Image Captioning

a small dog is sitting on a beach
logprob: -11.86

Fast Campus
Start Deep Learning with Tensorflow

Image Captioning?

- https://github.com/tensorflow/models/tree/master/research/im2txt

A person on a beach flying a kite.

A black and white photo of a train on a train track.

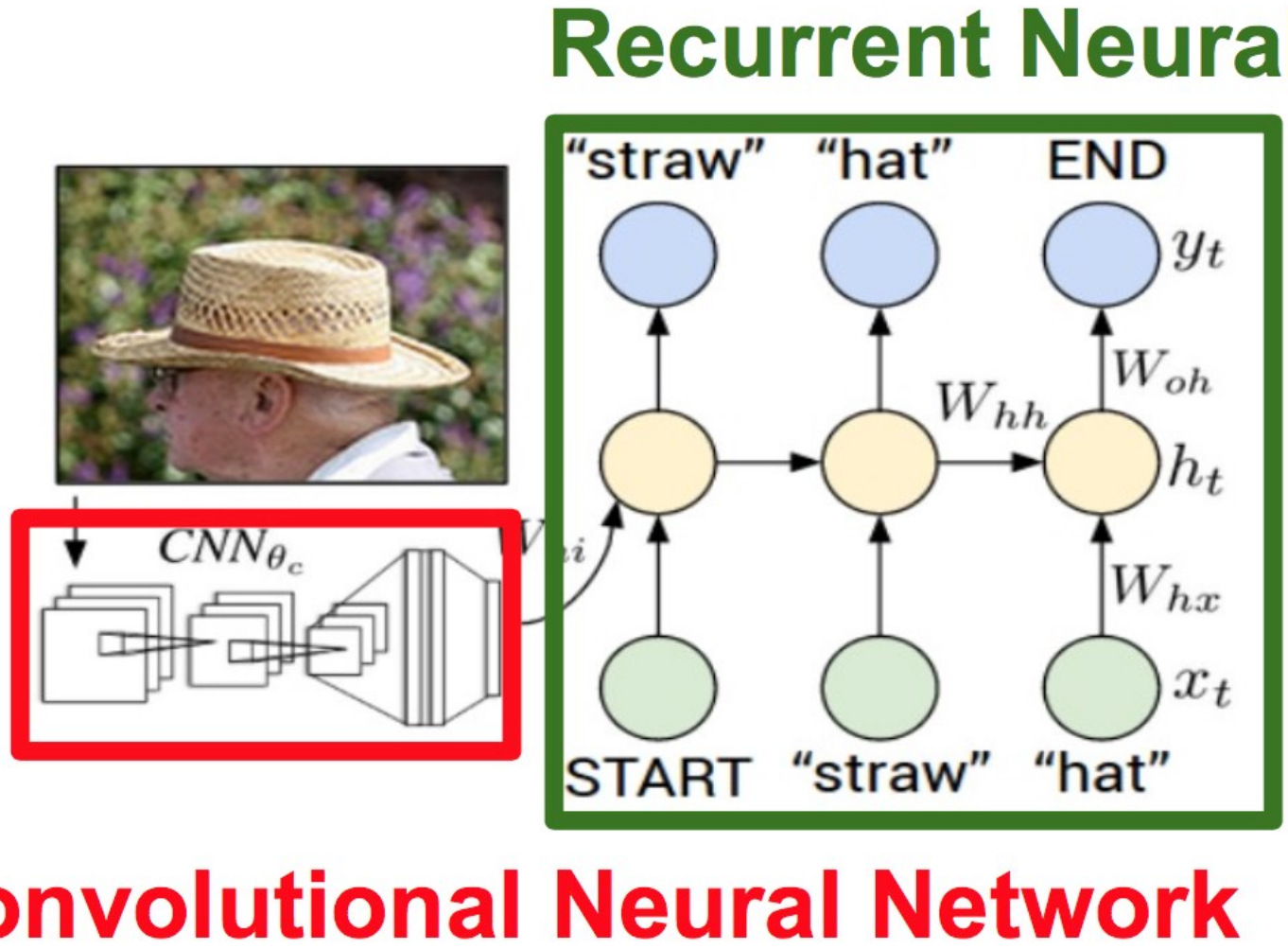
A person skiing down a snow covered slope.

A group of giraffe standing next to each other.

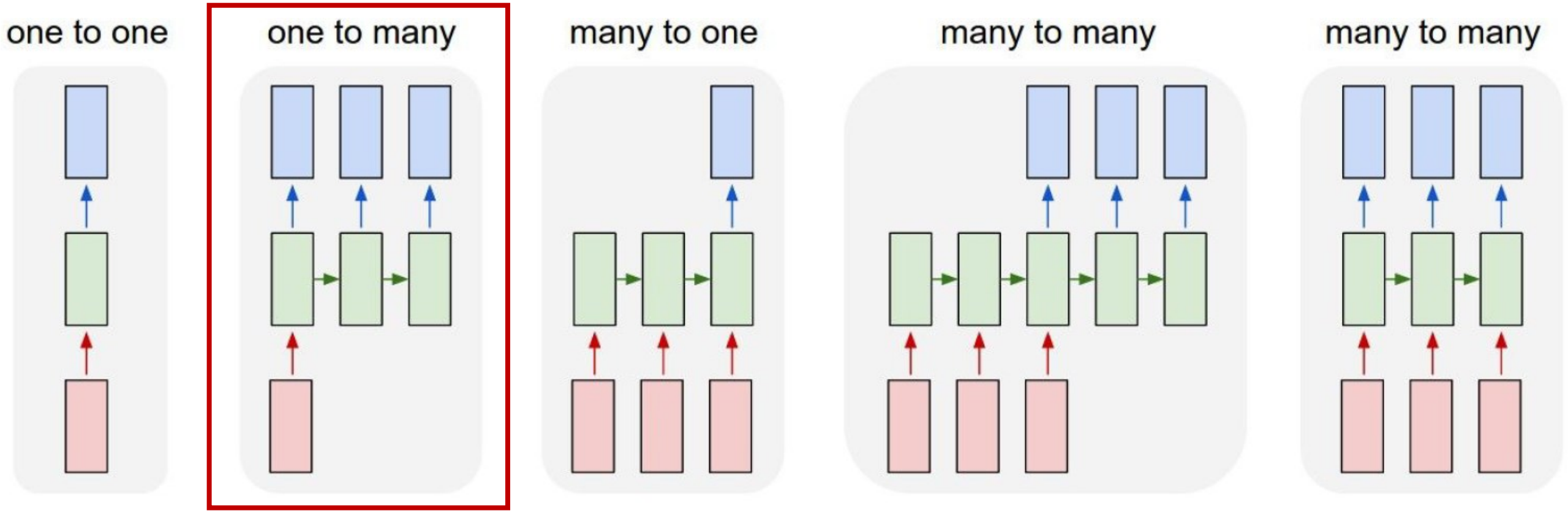
Fei Fei Li – Ted Talk

https://www.ted.com/talks/fei_fei_li_how_we_re_teaching_computers_to_understand_pictures?language=ko

Overall Architecture

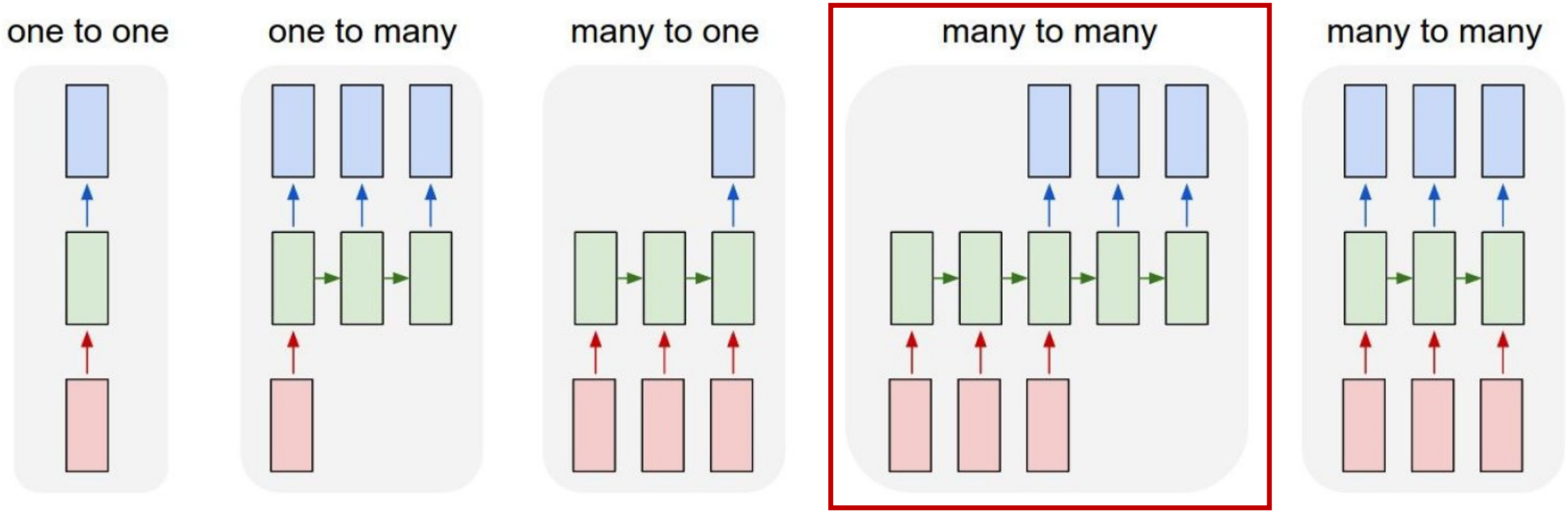


Recurrent Neural Network



e.g. **Image Captioning**
image -> sequence of words

Recurrent Neural Network

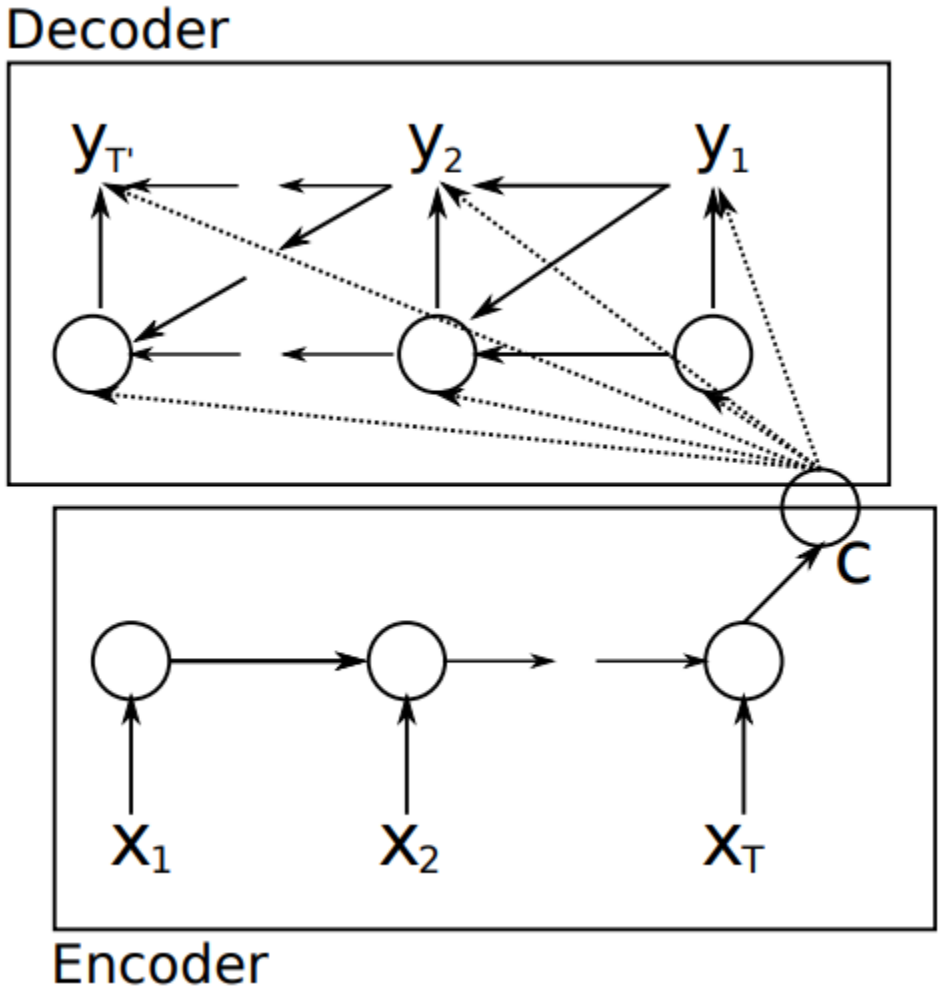


e.g. **Machine Translation**
seq of words -> seq of words

Sequence to Sequence – NMT

- Kyunghyun Cho, "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation"

- Encoder-decoder model
- GRU!

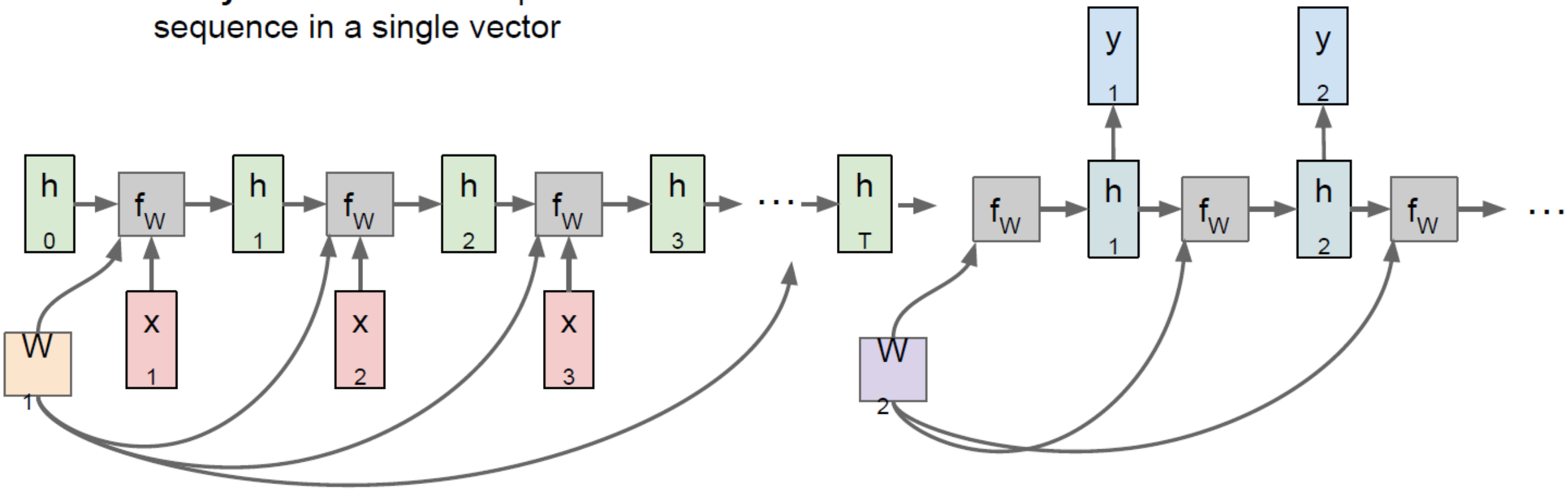


Sequence to Sequence

- Many to one + one to many

Many to one: Encode input sequence in a single vector

One to many: Produce output sequence from single input vector

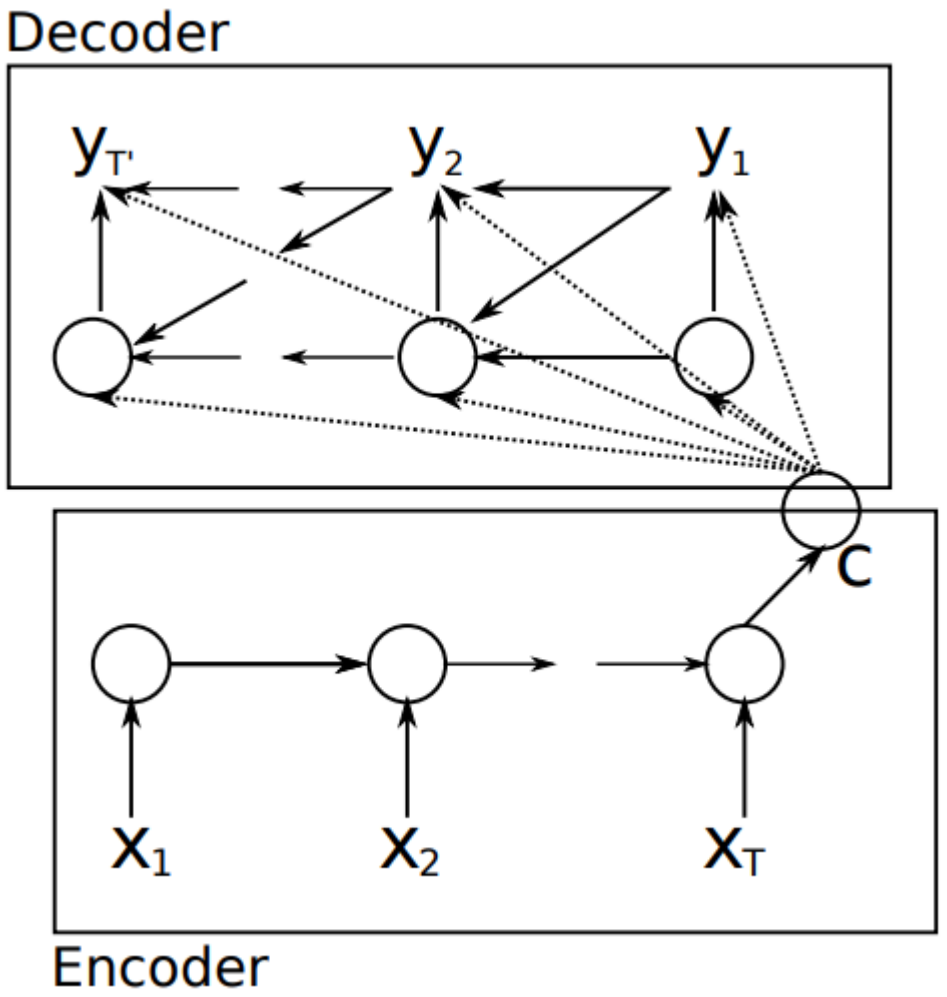


Encoder

$$h_j^{\langle t \rangle} = z_j h_j^{\langle t-1 \rangle} + (1 - z_j)\tilde{h}_j^{\langle t \rangle},$$

