

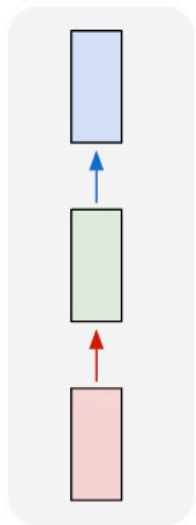
# Recurrent Neural Nets & Visual Captioning

## Lecture 7

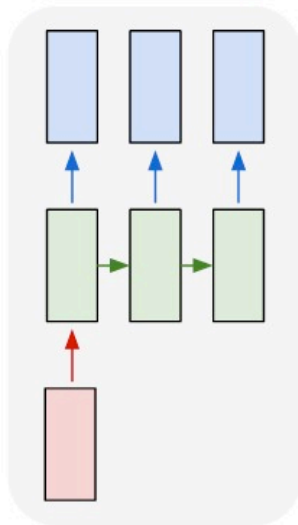
Slides from: Dhruv Bhatra, Fei-Fei Li, Justin Johnson,  
Serena Yeung, Andrej Karpathy

# Recurrent Neural Nets

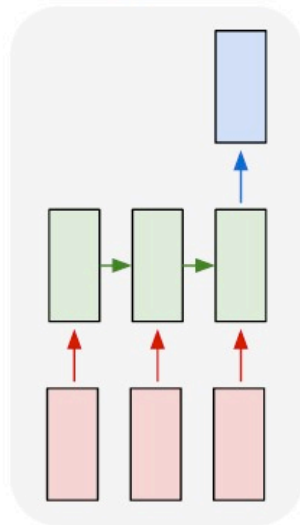
one to one



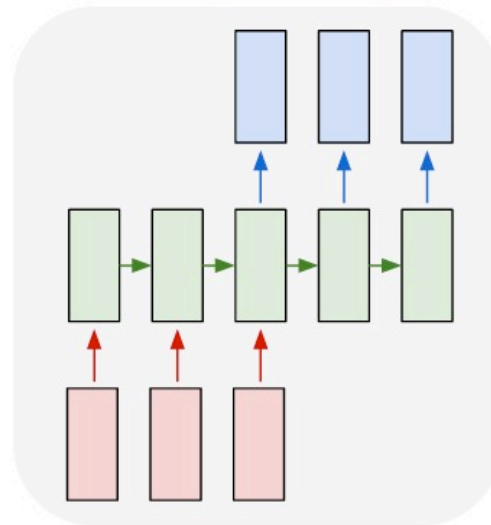
one to many



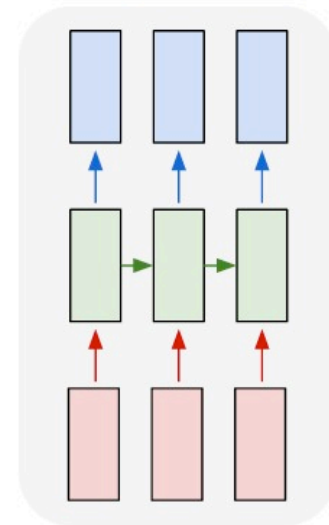
many to one



many to many



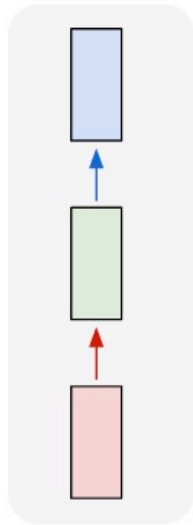
many to many





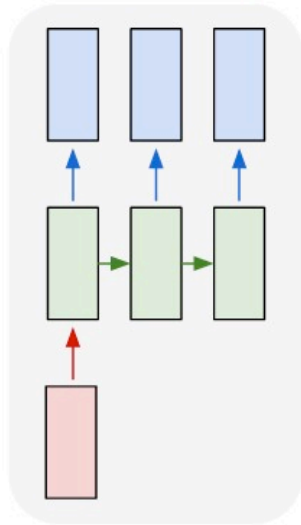
# Recurrent Neural Nets

one to one



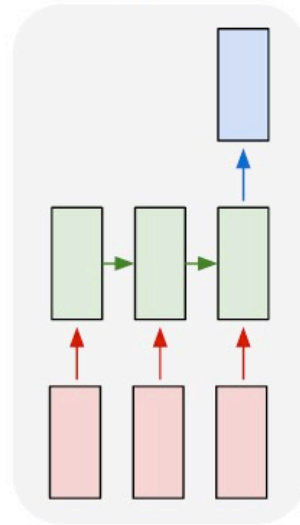
Input: No sequence  
Output: No sequence  
Example: "standard" classification / regression problems

one to many



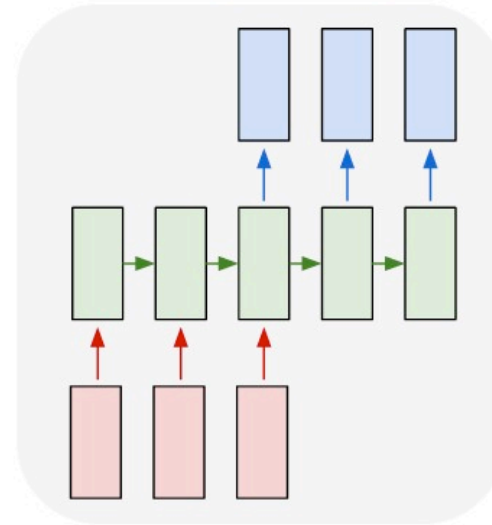
Input: No sequence  
Output: Sequence  
Example: Im2Caption

many to one



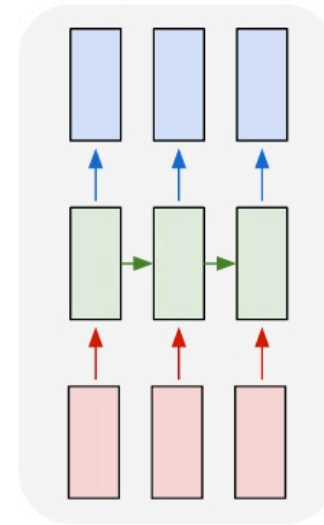
Input: Sequence  
Output: No sequence  
Example: sentence classification, multiple-choice question answering

many to many



Input: Sequence  
Output: Sequence  
Example: machine translation, video captioning, open-ended question answering, video question answering

many to many

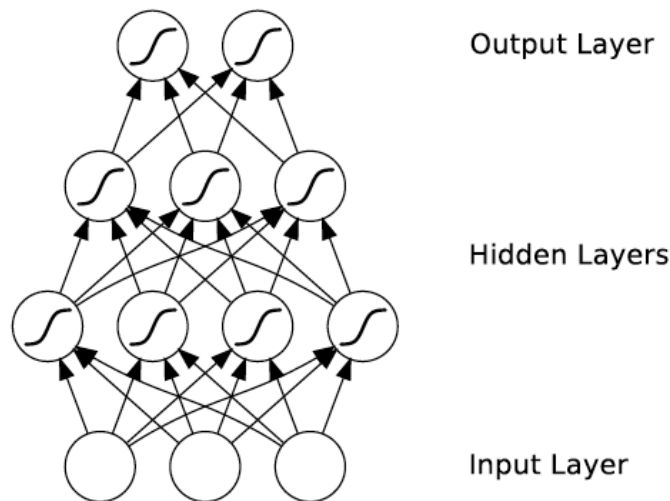


# Synonyms

- Recurrent Neural Networks (RNNs)
- Types:
  - “Vanilla” RNNs
  - Long Short Term Memory (LSTMs)
  - Gated Recurrent Units (GRUs)
  - ...
- Algorithms
  - BackProp Through Time (BPTT)

# What's wrong with MLPs/ConvNets?

- Problem 1: Can't model sequences
  - Fixed-sized Inputs & Outputs
  - No temporal structure
- Problem 2: Pure feed-forward processing
  - No “memory”, no feedback



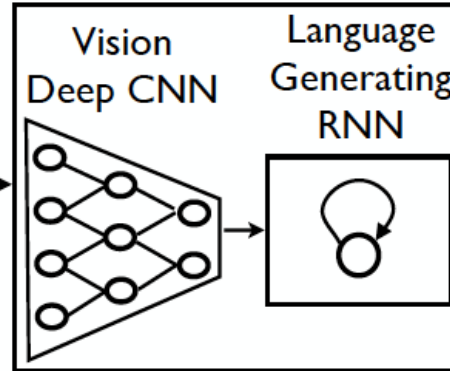
# Sequences are everywhere...

*Foreign Minister.* → FOREIGN MINISTER.

 → THE SOUND OF

$a_1=2$   $a_2=0$   $a_3=1$   $a_4=3$   $a_5=4$   $a_6=2$   $a_7=5$   
 $x =$  bringen sie bitte das auto zurück .  
↙  
 $y =$  please return the car .

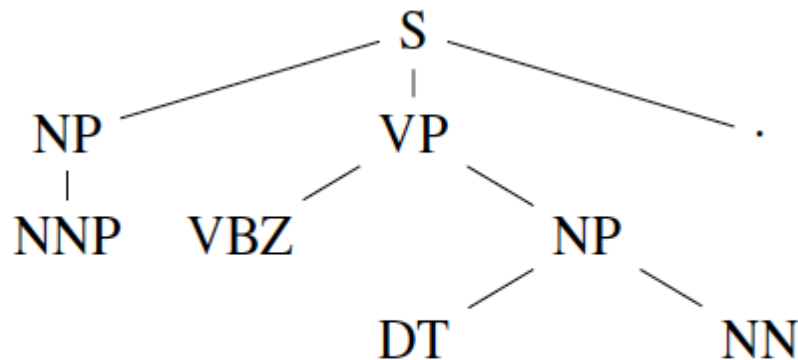
# Even where you might not expect a sequence...



**A group of people shopping at an outdoor market.**

**There are many vegetables at the fruit stand.**

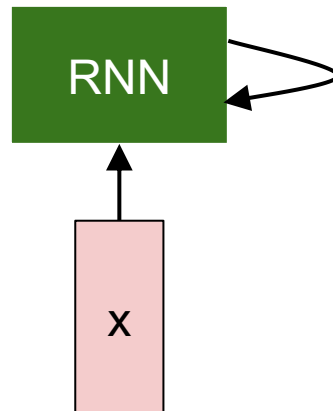
John has a dog .      →



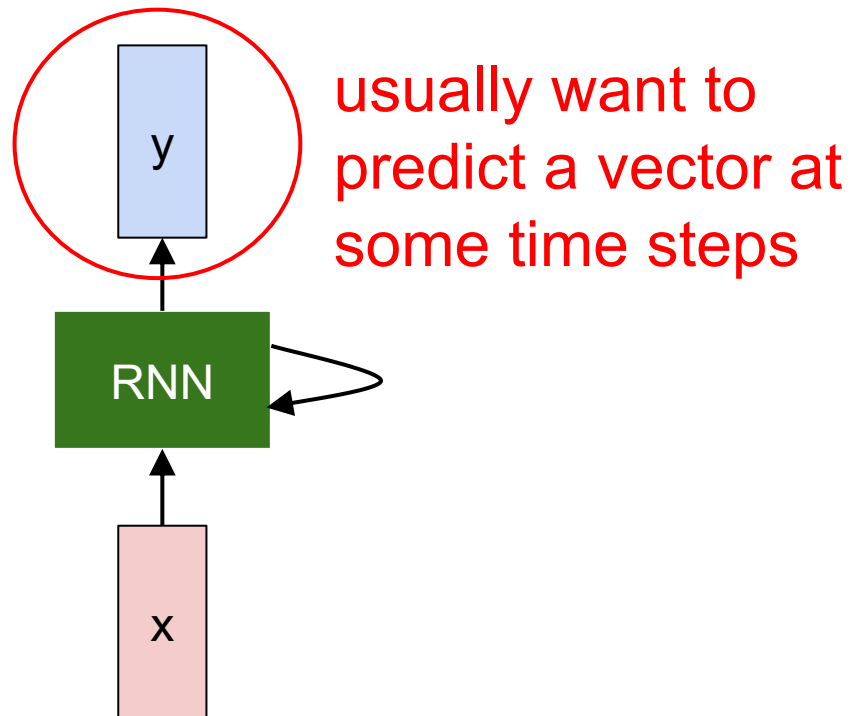
John has a dog .      →

$(S (NP NNP )_{NP} (VP VBZ (NP DT NN )_{NP} )_{VP} . )_S$

# Recurrent Neural Network



# Recurrent Neural Network



# Recurrent Neural Network

We can process a sequence of vectors  $\mathbf{x}$  by applying a **recurrence formula** at every time step:

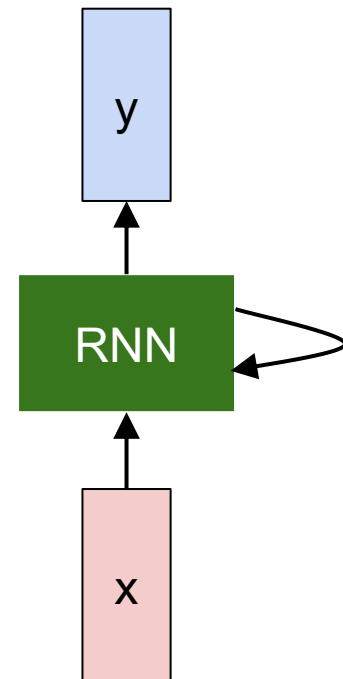
$$\boxed{h_t} = \boxed{f_W}(\boxed{h_{t-1}}, \boxed{x_t})$$

new state

some function with parameters  $W$

old state

input vector at some time step



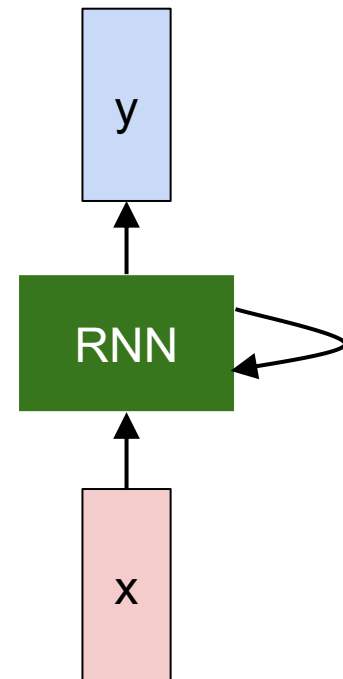


# Recurrent Neural Network

We can process a sequence of vectors  $\mathbf{x}$  by applying a **recurrence formula** at every time step:

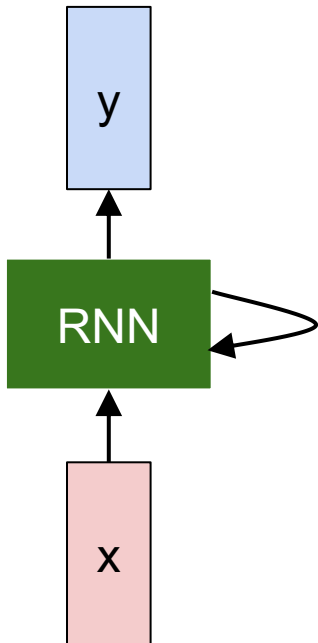
$$h_t = f_W(h_{t-1}, x_t)$$

Notice: the same function and the same set of parameters are used at every time step.



# (Vanilla) Recurrent Neural Network

The state consists of a single “*hidden*” vector  $\mathbf{h}$ :



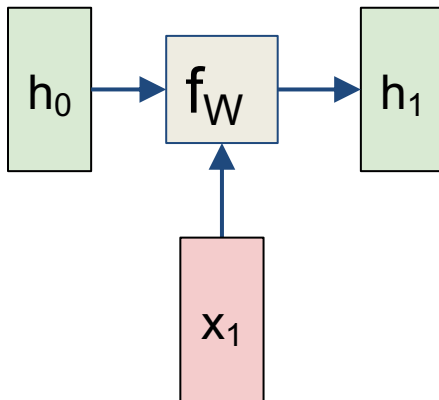
$$y_t = W_{hy}h_t + b_y$$

$$h_t = f_W(h_{t-1}, x_t)$$

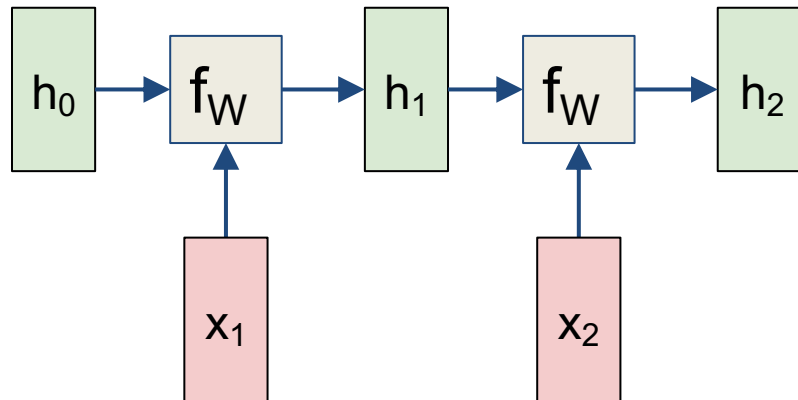


$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t + b_h)$$

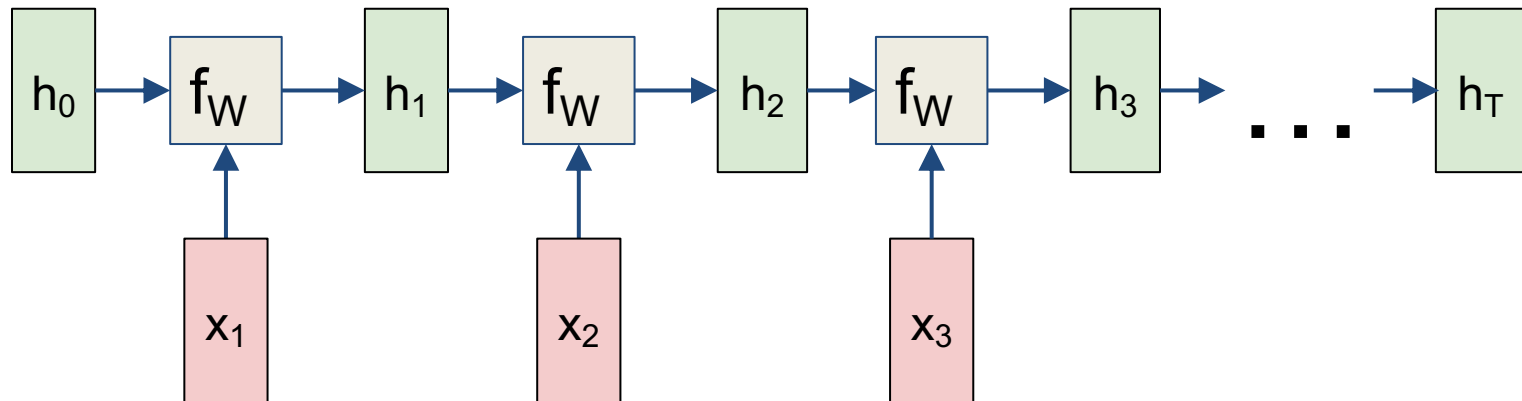
# RNN: Computational Graph



# RNN: Computational Graph

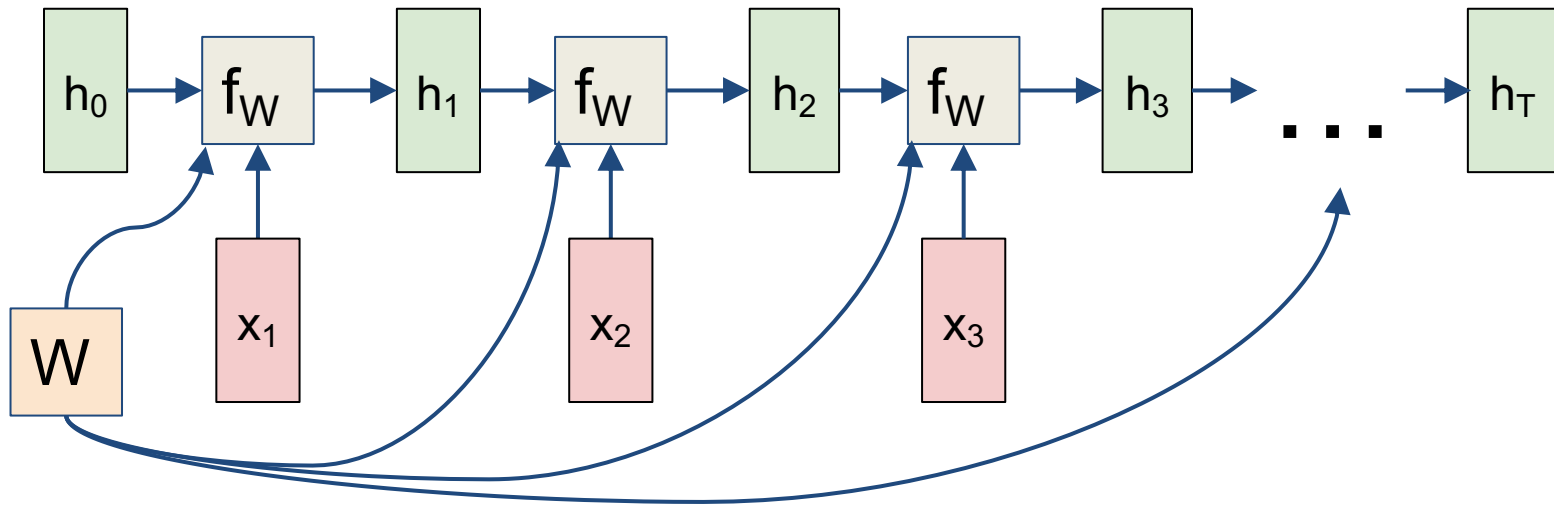


# RNN: Computational Graph

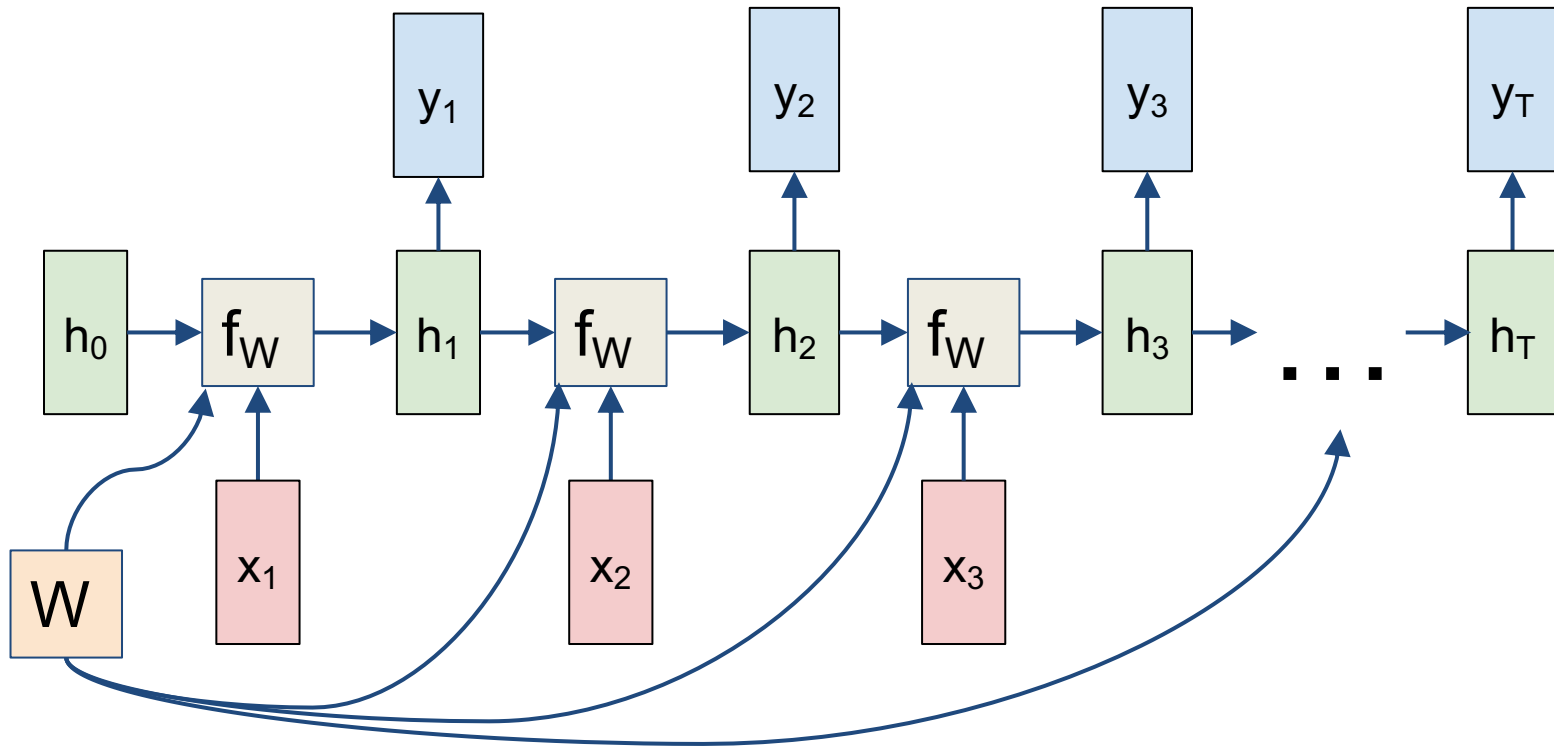


# RNN: Computational Graph

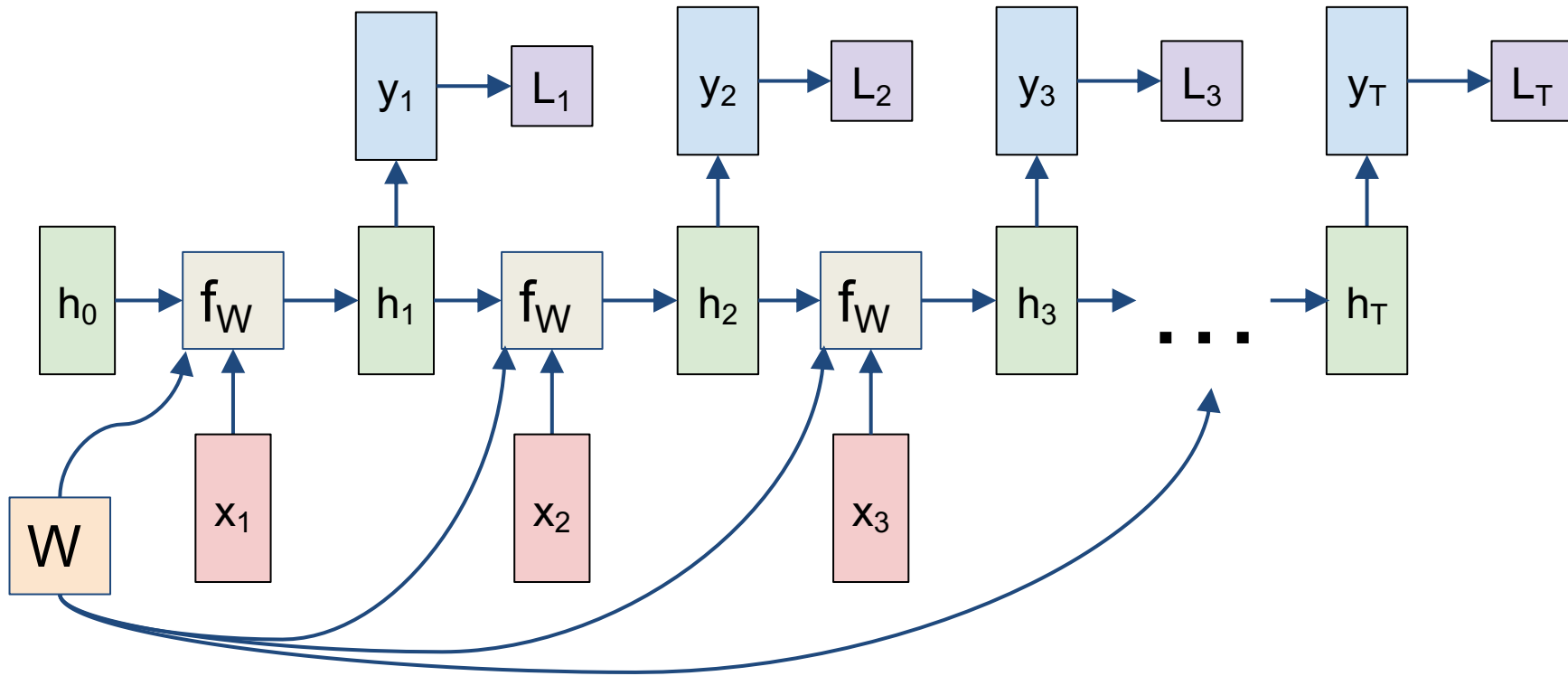
Re-use the same weight matrix at every time-step



