

# CS194/294-129: Designing, Visualizing and Understanding Deep Neural Networks

**John Canny**

Spring 2018

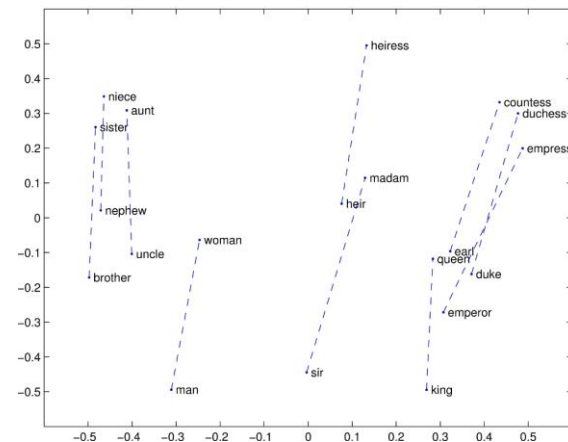
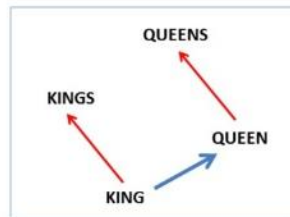
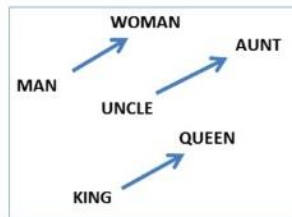
Lecture 13: Attention

# Last Time: Word Embeddings



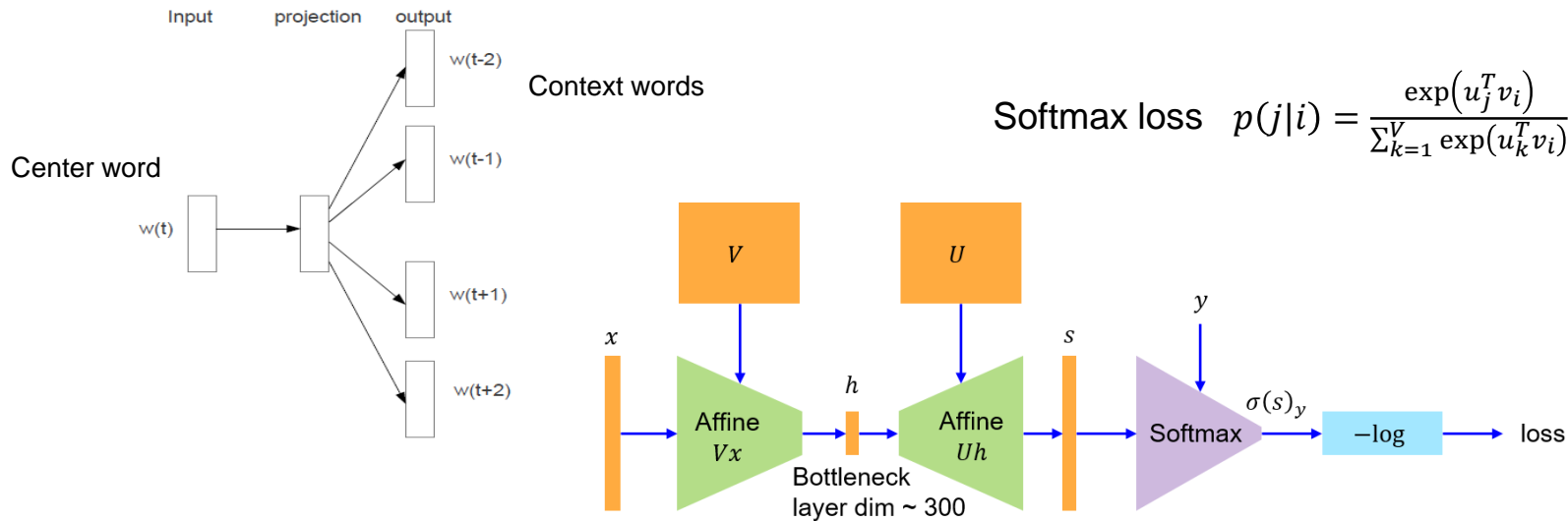
t-SNE of  
word vectors

Relations among  
word embeddings



# Last Time: Word2vec: Local context

The pairs of center word/context word are called “**skip-grams.**” Typical distances are 3-5 word positions. Skip-gram model:



Word2vec as a deep network. Input is (x, y) (center, context) word pairs.

# Last Time: GloVe: Word embedding for analogies

Let  $C_{ij}$  denote the number of times that word  $j$  occurs in the context of word  $i$ .

Glove loss is:

$$J(\theta) = \sum_{i,j=1}^V f(C_{ij})(u_i^T v_j + b_i + \tilde{b}_j - \log C_{ij})^2$$

A sensible choice of  $f(\cdot)$  is :  $f(x) = \begin{cases} (x/x_{\max})^\alpha & \text{if } x < x_{\max} \\ 1 & \text{otherwise} \end{cases}$  . typical  $\alpha = 3/4$ ,  $x_{\max}=100$

Nearest words to

frog:

1. frogs
2. toad
3. litoria
4. leptodactylidae
5. rana
6. lizard
7. eleutherodactylus



litoria



leptodactylidae



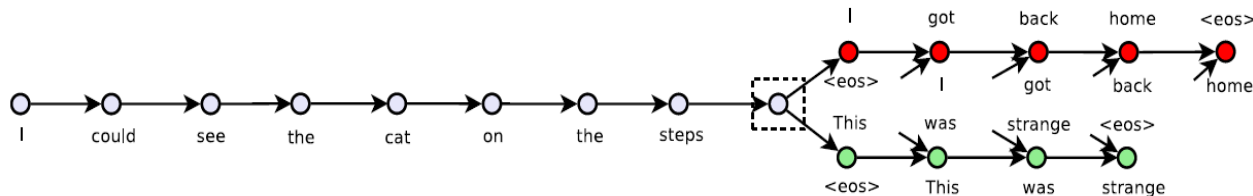
rana



eleutherodactylus

# Last Time: Skip-Thought Vectors

Skip-thought embeddings use sequence-to-sequence RNNs to predict the next and previous *sentences*.



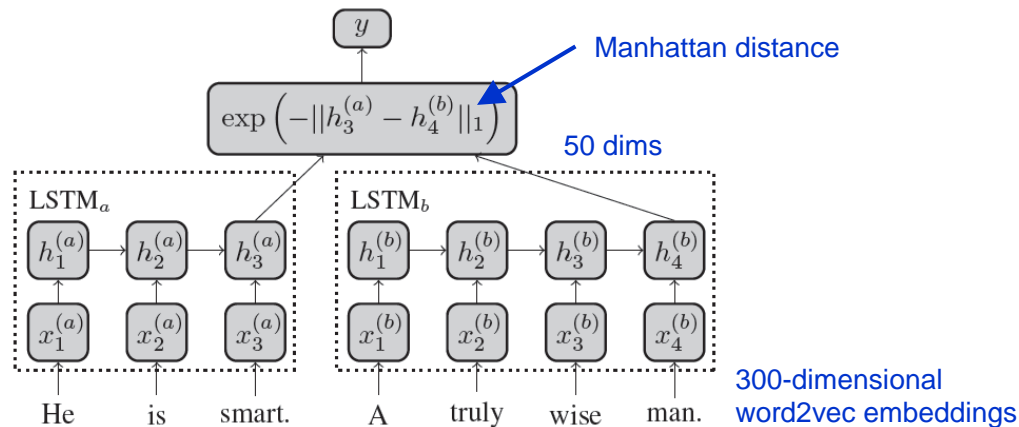
The output state vector of the boundary layer (dotted box) forms the embedding. RNN units are GRU units.

Once the network is trained, we discard the red and green sections of the network, and use the white section to embed new sentences.

From "Skip-Thought Vectors," Ryan Kiros et al., Arxiv 2015.

# Last Time: Siamese Networks for Semantic Relatedness

This network is trained on pairs of sentences a, b with a similarity label y.



Parameters are shared between the two networks.

From "Siamese Recurrent Architectures for Learning Sentence Similarity" Jonas Mueller, Aditya Thyagarajan, AAAI-2016

# This Time: Attention

Defn: “the regarding of someone or something as interesting or important.”

Attention is one of the most important ideas in deep networks in the last decade...

It cross-cuts computer vision, NLP, speech, RL,...





# Early attention models

Larochelle and Hinton, 2010, “Learning to combine foveal glimpses with a third-order Boltzmann machine”

Misha Denil et al, 2011, “Learning where to Attend with Deep Architectures for Image Tracking”

# 2014: Neural Translation Breakthroughs

- Devlin et al, ACL'2014
- Cho et al EMNLP'2014
- Bahdanau, Cho & Bengio, arXiv sept. 2014
- Jean, Cho, Memisevic & Bengio, arXiv dec. 2014
- Sutskever et al NIPS'2014

# Other Applications

- Ba et al 2014, **Visual attention for recognition**
- Mnih et al 2014, **Visual attention for recognition**
- Chorowski et al, 2014, **Speech recognition**
- Graves et al 2014, **Neural Turing machines**
- Yao et al 2015, **Video description generation**
- Vinyals et al, 2015, **Conversational Agents**
- Xu et al 2015, **Image caption generation**
- Xu et al 2015, **Visual Question Answering**

# Soft vs Hard Attention Models

## Hard attention:

Attend to a single input location.

Can't use gradient descent.

Need **reinforcement learning**.

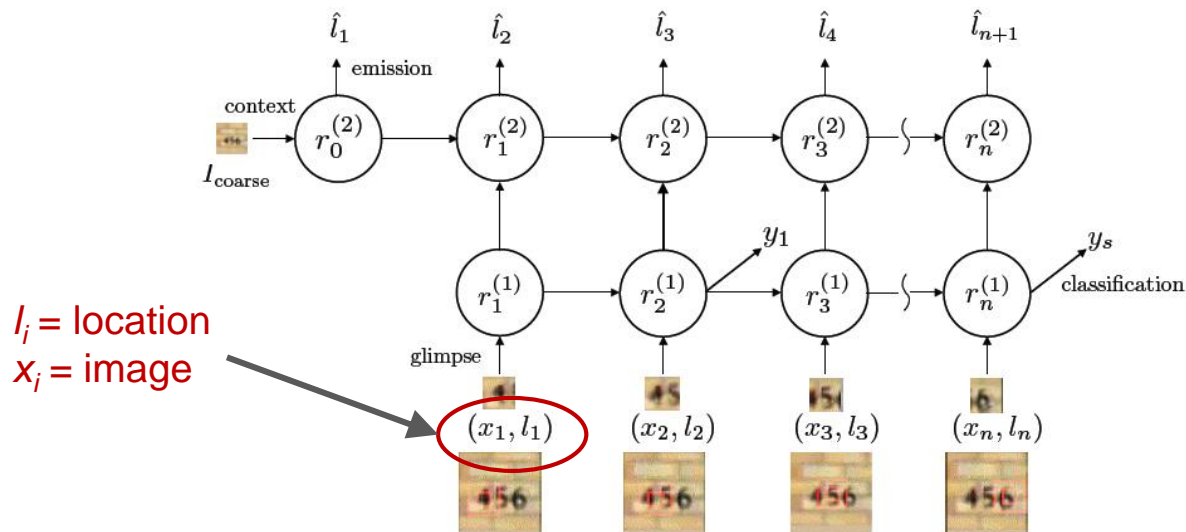
## Soft attention:

Compute a weighted combination (attention) over some inputs using an attention network.

Can use backpropagation to train end-to-end.

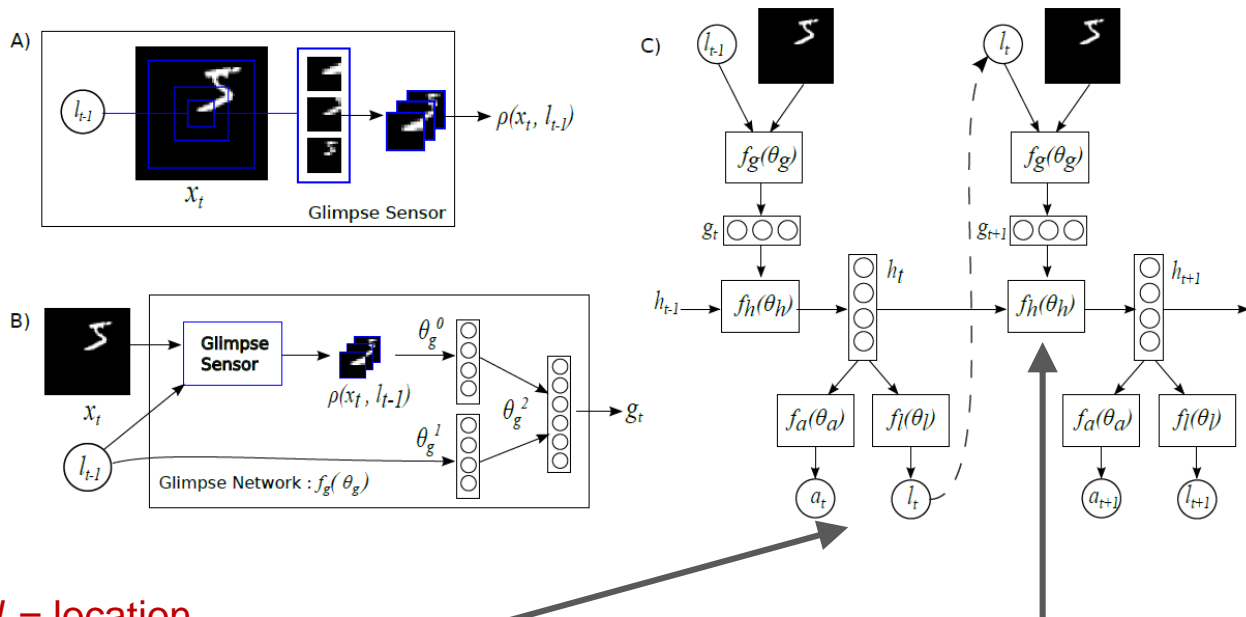
# Attention for Recognition (Ba et al 2014)

- RNN-based model.
- Hard attention.
- Required reinforcement learning.