$$\begin{aligned}
\tilde{h}_j^{\langle t \rangle} &= \tanh\left(\left[\mathbf{W} e(\mathbf{x}_t) \right]_j + \left[\mathbf{U} \left(\mathbf{r} \odot \mathbf{h}_{\langle t-1 \rangle} \right) \right]_j \right), \\
z_j &= \sigma\left(\left[\mathbf{W}_z e(\mathbf{x}_t) \right]_j + \left[\mathbf{U}_z \mathbf{h}_{\langle t-1 \rangle} \right]_j \right), \\
r_j &= \sigma\left(\left[\mathbf{W}_r e(\mathbf{x}_t) \right]_j + \left[\mathbf{U}_r \mathbf{h}_{\langle t-1 \rangle} \right]_j \right).
\end{aligned}$$

$$\mathbf{c} = \tanh\left(\mathbf{V} \mathbf{h}^{\langle N \rangle} \right).$$

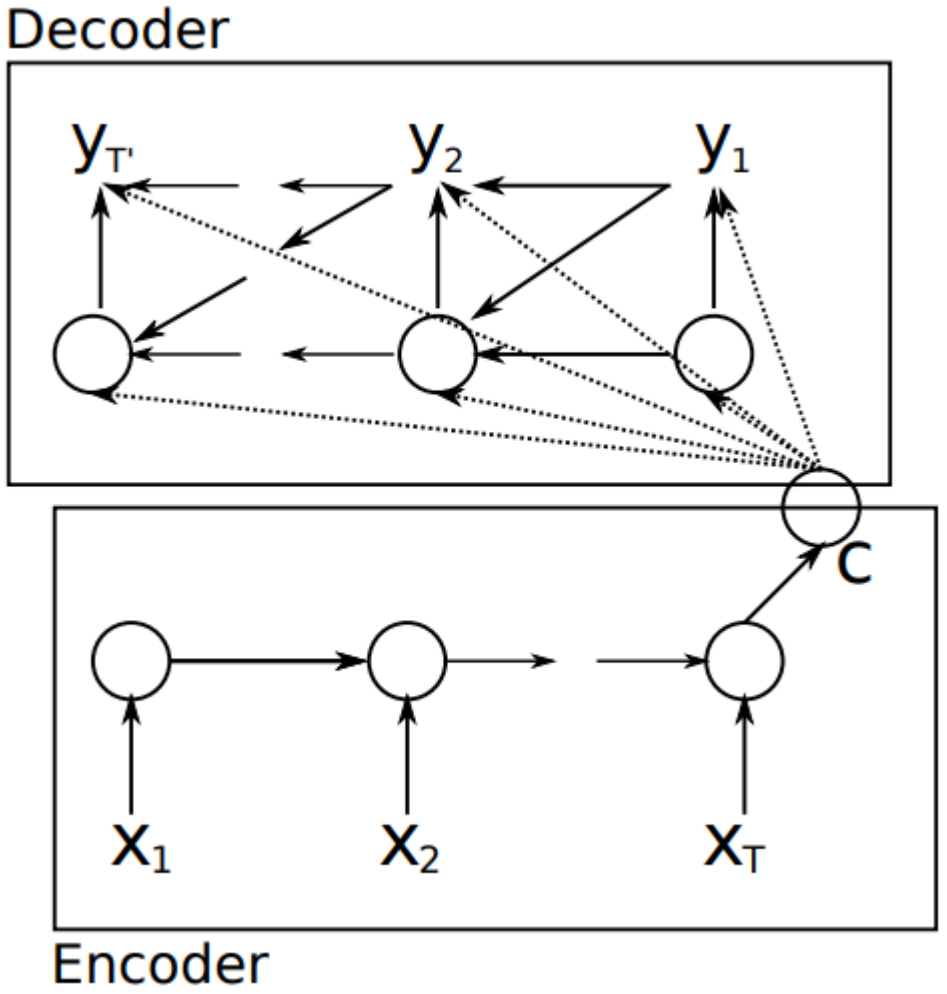


Decoder

$$\mathbf{h}'^{\langle 0 \rangle} = \tanh\left(\mathbf{V}'\mathbf{c}\right),$$

$$h'^{\langle t \rangle}_j = z'_j h'^{\langle t-1 \rangle}_j + (1 - z'_j)\tilde{h}'^{\langle t \rangle}_j,$$

$$\begin{aligned}
\tilde{h}'^{\langle t \rangle}_j &= \tanh\left(\left[\mathbf{W}'e(\mathbf{y}_{t-1})\right]_j + r'_j\left[\mathbf{U}'\mathbf{h}'_{\langle t-1 \rangle} + \mathbf{C}\mathbf{c}\right]\right), \\
z'_j &= \sigma\left(\left[\mathbf{W}'_z e(\mathbf{y}_{t-1})\right]_j + \left[\mathbf{U}'_z \mathbf{h}'_{\langle t-1 \rangle}\right]_j + \left[\mathbf{C}_z \mathbf{c}\right]_j\right), \\
r'_j &= \sigma\left(\left[\mathbf{W}'_r e(\mathbf{y}_{t-1})\right]_j + \left[\mathbf{U}'_r \mathbf{h}'_{\langle t-1 \rangle}\right]_j + \left[\mathbf{C}_r \mathbf{c}\right]_j\right),
\end{aligned}$$



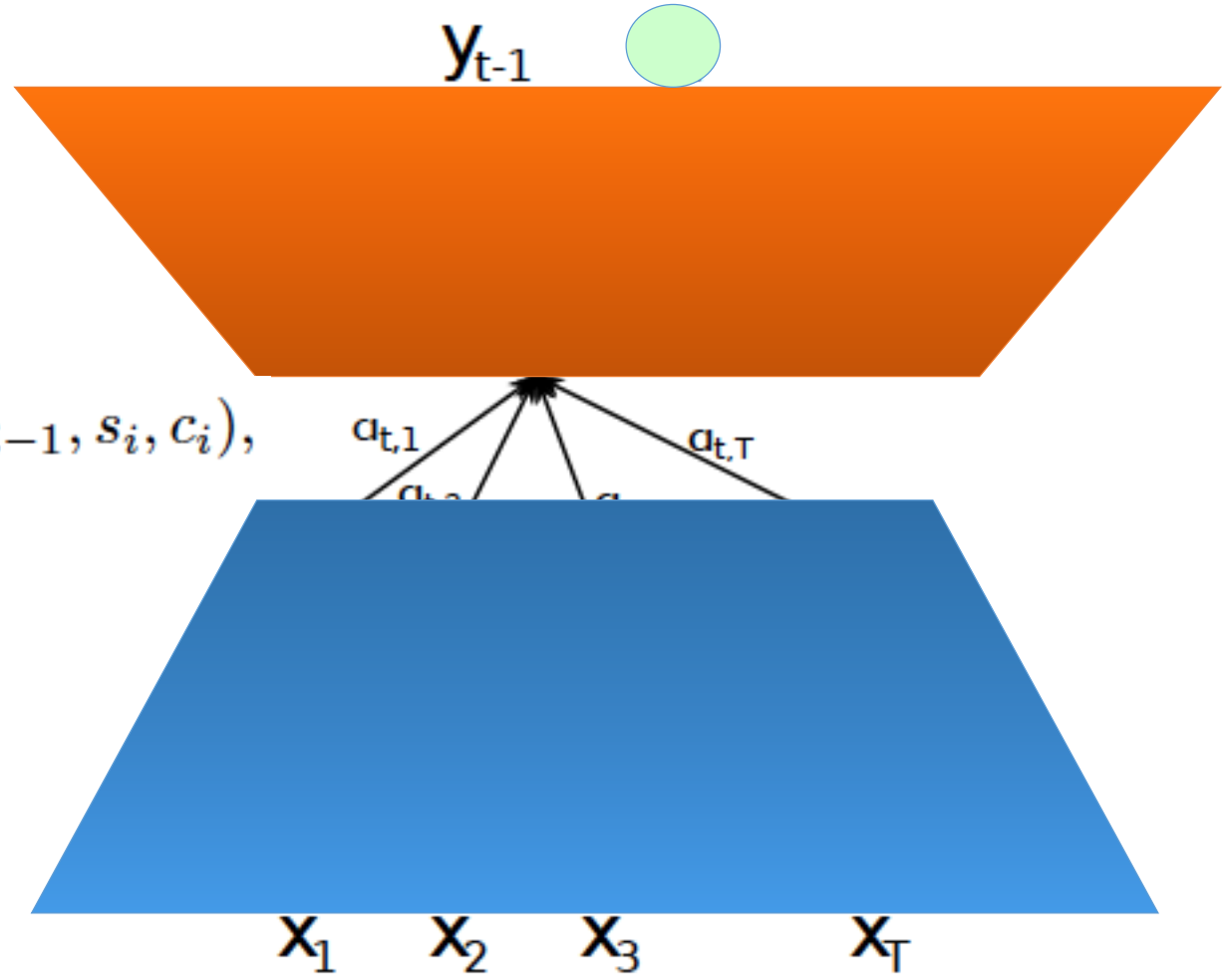
Sequence to Sequence with Attention

- Encoder-decoder model encodes a source sentence into a fixed-length vector from which a decoder generates a translation
- Fixed length vector representation is enough?

Jointly Learning to Align and Translate

- Attention mechanism

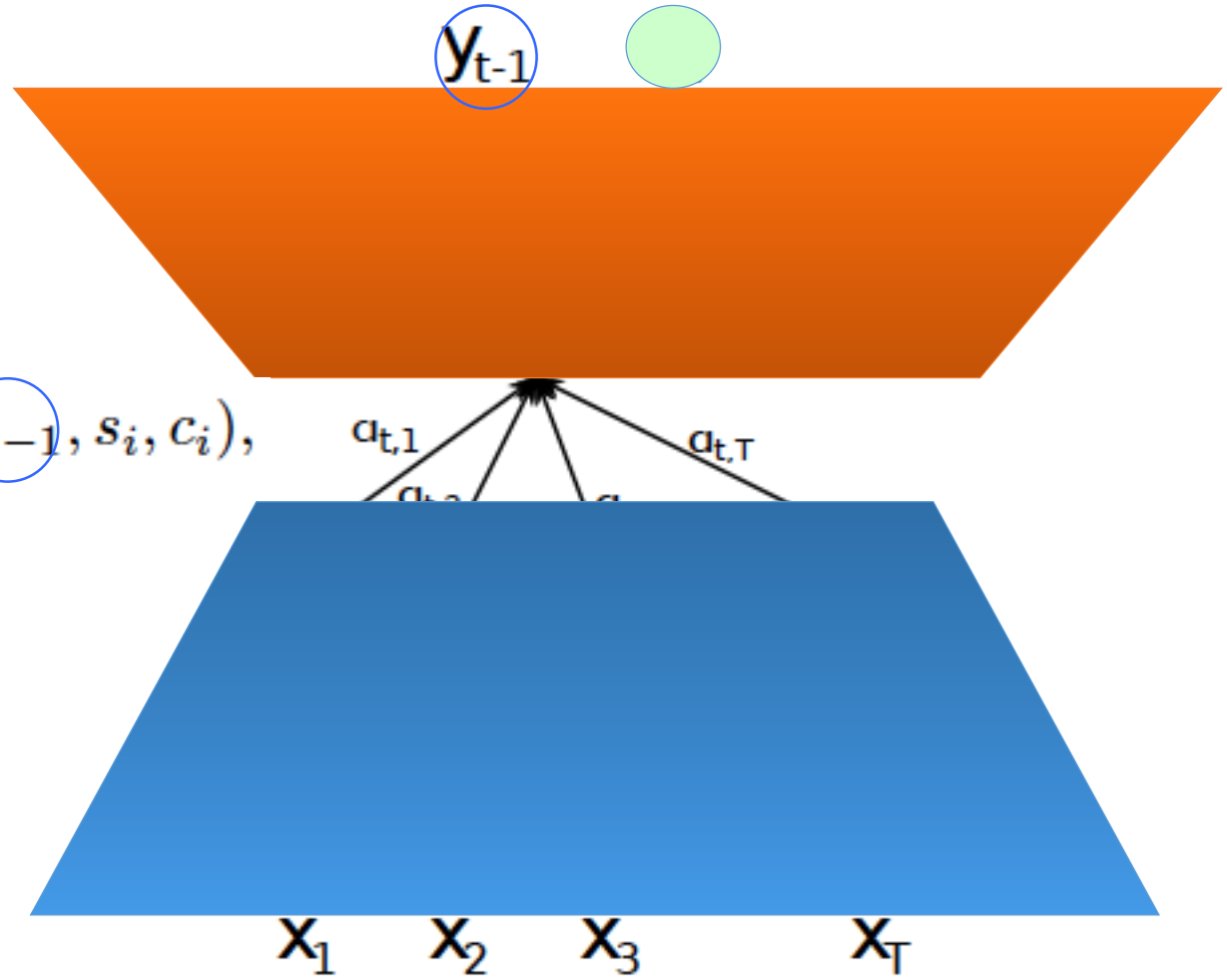
$$p(y_i \mid y_1, \ldots, y_{i-1}, \mathbf{x}) = g(y_{i-1}, s_i, c_i),$$



Jointly Learning to Align and Translate

- Attention mechanism

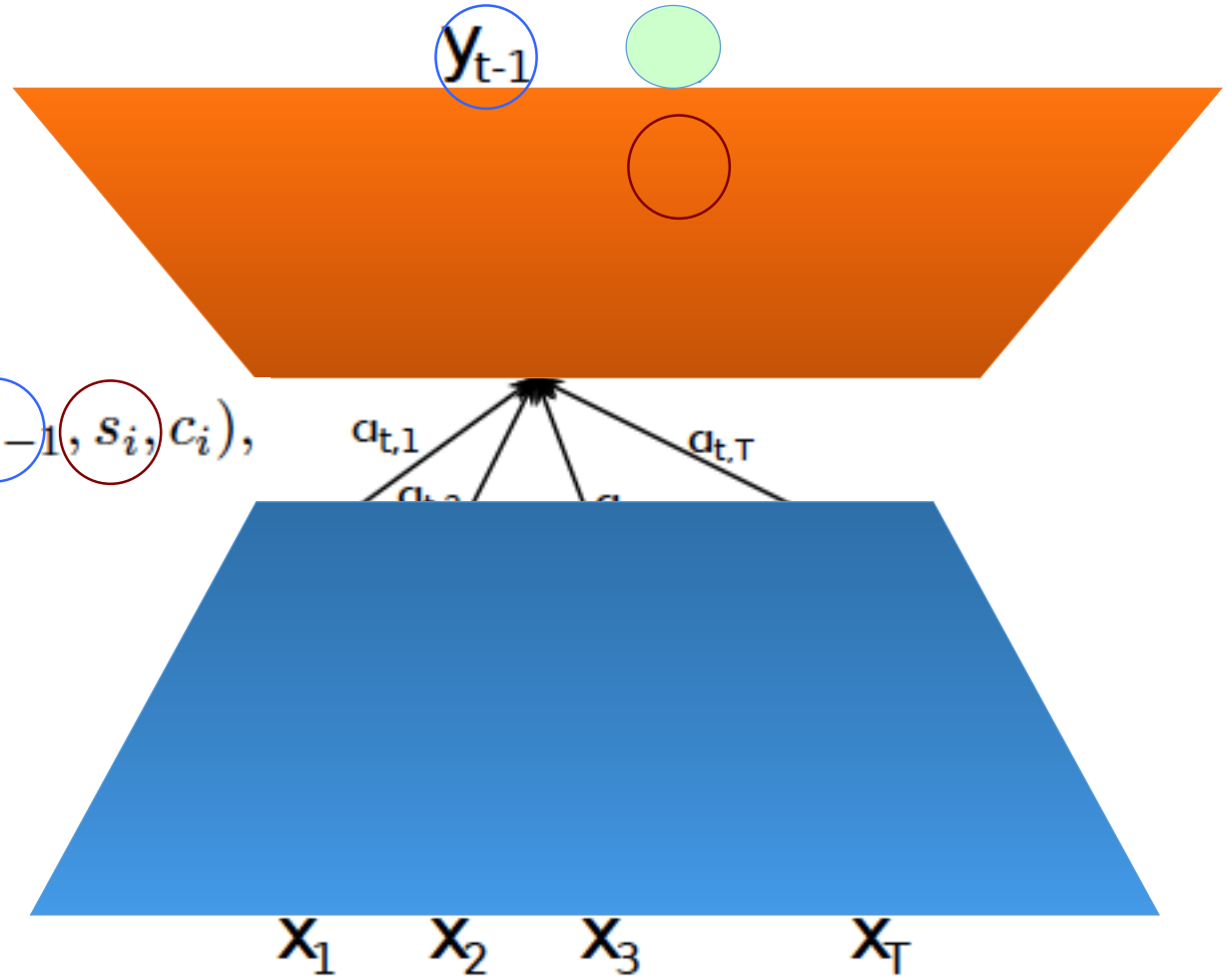
$$p(y_i \mid y_1, \ldots, y_{i-1}, \mathbf{x}) = g(y_{i-1}, s_i, c_i),$$



