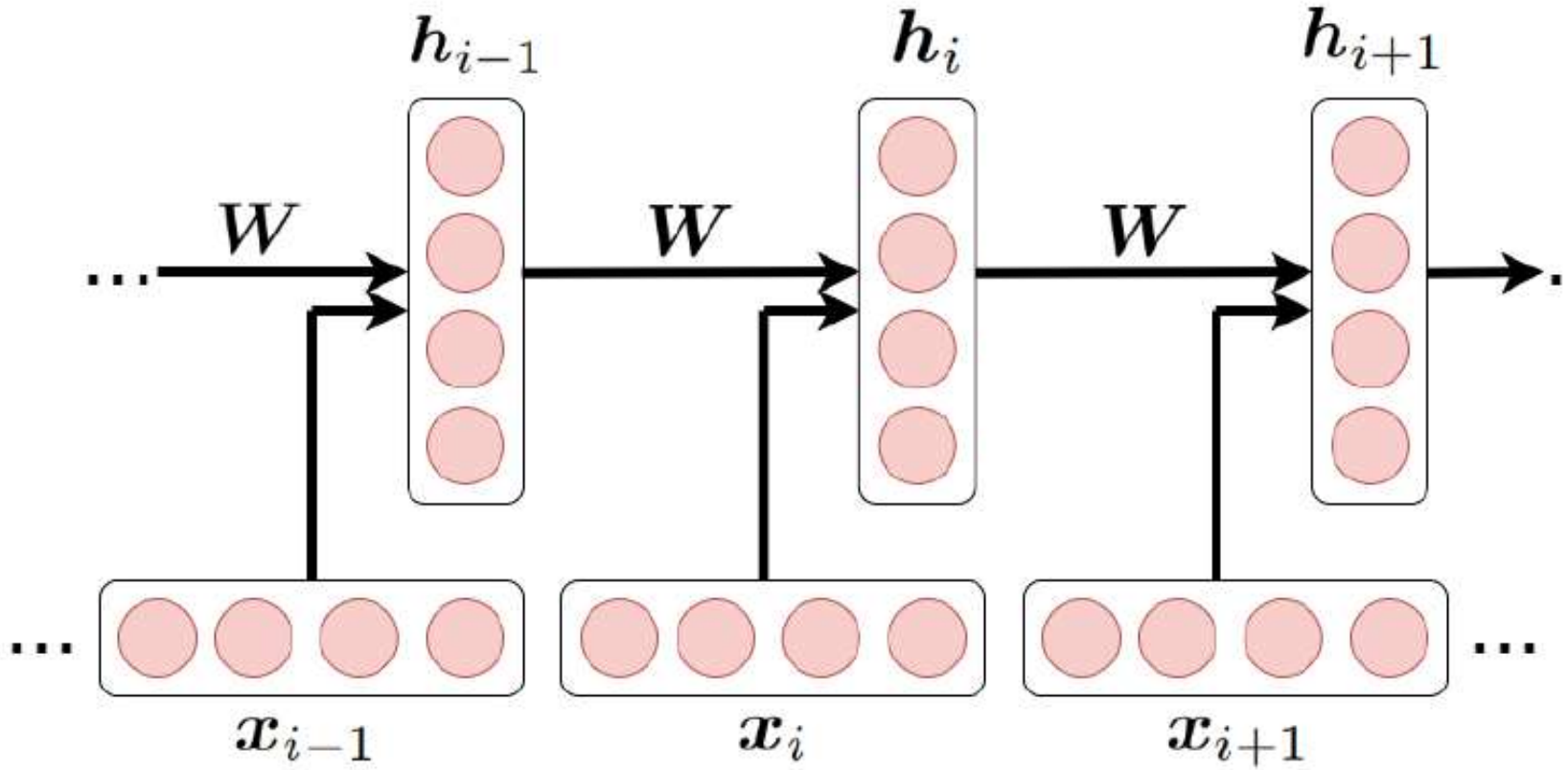
Attention Mechanism

Common representation

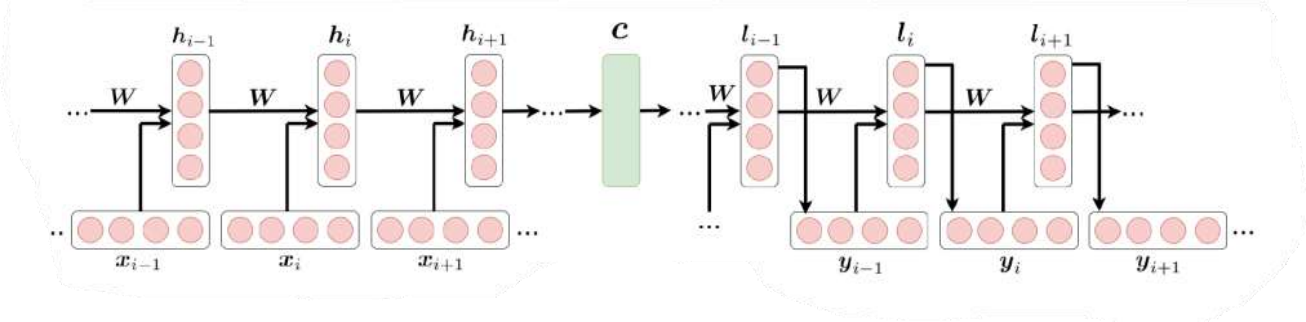
- One 'concept' represented universally, transcending language or form.
- **Encoder:** Analyzes 'elephant' from source
- **Output:** Universal representation vector (The 'concept' of an elephant).
- **Decoders:** Interpret the concept into various domains.
- The 'concept' is abstract, existing beyond language or specific representation.



