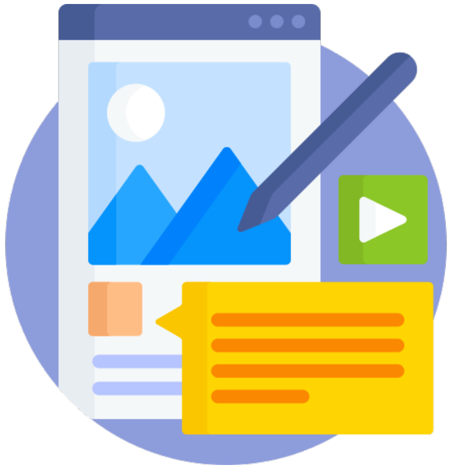
Image Captioning

Models Introduction

Date: Oct 27, 2021

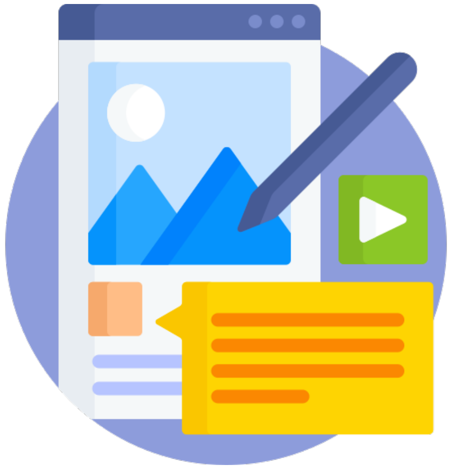
Presenter: Everley Tseng



Outline

- Show Attend and Tell
  - Introduction to attention
  - NLP backgrounds
  - Model structure
  - Soft attention & hard attention
  - Loss function
  - Experimental results

- What's new in Image Captioning?



Why Attention?

Please write the caption for this image

Image source: https://www.sandiegouniontribune.com/business/real-estate/story/2020-08-21/will-san-diego-stay-at-home-workers-leave-if-given-the-opportunity

Why Attention?

- Important Components
  - Objects
  - Environments
- Relationships
  - Spatial
  - Interactive
- Details
  - Adjectives
  - Adverbs

Why Attention?

- Important Components
  - Objects
  - Environments
- Relationships
  - Spatial
  - Interactive
- Details
  - Adjectives
  - Adverbs

Why Attention?

- Important Components
  - Objects **Dog** **Dog**
  - Environments
- Relationships
  - Spatial
  - Interactive
- Details
  - Adjectives
  - Adverbs

Why Attention?

- Important Components
  - Objects **Dog Dog**
  - Environments
- Relationships
  - Spatial
  - Interactive
- Details
  - Adjectives **Brown**
    Colorful Collar
  - Adverbs **Happy**

Why Attention?

- Important Components
  - Objects **Dog Dog**
  - Environments
- Relationships **Next to... Looking at... Walking toward...**
  - Spatial
  - Interactive
- Details
  - Adjectives **Brown Colorful Collar**
  - Adverbs **Happy**

Why Attention?

- Important Components
  - Objects **Dog Dog**
  - Environments
- Relationships **Next to… Looking at… Walking toward…**
  - Spatial
  - Interactive
- Details
  - Adjectives **Brown Colorful Collar**
  - Adverbs **Happy**

As we write down words, our attention moves across the image.

Show and Tell

O. Vinyals, A. Toshev, S. Bengio and D. Erhan, "Show and tell: A neural image caption generator," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015

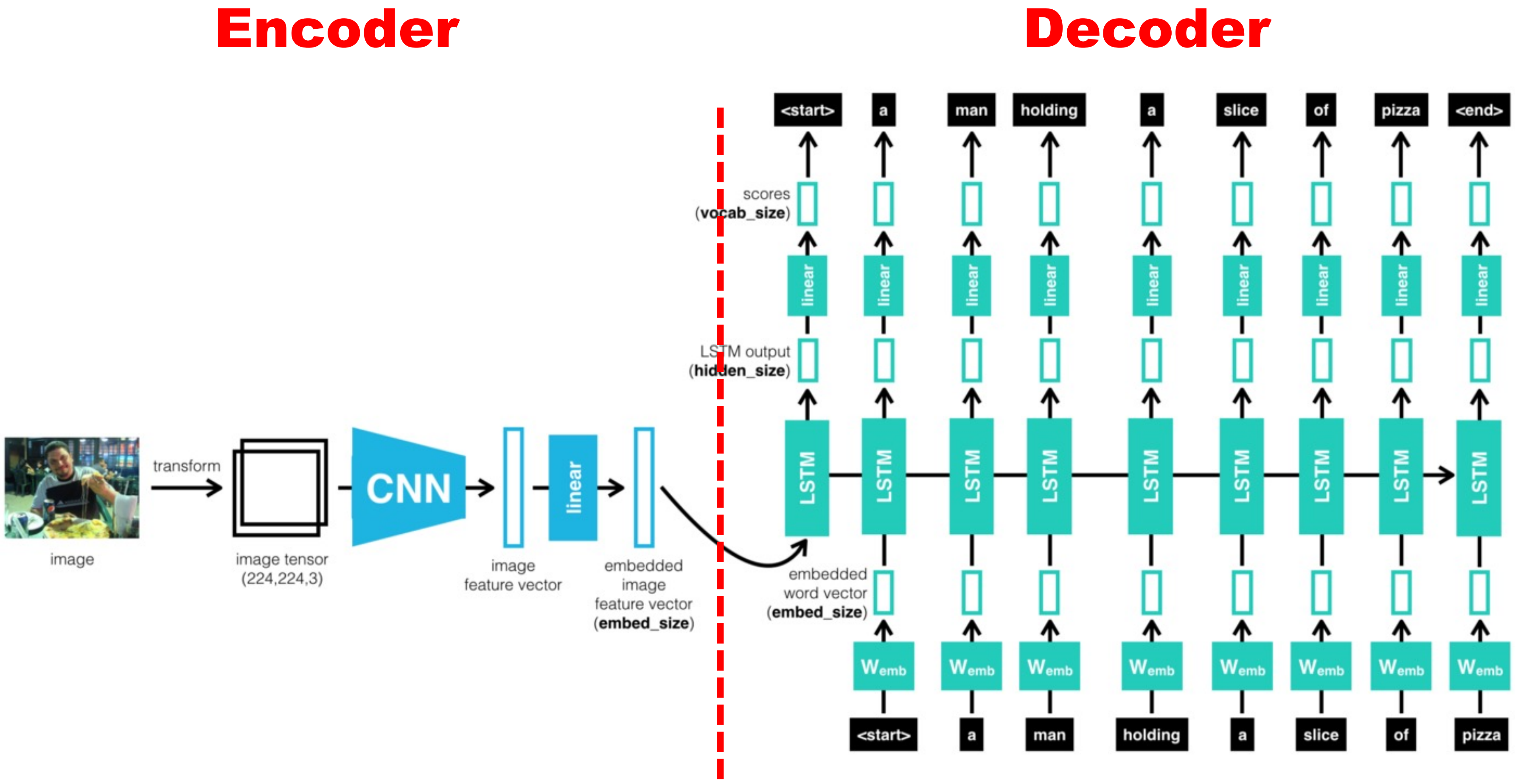
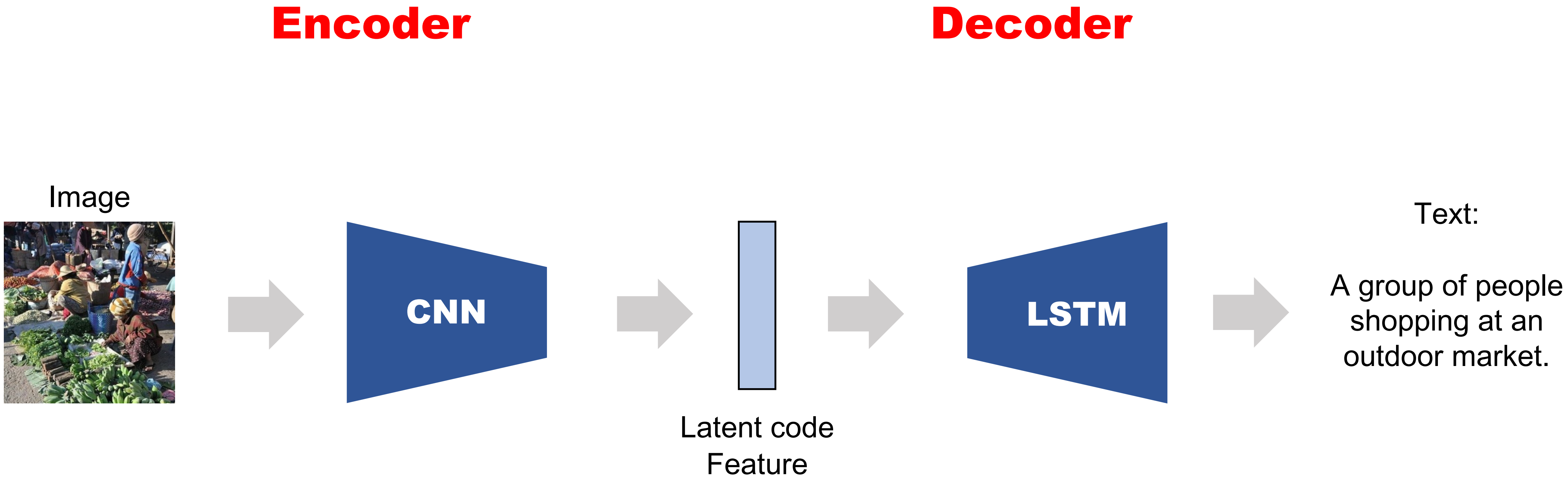


Image source: https://medium.com/swlh/image-caption-generation-with-visual-attention-c782dfc0634b

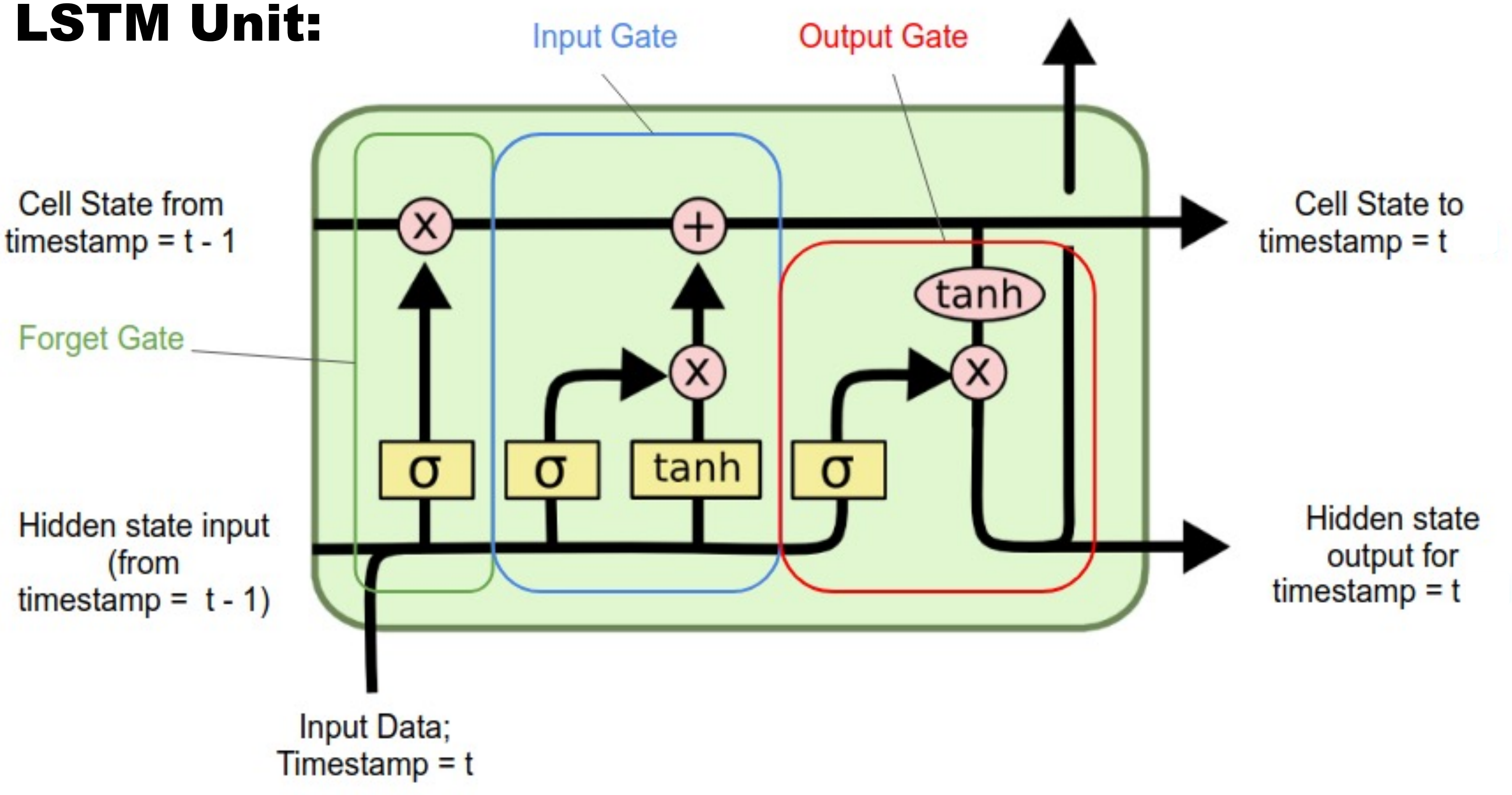
Show and Tell

O. Vinyals, A. Toshev, S. Bengio and D. Erhan, "Show and tell: A neural image caption generator," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015



Recall – LSTM

LSTM Unit:



LSTM – Encoder-Decoder in NLP

E.g., language translation

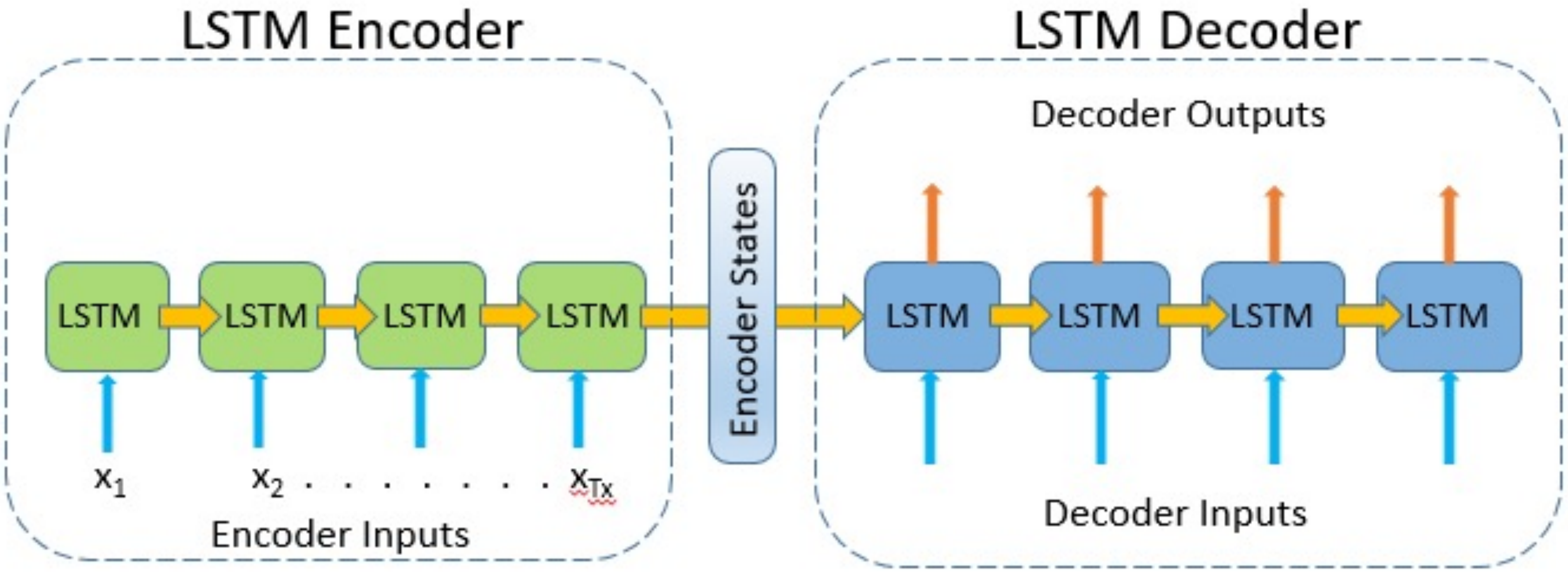


Image source: https://towardsdatascience.com/time-series-forecasting-with-deep-stacked-unidirectional-and-bidirectional-lstms-de7c099bd918

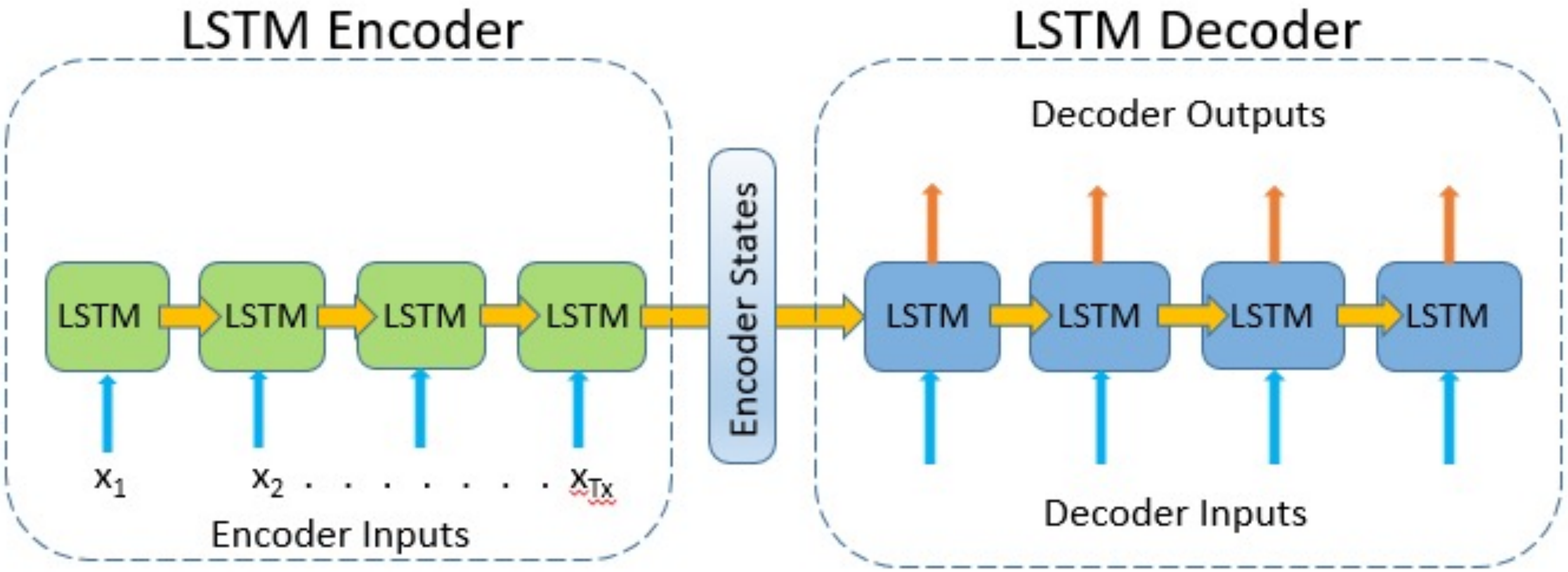
LSTM – Encoder-Decoder in NLP

How do we actually input/output words?

E.g., language translation

Language B

word1 word2 word3 wordn



word1 word2 word3 wordn

Language A

Image source: https://towardsdatascience.com/time-series-forecasting-with-deep-stacked-unidirectional-and-bidirectional-lstms-de7c099bd918

NLP – Word Representation

- One-hot encoding
- Word embedding

```
                    Paris
              Rome                         word V
Rome   = [1, 0, 0, 0, 0, 0, …, 0]
Paris  = [0, 1, 0, 0, 0, 0, …, 0]
Italy  = [0, 0, 1, 0, 0, 0, …, 0]
France = [0, 0, 0, 1, 0, 0, …, 0]
```

NLP – Word Representation

- One-hot encoding
- Word embedding

$$\begin{array}{lccccccccc}
 & & \text{Rome} & \text{Paris} & & & & & & \text{word } V \\
\text{Rome} & = & [1, & 0, & 0, & 0, & 0, & 0, & \ldots, & 0] \\
\text{Paris} & = & [0, & 1, & 0, & 0, & 0, & 0, & \ldots, & 0] \\
\text{Italy} & = & [0, & 0, & 1, & 0, & 0, & 0, & \ldots, & 0] \\
\text{France} & = & [0, & 0, & 0, & 1, & 0, & 0, & \ldots, & 0]
\end{array}$$

My Corpus:

1. I live in Rome
2. I live in Paris
3. I live in Italy
4. I live in France

What is the length of each word vector?