# RNN: Computational Graph: Many to Many

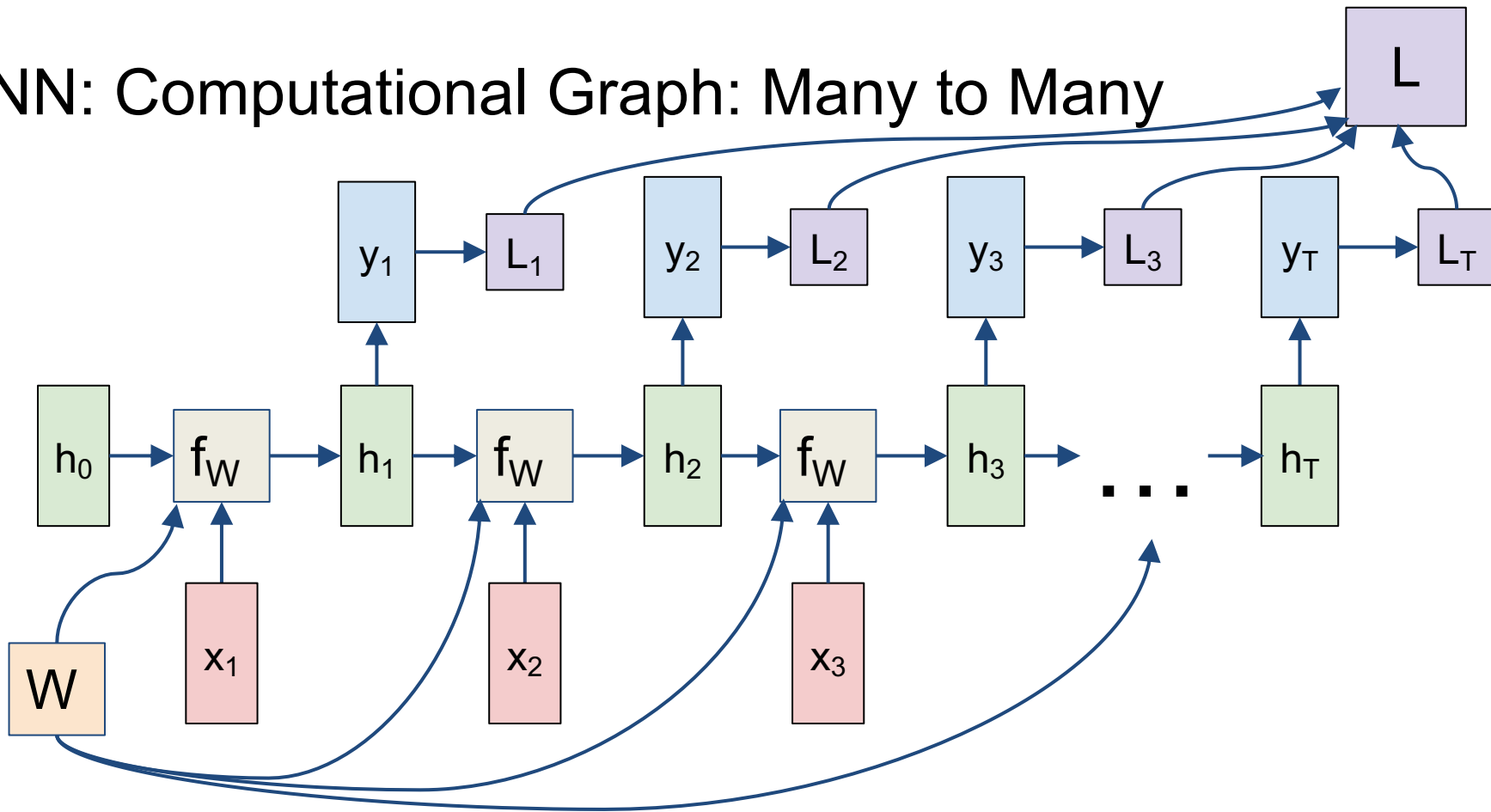


# RNN: Computational Graph: Many to Many

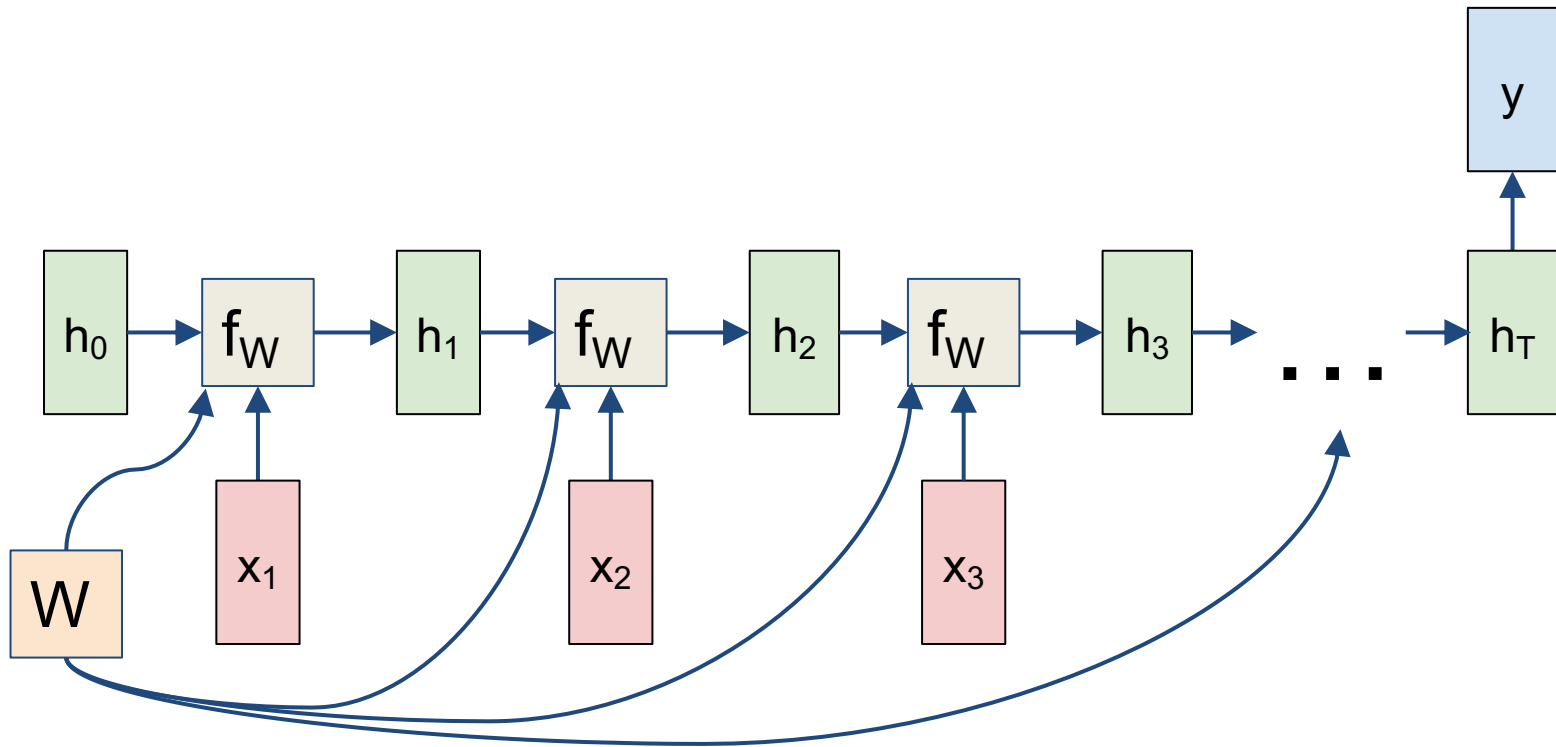




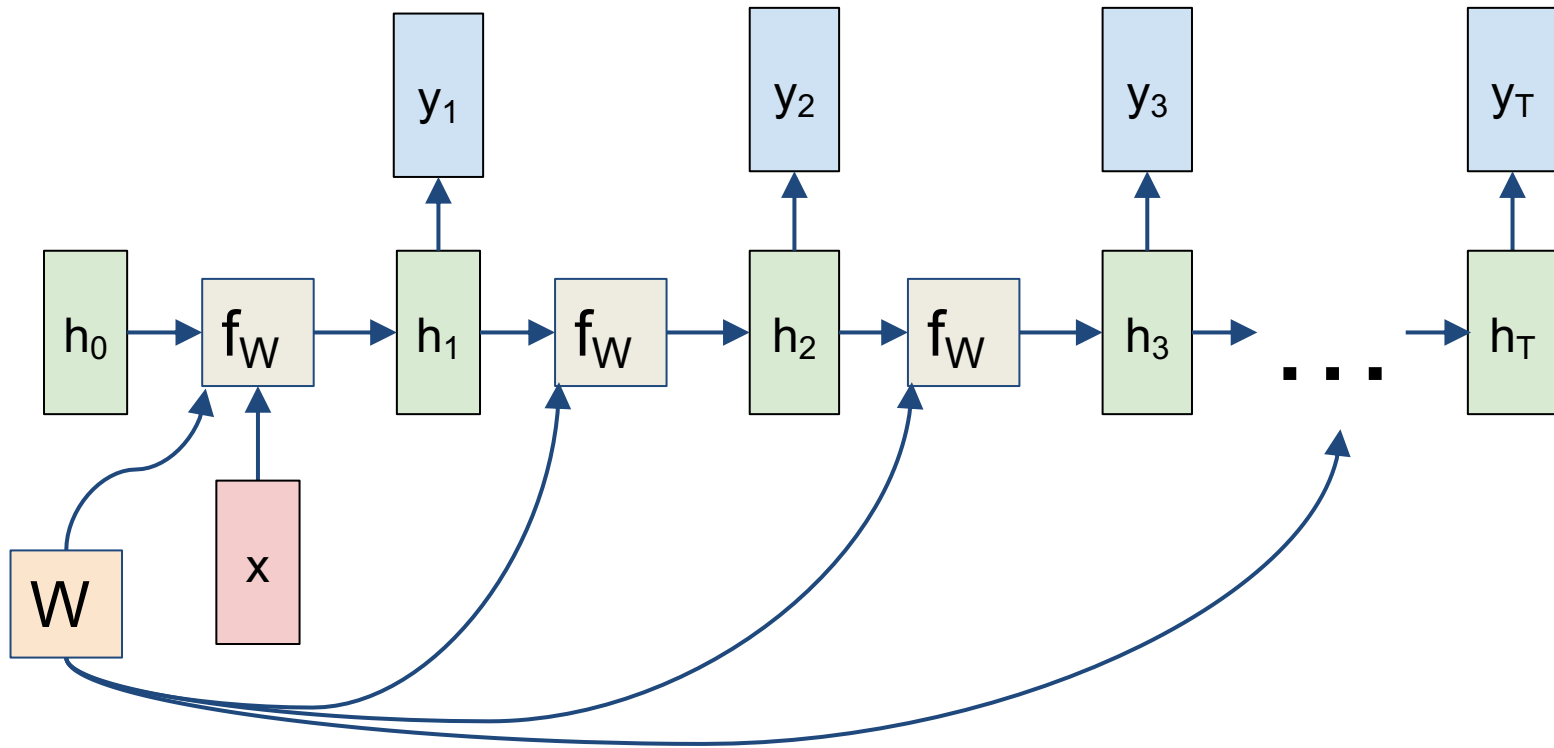
# RNN: Computational Graph: Many to Many



# RNN: Computational Graph: Many to One

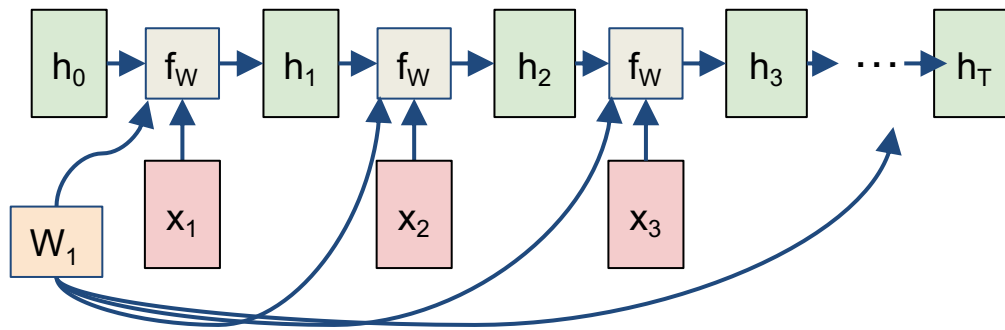


# RNN: Computational Graph: One to Many



# Sequence to Sequence: Many-to-one + one-to-many

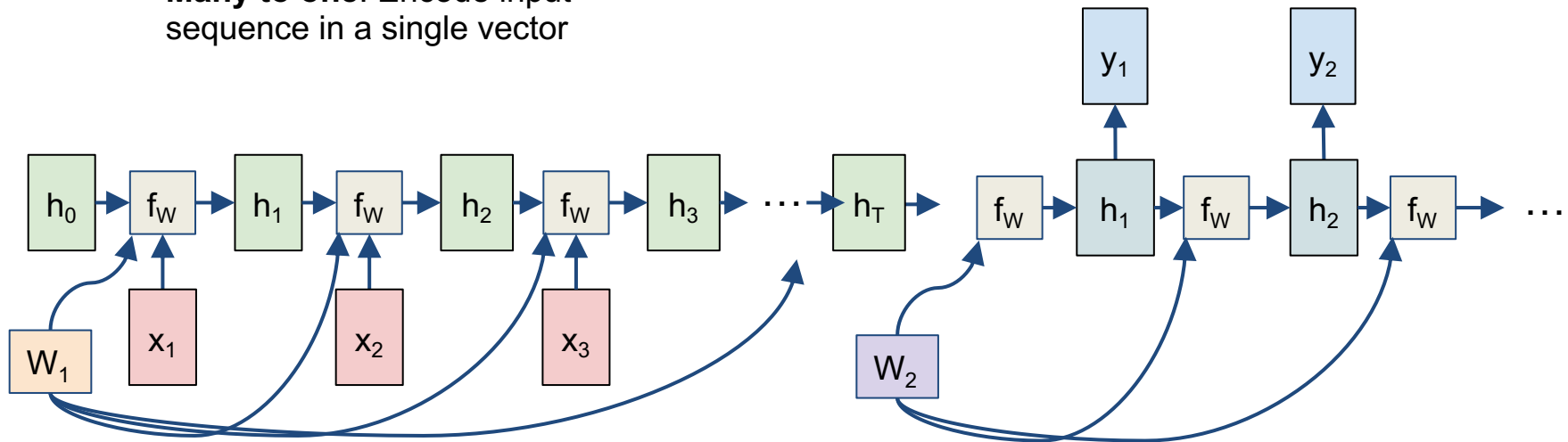
**Many to one:** Encode input sequence in a single vector



# Sequence to Sequence: Many-to-one + one-to-many

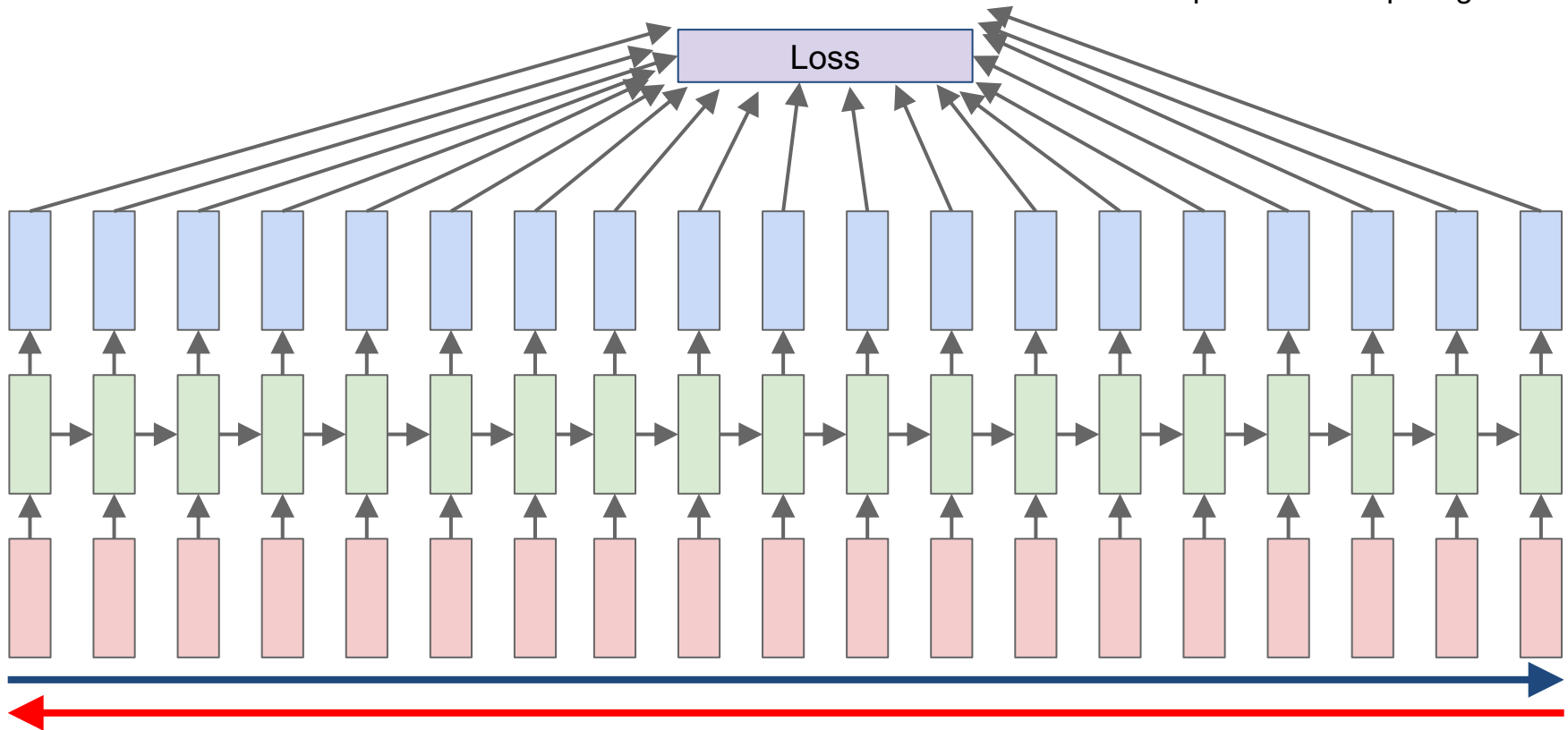
**Many to one:** Encode input sequence in a single vector

**One to many:** Produce output sequence from single input vector

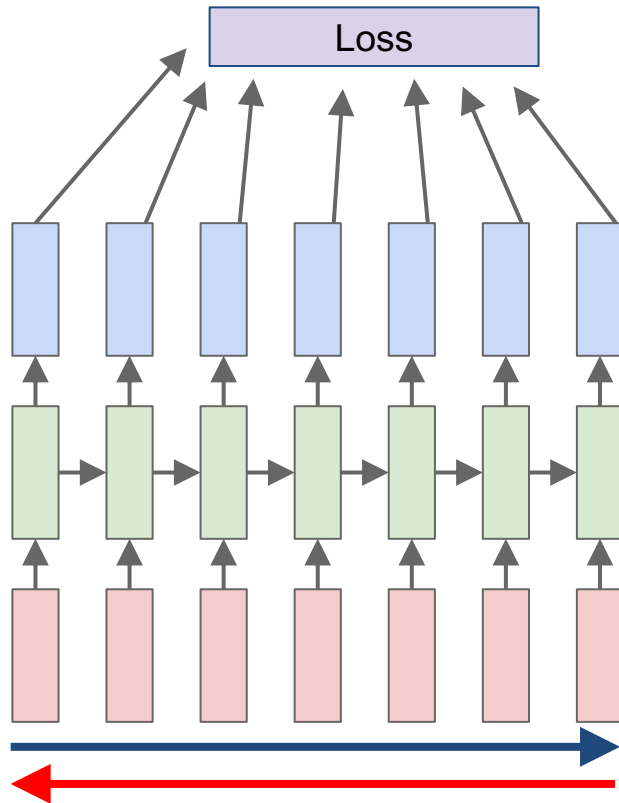


# Backpropagation through time

Forward through entire sequence to compute loss, then backward through entire sequence to compute gradient

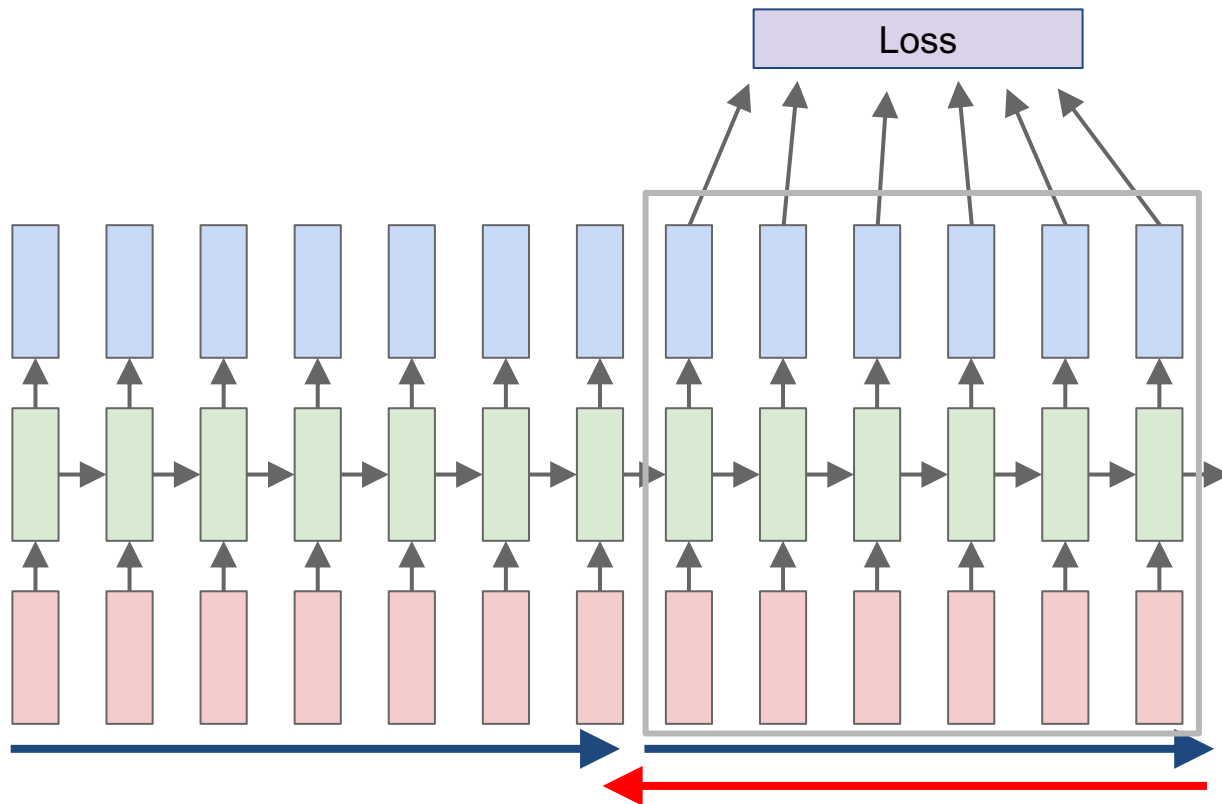


# Truncated Backpropagation through time



Run forward and backward through chunks of the sequence instead of whole sequence