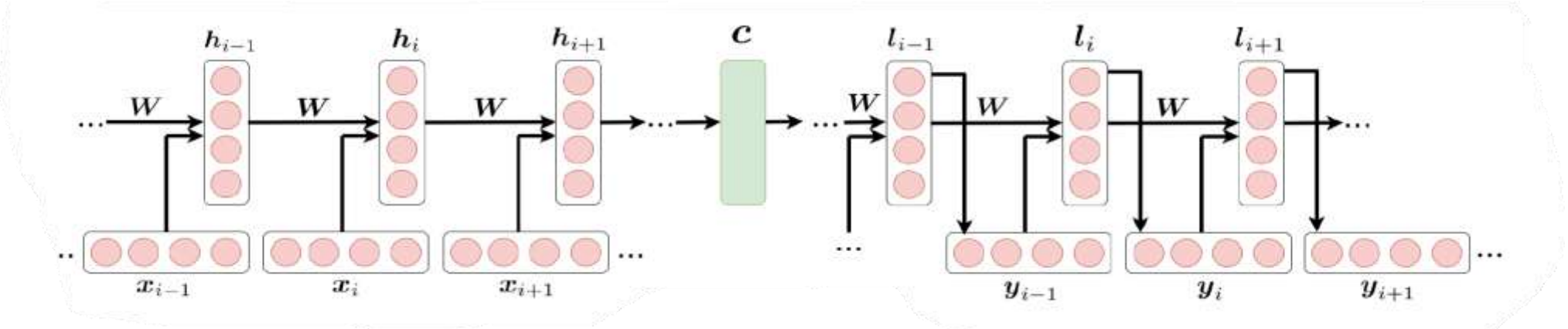
Recurrent neural network



Sequence-to-Sequence Model



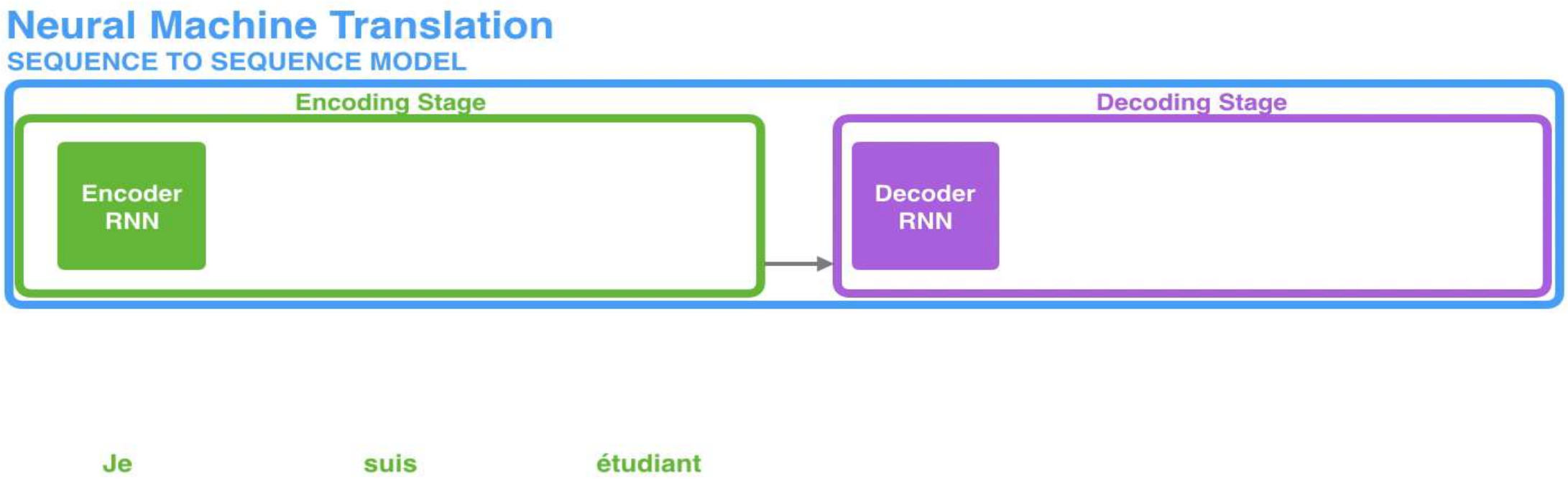
Sequence-to-Sequence Model



In the sequence-to-sequence model, every word x_i produces a hidden vector h_i in the encoder part of the autoencoder. The hidden vector of every word, h_i, is fed to the next hidden vector, h_{i+1}, by a projection matrix W.

In this model, for the whole sequence, there is only one context vector c which is equal to the last hidden vector of the encoder, i.e., $c = h_n$.

Sequence to sequence



Credit: Jay Alammarz
https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/

Challenges

- The long-range dependencies.

- The sequential nature of the model architecture prevents parallelization.

Attention

- The basic idea behind the attention mechanism is directing the focus on important factors when processing data.



Attention

- The basic idea behind the attention mechanism is directing the focus on important factors when processing data.

- Attention is a fancy name for weighted average.



Attention

- The basic idea behind the attention mechanism is directing the focus on important factors when processing data.

- Attention is a fancy name for weighted average.

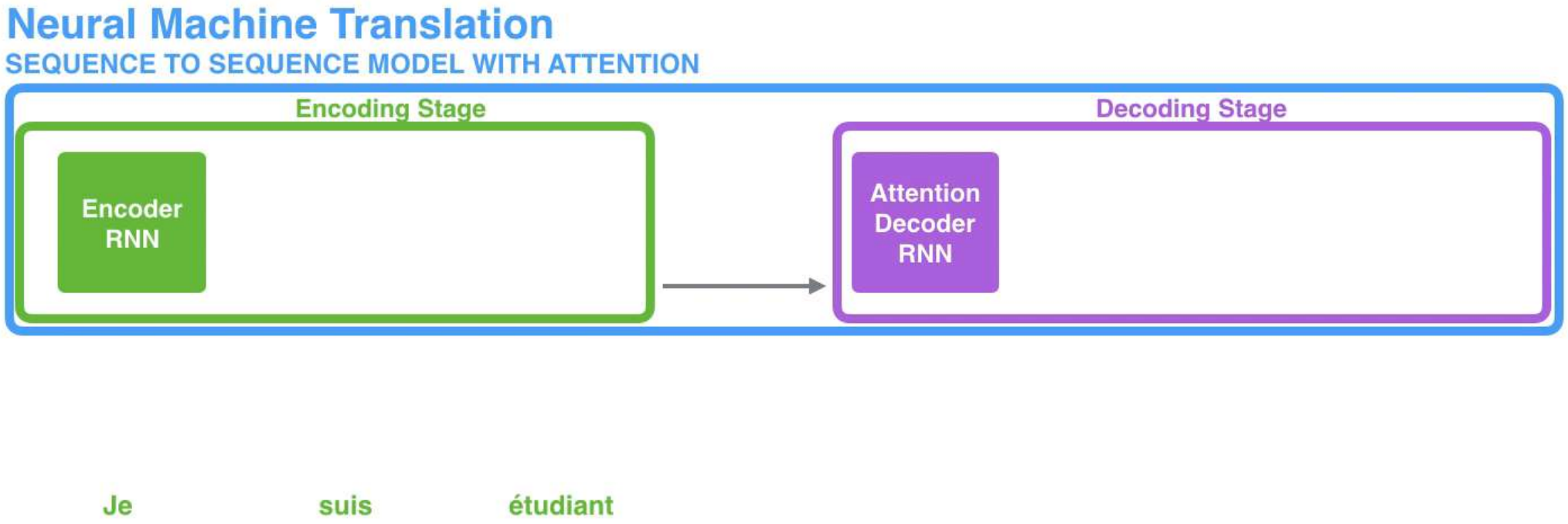


- Bahdanau et al., 2014

and

- Luong et al., 2015

Sequence to sequence with attention



Credit: Jay Alammarz

Attention is weighted average

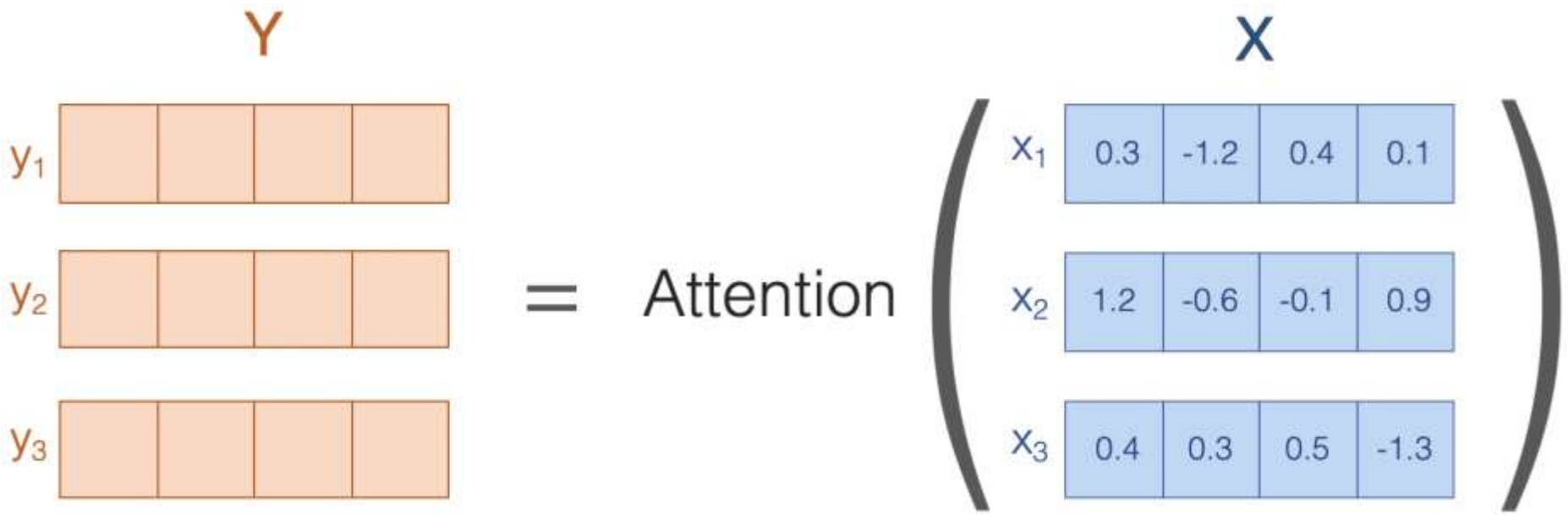


Figure: Jay Alammarz

Neural machine translation by jointly learning to align and translate (2014)

