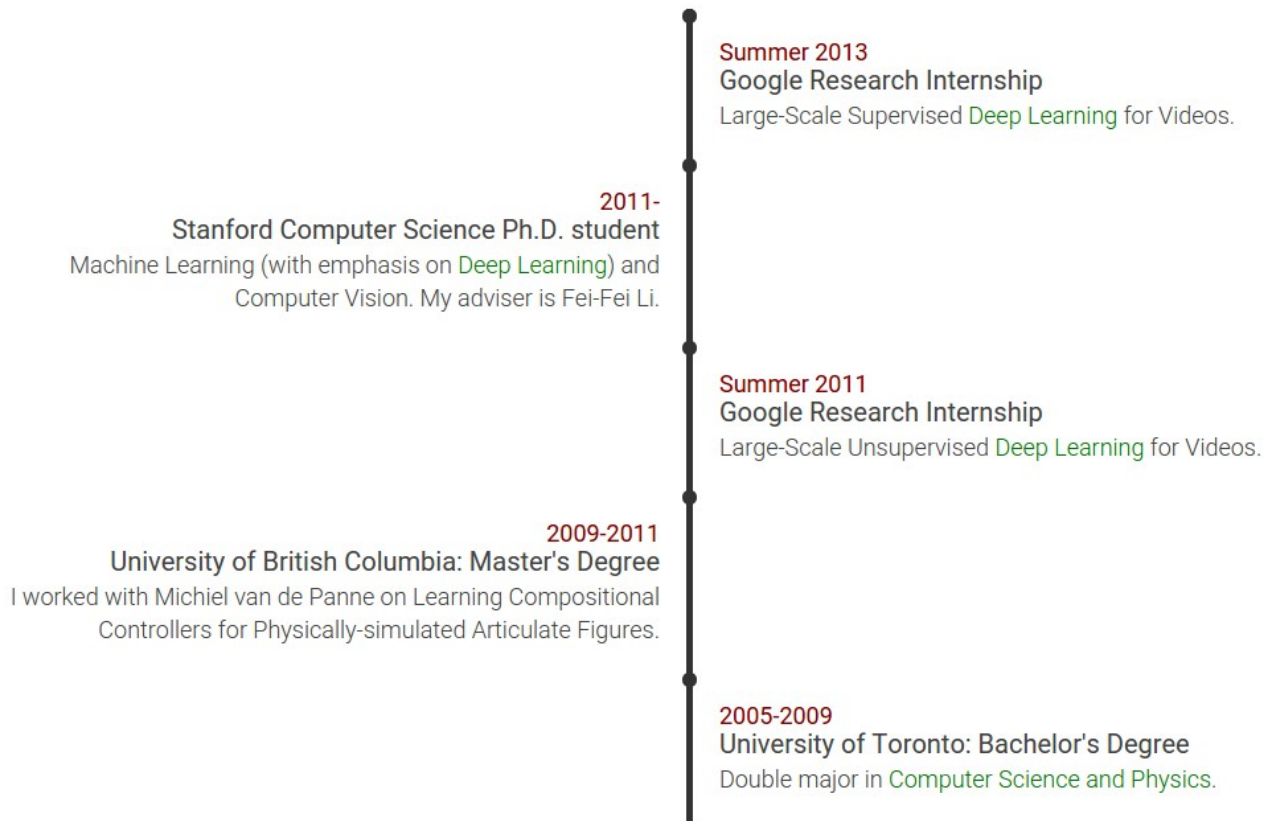


Automated Image Captioning with ConvNets and Recurrent Nets

Andrej Karpathy, Fei-Fei Li





"I like my data large, my algorithms simple, and my labels weak."