NLP – Word Representation

- One-hot encoding
- Word embedding

Rome Paris word V

```
Rome   = [1, 0, 0, 0, 0, 0, ..., 0]
Paris  = [0, 1, 0, 0, 0, 0, ..., 0]
Italy  = [0, 0, 1, 0, 0, 0, ..., 0]
France = [0, 0, 0, 1, 0, 0, ..., 0]
```

My Corpus:

1. I live in Rome
2. I live in Paris
3. I live in Italy
4. I live in France

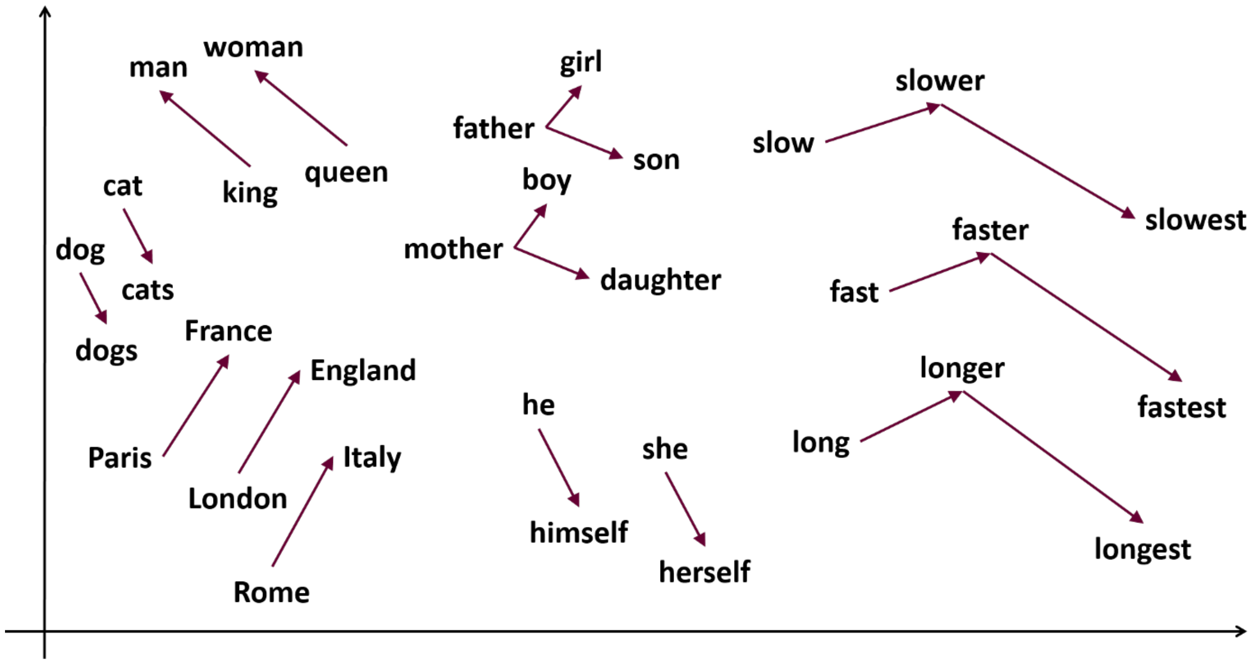
What is the length of each word vector? **Length = 7**

Dictionary: [‘I’, ‘live’, ‘in’, ‘Rome’, ‘Paris’, ‘Italy’, ‘France’]

Image source:
https://www.usna.edu/Users/cs/nchamber/courses/nlp/f20/labs/lab5/index.html

NLP – Word Representation

- One-hot encoding
- Word embedding
  - word2vec
  - BERT
  - GLoVe



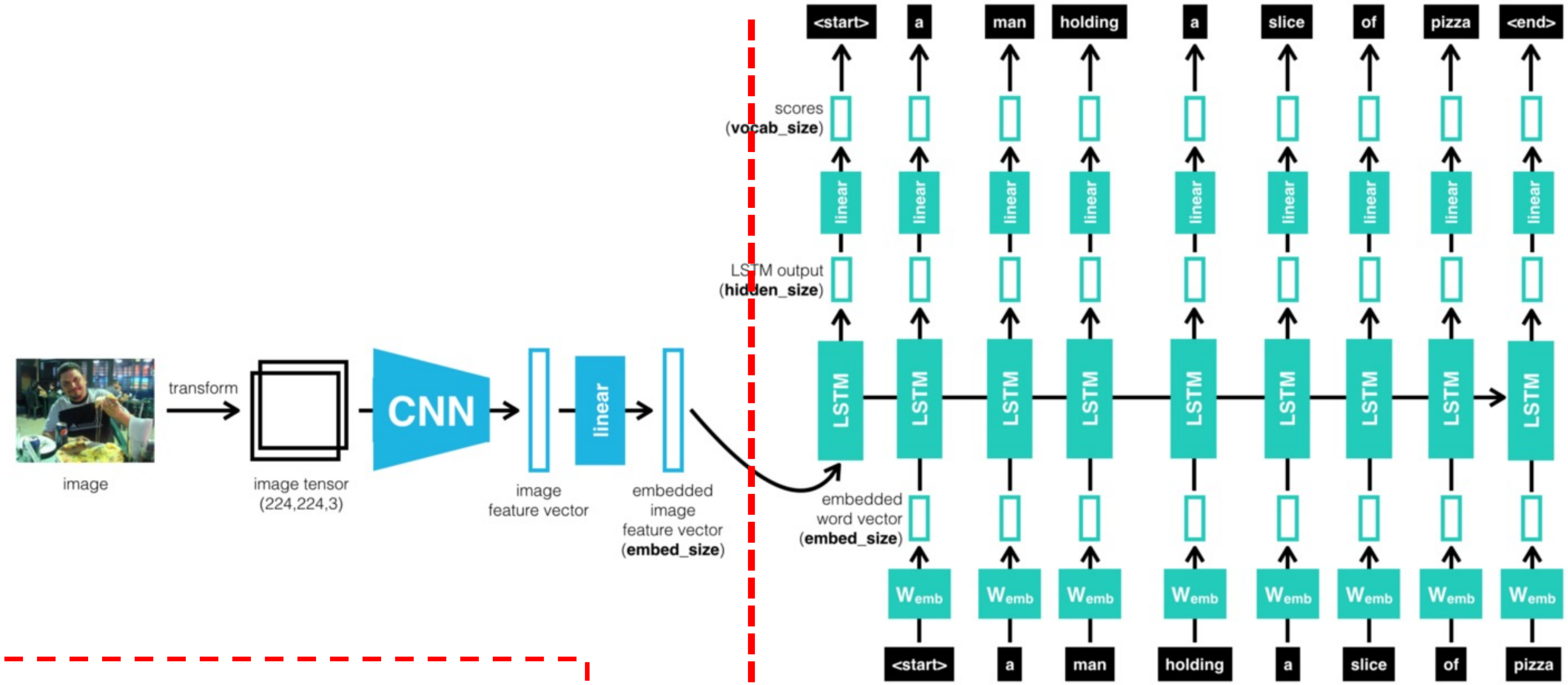
The semantic meaning of the words are embedded in the latent space.

Show and Tell

O. Vinyals, A. Toshev, S. Bengio and D. Erhan, "Show and tell: A neural image caption generator," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015

Encoder

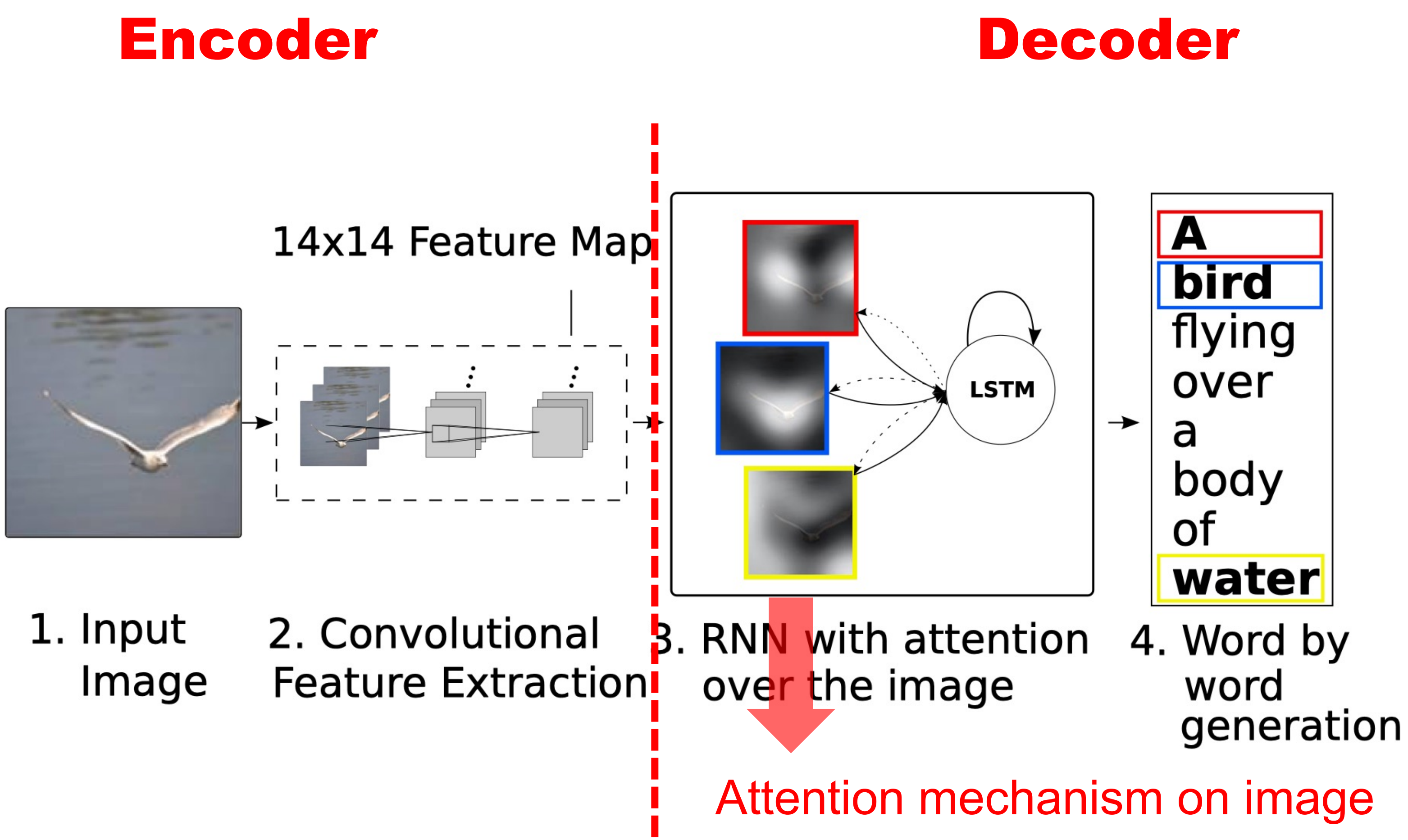
Decoder



What are the issues of this model?

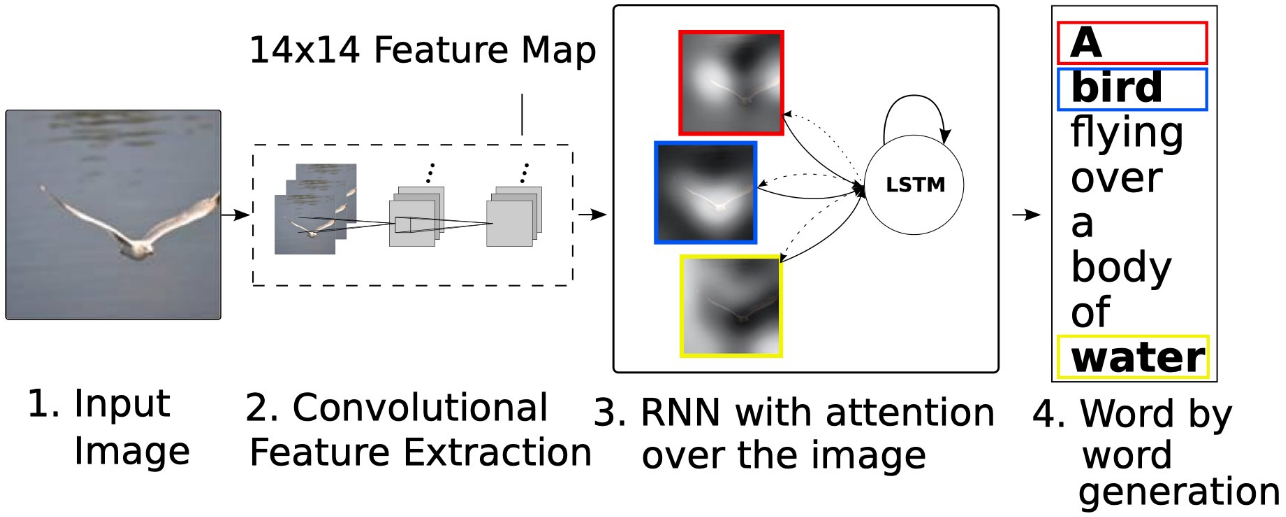
Image source: https://medium.com/swlh/image-caption-generation-with-visual-attention-c782dfc0634b

Show, Attend and Tell

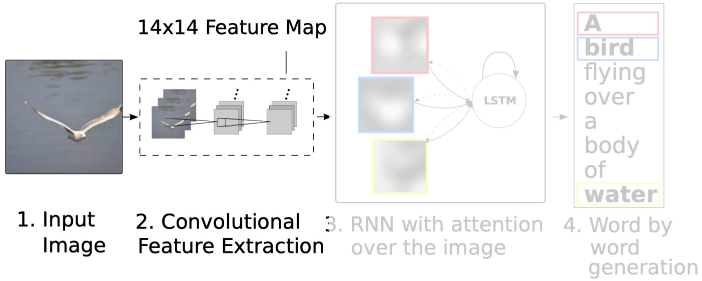


Show Attend and Tell

Model Structure



CNN Encoder



VGG-16

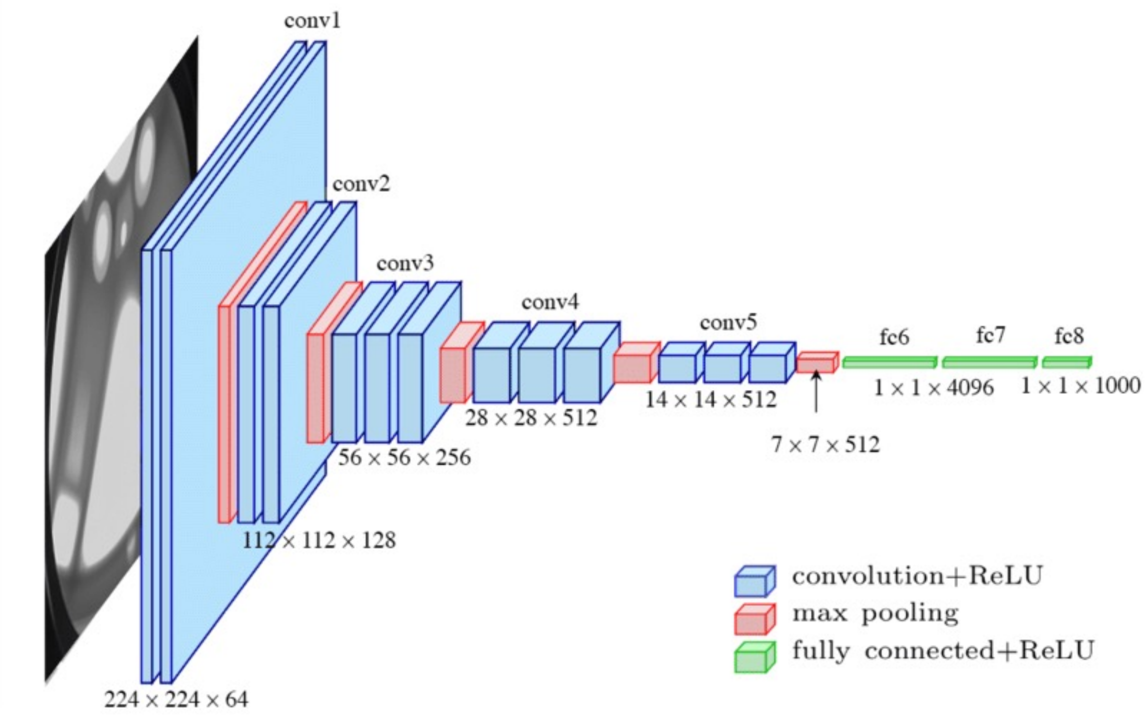
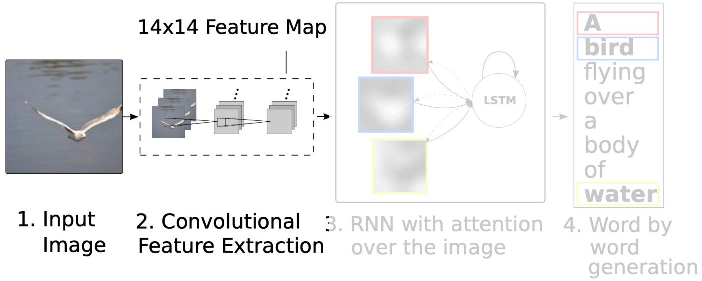
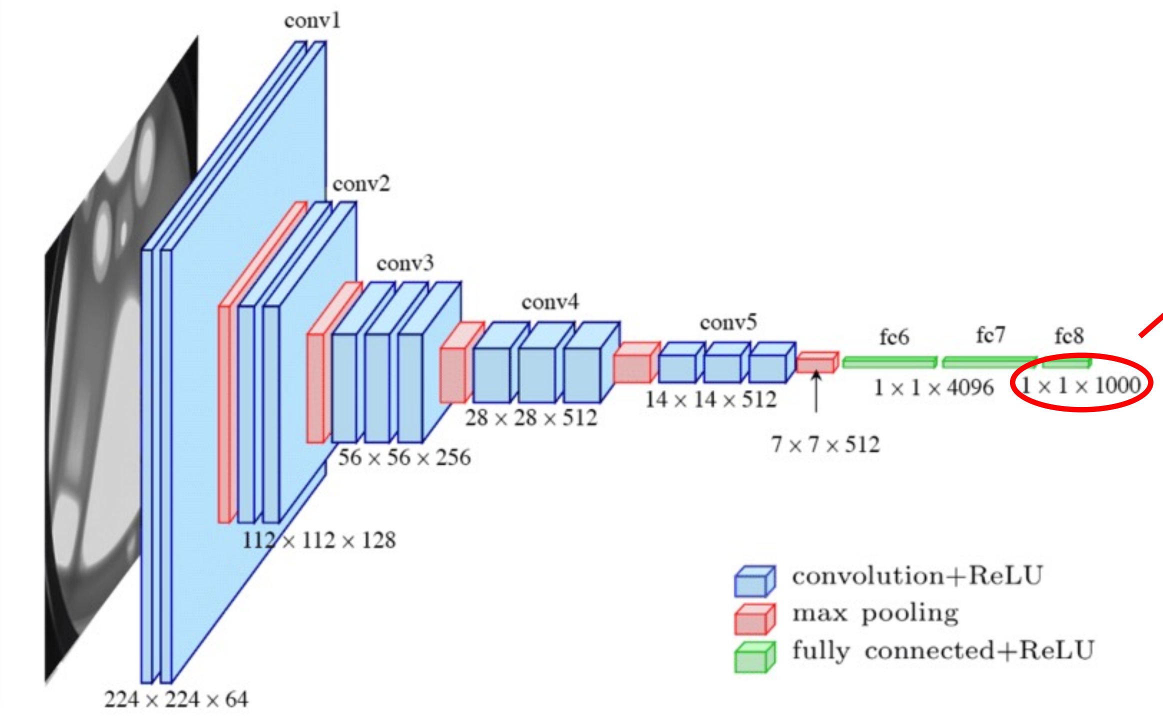


Image source: https://www.researchgate.net/figure/Fig-A1-The-standard-VGG-16-network-architecture-as-proposed-in-32-Note-that-only_fig3_322512435