- Sequence to sequence model:

$p(y_i|y_1, \ldots, y_{i-1})$

Neural machine translation by jointly learning to align and translate (2014)

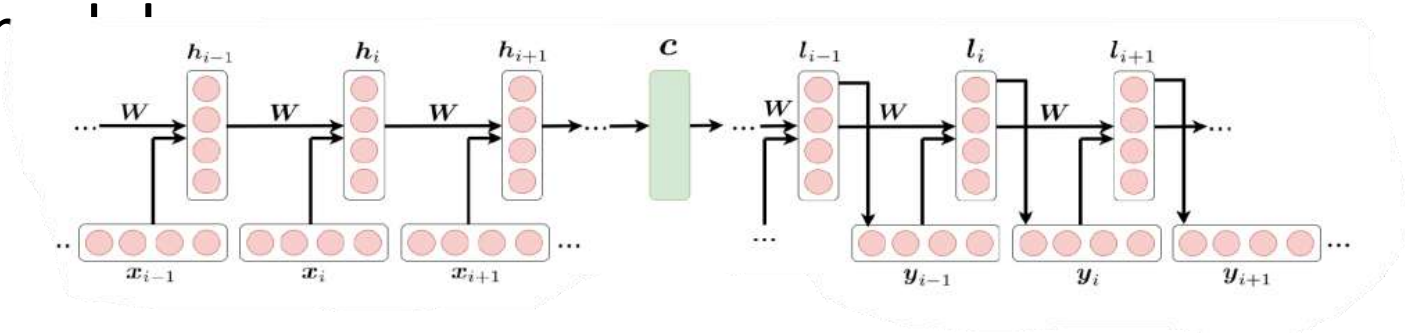
- Sequence to sequence model:

$$p(y_i|y_1, \ldots, y_{i-1}) = g(y_{i-1}, l_i, c\ \ )$$

Neural machine translation by jointly learning to align and translate (2014)

- Sequence to sequence m

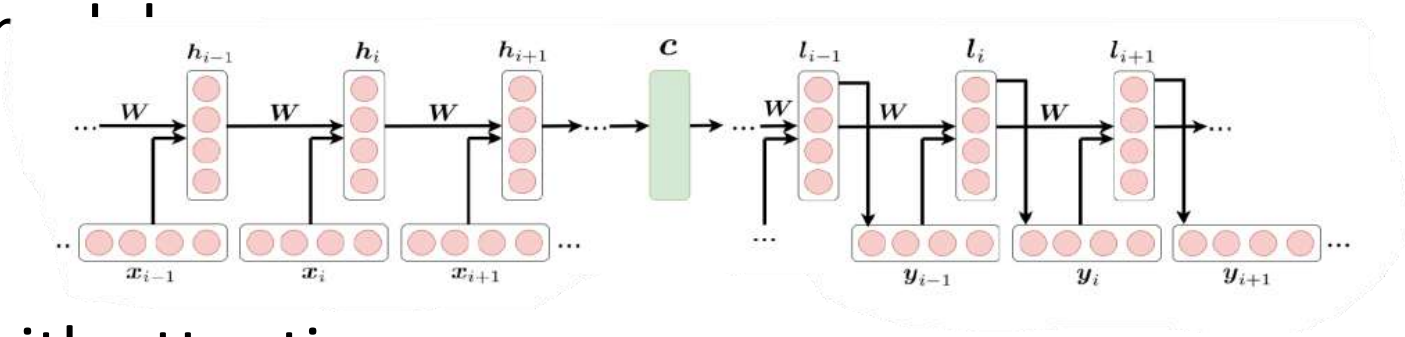
$$p(y_i|y_1,\ldots,y_{i-1}) = g(y_{i-1}, l_i, c\ )$$



Neural machine translation by jointly learning to align and translate (2014)

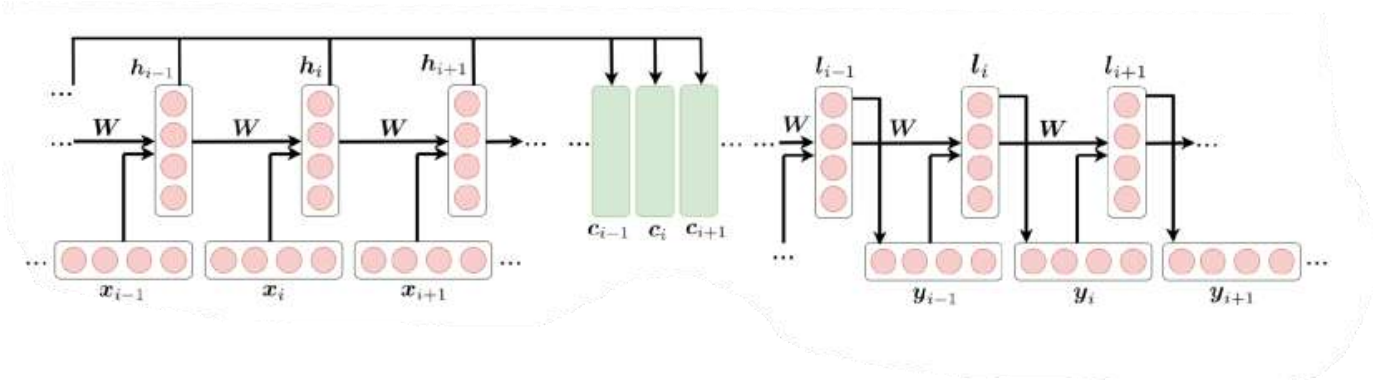
- Sequence to sequence m

$$p(y_i|y_1,\ldots,y_{i-1}) = g(y_{i-1}, l_i, c\ \ )$$

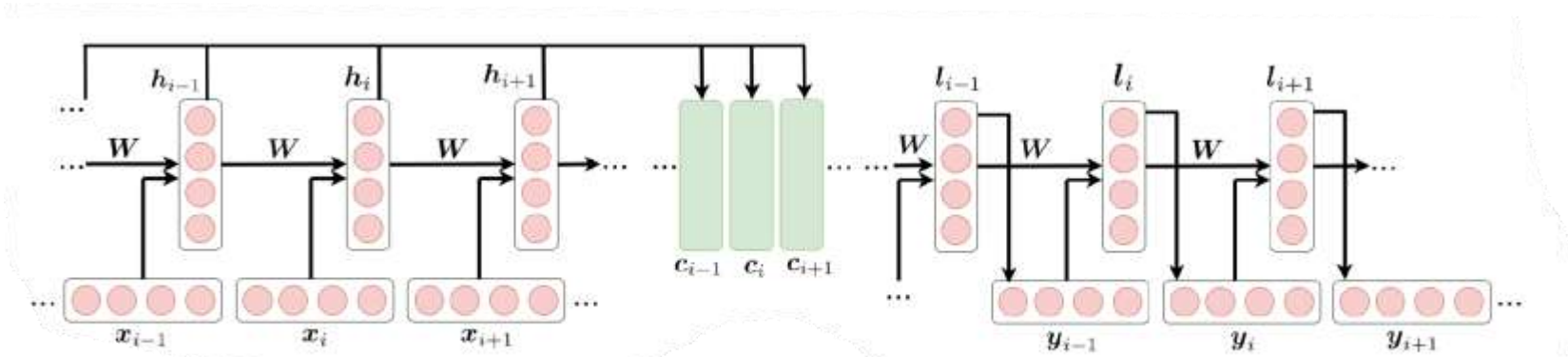


- Sequence to sequence with attention:

$$p(y_i|y_1,\ldots,y_{i-1}) = g(y_{i-1}, l_i, c_i)$$



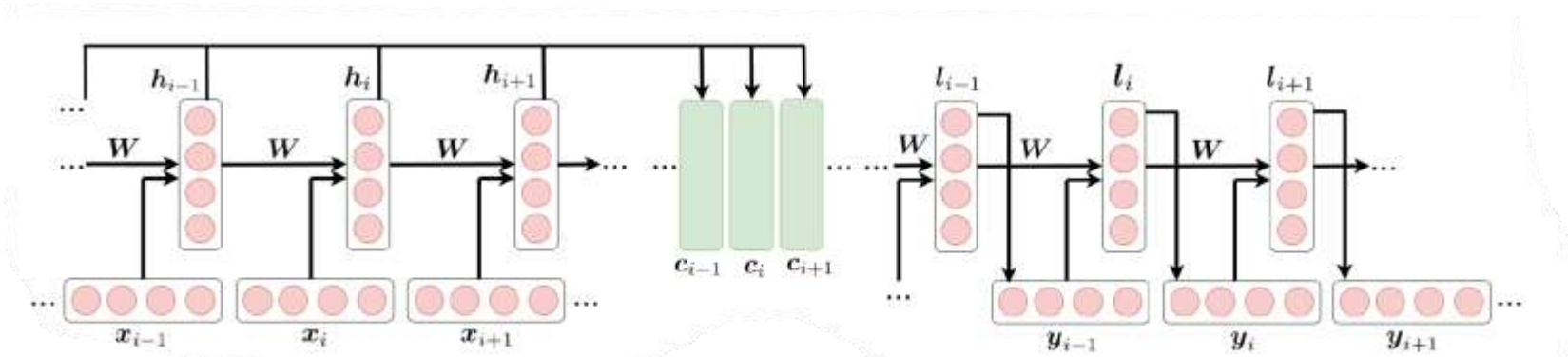
Sequence to Sequence model with attention



- Sequence to sequence with attention:

$$p(y_i|y_1, \ldots, y_{i-1}) = g(y_{i-1}, l_i, c_i)$$

Sequence to Sequence model with attention

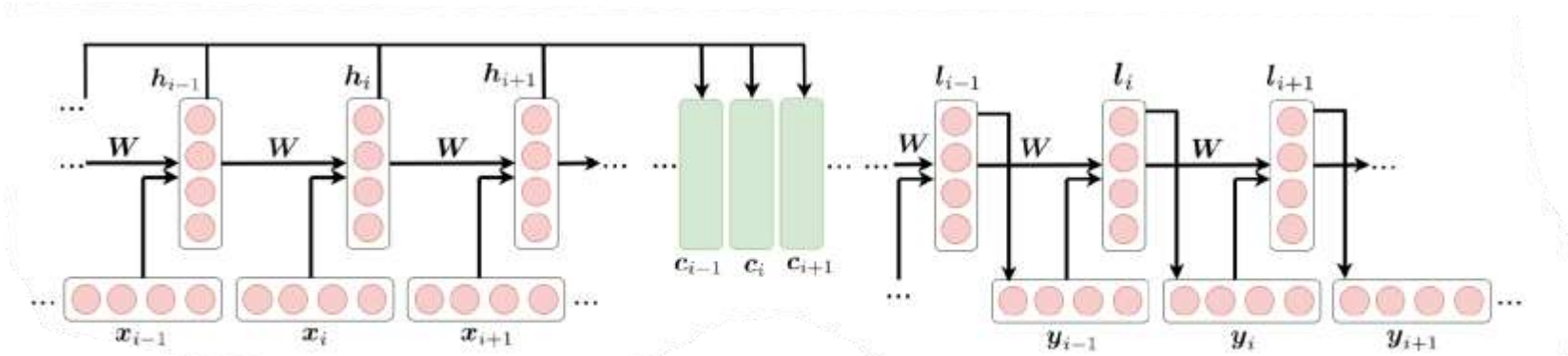


- Sequence to sequence with attention:

$$p(y_i|y_1, \ldots, y_{i-1}) = g(y_{i-1}, l_i, c_i)$$

$$s_{ij} = similarity(l_{i-1}, h_j)$$

Sequence to Sequence model with attention



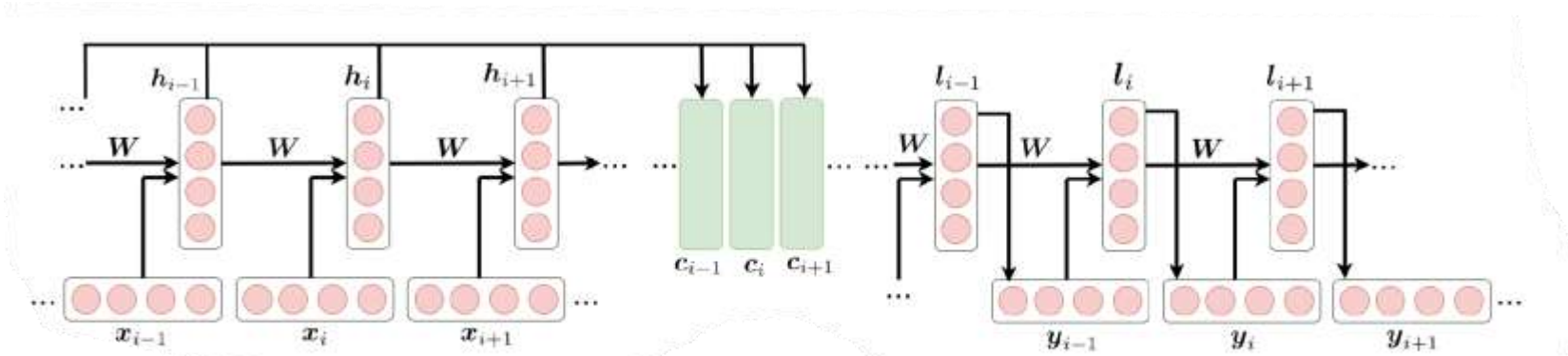
- Sequence to sequence with attention:

$$p(y_i|y_1,\ldots,y_{i-1}) = g(y_{i-1}, l_i, c_i)$$

$$s_{ij} = similarity(l_{i-1}, h_j)$$

$$a_{ij} = \frac{e^{s_{ij}}}{\sum_{k=1} e^{s_{ik}}}$$

Sequence to Sequence model with attention



- Sequence to sequence with attention:

$$p(y_i|y_1,\ldots,y_{i-1}) = g(y_{i-1}, l_i, c_i)$$

$$s_{ij} = similarity(l_{i-1}, h_j)$$

$$a_{ij} = \frac{e^{s_{ij}}}{\sum_{k=1} e^{s_{ik}}}$$

$$c_i = \sum_{j=1} a_{ij} h_j$$

Sequence to Sequence model with attention

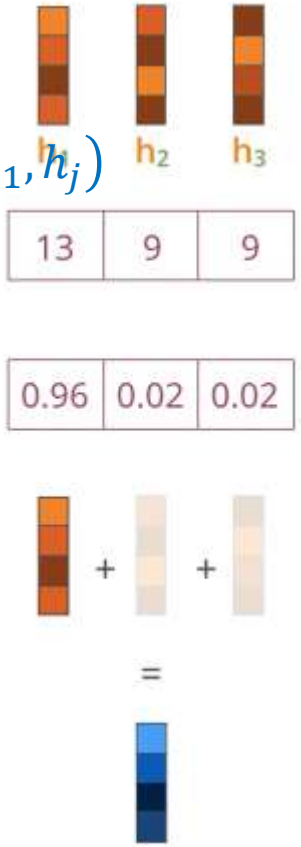
- The effectiveness of the correlation between inputs around position j and the output at position i is crucial.
- This score is determined based on:
  - The RNN hidden state l_{i-1} just before emitting y_i.
  - The j^{th} hidden state h_j of the input sentence.