# Attention for Recognition (Mnih et al 2014)

- Glimpses are retinal (graded resolution) images

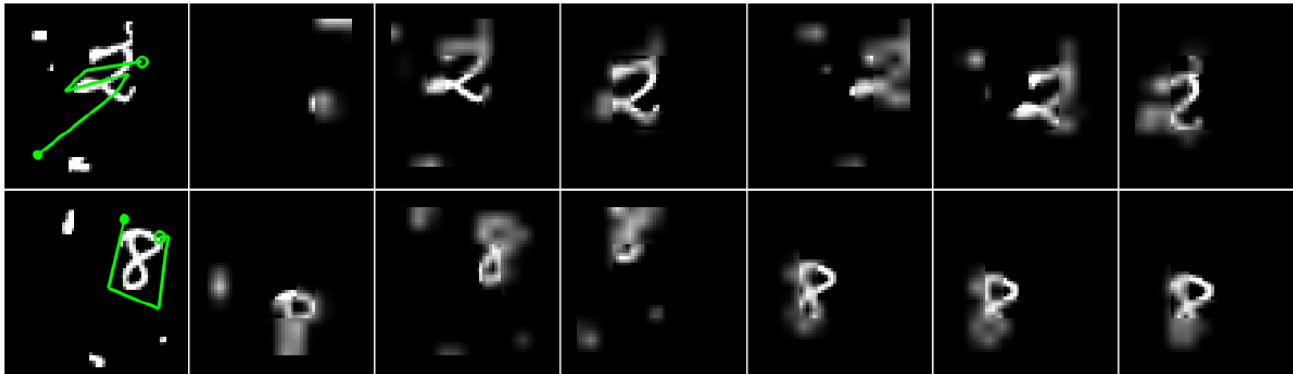


$l_i$  = location  
 $a_i$  = action (classification)

$f_h()$  = 256-dimensional LSTM

# Attention for Recognition (Mnih et al 2014)

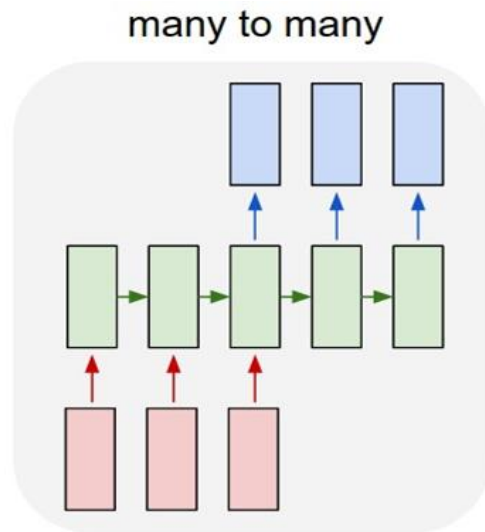
- Glimpse trace on some digit images:
- Green line shows trajectory, other images are the glimpses themselves.



# Soft Attention for Translation

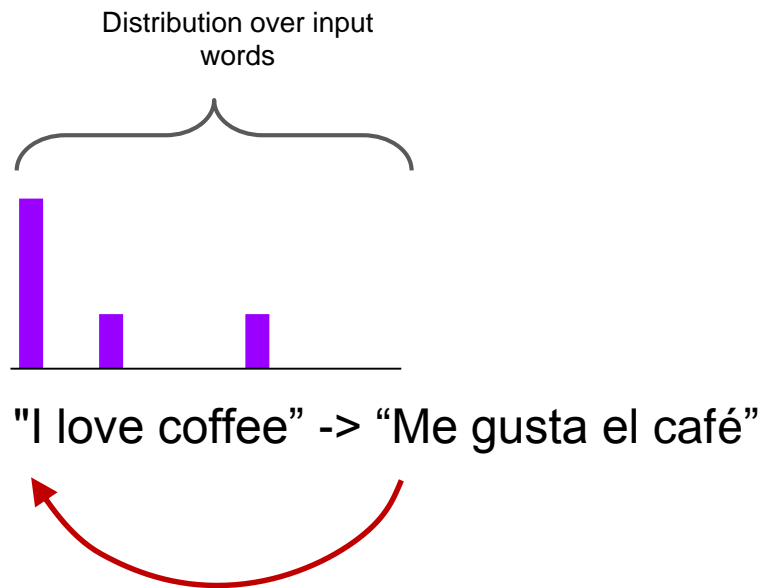
“I love coffee” -> “Me gusta el café”

Bahdanau et al, “Neural Machine Translation by  
Jointly Learning to Align and Translate”, ICLR 2015

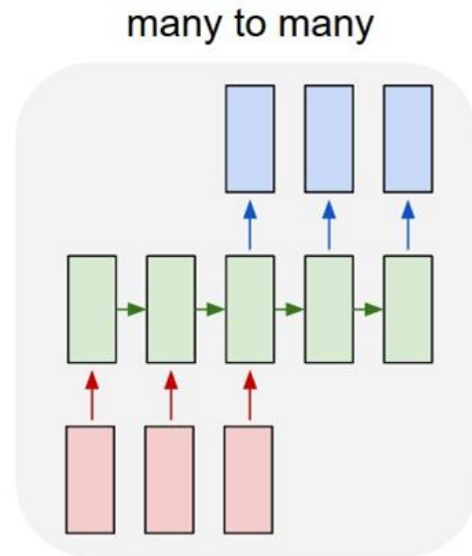




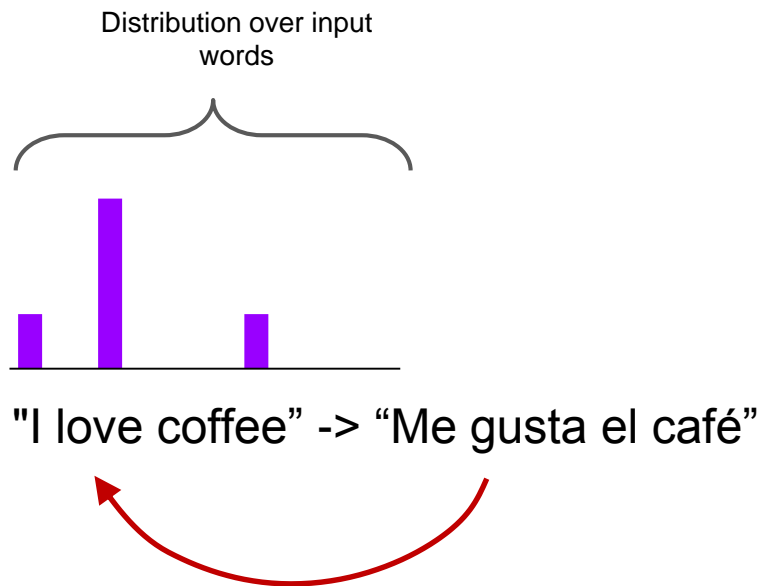
# Soft Attention for Translation



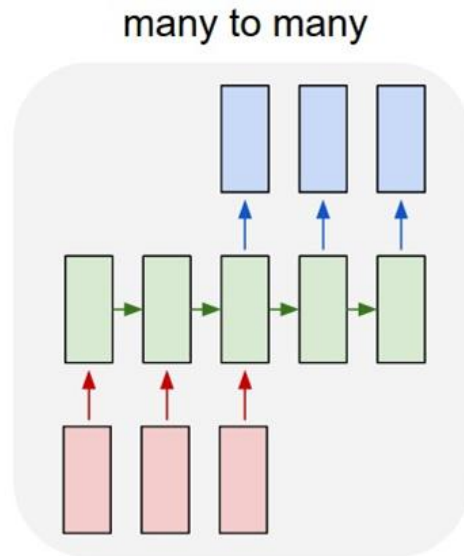
Bahdanau et al, "Neural Machine Translation by Jointly Learning to Align and Translate", ICLR 2015



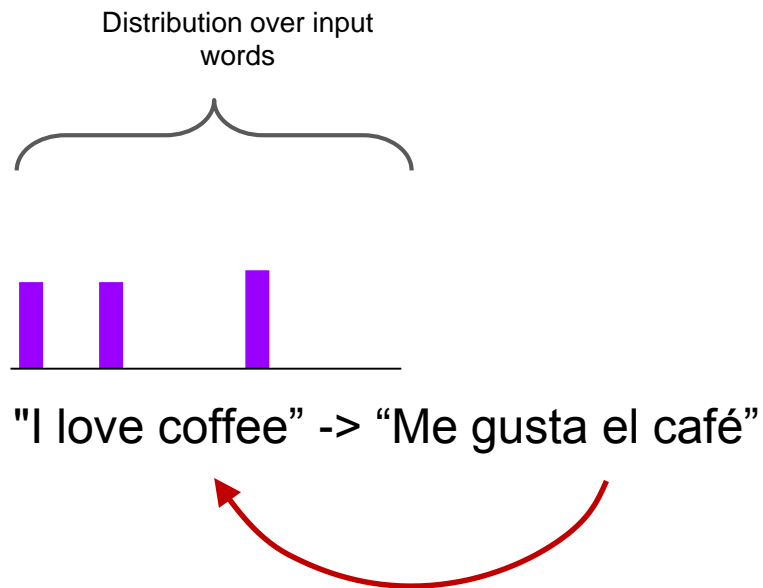
# Soft Attention for Translation



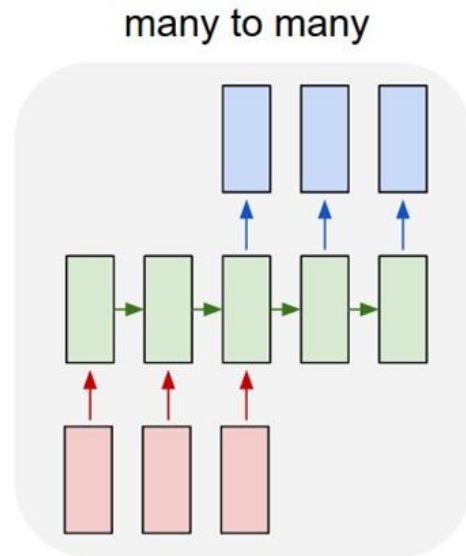
Bahdanau et al, "Neural Machine Translation by Jointly Learning to Align and Translate", ICLR 2015



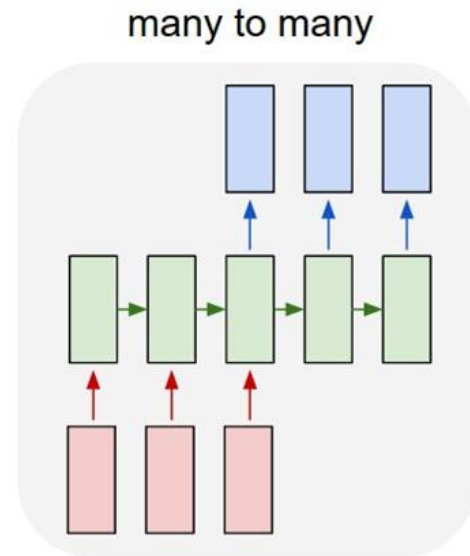
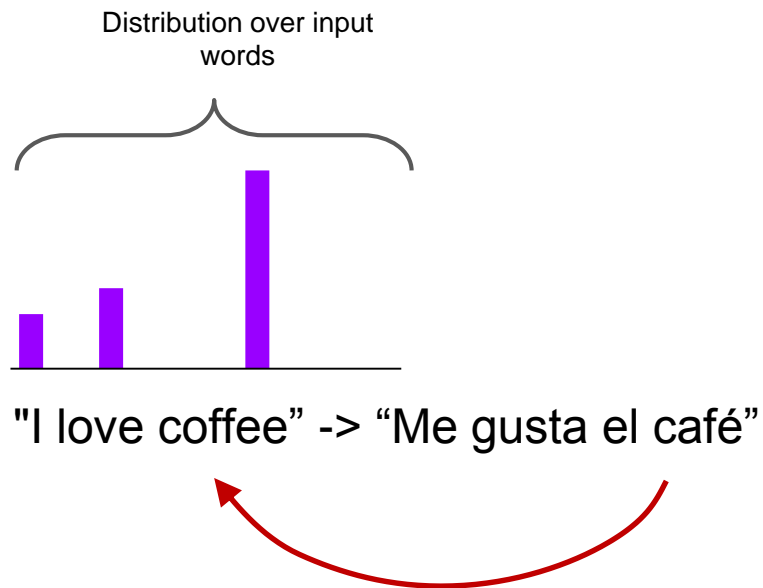
# Soft Attention for Translation



Bahdanau et al, "Neural Machine Translation by Jointly Learning to Align and Translate", ICLR 2015

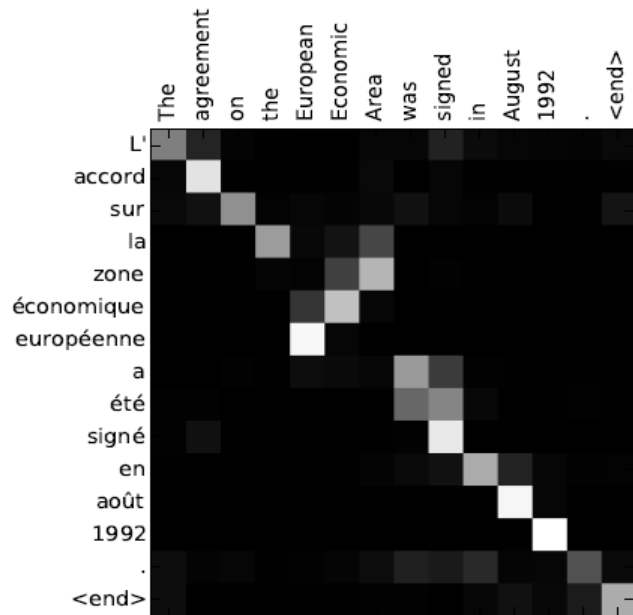


# Soft Attention for Translation

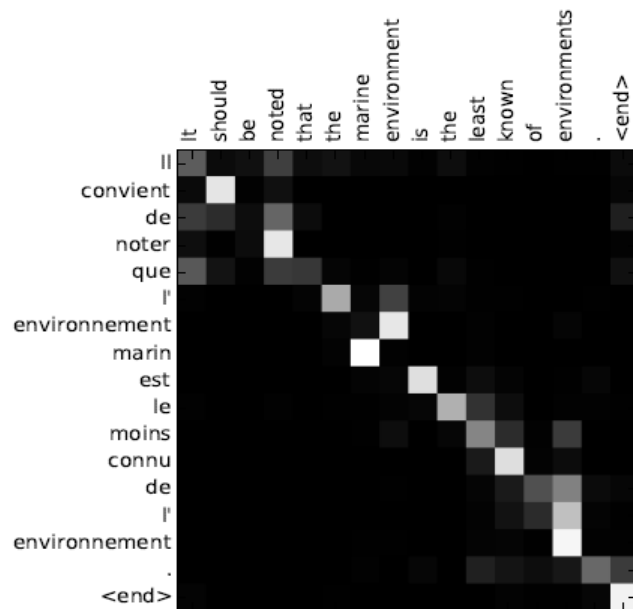


Bahdanau et al, "Neural Machine Translation by  
Jointly Learning to Align and Translate", ICLR 2015

# Soft Attention for Translation



(a)



(b)

Bahdanau et al, "Neural Machine Translation by Jointly Learning to Align and Translate", ICLR 2015

# Soft Attention for Translation

Reached State of the art in one year:

(a) English→French (WMT-14)

	NMT(A)	Google	P-SMT
NMT	32.68	30.6*	37.03*
+Cand	33.28	—	
+UNK	33.99	32.7°	
+Ens	36.71	36.9°	

(b) English→German (WMT-15)

Model	Note
24.8	Neural MT
24.0	U.Edinburgh, Syntactic SMT
23.6	LIMS/KIT
22.8	U.Edinburgh, Phrase SMT
22.7	KIT, Phrase SMT

(c) English→Czech (WMT-15)

Model	Note
18.3	Neural MT
18.2	JHU, SMT+LM+OSM+Sparse
17.6	CU, Phrase SMT
17.4	U.Edinburgh, Phrase SMT
16.1	U.Edinburgh, Syntactic SMT