Jointly Learning to Align and Translate

- Attention mechanism

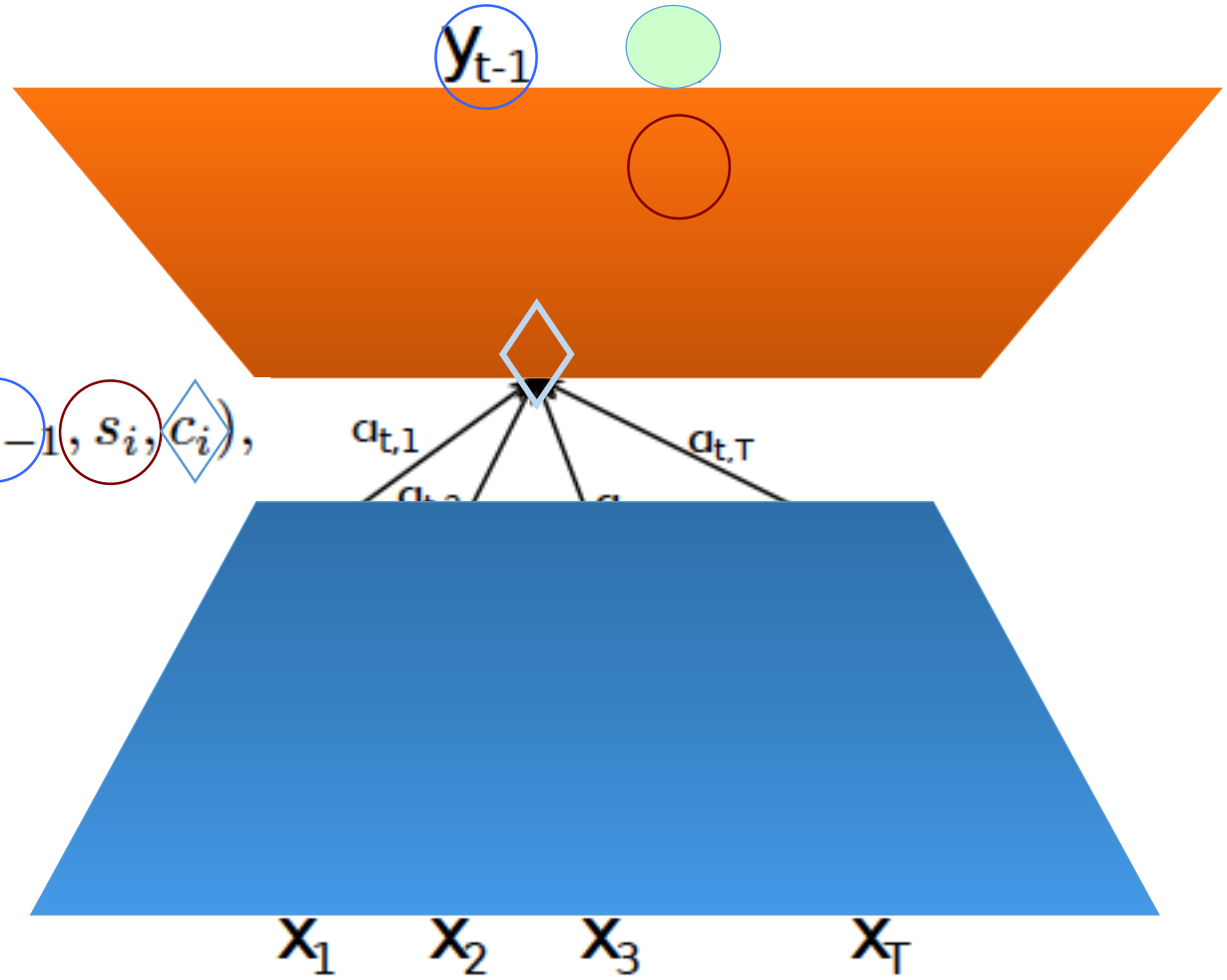
$$p(y_1, \ldots, y_{i-1}, \mathbf{x}) = g(y_{i-1}, s_i, c_i),$$



Jointly Learning to Align and Translate

- Attention mechanism

$$p(y_1, \ldots, y_{i-1}, \mathbf{x}) = g(y_{i-1}, s_i, c_i),$$

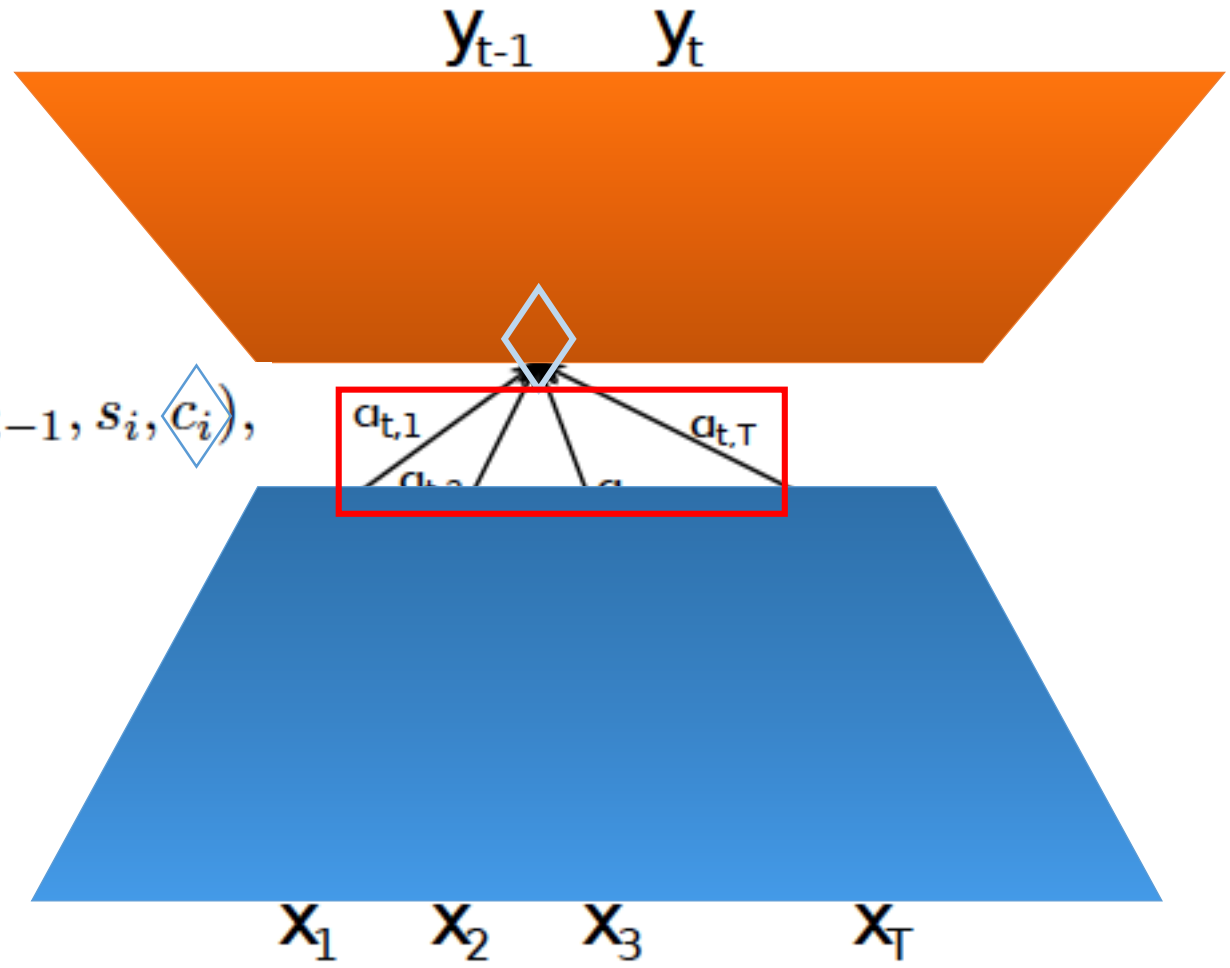


Jointly Learning to Align and Translate

- Attention mechanism

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

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



Jointly Learning to Align and Translate

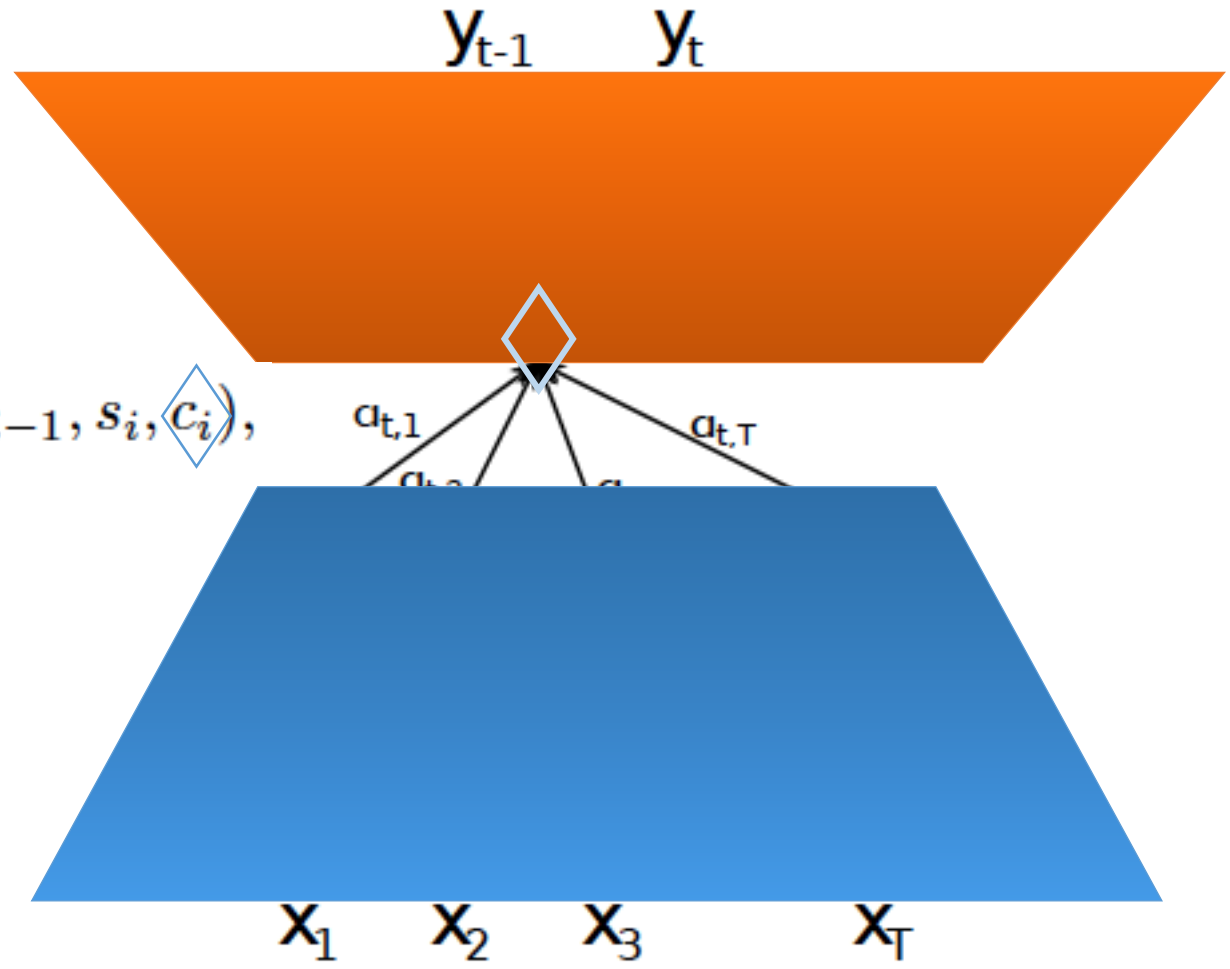
- Attention mechanism

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

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

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})},$$

$$e_{ij} = a(s_{i-1}, h_j)$$



Jointly Learning to Align and Translate

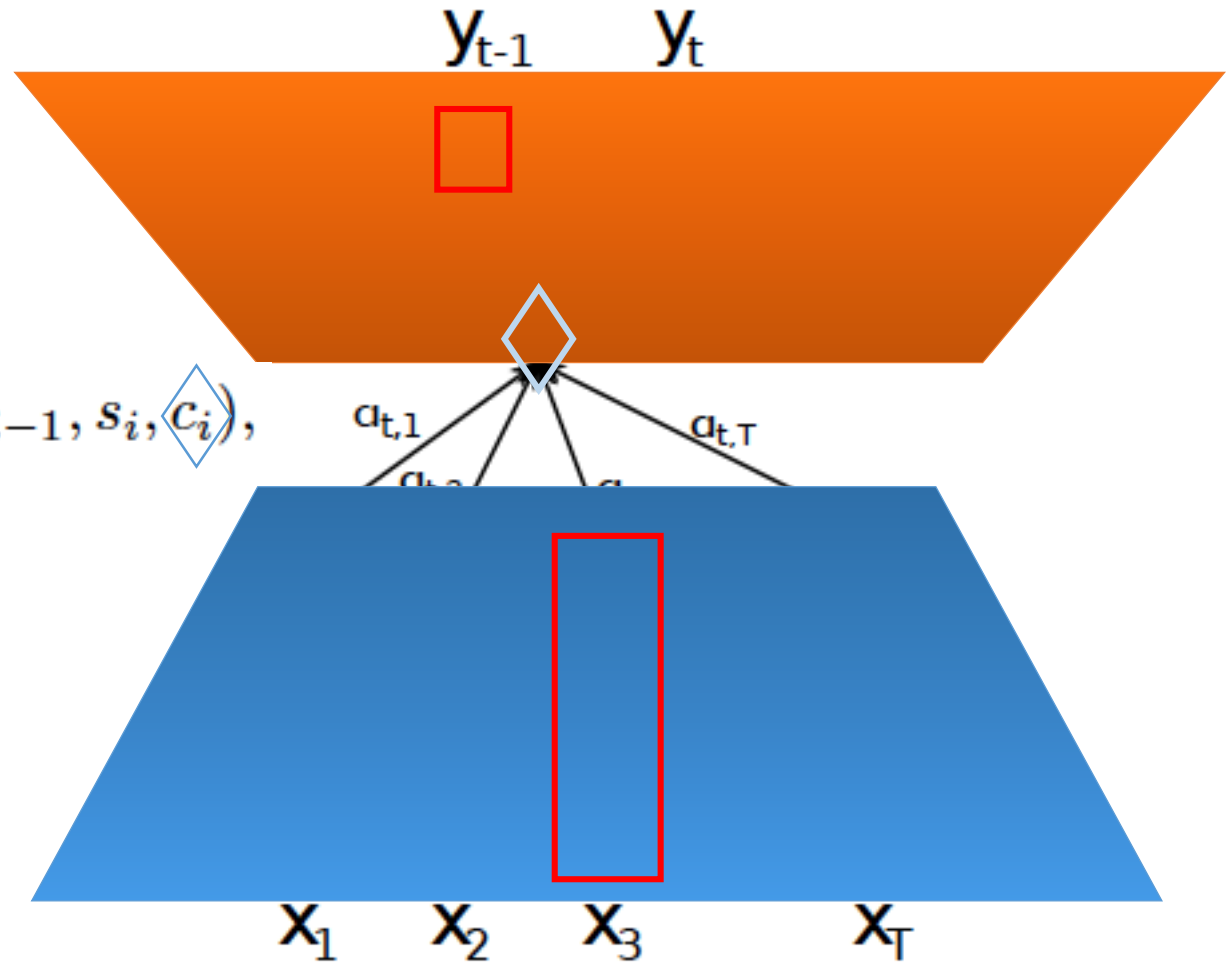
- Attention mechanism

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

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

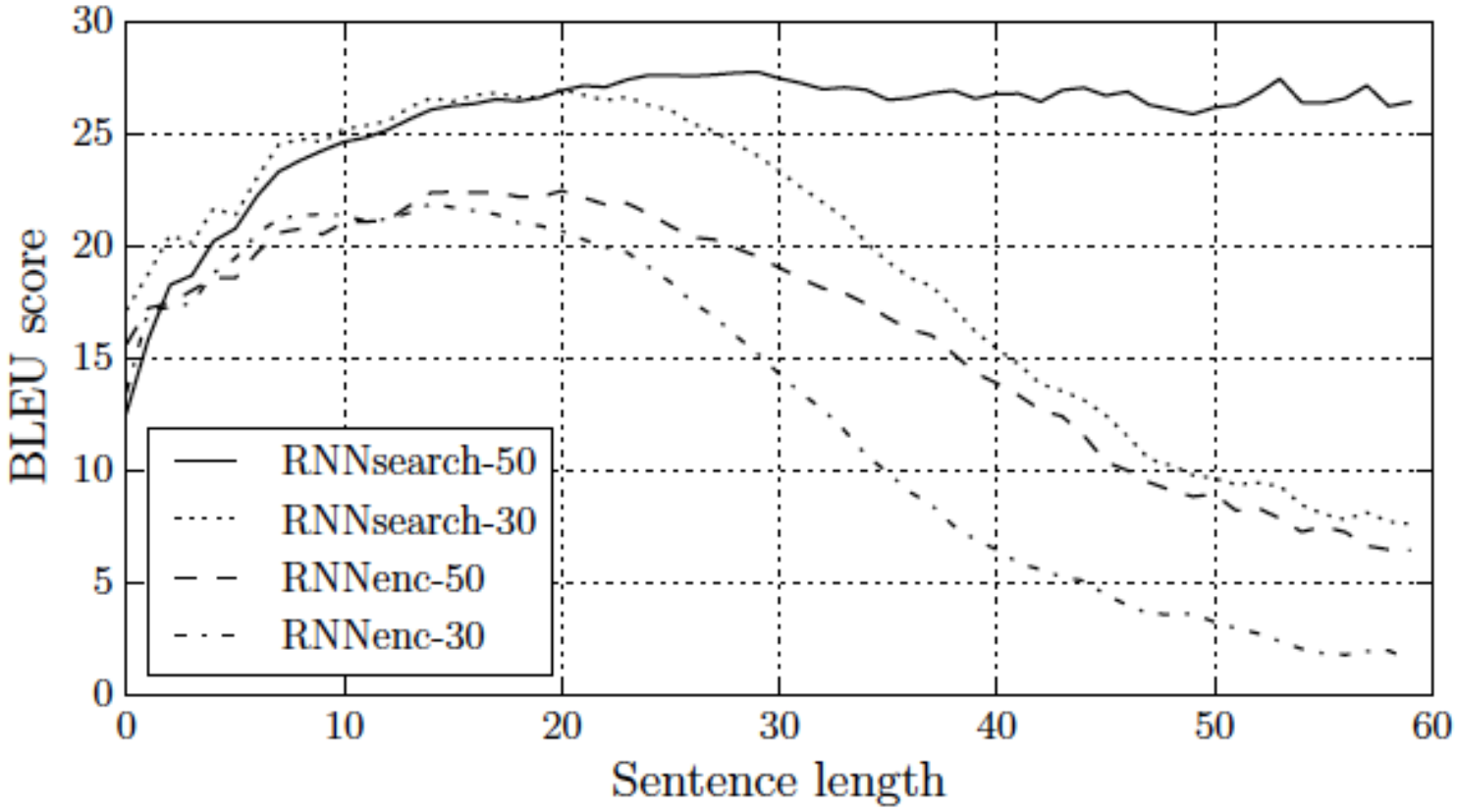
$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})},$$