Attention is weighted average

$$s_{ij} = similarity(l_{i-1}, h_j)$$

$$a_{ij} = \frac{e^{s_{ij}}}{\sum_{k=1} e^{s_{ik}}}$$

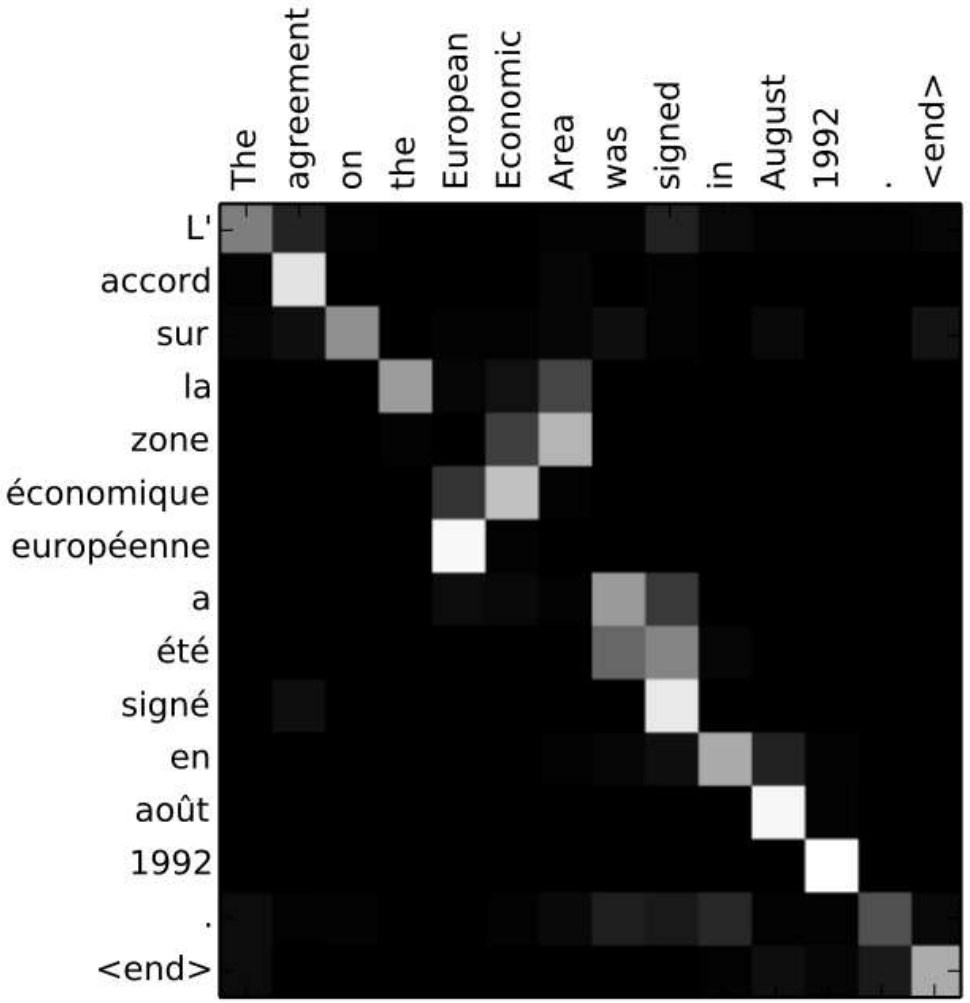
$$c_i = \sum_{j=1} a_{ij} h_j$$



$$p(y_i | y_1, \ldots, y_{i-1}) = g(y_{i-1}, l_i, c_i)$$

Figure: Jay Alammarz

You can see how the model paid attention correctly when outputting "European Economic Area". In French, the order of these words is reversed ("européenne économique zone") as compared to English. Every other word in the sentence is in similar order.



Exact Match

Key	Value
Alice Johnson	555-1234
Bob Smith	555-5678
Carol Davis	555-8765
Dave Martin	555-4321
Eve Clark	555-3456

Example Query 1:

- **Query:** Find the phone number for Bob Smith.
- **Result:** 555-5678

Approximate Match

Database of Box Sizes and Weights

Key (Box Size in cubic inches)	Value (Weight in pounds)
300	30.0
400	40.0

Query: Box Size "360" cubic inches

Calculate Similarity Scores:

- Similarity of size "300": $\frac{1}{|360-300|} = \frac{1}{60}$
- Similarity of size "400": $\frac{1}{|360-400|} = \frac{1}{40}$

Calculate Coefficients:

- $a_1 = \frac{\frac{1}{60}}{\frac{1}{60}+\frac{1}{40}} = \frac{2}{5}$
- $a_2 = \frac{\frac{1}{40}}{\frac{1}{60}+\frac{1}{40}} = \frac{3}{5}$

Calculate Weighted Average Weight:

$$\text{Weighted Average Weight} = (a_1 \cdot 30.0) + (a_2 \cdot 40.0) = 12.0 + 24.0 = 36.0$$

Estimation: The estimated weight of a box with a size of "360" cubic inches is "36.0" pounds.

Self-Attention

- Word embedding

- Composite embeddings (weighted averages)

Self-Attention

- Long Short-Term Memory-Networks for Machine Reading (2016)

By Jianpeng Cheng, Li Dong, Mirella Lapat

Self-Attention

- Long Short-Term Memory-Networks for Machine Reading (2016)

By Jianpeng Cheng, Li Dong, Mirella Lapat

- « *We propose a machine reading simulator which processes text incrementally from left to right and performs shallow reasoning with memory and attention.* »

Self-Attention

- Long Short-Term Memory-Networks for Machine Reading (2016)

By Jianpeng Cheng, Li Dong, Mirella Lapat

- « *We propose a machine reading simulator which processes text incrementally from left to right and performs shallow reasoning with memory and attention.* »

- Offering a way to induce relations among words.

Self-Attention

- Understanding the individual words in a sentence is not sufficient to understand the sentence.
- Need to understand how the words relate to each other.
- The attention mechanism forms composite representations.

Self-Attention

The early bird catches the worm.

- **Composite Concepts:**
- "Early Bird" - Represents promptness or being ahead.
- "Catches the Worm" - Implies a reward for prompt action.

Self-Attention

- **Self-Attention Mechanism:**
- Analyzes pairs of words like "early" + "bird" and "catches" + "worm."
- Forms higher-level composite meanings essential for full phrase understanding.