CNN Encoder



VGG-16



Show and Tell

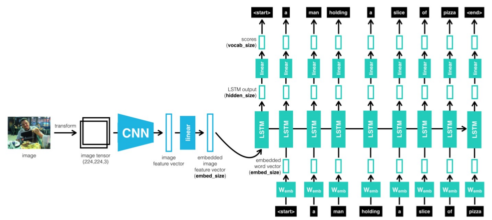
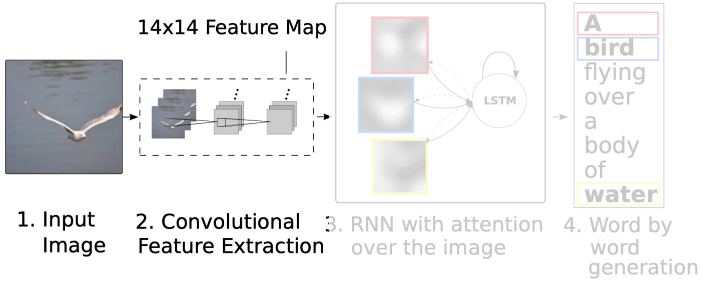
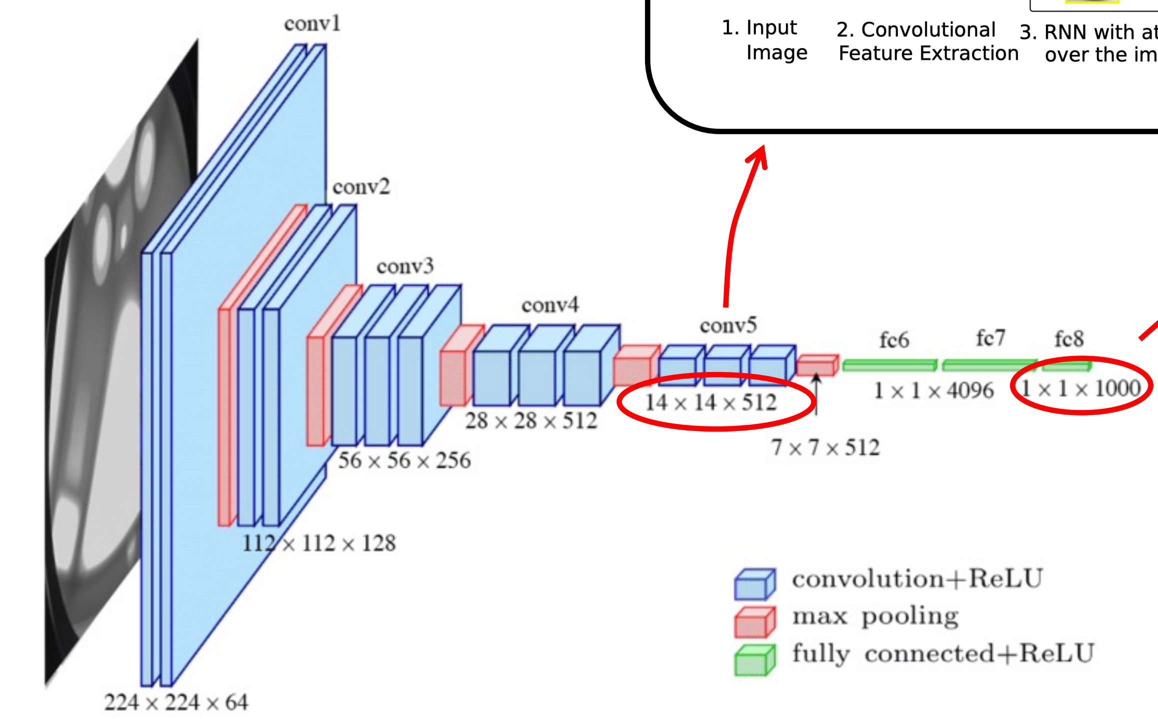


Image source: https://www.researchgate.net/figure/Fig-A1-The-standard-VGG-16-network-architecture-as-proposed-in-32-Note-that-only_fig3_322512435

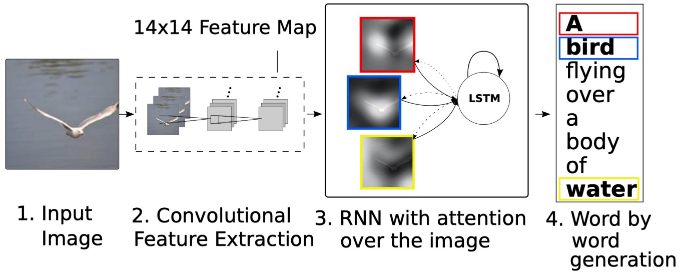


CNN Encoder

VGG-16



Show Attend and Tell



Spatial information is preserved for spatial attention

Show and Tell

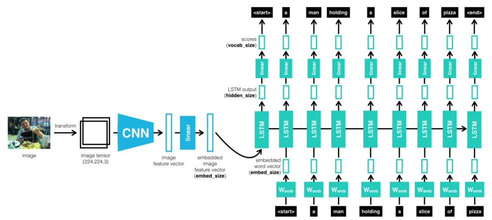
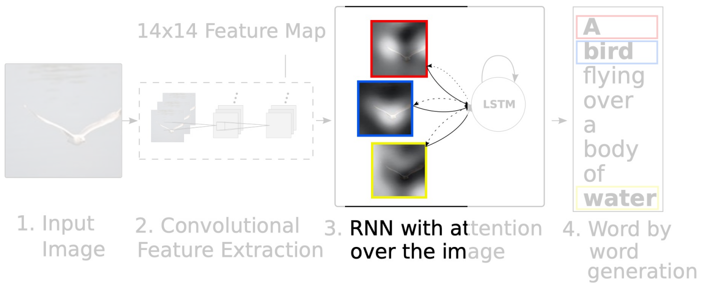
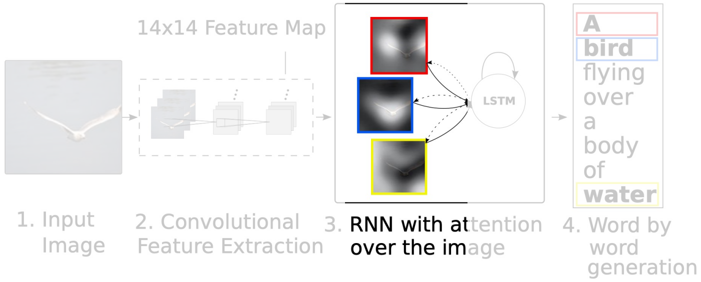


Image source: https://www.researchgate.net/figure/Fig-A1-The-standard-VGG-16-network-architecture-as-proposed-in-32-Note-that-only_fig3_322512435

Attention

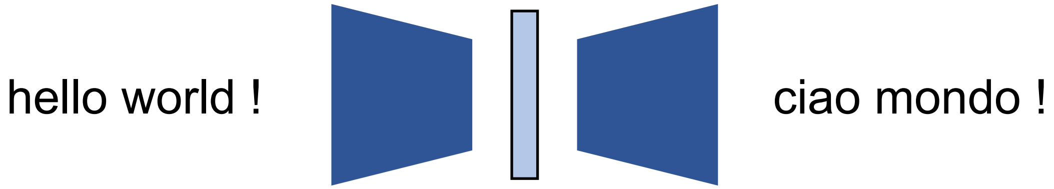


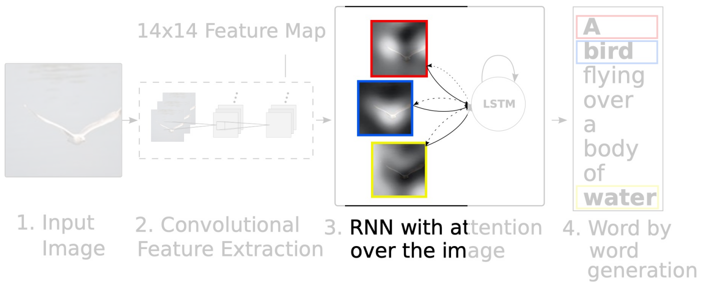
- Soft attention
- Hard attention



Soft Attention

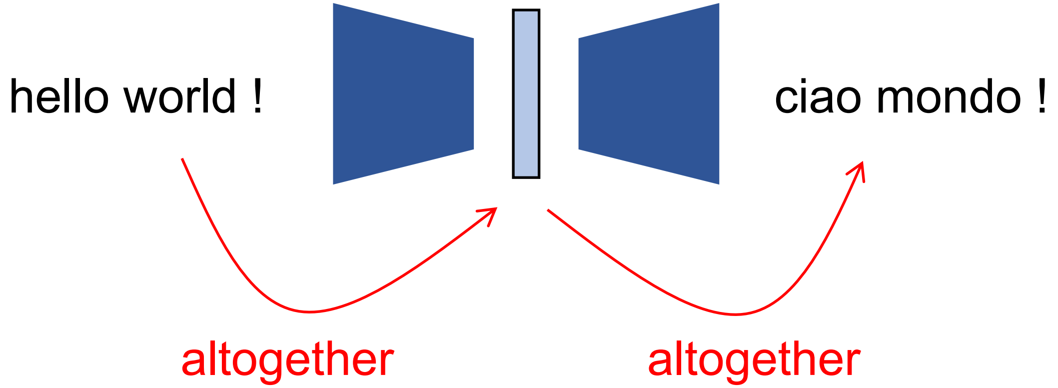
- Attention Is All You Need, 2017
  - Natural Language Processing (NLP)
  - Language translation



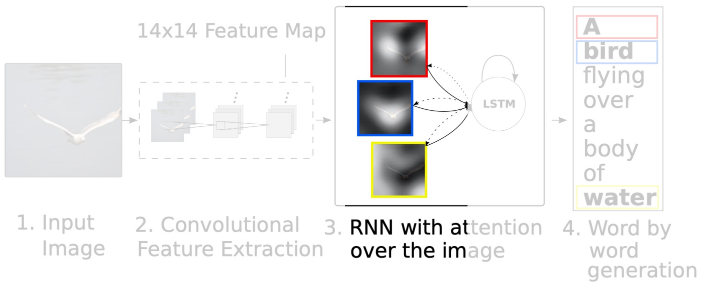


Soft Attention

- Attention Is All You Need, 2017
  - Natural Language Processing (NLP)
  - Language translation

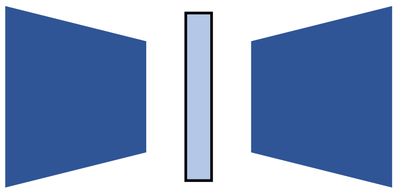


Soft Attention



- Attention Is All You Need, 2017
  - Natural Language Processing (NLP)
  - Language translation

hello world !



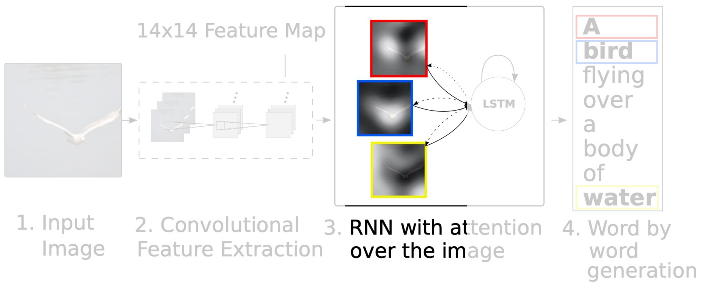
ciao mondo !

hello world !

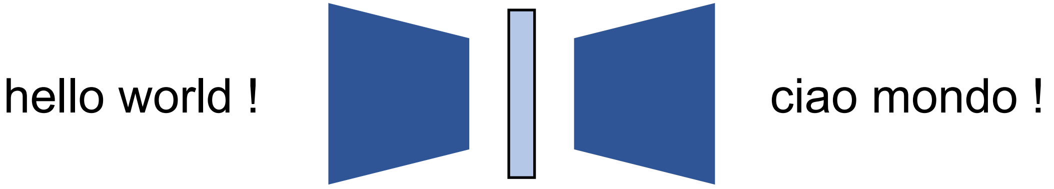
ciao mondo !

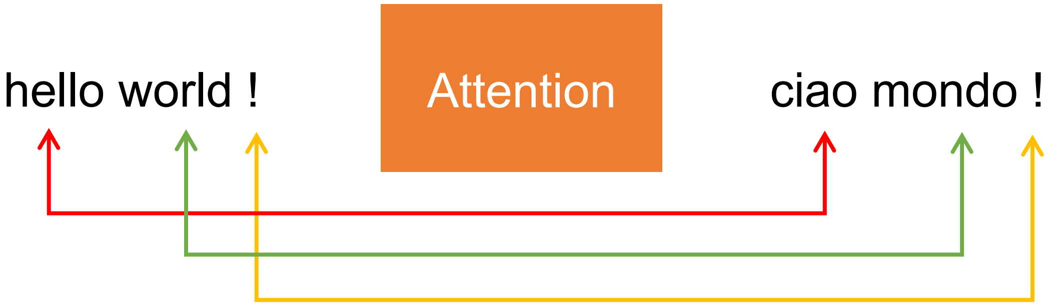
Should the contribution be equal?
Which word do you think should contribute to “ciao” more?

Soft Attention



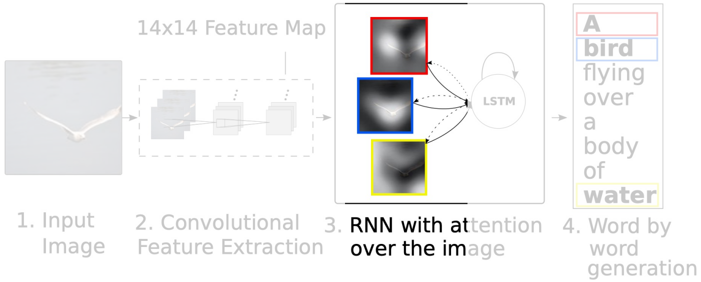
- Attention Is All You Need, 2017
  - Natural Language Processing (NLP)
  - Language translation





Attention helps the model to understand these relationships

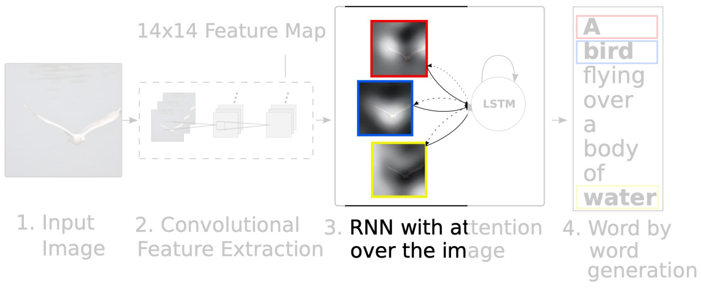
Soft Attention



- Attention Is All You Need, 2017
  - Natural Language Processing (NLP)
  - Language translation

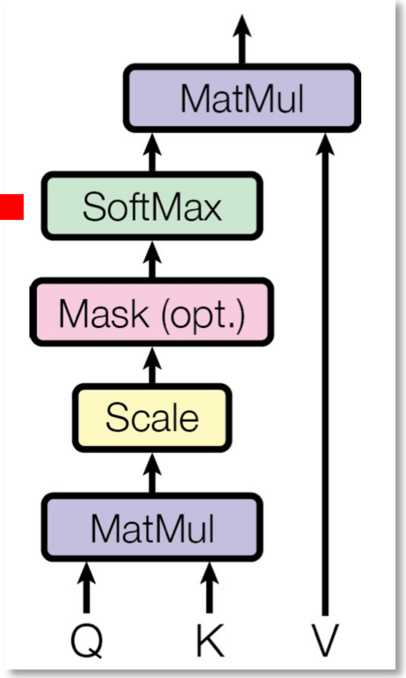
Attention mechanism:
How to calculate the influence on ***ciao*** from ***hello***?

Soft Attention



- Transformer: Attention Is All You Need, 2017
  - Natural Language Processing (NLP)
  - Language translation

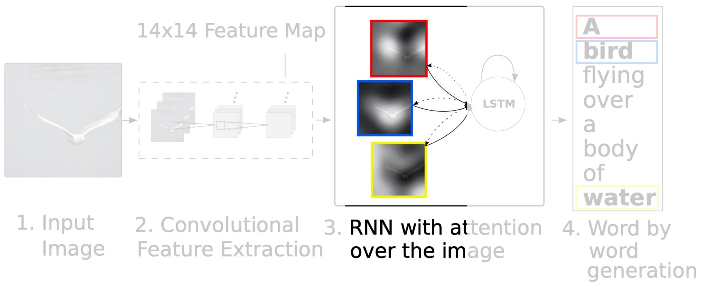
Attention values



Attention mechanism:
How to calculate the influence on ***ciao*** from ***hello***?

Query: information about ***hello*** as a feature candidate
Key: information about ***ciao*** as the output
Value: original feature vector of ***hello***

Soft Attention



- Image attention

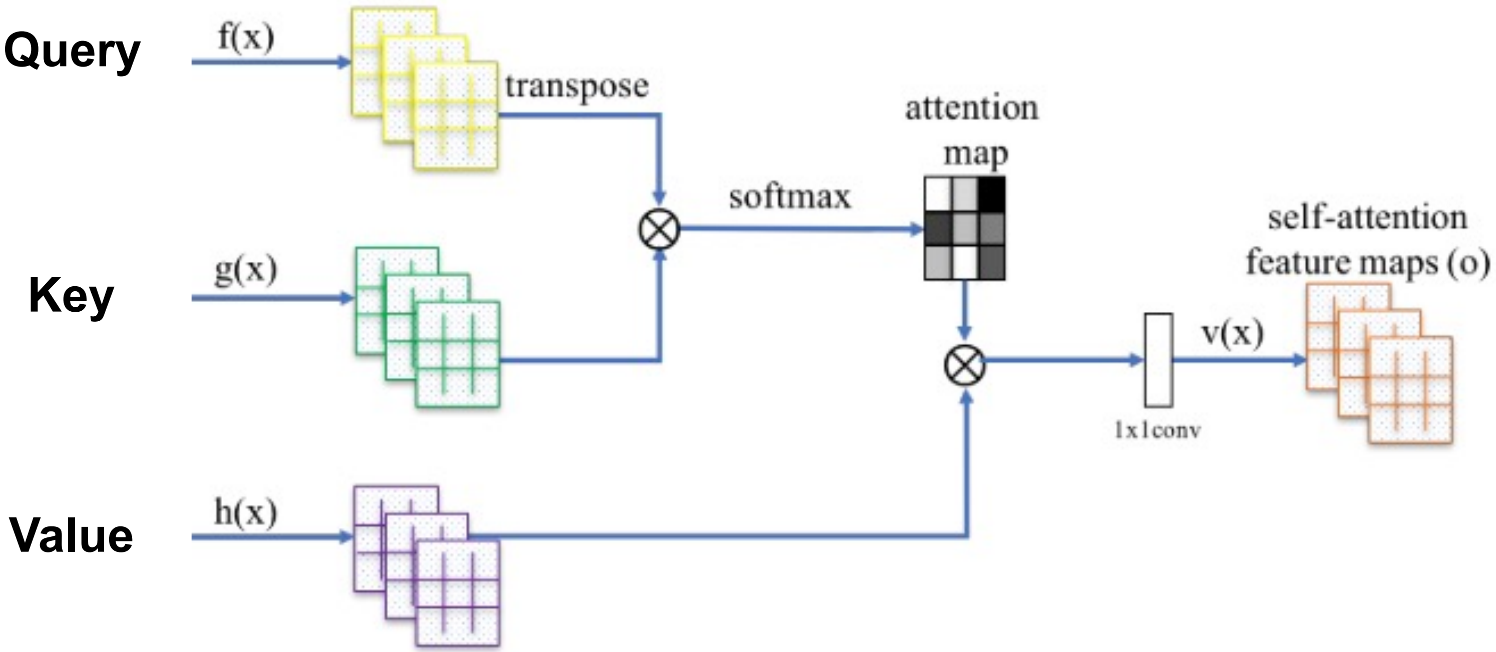
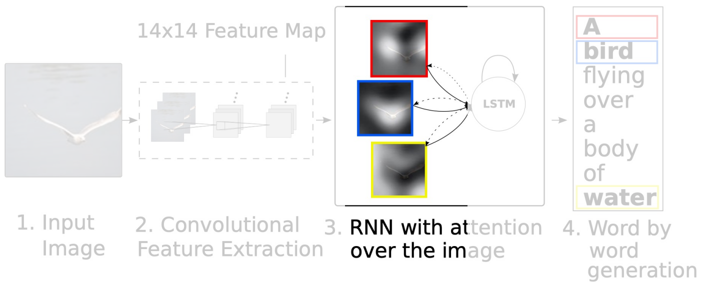


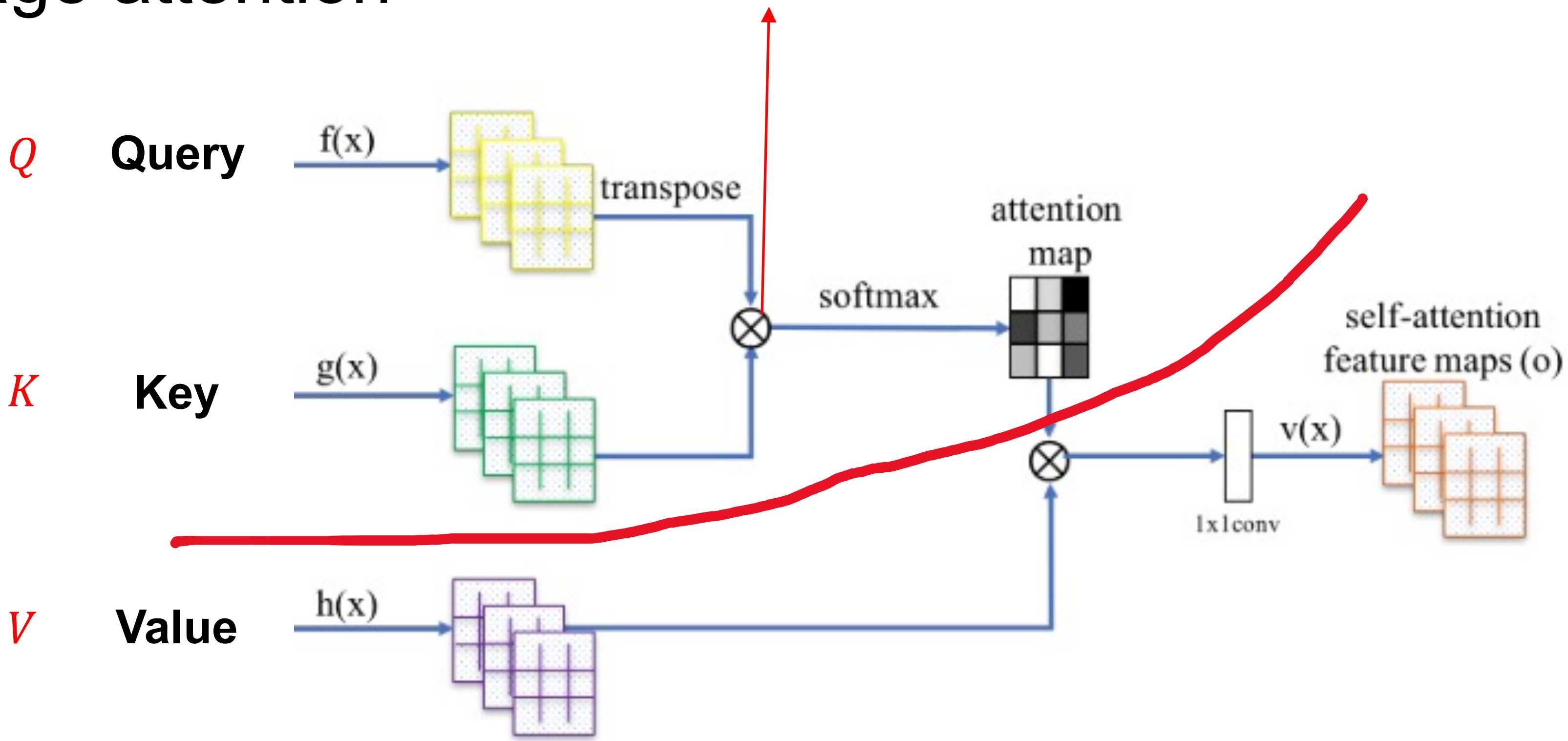
Image source:
https://medium.com/mlearning-ai/self-attention-in-convolutional-neural-networks-172d947afc00