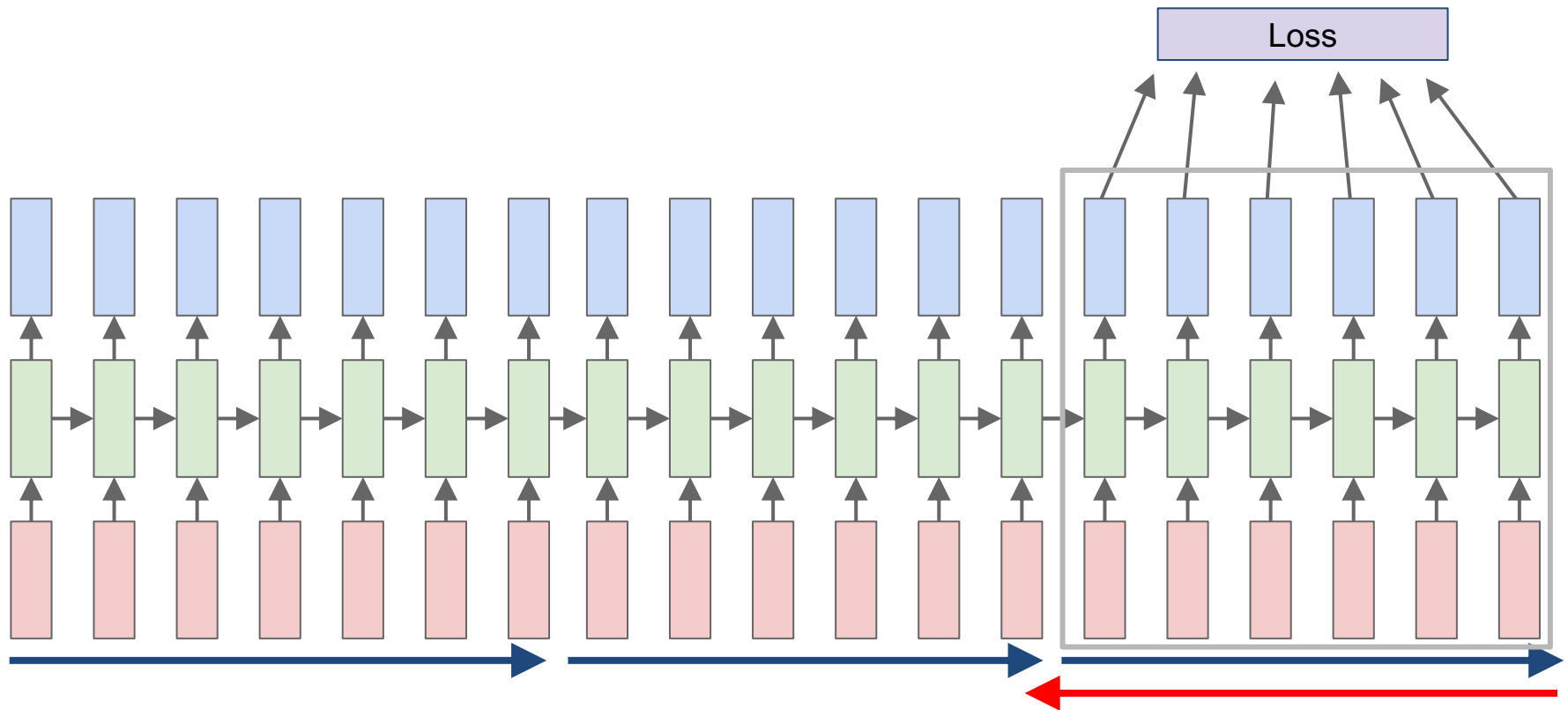
# Truncated Backpropagation through time



Carry hidden states forward in time forever, but only backpropagate for some smaller number of steps



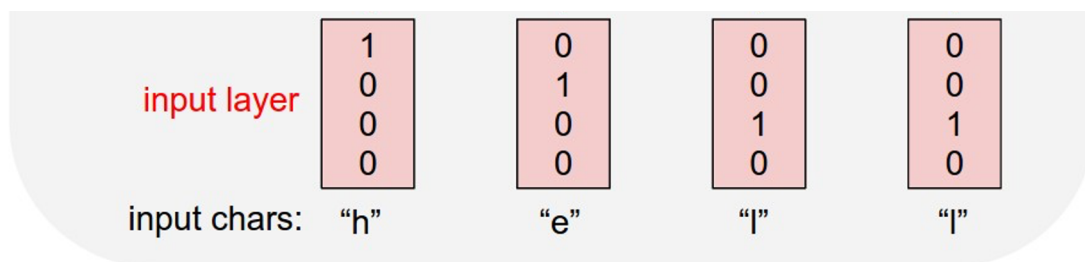
# Truncated Backpropagation through time



# Example: Character-level Language Model

Vocabulary:  
[h,e,l,o]

Example training  
sequence:  
“hello”

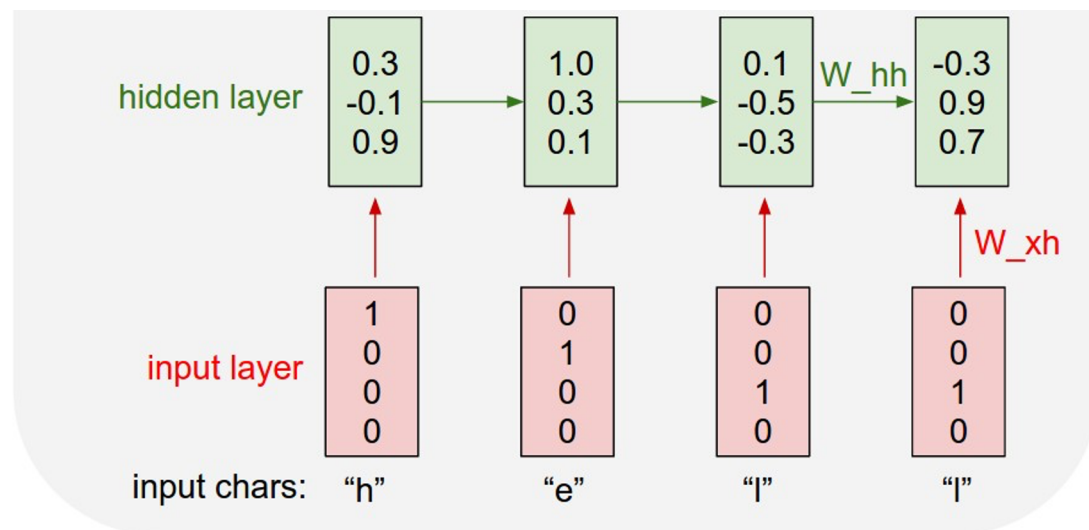


# Example: Character-level Language Model

Vocabulary:  
[h,e,l,o]

Example training  
sequence:  
“hello”

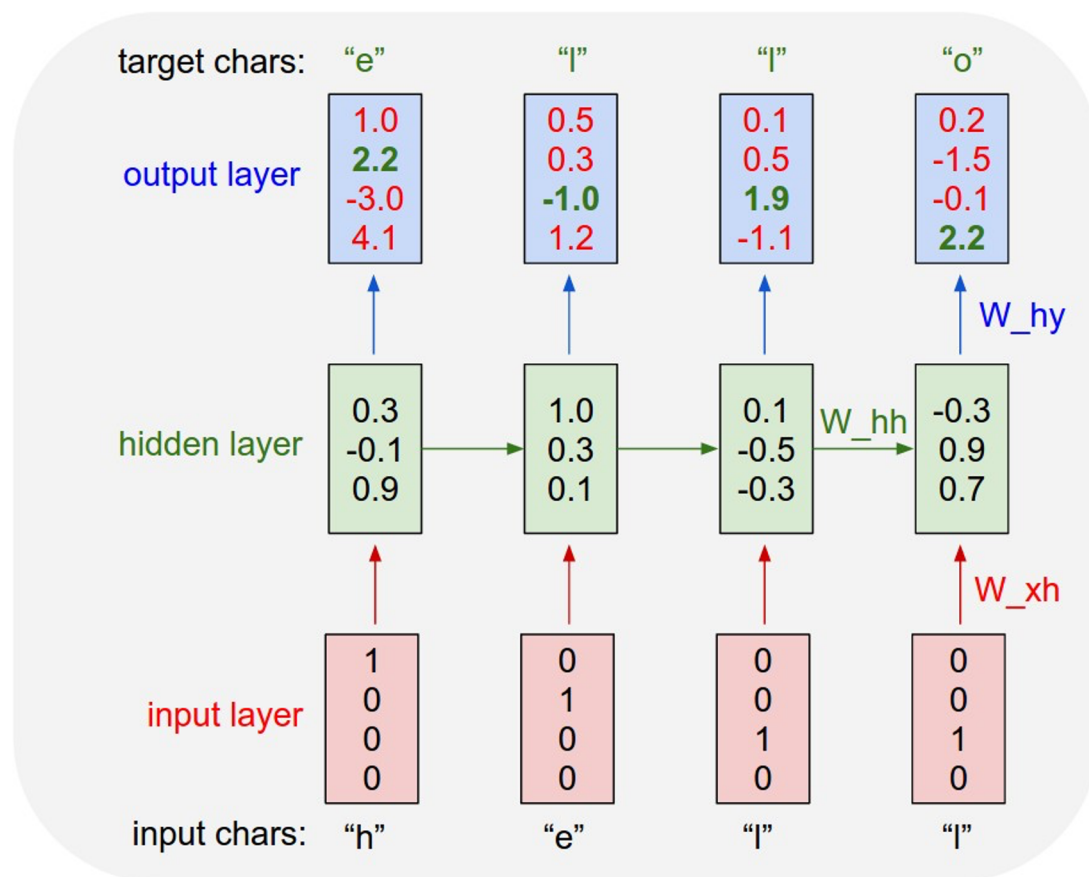
$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t + b_h)$$



# Example: Character-level Language Model

Vocabulary:  
[h,e,l,o]

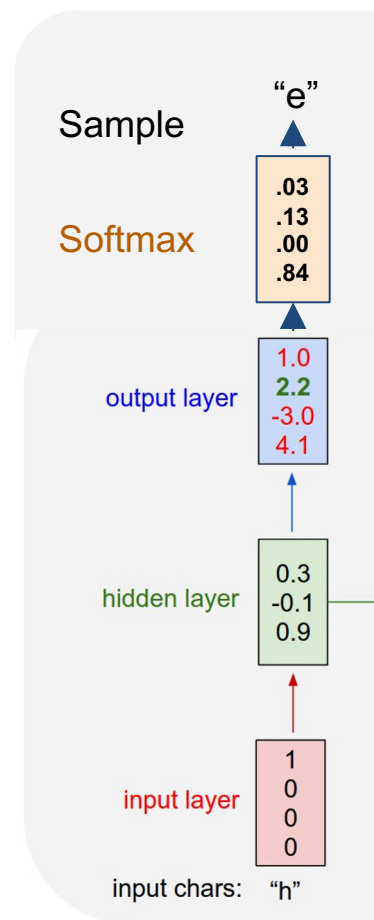
Example training  
sequence:  
“hello”



# Example: Character-level Language Model Sampling

Vocabulary:  
[h,e,l,o]

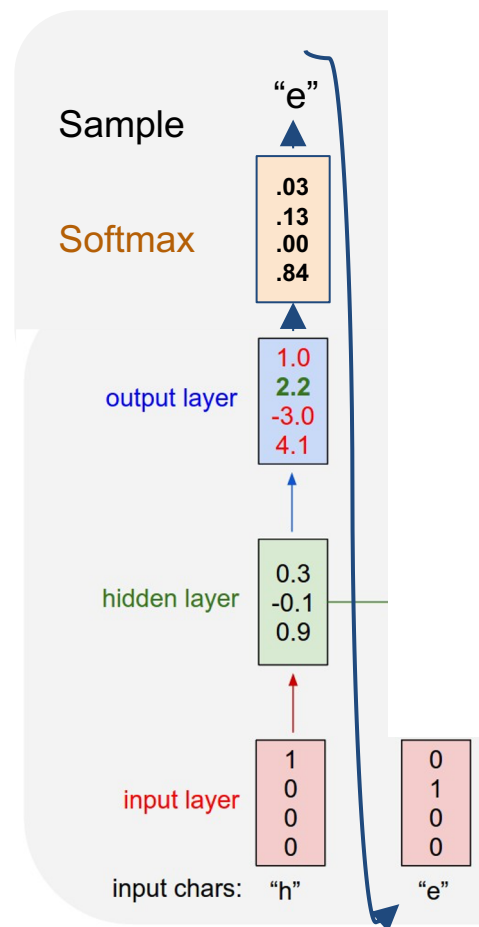
At test-time sample  
characters one at a  
time, feed back to  
model



# Example: Character-level Language Model Sampling

Vocabulary:  
[h,e,l,o]

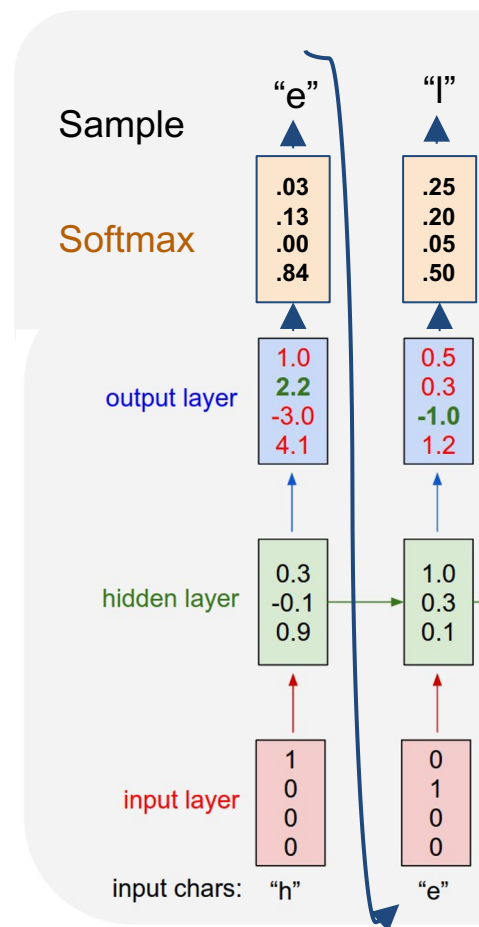
At test-time sample  
characters one at a  
time, feed back to  
model



# Example: Character-level Language Model Sampling

Vocabulary:  
[h,e,l,o]

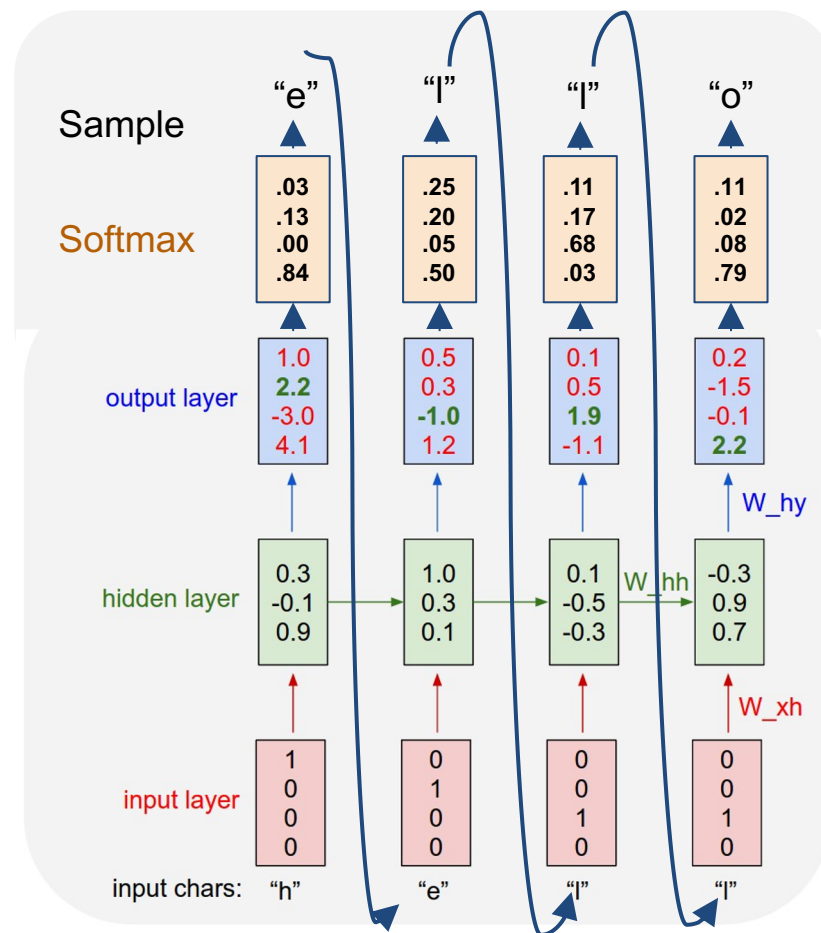
At test-time sample  
characters one at a  
time, feed back to  
model



# Example: Character-level Language Model Sampling

Vocabulary:  
[h,e,l,o]

At test-time sample  
characters one at a  
time, feed back to  
model





## [min-char-rnn.py gist: 112 lines of Python](https://gist.github.com/karpathy/d4dee566867f8291f086)

```
1  """
2  Minimal character-level Vanilla RNN model. Written by Andrej Karpathy (@karpathy)
3  BSD License
4  """
5  import numpy as np
6
7  # data I/O
8  data = open('input.txt', 'r').read() # should be simple plain text file
9  chars = list(set(data))
10 data_size, vocab_size = len(data), len(chars)
11 print 'data has %d characters, %d unique.' % (data_size, vocab_size)
12 char_to_ix = { ch:i for i,ch in enumerate(chars) }
13 ix_to_char = { i:ch for i,ch in enumerate(chars) }
14
15 # hyperparameters
16 hidden_size = 100 # size of hidden layer of neurons
17 seq_length = 25 # number of steps to unroll the RNN for
18 learning_rate = 1e-1
19
20 # model parameters
21 Wxh = np.random.randn(hidden_size, vocab_size)*0.01 # input to hidden
22 Whh = np.random.randn(hidden_size, hidden_size)*0.01 # hidden to hidden
23 Why = np.random.randn(vocab_size, hidden_size)*0.01 # hidden to output
24 bh = np.zeros((hidden_size, 1)) # hidden bias
25 by = np.zeros((vocab_size, 1)) # output bias
26
27 def lossFun(inputs, targets, hprev):
28     """
29     inputs, targets are both list of integers.
30     hprev is Nx1 array of initial hidden state
31     returns the loss, gradients on model parameters, and last hidden state
32     """
33     xs, hs, ys, ps = {}, {}, {}, {}
34     hs[-1] = np.copy(hprev)
35     loss = 0
36     # forward pass
37     for t in xrange(len(inputs)):
38         xs[t] = np.zeros((vocab_size,1)) # encode in 1-of-k representation
39         xs[t][inputs[t]] = 1
40         hs[t] = np.tanh(np.dot(Wxh, xs[t]) + np.dot(Whh, hs[-1]) + bh) # hidden state
41         ys[t] = np.dot(Why, hs[t]) + by # unnormalized log probabilities for next chars
42         ps[t] = np.exp(ys[t]) / np.sum(np.exp(ys[t])) # probabilities for next chars
43         loss += -np.log(ps[t][targets[t],0]) # softmax (cross-entropy) loss
44     # backward pass: compute gradients going backwards
45     dWxh, dWhh, dWhy = np.zeros_like(Wxh), np.zeros_like(Whh), np.zeros_like(Why)
46     dbh, dby = np.zeros_like(bh), np.zeros_like(by)
47     dhnext = np.zeros_like(hs[0])
48     for t in reversed(xrange(len(inputs))):
49         dy = np.copy(ps[t])
50         dy[targets[t]] -= 1 # backprop into y
51         dWhy += np.dot(dy, hs[t].T)
52         dby += dy
53         dh = np.dot(Why.T, dy) + dhnext # backprop into h
54         dhraw = (1 - hs[t]**2) * dh # backprop through tanh nonlinearity
55         dbh += dhraw
56         dWxh += np.dot(dhraw, xs[t].T)
57         dWhh += np.dot(dhraw, hs[t-1].T)
58         dhnext = np.dot(Whh.T, dhraw)
59     for dparam in [dWxh, dWhh, dWhy, dbh, dby]:
60         np.clip(dparam, -5, 5, out=dparam) # clip to mitigate exploding gradients
61     return loss, dWxh, dWhh, dWhy, dbh, dby, hs[len(inputs)-1]
62
63 def sample(h, seed_ix, n):
64     """
65     sample a sequence of integers from the model
66     h is memory state, seed_ix is seed letter for first time step
67     """
68     x = np.zeros((vocab_size, 1))
69     x[seed_ix] = 1
70     ixes = []
71     for t in xrange(n):
72         h = np.tanh(np.dot(Wxh, x) + np.dot(Whh, h) + bh)
73         y = np.dot(Why, h) + by
74         p = np.exp(y) / np.sum(np.exp(y))
75         ix = np.random.choice(range(vocab_size), p=p.ravel())
76         x = np.zeros((vocab_size, 1))
77         x[ix] = 1
78         ixes.append(ix)
79     return ixes
80
81 n, p = 0, 0
82 mWxh, mWhh, mWhy = np.zeros_like(Wxh), np.zeros_like(Whh), np.zeros_like(Why)
83 mbh, mby = np.zeros_like(bh), np.zeros_like(by) # memory variables for Adagrad
84 smooth_loss = -np.log(1.0/vocab_size)*seq_length # loss at iteration 0
85 while True:
86     # prepare inputs (we're sweeping from left to right in steps seq_length long)
87     if p+seq_length+1 >= len(data) or n == 0:
88         hprev = np.zeros((hidden_size,1)) # reset RNN memory
89         p = 0 # go from start of data
90     inputs = [char_to_ix[ch] for ch in data[p:p+seq_length]]
91     targets = [char_to_ix[ch] for ch in data[p+1:p+seq_length+1]]
92
93     # sample from the model now and then
94     if n % 100 == 0:
95         sample_ix = sample(hprev, inputs[0], 200)
96         txt = ''.join(ix_to_char[ix] for ix in sample_ix)
97         print '....\n%s\n....' % (txt, )
98
99     # forward seq_length characters through the net and fetch gradient
100    loss, dWxh, dWhh, dWhy, dbh, dby, hprev = lossFun(inputs, targets, hprev)
101    smooth_loss = smooth_loss * 0.999 + loss * 0.001
102    if n % 100 == 0: print 'iter %d, loss: %f' % (n, smooth_loss) # print progress
103
104    # perform parameter update with Adagrad
105    for param, dparam, mem in zip([Wxh, Whh, Why, bh, by],
106                                  [dWxh, dWhh, dWhy, dbh, dby],
107                                  [mWxh, mWhh, mWhy, mbh, mby]):
108        mem += dparam * dparam
109        param += -learning_rate * dparam / np.sqrt(mem + 1e-8) # adagrad update
110
111    p += seq_length # move data pointer
112    n += 1 # iteration counter
```

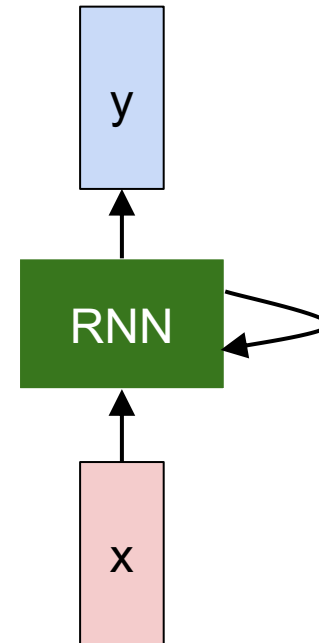
(<https://gist.github.com/karpathy/d4dee566867f8291f086>)

# THE SONNETS

by William Shakespeare

From fairest creatures we desire increase,  
That thereby beauty's rose might never die,  
But as the ripper should by time decease,  
His tender heir might bear his memory:  
But thou, contracted to thine own bright eyes,  
Feed'st thy light's flame with self-substantial fuel,  
Making a famine where abundance lies,  
Thyself thy foe, to thy sweet self too cruel:  
Thou that art now the world's fresh ornament,  
And only herald to the gaudy spring,  
Within thine own bud buriest thy content,  
And tender churl mak'st waste in niggarding:  
    Pity the world, or else this glutton be,  
    To eat the world's due, by the grave and thee.

When forty winters shall besiege thy brow,  
And dig deep trenches in thy beauty's field,  
Thy youth's proud livery so gazed on now,  
Will be a tatter'd weed of small worth held:  
Then being asked, where all thy beauty lies,  
Where all the treasure of thy lusty days;  
To say, within thine own deep sunken eyes,  
Were an all-eating shame, and thriftless praise.  
How much more praise deserv'd thy beauty's use,  
If thou couldst answer 'This fair child of mine  
Shall sum my count, and make my old excuse,'  
Proving his beauty by succession thine!  
    This were to be new made when thou art old,  
    And see thy blood warm when thou feel'st it cold.



at first:

tyntd-iafhatawiaoihrdemot lytdws e ,tfti, astai f ogoh eoase rrranbyne 'nhthnee e  
plia tklrqd t o idoe ns,smtt h ne etie h,hregtrs nigtiqe,aoaenns lng



train more

"Tmont thithey" fomesscerliund  
Keushey. Thom here  
sheulke, anmerenith ol sivh I lalterthend Bleipile shuw y fil on aseterlome  
coaniogennc Phe lism thond hon at. MeiDimorotion in ther thize."



train more

Aftair fall unsuch that the hall for Prince Velzonski's that me of  
her hearly, and behs to so arwage fiving were to it beloge, pavu say falling misfort  
how, and Gogition is so overelical and ofter.