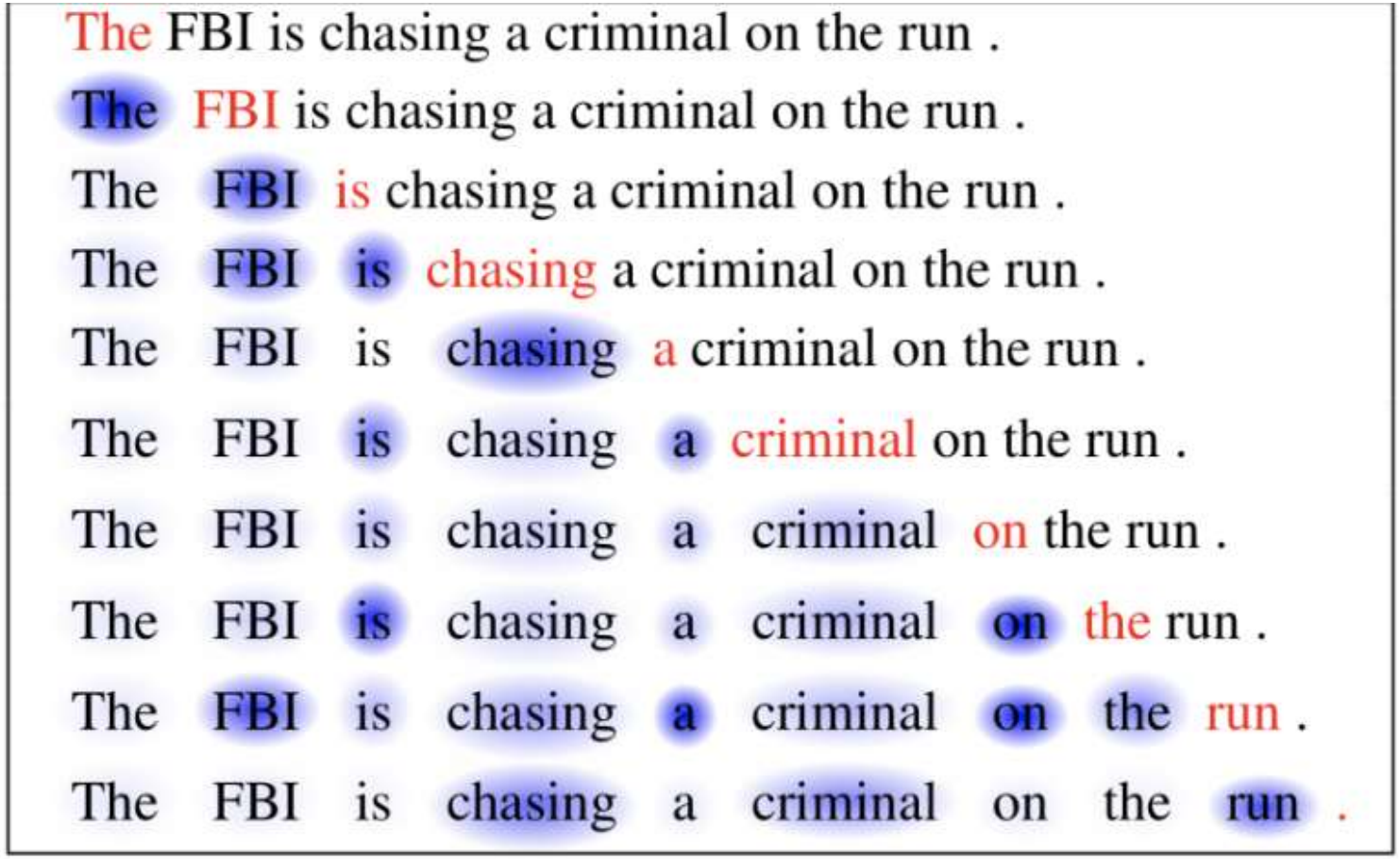
Self-Attention

- **Self-Attention Mechanism:**
- Analyzes pairs of words like "early" + "bird" and "catches" + "worm."
- Forms higher-level composite meanings essential for full phrase understanding.
- **Process:**
- Attention scores calculated between word pairs.
- Multiple layers help form complex concept representations.

If you're a bird, be an early bird and catch the worm for your breakfast plate. If you're a bird, be an early early bird-- But if you're a worm, sleep late.

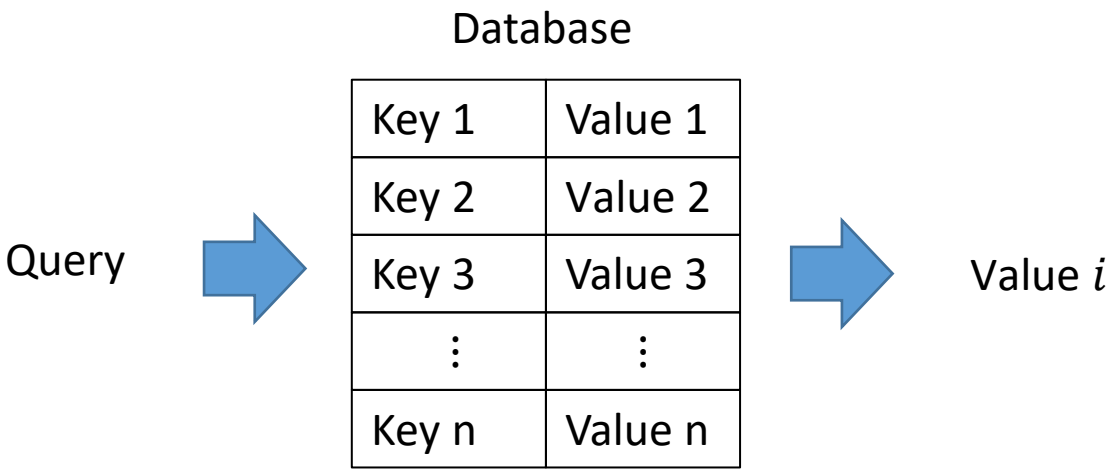
If you're a bird, be an early bird and catch the worm for your breakfast plate. If you're a bird, be an early early bird-- But if you're a worm, sleep late.

Shel Silverstein


The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .

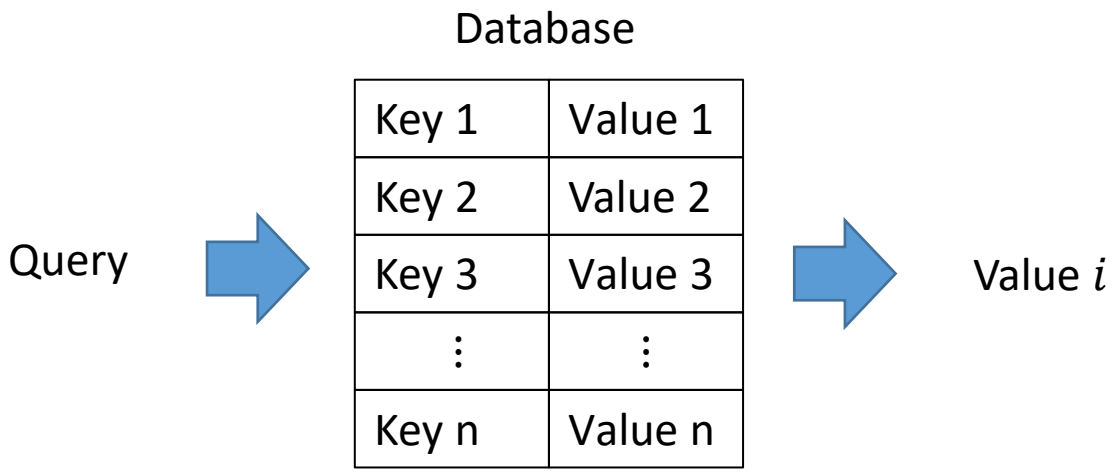
Generalized definition

- Mimics the retrieval of a **value** v_i for a **query** q based on **key** k_i in database.



Generalized definition

- Mimics the retrieval of a **value** v_i for a **query** q based on **key** k_i in database.



$$attention(q, \boldsymbol{k}, \boldsymbol{v}) = \sum_i similarity(q, k_i) \times v_i$$

Generalized definition

- To calculate the Attention of a target word with respect to the input word
- Use the **Query** of the target and the **Key** of the input
- Calculate a matching score
- These matching scores act as the weights of the **Value** vectors.