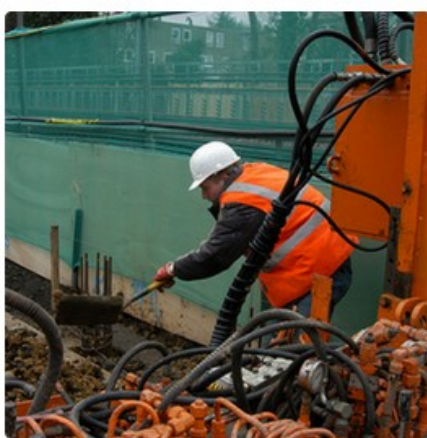
Automated Image Captioning with ConvNets and Recurrent Nets

Andrej Karpathy, Fei-Fei Li





"man in black shirt is playing guitar."



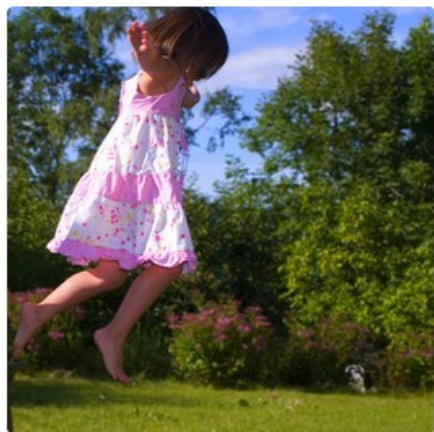
"construction worker in orange safety vest is working on road."



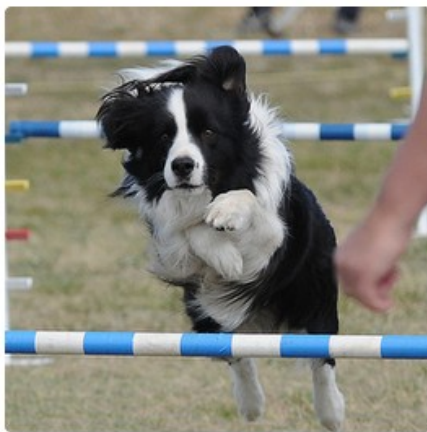
"two young girls are playing with lego toy."



"boy is doing backflip on wakeboard."



"girl in pink dress is jumping in air."



"black and white dog jumps over bar."



"young girl in pink shirt is swinging on swing."



"man in blue wetsuit is surfing on wave."





natural language



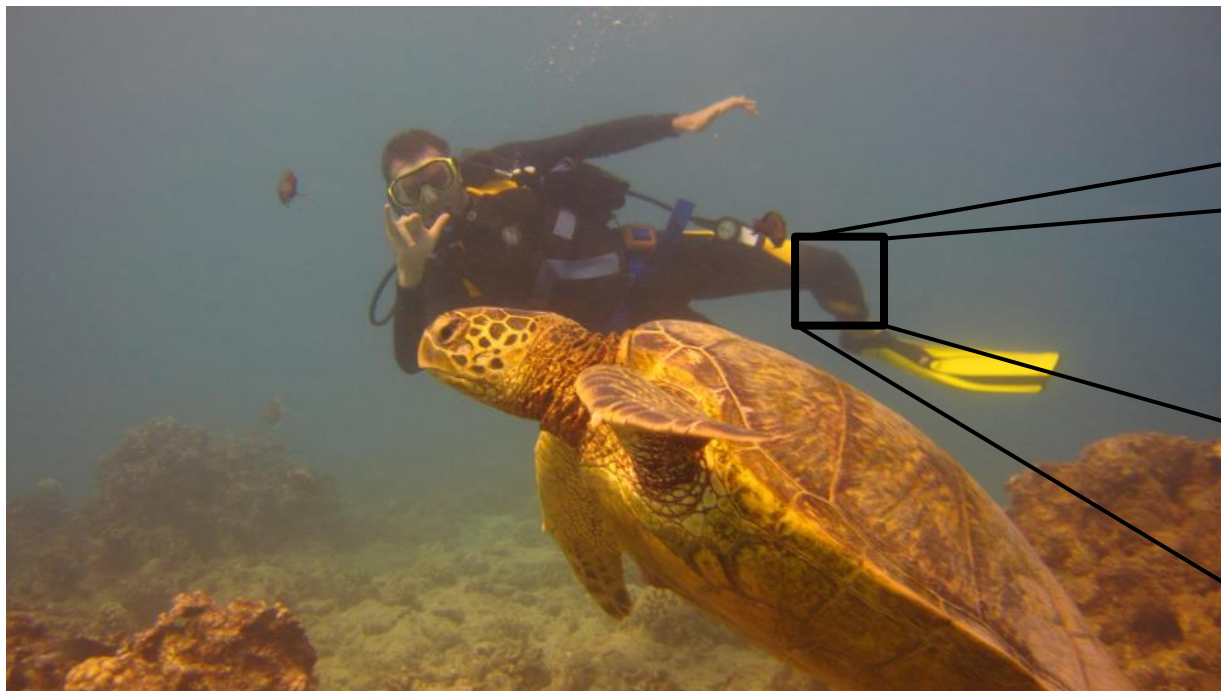
images of me scuba diving next to turtle