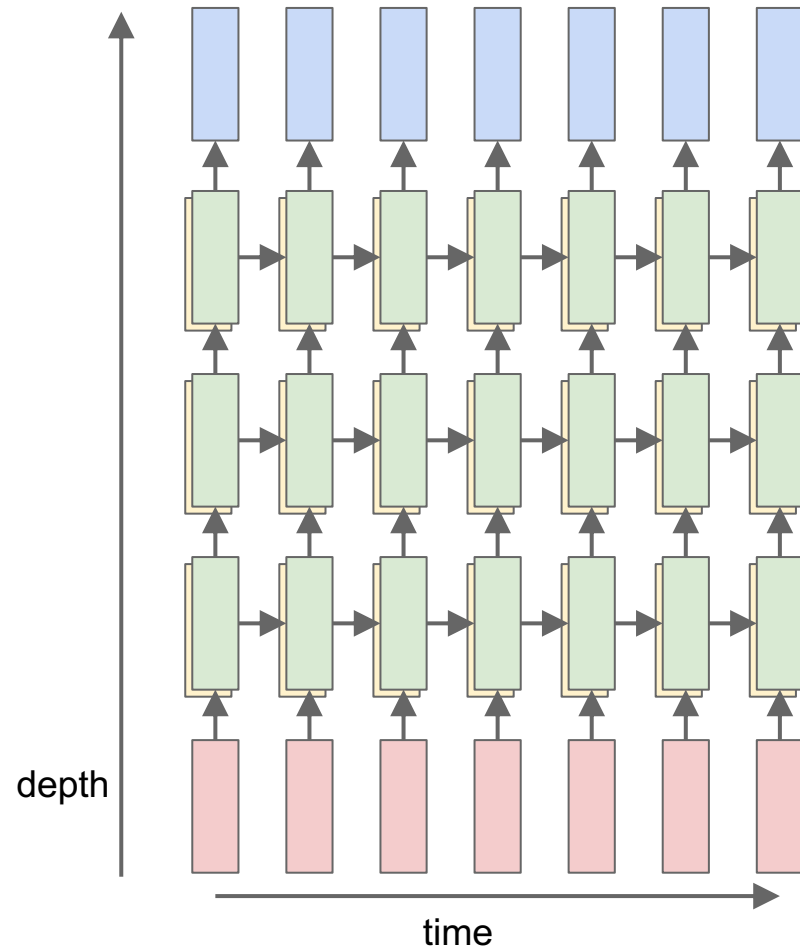
train more

"Why do what that day," replied Natasha, and wishing to himself the fact the  
princess, Princess Mary was easier, fed in had oftended him.  
Pierre aking his soul came to the packs and drove up his father-in-law women.

# Multilayer RNNs

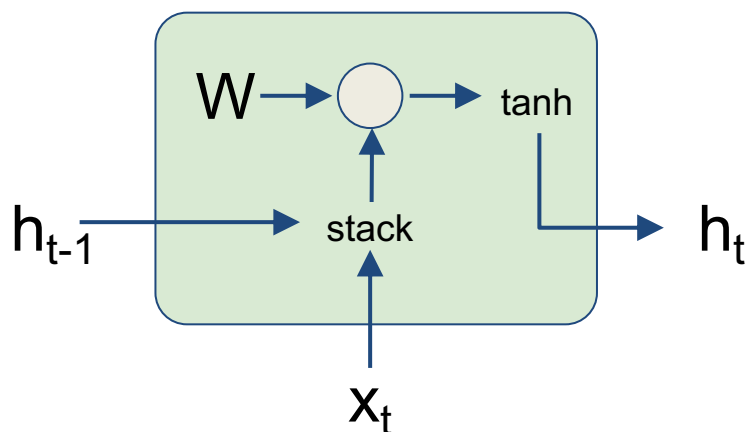
$$h_t^l = \tanh W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$

$h \in \mathbb{R}^n$        $W^l [n \times 2n]$



# Vanilla RNN Gradient Flow

Bengio et al, "Learning long-term dependencies with gradient descent is difficult", IEEE Transactions on Neural Networks, 1994  
Pascanu et al, "On the difficulty of training recurrent neural networks", ICML 2013



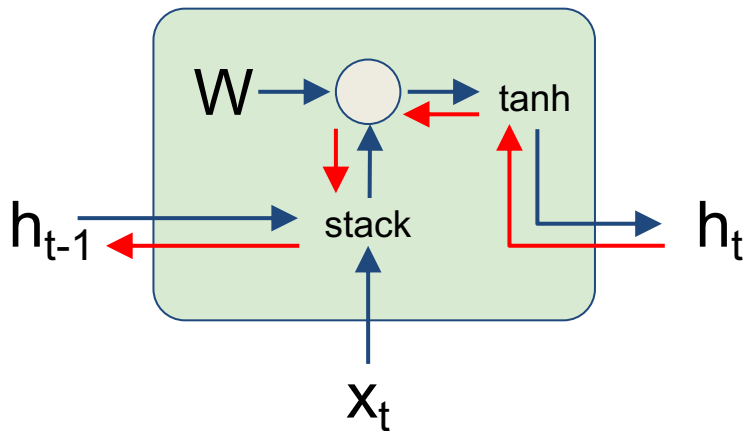
$$\begin{aligned} h_t &= \tanh(W_{hh}h_{t-1} + W_{hx}x_t) \\ &= \tanh\left((W_{hh} \quad W_{hx}) \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right) \\ &= \tanh\left(W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right) \end{aligned}$$

# Vanilla RNN Gradient Flow

Bengio et al, "Learning long-term dependencies with gradient descent is difficult", IEEE Transactions on Neural Networks, 1994

Pascanu et al, "On the difficulty of training recurrent neural networks", ICML 2013

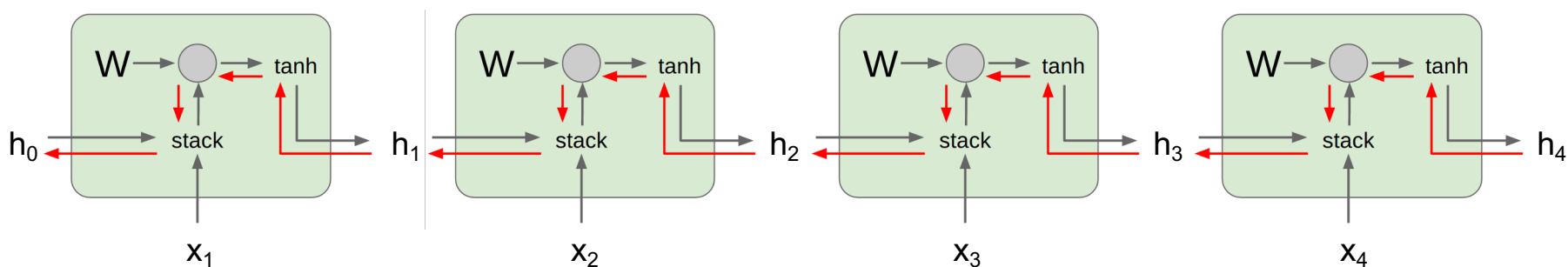
Backpropagation from  $h_t$  to  $h_{t-1}$  multiplies by  $W$  (actually  $W_{hh}^T$ )



$$\begin{aligned} h_t &= \tanh(W_{hh}h_{t-1} + W_{xh}x_t) \\ &= \tanh\left((W_{hh} \quad W_{hx}) \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right) \\ &= \tanh\left(W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right) \end{aligned}$$

# Vanilla RNN Gradient Flow

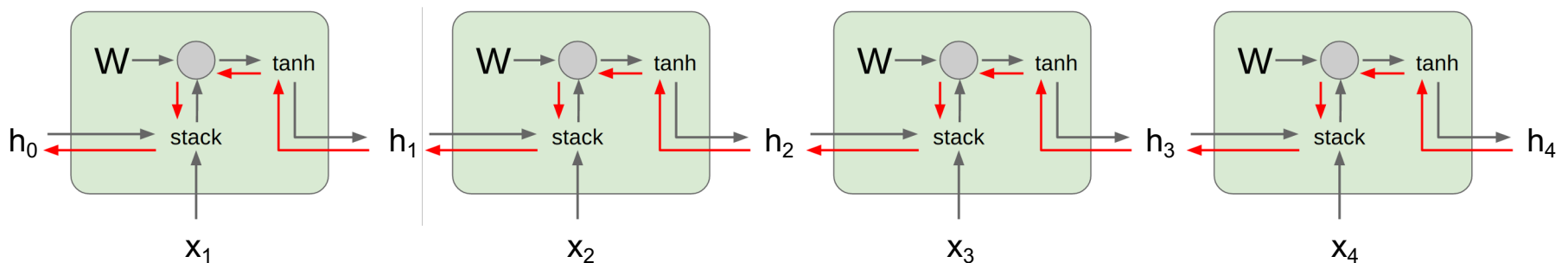
Bengio et al, "Learning long-term dependencies with gradient descent is difficult", IEEE Transactions on Neural Networks, 1994  
Pascanu et al, "On the difficulty of training recurrent neural networks", ICML 2013



Computing gradient  
of  $h_0$  involves many  
factors of  $W$   
(and repeated  $\tanh$ )

# Vanilla RNN Gradient Flow

Bengio et al, "Learning long-term dependencies with gradient descent is difficult", IEEE Transactions on Neural Networks, 1994  
Pascanu et al, "On the difficulty of training recurrent neural networks", ICML 2013



Computing gradient of  $h_0$  involves many factors of  $W$  (and repeated tanh)

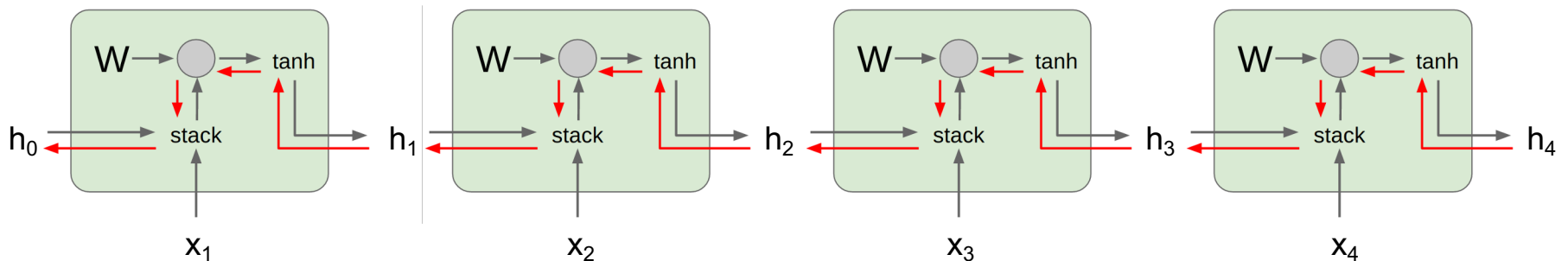
Largest singular value  $> 1$ :  
**Exploding gradients**

Largest singular value  $< 1$ :  
**Vanishing gradients**



# Vanilla RNN Gradient Flow

Bengio et al, "Learning long-term dependencies with gradient descent is difficult", IEEE Transactions on Neural Networks, 1994  
Pascanu et al, "On the difficulty of training recurrent neural networks", ICML 2013



Computing gradient of  $h_0$  involves many factors of  $W$  (and repeated tanh)

Largest singular value  $> 1$ :  
**Exploding gradients**

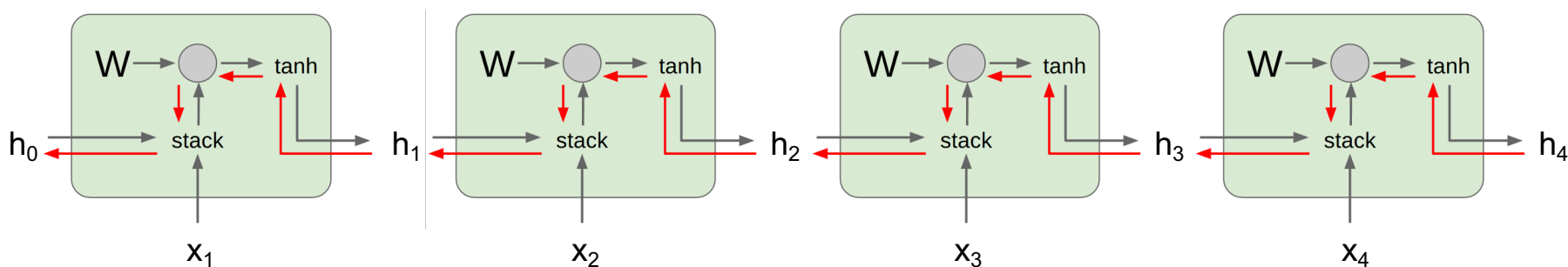
Largest singular value  $< 1$ :  
**Vanishing gradients**

➔ **Gradient clipping:** Scale gradient if its norm is too big

```
grad_norm = np.sum(grad * grad)
if grad_norm > threshold:
    grad *= (threshold / grad_norm)
```

# Vanilla RNN Gradient Flow

Bengio et al, "Learning long-term dependencies with gradient descent is difficult", IEEE Transactions on Neural Networks, 1994  
Pascanu et al, "On the difficulty of training recurrent neural networks", ICML 2013



Computing gradient of  $h_0$  involves many factors of  $W$  (and repeated  $\tanh$ )

Largest singular value  $> 1$ :  
**Exploding gradients**

Largest singular value  $< 1$ :  
**Vanishing gradients**

→ Change RNN architecture

# Long Short Term Memory (LSTM)

## Vanilla RNN

$$h_t = \tanh \left( W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix} \right)$$

## LSTM

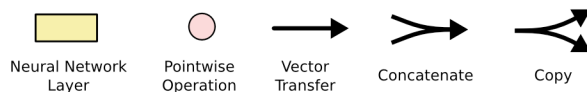
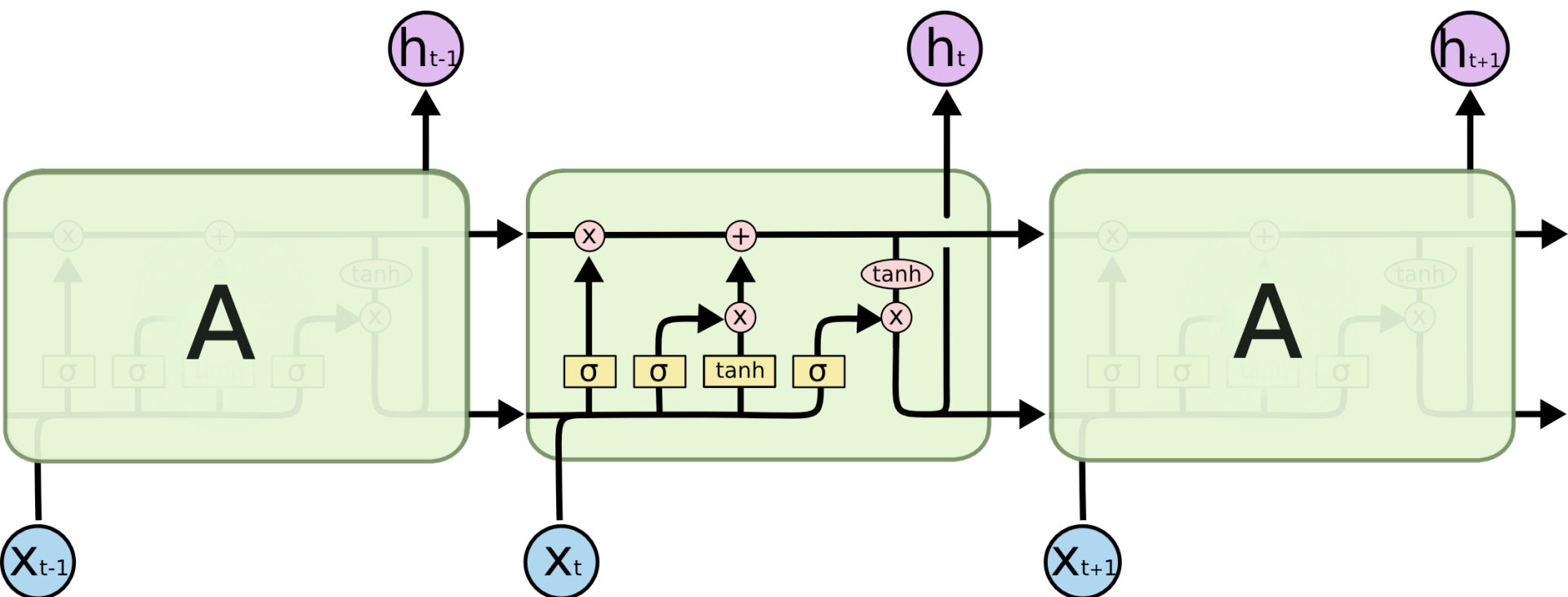
$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

$$c_t = f \odot c_{t-1} + i \odot g$$

$$h_t = o \odot \tanh(c_t)$$

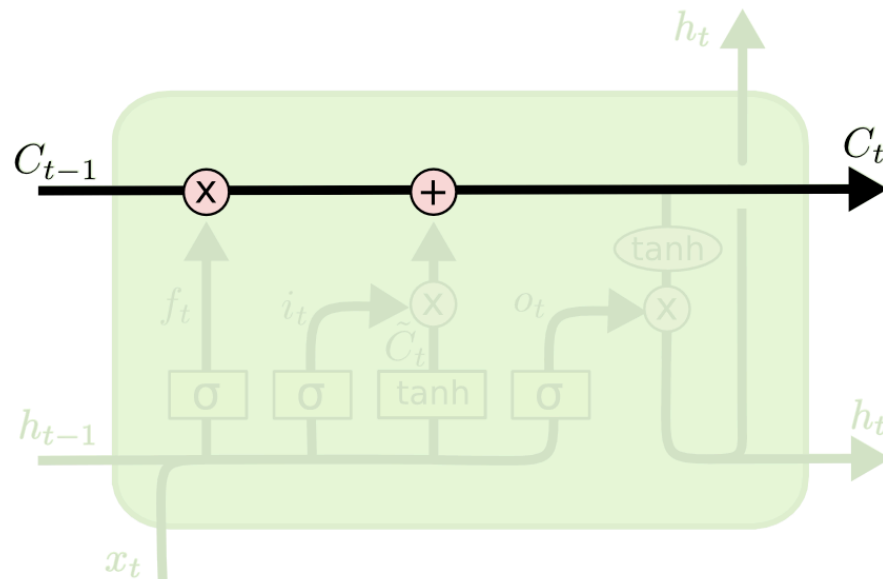
Hochreiter and Schmidhuber, "Long Short Term Memory", Neural Computation  
1997

# Meet LSTMs



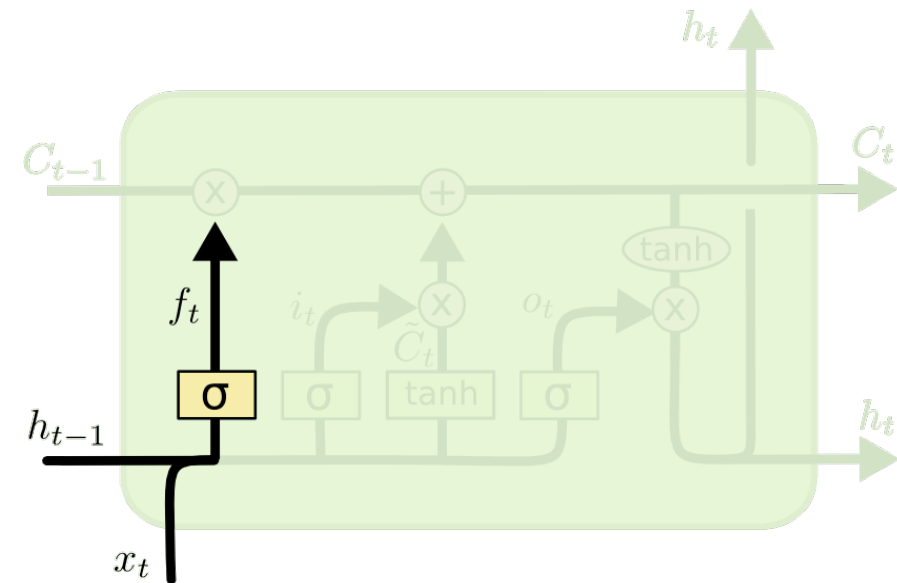
# LSTMs Intuition: Memory

- Cell State / Memory



# LSTMs Intuition: Forget Gate

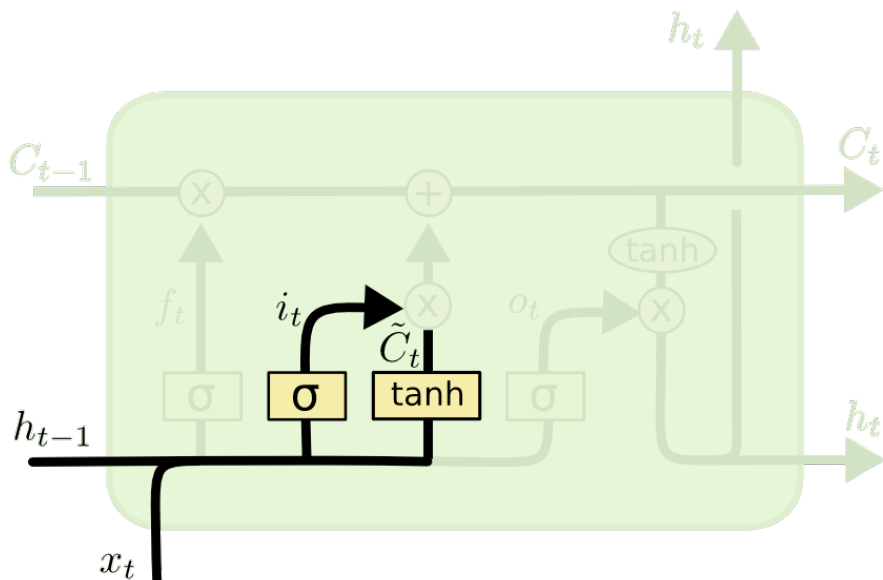
- Should we continue to remember this “bit” of information or not?



$$f_t = \sigma (W_f \cdot [h_{t-1}, x_t] + b_f)$$

# LSTMs Intuition: Input Gate

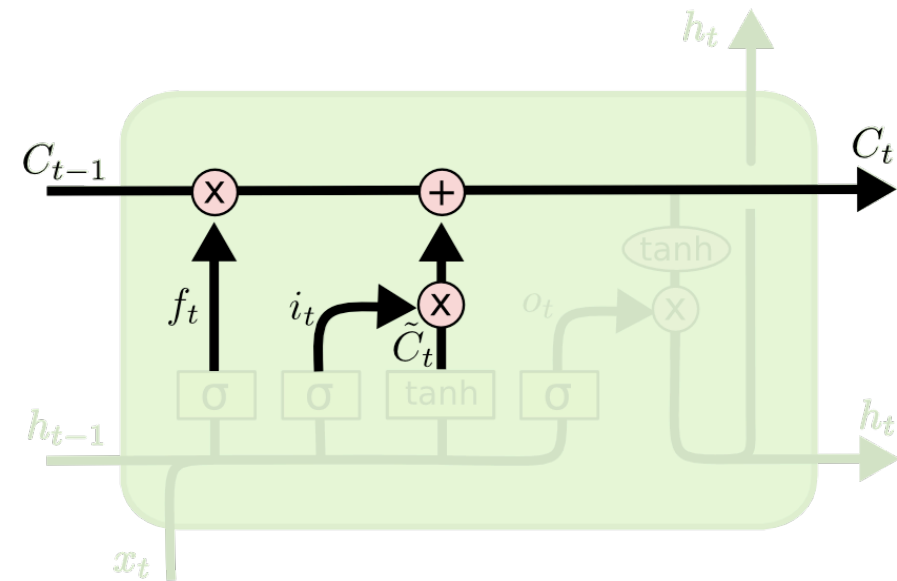
- Should we update this “bit” of information or not?
  - If so, with what?