Query	Key	Value
The	The	The
early	early	early
bird	bird	bird
catches	catches	catches
the	the	the
worm	worm	worm

Query	Key	Value
x_{The}	x_{The}	x_{The}
x_{early}	x_{early}	x_{early}
x_{bird}	x_{bird}	x_{bird}
x_{catches}	x_{catches}	x_{catches}
x_{the}	x_{the}	x_{the}
x_{worm}	x_{worm}	x_{worm}

Query	Key	Value
$W_Q^T x_{\text{The}}$	$W_K^T x_{\text{The}}$	$W_V^T x_{\text{The}}$
$W_Q^T x_{\text{early}}$	$W_K^T x_{\text{early}}$	$W_V^T x_{\text{early}}$
$W_Q^T x_{\text{bird}}$	$W_K^T x_{\text{bird}}$	$W_V^T x_{\text{bird}}$
$W_Q^T x_{\text{catches}}$	$W_K^T x_{\text{catches}}$	$W_V^T x_{\text{catches}}$
$W_Q^T x_{\text{the}}$	$W_K^T x_{\text{the}}$	$W_V^T x_{\text{the}}$
$W_Q^T x_{\text{worm}}$	$W_K^T x_{\text{worm}}$	$W_V^T x_{\text{worm}}$

Calculating Attention for the Word "bird"

Objective:

- Calculate the attention weights for the word "**bird**" within the sentence context.

Procedure:

- Focus on the word "**bird**" as our point of interest.

Attention Calculation:

- Calculate a similarity between the query vector of "bird" and the key vectors of other words in the sentence.

"The", "early", "bird", "catches", "the", "worm".

Generalized definition (example)

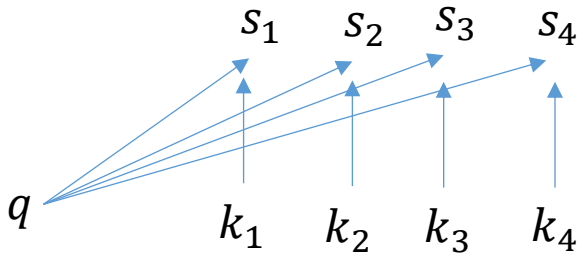
$$\alpha_1 = \text{softmax}\left(\langle W_Q^T x_{\text{bird}}, W_K^T x_{\text{The}} \rangle\right)$$
$$\alpha_2 = \text{softmax}\left(\langle W_Q^T x_{\text{bird}}, W_K^T x_{\text{early}} \rangle\right)$$
$$\alpha_3 = \text{softmax}\left(\langle W_Q^T x_{\text{bird}}, W_K^T x_{\text{catches}} \rangle\right)$$
$$\alpha_4 = \text{softmax}\left(\langle W_Q^T x_{\text{bird}}, W_K^T x_{\text{the}} \rangle\right)$$
$$\alpha_5 = \text{softmax}\left(\langle W_Q^T x_{\text{bird}}, W_K^T x_{\text{worm}} \rangle\right)$$

Generalized definition (example)

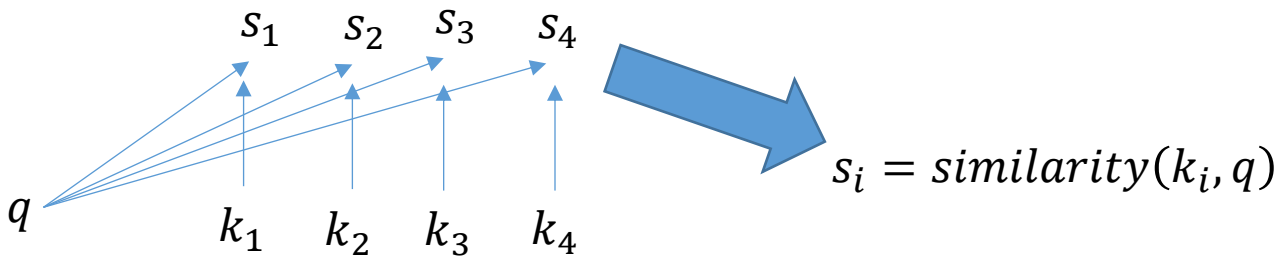
$$\alpha_1 = \text{softmax}\left(\langle W_Q^T x_{\text{bird}}, W_K^T x_{\text{The}} \rangle\right)$$
$$\alpha_2 = \text{softmax}\left(\langle W_Q^T x_{\text{bird}}, W_K^T x_{\text{early}} \rangle\right)$$
$$\alpha_3 = \text{softmax}\left(\langle W_Q^T x_{\text{bird}}, W_K^T x_{\text{catches}} \rangle\right)$$
$$\alpha_4 = \text{softmax}\left(\langle W_Q^T x_{\text{bird}}, W_K^T x_{\text{the}} \rangle\right)$$
$$\alpha_5 = \text{softmax}\left(\langle W_Q^T x_{\text{bird}}, W_K^T x_{\text{worm}} \rangle\right)$$

$$v_{\text{bird}} = \alpha_1 V_{\text{The}} + \alpha_2 V_{\text{early}} + \alpha_3 V_{\text{catches}} + \cdots$$

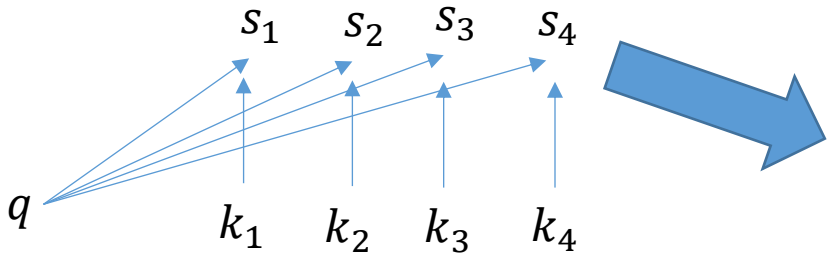
Neural architecture



Neural architecture



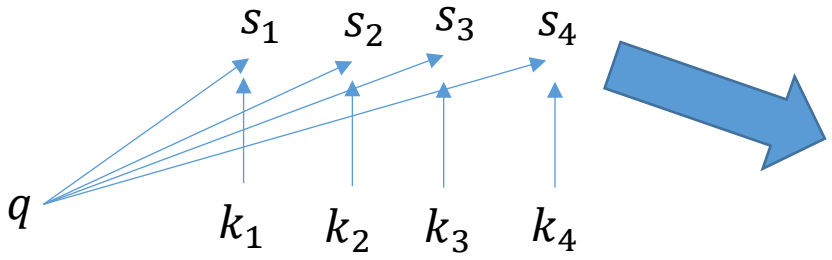
Neural architecture



$$s_i = similarity(k_i, q) = \begin{cases} q^T k_i \\ q^T k_i / \sqrt{p} \\ q^T W k_i \\ \vdots \end{cases}$$

(p is the dimensionality of k)

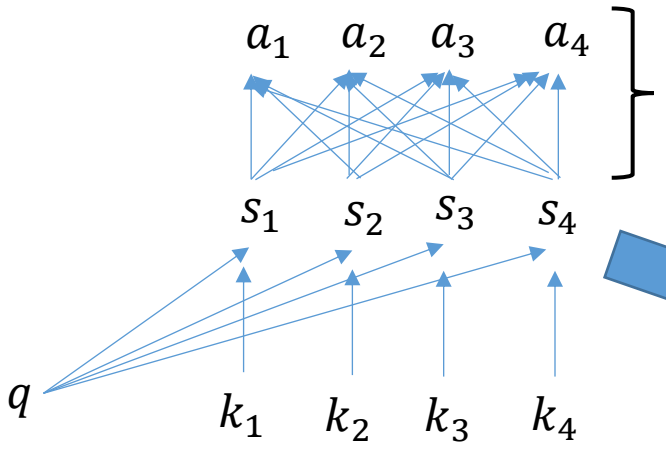
Neural architecture



$$s_i = similarity(k_i, q) = \begin{cases} q^T k_i \\ q^T k_i / \sqrt{p} \\ q^T W k_i \\ \vdots \end{cases}$$

(p is the dimensionality of k)