# RNN for Captioning



Image:  
 $H \times W \times 3$

# RNN for Captioning



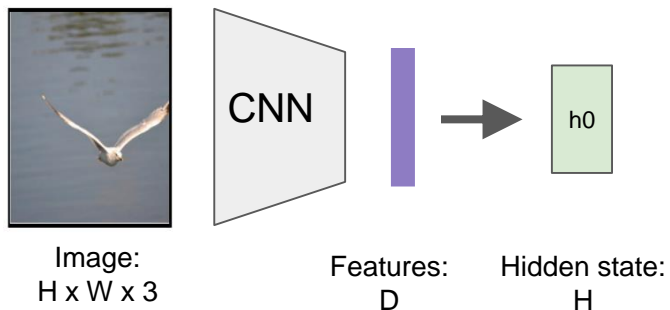
Image:  
 $H \times W \times 3$



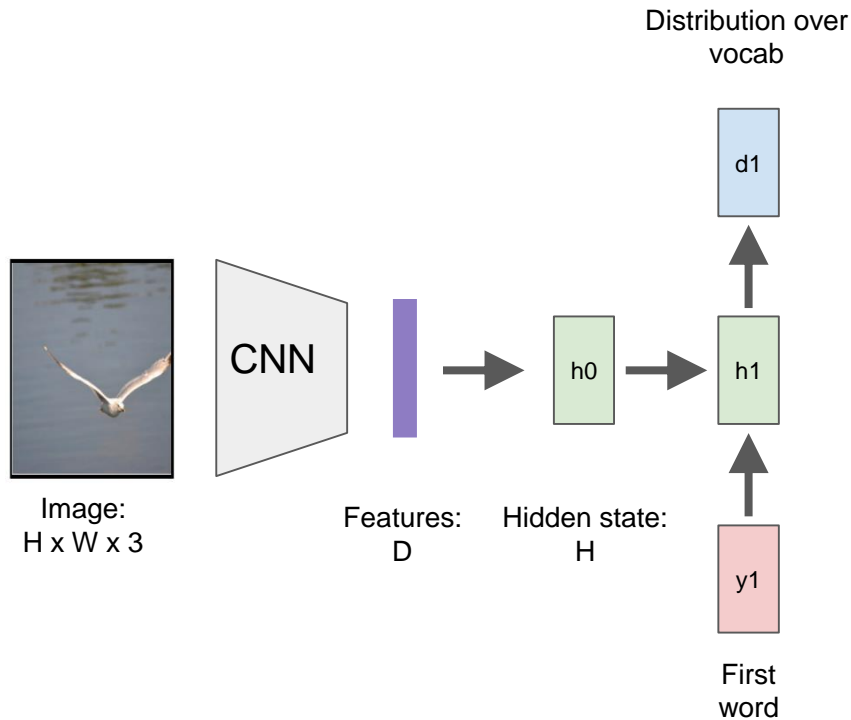
Features:  
 $D$



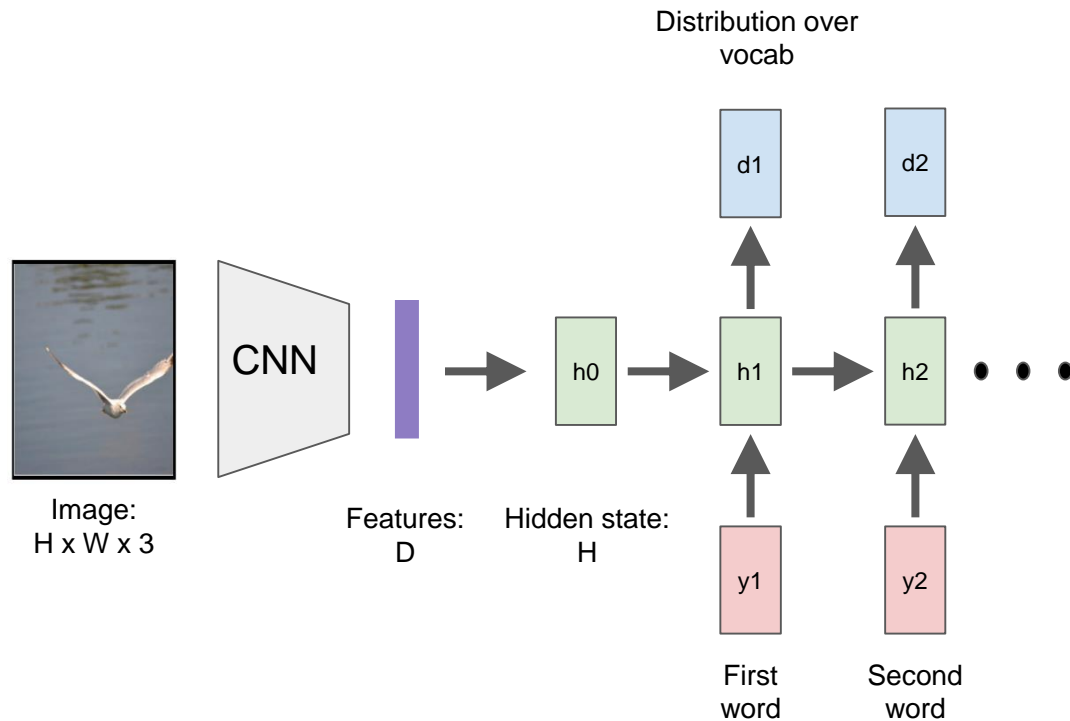
# RNN for Captioning



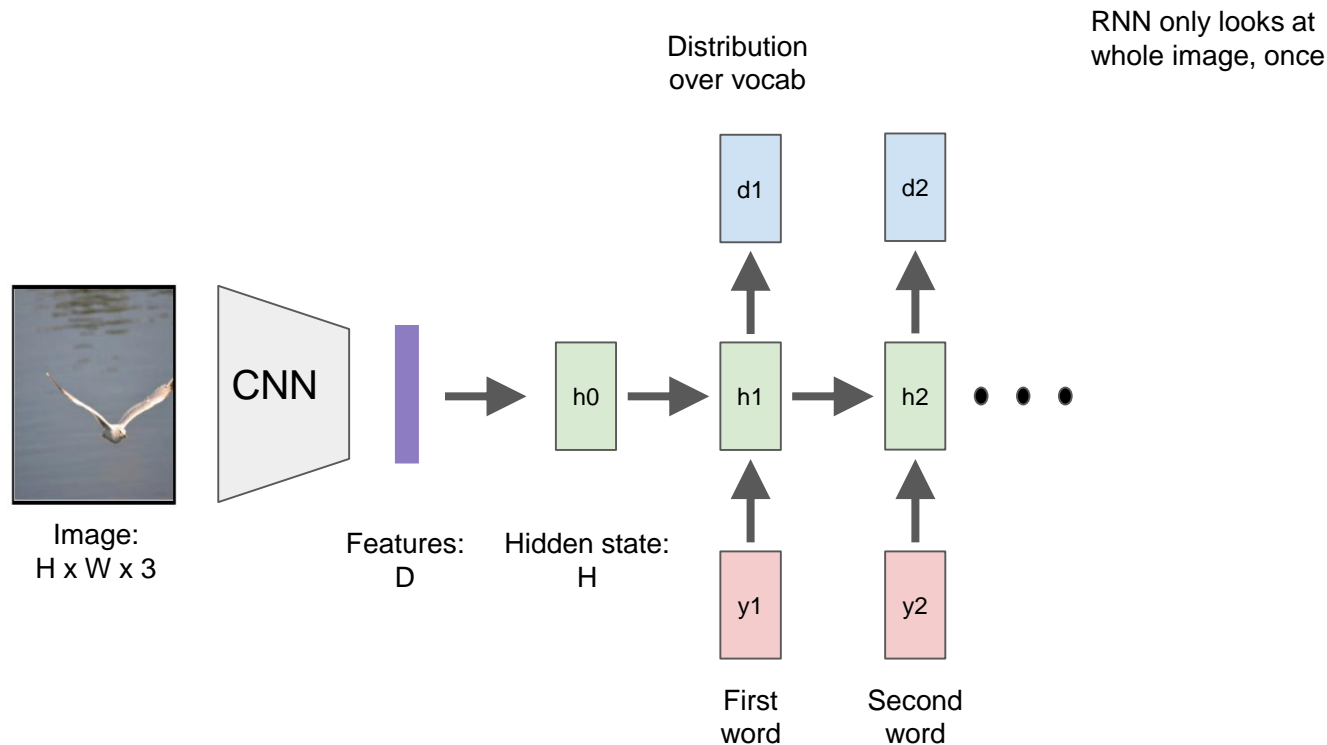
# RNN for Captioning



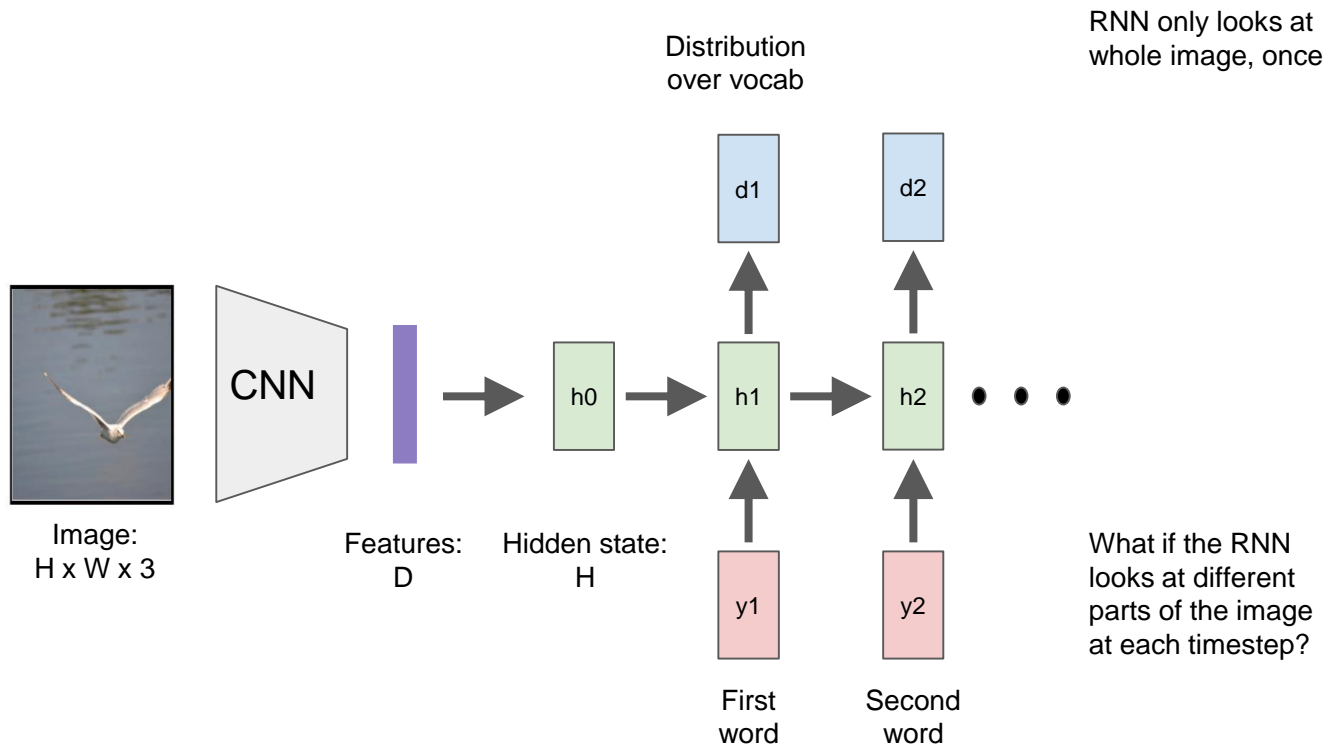
# RNN for Captioning



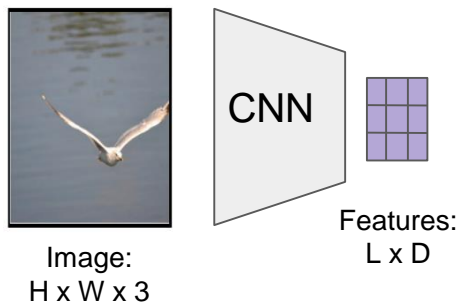
# RNN for Captioning



# RNN for Captioning

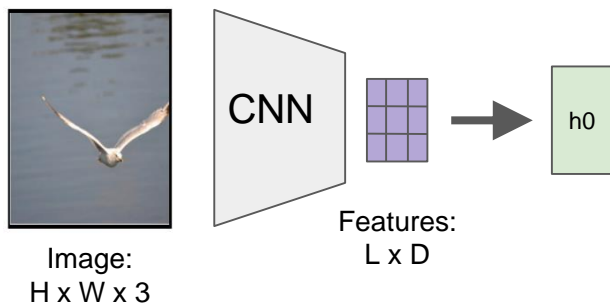


# RNN for Captioning



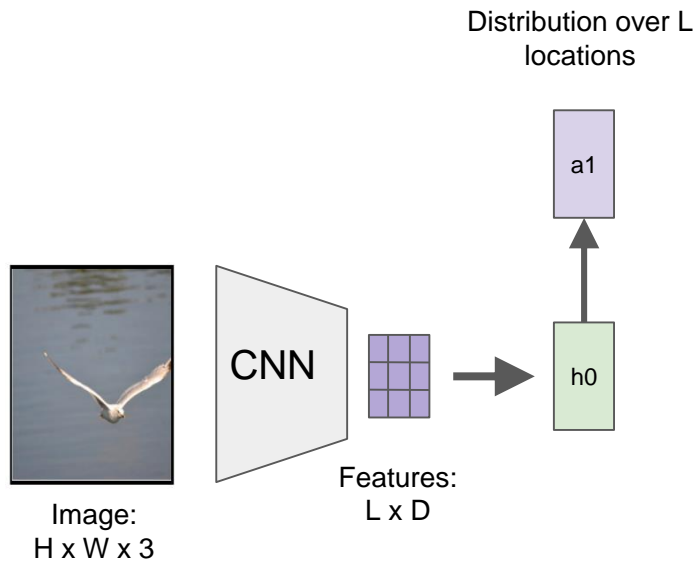
Xu et al, "Show, Attend and Tell:  
Neural Image Caption Generation  
with Visual Attention", ICML 2015

# RNN for Captioning



Xu et al, "Show, Attend and Tell:  
Neural Image Caption Generation  
with Visual Attention", ICML 2015