$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

# LSTMs Intuition: Memory Update

- Forget that + memorize this

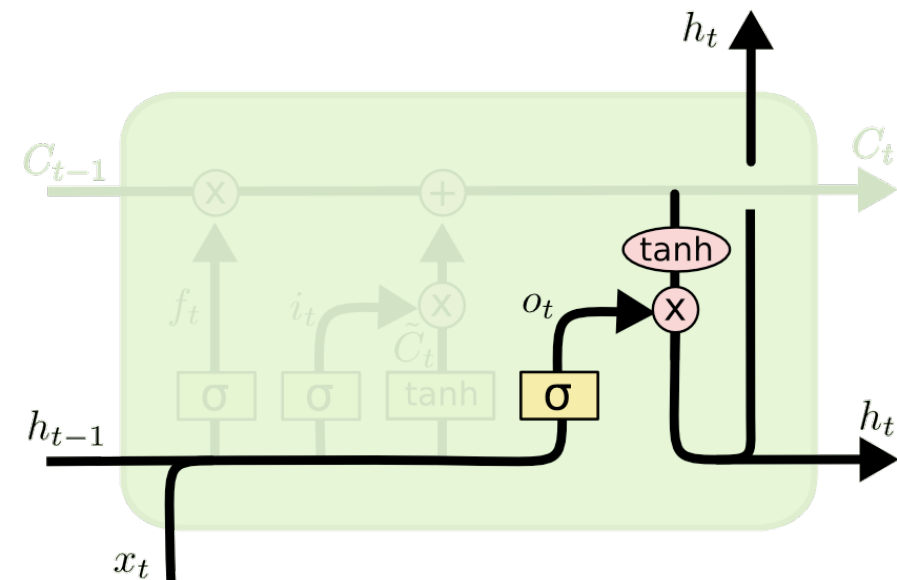


$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$



# LSTMs Intuition: Output Gate

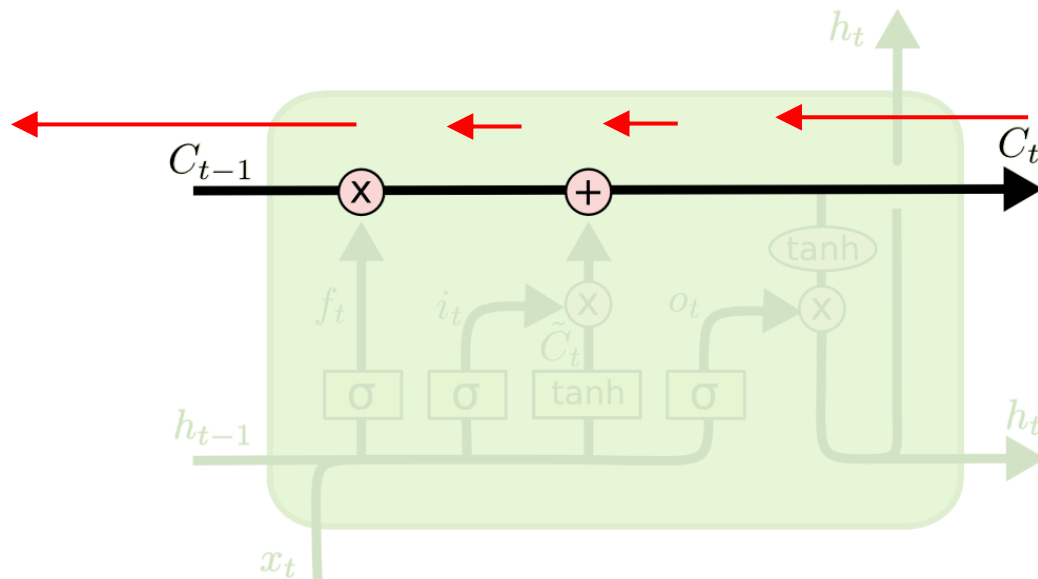
- Should we output this “bit” of information to “deeper” layers?



$$o_t = \sigma(W_o [h_{t-1}, x_t] + b_o)$$

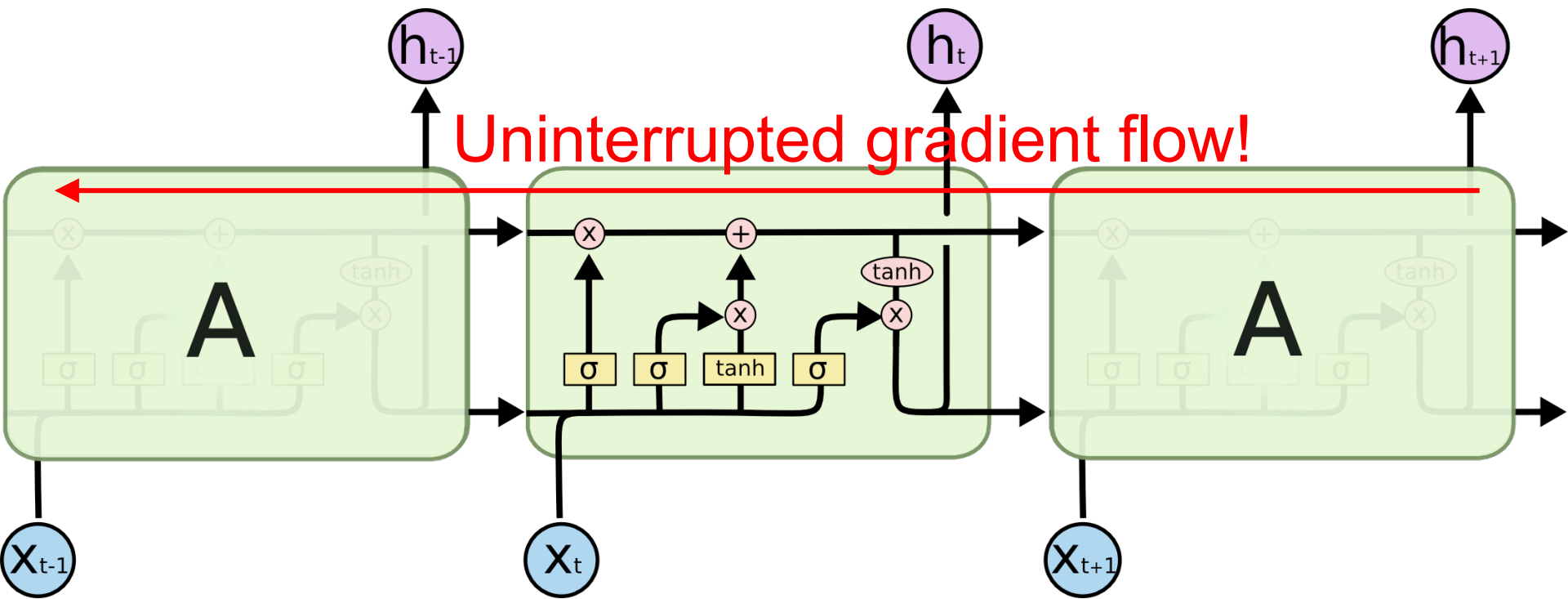
$$h_t = o_t * \tanh(C_t)$$

# LSTMs Intuition: Additive Updates

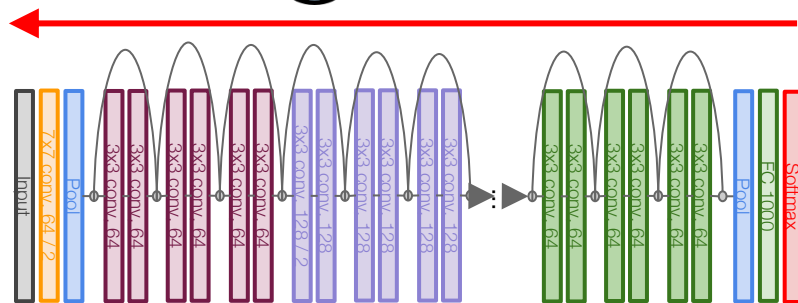
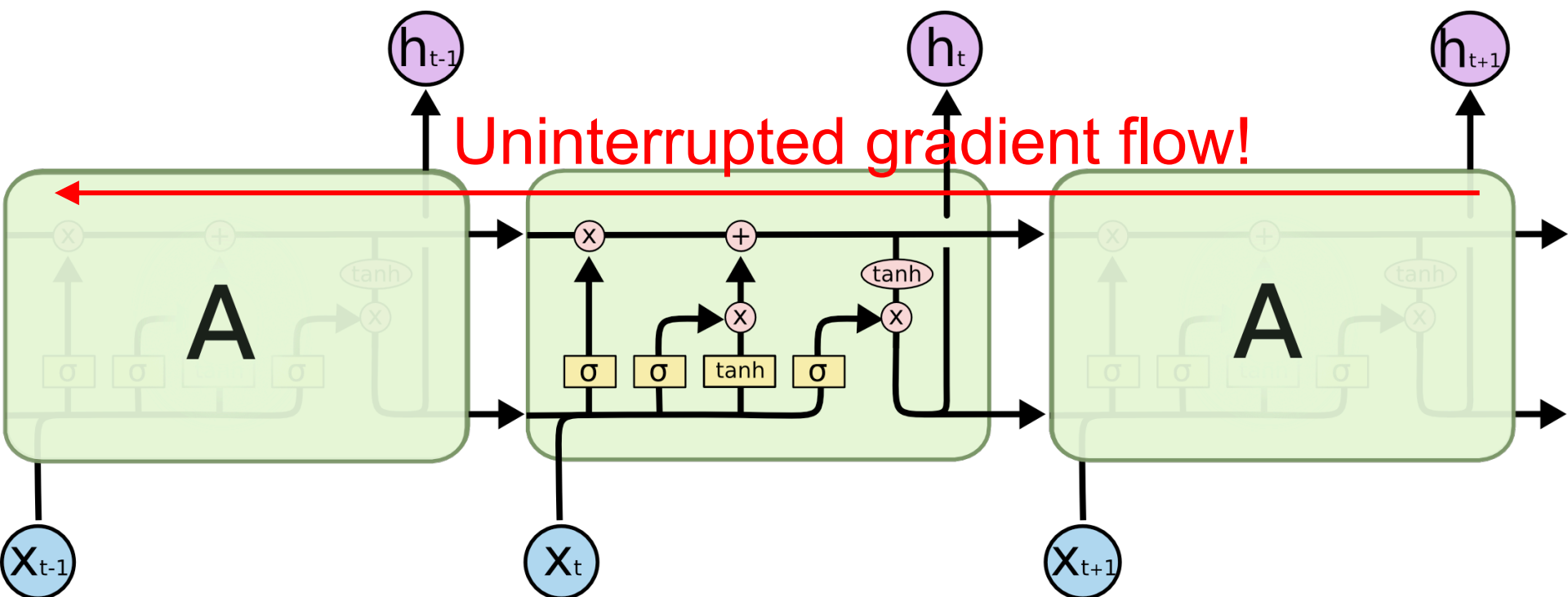


Backpropagation from  $c_t$  to  $c_{t-1}$  only  
elementwise  
multiplication by  $f$ , no  
matrix multiply by  $W$

# LSTMs Intuition: Additive Updates



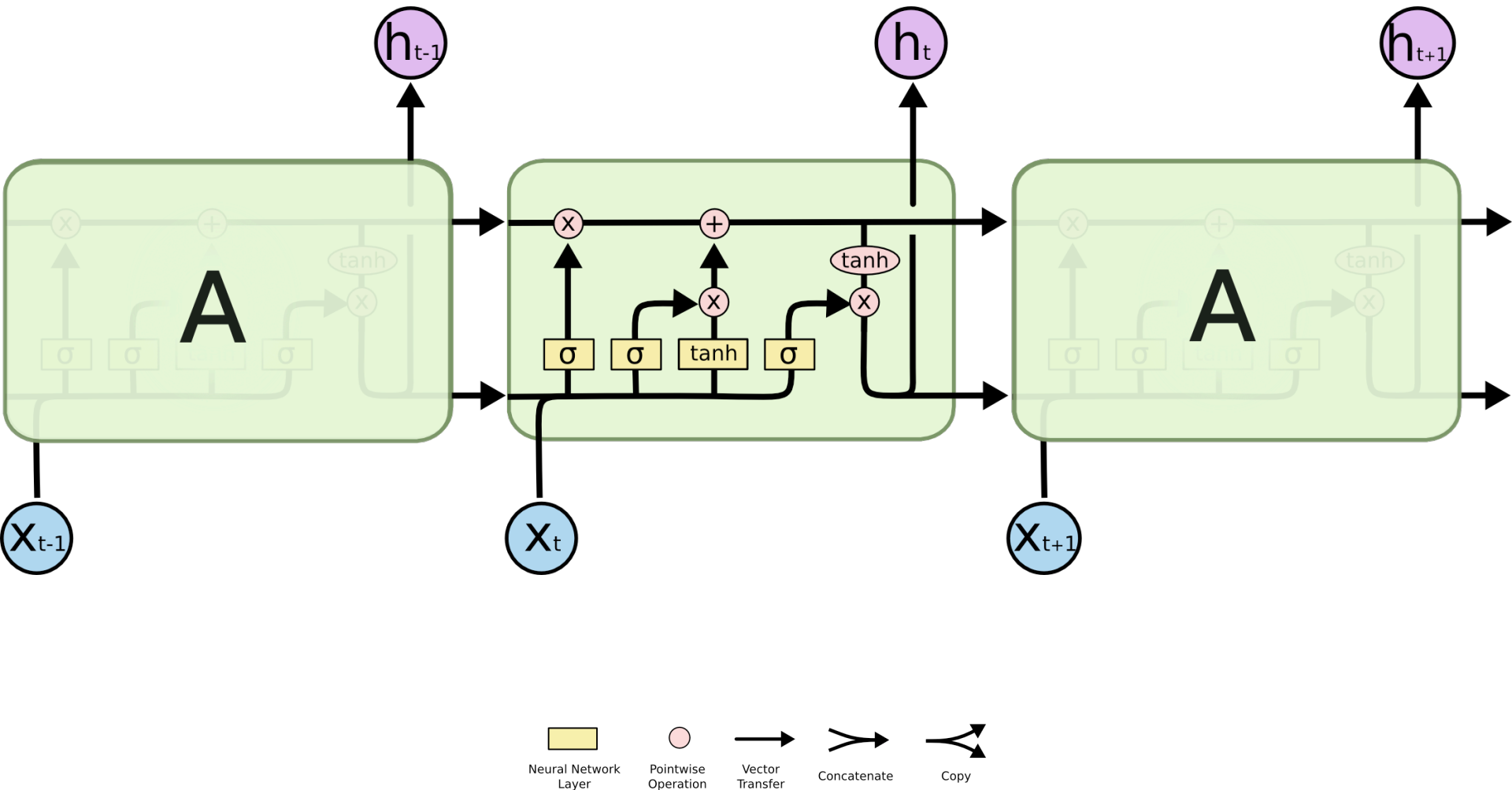
# LSTMs Intuition: Additive Updates



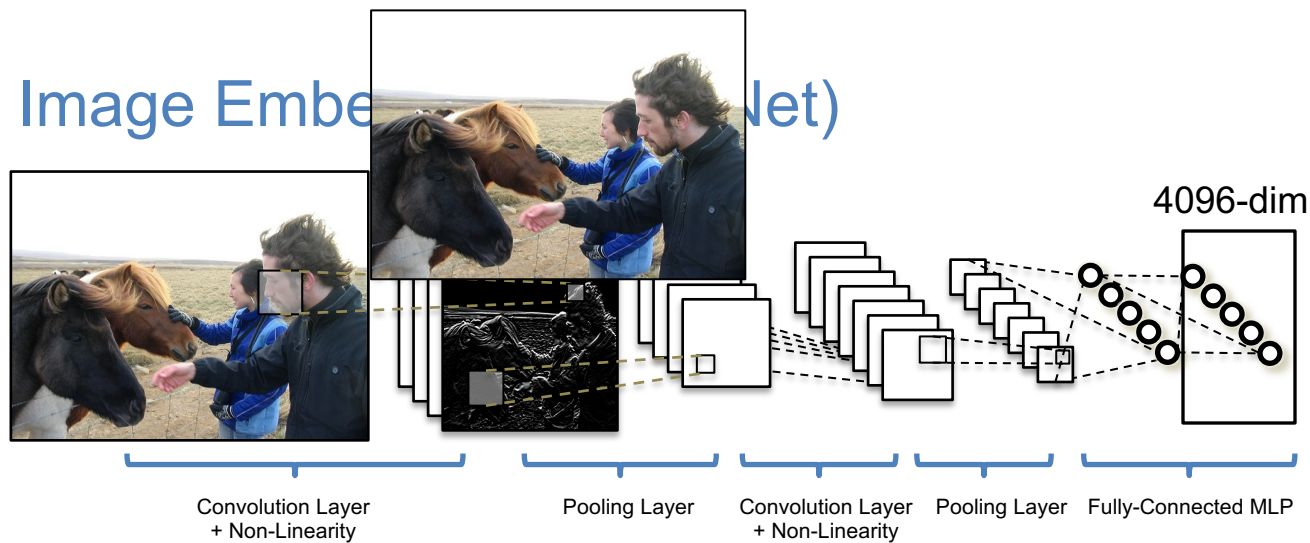
Similar to ResNet!

# LSTMs

- A pretty sophisticated cell

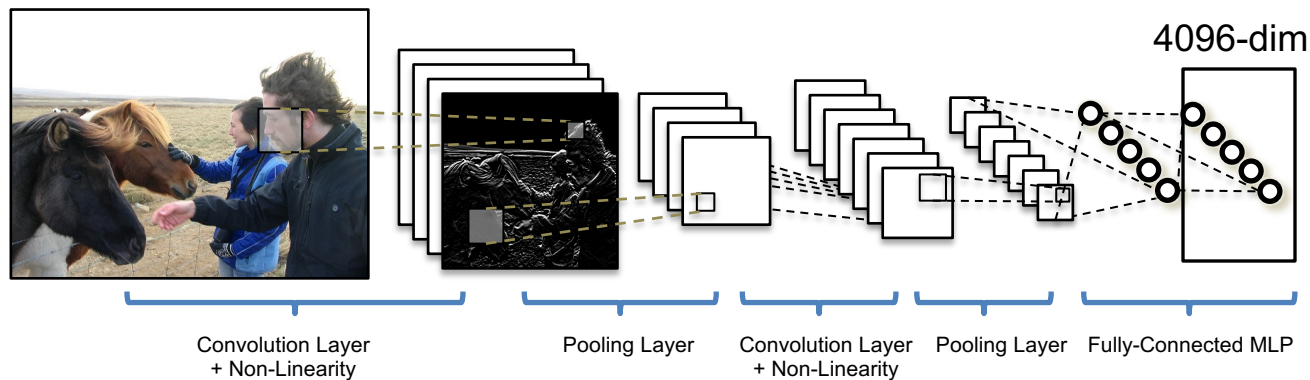


# Neural Image Captioning

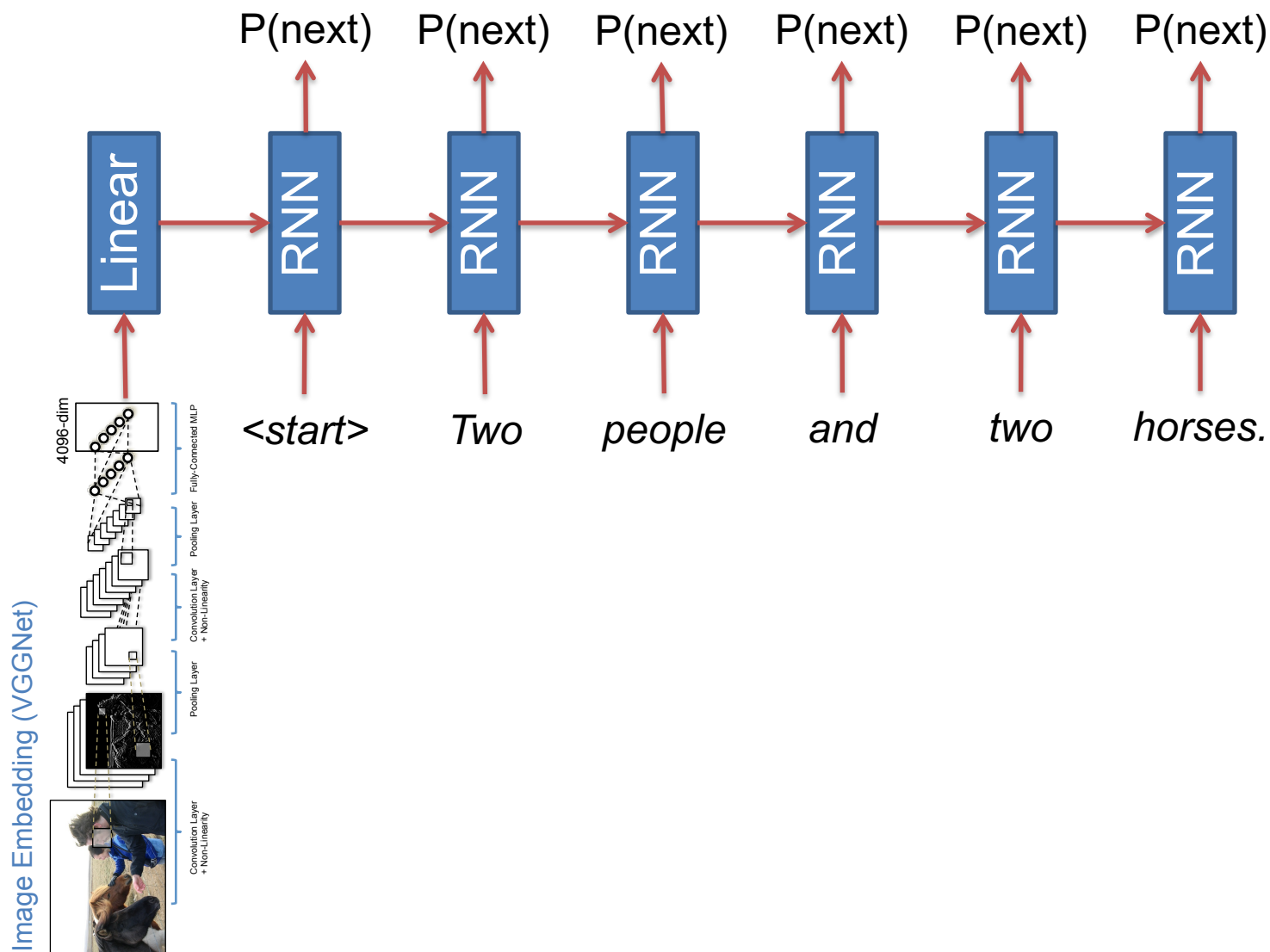


# Neural Image Captioning

## Image Embedding (VGGNet)

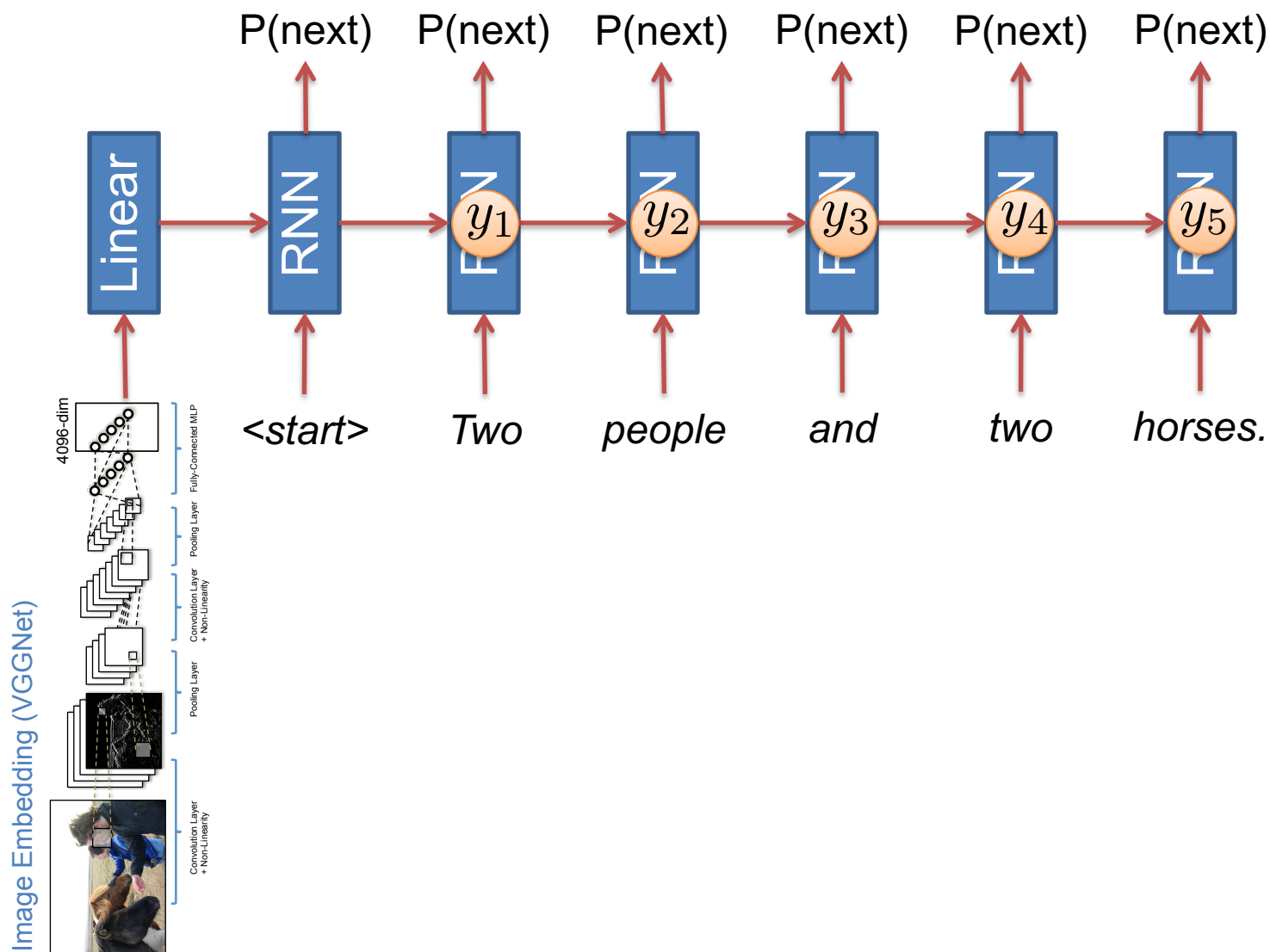


# Neural Image Captioning

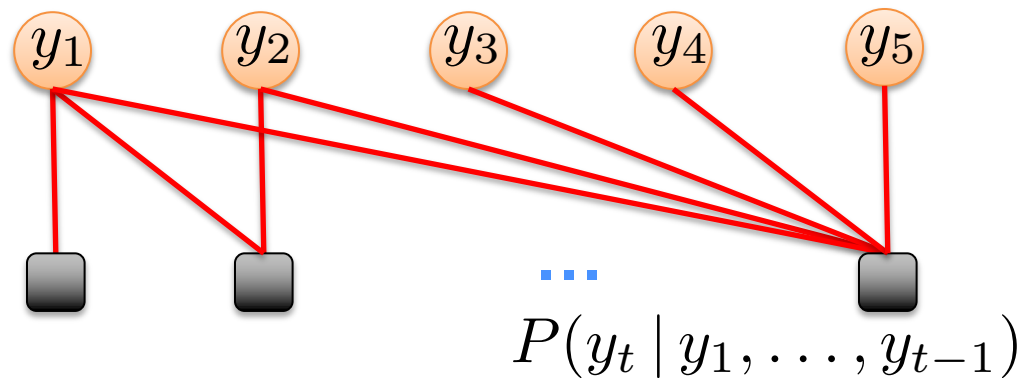




# Neural Image Captioning



# Sequence Model Factor Graph



# Beam Search Demo

- <http://dbs.cloudcv.org/captioning>

# Image Captioning

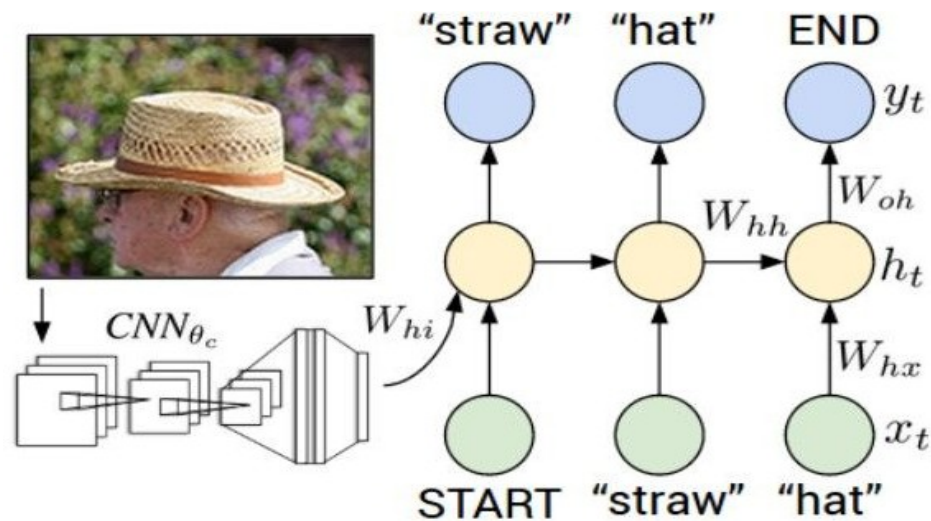
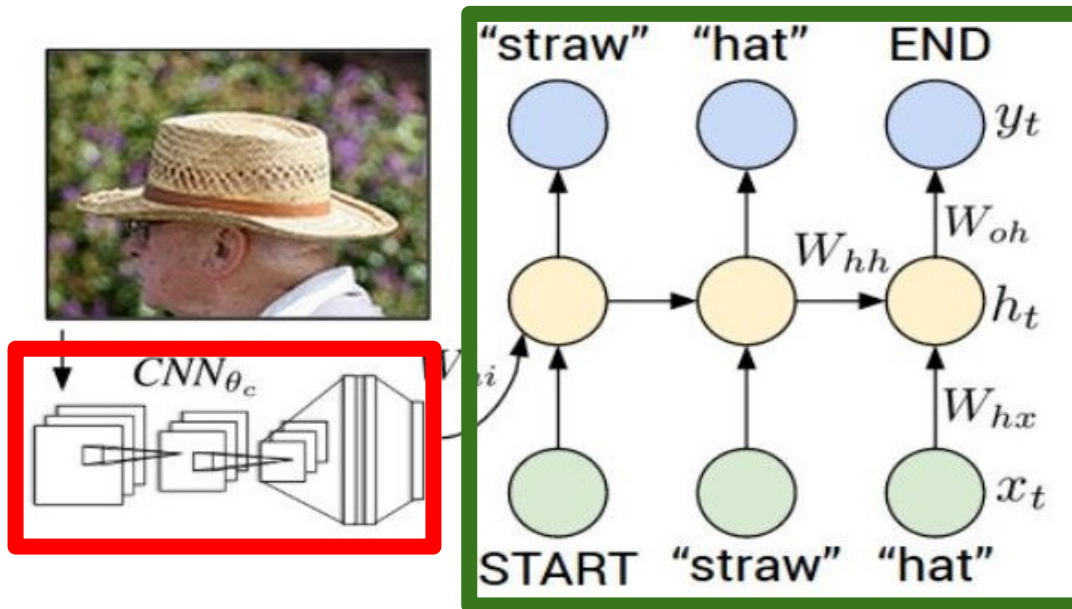


Figure from Karpathy et al, "Deep Visual-Semantic Alignments for Generating Image Descriptions", CVPR 2015; figure copyright IEEE, 2015.  
Reproduced for educational purposes.