Neural architecture

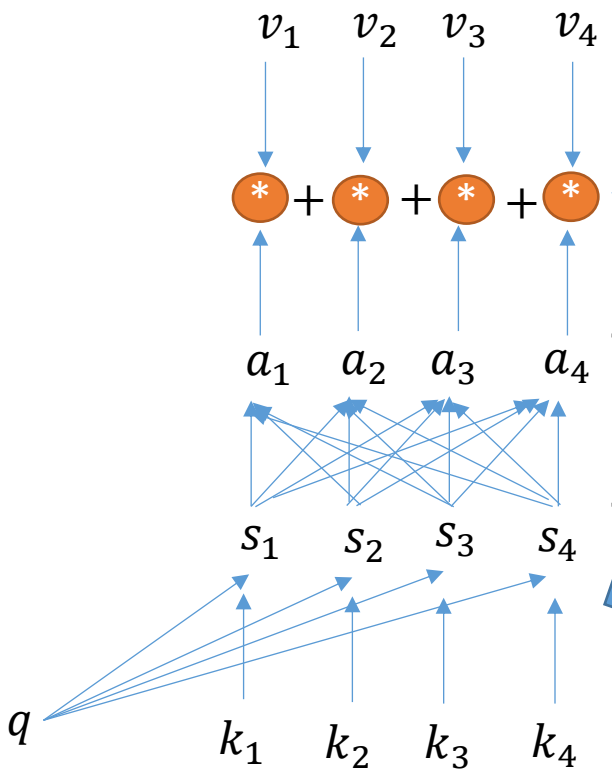


$$softmax:\ a_i = \frac{e^{s_i}}{\sum_j e^{s_j}}$$

$$s_i = similarity(k_i, q) = \begin{cases} q^T k_i \\ q^T k_i / \sqrt{p} \\ q^T W k_i \\ \vdots \end{cases}$$

(p is the dimensionality of k)

Neural architecture



$$attention\ value = \sum_i a_i v_i$$

$$softmax\colon\ a_i = \frac{e^{s_i}}{\sum_j e^{s_j}}$$

$$s_i = similarity(k_i, q) = \begin{cases} q^T k_i \\ q^T k_i / \sqrt{p} \\ q^T W k_i \\ \vdots \end{cases}$$

(p is the dimensionality of k)

Matrix Forms:

- Words in sequence: $X = [x_1, \ldots, x_n] \in \mathbb{R}^{d \times n}$
- Queries: $Q = [q_1, \ldots, q_n] \in \mathbb{R}^{p \times n}$
- Keys: $K = [k_1, \ldots, k_n] \in \mathbb{R}^{p \times n}$
- Values: $V = [v_1, \ldots, v_n] \in \mathbb{R}^{r \times n}$

Projection into Subspaces (Post Definitions):

- Queries: $q_i = W_Q^T x_i$
- Keys: $k_i = W_K^T x_i$
- Values: $v_i = W_V^T x_i$

Matrix Form

Similarity Measures:

- Inner product: $x_i^T W_Q (W_K)^T x_i$
- Acts like a kernel matrix, measuring similarity.

Attention Computation:

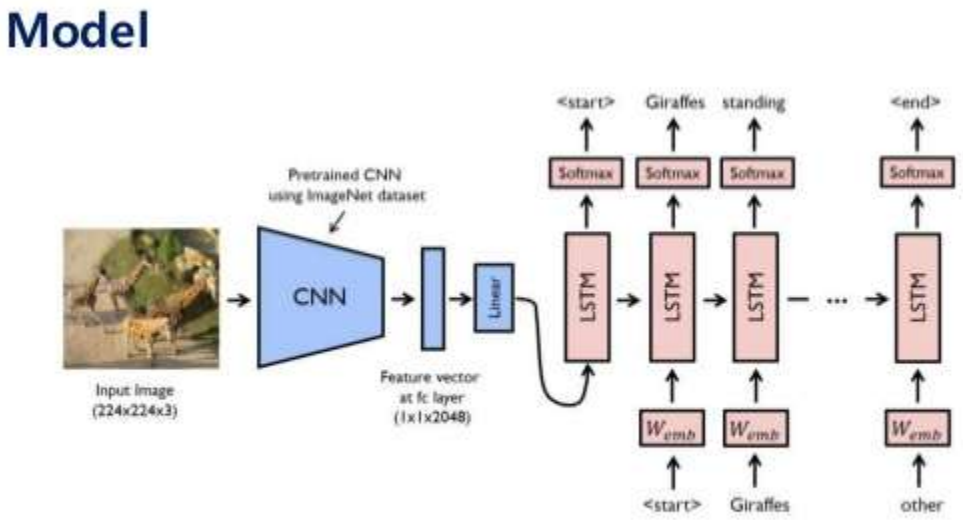
- $Z := \text{attention}(Q, K, V) = V \text{softmax}\left(\frac{1}{\sqrt{p}} Q^T K\right)$

Attention in computer vision

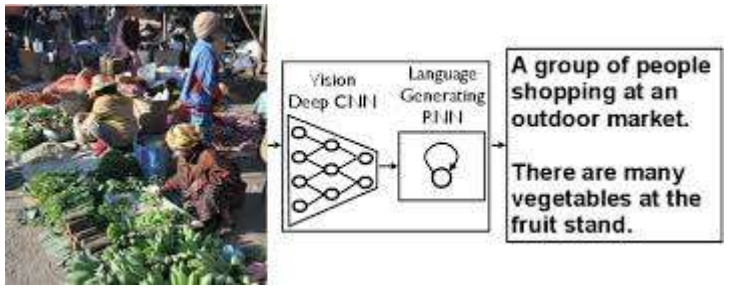
- **Image Captioning**

- Show, Attend and Tell: Neural Image Caption Generation with Visual Attention (Kelvin Xu, et al 2016)

Attention in computer vision



Attention in computer vision



Annotation vector

Spatial information of an image is captured by lower convolutional layer of a CNN.

$V = \{v_1, \ldots, v_l\}, v_i \in R^d$



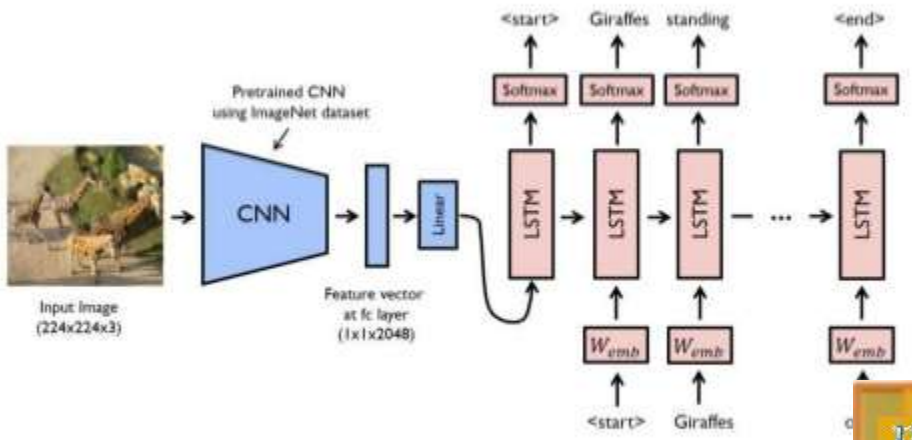
Attention in computer vision

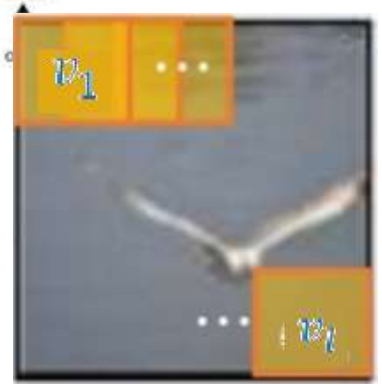
$$v = \{v_1, \ldots, v_l\}, v_i \in R^d$$

$$s_{ti} = f_{att}(v_i, h_{t-1})$$

$$a_{ti} = \frac{e^{s_{ti}}}{\sum_j e^{s_{tj}}}$$

$$z_t = \Phi(\{v_i\}, \{a_i\})$$





Attention in computer vision