images of me scuba diving next to turtle



Very hard task

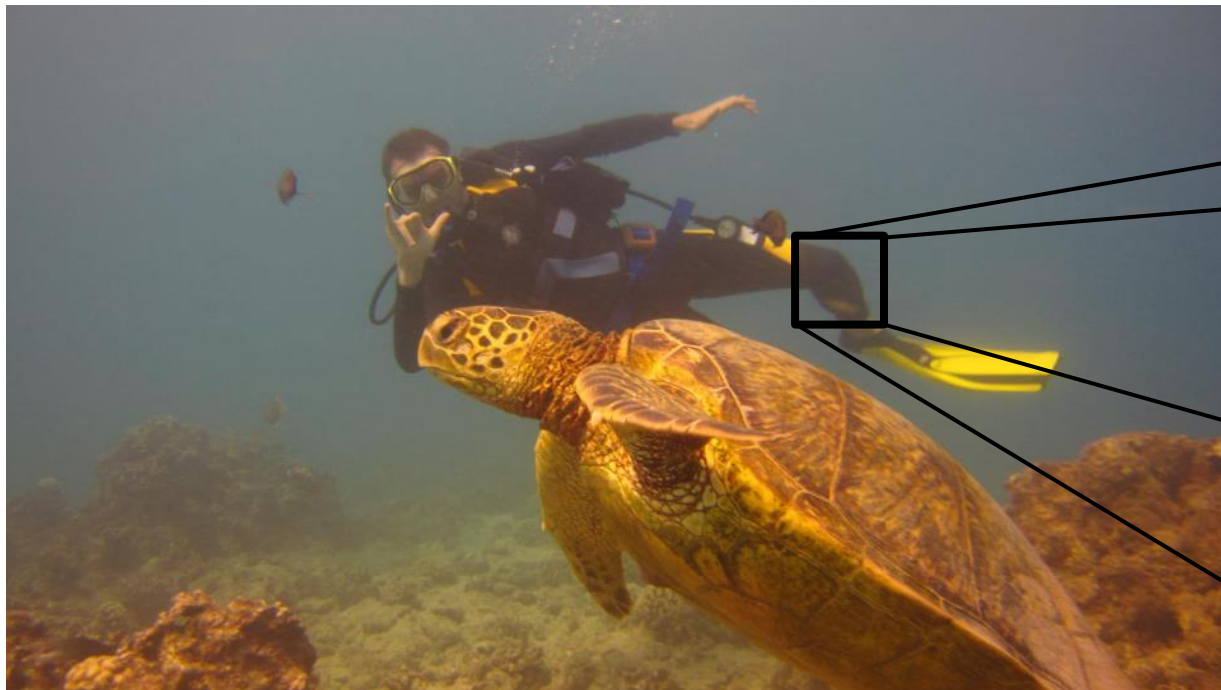
images of me scuba diving next to turtle