- Many recent works on this:
- Baidu/UCLA: Explain Images with Multimodal Recurrent Neural Networks
- Toronto: Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models
- Berkeley: Long-term Recurrent Convolutional Networks for Visual Recognition and Description
- Google: Show and Tell: A Neural Image Caption Generator
- Stanford: Deep Visual-Semantic Alignments for Generating Image Description
- UML/UT: Translating Videos to Natural Language Using Deep Recurrent Neural Networks
- Microsoft/CMU: Learning a Recurrent Visual Representation for Image Caption Generation
- Microsoft: From Captions to Visual Concepts and Back

# Recurrent Neural Network



## Convolutional Neural Network



test image

[This image is CC0 public domain](#)

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

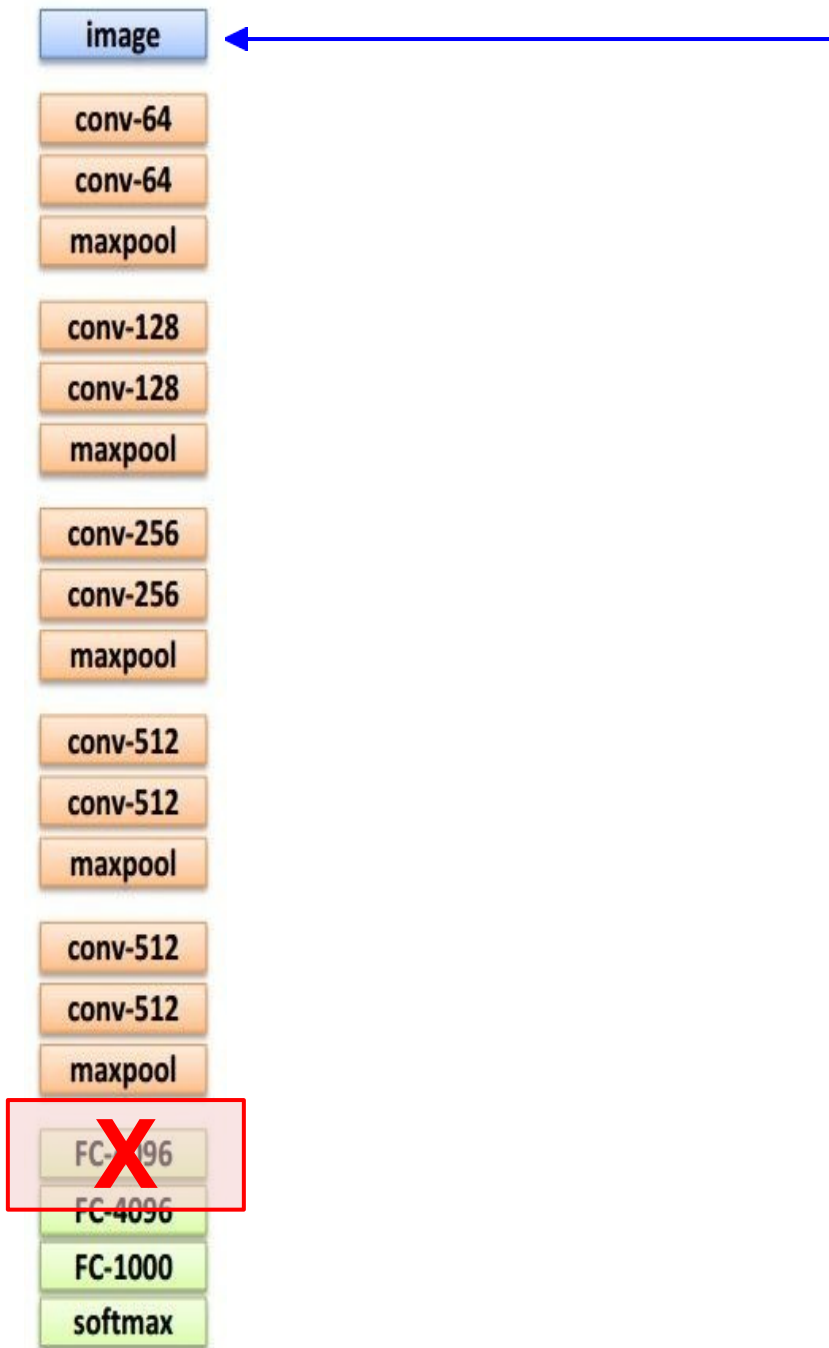
FC-4096

FC-1000

softmax



test image



test image



image



conv-64

conv-64

maxpool

conv-128

conv-128

maxpool

conv-256

conv-256

maxpool

conv-512

conv-512

maxpool

conv-512

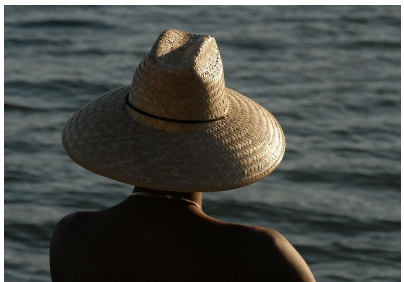
conv-512

maxpool

FC-4096

FC-1000

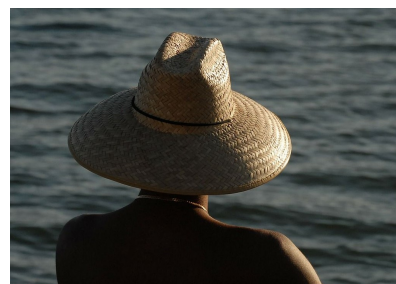
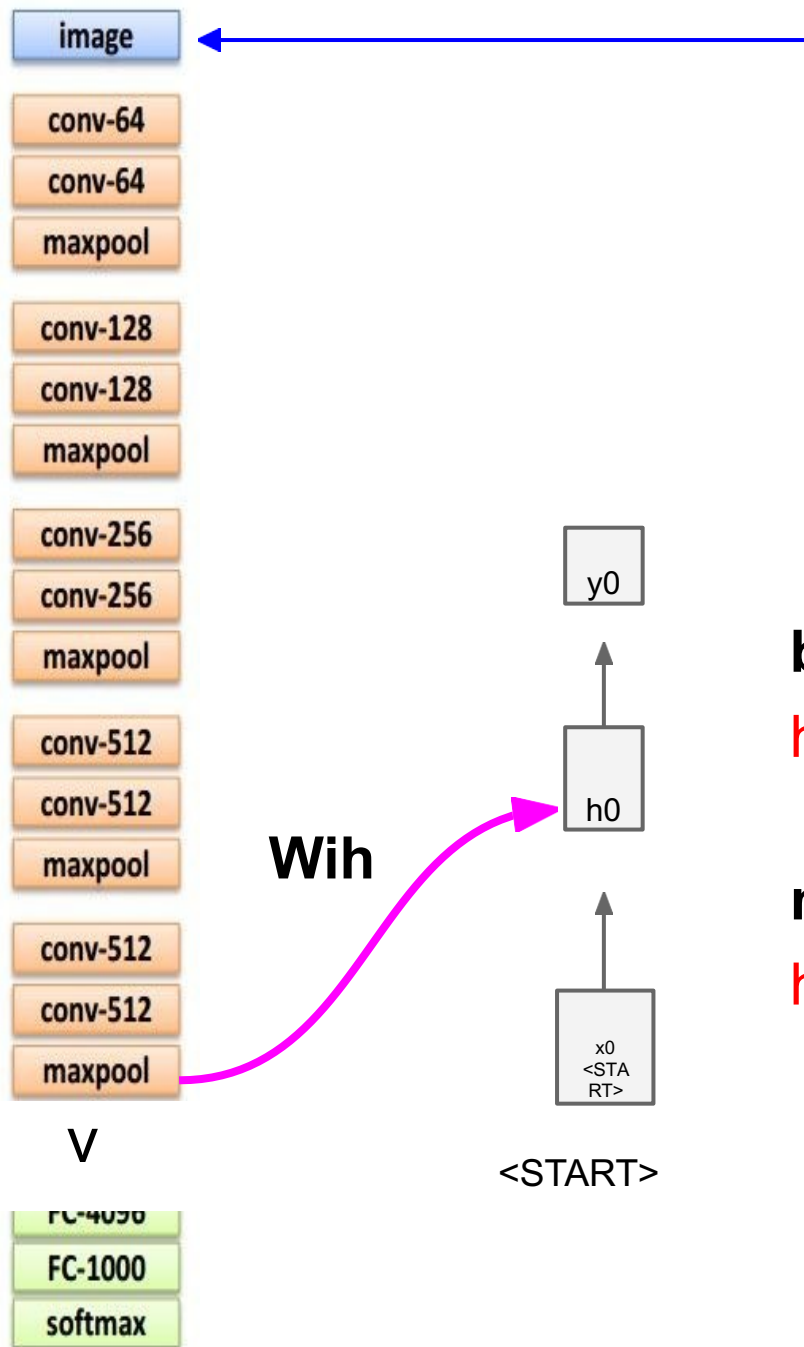
softmax



test image

x0  
<STA  
RT>

<START>



test image

**before:**

$$h = \tanh(W_{xh} * x + W_{hh} * h)$$

**now:**

$$h = \tanh(W_{xh} * x + W_{hh} * h + W_{ih} * v)$$

image

conv-64

conv-64

maxpool

conv-128

conv-128

maxpool

conv-256

conv-256

maxpool

conv-512

conv-512

maxpool

conv-512

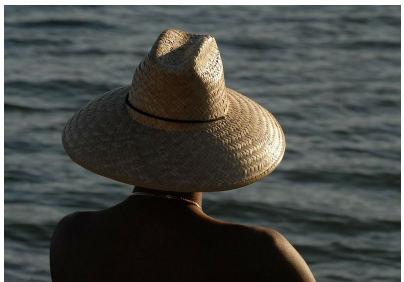
conv-512

maxpool

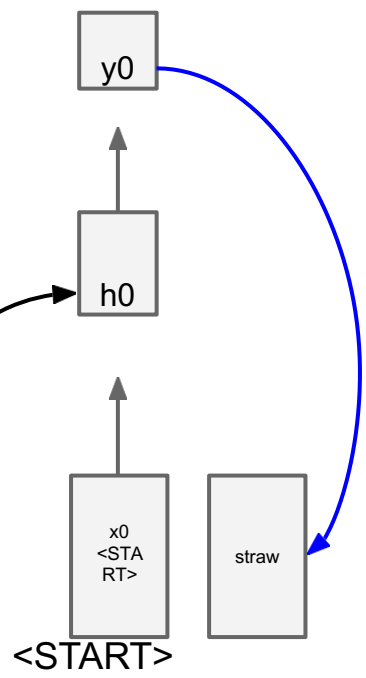
FC-4096

FC-1000

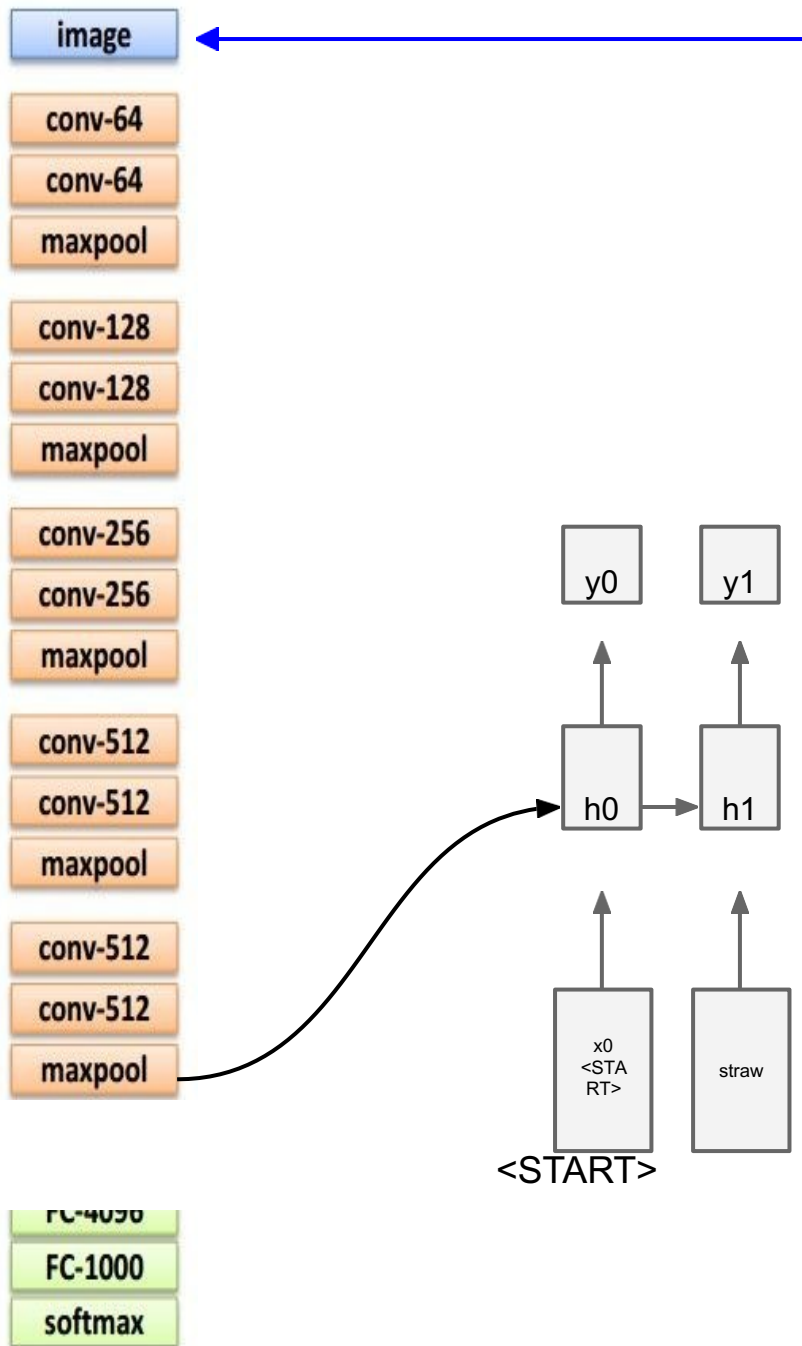
softmax



test image



sample!



test image

image



conv-64

conv-64

maxpool

conv-128

conv-128

maxpool

conv-256

conv-256

maxpool

conv-512

conv-512

maxpool

conv-512

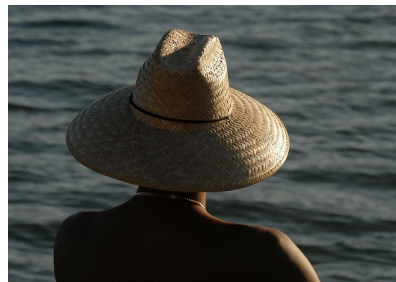
conv-512

maxpool

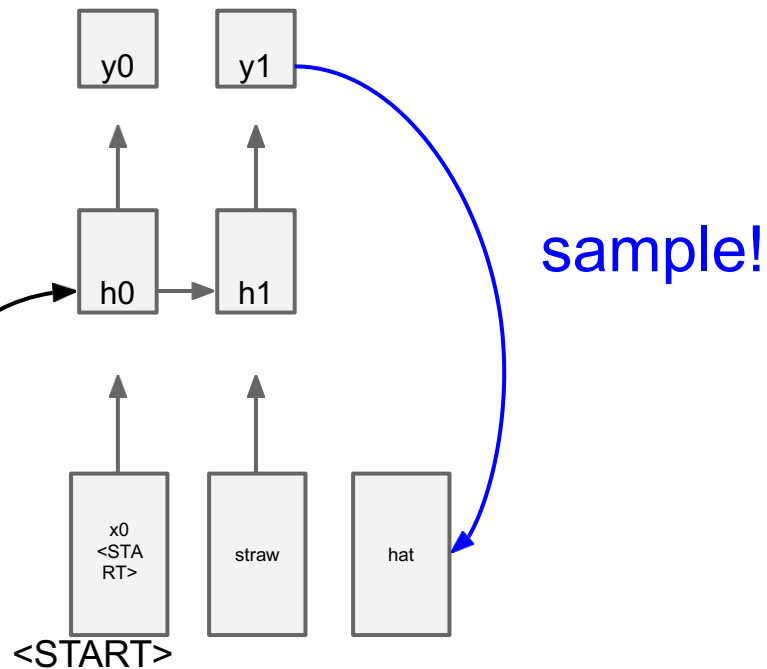
FC-4096

FC-1000

softmax



test image



image



conv-64

conv-64

maxpool

conv-128

conv-128

maxpool

conv-256

conv-256

maxpool

conv-512

conv-512

maxpool

conv-512

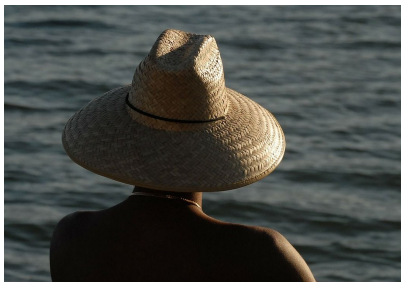
conv-512

maxpool

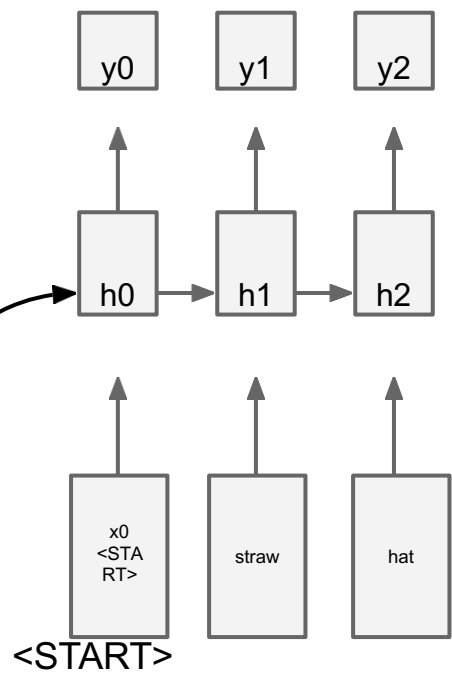
FC-4096

FC-1000

softmax



test image



image

conv-64

conv-64

maxpool

conv-128

conv-128

maxpool

conv-256

conv-256

maxpool

conv-512

conv-512

maxpool

conv-512

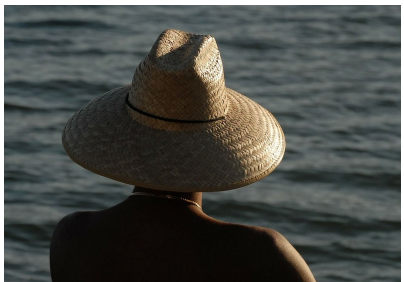
conv-512

maxpool

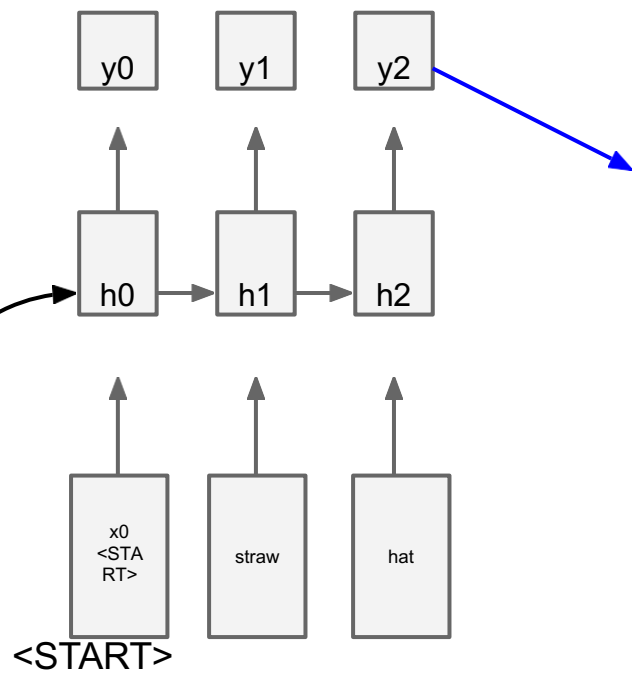
FC-4096

FC-1000

softmax



test image



sample  
<END> token  
=> finish.



# Image Captioning: Example Results

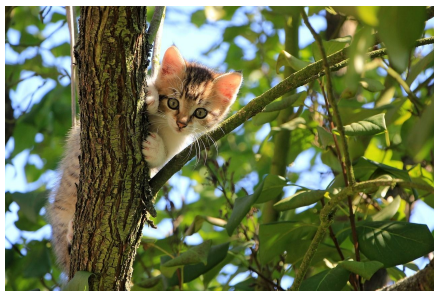
Captions generated using [neuraltalk2](#)

All images are [CC0 Public domain](#)

[cat](#) [suitcase](#) [cat](#) [tree](#) [dog](#) [bear](#)  
[surfers](#) [tennis](#) [giraffe](#) [motorcycle](#)



*A cat sitting on a  
suitcase on the floor*



*A cat is sitting on a tree  
branch*



*A dog is running in the  
grass with a frisbee*



*A white teddy bear sitting in  
the grass*



*Two people walking on  
the beach with surfboards*



*A tennis player in action  
on the court*



*Two giraffes standing in a  
grassy field*



*A man riding a dirt bike on  
a dirt track*



# Image Captioning: Failure Cases

Captions generated using [neuraltalk2](#)  
All images are [CC0 Public domain](#): [fur](#),  
[coat](#), [handstand](#), [spider.web](#), [baseball](#)



*A woman is holding a cat in her hand*



*A person holding a computer mouse on a desk*



*A woman standing on a beach holding a surfboard*

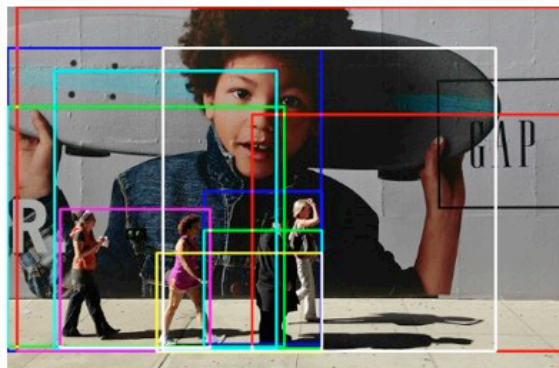


*A bird is perched on a tree branch*



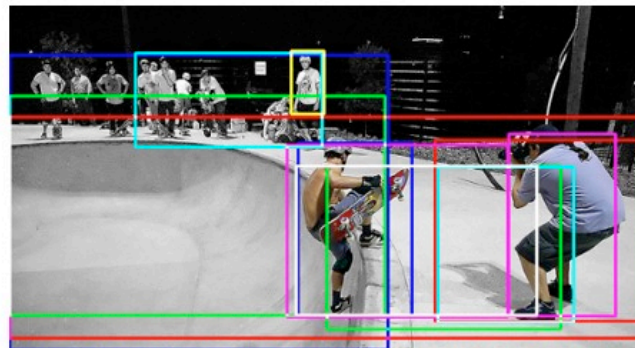
*A man in a baseball uniform throwing a ball*

# More Image Captioning Examples



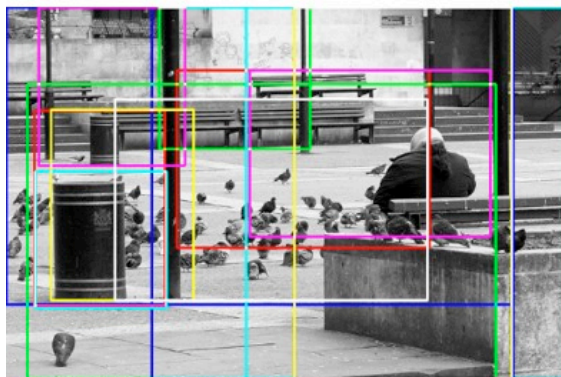
[men (0.59)] [group (0.66)] [woman (0.64)]  
 [people (0.89)] [holding (0.60)] [playing (0.61)] [tennis (0.69)]  
 [court (0.51)] [standing (0.59)] [skis (0.58)] [street (0.52)]  
 [man (0.77)] [skateboard (0.67)]

a group of people standing next to each other  
 people stand outside a large ad for gap featuring a young boy



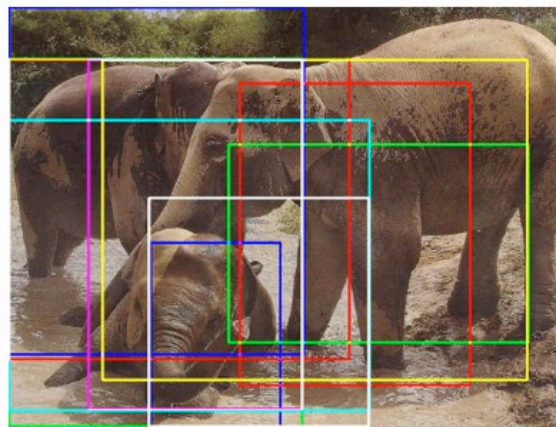
[person (0.55)] [street (0.53)] [holding (0.55)] [group (0.63)] [slope (0.51)]  
 [standing (0.62)] [snow (0.91)] [skis (0.74)] [player (0.54)]  
 [people (0.85)] [men (0.57)] [skiing (0.51)]  
 [skateboard (0.89)] [riding (0.75)] [tennis (0.74)] [trick (0.53)] [skate (0.52)]  
 [woman (0.52)] [man (0.86)] [down (0.61)]

a group of people riding skis down a snow covered slope  
 a guy on a skate board on the side of a ramp



[umbrella (0.59)] [woman (0.52)]  
 [fire (0.96)] [hydrant (0.96)] [street (0.79)] [old (0.50)]  
 [bench (0.81)] [building (0.75)] [standing (0.57)] [baseball (0.55)]  
 [white (0.82)] [sitting (0.65)] [people (0.79)] [photo (0.53)]  
 [black (0.84)] [kitchen (0.54)] [man (0.72)] [water (0.56)]

a black and white photo of a fire hydrant  
 a courtyard full of poles pigeons and garbage cans also has benches on  
 either side of it one of which shows the back of a large person facin  
 g in the direction of the pigeons



[horse (0.53)] [bear (0.71)] [elephant (0.99)] [elephants (0.95)]  
 [brown (0.68)] [baby (0.62)] [walking (0.57)] [laying (0.61)]  
 [man (0.57)] [standing (0.79)] [field (0.65)]  
 [water (0.83)] [large (0.71)] [dirt (0.65)] [river (0.58)]

a baby elephant standing next to each other on a field  
 elephants are playing together in a shallow watering hole



# How to Evaluate different captions?



1. A woman in a green shirt is getting food ready with a child , while sitting on rocks .
2. A mother and child having a picnic on a big rock with blue utensils .
3. A woman serving food for a little boy outside on a large rock .
4. A woman and a baby eating ( having a picnic ) .
5. A mother and child picnic on some rocks .