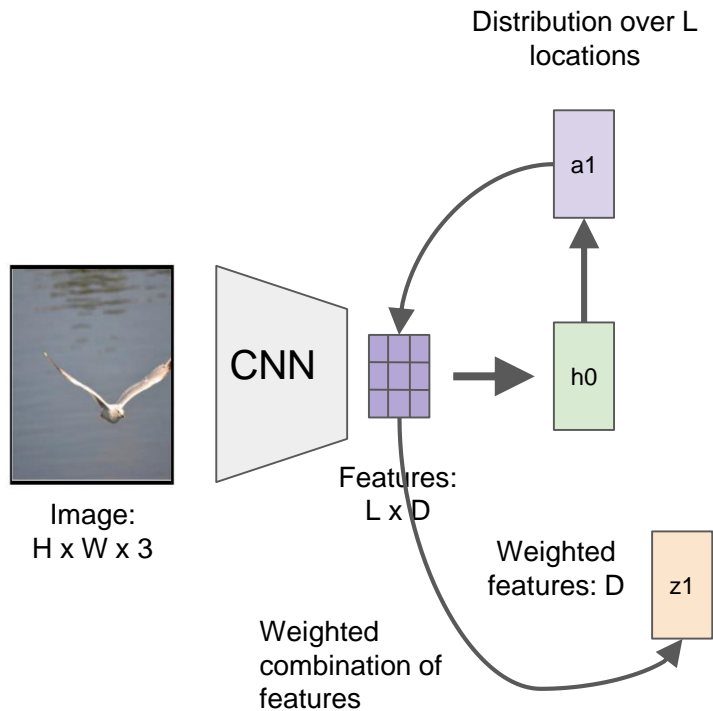
# RNN for Captioning



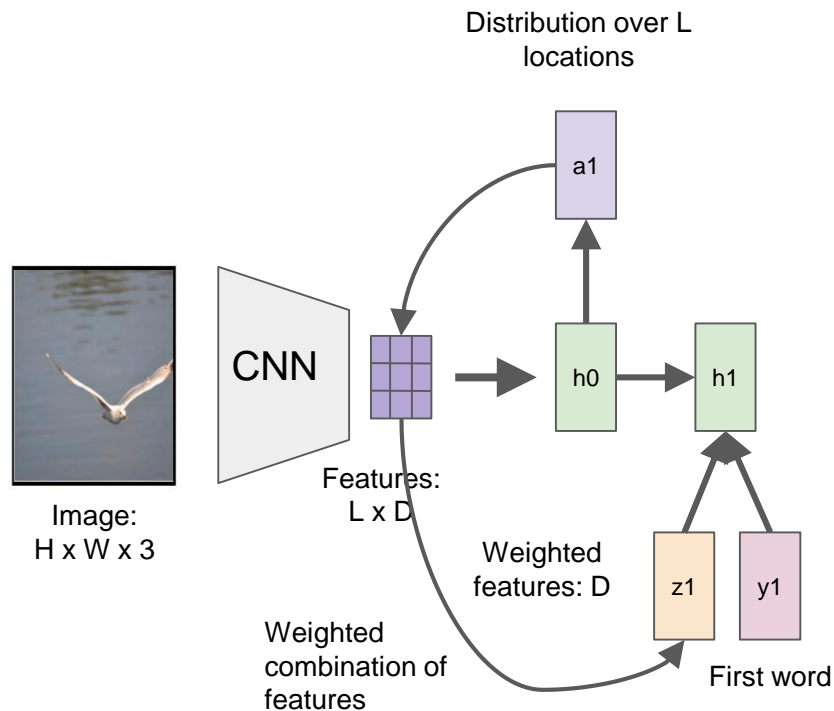
Xu et al, "Show, Attend and Tell:  
Neural Image Caption Generation  
with Visual Attention", ICML 2015



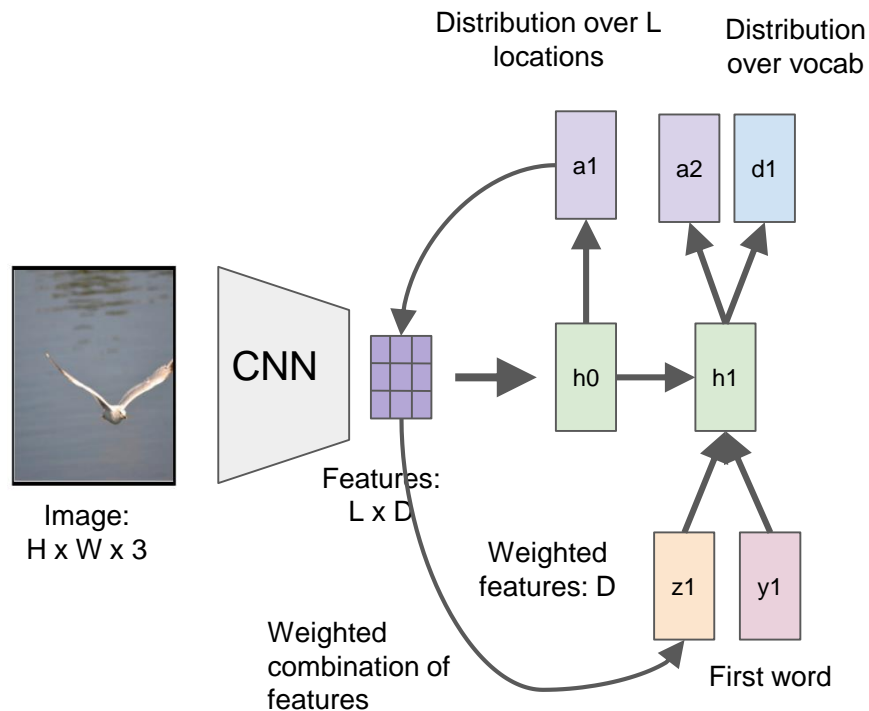
# RNN for Captioning



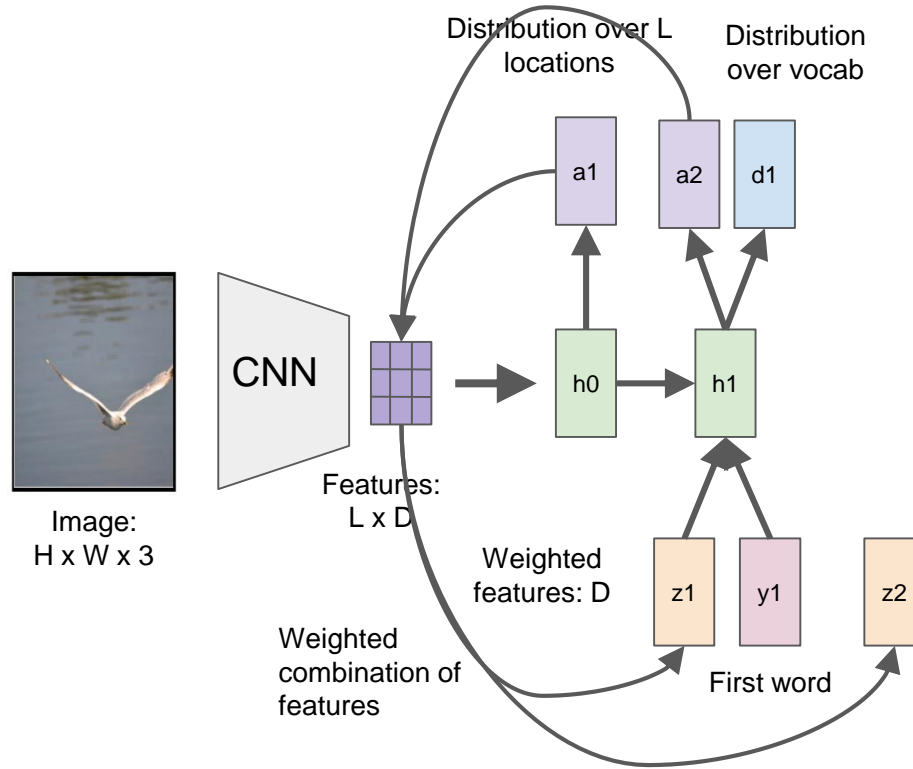
# RNN for Captioning



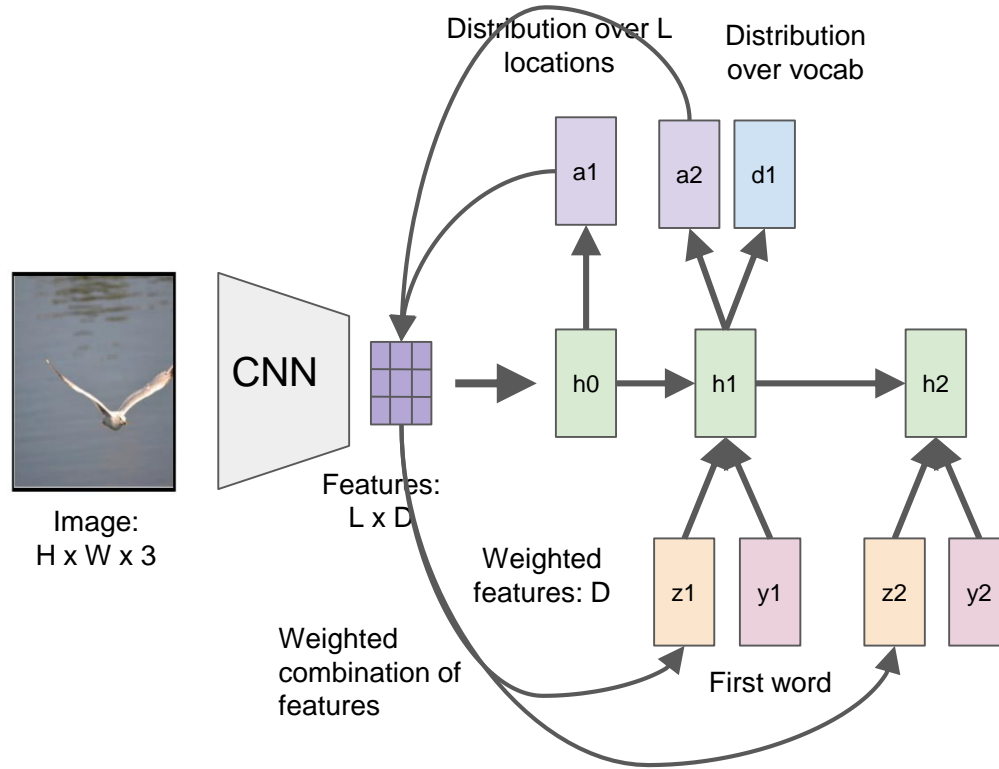
# RNN for Captioning



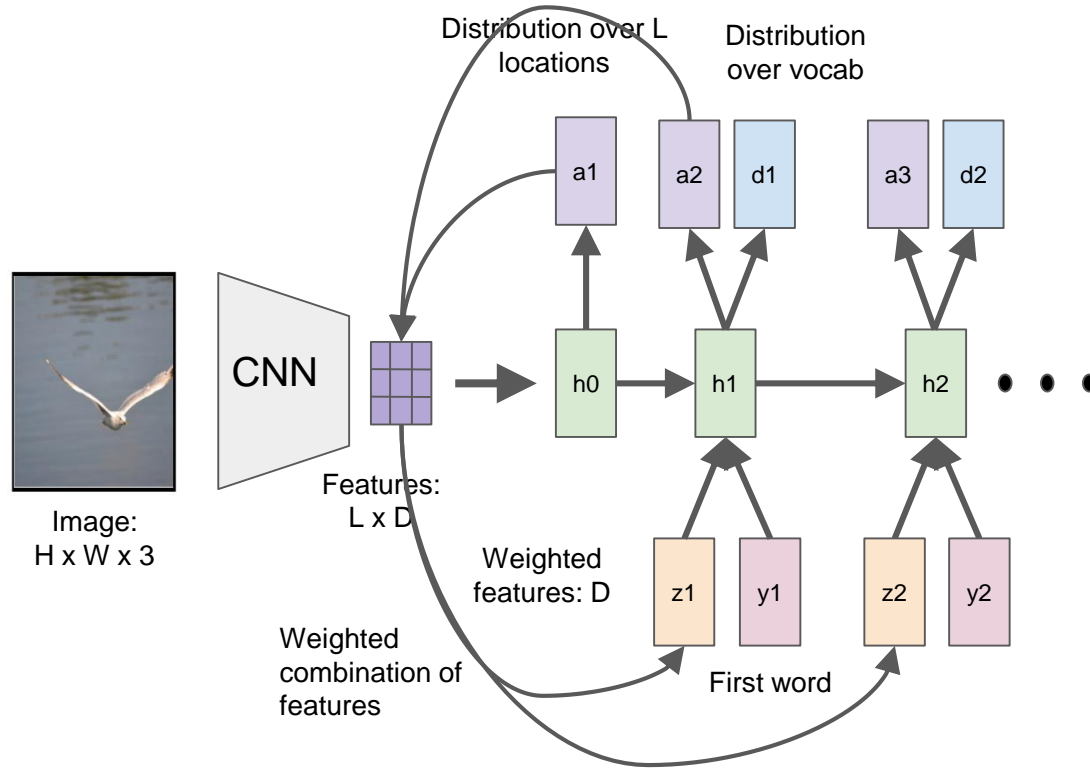
# RNN for Captioning



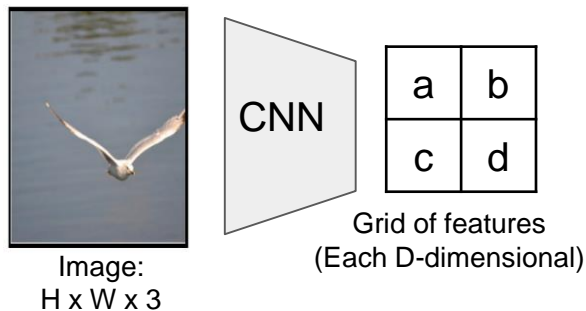
# RNN for Captioning



# RNN for Captioning



# Soft vs. Hard Attention



From  
RNN:



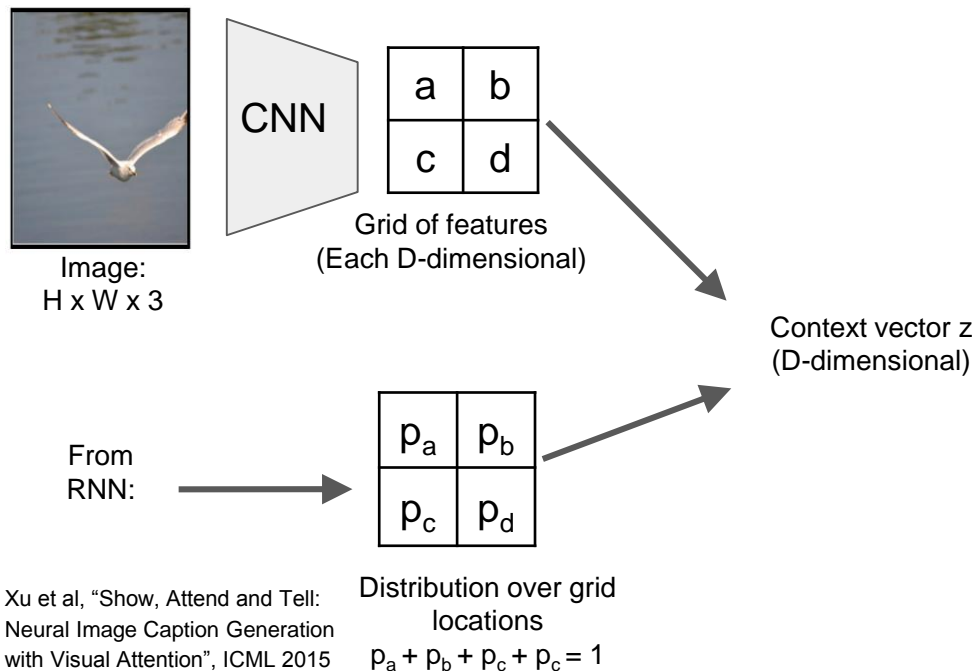
$p_a$	$p_b$
$p_c$	$p_d$

Xu et al, "Show, Attend and Tell:  
Neural Image Caption Generation  
with Visual Attention", ICML 2015