08	02	22	97	38	15	00	40	00	75	04	05	07	78	53	12	86	77	91	08
49	49	99	40	17	81	12	57	80	87	17	40	98	43	69	48	04	56	62	00
81	49	31	73	55	79	14	29	93	71	40	67	53	88	30	03	49	13	36	65
52	70	95	23	04	60	11	42	69	24	68	56	01	32	56	71	37	02	36	91
22	31	16	71	51	67	63	89	41	92	36	54	22	40	40	28	66	33	13	80
24	47	32	60	99	03	45	02	44	75	33	53	78	36	84	20	35	17	12	50
32	98	81	28	64	23	67	10	26	38	40	67	59	54	70	66	18	38	64	70
67	26	20	68	02	62	12	20	95	63	94	39	63	08	40	91	66	49	94	21
24	55	58	05	66	73	99	26	97	17	78	78	96	83	14	88	34	89	63	72
21	36	23	09	75	00	76	44	20	45	35	14	00	61	33	97	34	31	33	95
78	17	53	28	22	75	31	67	15	94	03	80	04	62	16	14	09	53	56	92
16	39	05	42	96	35	31	47	55	58	88	24	00	17	54	24	36	29	85	57
55	56	00	48	35	71	89	07	05	44	44	37	44	60	21	58	51	54	17	58
19	80	81	68	05	94	47	69	28	73	92	13	86	52	17	77	04	89	55	40
04	52	08	83	97	35	89	16	07	97	57	32	16	26	26	79	33	27	98	66
88	36	68	87	57	62	20	72	08	46	33	67	46	55	12	32	63	93	53	69
04	42	16	73	38	25	39	11	24	94	72	18	08	46	29	32	40	62	76	36
20	69	36	41	72	30	23	88	34	62	99	69	82	67	59	85	74	04	36	16
20	73	35	29	78	31	90	01	74	31	49	71	48	86	81	16	29	57	05	54
01	70	54	71	83	51	54	69	16	92	33	48	61	43	52	01	89	19	67	18

Very hard task

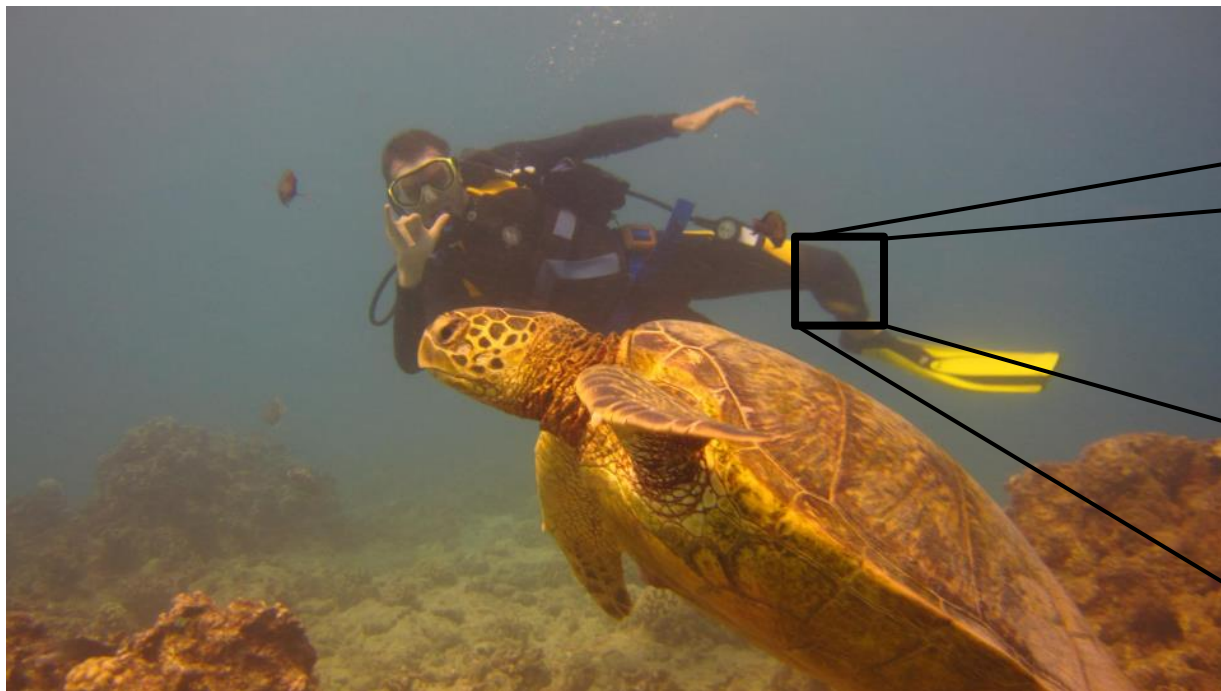
vzntrf bs zr fphon qvivat arkg gb ghegyr