# BLEU (BiLingual Evaluation Understudy)

(Papineni et al., 2002)

- *“The closer a machine translation is to a professional human translation, the better it is.”*
- Analyzes co-occurrences of  $n$ -grams between candidate and reference sentences
  - Modified (clipped)  $n$ -gram precision
  - Brevity penalty to penalize short candidate sentences
- Has been shown in MT literature to be an insufficient metric (Callison-Burch et al., 2006)
  - Many large variations of a generated sentence can score identically
  - Higher BLEU score is not necessarily indicative of higher human-judged quality

Candidate: the the the the the the the.

Reference 1: The cat is on the mat.

Reference 2: There is a cat on the mat.

Modified Unigram Precision =  $2/7$ .



### Reference captions:

1. Latino man holding sign on the sidewalk outside promoting Quiznos-Subs .
2. A man is holding an advertisement for Quiznos Subs .
3. A man is holding a Quiznos sign next to a street .
4. A man is holding a Quiznos Sub sign .

### Candidate caption:

? Quiznos worker wearing sign .

BLEU-4 = 0.106

# METEOR

(Banerjee & Lavie, 2005)

More flexible MT metric that calculates sentence-level similarity scores as a harmonic mean of unigram precision & recall, based on:

- Exact token matching
- Stemmed tokens
- WordNet synonyms
- Paraphrases

SYSTEM	Jim went home
REFERENCE	Joe goes home

SYSTEM	Jim walks home
REFERENCE	Joe goes home

Examples from [Statistical Machine Translation slides](#)

Banerjee & Lavie, [METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments](#), ACL 2005

# CIDEr: Consensus-based Image Description Evaluation

(Vedantam et al., 2015)

- “Does a caption describe an image as most people tend to describe it?”
- Automatically evaluate for image  $I_i$  how well a candidate sentence  $c_i$  matches the **consensus** of a set of image descriptions  $S_i = \{s_{i1}, \dots, s_{im}\}$
- Intuitively, a measure of consensus should:
  - Encode how often  $n$ -grams in the candidate sentence are present in the reference sentences
  - $n$ -grams not present in the reference sentences should not be in the candidate sentence
  - $n$ -grams that commonly occur across all images in the dataset should be given lower weight, since they are likely to be less informative

In practice: perform a **Term Frequency Inverse Document Frequency (TF-IDF)**

(Robertson, 2004) weighting for each  $n$ -gram



	CIDEr-D	Meteor	ROUGE-L	BLEU-1	BLEU-2	BLEU-3	BLEU-4	date
Watson Multimodal <sup>[46]</sup>	1.123	0.268	0.559	0.773	0.609	0.461	0.344	2016-11-16
MSM@MSRA <sup>[29]</sup>	1.049	0.266	0.552	0.751	0.588	0.449	0.343	2016-10-25
G-RMI(PG-SPIDEr-TAG) <sup>[17]</sup>	1.042	0.255	0.551	0.751	0.591	0.445	0.331	2016-11-11
MetaMind/VT_GT <sup>[25]</sup>	1.042	0.264	0.55	0.748	0.584	0.444	0.336	2016-12-01
ATT-IMG (MSM@MSRA) <sup>[5]</sup>	1.023	0.262	0.551	0.752	0.59	0.449	0.34	2016-06-13
G-RMI (PG-BCMR) <sup>[16]</sup>	1.013	0.257	0.55	0.754	0.591	0.445	0.332	2016-10-30
DONOT_FAIL_AGAIN <sup>[13]</sup>	1.01	0.262	0.542	0.734	0.564	0.425	0.32	2016-11-22
DLTC@MSR <sup>[12]</sup>	1.003	0.257	0.543	0.74	0.575	0.436	0.331	2016-09-04
Postech_CV <sup>[38]</sup>	0.987	0.255	0.539	0.743	0.575	0.431	0.321	2016-06-13
feng <sup>[15]</sup>	0.986	0.255	0.54	0.743	0.578	0.434	0.323	2016-11-06
...								
Human <sup>[21]</sup>	0.854	0.252	0.484	0.663	0.469	0.321	0.217	2015-03-23

According to CIDEr, humans are in 38<sup>th</sup> place!! 🤯



# Project Abstracts due tomorrow!

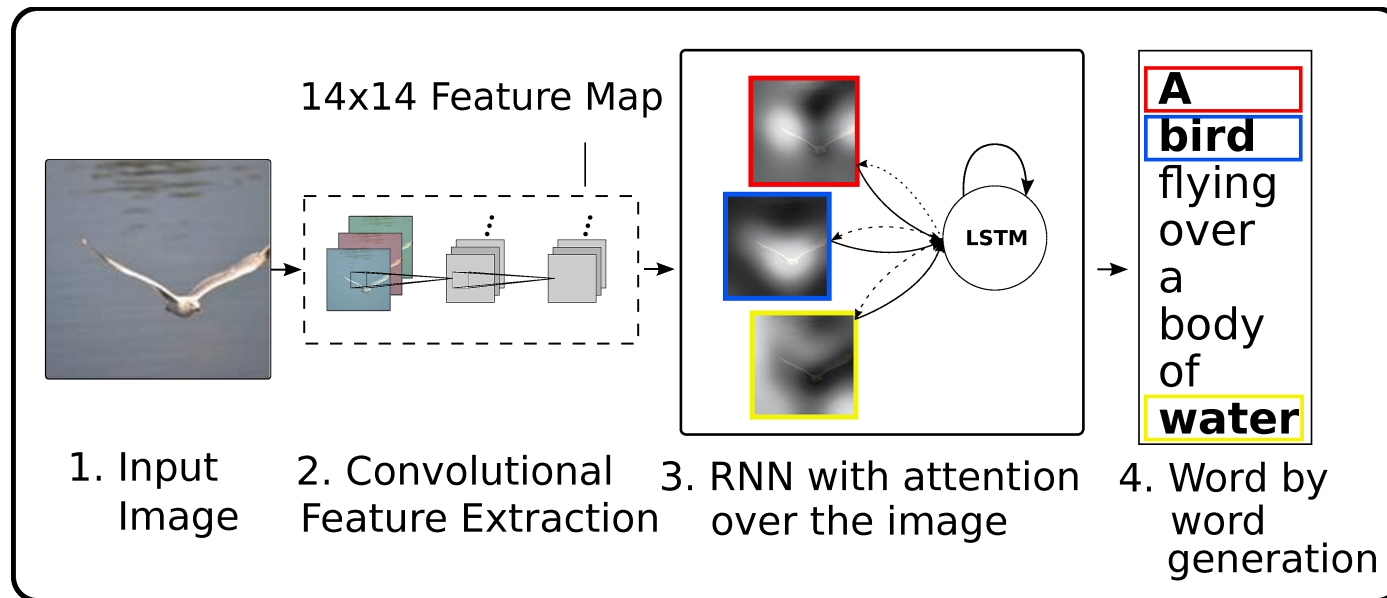
- Just need a paragraph about what you are doing.
- Email to me.
- List other people on team.
- One email per team.

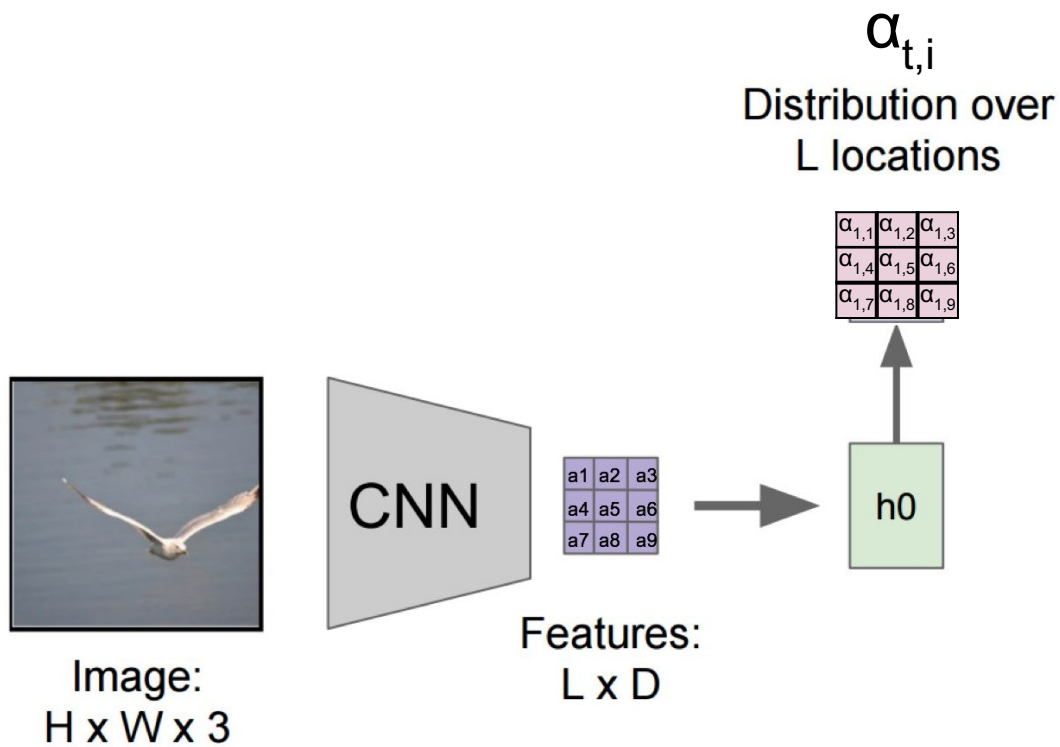
# Show, Attend and Tell

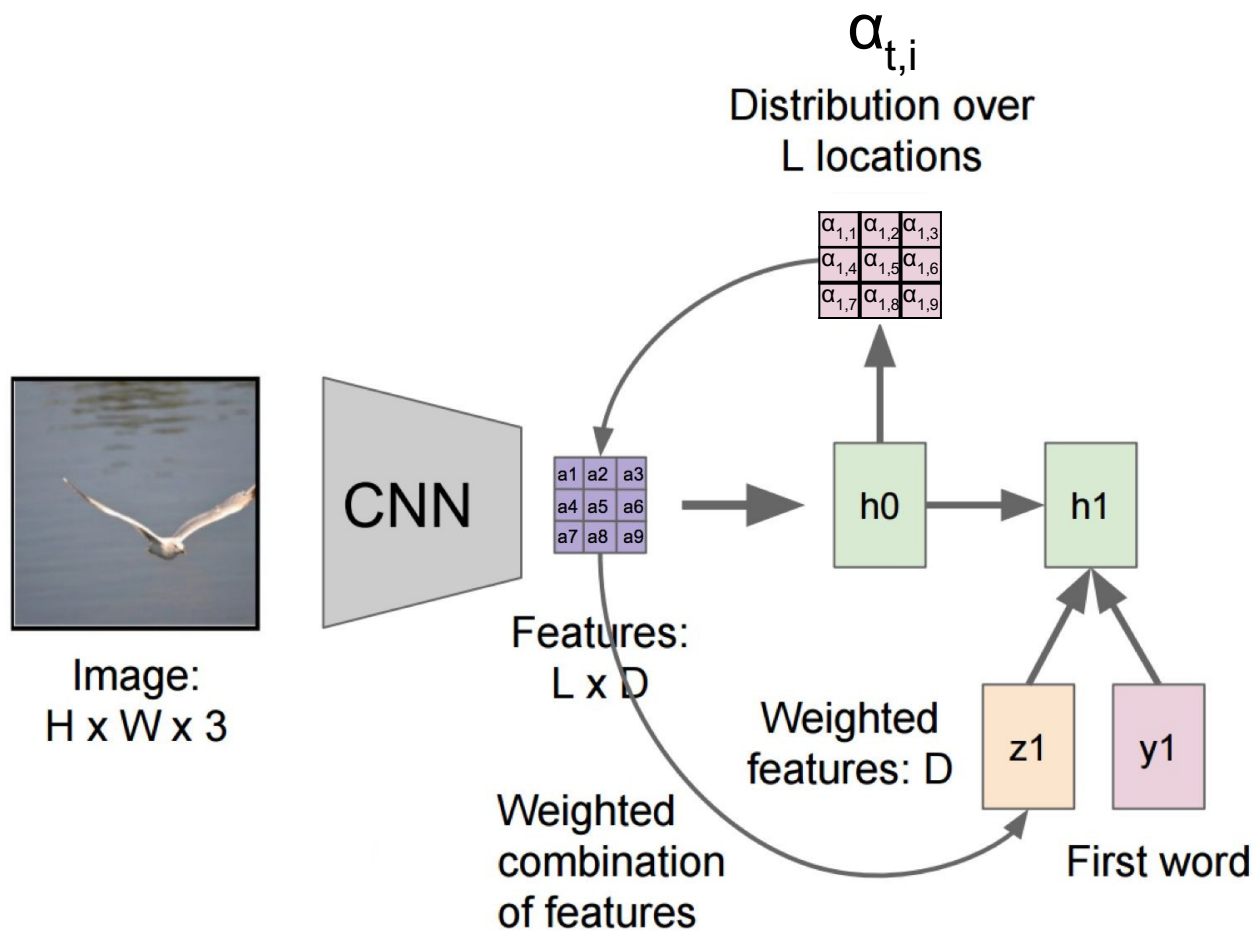
(Xu et al., 2015)

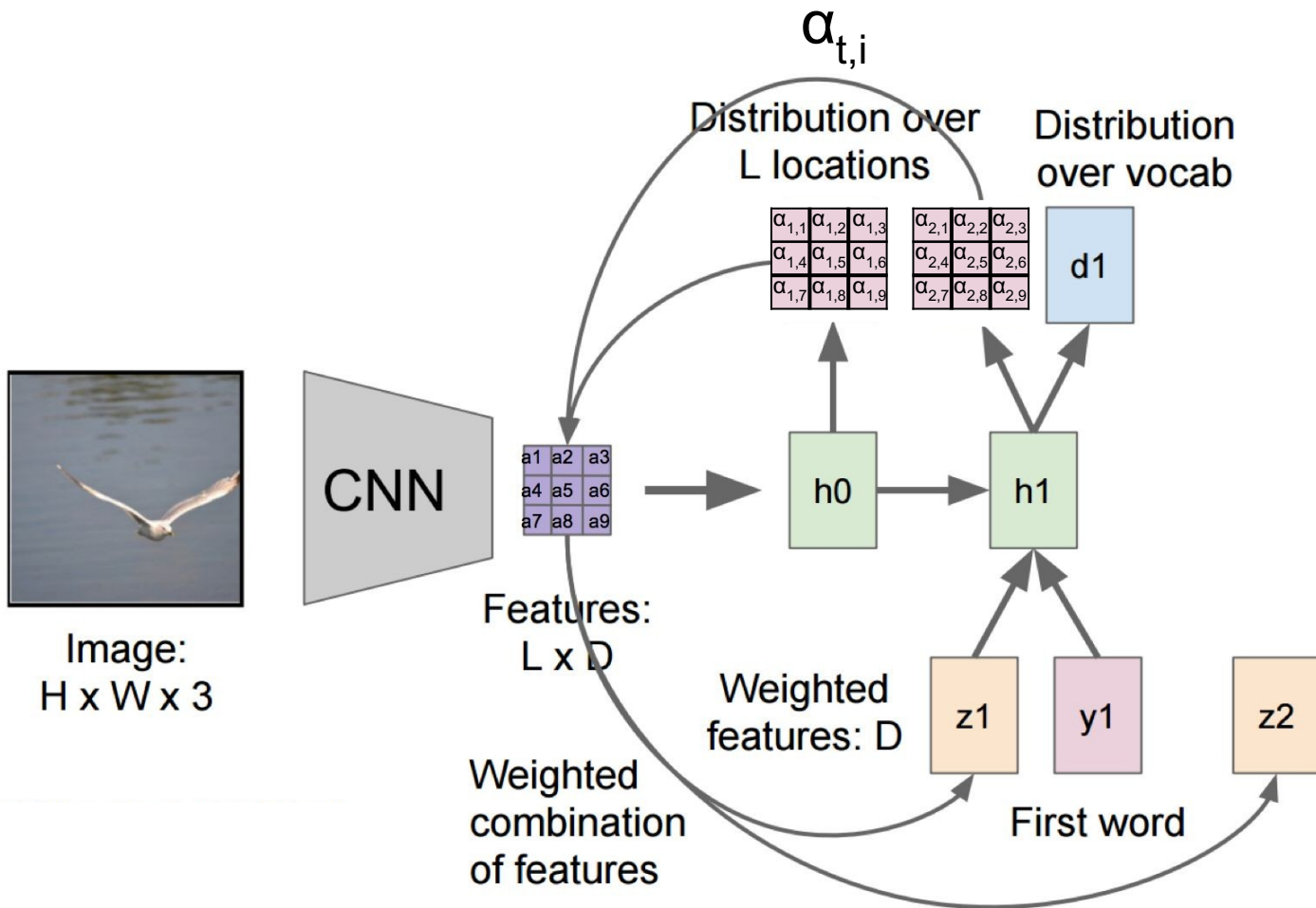
Instead of learning word detectors over image regions, consider learning an **attention model** instead

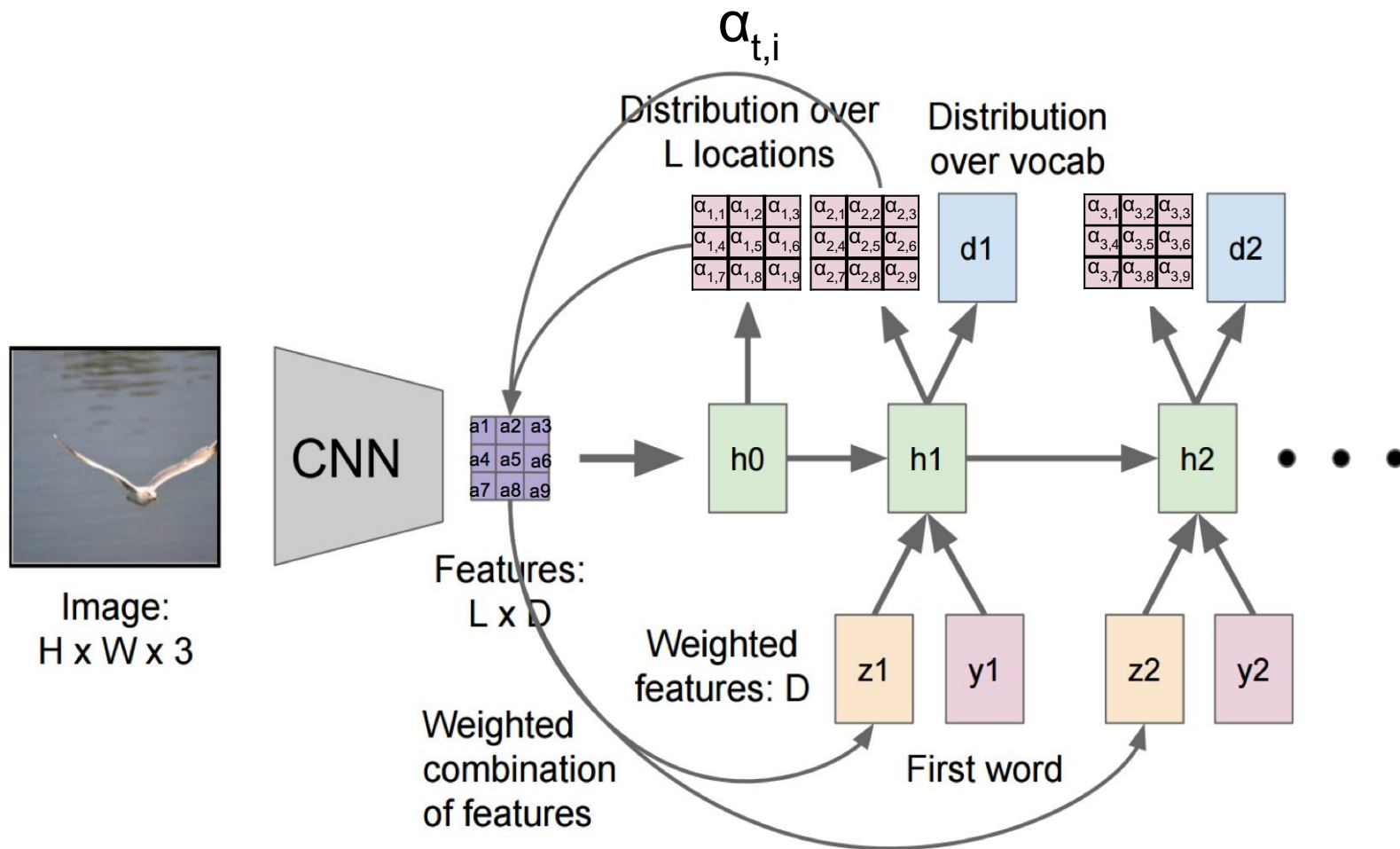
- What is visual attention?
- How to augment Show and Tell with visual attention
- Soft vs. hard attention











# Soft Attention

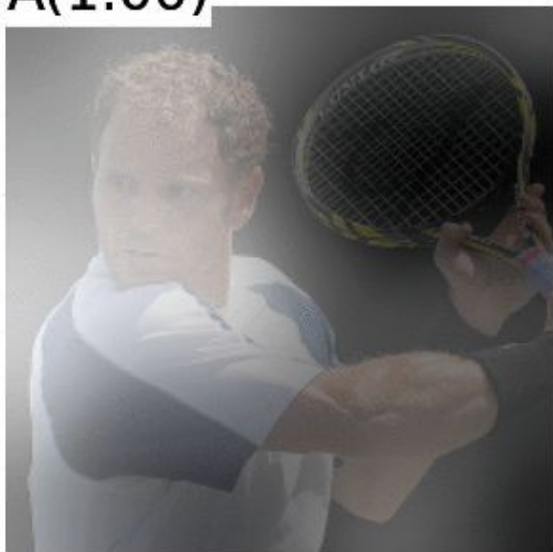
$\mathbf{z}_t$  is calculated by taking the weighted sum of all feature vectors  $\mathbf{a}$

$$\mathbf{z}_t = \sum_{i=1}^L \alpha_t[i] \cdot \mathbf{a}_i$$

- Differentiable
- Deterministic:  $\alpha_i$ 's assign relative importance to give to location  $i$  in blending the  $\mathbf{a}_i$ 's together
- Learned using standard backpropagation

# Soft Attention: Examples

A(1.00)



A(0.99)



More examples at [project website](#)

Xu et al., [Show, Attend, and Tell: Neural Image Caption Generation with Visual Attention](#), ICML 2015