Distribution over grid  
locations

$$p_a + p_b + p_c + p_d = 1$$

# Soft vs. Hard Attention

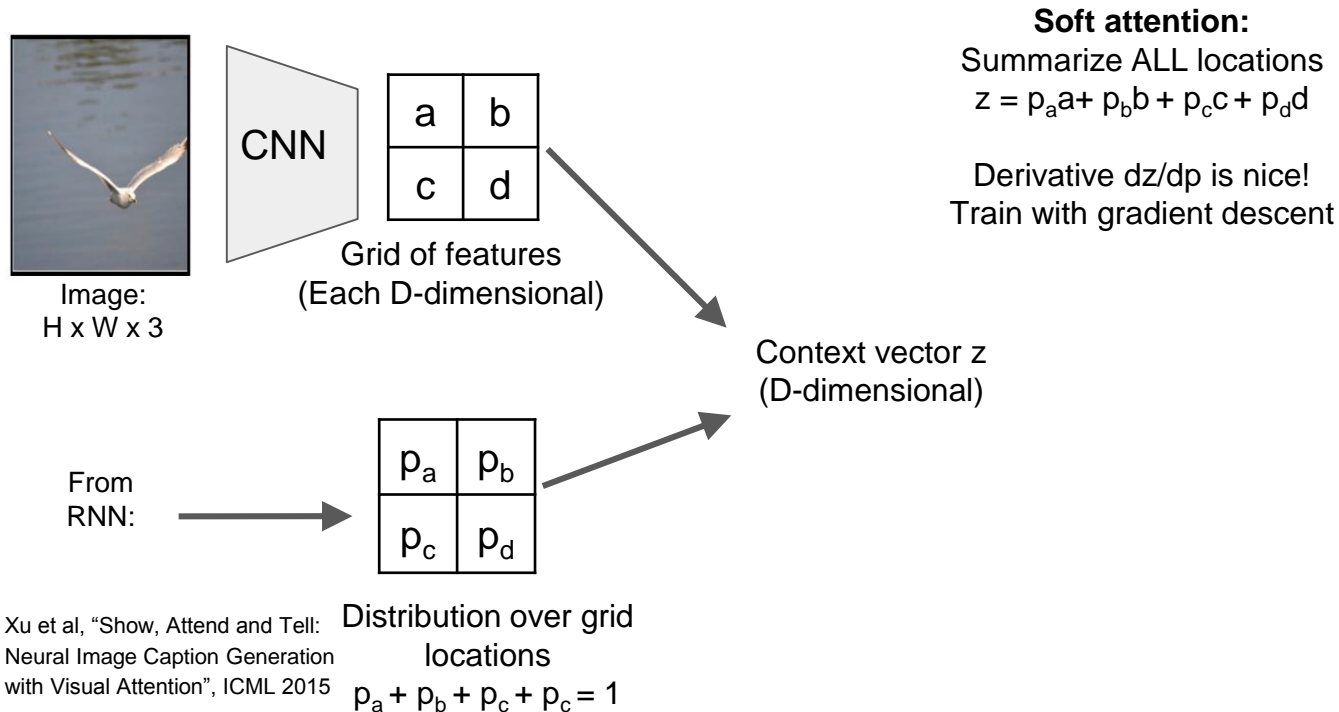


Xu et al, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention", ICML 2015

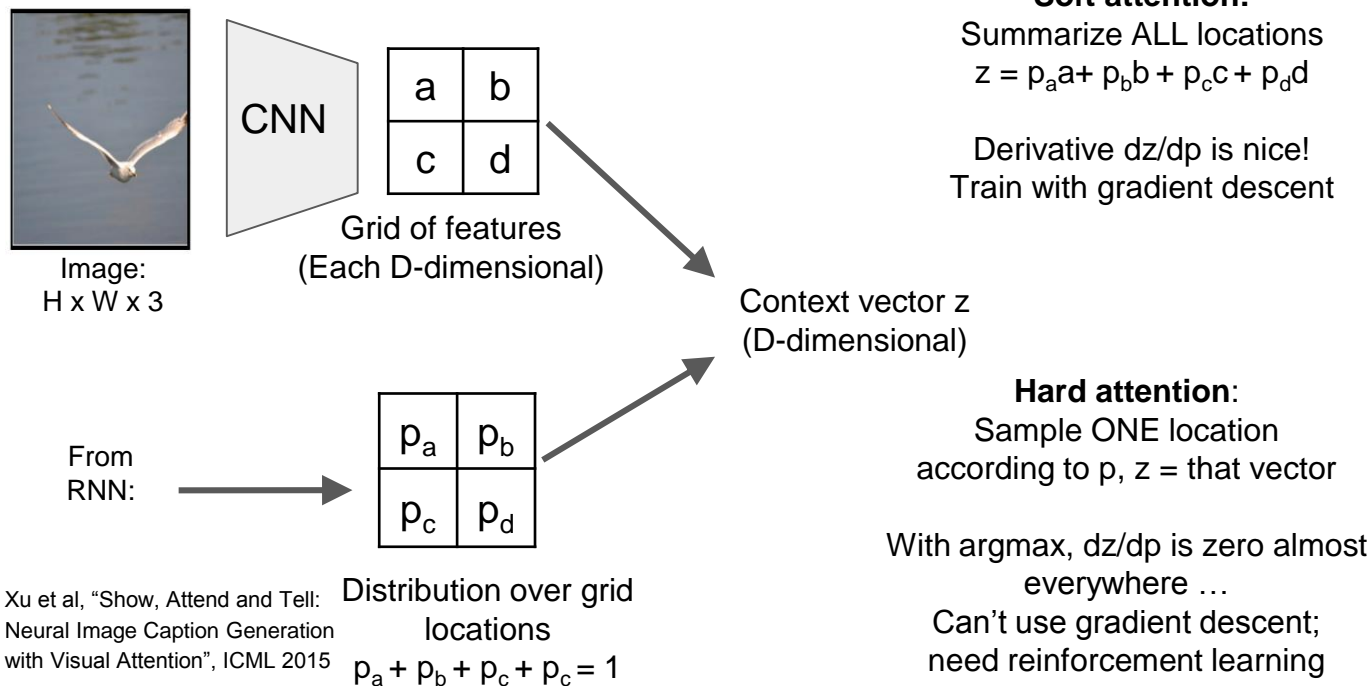
Distribution over grid locations  
 $p_a + p_b + p_c + p_d = 1$



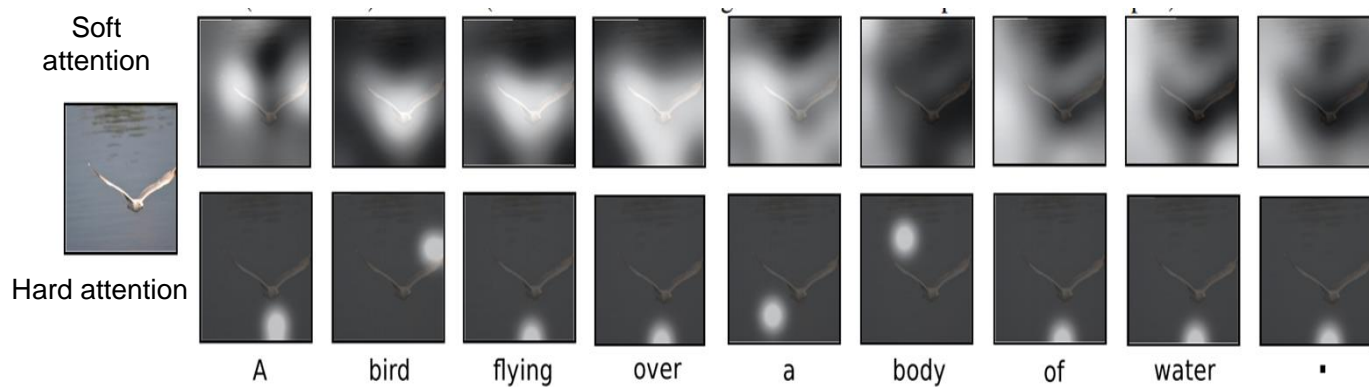
# Soft vs. Hard Attention



# Soft vs. Hard Attention



# Soft Attention for Captioning



# Soft Attention for Captioning



A woman is throwing a frisbee in a park.



A dog is standing on a hardwood floor.



A stop sign is on a road with a mountain in the background.



A little girl sitting on a bed with a teddy bear.



A group of people sitting on a boat in the water.



A giraffe standing in a forest with trees in the background.

Xu et al, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention", ICML 2015

# Soft Attention for Diagnosis

Figure 5. Examples of mistakes where we can use attention to gain intuition into what the model saw.



A large white bird standing in a forest.



A woman holding a clock in her hand.



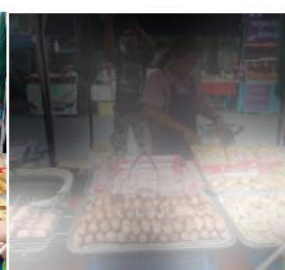
A man wearing a hat and a hat on a skateboard.



A person is standing on a beach with a surfboard.



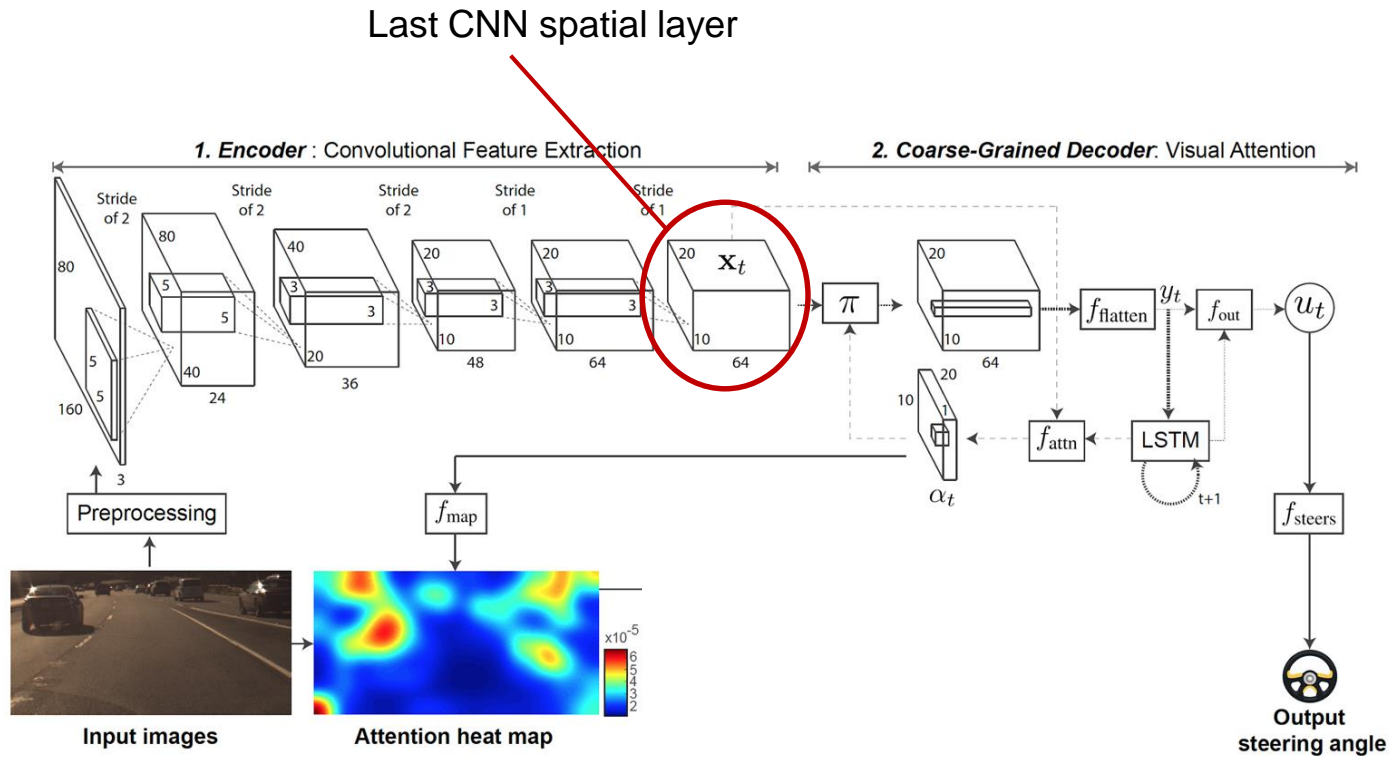
A woman is sitting at a table with a large pizza.



A man is talking on his cell phone while another man watches.