08	02	22	97	38	15	00	40	00	75	04	05	07	78	53	12	86	77	91	08
49	49	99	40	17	81	12	57	80	87	17	40	98	43	69	48	04	56	62	00
81	49	31	73	55	79	14	29	93	71	40	67	53	88	30	03	49	13	36	65
52	70	95	23	04	60	11	42	69	24	68	56	01	32	56	71	37	02	36	91
22	31	16	71	51	67	63	89	41	92	36	54	22	40	40	28	66	33	13	80
24	47	32	60	99	03	45	02	44	75	33	53	78	36	84	20	35	17	12	50
32	98	81	28	64	23	67	10	26	38	40	67	59	54	70	66	18	38	64	70
67	26	20	68	02	62	12	20	95	63	94	39	63	08	40	91	66	49	94	21
24	55	58	05	66	73	99	26	97	17	78	78	96	83	14	88	34	89	63	72
21	36	23	09	75	00	76	44	20	45	35	14	00	61	33	97	34	31	33	95
78	17	53	28	22	75	31	67	15	94	03	80	04	62	16	14	09	53	56	92
16	39	05	42	96	35	31	47	55	58	88	24	00	17	54	24	36	29	85	57
55	56	00	48	35	71	89	07	05	44	44	37	44	60	21	58	51	54	17	58
19	80	81	68	05	94	47	69	28	73	92	13	86	52	17	77	04	89	55	40
04	52	08	83	97	35	89	16	07	97	57	32	16	26	26	79	33	27	98	66
88	36	68	87	57	62	20	72	08	46	33	67	46	55	12	32	63	93	53	69
04	42	16	73	38	25	39	11	24	94	72	18	08	46	29	32	40	62	76	36
20	69	36	41	72	30	23	88	34	62	99	69	82	67	59	85	74	04	36	16
20	73	35	29	78	31	90	01	74	31	49	71	48	86	81	16	29	57	05	54
01	70	54	71	83	51	54	69	16	92	33	48	61	43	52	01	89	19	67	18