Soft Attention



- Image attention

Attention function: $e_{ti} = f_{att}(Q, K)$



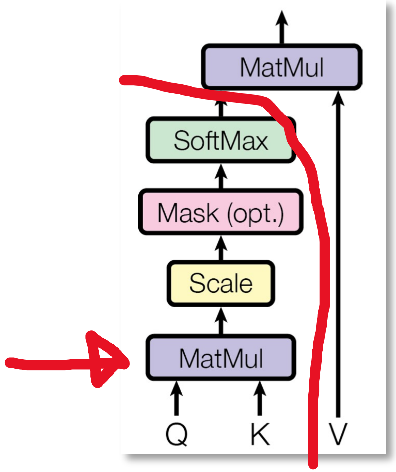
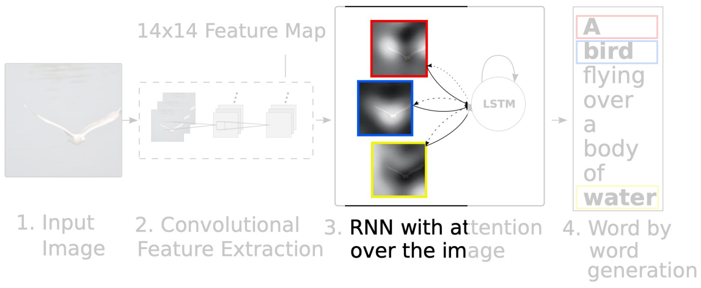


Image source: https://medium.com/mlearning-ai/self-attention-in-convolutional-neural-networks-172d947afc00

Soft Attention



- Image attention

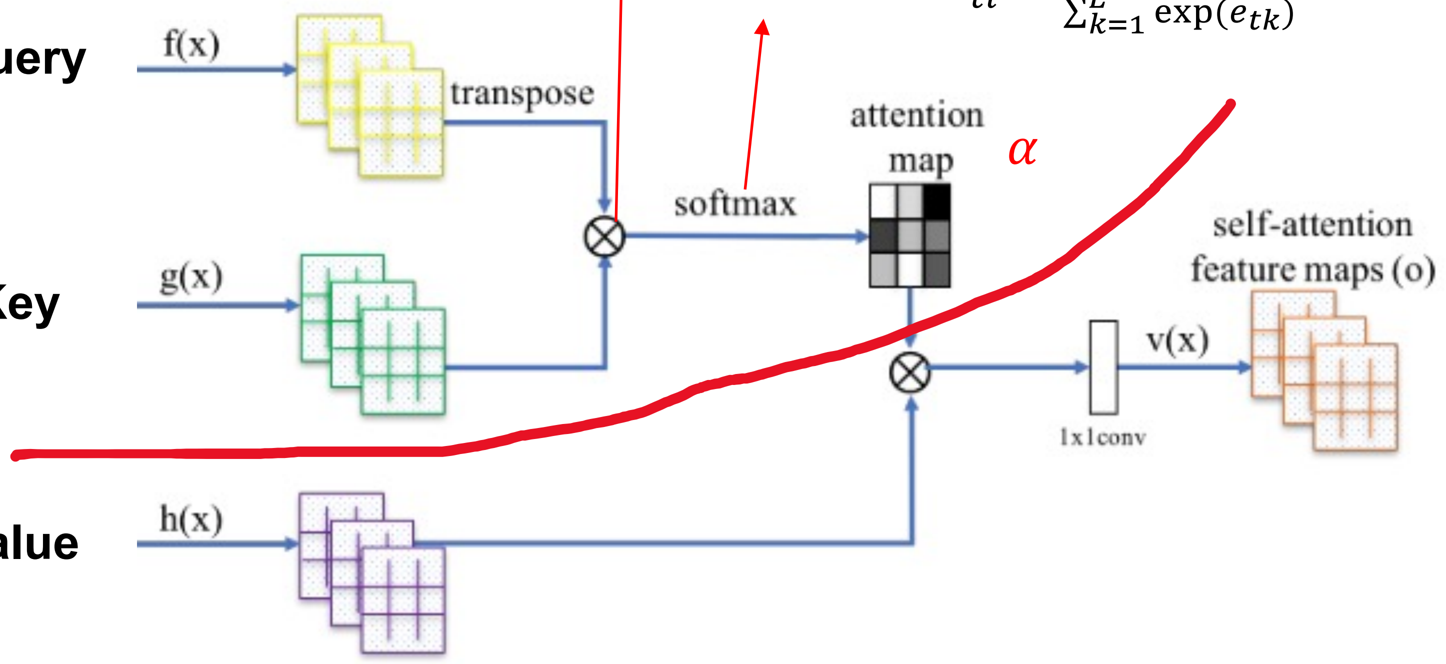
Attention function: $e_{ti} = f_{att}(Q, K)$

Softmax: $\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^{L} \exp(e_{tk})}$

Q **Query**

K **Key**

V **Value**



α

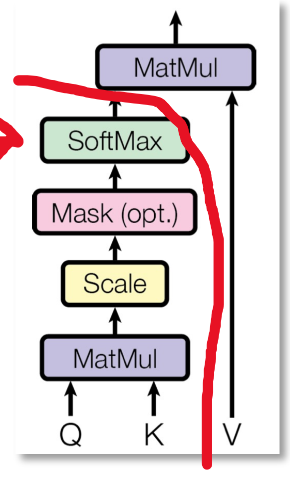
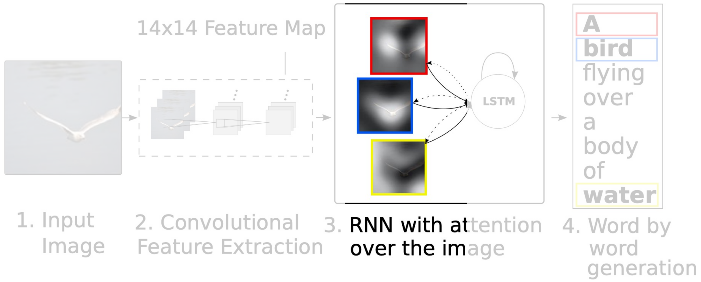


Image source: https://medium.com/mlearning-ai/self-attention-in-convolutional-neural-networks-172d947afc00

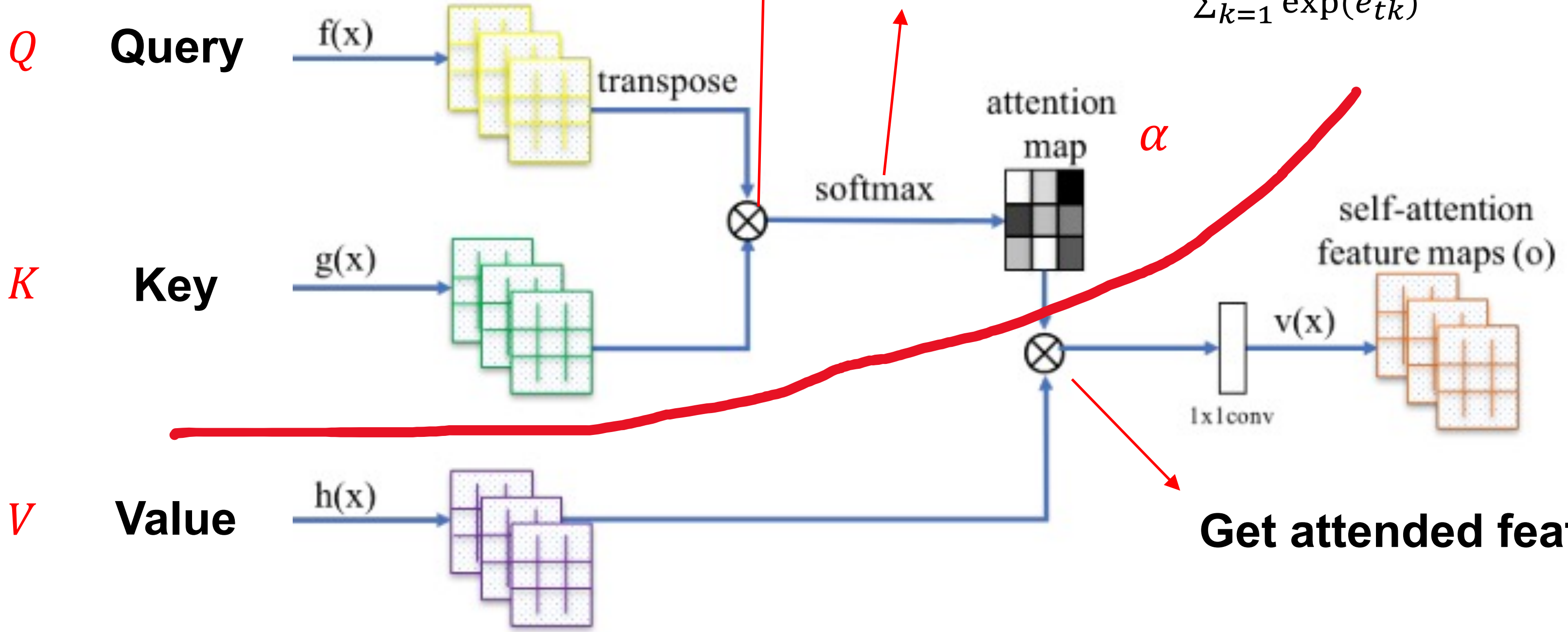
Soft Attention

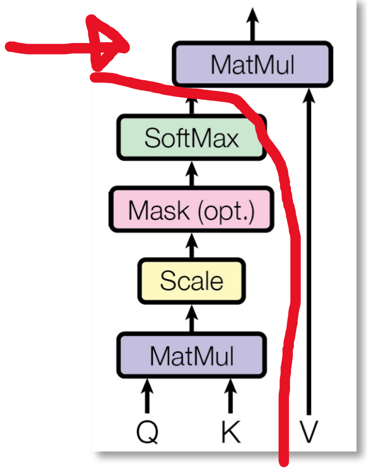


- Image attention

Attention function: $e_{ti} = f_{att}(Q, K)$

Softmax: $\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^{L} \exp(e_{tk})}$

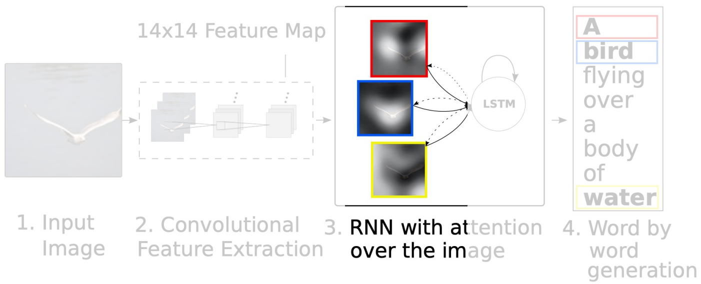




Get attended feature: $\emptyset(V, \alpha)$

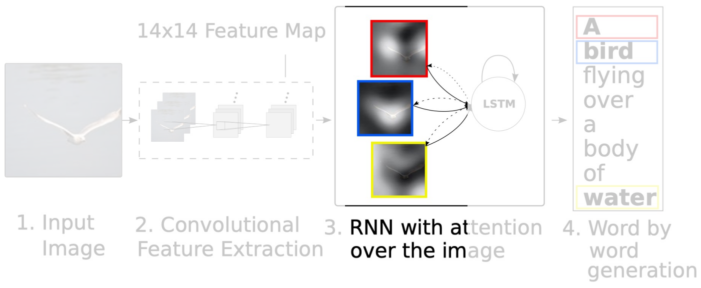
Image source: https://medium.com/mlearning-ai/self-attention-in-convolutional-neural-networks-172d947afc00

Hard Attention



- Binary, One-hot

Hard Attention



- Binary, One-hot

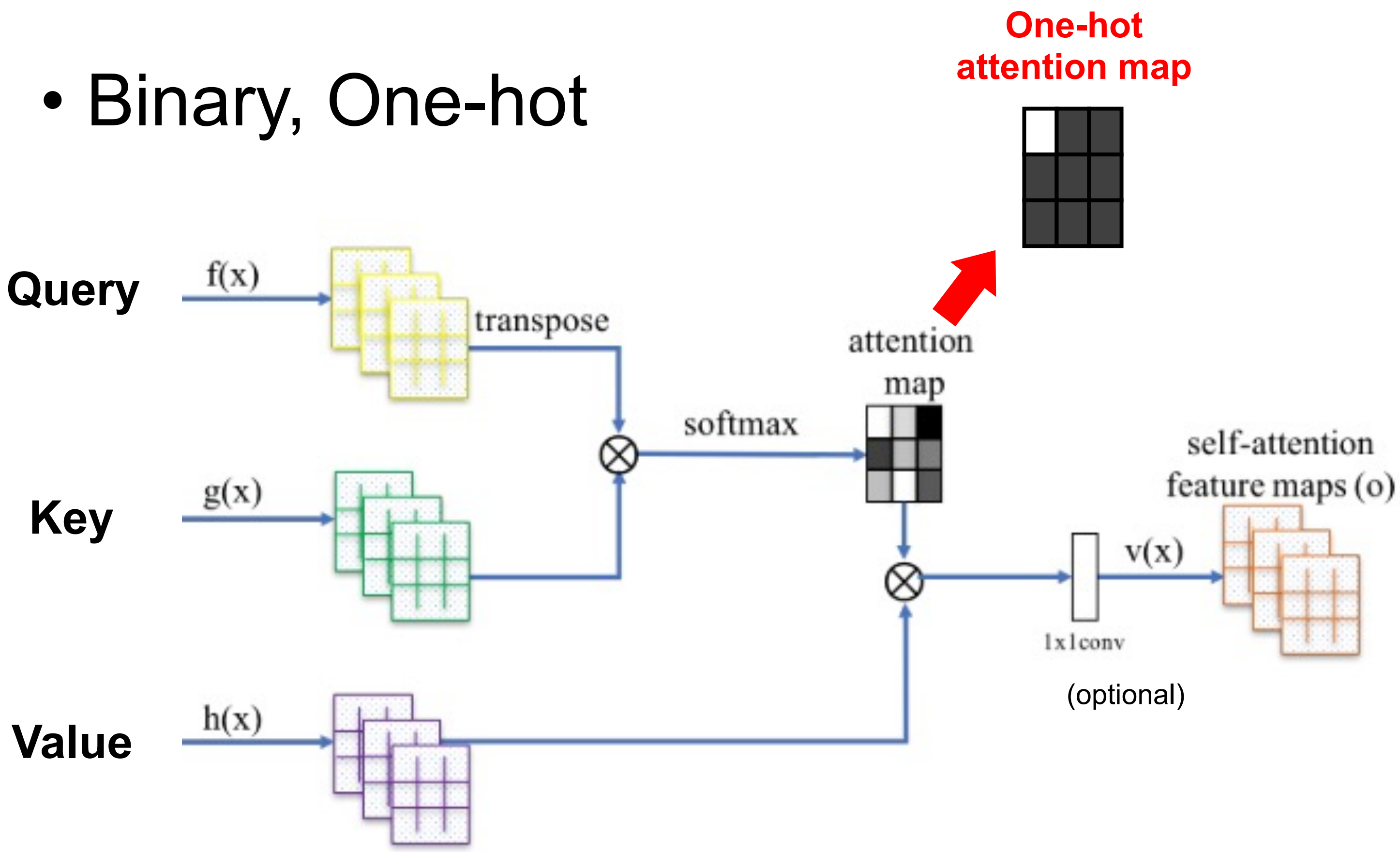
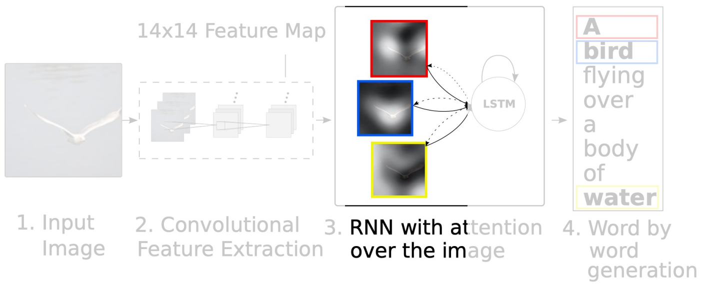


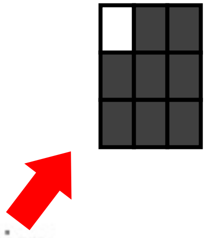
Image source:
https://medium.com/mlearning-ai/self-attention-in-convolutional-neural-networks-172d947afc00

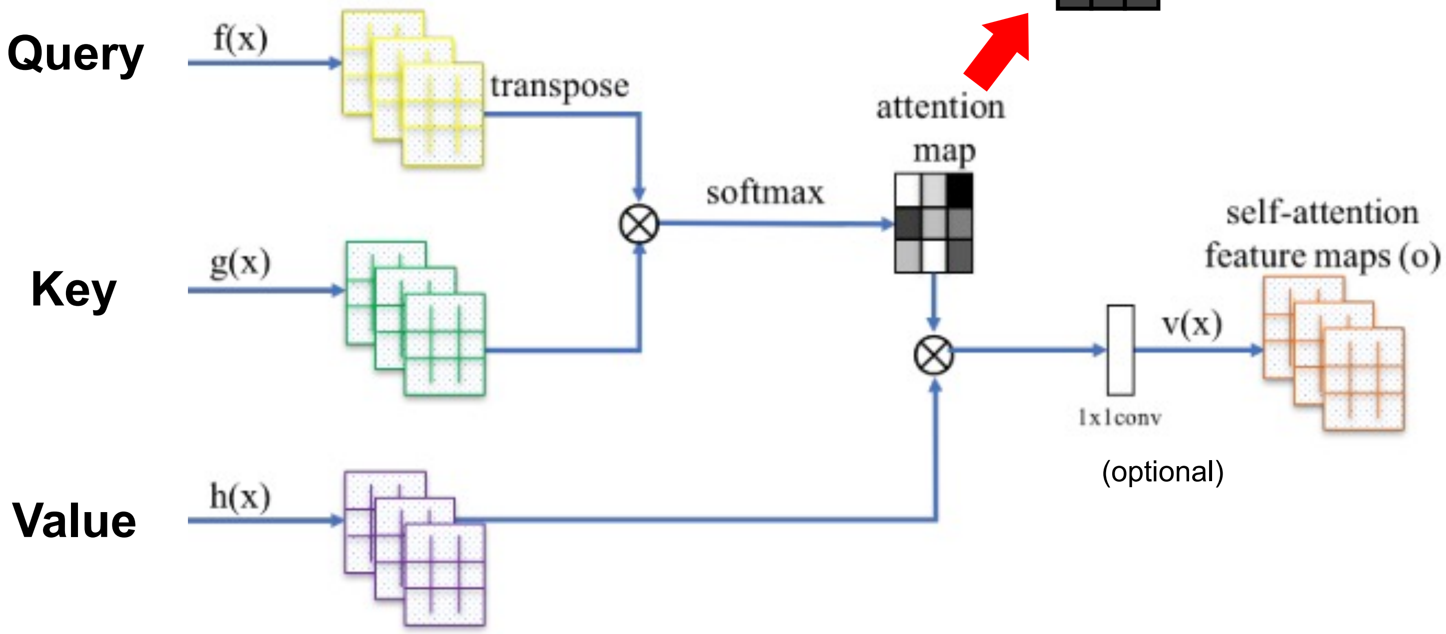
Hard Attention



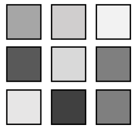
- Binary, One-hot

One-hot attention map



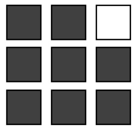


Soft Attention:



Softmax → sum = 1

Hard Attention:

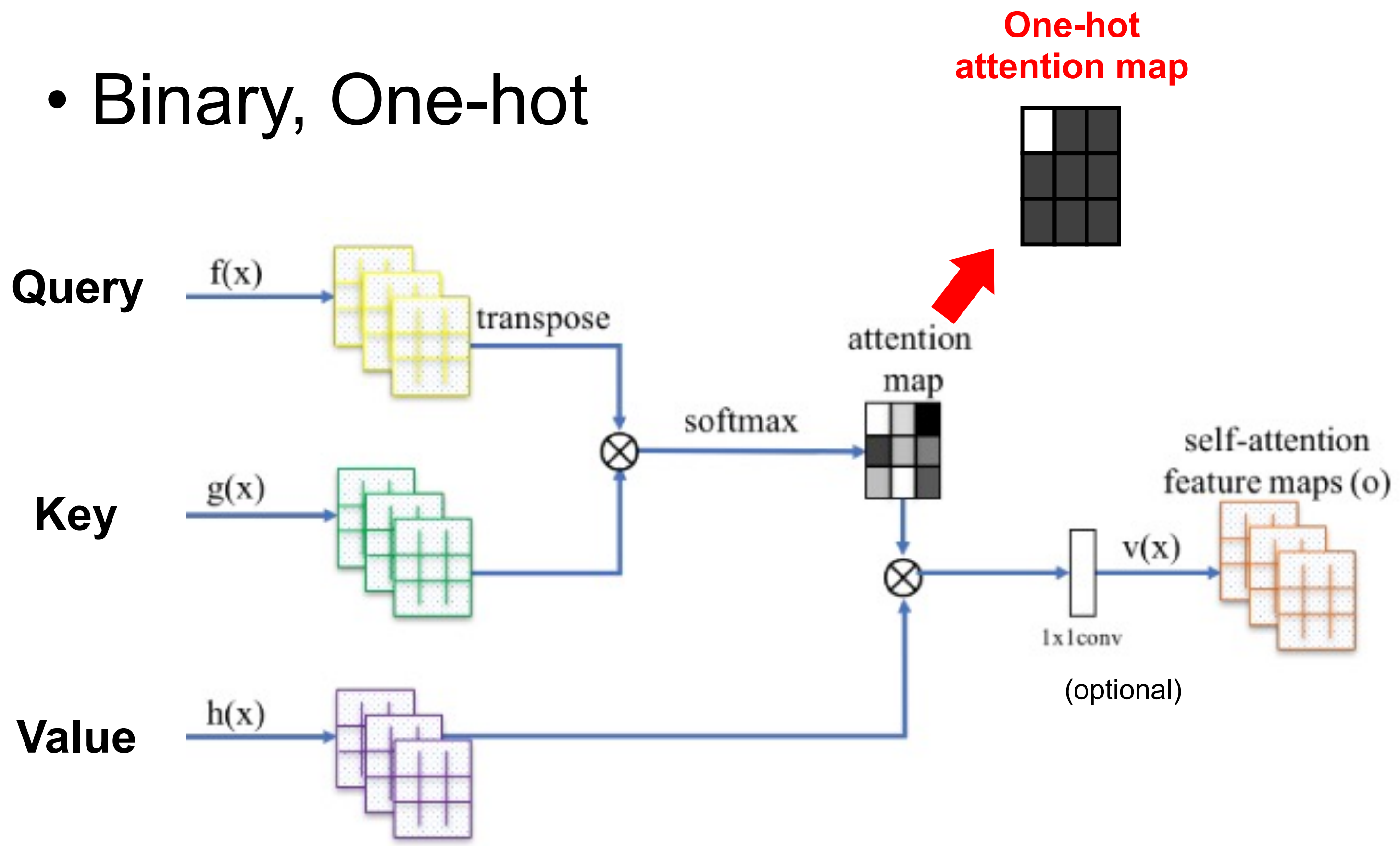


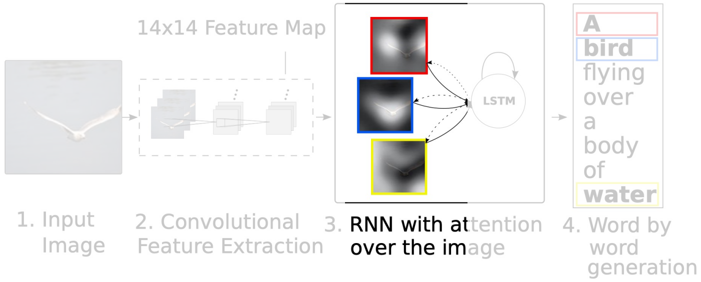
One hot → sum = 1

Image source: https://medium.com/mlearning-ai/self-attention-in-convolutional-neural-networks-172d947afc00

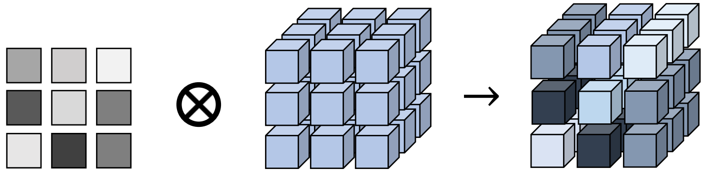
Hard Attention

- Binary, One-hot





Soft Attention:



Hard Attention:

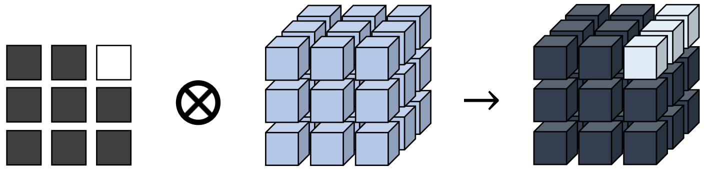
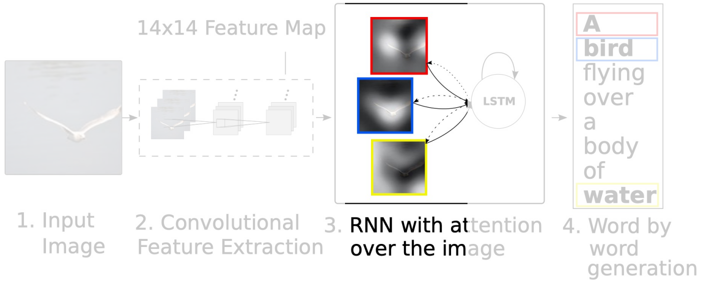


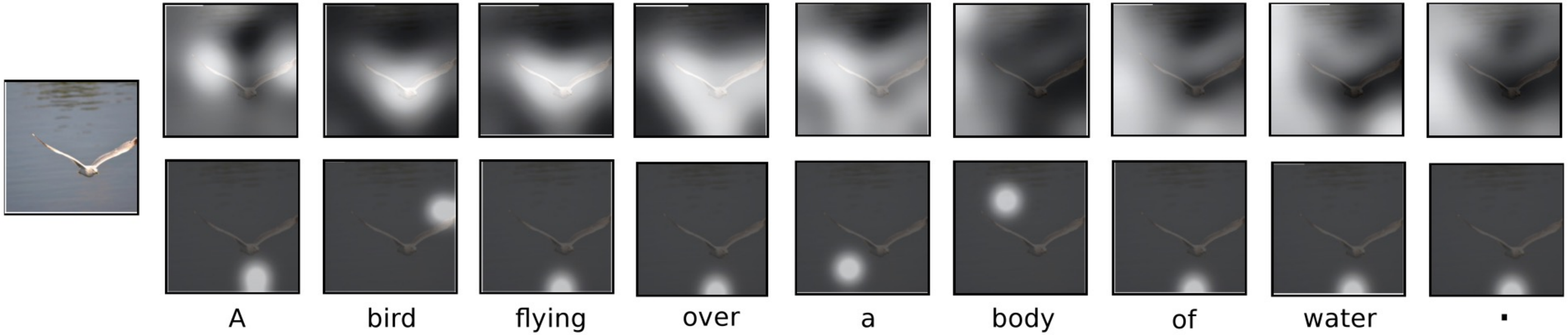
Image source: https://medium.com/mlearning-ai/self-attention-in-convolutional-neural-networks-172d947afc00

Attention



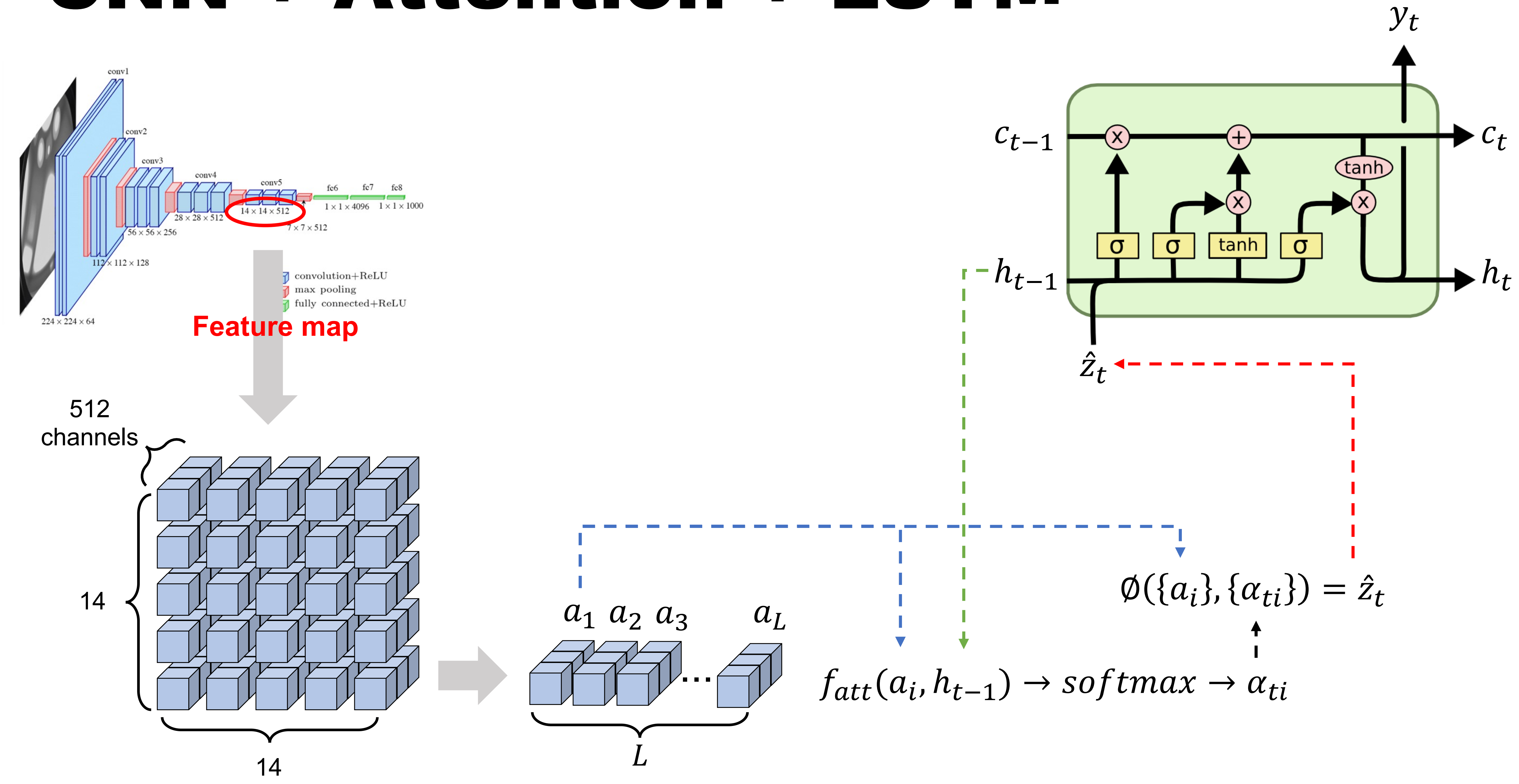
- Soft attention
- Hard attention

Soft attention: the attention map is a heatmap-like image

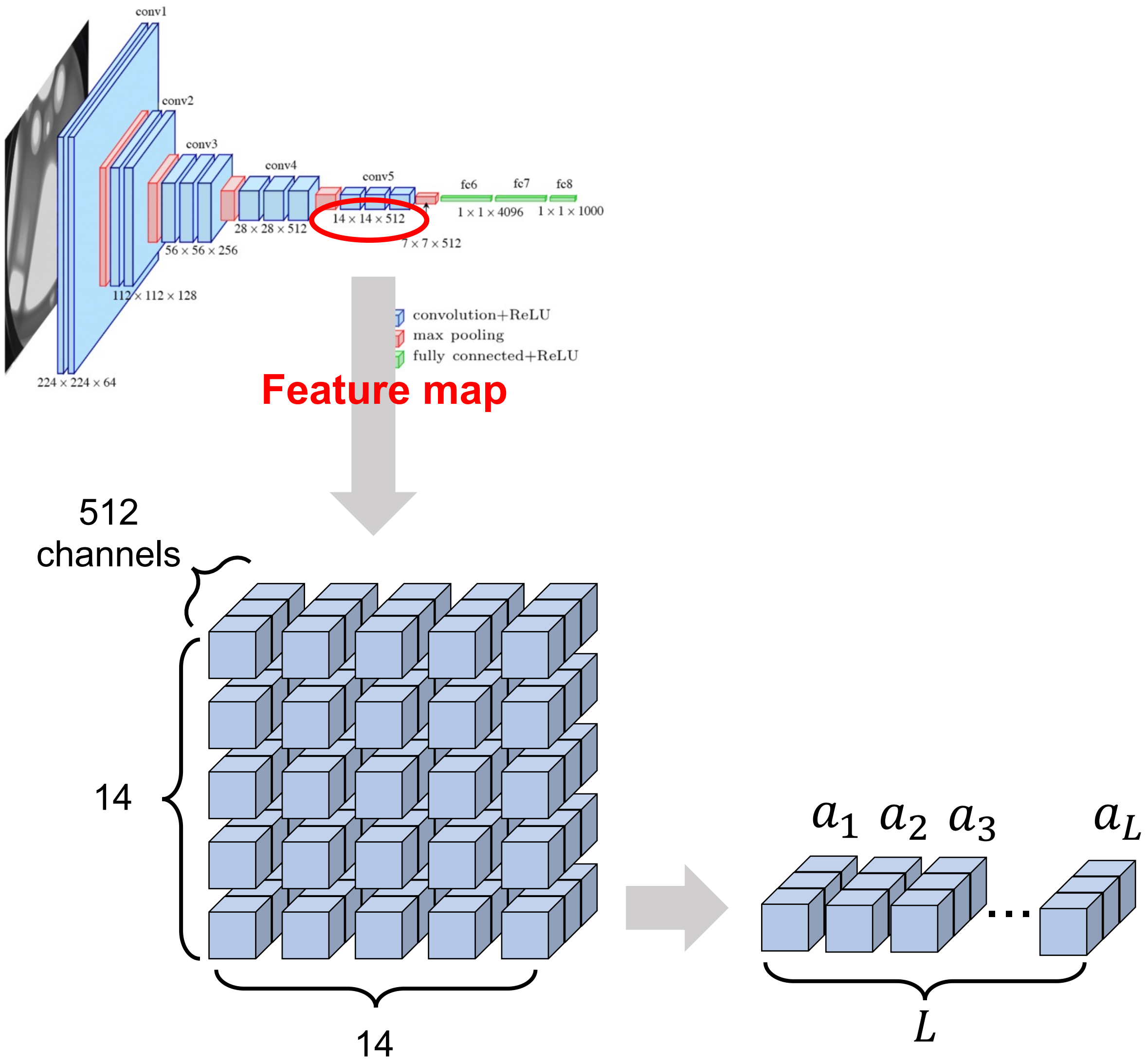


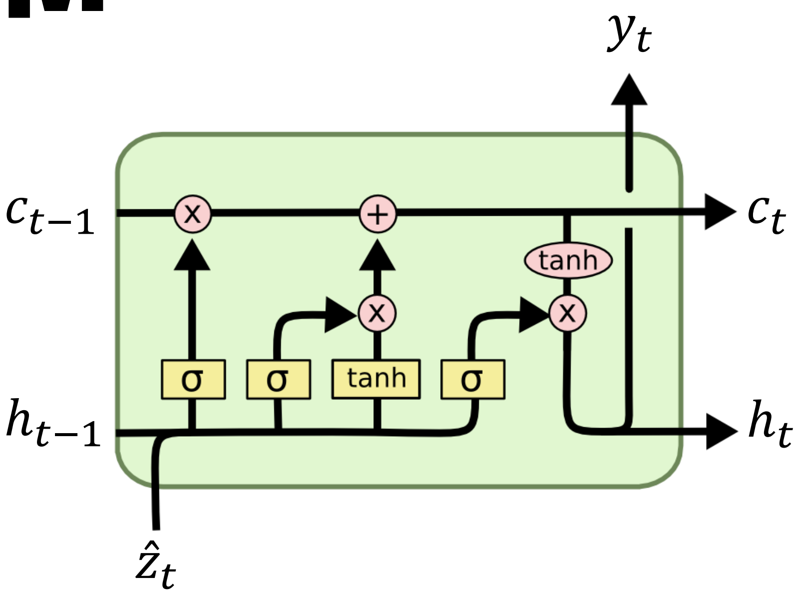
Hard attention: a single attention spot is selected on the map

CNN + Attention + LSTM



CNN + Attention + LSTM

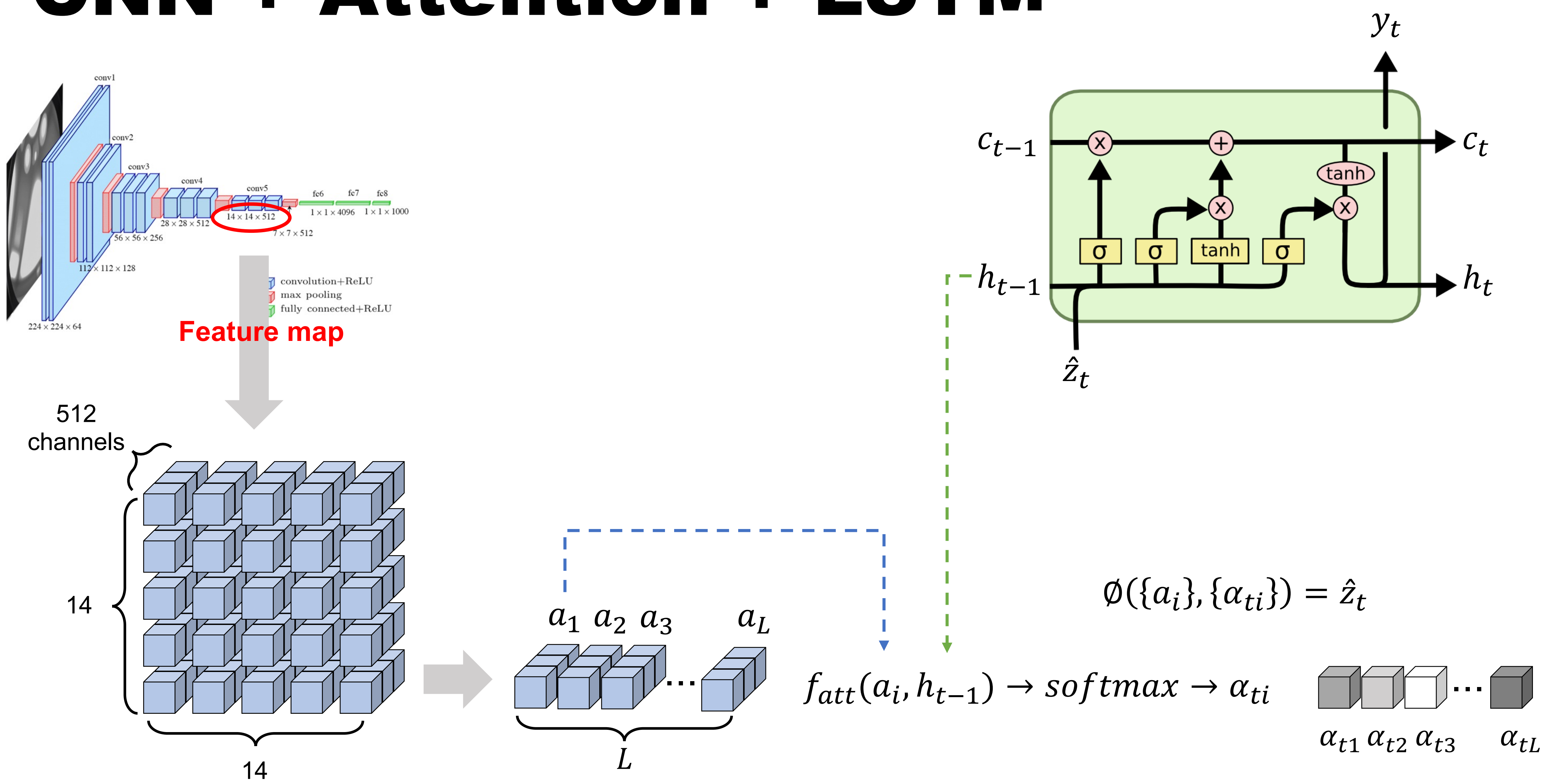




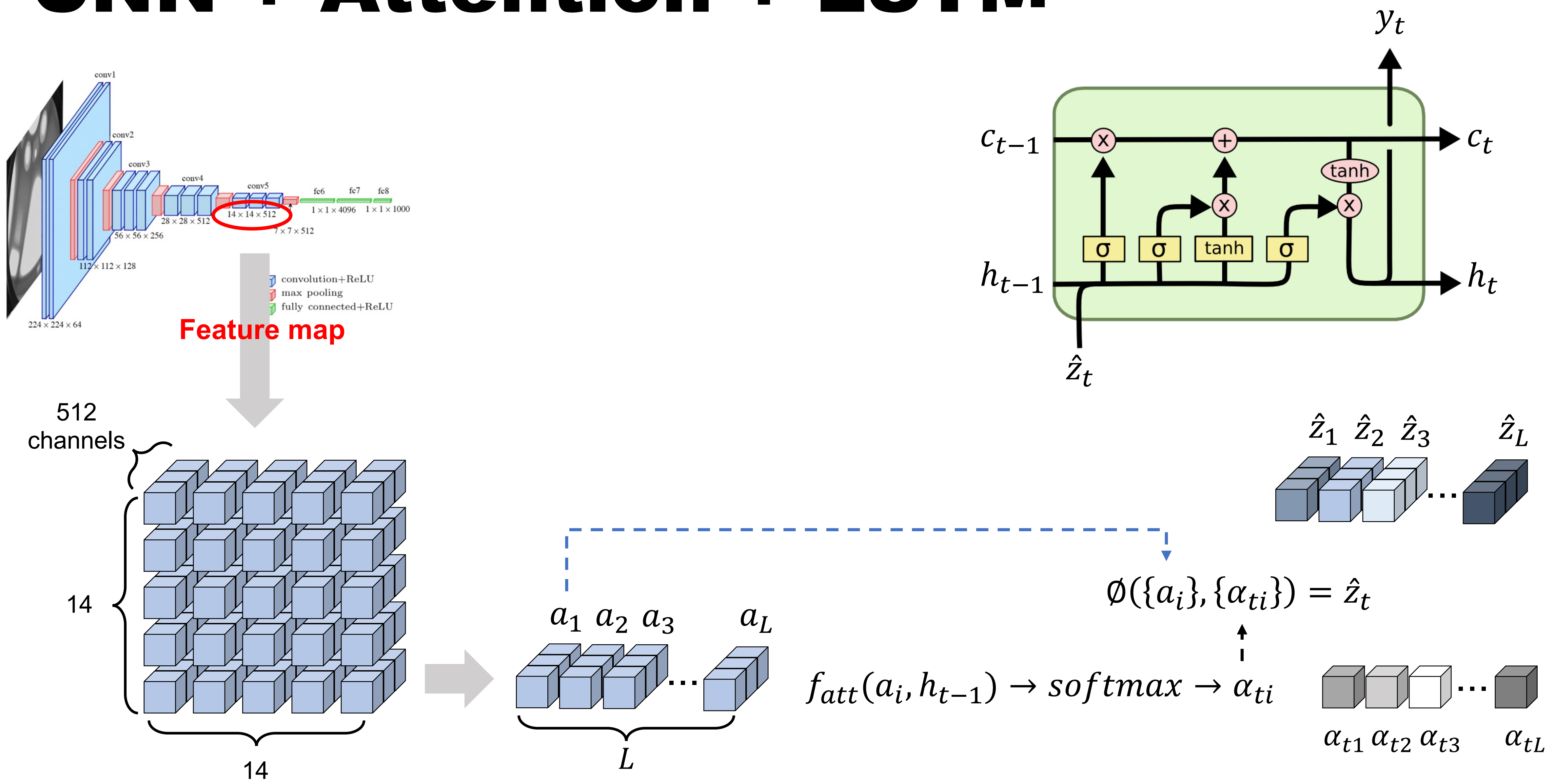
$$\emptyset(\{a_i\}, \{\alpha_{ti}\}) = \hat{z}_t$$