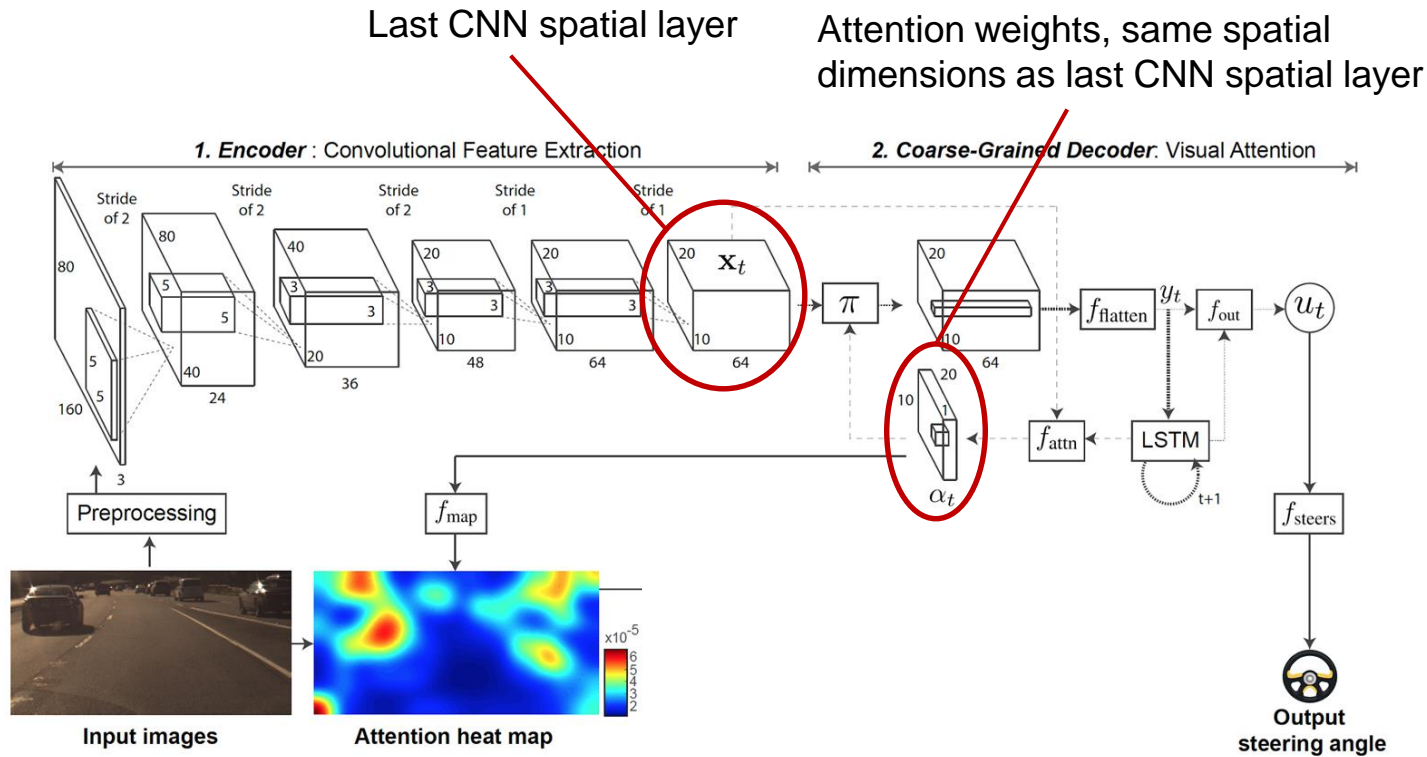


# Attention Mechanics



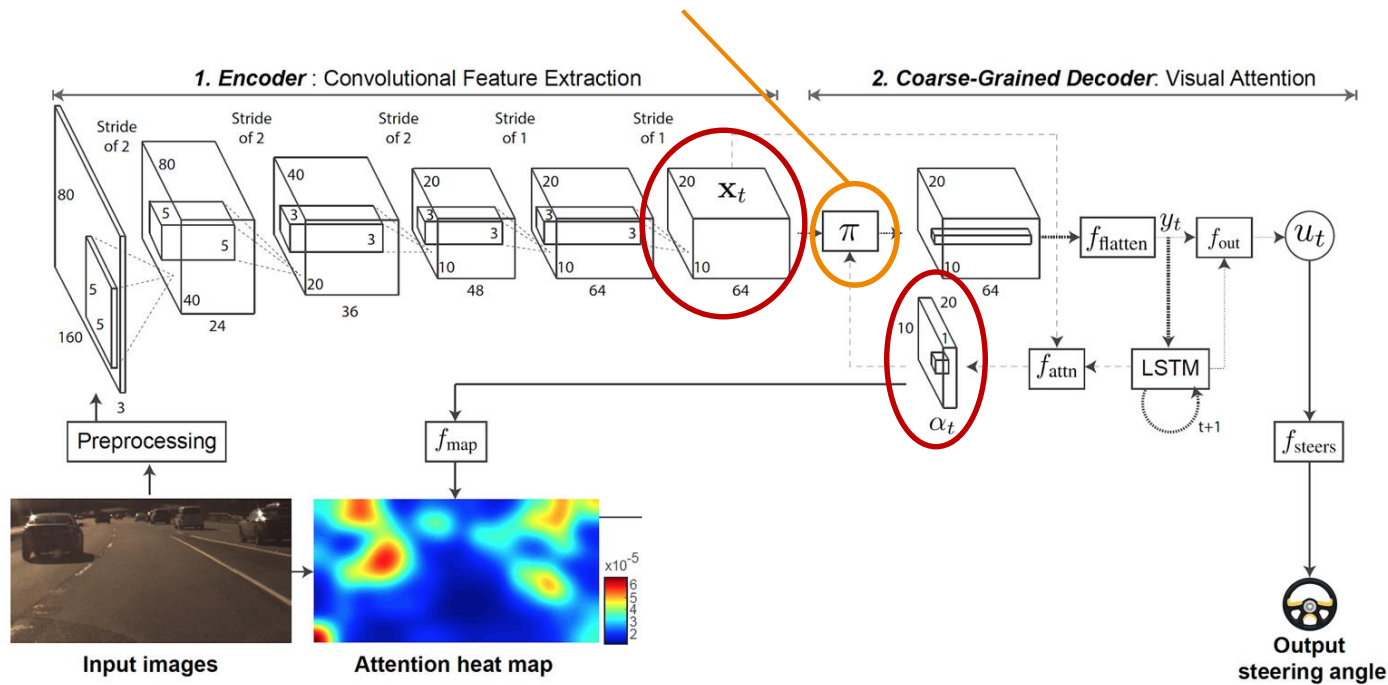


# Attention Mechanics



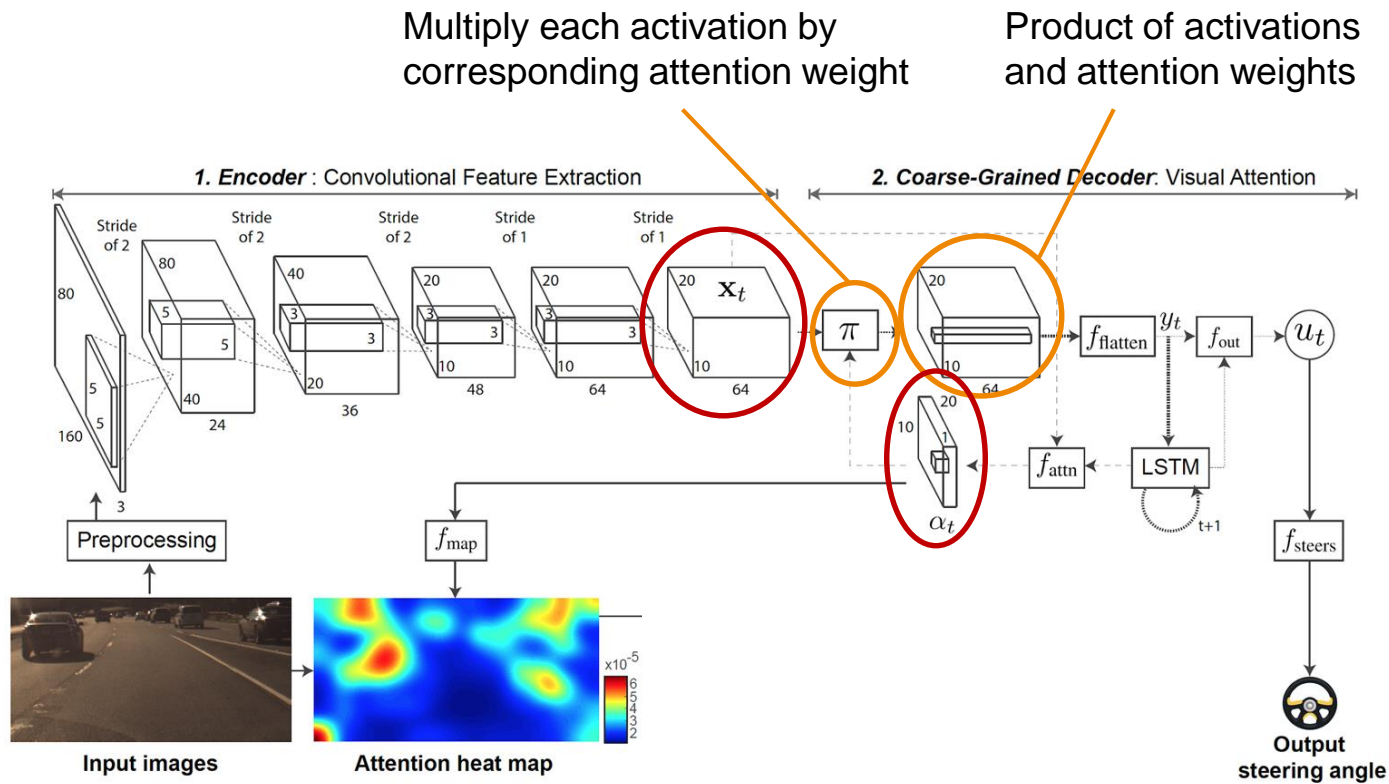
# Attention Mechanics

Multiply each activation  
by corresponding spatial weight

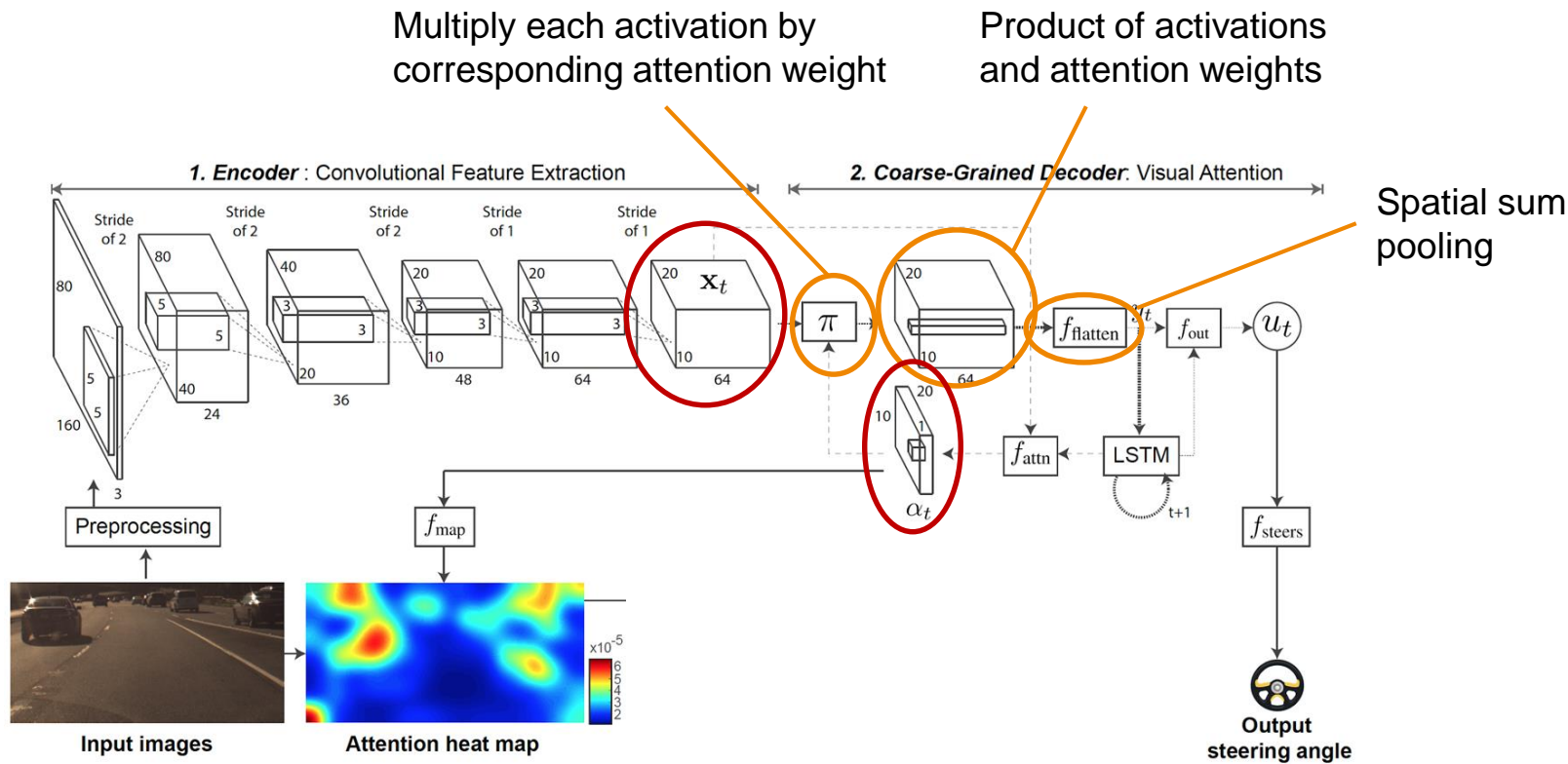




# Attention Mechanics

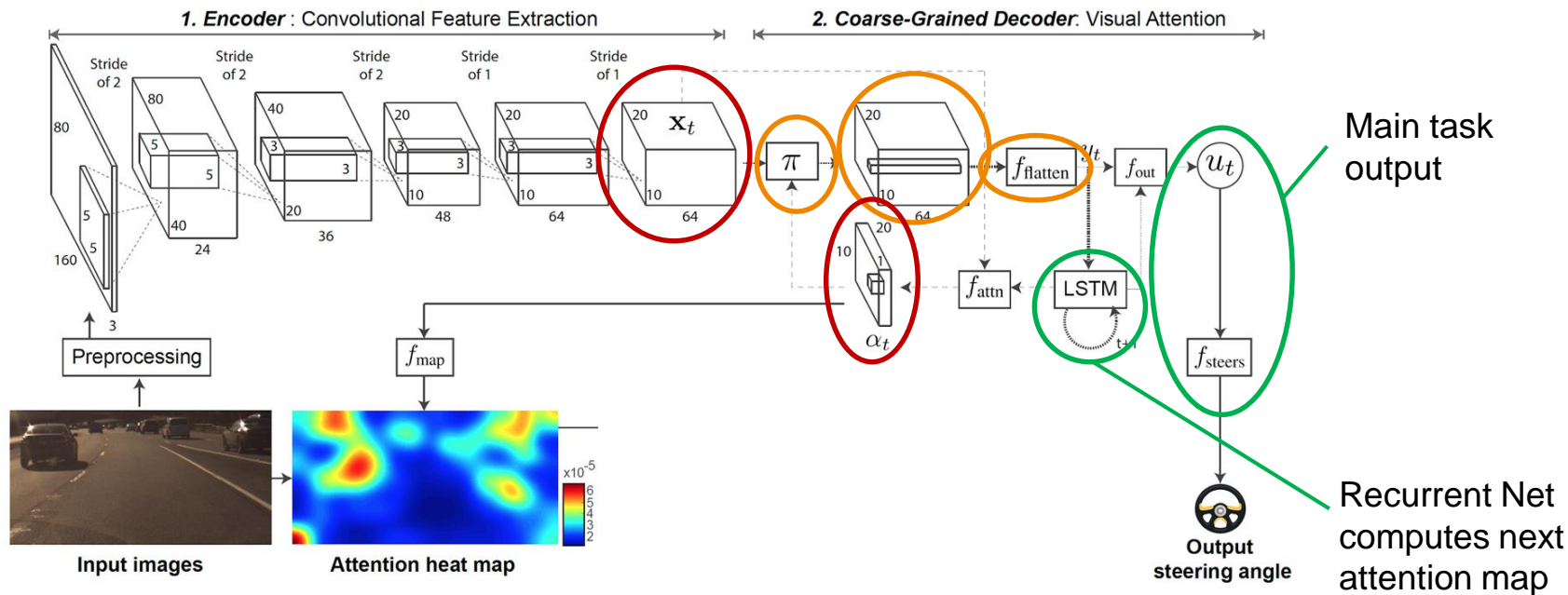


# Attention Mechanics



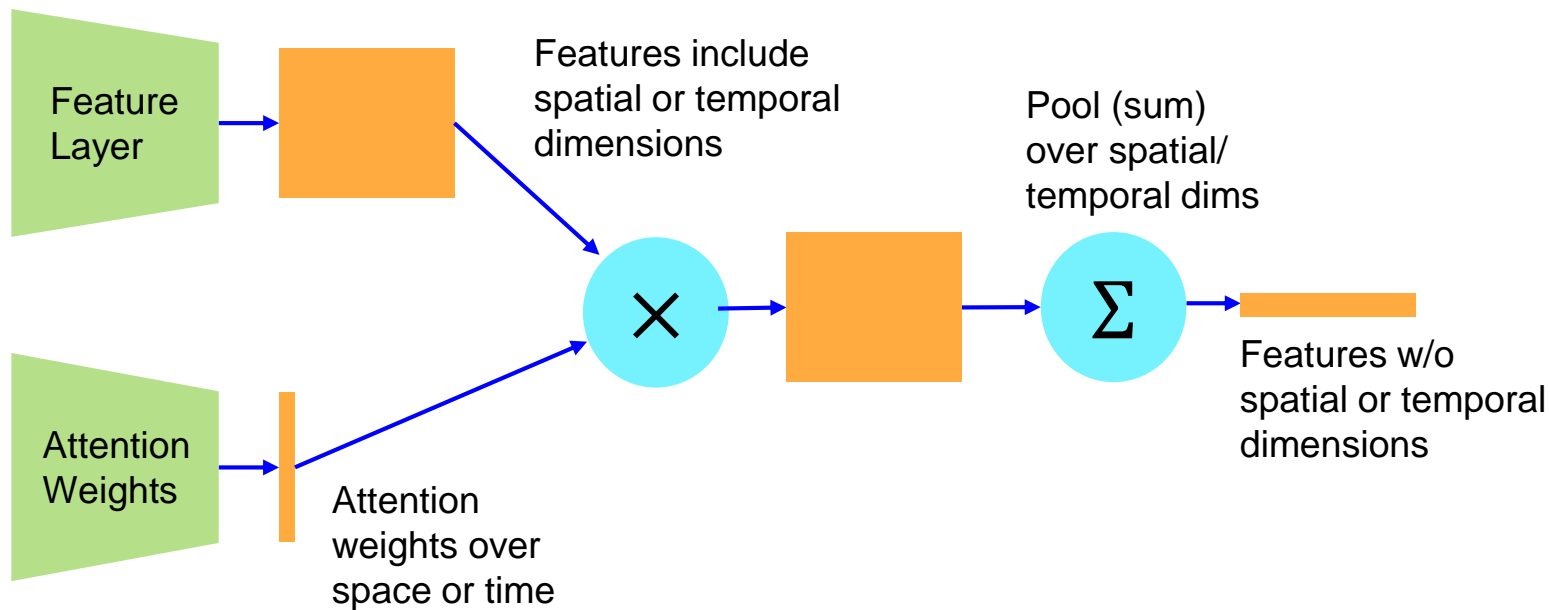
# Attention Mechanics

Multiply each activation by  
corresponding attention weight



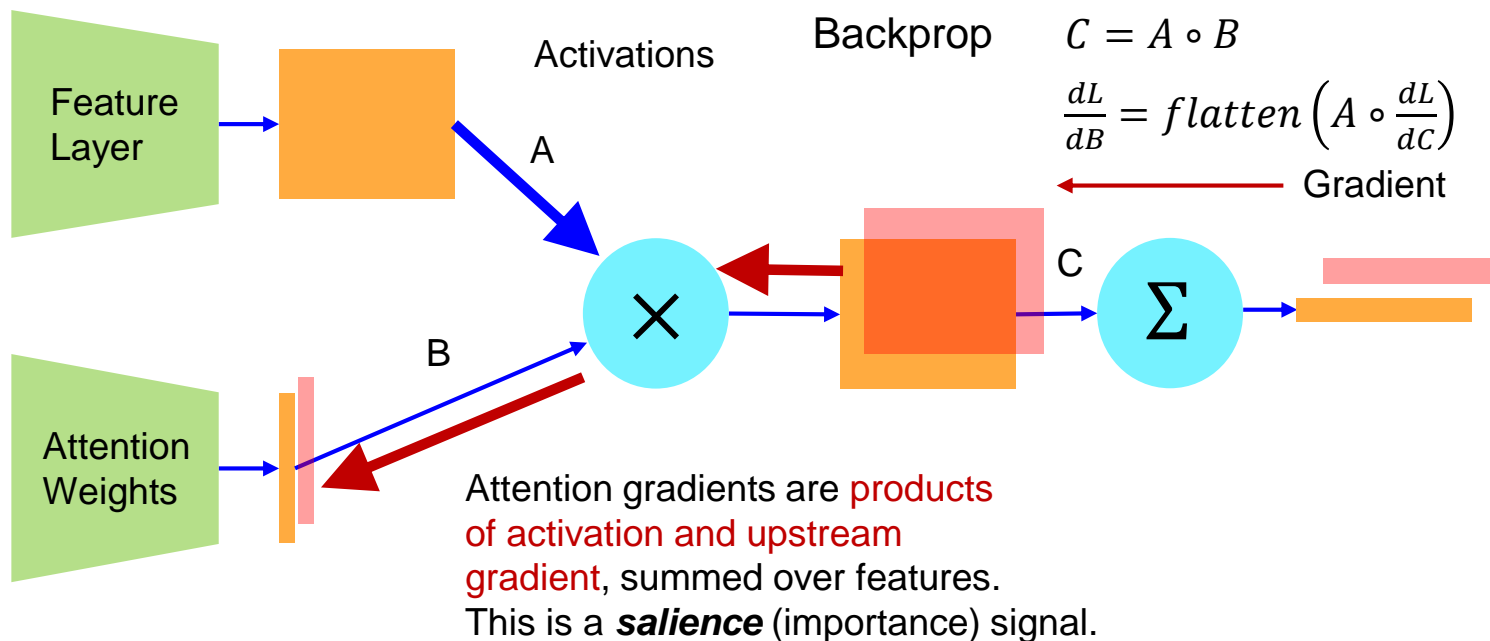
# Attention Mechanics

Typically, soft attention involves a feature layer, a weight predictor, and (optionally) pooling:



# Attention Mechanics: Saliency

During training, the attention layers receives gradients which are the **product of the upstream gradient and the feature layer activations** (saliency).

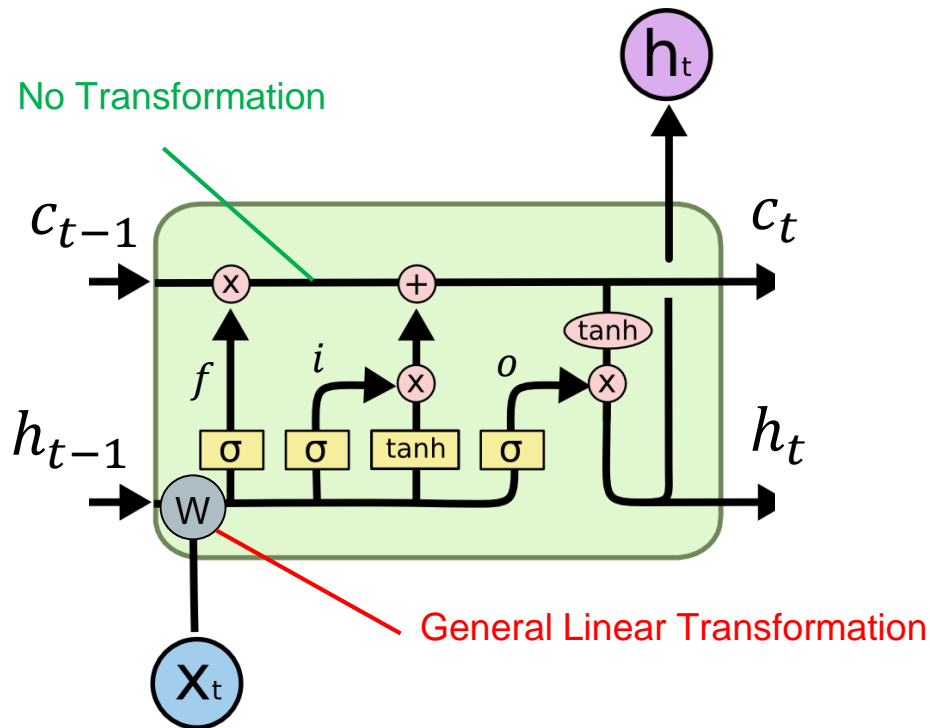


# Attention and LSTMs

We saw something similar in LSTMs:  $i, f, o$  nodes learn to weight features.

They receive a **salience gradient** during training.

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \text{tanh} \end{pmatrix} W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$