$$e_{ij} = a(s_{i-1}, h_j)$$



Jointly Learning to Align and Translate

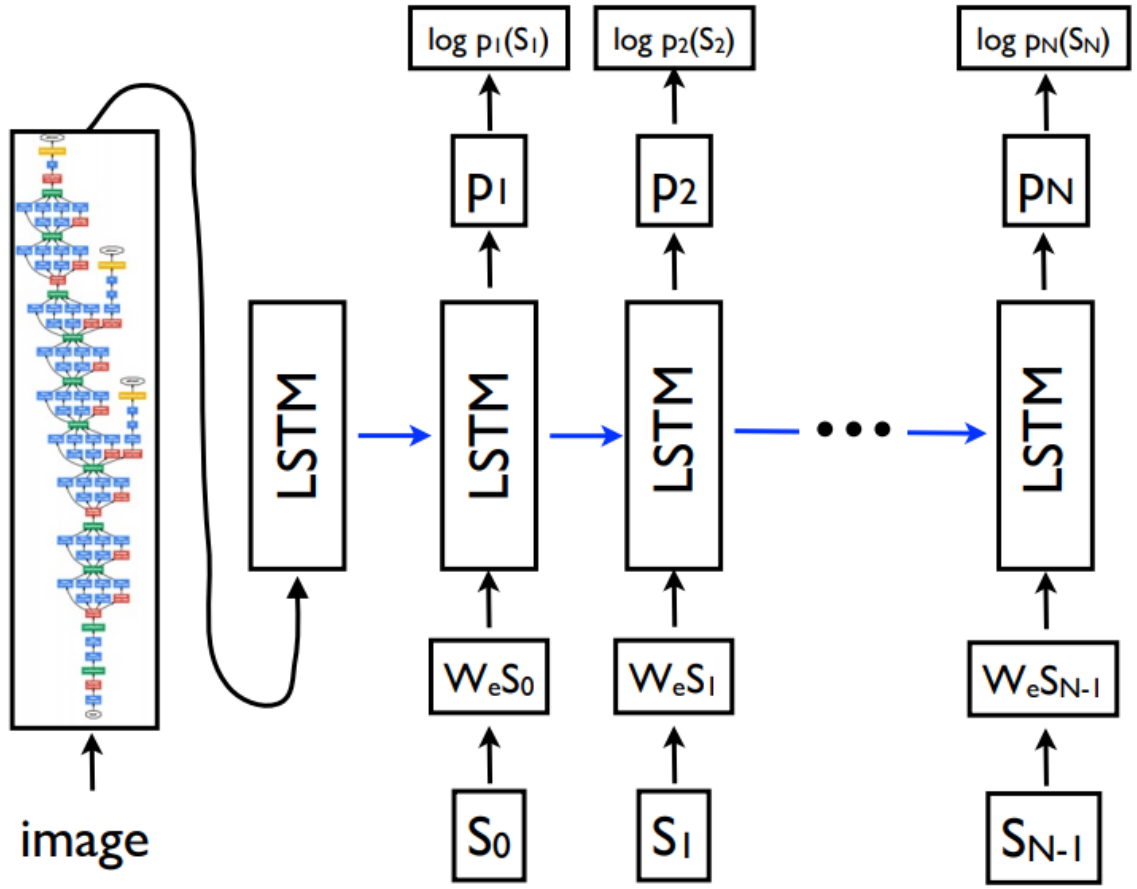
- Long sentences



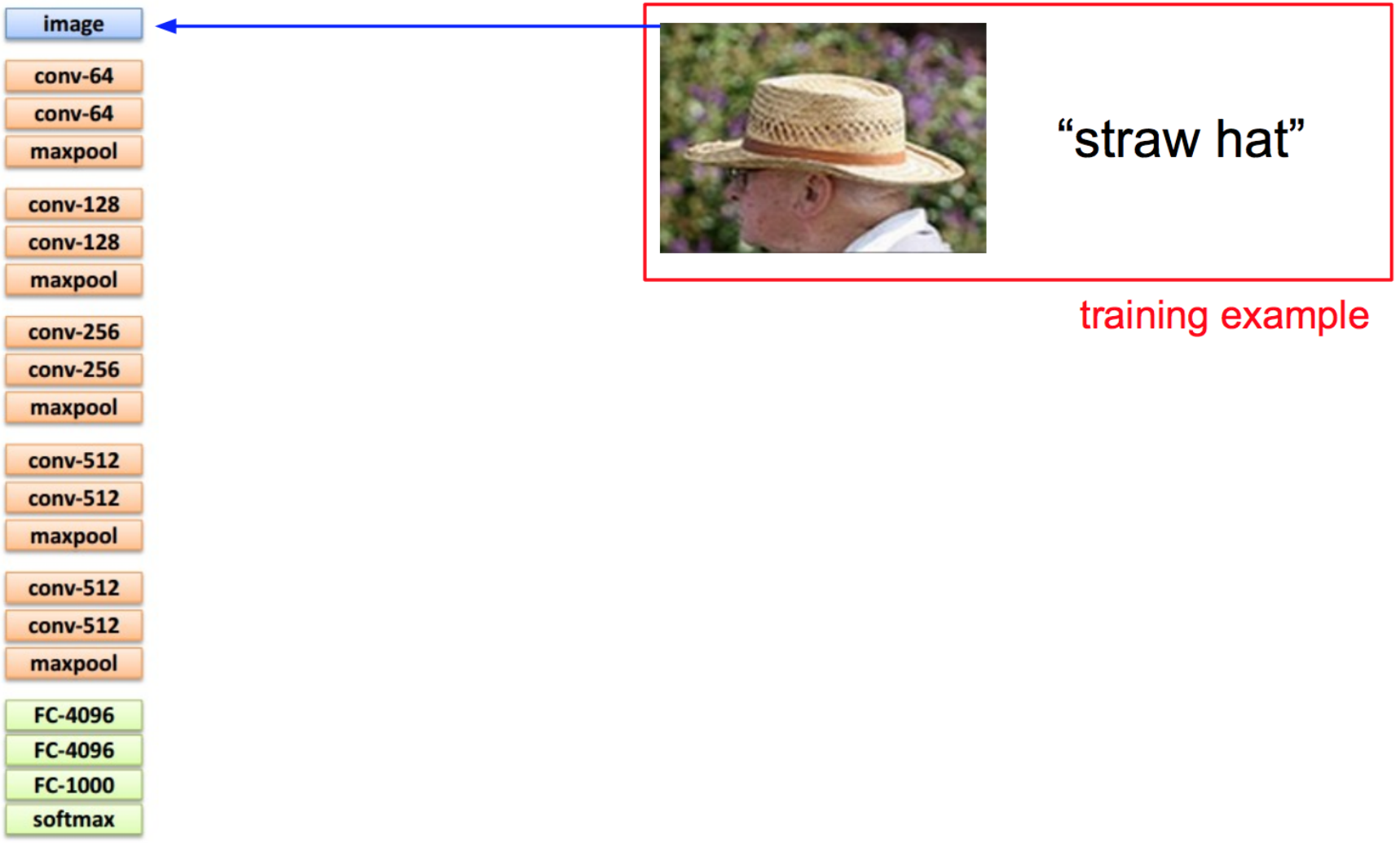
Bahdanau et al. 2014

Show and Tell

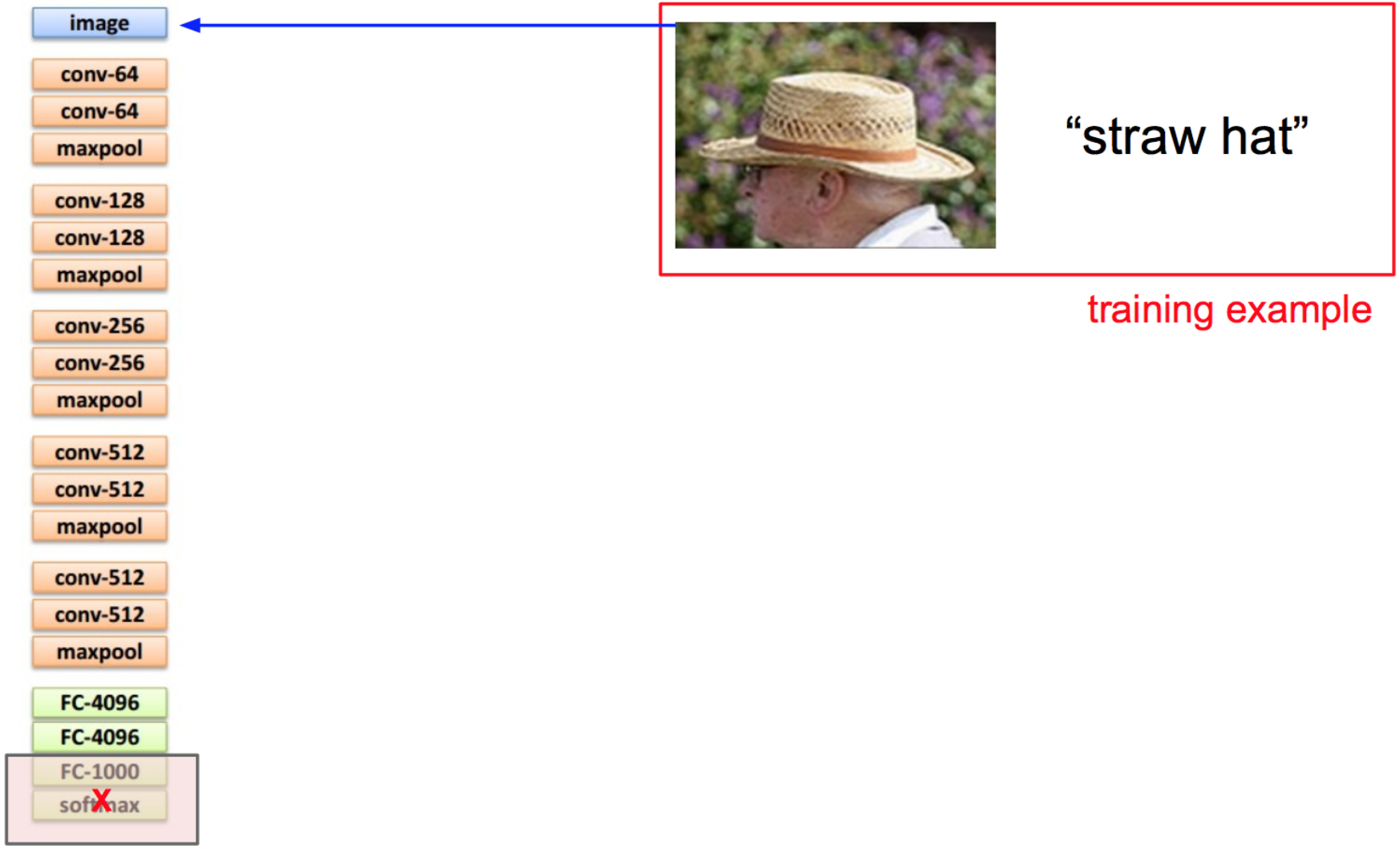
- Neural Image Caption(NIC)
  - CNN : 22 layer GoogLeNet
  - LSTM for modeling
  - Word embedding(word2vec)



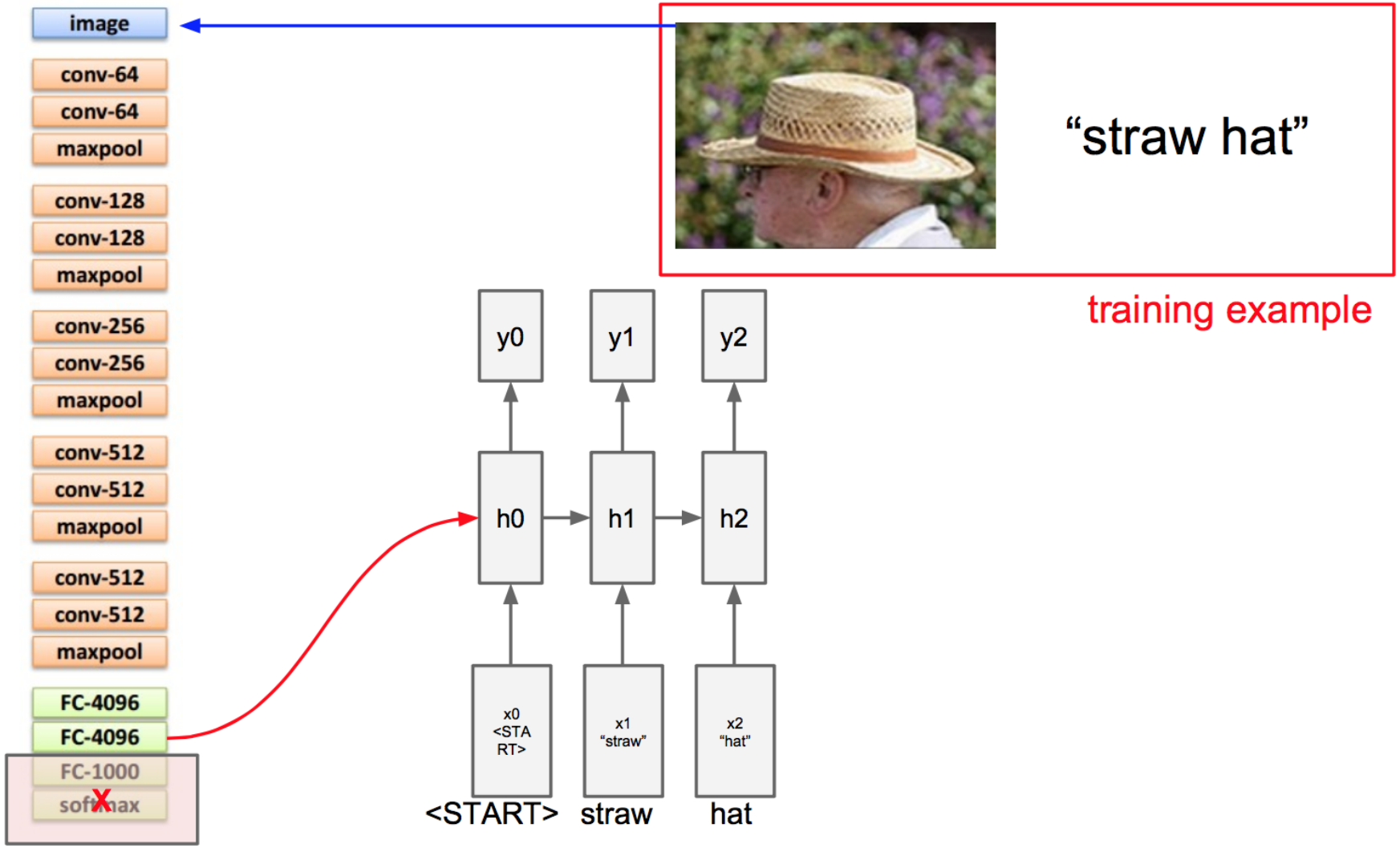
Training phase(example)