$$f_{att}(a_i, h_{t-1}) \rightarrow softmax \rightarrow \alpha_{ti}$$

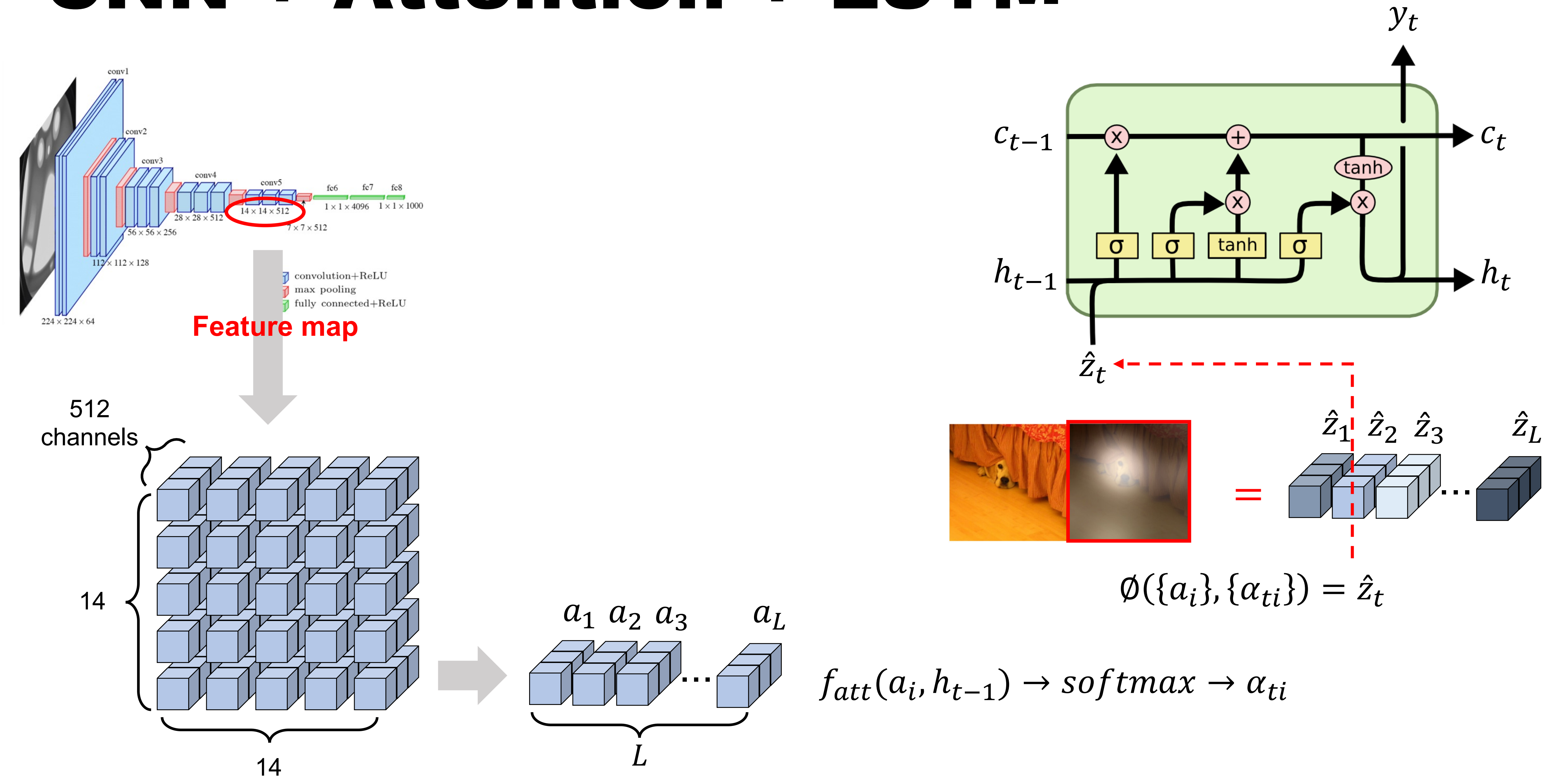
CNN + Attention + LSTM



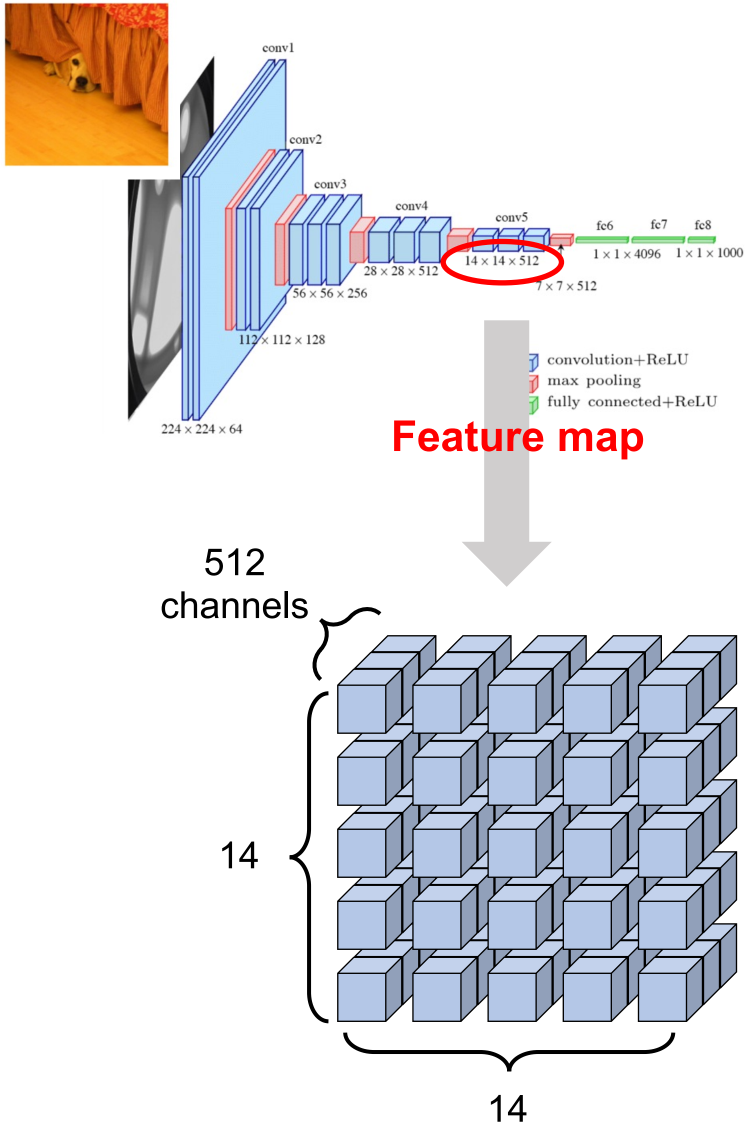
CNN + Attention + LSTM

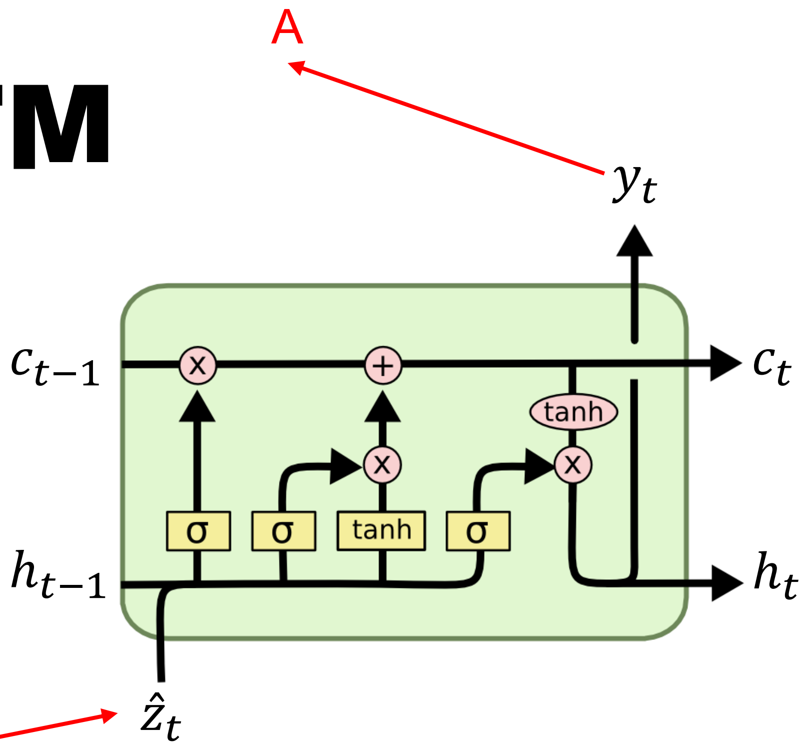


CNN + Attention + LSTM



CNN + Attention + LSTM

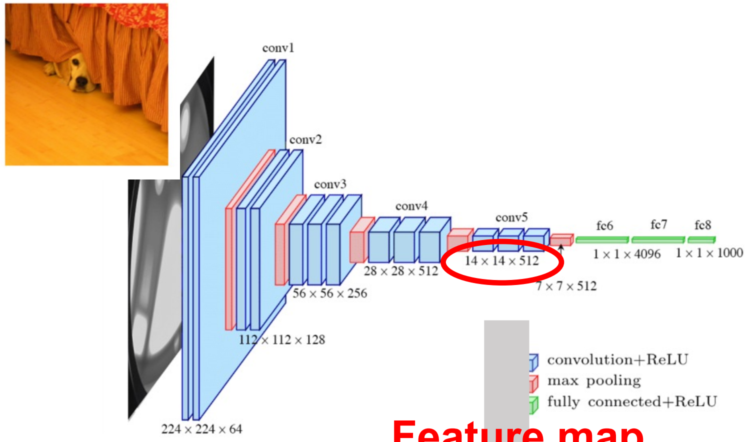




$$\emptyset(\{a_i\},\{\alpha_{ti}\}) = \hat{z}_t$$

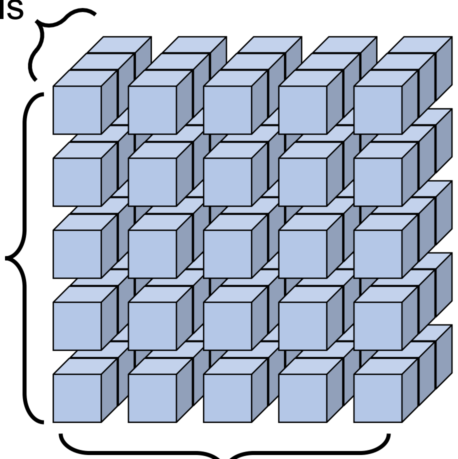
$$f_{att}(a_i, h_{t-1}) \rightarrow softmax \rightarrow \alpha_{ti}$$