$$c_t^l = f \odot c_{t-1}^l + i \odot g$$
$$h_t^l = o \odot \tanh(c_t^l)$$

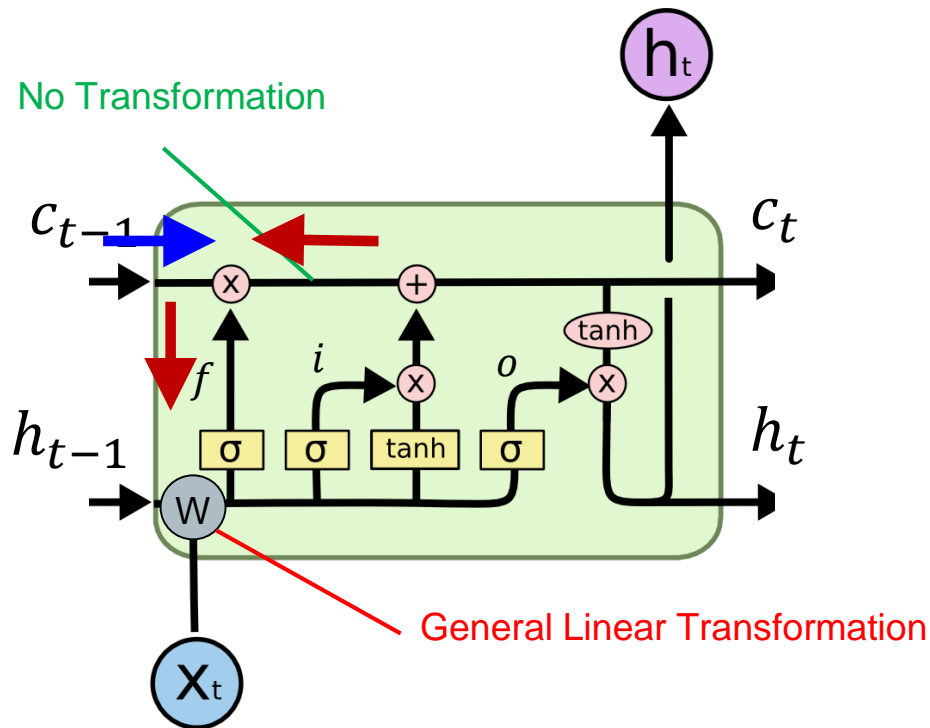


# Attention and LSTMs

We saw something similar in LSTMs:  $i, f, o$  nodes learn to weight features.

They receive a **salience gradient** during training.

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \text{tanh} \end{pmatrix} W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$
$$c_t^l = f \odot c_{t-1}^l + i \odot g$$
$$h_t^l = o \odot \tanh(c_t^l)$$

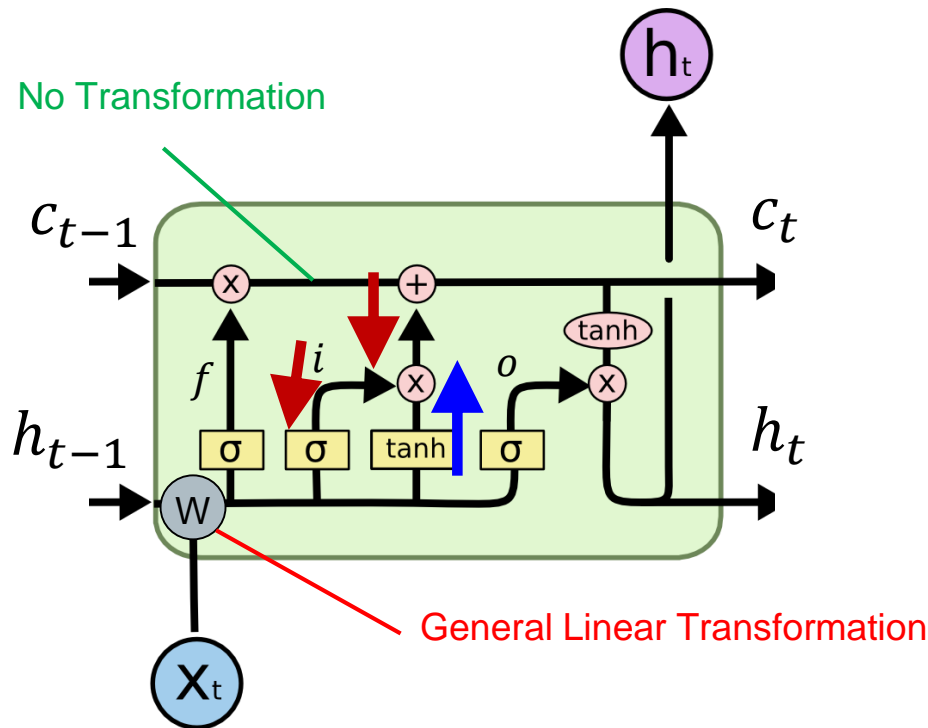


# Attention and LSTMs

We saw something similar in LSTMs:  $i, f, o$  nodes learn to weight features.

They receive a **salience gradient** during training.

$$\begin{pmatrix} i \\ f \\ o \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \text{tanh} \end{pmatrix} W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$
$$c_t^l = f \odot c_{t-1}^l + i \odot g$$
$$h_t^l = o \odot \tanh(c_t^l)$$



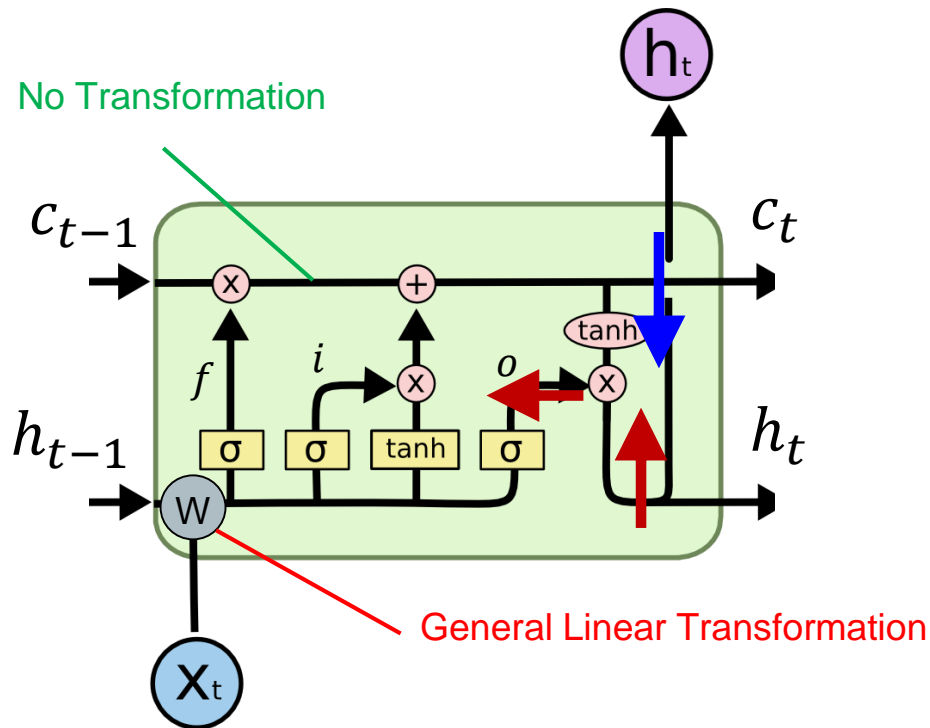


# Attention and LSTMs

We saw something similar in LSTMs:  $i, f, o$  nodes learn to weight features.