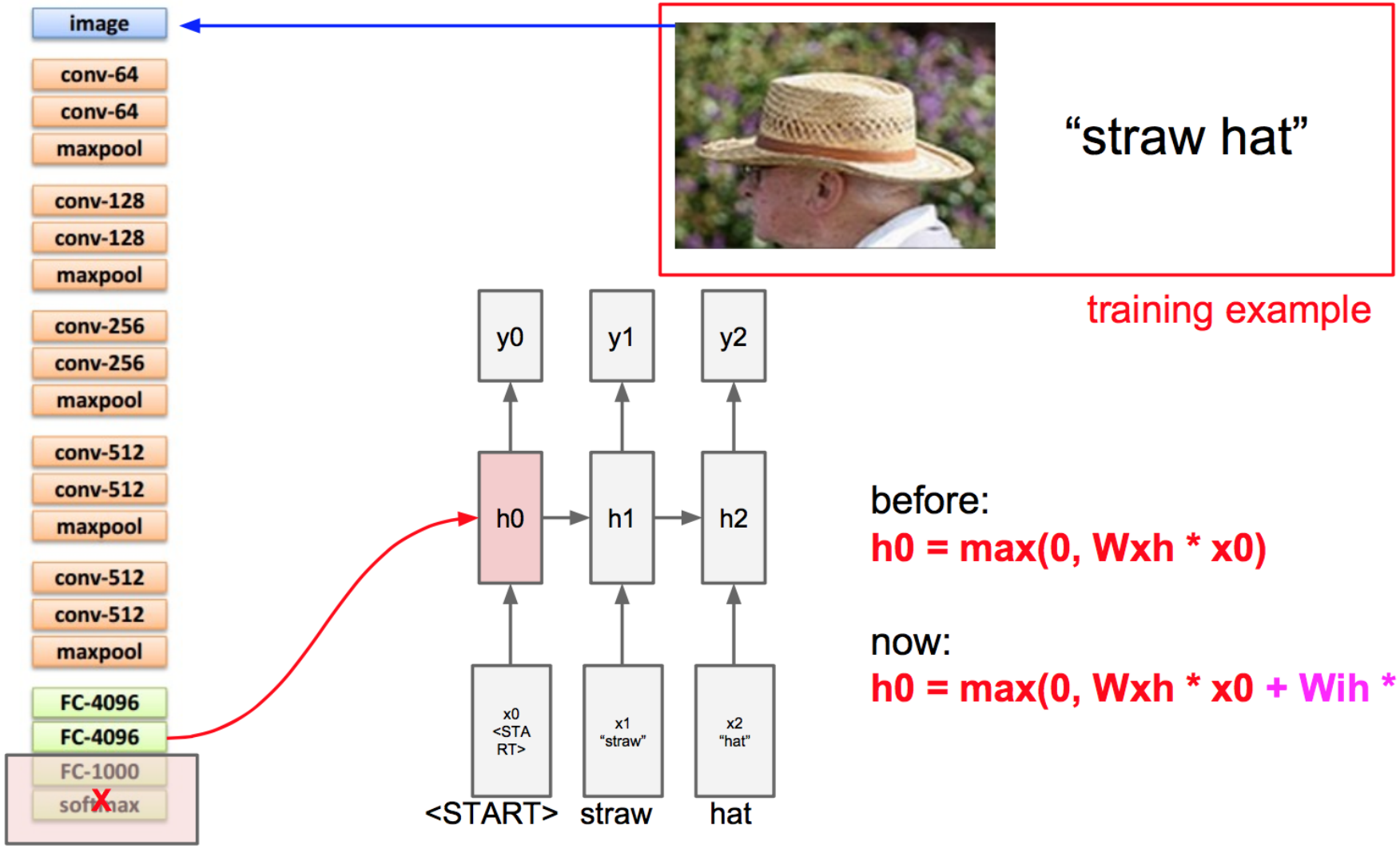
Training phase



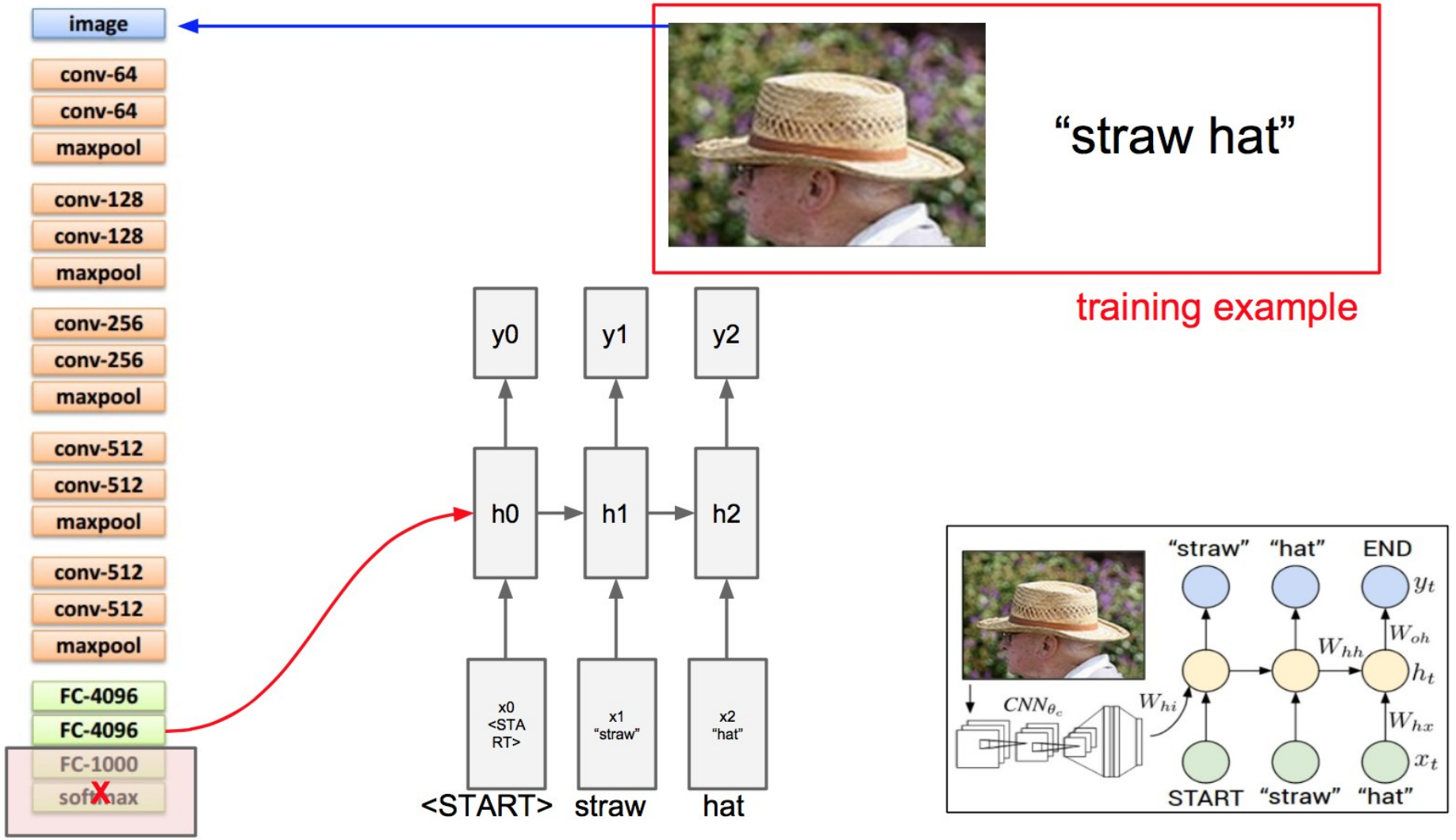
Training phase



Training phase



Training phase



Test phase

test image

Test phase

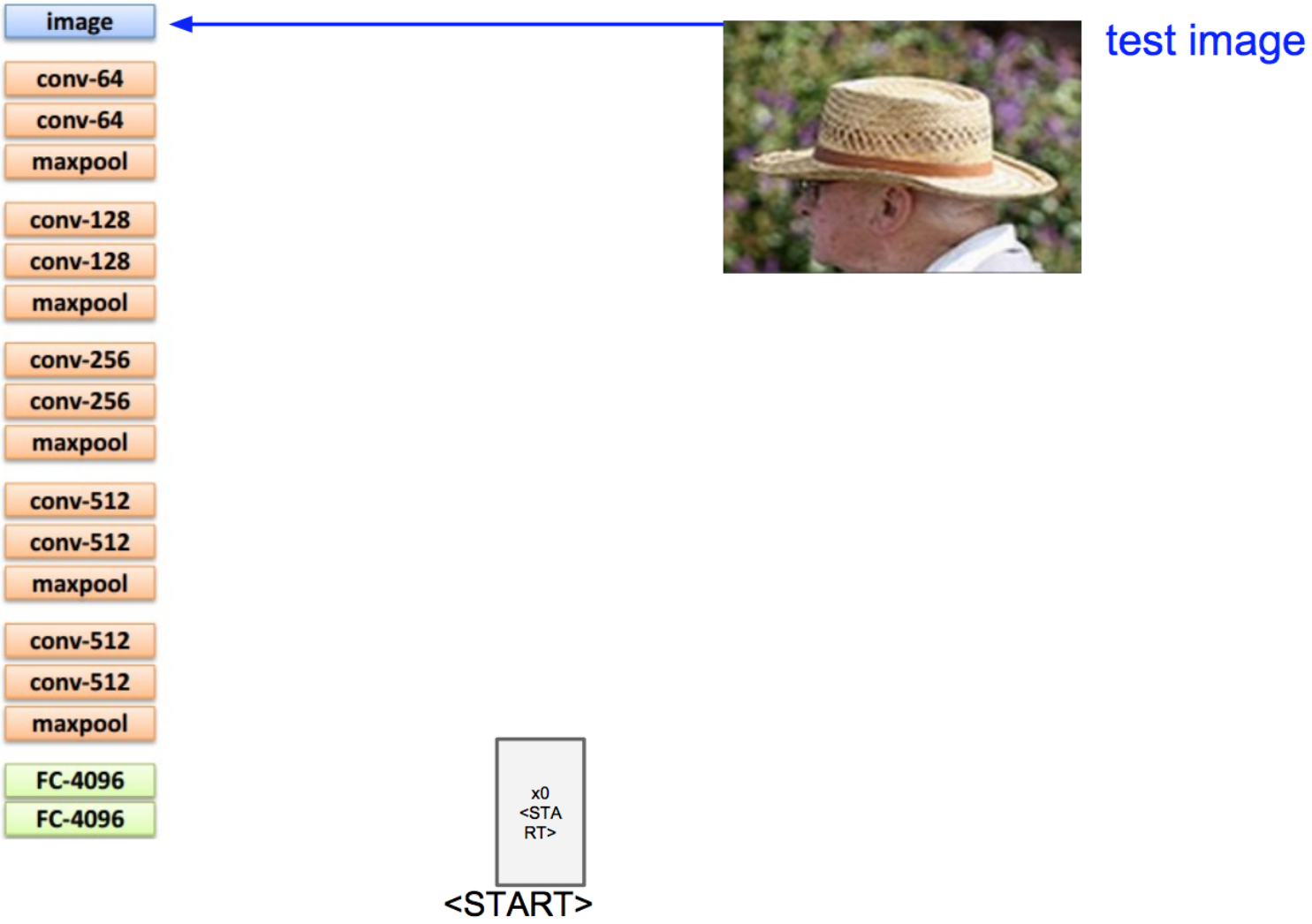


image
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096
test image
x0
<STA
RT>
<START>

Test phase

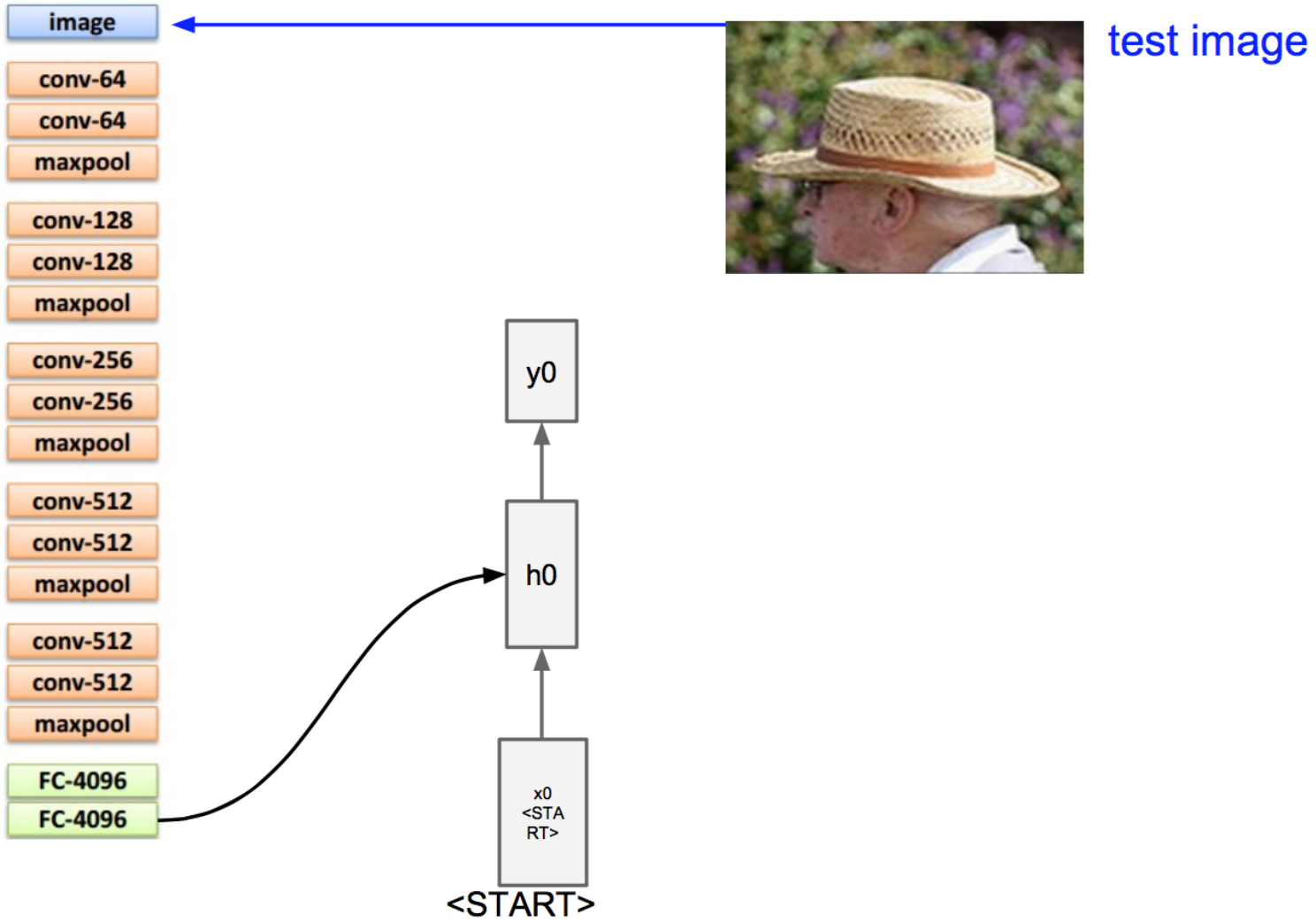


image
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096
y0
h0
x0
<STA
RT>
<START>
test image

Test phase

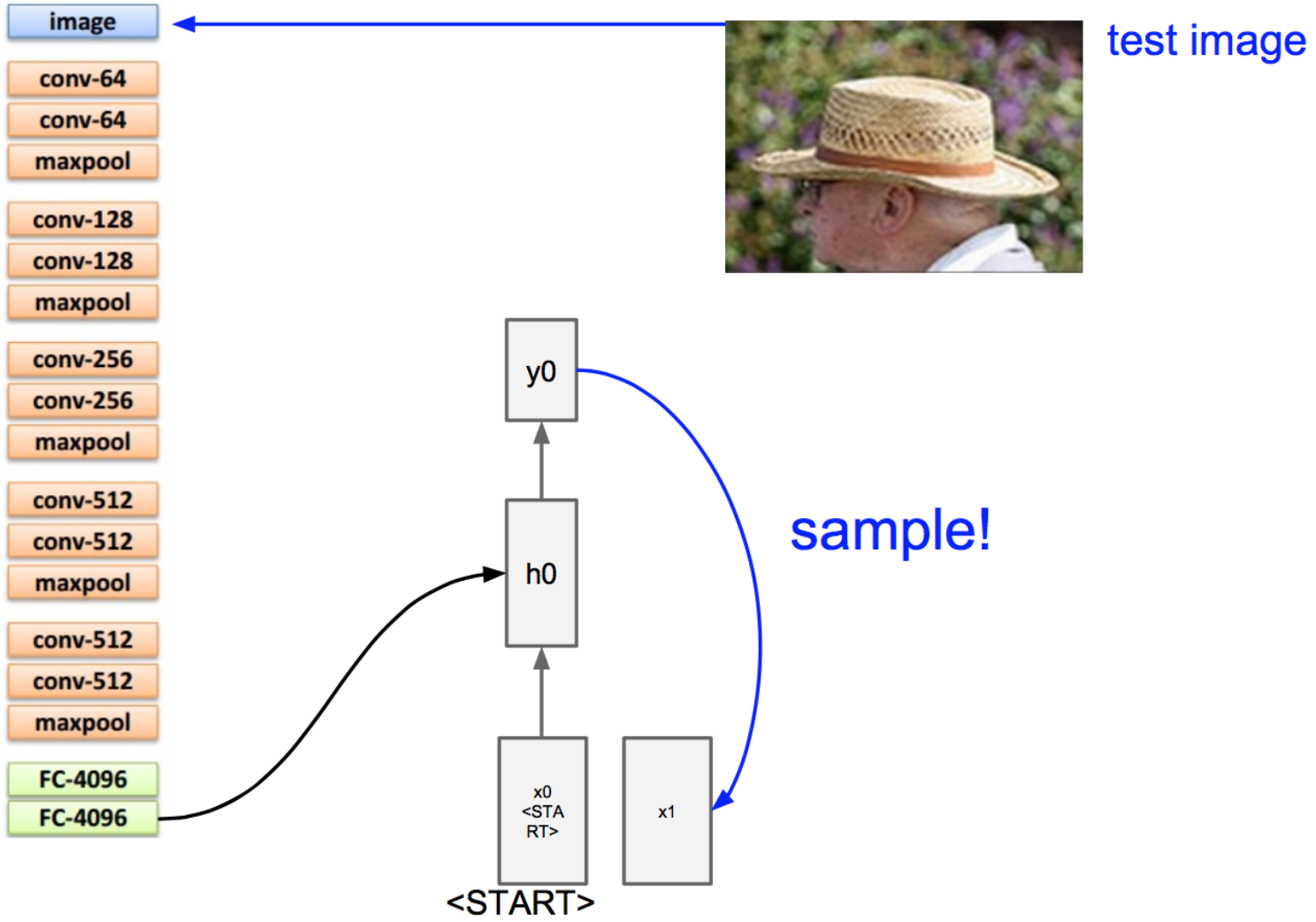


image
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096
test image
y0
h0
x0
<STA
RT>
x1
<START>
sample!