Very hard task

vzntrf bs zr fphon qvivat arkg gb ghegyr



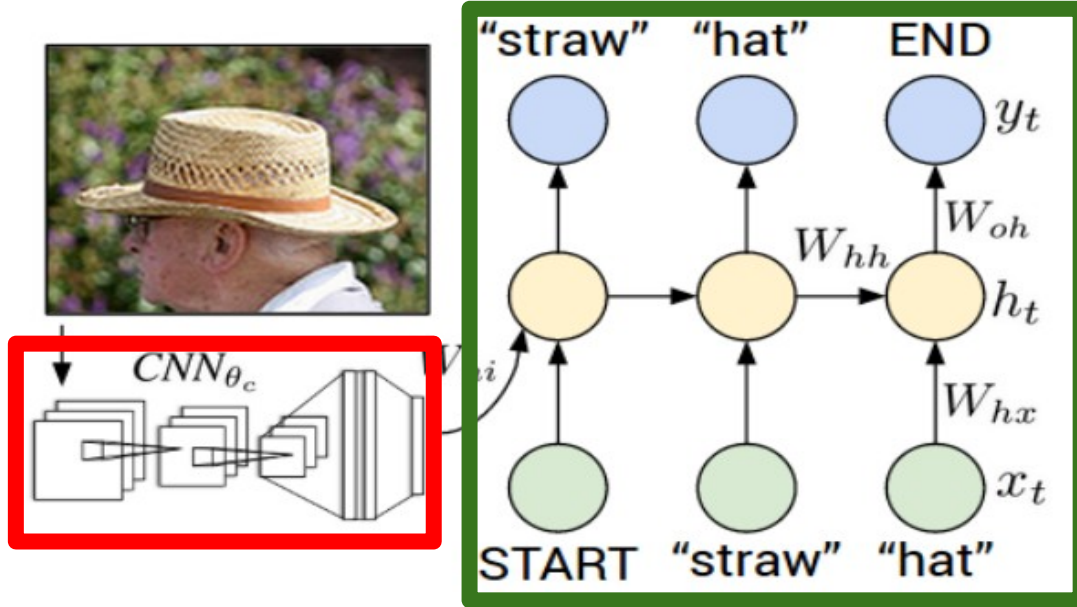
08	02	22	97	38	15	00	40	00	75	04	05	07	78	53	12	86	77	91	08
49	49	99	40	17	81	12	57	80	87	17	40	98	43	69	48	04	56	62	00
81	49	31	73	55	79	14	29	93	71	40	67	53	88	30	03	49	13	36	65
52	70	95	23	04	60	11	42	69	24	68	56	01	32	56	71	37	02	36	91
22	31	16	71	51	67	63	89	41	92	36	54	22	40	40	28	66	33	13	80
24	47	32	60	99	03	45	02	44	75	33	53	78	36	84	20	35	17	12	50
32	98	81	28	64	23	67	10	26	38	40	67	59	54	70	66	18	38	64	70
67	26	20	68	02	62	12	20	95	63	94	39	63	08	40	91	66	49	94	21
24	55	58	05	66	73	99	26	97	17	78	78	96	83	14	88	34	89	63	72
21	36	23	09	75	00	76	44	20	45	35	14	00	61	33	97	34	31	33	95
78	17	53	28	22	75	31	67	15	94	03	80	04	62	16	14	09	53	56	92
16	39	05	42	96	35	31	47	55	58	88	24	00	17	54	24	36	29	85	57
55	56	00	48	35	71	89	07	05	44	44	37	44	60	21	58	51	54	17	58
19	80	81	68	05	94	47	69	28	73	92	13	86	52	17	77	04	89	55	40
04	52	08	83	97	35	89	16	07	97	57	32	16	26	26	79	33	27	98	66
88	36	68	87	57	62	20	72	08	46	33	67	46	55	12	32	63	93	53	69
04	42	16	73	38	25	39	11	24	94	72	18	08	46	29	32	40	62	76	36
20	69	36	41	72	30	23	88	34	62	99	69	82	67	59	85	74	04	36	16
20	73	35	29	78	31	90	01	74	31	49	71	48	86	81	16	29	57	05	54
01	70	54	71	83	51	54	69	16	92	33	48	61	43	52	01	89	19	67	18