They receive a **salience gradient** during training.

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \text{tanh} \end{pmatrix} W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$
$$c_t^l = f \odot c_{t-1}^l + i \odot g$$
$$h_t^l = o \odot \tanh(c_t^l)$$



# Attention as Explanation

Deep Network behavior is generally inscrutable.

Deep Networks do not model data like classical ML models.

Activations don't have obvious meaning (mostly).

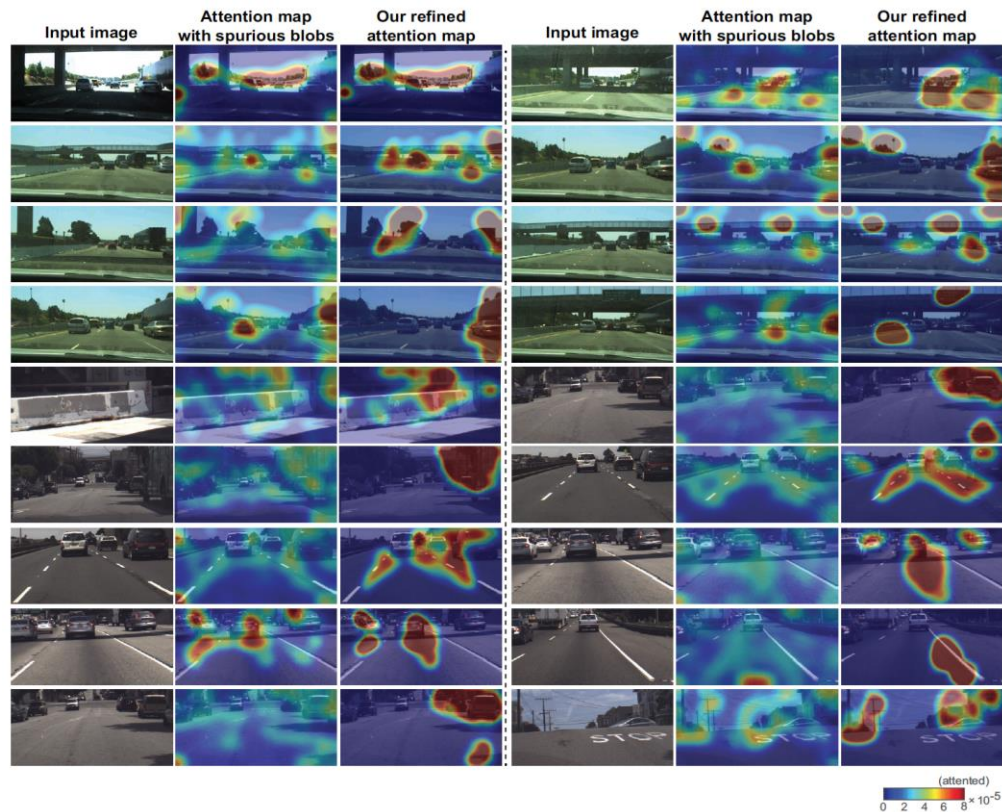
Attention maps are **explanations of net behavior** because they identify the influential parts of the input stream.

Example videos.

# Attention and True Saliency

Attention models must be conservative: they should attend to inputs that **might be salient**.

Testing “blobs” of attention for true saliency can reduce the clutter in the attention map:



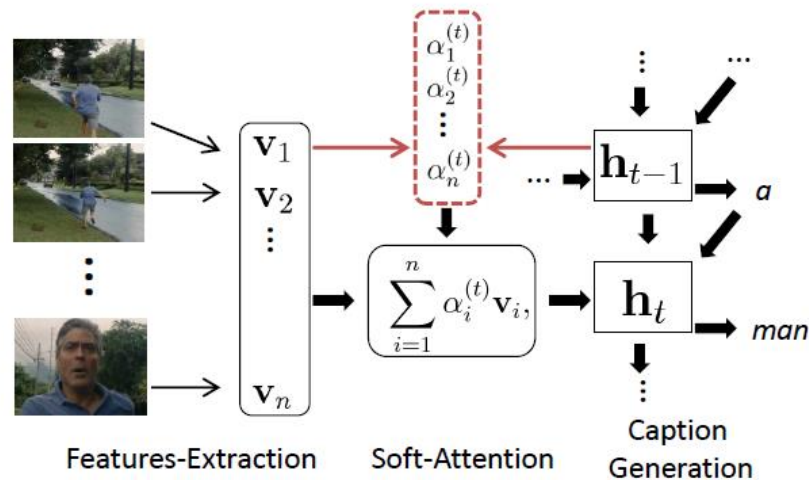
# Soft Attention for Video

“Describing Videos by Exploiting Temporal Structure,” Li Yao et al, arXiv 2015.



# Soft Attention for Video

The attention model:



“Describing Videos by Exploiting Temporal Structure,” Li Yao et al, arXiv 2015.

# Examples



**+Local+Global:** A **man** and a **woman** are **talking** on the **road**

**Ref:** A man and a woman ride a motorcycle



**+Local+Global:** **Someone** is **frying** a **fish** in a **pot**

**+Local:** Someone is frying something

**+Global:** The person is cooking

**Basic:** A man cooking its kitchen

**Ref:** A woman is frying food



**+Local+Global:** the **girl** **grins** at **him**

**Ref:** SOMEONE and SOMEONE swap a look



**+Local+Global:** as **SOMEONE** **sits** on the **table**,  
**SOMEONE** shifts his **gaze** to **SOMEONE**

**+Local:** with a smile SOMEONE arrives

**+Global:** SOMEONE sits at a table

**Basic:** now, SOMEONE grins

**Ref:** SOMEONE gaze at SOMEONE

# Soft Attention for Video

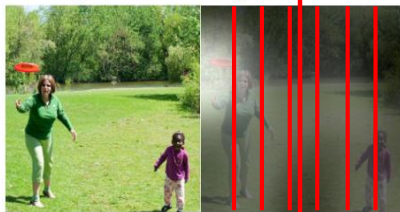
Table 1. Performance of different variants of the model on the Youtube2Text and DVS datasets.

Model	Youtube2Text				DVS			
	BLEU	METEOR	CIDEr	Perplexity	BLEU	METEOR	CIDEr	Perplexity
Enc-Dec (Basic)	0.3869	0.2868	0.4478	33.09	0.003	0.044	0.044	88.28
+ Local (3-D CNN)	0.3875	0.2832	0.5087	33.42	0.004	0.051	0.050	84.41
+ Global (Temporal Attention)	0.4028	0.2900	0.4801	27.89	0.003	0.040	0.047	66.63
+ Local + Global	<b>0.4192</b>	<b>0.2960</b>	<b>0.5167</b>	<b>27.55</b>	<b>0.007</b>	<b>0.057</b>	<b>0.061</b>	<b>65.44</b>
Venugopalan <i>et al.</i> [41]	0.3119	0.2687	-	-	-	-	-	-
+ Extra Data (Flickr30k, COCO)	0.3329	0.2907	-	-	-	-	-	-
Thomason <i>et al.</i> [37]	0.1368	0.2390	-	-	-	-	-	-

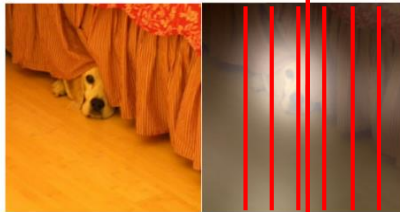


# Soft Attention for Captioning

Attention constrained to fixed grid!



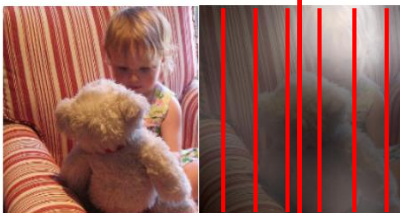
A woman is throwing a frisbee in a park.



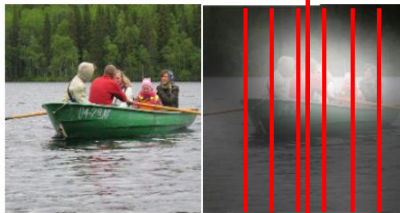
A dog is standing on a hardwood floor.



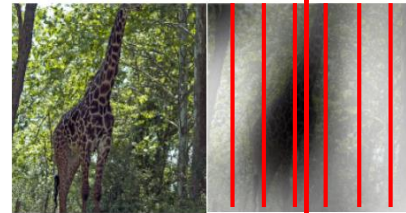
A stop sign is on a road with a mountain in the background.



A little girl sitting on a bed with a teddy bear.



A group of people sitting on a boat in the water.



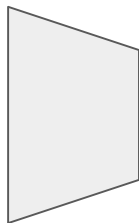
A giraffe standing in a forest with trees in the background.



# Attending to arbitrary regions?



Image:  
 $H \times W \times 3$



Features:  
 $L \times D$

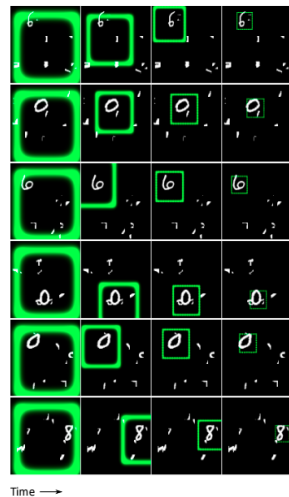


A woman is throwing a frisbee in a park.

Attention mechanism from Show, Attend, and Tell only lets us softly attend to fixed grid positions ... can we do better?

# Attending to Arbitrary Regions: DRAW

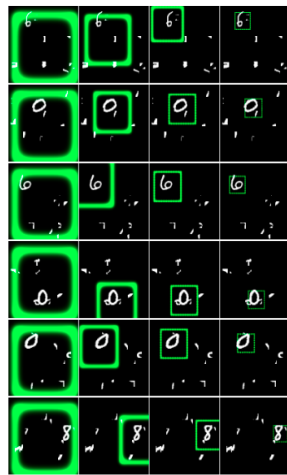
**Classify** images by attending to arbitrary regions of the *input*



Gregor et al, "DRAW: A Recurrent Neural Network For Image Generation", ICML 2015

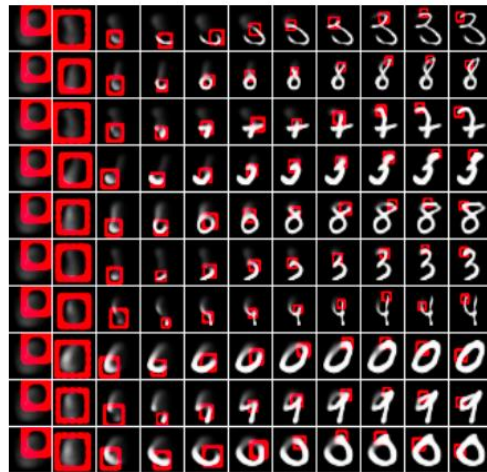
# Attending to Arbitrary Regions: DRAW

**Classify** images by attending to arbitrary regions of the *input*



Time →

**Generate** images by attending to arbitrary regions of the *output*

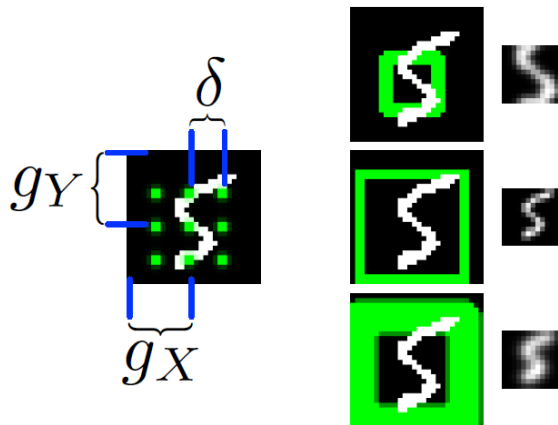


Time →

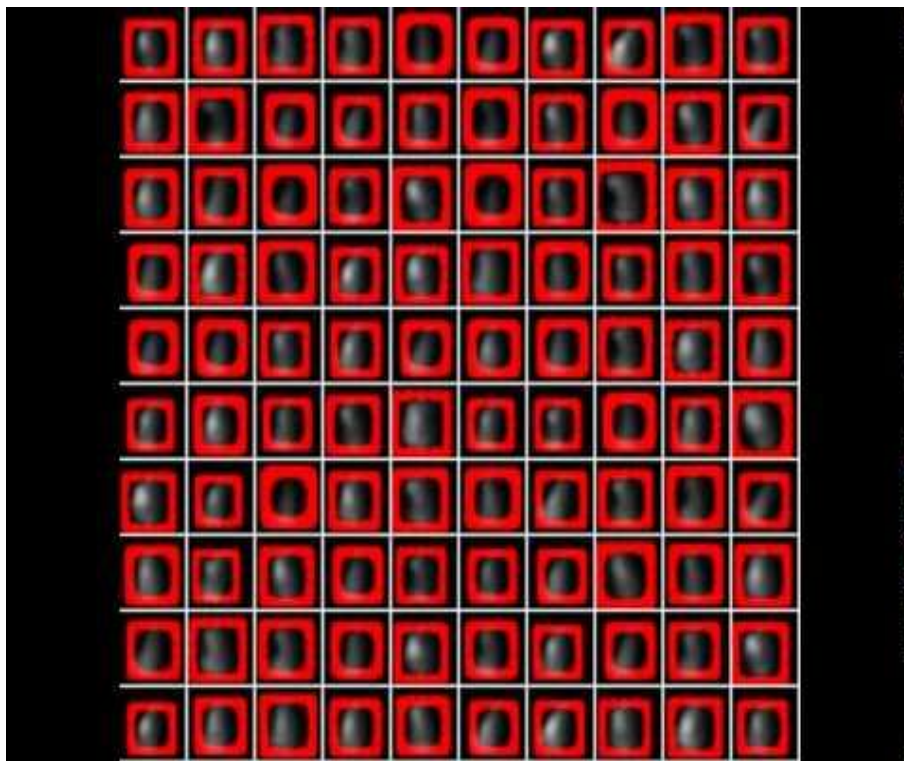
Gregor et al, "DRAW: A Recurrent Neural Network For Image Generation", ICML 2015

# Attending to Arbitrary Regions: DRAW

Attention is a parametric distribution: both location and scale can vary:



Gregor et al, "DRAW: A Recurrent Neural Network For Image Generation", ICML 2015



Based on cs231n by Fei-Fei Li & Andrej Karpathy & Justin Johnson

# Attention Takeaways

## Performance:

Attention models can *improve accuracy* and *reduce computation* at the same time.

## Saliency:

Attention models learn to predict saliency, i.e. to emphasize relevant input data across space or time.



# Attention Takeaways

## Explainability:

Attention models encode explanations.

Both locus and trajectory help understand what's going on.

## Hard vs. Soft:

Soft models are easier to train, hard models require reinforcement learning.