CNN + Attention + LSTM



Feature map

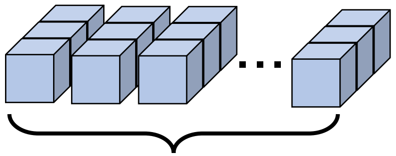
512 channels



14

14

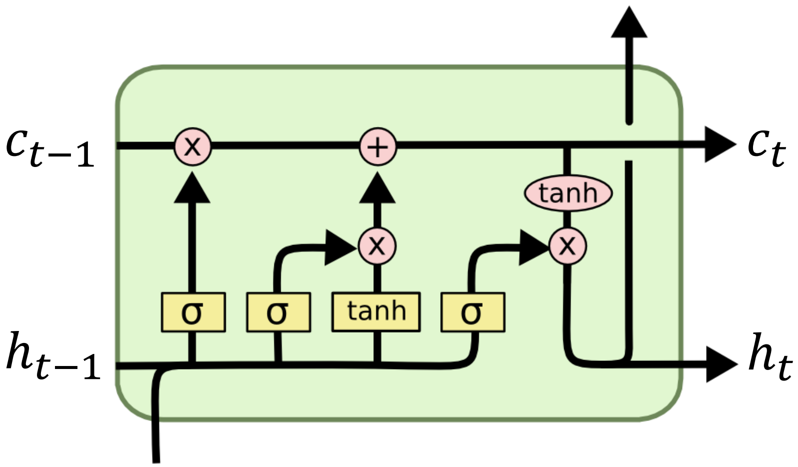
a_1 a_2 a_3 a_L



L

A DOG

y_t

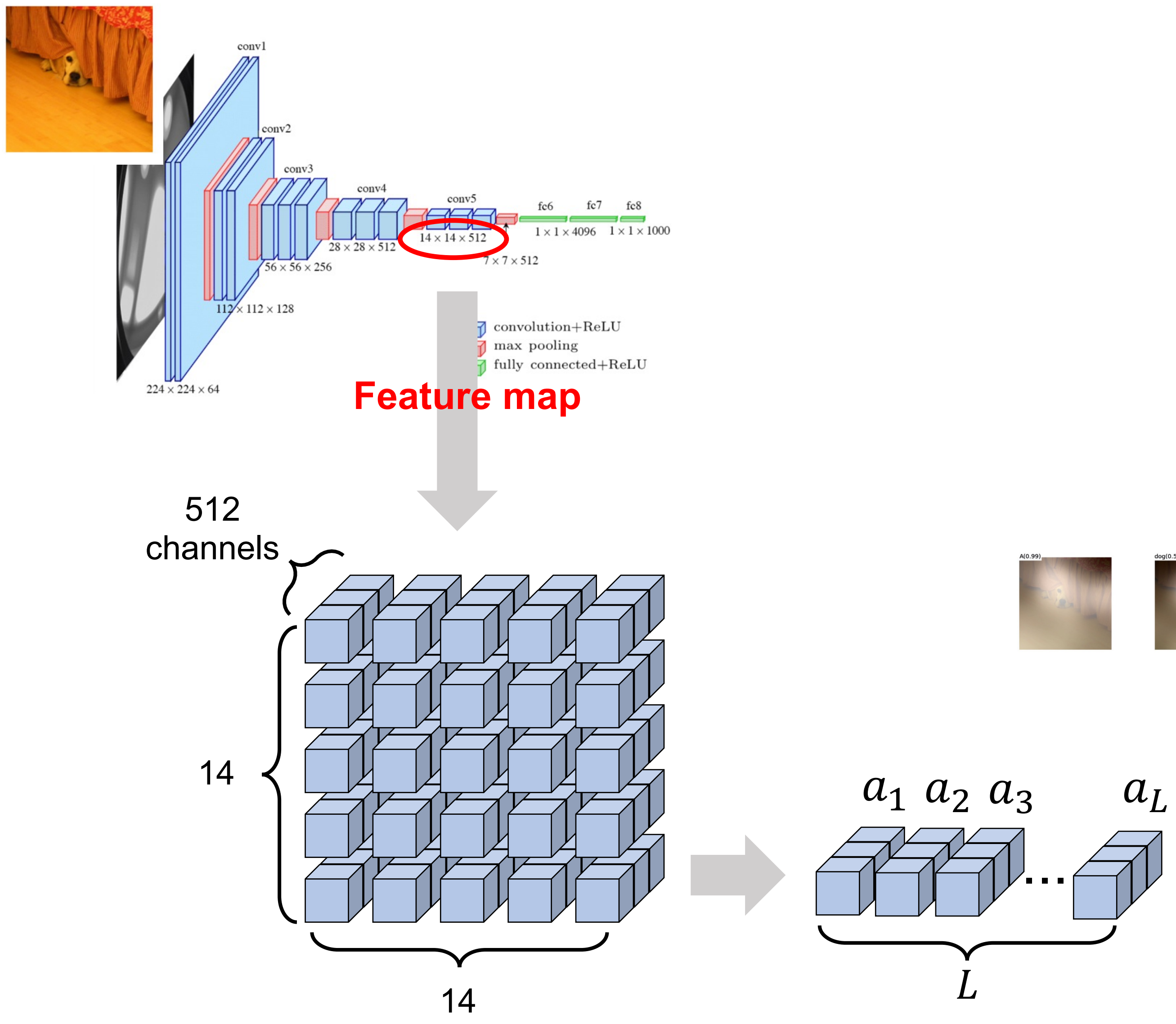


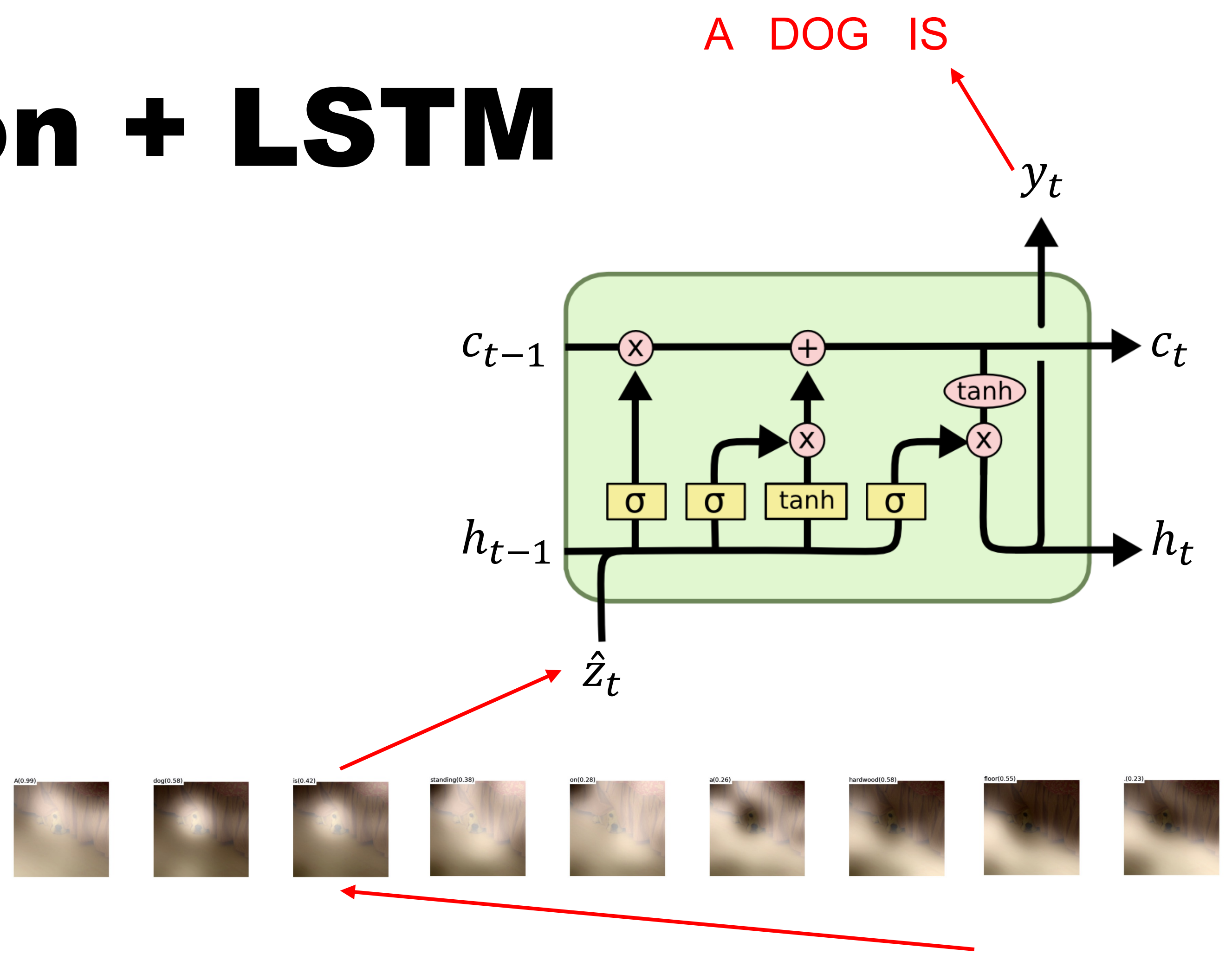
$\hat{z}_t$

$$\emptyset(\{a_i\}, \{\alpha_{ti}\}) = \hat{z}_t$$

$$f_{att}(a_i, h_{t-1}) \rightarrow softmax \rightarrow \alpha_{ti}$$

CNN + Attention + LSTM



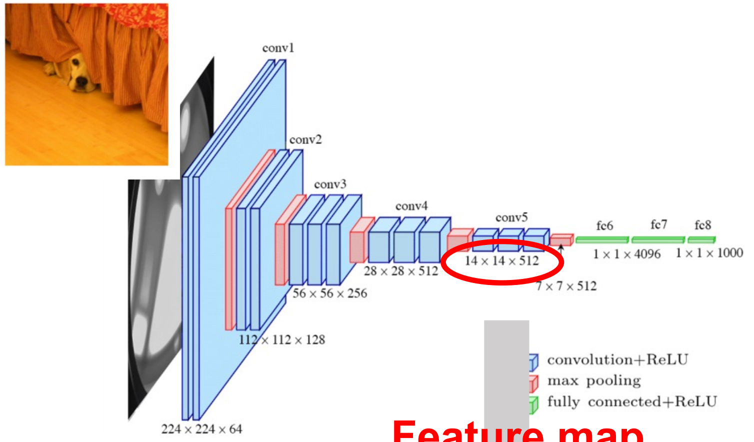


$$\emptyset(\{a_i\},\{\alpha_{ti}\}) = \hat{z}_t$$

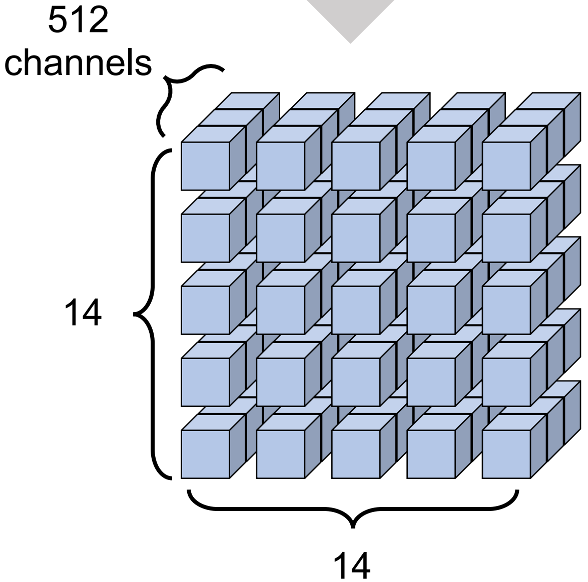
$$f_{att}(a_i, h_{t-1}) \rightarrow softmax \rightarrow \alpha_{ti}$$

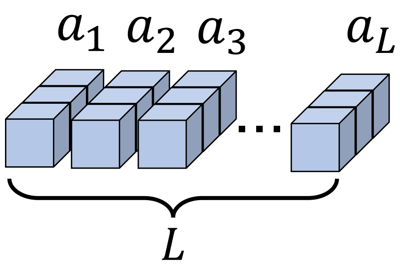
A DOG IS .... FLOOR .

CNN + Attention + LSTM

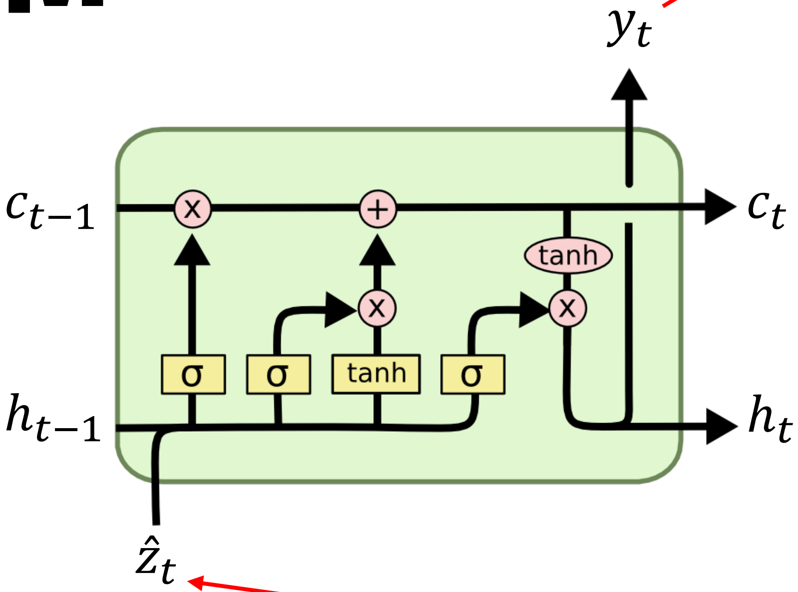


Feature map





$f_{att}(a_i, h_{t-1}) \rightarrow softmax \rightarrow \alpha_{ti}$



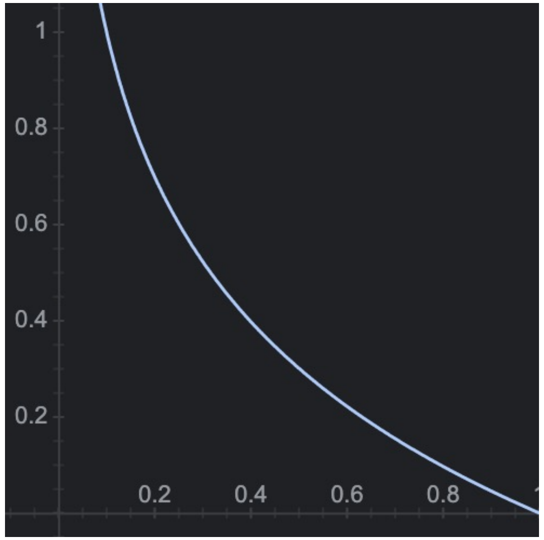
$\phi(\{a_i\}, \{\alpha_{ti}\}) = \hat{z}_t$

Loss Function

$$L_d = -\underline{\log(P(\mathbf{y}|\mathbf{x}))} + \lambda \sum_{i}^{L} (1 - \sum_{t}^{C} \alpha_{ti})^2$$

negative log-likelihood

penalty



P(y|x)

x: 

y: The dog is laying under a bed

Loss Function

$$L_d = -\log(P(\mathbf{y}|\mathbf{x})) + \underline{\lambda \sum_i^L (1 - \sum_t^C \alpha_{ti})^2}$$

$$e_{ti} = f_{\text{att}}(\mathbf{a}_i, \mathbf{h}_{t-1})$$

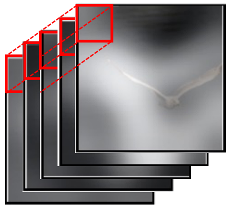
$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^{L} \exp(e_{tk})}$$

Softmax, so $\sum_i \alpha_{ti} = 1$

→ Sum over positions = 1

Loss Function

encourage $\sum_t \alpha_{ti} \approx 1$



→ Sum over time = 1

$$L_d = -\log(P(\mathbf{y}|\mathbf{x})) + \underline{\lambda \sum_i^L (1 - \sum_t^C \alpha_{ti})^2}$$

$$e_{ti} = f_{\text{att}}(\mathbf{a}_i, \mathbf{h}_{t-1})$$

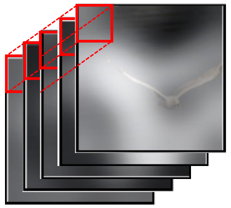
$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^L \exp(e_{tk})}$$

Softmax, so $\sum_i \alpha_{ti} = 1$

→ Sum over positions = 1

Loss Function

encourage $\sum_t \alpha_{ti} \approx 1$



→ Sum over time = 1

$$L_d = -\log(P(\mathbf{y}|\mathbf{x})) + \lambda \underline{\sum_{i}^{L}(1 - \sum_{t}^{C}\alpha_{ti})^2}$$

Practice: Calculate the loss for these attention maps.

α_t: α_1 α_2 α_3

α_1:

0.1	0.4
0.2	0.3

α_2:

0.1	0.1
0.1	0.7

α_3:

0.0	0.4
0.3	0.3

$$e_{ti} = f_{\text{att}}(\mathbf{a}_i, \mathbf{h}_{t-1})$$

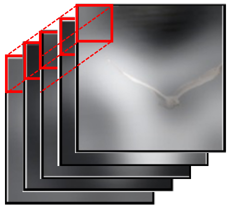
$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^{L} \exp(e_{tk})}$$

Softmax, so $\sum_i \alpha_{ti} = 1$

→ Sum over positions = 1

Loss Function

encourage $\sum_t \alpha_{ti} \approx 1$



→ Sum over time = 1

$$L_d = -\log(P(\mathbf{y}|\mathbf{x})) + \lambda \underline{\sum_i^L (1 - \sum_t^C \alpha_{ti})^2}$$

$$e_{ti} = f_{\text{att}}(\mathbf{a}_i, \mathbf{h}_{t-1})$$

$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^{L} \exp(e_{tk})}$$

Softmax, so $\sum_i \alpha_{ti} = 1$

→ Sum over positions = 1

Practice: Calculate the loss for these attention maps.

α_t:

α_1

0.1	0.4
0.2	0.3

α_2

0.1	0.1
0.1	0.7

α_3

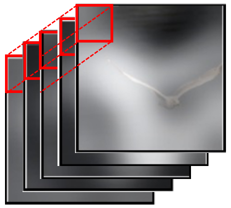
0.0	0.4
0.3	0.3

$1 - \sum_t^C \alpha_{ti}$

?	

Loss Function

encourage $\sum_t \alpha_{ti} \approx 1$



→ Sum over time = 1

$$L_d = -\log(P(\mathbf{y}|\mathbf{x})) + \lambda \sum_{i}^{L} (1 - \sum_{t}^{C} \alpha_{ti})^2$$

$$e_{ti} = f_{\text{att}}(\mathbf{a}_i, \mathbf{h}_{t-1})$$

$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^{L} \exp(e_{tk})}$$

Softmax, so $\sum_i \alpha_{ti} = 1$

→ Sum over positions = 1

Practice: Calculate the loss for these attention maps.

α_t: α_1

0.1	0.4
0.2	0.3

α_2

0.1	0.1
0.1	0.7

α_3

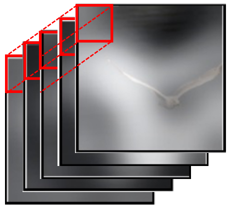
0.0	0.4
0.3	0.3

$1 - \sum_{t}^{C} \alpha_{ti}$

0.8	?

Loss Function

encourage $\sum_t \alpha_{ti} \approx 1$



→ Sum over time = 1

$$L_d = -\log(P(\mathbf{y}|\mathbf{x})) + \underline{\lambda \sum_{i}^{L}(1 - \sum_{t}^{C} \alpha_{ti})^2}$$

$$e_{ti} = f_{\text{att}}(\mathbf{a}_i, \mathbf{h}_{t-1})$$

$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^{L} \exp(e_{tk})}$$

Softmax, so $\sum_i \alpha_{ti} = 1$

→ Sum over positions = 1

Practice: Calculate the loss for these attention maps.

α_t:

α_1

0.1	0.4
0.2	0.3

α_2

0.1	0.1
0.1	0.7

α_3

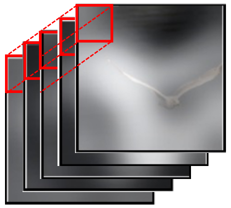
0.0	0.4
0.3	0.3

$1 - \sum_{t}^{C} \alpha_{ti}$

0.8	0.1
?	

Loss Function

encourage $\sum_t \alpha_{ti} \approx 1$



→ Sum over time = 1

$$L_d = -\log(P(\mathbf{y}|\mathbf{x})) + \lambda \underline{\sum_{i}^{L}(1 - \sum_{t}^{C} \alpha_{ti})^2}$$

$$e_{ti} = f_{\text{att}}(\mathbf{a}_i, \mathbf{h}_{t-1})$$

$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^{L} \exp(e_{tk})}$$

Softmax, so $\sum_i \alpha_{ti} = 1$

→ Sum over positions = 1

Practice: Calculate the loss for these attention maps.

α_t:

α_1

0.1	0.4
0.2	0.3

α_2

0.1	0.1
0.1	0.7

α_3

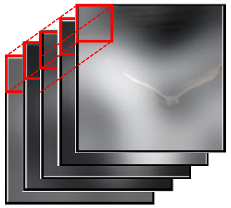
0.0	0.4
0.3	0.3

$1 - \sum_t^C \alpha_{ti}$

0.8	0.1
0.4	?

Loss Function

encourage $\sum_t \alpha_{ti} \approx 1$



→ Sum over time = 1

$$L_d = -\log(P(\mathbf{y}|\mathbf{x})) + \underline{\lambda \sum_{i}^{L}(1 - \sum_{t}^{C} \alpha_{ti})^2}$$

$$e_{ti} = f_{\text{att}}(\mathbf{a}_i, \mathbf{h}_{t-1})$$

$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^{L} \exp(e_{tk})}$$

Softmax, so $\sum_i \alpha_{ti} = 1$

→ Sum over positions = 1

Practice: Calculate the loss for these attention maps.

α_t: α_1

0.1	0.4
0.2	0.3

α_2

0.1	0.1
0.1	0.7

α_3

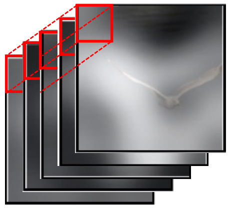
0.0	0.4
0.3	0.3

$1 - \sum_{t}^{C} \alpha_{ti}$

0.8	0.1
0.4	-0.3

Loss Function

encourage $\sum_t \alpha_{ti} \approx 1$



→ Sum over time = 1

$$L_d = -\log(P(\mathbf{y}|\mathbf{x})) + \lambda \underline{\sum_{i}^{L}(1 - \sum_{t}^{C} \alpha_{ti})^2}$$

$$e_{ti} = f_{\text{att}}(\mathbf{a}_i, \mathbf{h}_{t-1})$$

$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^{L} \exp(e_{tk})}$$

Softmax, so $\sum_i \alpha_{ti} = 1$

→ Sum over positions = 1

Practice: Calculate the loss for these attention maps.

α_t: α_1

0.1	0.4
0.2	0.3

α_2

0.1	0.1
0.1	0.7

α_3

0.0	0.4
0.3	0.3

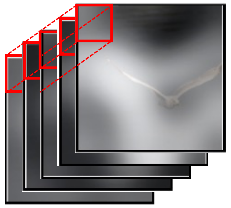
$1 - \sum_{t}^{C} \alpha_{ti}$

0.8	0.1
0.4	-0.3

$$\sum_{i}^{L}(1 - \sum_{t}^{C} \alpha_{ti})^2 = ?$$

Loss Function

encoding $\sum_t \alpha_{ti} \approx 1$



→ Sum over time = 1

$$L_d = -\log(P(\mathbf{y}|\mathbf{x})) + \underline{\lambda \sum_i^L (1 - \sum_t^C \alpha_{ti})^2}$$

$$e_{ti} = f_{\text{att}}(\mathbf{a}_i, \mathbf{h}_{t-1})$$

$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^L \exp(e_{tk})}$$

Softmax, so $\sum_i \alpha_{ti} = 1$

→ Sum over positions = 1

Practice: Calculate the loss for these attention maps.

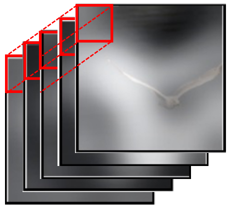
α_t: α_1 α_2 α_3 $1 - \sum_t^C \alpha_{ti}$

α_1		α_2		α_3		$1 - \sum_t^C \alpha_{ti}$	
0.1	0.4	0.1	0.1	0.0	0.4	0.8	0.1
0.2	0.3	0.1	0.7	0.3	0.3	0.4	-0.3

$$\sum_i^L (1 - \sum_t^C \alpha_{ti})^2 = 0.8^2 + 0.1^2 + 0.4^2 + 0.3^2$$
$$= 0.9$$

Loss Function

encourage $\sum_t \alpha_{ti} \approx 1$



→ Sum over time = 1

$$L_d = -\log(P(\mathbf{y}|\mathbf{x})) + \lambda \sum_i^L (1 - \sum_t^C \alpha_{ti})^2$$

$$e_{ti} = f_{\text{att}}(\mathbf{a}_i, \mathbf{h}_{t-1})$$

$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^L \exp(e_{tk})}$$

Softmax, so $\sum_i \alpha_{ti} = 1$

→ Sum over positions = 1

Practice: What is the max value of $\sum_i^L (1 - \sum_t^C \alpha_{ti})^2$?

α_t: α_1 α_2 α_3

α_1:

?	?
?	?

α_2:

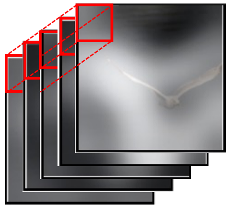
?	?
?	?

α_3:

?	?
?	?

Loss Function

encourage $\sum_t \alpha_{ti} \approx 1$



→ Sum over time = 1

$$L_d = -\log(P(\mathbf{y}|\mathbf{x})) + \lambda \underline{\sum_{i}^{L}(1 - \sum_{t}^{C} \alpha_{ti})^2}$$

$$e_{ti} = f_{\text{att}}(\mathbf{a}_i, \mathbf{h}_{t-1})$$

$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^{L} \exp(e_{tk})}$$

Softmax, so $\sum_i \alpha_{ti} = 1$

→ Sum over positions = 1

Practice: What is the max value of $\sum_{i}^{L}(1 - \sum_{t}^{C} \alpha_{ti})^2$?

α_t: α_1 α_2 α_3

?	?
?	?

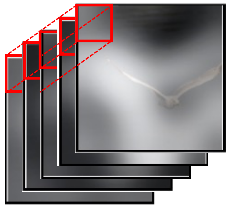
?	?
?	?

?	?
?	?

$$\sum_{t}^{C} \alpha_{ti}$$

Loss Function

encourage $\sum_t \alpha_{ti} \approx 1$



→ Sum over time = 1

$$L_d = -\log(P(\mathbf{y}|\mathbf{x})) + \lambda \underline{\sum_i^L (1 - \sum_t^C \alpha_{ti})^2}$$

Practice: What is the max value of $\sum_i^L (1 - \sum_t^C \alpha_{ti})^2$?

α_t: α_1 α_2 α_3

?	?		?	?		?	?
?	?		?	?		?	?

$$0 \leq \sum_t^C \alpha_{ti} \leq 3$$

$$e_{ti} = f_{\text{att}}(\mathbf{a}_i, \mathbf{h}_{t-1})$$

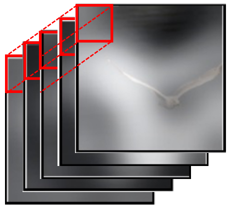
$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^L \exp(e_{tk})}$$

Softmax, so $\sum_i \alpha_{ti} = 1$

→ Sum over positions = 1

Loss Function

encourage $\sum_{t} \alpha_{ti} \approx 1$



→ Sum over time = 1

$$L_d = -\log(P(\mathbf{y}|\mathbf{x})) + \lambda \underline{\sum_i^L (1 - \sum_t^C \alpha_{ti})^2}$$

Practice: What is the max value of $\sum_i^L (1 - \sum_t^C \alpha_{ti})^2$?

α_t: α_1 α_2 α_3

?	?		?	?		?	?
?	?		?	?		?	?

$$e_{ti} = f_{\text{att}}(\mathbf{a}_i, \mathbf{h}_{t-1})$$

$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^L \exp(e_{tk})}$$

$0 \leq \sum_t^C \alpha_{ti} \leq 3$

$|1 - \sum_t^C \alpha_{ti}|$

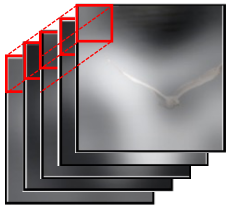
Softmax, so $\sum_i \alpha_{ti} = 1$

→ Sum over positions = 1

Loss Function

$$L_d = -\log(P(\mathbf{y}|\mathbf{x})) + \lambda \underline{\sum_{i}^{L}(1 - \sum_{t}^{C} \alpha_{ti})^2}$$

encourage $\sum_t \alpha_{ti} \approx 1$



→ Sum over time = 1

$$e_{ti} = f_{\text{att}}(\mathbf{a}_i, \mathbf{h}_{t-1})$$

$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^{L} \exp(e_{tk})}$$

Softmax, so $\sum_i \alpha_{ti} = 1$

→ Sum over positions = 1

Practice: What is the max value of $\sum_{i}^{L}(1 - \sum_{t}^{C} \alpha_{ti})^2$?