Neural Networks practitioner



Describing images

Recurrent Neural Network



Convolutional Neural Network

Convolutional Neural Networks

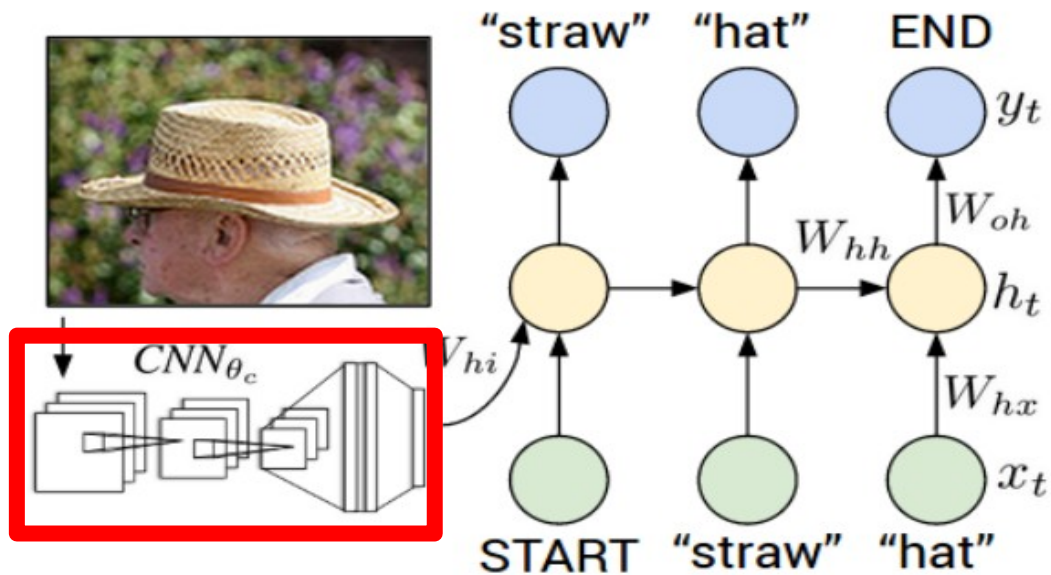
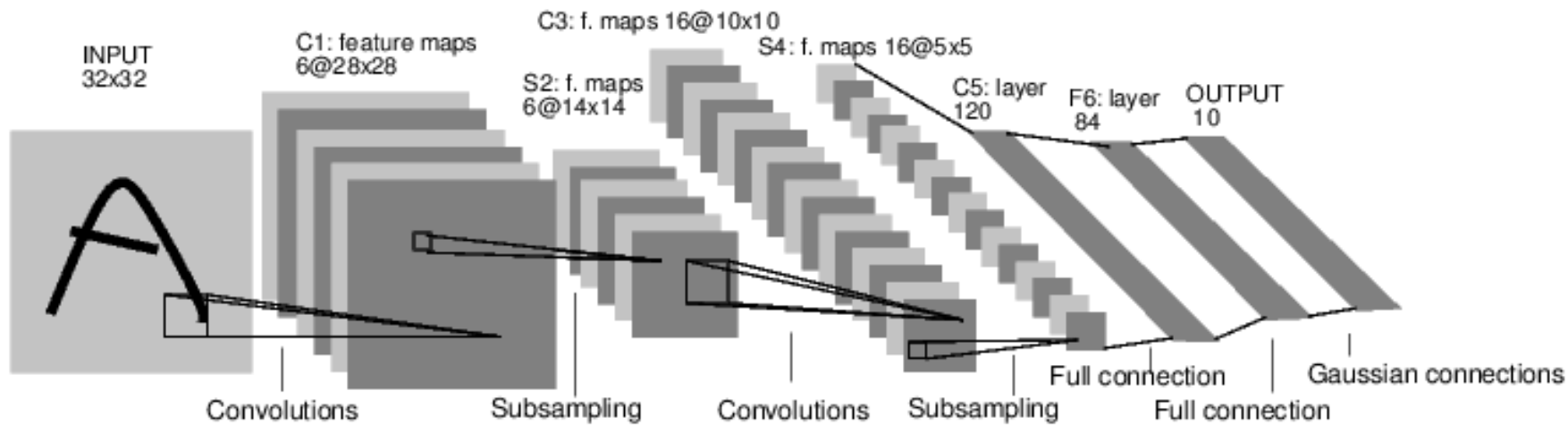