Test phase

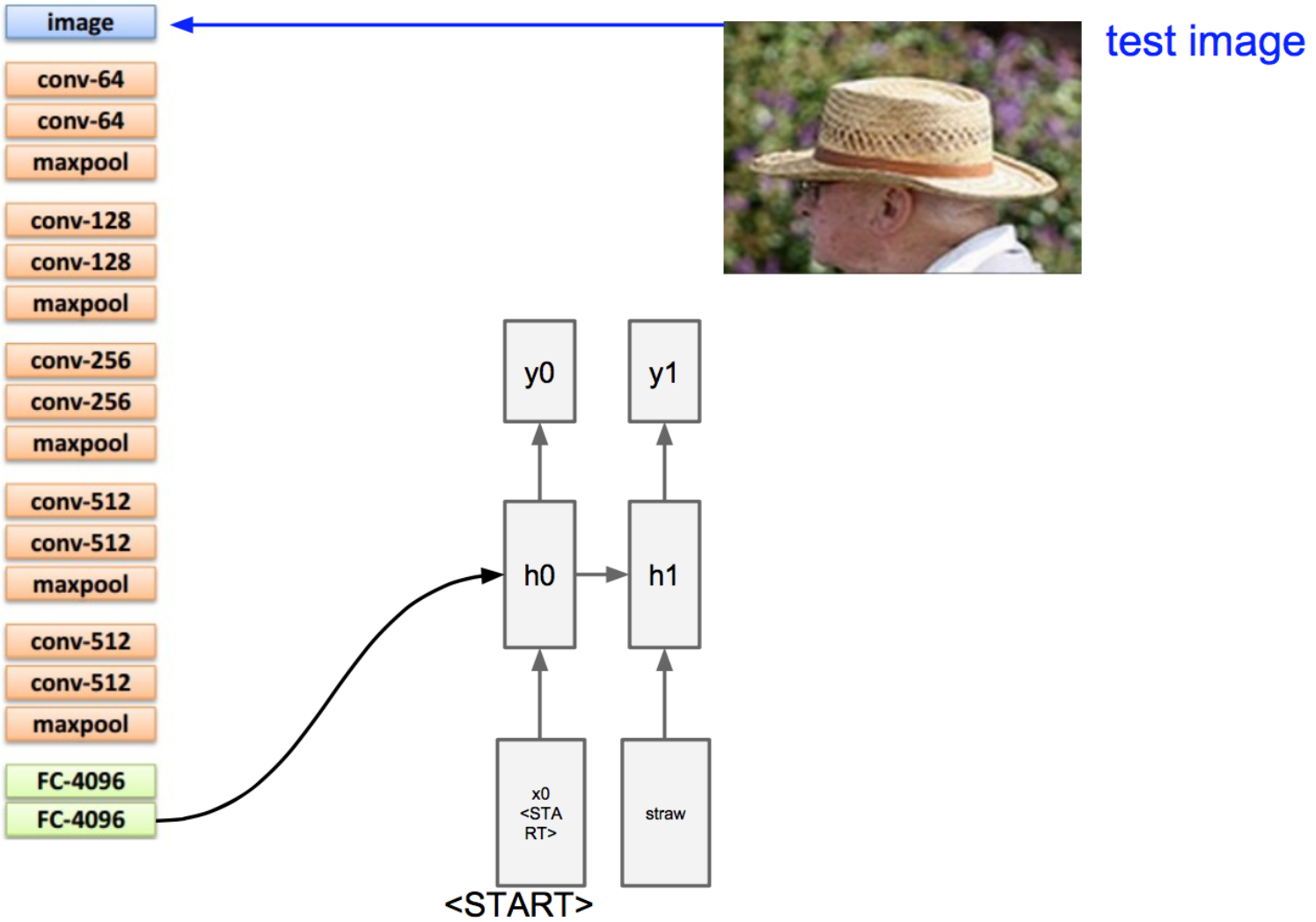


image
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096
test image
y0
y1
h0
h1
x0
<STA
RT>
straw
<START>

Test phase

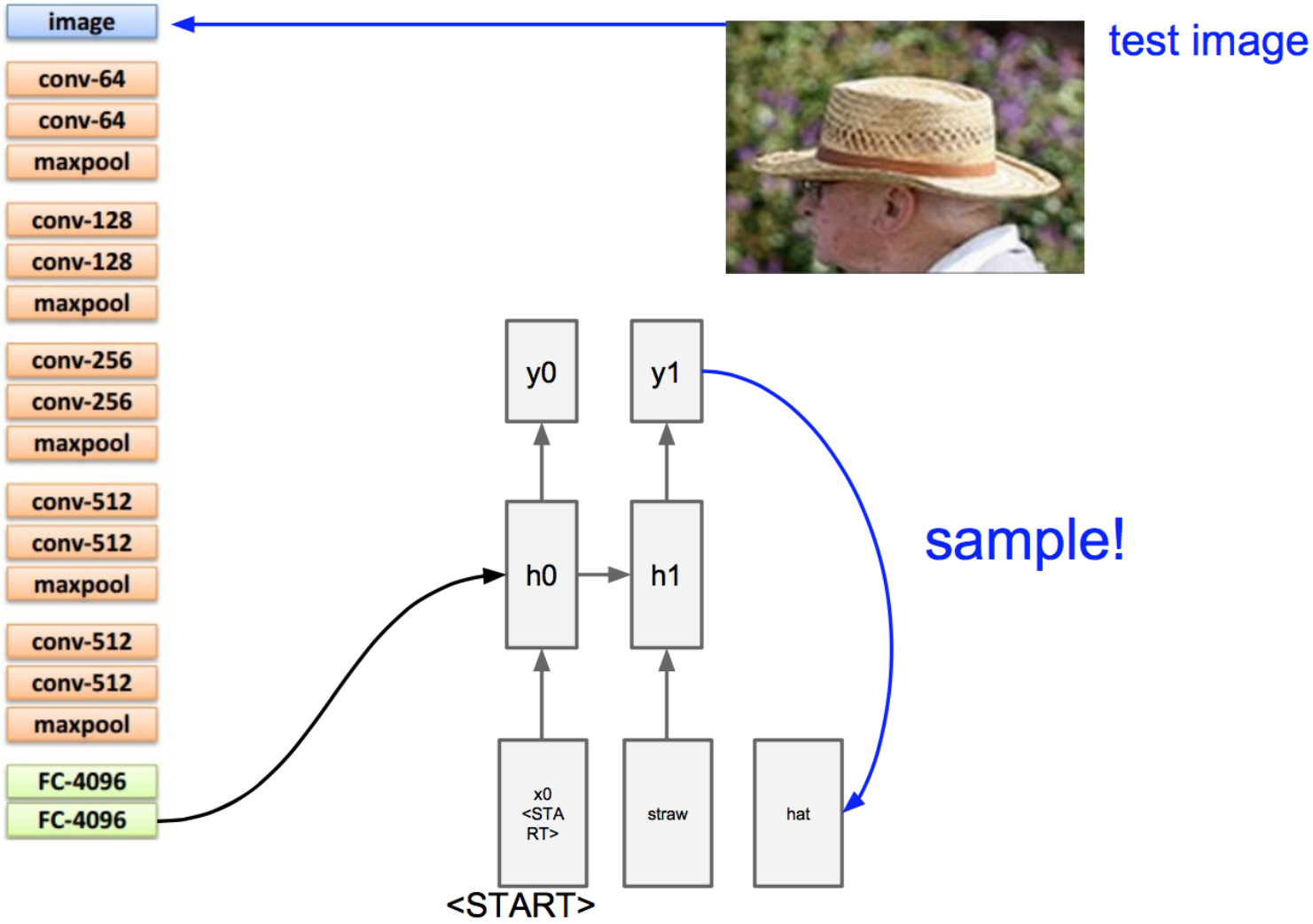


image
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096
test image
y0
y1
h0
h1
x0
<STA
RT>
straw
hat
<START>
sample!

Test phase

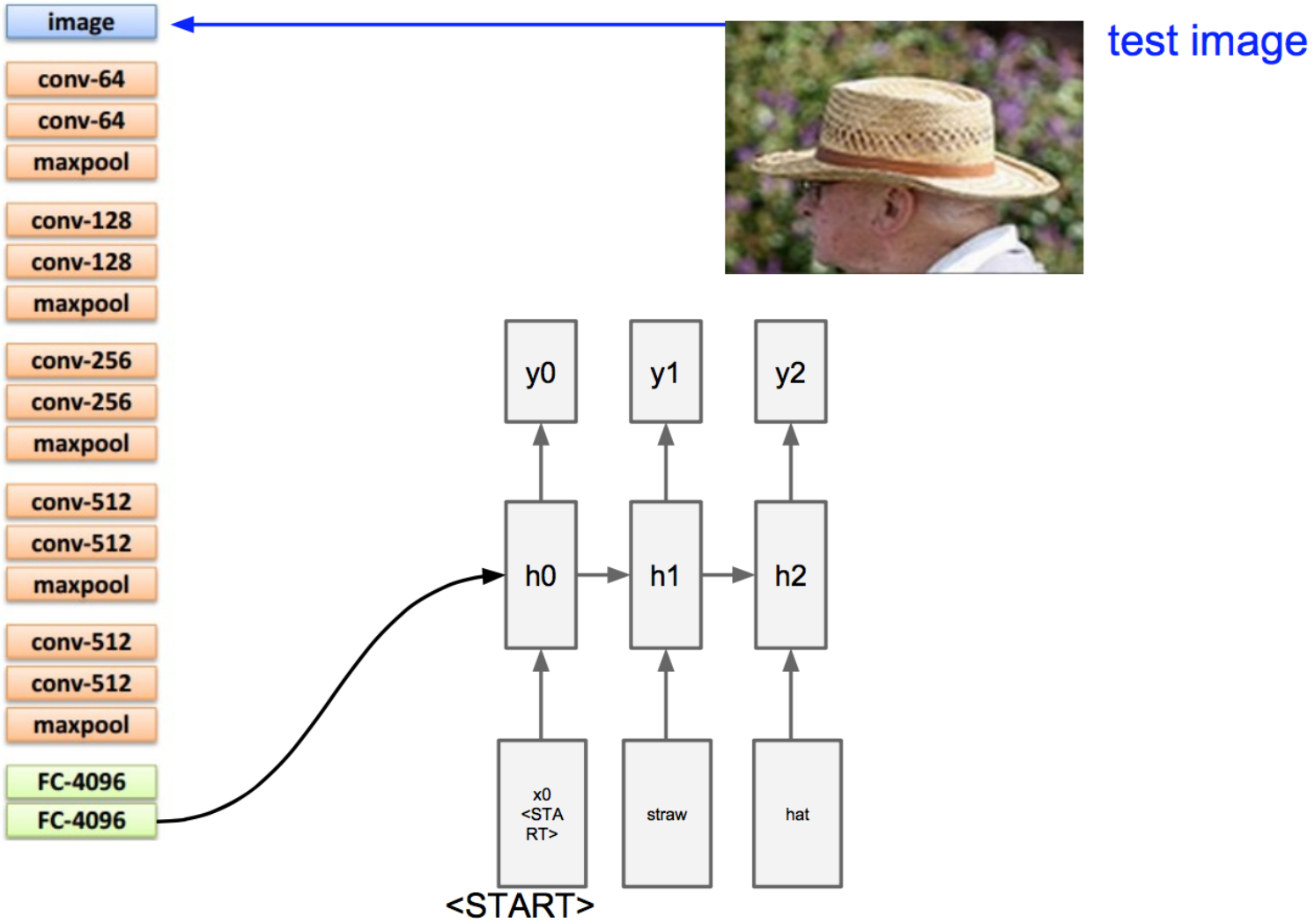


image
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096
test image
y0
y1
y2
h0
h1
h2
x0 <START>
straw
hat
<START>

Test phase

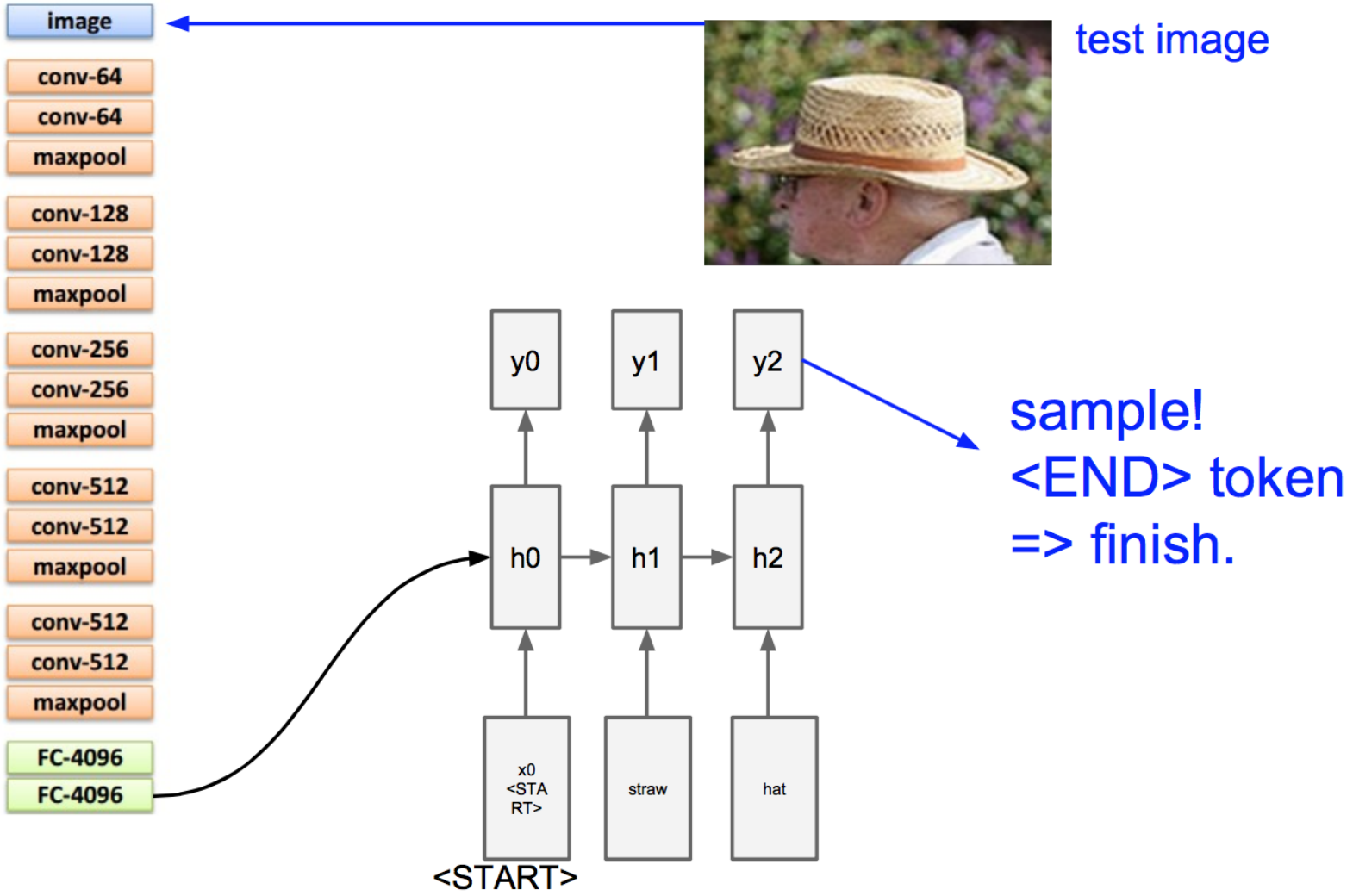
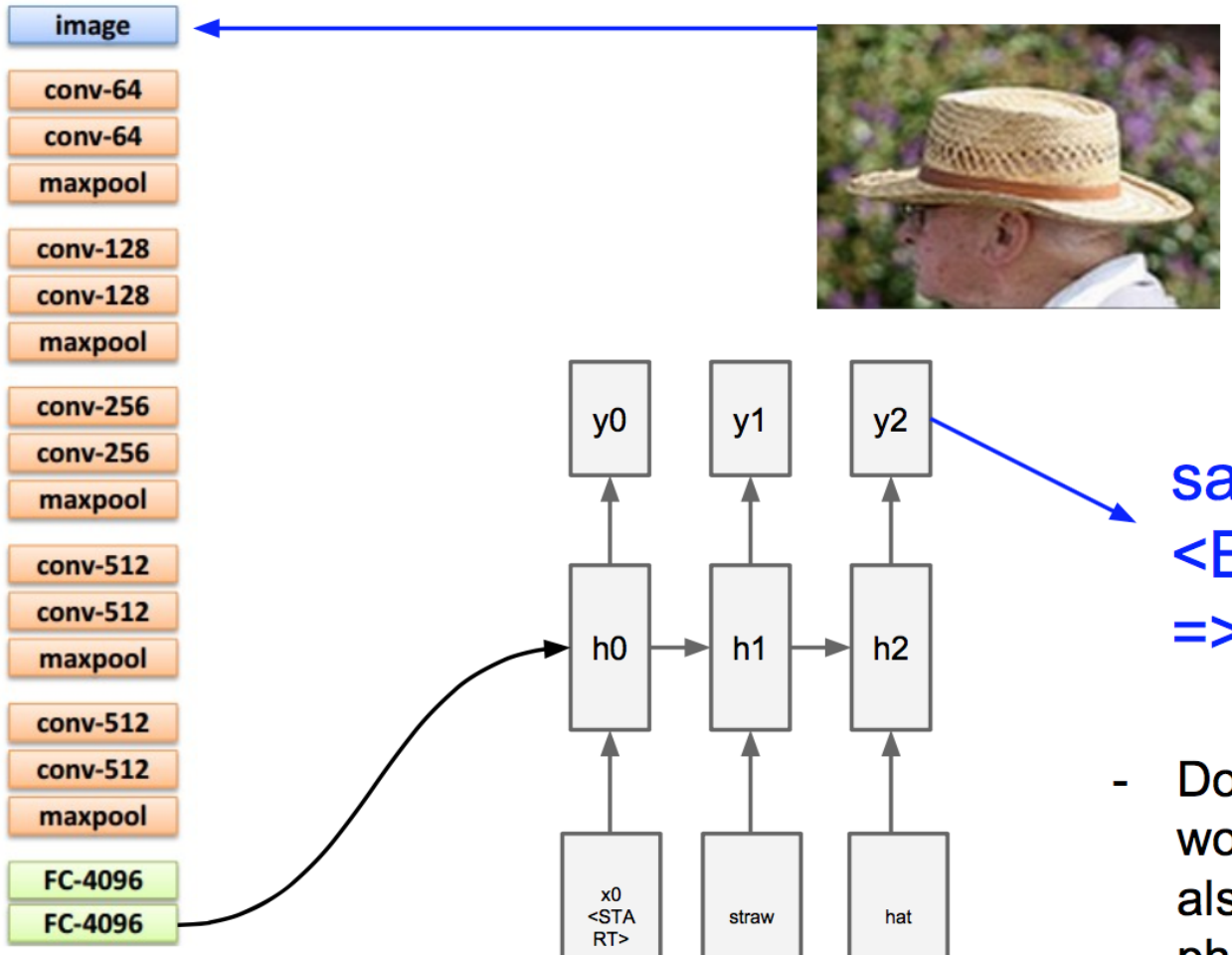