# Hard Attention

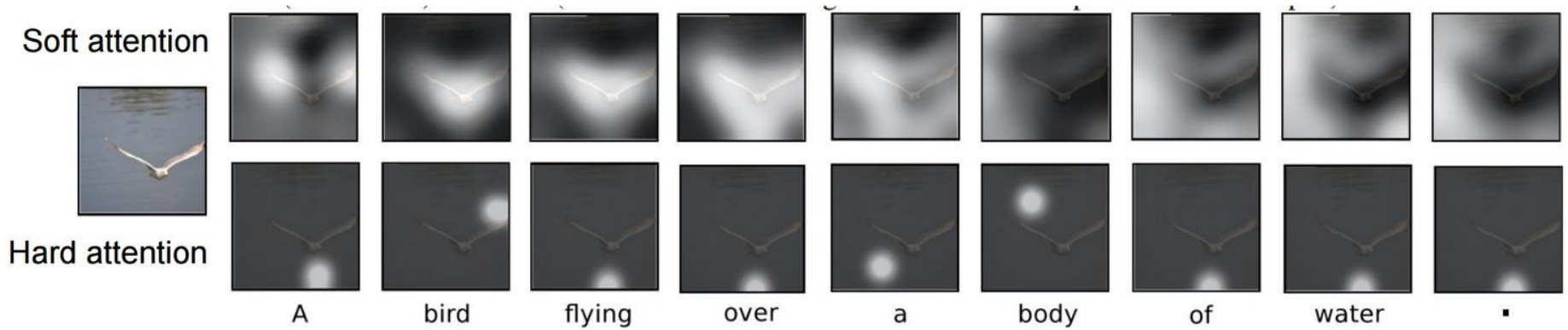
At time step  $t$ , the index into the feature vectors is sampled from the current location distribution vector  $\alpha_t$

$$k = \text{sample}(\alpha_t)$$

$$z_t = a_k$$

- Stochastic:  $\alpha_i$ 's assign probability that location  $i$  is the right place to focus for producing the next word
- Focuses on one image region at a time
- Non-differentiable due to sampling
- Set up as reinforcement learning problem:
  - Action = which area to attend to next
  - Reward = log-likelihood of caption wrt to target sentence

# Soft vs. Hard Attention

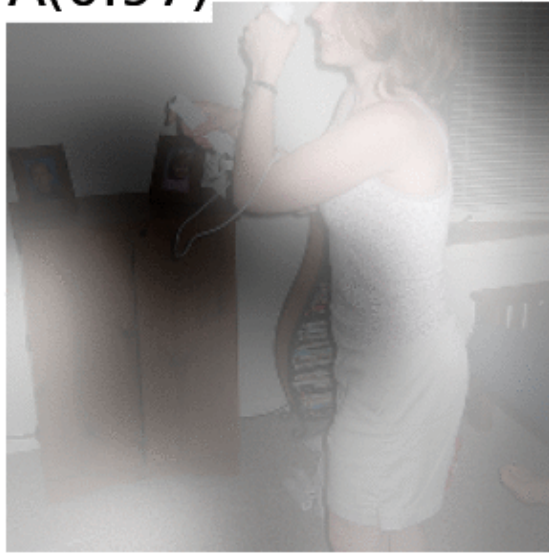


# Examples

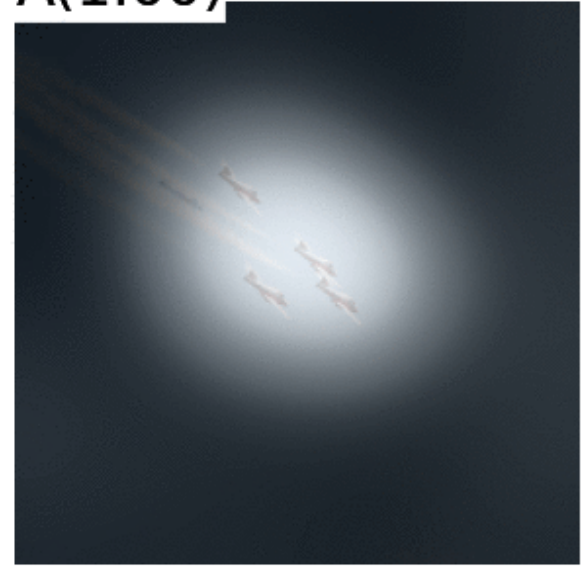
A(0.99)



A(0.97)



A(1.00)





# Engaging Image Captioning Via Personality

Kurt Shuster, Samuel Humeau, Hexiang Hu,  
Antoine Bordes, Jason Weston

# Standard (COCO) Image Captioning Models



Man in black shirt is playing guitar.

# Standard (COCO) Image Captioning Models



Man in black shirt is playing guitar.



A plate with a sandwich and salad on it.

**Good for:** testing if model understands image content

**Bad for:** engaging human reader

# Standard (COCO) Image Captioning Models



Man in black shirt is playing guitar.



A plate with a sandwich and salad on it.

**Good for:** testing if model understands image content

**Bad for:** engaging human reader



*Want to be good at both of these!!!*

What makes an utterance engaging? One answer: personality, emotion  
& style traits  
(not always just neutral, factual tone)



# Existing Work

## Neutral, factual captions:

- COCO (Chen et al., 2015) and Flickr30k (Young et al., 2014)
- Many models developed for them (see above).

## Funny captions:

- wordplay (puns) (Chandrasekaran et al., 2017)
- or training on data from humour websites (Yoshida et al., 2018).

## Using user features:

- location and age (Denton et al., 2015)
- or knowledge of the reader's active vocabulary (Park et al., 2017).

## Style transfer:

- unsupervised (Mathews et al., 2018).
- Small datasets, e.g. Senticap (800 examples), (Mathews et al., 2016)
- romantic and humorous only - FlickrStyle10K , 10k examples - Gan et al. (2017)

## MIT Personality List - 638 Traits



ideonomy.mit.edu/essays/traits.html

fb work RL BlueJeans Network Mail - jase@fb.com Calendar Workplace [708.05866] A B... Mssngr ir

### 638 Primary Personality Traits

#### Positive Traits (234 = 37%)

1. Accessible
2. Active
3. Adaptable
4. Admirable
5. Adventurous
6. Agreeable
7. Alert
8. Allocentric
9. Amiable
10. Anticipative
11. Appreciative
12. Articulate
13. Aspiring
14. Athletic
15. Attractive
16. Balanced
17. Benevolent
18. Brilliant
19. Calm
20. Capable
21. Captivating
22. Caring
23. Challenging
24. Charismatic
25. Charming
26. Cheerful

215. Hey
216. Tolerant
217. Tractable
218. Trusting
219. Uncomplaining
220. Understanding
221. Undogmatic
222. Unfoolable
223. Upright
224. Urbane
225. Venturesome
226. Vivacious
227. Warm
228. Well-bred
229. Well-read
230. Well-rounded
231. Winning
232. Wise
233. Witty
234. Youthful

#### Neutral Traits (292 = 18%)

1. Absentminded
2. Aggressive
3. Ambitious
4. Amusing
5. Artful
6. Ascetic
7. Authoritarian
8. Big-thinking
9. Boyish
10. Breezy
11. Businesslike
12. Busy
13. Casual

#### Negative Traits (292 = 46%)

1. Abrasive
2. Abrupt
3. Agonizing
4. Aimless
5. Airy
6. Aloof
7. Amoral
8. Angry
9. Anxious
10. Apathetic
11. Arbitrary
12. Argumentative
13. Arrogant
14. Artificial
15. Asocial
16. Assertive
17. Astigmatic
18. Barbaric
19. Bewildered
20. Bizarre
21. Bland
22. Blunt
23. Biosterous

# Step 1: build a dataset



Your personality: **Sarcastic**

Your comment:

**Can this island get any smaller?**

- Selected 215 personality traits
- Images from YFFC100M
- Collect captions via annotators

## Examples from the dataset



*Sarcastic*

Yes please sit by me



*Mellow*

Look at that smooth easy catch of the ball. like ballet.



*Zany*

I wish I could just run down this shore!



*Contradictory*

Love what you did with the place!



*Contemptible*

I can't believe no one has been taking care of this plant. Terrible



*Energetic*

About to play the best tune you've ever heard in your life. Get ready!

## Examples from the dataset



*Kind*  
they left me a parking spot



*Spirited*  
That is one motor cycle enthusiast!!!



*Creative*  
Falck alarm, everyone. Just a Falck alarm.



*Crazy*  
I drove down this road backwards at 90 miles per hour three times



*Morbid*  
I hope this car doesn't get into a wreck.



*Questioning*  
Why do people think its cool to smoke cigarettes?

# Step 1: Collect a large supervised dataset

Table 1: PERSONALITY-CAPTIONS dataset statistics.

Split	train	valid	test
Number of Examples	186,858	5,000	10,000
Number of Personality Types	215	215	215
Vocabulary Size	35559	5557	8137
Average Tokens per Caption	11.6	11.2	11.4

## Step 2: Build strong models

*We make use of state-of-the-art in vision and language domains to build our models:*

### Image Encoder:

- ResNeXt (Xie et al., 2016) trained on 3.5 billion Instagram pictures following Mahajan et al. (2018), which we call *ResNeXt-IG-3.5B*.
- *Shown to work very well on ImageNet classification (but not captioning).*

### Text Encoder:

- Transformer (Vaswani et al., 2017) trained on 1.7 billion Reddit dialogue examples, following (Mazare' et al., 2018).
- *Shown to work very well for PersonaChat dialogue (but not captioning).*

Models: we consider both generative and retrieval models.

- Generative: *consider three widely used architectures:*

- ShowTell (Vinyals et al., 2015)
- ShowAttTell (Xu et al., 2015)
- UpDown (Anderson et al., 2018)



*Use ResNeXt-IG-3.5B and add learnt personality features to each decoder step*



Models: we consider both generative and retrieval models.

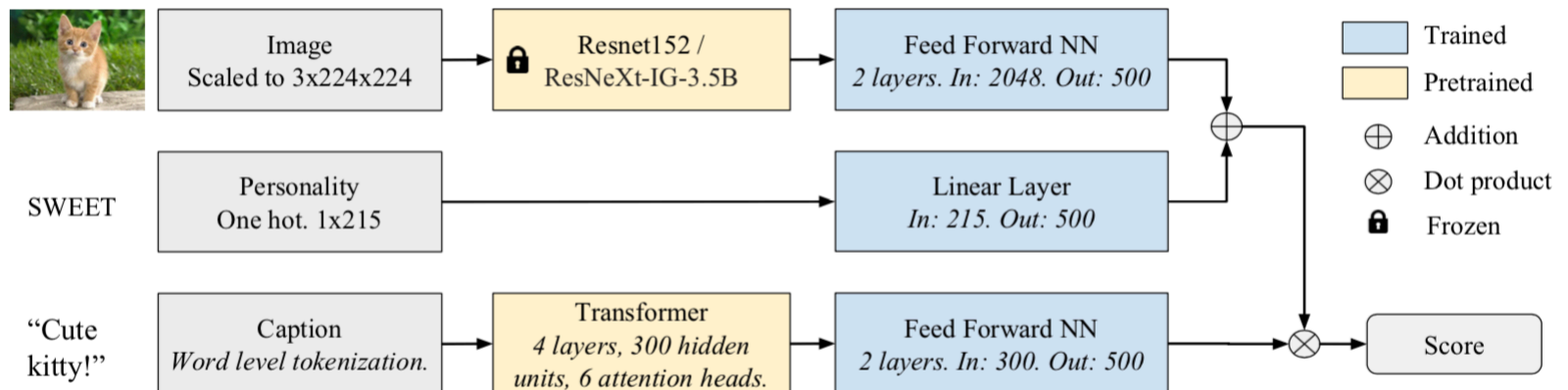
- Generative: *consider three recent best architectures:*

- ShowTell (Vinyals et al., 2015)
- ShowAttTell (Xu et al., 2015)
- UpDown (Anderson et al., 2018)

*Use ResNeXt-IG-3.5B and add learnt personality features to each decoder step*

- Retrieval:

### TransResNet



Our generative models are good at understanding image content.

Table 3: Generative model performance on COCO caption using the test split of (Karpathy & Fei-Fei, 2015)

Method	Image Encoder	BLEU1	BLEU4	ROUGE-L	CIDEr	SPICE
Adaptive (Lu et al., 2017)	ResNet	74.2	32.5	-	108.5	19.5
Att2in (Rennie et al., 2017)	ResNet	-	33.3	55.3	111.4	-
NBT (Lu et al., 2018)	ResNet	75.5	34.7	-	107.2	20.1
UPDOWN (Anderson et al., 2018)	ResNet FRCNN	<b>79.8</b>	36.3	56.9	120.1	<b>21.4</b>
SHOWTELL (Our)	ResNet152	75.2	31.5	54.2	103.9	18.4
SHOWATTTELL (Our)	ResNet152	76.5	32.4	55.1	109.7	19.2
UPDOWN (Our)	ResNet152	77.0	33.9	55.6	112.7	19.6
SHOWTELL (Our)	ResNeXt-IG-3.5B	78.2	35.0	56.6	119.9	20.8
SHOWATTTELL (Our)	ResNeXt-IG-3.5B	78.8	35.6	57.1	121.8	20.6
UPDOWN (Our)	ResNeXt-IG-3.5B	79.3	<b>36.4</b>	<b>57.5</b>	<b>124.0</b>	21.2

## Our retrieval models are good at understanding image content.

Table 4: Retrieval model performance on Flickr30k and COCO caption using the splits of (Karpathy & Fei-Fei, 2015). COCO caption performance is measured on the 1k image test split.

Model	Text Pre-training	Flickr30k			COCO		
		R@1	R@5	R@10	R@1	R@5	R@10
UVS (Kiros et al., 2014)	-	23.0	50.7	62.9	43.4	75.7	85.8
Embedding Net (Wang et al., 2018)	-	40.7	69.7	79.2	50.4	79.3	69.4
sm-LSTM (Huang et al., 2016)	-	42.5	71.9	81.5	53.2	83.1	91.5
VSE++ (ResNet, FT) (Faghri et al., 2017)	-	52.9	80.5	87.2	64.6	90.0	95.7
GXN (i2t+t2i) (Gu et al., 2017)	-	56.8	-	89.6	<b>68.5</b>	-	<b>97.9</b>
<i>TransResNet model variants:</i>							
Transformer, ResNet152	Full	10.3	27.3	38.8	21.7	45.6	58.9
Bag of words ResNeXt-IG-3.5B	None	50.0	81.1	90.0	51.6	85.3	93.4
Transformer ResNeXt-IG-3.5B	None	55.6	83.2	90.5	64.0	90.6	96.3
Bag of words ResNeXt-IG-3.5B	Word	58.6	87.2	92.9	54.7	87.1	94.5
Transformer ResNeXt-IG-3.5B	Word	<b>68.4</b>	<b>90.6</b>	<b>95.3</b>	67.3	<b>91.7</b>	96.5

## Our generative models are good at using personality

Table 5: Generative model caption performance on the PERSONALITY-CAPTIONS test set.

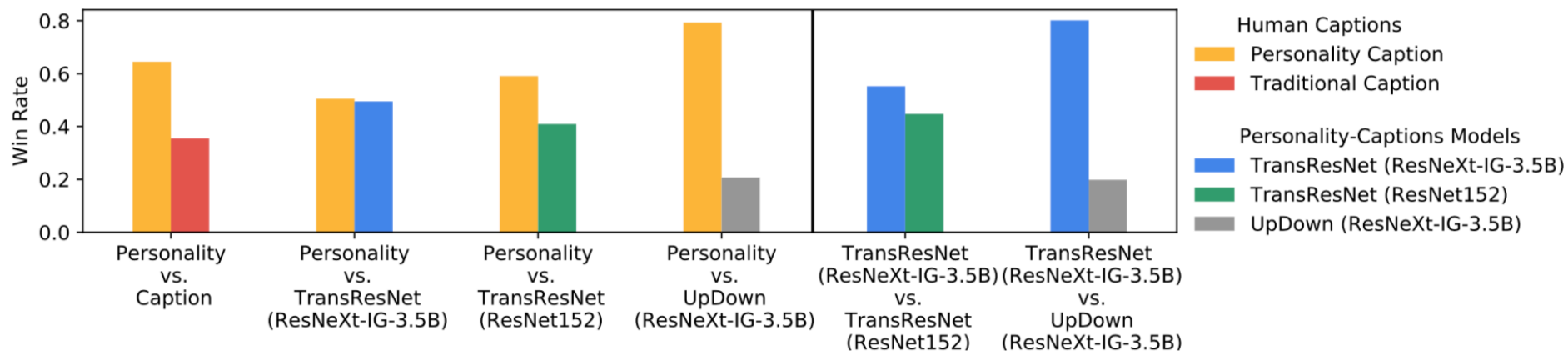
Method	Image Encoder	Personality Encoder	BLEU1	BLEU4	ROUGE-L	CIDE <sub>r</sub>	SPICE
SHOWTELL	ResNet152	Yes	12.4	1.4	13.2	14.5	1.6
SHOWATTTELL	ResNet152	Yes	15.3	1.3	13.1	15.2	3.4
UPDOWN	ResNet152	Yes	15.4	1.4	14.6	16.9	4.9
SHOWTELL	ResNeXt-IG-3.5B	No	15.2	0.9	13.3	14.4	4.6
SHOWATTTELL	ResNeXt-IG-3.5B	No	13.8	0.9	13.1	17.6	5.4
UPDOWN	ResNeXt-IG-3.5B	No	14.3	1.0	13.5	18.0	7.0
SHOWTELL	ResNeXt-IG-3.5B	Yes	14.2	1.2	14.5	15.4	2.2
SHOWATTTELL	ResNeXt-IG-3.5B	Yes	15.0	1.4	14.6	18.8	5.9
UPDOWN	ResNeXt-IG-3.5B	Yes	<b>15.6</b>	<b>1.6</b>	<b>15.0</b>	<b>22.0</b>	<b>7.3</b>

## Our retrieval models are good at using **personality**

Table 6: Results for TransResNet retrieval variants on the PERSONALITY-CAPTIONS test set.

Text Encoder	Pre-training	Image Encoder	Personality Encoder	R@1
Transformer	Full	ResNet152	No	16.6
Bag of Words	None	ResNet152	Yes	24.2
Transformer	None	ResNet152	Yes	26.8
Bag of Words	Word	ResNet152	Yes	28.5
Transformer	Full	ResNet152	Yes	34.4
Transformer	Full	ResNeXt-IG-3.5B	No	38.5
Bag of Words	None	ResNeXt-IG-3.5B	Yes	38.6
Transformer	None	ResNeXt-IG-3.5B	Yes	42.9
Bag of Words	Word	ResNeXt-IG-3.5B	Yes	45.7
Transformer	Full	ResNeXt-IG-3.5B	Yes	<b>53.5</b>

Human evaluation studies: *our best model is close to matching human performance*





---

**Standard captioning output:** A plate with a sandwich and salad on it.

**Our model with different personality traits:**

<i>Sweet</i>	That is a lovely sandwich.
<i>Dramatic</i>	This sandwich looks so delicious! My goodness!
<i>Anxious</i>	I'm afraid this might make me sick if I eat it.
<i>Sympathetic</i>	I feel so bad for that carrot, about to be consumed.
<i>Arrogant</i>	I make better food than this
<i>Optimistic</i>	It will taste positively wonderful!
<i>Money-minded</i>	I would totally pay \$100 for this plate.

---



**Standard Captioning Model:** man in black shirt is playing guitar

**Our model with different personas:**

**Artful:** He has the most perfect technique of any solo artist

**Overimaginative:** I'm thinking he could grab that guitar and bust out Stairway to Heaven right now. In doing so, he could summon all the long-gone rock greats.




**Romantic:** This guitarist is so cute I want to take him on a date!

**Arrogant:** He holds the guitar wrong. I would do a much better job if I was in the photo.

**Absentminded:** Okay guys. What do I do now? Is this a banjo?



## More examples of our best model

Image	Personality	Generated comment
	Sweet	I love, love, love these chairs! I want the big one in my house!
	Vague	This chair is either covered in snow or the snow is covered in the chair.
	Cultured	These chairs remind me of the Swedish interior design revolution of the 70's.
	Paranoid	What if someone fell off those chairs.
	Overimaginative	Those chairs look like they could be in a doll house.
	Arrogant	I've seen better sunsets elsewhere.
	Overimaginative	that sunset is so orange it could be a fruit
	Vague	It's the sunset.
	Optimistic	The sunset makes look forward to a happy tomorrow.
	Charming	The way the sun is hitting the water makes for a romantic evening.
	Sweet	What a cute puppy, reminds me of my friends.
	Skeptical	I don't think this dog will bite me.
	Sympathetic	poor dog! It looks so hungry :c
	Vague	it's a dog
	Wishful	I wish that I had a dog as cute as him.



## More examples of our best model



---

Cultured	I love a cultural celebration.
Skeptical	I'm not sure if these are guys in costumes or time travelers.
Sweet	I love that they are celebrating their traditions and culture.
Overimaginative	They look like they could be dancers in a fantasy movie with dragons!
Sympathetic	I feel sorry for him having to wear that

---



Romantic	If I was an insect, I would definitely make this my mate.
Humble	I am grateful that spiders eat these disgusting bugs.
Paranoid	What is going on? Are these insects dangerous?
Creative	I made something like this from colored toothpicks once
Money-minded	how much are those? those looks expensive




---



Happy	That is so cool! I I love street art!
Optimistic	The future is bright for people who can dream in artistic ways.
Critical	I do believe this taggers verbage is a tad junvenile
Charming	What a charming wall.
Adventurous	I think I could create art like that, I will go learn and take action.

---

## More examples of our best model

Image	Personality	Generated comment
	Adventurous	This biking event looks like something that I would try!
	Vague	Those people are riding a bike.
	Charming	I bet a wonderful couple uses this bike to tour the countryside together.
	Optimistic	A hopeful cyclist trying to catch up to the pack
	Paranoid	What if all those bikes just tipped over!
	Adventurous	I am so ready for the conference.
	Cultured	This conference is one of the most important ones in the country.
	Vague	The organization on that table is uncertain.
	Dramatic	OMG!! This ceremony is frightening!
	Sympathetic	I feel bad for these people being so cramped in this room.
	Old-fashioned	Such old fashioned script, a true lost art.
	Charming	I could use these to write to my loved ones.
	Argumentative	Can you even read this through all the jpeg artifacts?
	Anxious	I hope this paper doesnt tear, history will be destroyed.
	Dramatic	Some of the most profound things ever written have been on linen.

## More examples of our best model

---



Happy	It finally snowed, it makes me feel awesome
Wishful	I wish there was enough for snow angels.
Boyish	Can I go sledding now?
Romantic	What a beautiful frost! Looks like the perfect place to fall in love!
Cultured	The white of the snow provides a glistening contrast to the dead trees.

---



Wishful	I wish I could have a life as easy as a plant.
Money-minded	This plant is probably worth a lot of money
Critical	the leaf is ruining the picture
Humble	This plant is a symbol of life in humble opinion. Just gorgeous!
Paranoid	If you eat this leaf it definitely will not poison you. Or will it...


---



Romantic	This valentine concert is for lovers.
Boyish	It's always fun to get down and jam with the boys!
Creative	musician performing a song of theirs
Sweet	oh what lovely young musicians
Money-minded	I wonder how much the musicians have in student loan debt.

---

# Human Evaluation Examples

Image and Pers.	Use pers.	Captioning	Caption
	No	Standard	A city on the background, a lake on the front, during a sunset.
	No	Engaging	Talk about summer fun! Can I join? :)
	Yes	Human	i feel moved by the sunset
	Yes	TransResNet	The water at night is a beautiful sight.
	Yes	UPDOWN	This is a beautiful sunset!
Spirited			
	No	Standard	Rose colored soft yarn.
	No	Engaging	I really want to untangle that yarn.
	Yes	Human	I cannot believe how yummy that looks.
	Yes	TransResNet	What is up with all the knitting on my feed
	Yes	UPDOWN	I would love to be a of that fruit!
Ridiculous			
	No	Standard	A beautiful mesa town built into the cliffs.
	No	Engaging	That is a strange cave
	Yes	Human	It must be very dangerous if children play there
	Yes	TransResNet	I hope my kids don't climb on this.
	Yes	UPDOWN	I hope this is a beautiful place.
Maternal			



Sophisticated

No	Standard	Hockey players competing for control of the hockey puck.
No	Engaging	Great save, goalie!!
Yes	Human	Hockey is a little too barbaric for my taste.
Yes	TransResNet	Hockey players gracefully skate across the ice.
Yes	UPDOWN	This hockey is like they are a great of the game.



Curious

No	Standard	two people walking through a snowy forest.
No	Engaging	Too cold for me.
Yes	Human	I wonder what's at the finish line for these guys?
Yes	TransResNet	I wonder why they are running.
Yes	UPDOWN	I wonder what they are a?



Happy

No	Standard	Hollywood Tower at Night
No	Engaging	I went to that theme park, but was too scared to get on that ride!
Yes	Human	I am so excited to be here!
Yes	TransResNet	I remember going to disney world, it was one of the best trips I've ever done.
Yes	UPDOWN	This looks like a beautiful view!



## Multi-Modal Open-Domain Dialogue

Kurt Shuster\*, Eric Michael Smith\*, Da Ju, Jason Weston

<https://arxiv.org/pdf/2010.01082.pdf>

- 2.7-billion-parameter Transformer sequence-to-sequence model based on the [BlenderBot](#) open-domain chatbot, trained on the image tasks [COCO Captions](#) and [Image-Chat](#), and using an image encoder from [Faster R-CNN](#).



These flowers are so beautiful, I would love to have a garden like this one day.

Yeah, me too! What kinds of flowers would you plant?

I'm not sure, I'd have to do a lot of research to find out what would grow well in my area.

That's a good point - honestly, I'm not sure what would grow well here. Do you have a big backyard for gardening?

I live in the suburbs, so I don't have a huge yard, but I do have a small garden in my front yard.

Oh, that's great. Would you grow vegetables as well or just flowers?

Vegetables, herbs, flowers, fruits and nuts are all things I grow in my garden.

That would be nice! I'd love to have apple trees and then eat fresh apples every fall.

I would grow all of the above. I love fresh fruit and veggies.

Do you like coconuts? People either love or hate them, I think.

I've never had a coconut. I've heard they're good for you though.



I would love to take this teddy bear for a spin on my motorcycle.

Ha, would the teddy bear ride in a basket in front of you or in a sidecar?

In front of me, of course. I don't want to get hit by a car.

Yes, good point - the sidecar would take up a lot of room

I think it would be fun to ride on the back of a motorbike with a stuffed animal in the basket.

Do you ride your motorbike often?

I used to ride a lot when I was younger, but I haven't been on one in a long time.