α_t: α_1 α_2 α_3

?	?
?	?

?	?
?	?

?	?
?	?

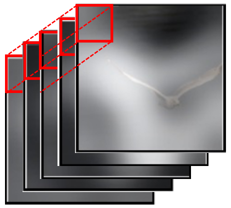
$$0 \leq \sum_{t}^{C} \alpha_{ti} \leq 3$$

$$0 \leq |1 - \sum_{t}^{C} \alpha_{ti}| \leq 2$$

Loss Function

$$L_d = -\log(P(\mathbf{y}|\mathbf{x})) + \lambda \underline{\sum_i^L (1 - \sum_t^C \alpha_{ti})^2}$$

encourage $\sum_t \alpha_{ti} \approx 1$



→ Sum over time = 1

$$e_{ti} = f_{\text{att}}(\mathbf{a}_i, \mathbf{h}_{t-1})$$

$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^{L} \exp(e_{tk})}$$

Softmax, so $\sum_i \alpha_{ti} = 1$

→ Sum over positions = 1

Practice: What is the max value of $\sum_i^L (1 - \sum_t^C \alpha_{ti})^2$?

α_t: α_1 α_2 α_3

?	?
?	?

?	?
?	?

?	?
?	?

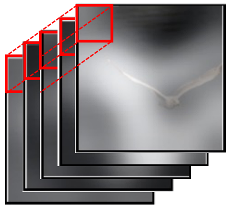
$$0 \le \sum_t^C \alpha_{ti} \le 3$$

$$0 \le |1 - \sum_t^C \alpha_{ti}| \le 2$$

$$(1 - \sum_t^C \alpha_{ti})^2$$

Loss Function

encouraging $\sum_t \alpha_{ti} \approx 1$



→ Sum over time = 1

$$L_d = -\log(P(\mathbf{y}|\mathbf{x})) + \lambda \underline{\sum_i^L (1 - \sum_t^C \alpha_{ti})^2}$$

$$e_{ti} = f_{\text{att}}(\mathbf{a}_i, \mathbf{h}_{t-1})$$

$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^{L} \exp(e_{tk})}$$

Softmax, so $\sum_i \alpha_{ti} = 1$

→ Sum over positions = 1

Practice: What is the max value of $\sum_i^L (1 - \sum_t^C \alpha_{ti})^2$?

α_t: α_1 α_2 α_3

?	?		?	?		?	?
?	?		?	?		?	?

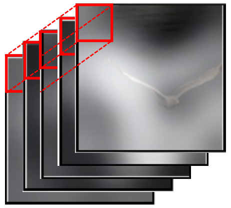
$$0 \le \sum_t^C \alpha_{ti} \le 3$$

$$0 \le |1 - \sum_t^C \alpha_{ti}| \le 2$$

$$0 \le (1 - \sum_t^C \alpha_{ti})^2 \le 4$$

Loss Function

encourage $\sum_t \alpha_{ti} \approx 1$



→ Sum over time = 1

$$L_d = -\log(P(\mathbf{y}|\mathbf{x})) + \lambda \underline{\sum_i^L (1 - \sum_t^C \alpha_{ti})^2}$$

$$e_{ti} = f_{\text{att}}(\mathbf{a}_i, \mathbf{h}_{t-1})$$

$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^L \exp(e_{tk})}$$

Softmax, so $\sum_i \alpha_{ti} = 1$

→ Sum over positions = 1

Practice: What is the max value of $\sum_i^L (1 - \sum_t^C \alpha_{ti})^2$?

α_t: α_1

0	1
0	0

α_2

0	1
0	0

α_3

0	1
0	0

$1 - \sum_t^C \alpha_{ti}$

1	-2
1	1

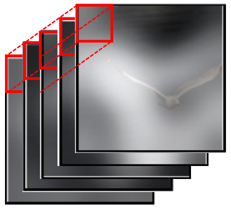
$$0 \leq \sum_t^C \alpha_{ti} \leq 3$$

$$0 \leq |1 - \sum_t^C \alpha_{ti}| \leq 2$$

$$0 \leq (1 - \sum_t^C \alpha_{ti})^2 \leq 4$$

Loss Function

encourage $\sum_t \alpha_{ti} \approx 1$



→ Sum over time = 1

$$L_d = -\log(P(\mathbf{y}|\mathbf{x})) + \lambda \underline{\sum_i^L (1 - \sum_t^C \alpha_{ti})^2}$$

$$e_{ti} = f_{\text{att}}(\mathbf{a}_i, \mathbf{h}_{t-1})$$

$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^{L} \exp(e_{tk})}$$

Softmax, so $\sum_i \alpha_{ti} = 1$

→ Sum over positions = 1

Practice: What is the max value of $\sum_i^L (1 - \sum_t^C \alpha_{ti})^2$?

α_t: α_1

0	1
0	0

α_2

0	1
0	0

α_3

0	1
0	0

$1 - \sum_t^C \alpha_{ti}$

1	-2
1	1

$(1 - \sum_t^C \alpha_{ti})^2$

1	4
1	1

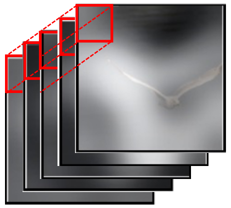
$0 \leq \sum_t^C \alpha_{ti} \leq 3$

$0 \leq |1 - \sum_t^C \alpha_{ti}| \leq 2$

$0 \leq (1 - \sum_t^C \alpha_{ti})^2 \leq 4$

Loss Function

encourage $\sum_t \alpha_{ti} \approx 1$



→ Sum over time = 1

$$L_d = -\log(P(\mathbf{y}|\mathbf{x})) + \lambda \underline{\sum_i^L (1 - \sum_t^C \alpha_{ti})^2}$$

$$e_{ti} = f_{\text{att}}(\mathbf{a}_i, \mathbf{h}_{t-1})$$

$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^L \exp(e_{tk})}$$

Softmax, so $\sum_i \alpha_{ti} = 1$

→ Sum over positions = 1

Practice: What is the max value of $\sum_i^L (1 - \sum_t^C \alpha_{ti})^2$?

α_t: α_1

0	1
0	0

α_2

0	1
0	0

α_3

0	1
0	0

$0 \leq \sum_t^C \alpha_{ti} \leq 3$

$0 \leq |1 - \sum_t^C \alpha_{ti}| \leq 2$

$0 \leq (1 - \sum_t^C \alpha_{ti})^2 \leq 4$

$1 - \sum_t^C \alpha_{ti}$

1	-2
1	1

$(1 - \sum_t^C \alpha_{ti})^2$

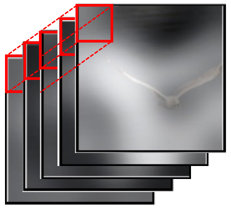
1	4
1	1

$\sum_i^L (1 - \sum_t^C \alpha_{ti})^2 = 7$

Loss Function

encourage $\sum_t \alpha_{ti} \approx 1$

$$L_d = -\log(P(\mathbf{y}|\mathbf{x})) + \underline{\lambda \sum_i^L (1 - \sum_t^C \alpha_{ti})^2}$$



→ Sum over time = 1

$$e_{ti} = f_{\text{att}}(\mathbf{a}_i, \mathbf{h}_{t-1})$$

$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^{L} \exp(e_{tk})}$$

Softmax, so $\sum_i \alpha_{ti} = 1$

→ Sum over positions = 1

Why do we need the second term?

- Encourage the model to ***pay equal attention*** to **every part** of the image over the course of generation
- This penalty was important quantitatively to improving overall ***BLEU score*** and that qualitatively this leads to more ***rich*** and ***descriptive*** captions

Experimental Results

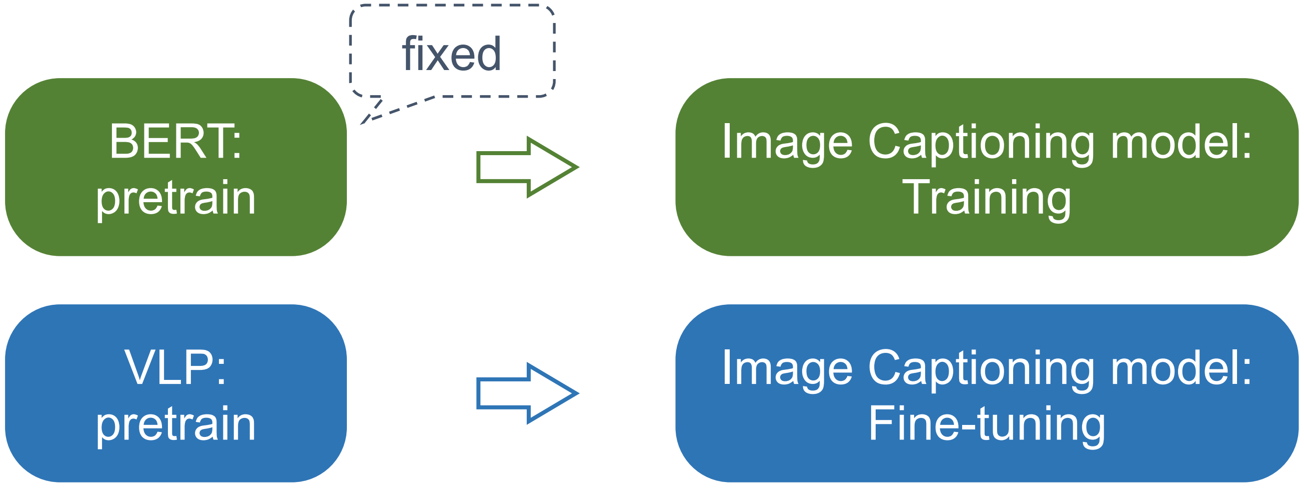
Dataset	Model	BLEU				
		BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR
Flickr8k	Google NIC(Vinyals et al., 2014)$^{\dagger\Sigma}$	63	41	27	—	—
	Log Bilinear (Kiros et al., 2014a)$^{\circ}$	65.6	42.4	27.7	17.7	17.31
	Soft-Attention	**67**	44.8	29.9	19.5	18.93
	Hard-Attention	**67**	**45.7**	**31.4**	**21.3**	**20.30**
Flickr30k	Google NIC$^{\dagger\circ\Sigma}$	66.3	42.3	27.7	18.3	—
	Log Bilinear	60.0	38	25.4	17.1	16.88
	Soft-Attention	66.7	43.4	28.8	19.1	**18.49**
	Hard-Attention	**66.9**	**43.9**	**29.6**	**19.9**	18.46
COCO	CMU/MS Research (Chen & Zitnick, 2014)^{a}	—	—	—	—	20.41
	MS Research (Fang et al., 2014)$^{\dagger a}$	—	—	—	—	20.71
	BRNN (Karpathy & Li, 2014)$^{\circ}$	64.2	45.1	30.4	20.3	—
	Google NIC$^{\dagger\circ\Sigma}$	66.6	46.1	32.9	24.6	—
	Log Bilinear$^{\circ}$	70.8	48.9	34.4	24.3	20.03
	Soft-Attention	70.7	49.2	34.4	24.3	**23.90**
	Hard-Attention	**71.8**	**50.4**	**35.7**	**25.0**	23.04

What’s new in Image Captioning?

- Vision-Language Pre-training (VLP)
- Object Anchors
- VIsual VOcabulary pre-training (VIVO)
- Generative Adversarial Networks (GAN)
- Deep Reinforcement Learning (RL) + Meta Learning

What’s new in Image Captioning?

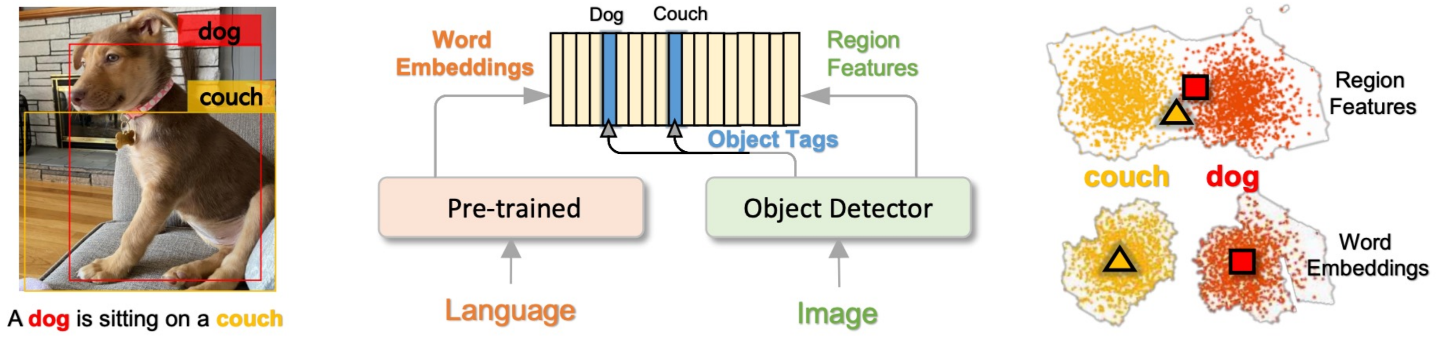
- Vision-Language Pre-training (VLP)
- Object Anchors
- VIsual VOcabulary pre-training (VIVO)
- Generative Adversarial Networks (GAN)
- Deep Reinforcement Learning (RL) + Meta Learning



End-to-end:
- Text embedding
- CNN backbone
- Encoder
- Decoder

What's new in Image Captioning?

- Vision-Language Pre-training (VLP)
- Object Anchors
- VIsual VOcabulary pre-training (VIVO)
- Generative Adversarial Networks (GAN)
- Deep Reinforcement Learning (RL) + Meta Learning

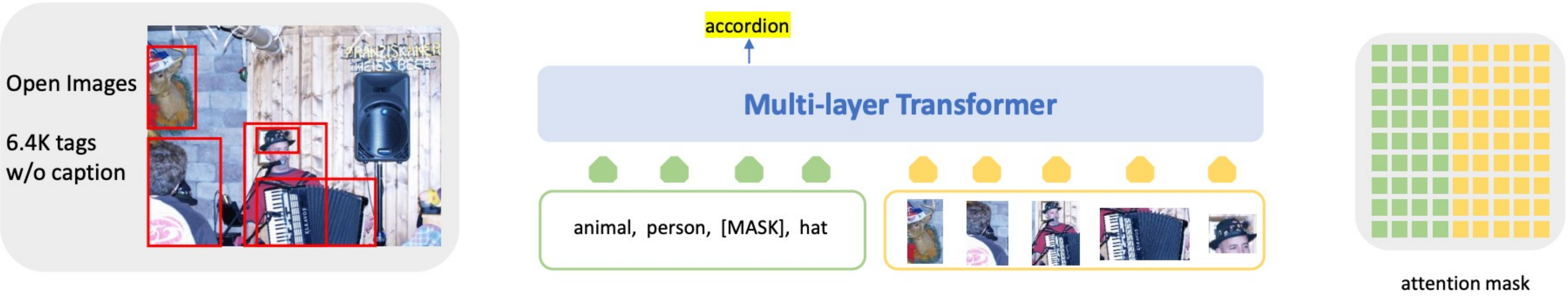


(a) Image-text pair (b) Objects as anchor points (c) Semantics spaces

What's new in Image Captioning?

- Vision-Language Pre-training (VLP)
- Object Anchors
- VIsual VOcabulary pre-training (VIVO)
- Generative Adversarial Networks (GAN)
- Deep Reinforcement Learning (RL) + Meta Learning

Some challenges require models trained without other image captioning dataset



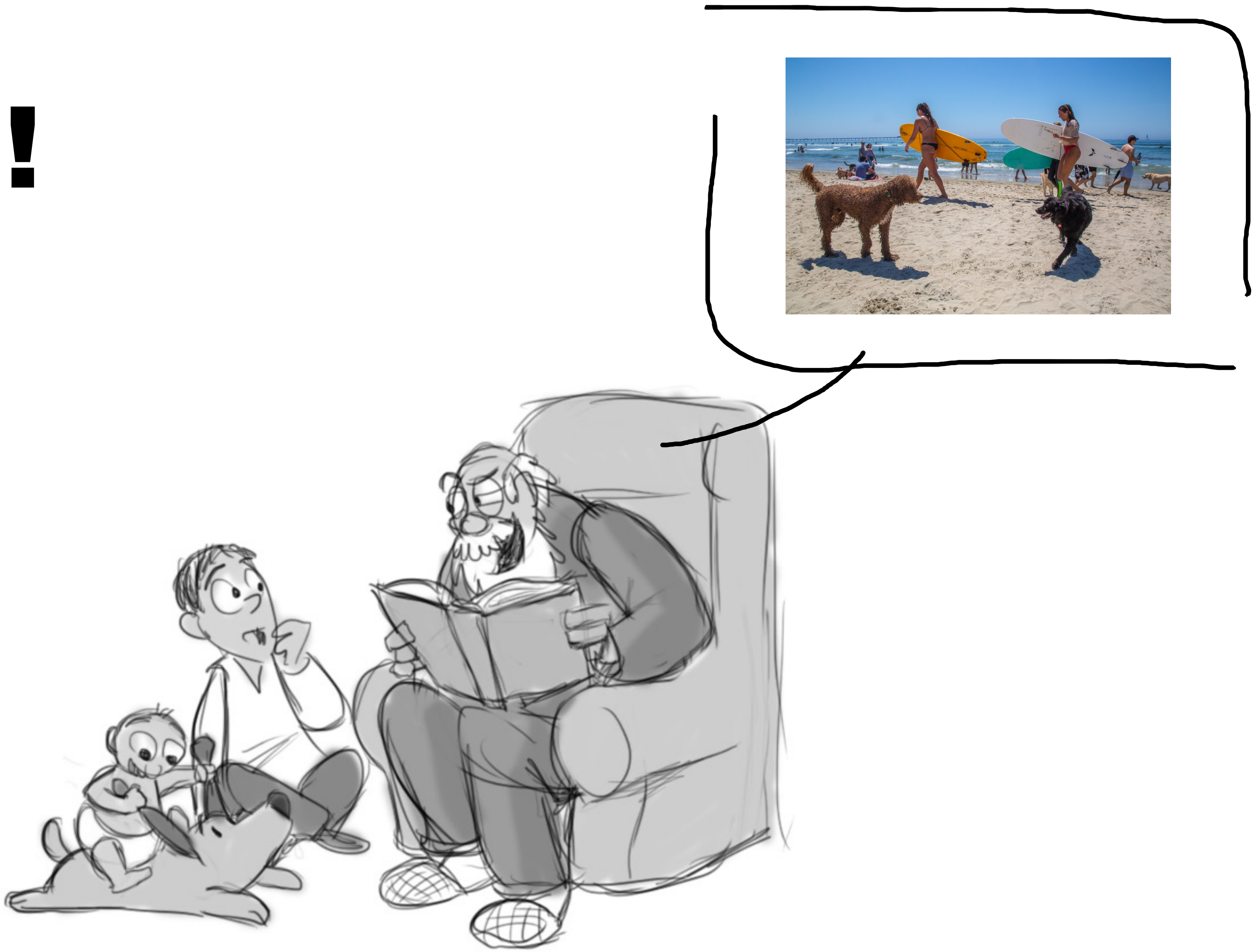
What’s new in Image Captioning?

- Vision-Language Pre-training (VLP)
- Object Anchors
- VIsual VOcabulary pre-training (VIVO)
- Generative Adversarial Networks (GAN)
- Deep Reinforcement Learning (RL) + Meta Learning

What’s new in Image Captioning?

- Vision-Language Pre-training (VLP)
- Object Anchors
- VIsual VOcabulary pre-training (VIVO)
- Generative Adversarial Networks (GAN)
- Deep Reinforcement Learning (RL) + Meta Learning

Thank you!



References

- Papers
  - https://arxiv.org/abs/1411.4555
  - https://arxiv.org/abs/1502.03044
- Others
  - https://becominghuman.ai/only-numpy-deriving-partial-forward-feed-lstm-on-show-attend-and-tell-neural-image-caption-4e44aa2b966d
  - https://www.youtube.com/watch?v=y1S3Ri7myMg
  - https://cv-tricks.com/artificial-intelligence/show-attend-tell-image-captioning-explained/

Image Sources

- https://www.sandiegouniontribune.com/business/real-estate/story/2020-08-21/will-san-diego-stay-at-home-workers-leave-if-given-the-opportunity
- https://medium.com/swlh/image-caption-generation-with-visual-attention-c782dfc0634b
- https://towardsdatascience.com/time-series-forecasting-with-deep-stacked-unidirectional-and-bidirectional-lstms-de7c099bd918
- https://www.usna.edu/Users/cs/nchamber/courses/nlp/f20/labs/lab5/index.html
- https://medium.com/intelligentmachines/word-embedding-and-one-hot-encoding-ad17b4bbe111
- https://medium.com/mlearning-ai/self-attention-in-convolutional-neural-networks-172d947afc00
- https://www.ecva.net/papers/eccv_2020/papers_ECCV/papers/123750120.pdf
- https://arxiv.org/pdf/2009.13682.pdf
- https://www.animatorisland.com/story-101-why-do-we-tell-stories/