image
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096
test image
y0
y1
y2
h0
h1
h2
x0 <START>
straw
hat
<START>
sample!
<END> token
=> finish.

Test phase



test image

sample!
<END> token
=> finish.

- Don't have to do greedy word-by-word sampling, can also search over longer phrases with **beam search**

Image Sentence Datasets

a man riding a bike on a dirt path through a forest.
bicyclist raises his fist as he rides on desert dirt trail.
this dirt bike rider is smiling and raising his fist in triumph.
a man riding a bicycle while pumping his fist in the air.
a mountain biker pumps his fist in celebration.

Microsoft COCO

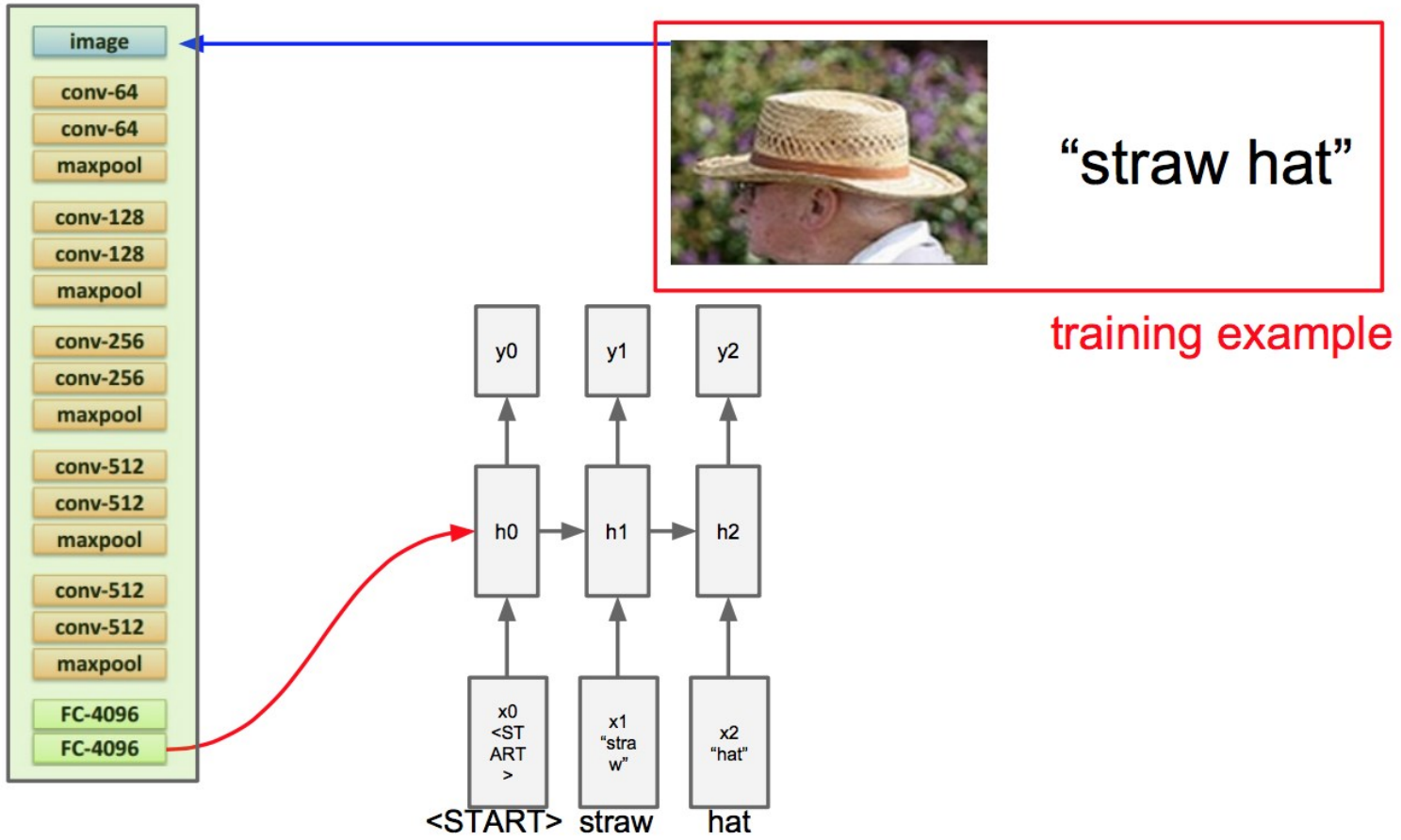
[Tsung-Yi Lin et al. 2014]

mscoco.org

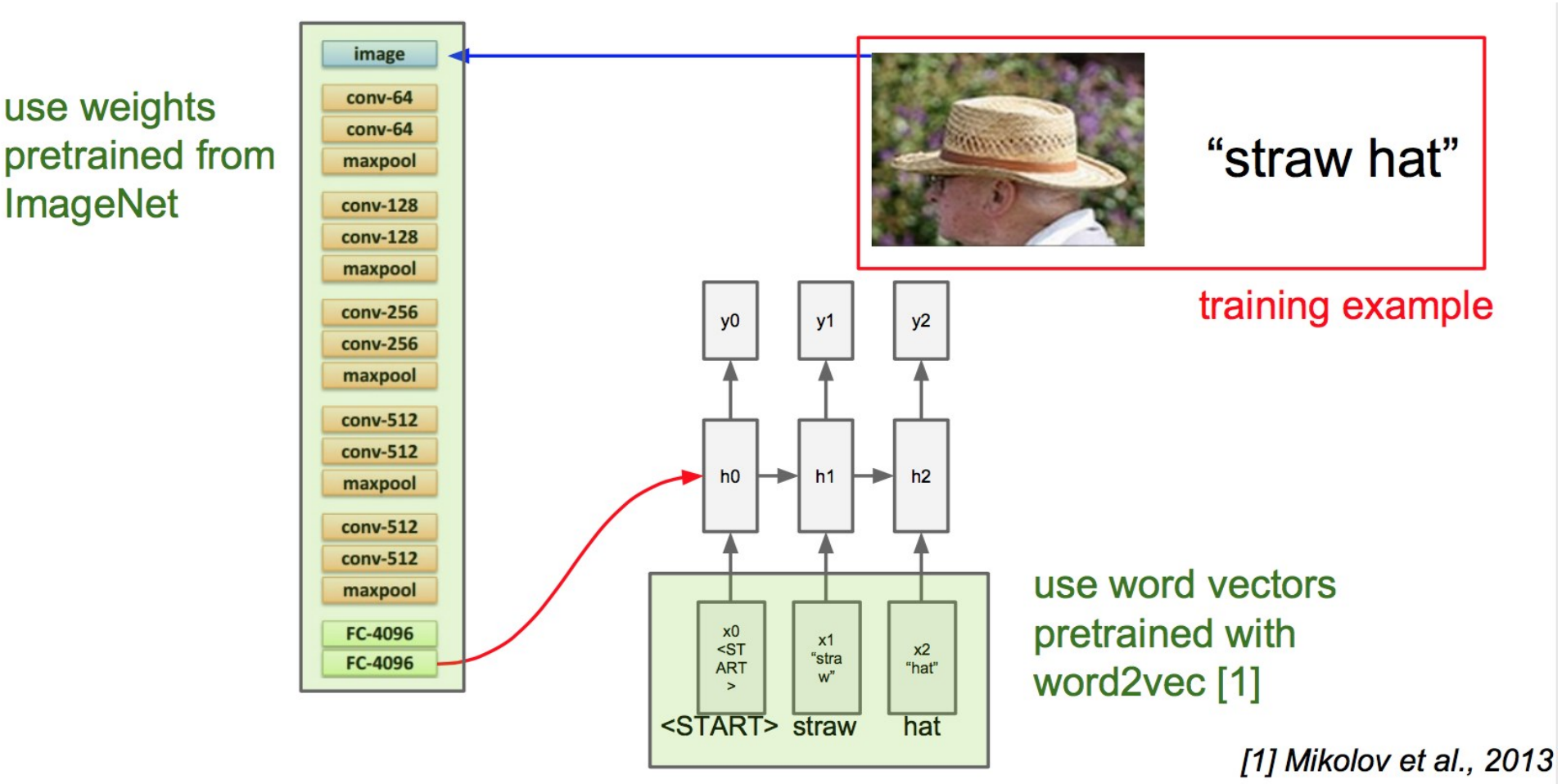
currently:
~120K images
~5 sentences each

Transfer learning

use weights pretrained from ImageNet



Transfer learning



Results

A person riding a motorcycle on a dirt road.

Two dogs play in the grass.

A skateboarder does a trick on a ramp.

A dog is jumping to catch a frisbee.

A group of young people playing a game of frisbee.

Two hockey players are fighting over the puck.

A little girl in a pink hat is blowing bubbles.

A refrigerator filled with lots of food and drinks.

A herd of elephants walking across a dry grass field.

A close up of a cat laying on a couch.

A red motorcycle parked on the side of the road.

A yellow school bus parked in a parking lot.

Describes without errors | Describes with minor errors | Somewhat related to the image | Unrelated to the image

Show, Attend and Tell

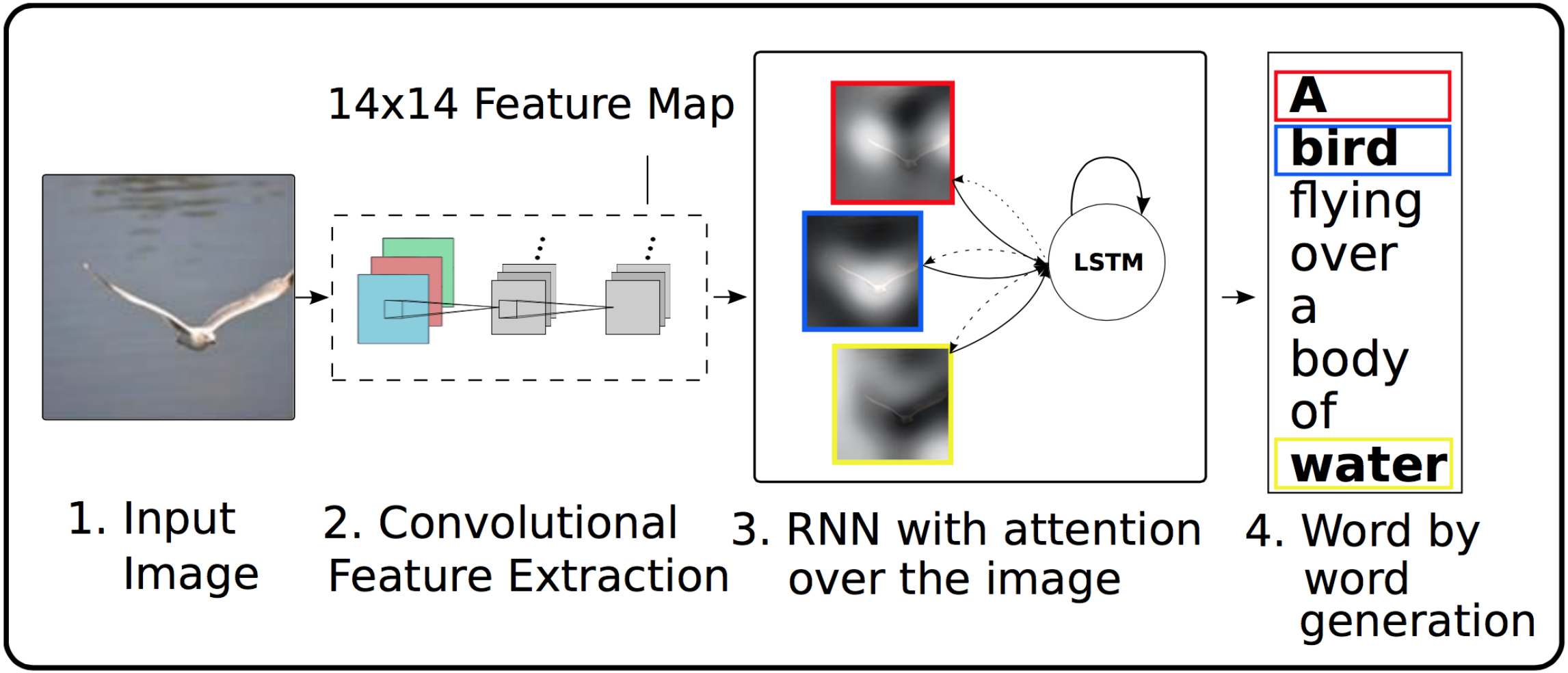
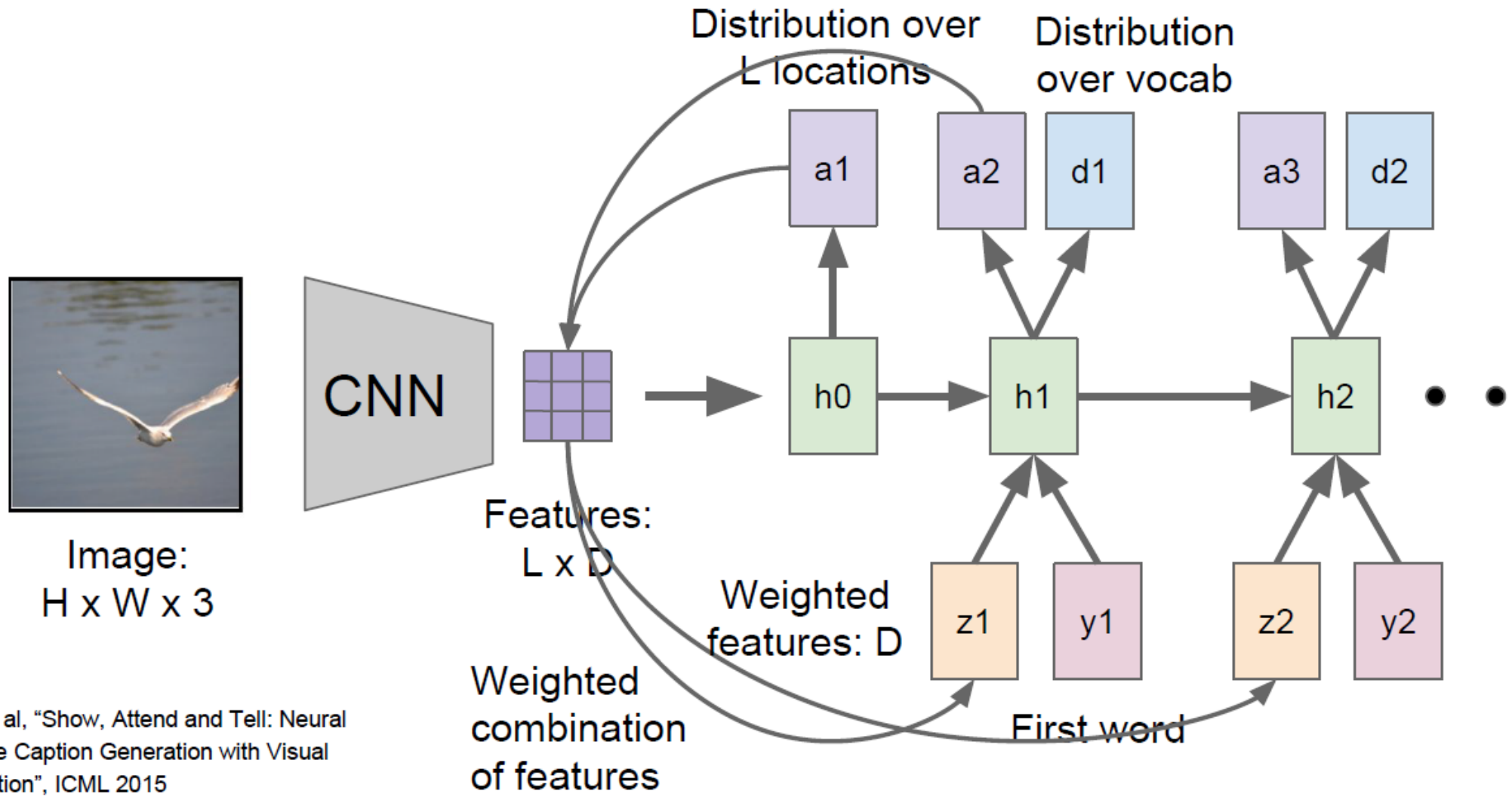


Image Captioning with Attention



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

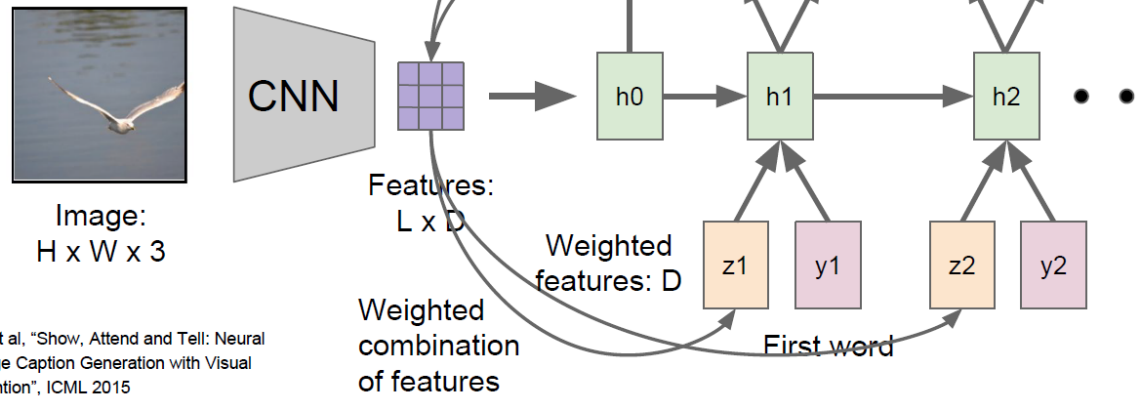
Image Captioning with Attention

$$e_{ti} = f_{\text{att}}(\mathbf{a}_i, \mathbf{h}_{t-1}) \quad (4)$$

$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^{L} \exp(e_{tk})}. \quad (5)$$

Once the weights (which sum to one) are computed, the context vector $\hat{z}_t$ is computed by

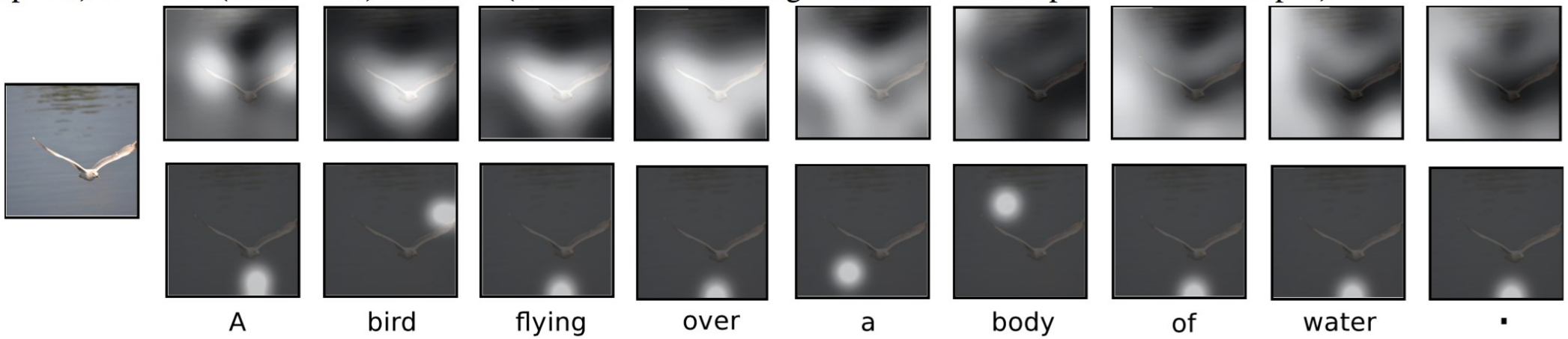
$$\hat{\mathbf{z}}_t = \phi\left(\{\mathbf{a}_i\}, \{\alpha_i\}\right), \quad (6)$$



u et al, "Show, Attend and Tell: Neural
nage Caption Generation with Visual
ttention", ICML 2015

Results

Figure 2. Attention over time. As the model generates each word, its attention changes to reflect the relevant parts of the image. "soft" (top row) vs "hard" (bottom row) attention. (Note that both models generated the same captions in this example.)



Results

Figure 3. Examples of attending to the correct object (*white* indicates the attended regions, *underlines* indicated the corresponding word)

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.

Results (mistakes)

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.

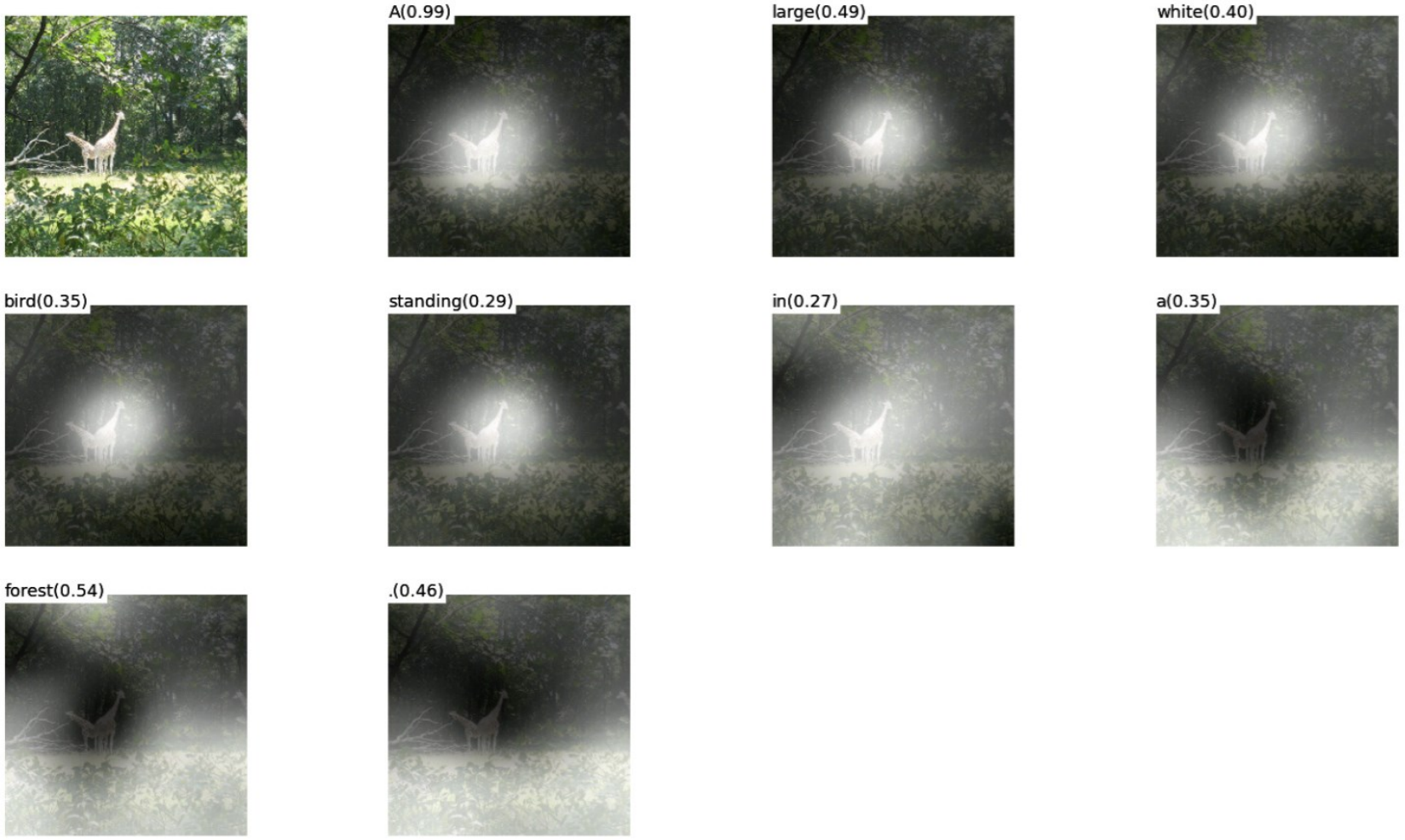
A woman is sitting at a table
with a large pizza.

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

Results

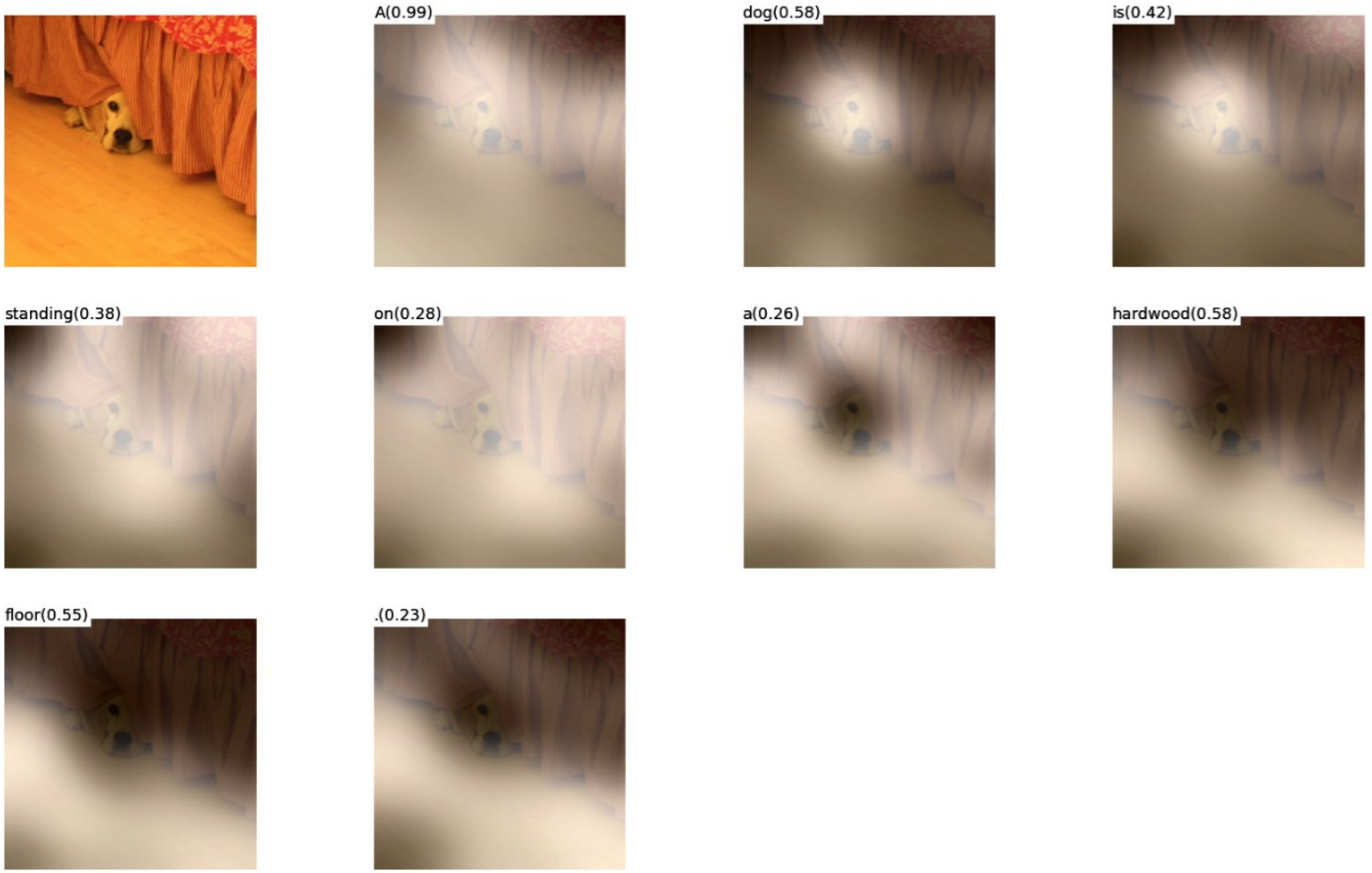
(b) A woman is throwing a frisbee in a park.

Results



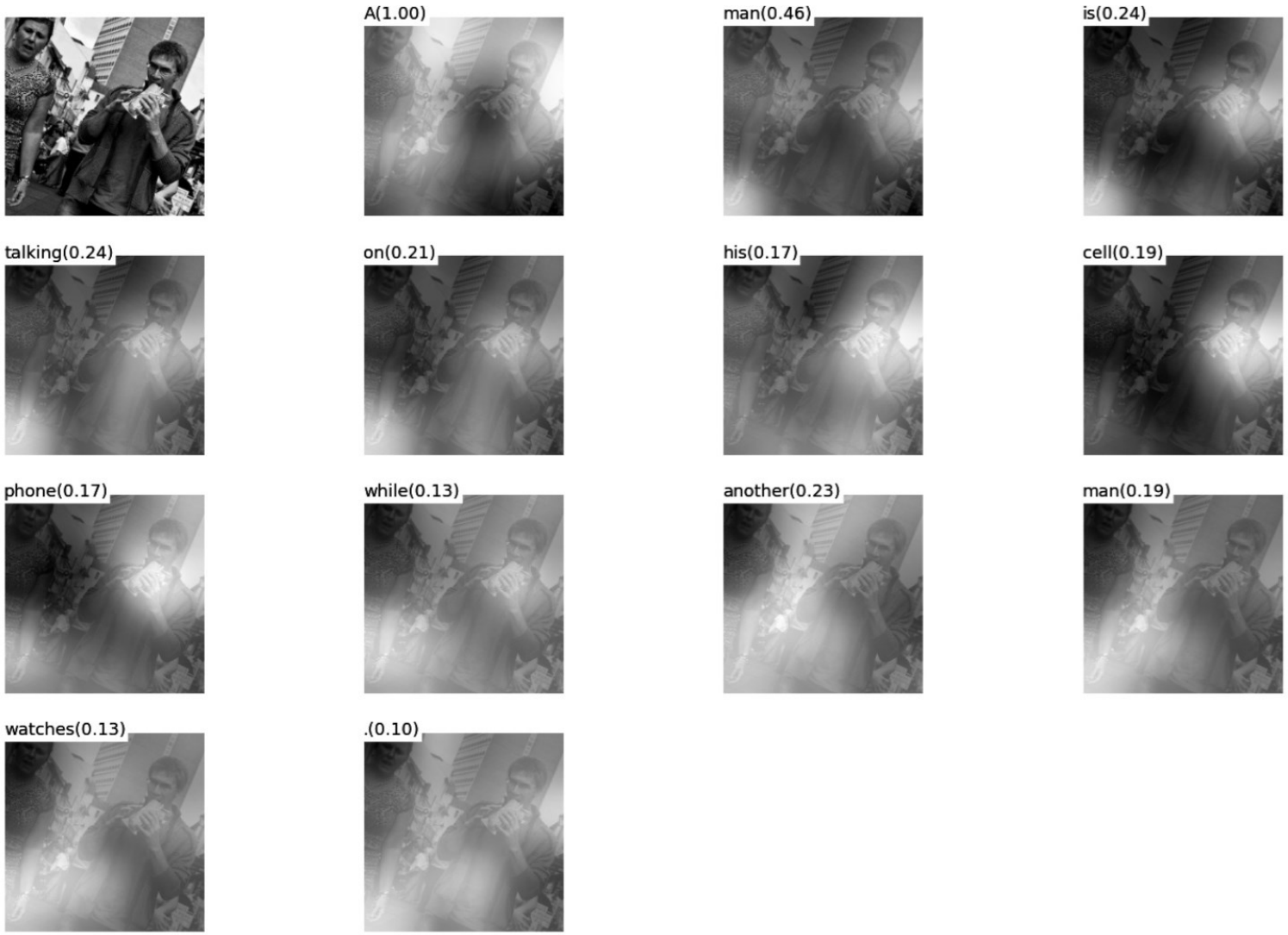
(b) A large white bird standing in a forest.

Results



(b) A dog is standing on a hardwood floor.

Results



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