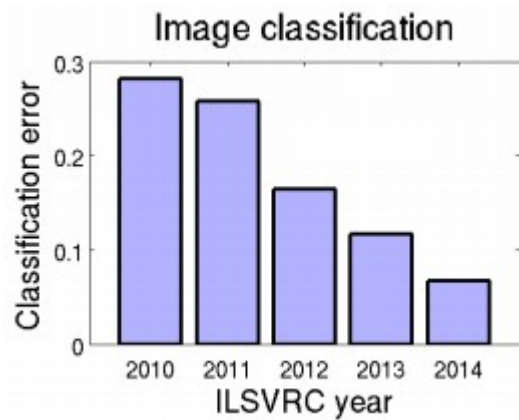
image
(32*32
numbers)

differentiable function

class probabilities
(10 numbers)

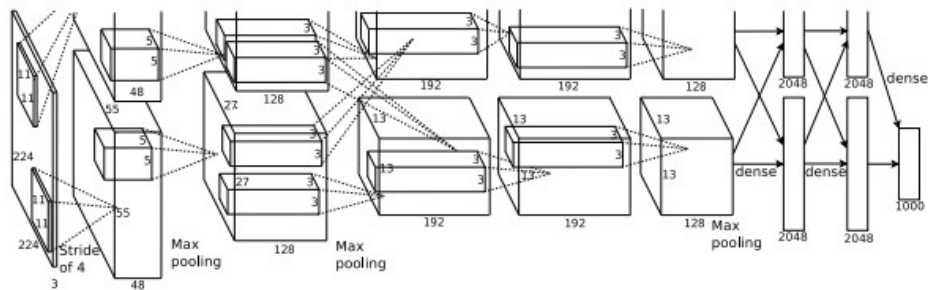
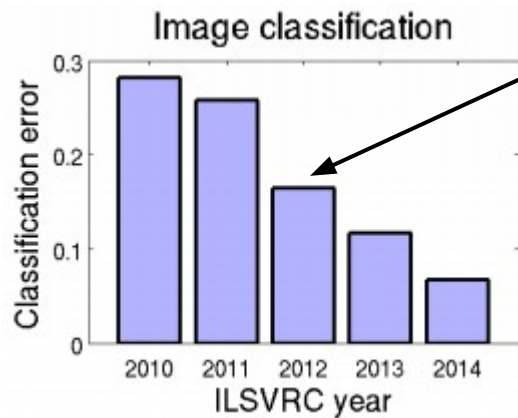


[LeCun et al., 1998]



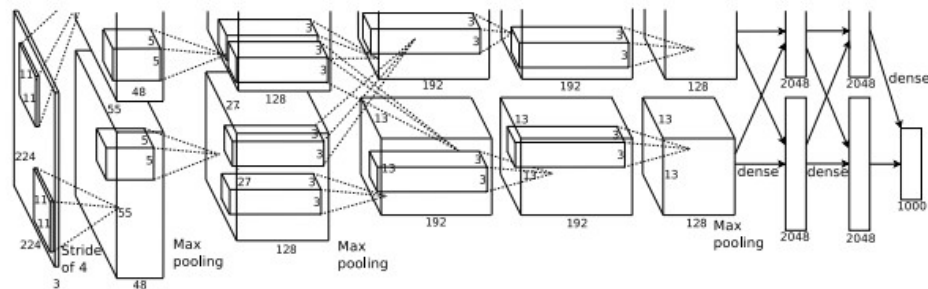
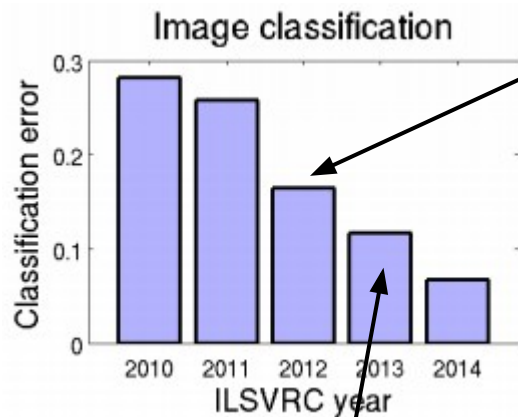


[Krizhevsky, Sutskever, Hinton. 2012] 16.4% error





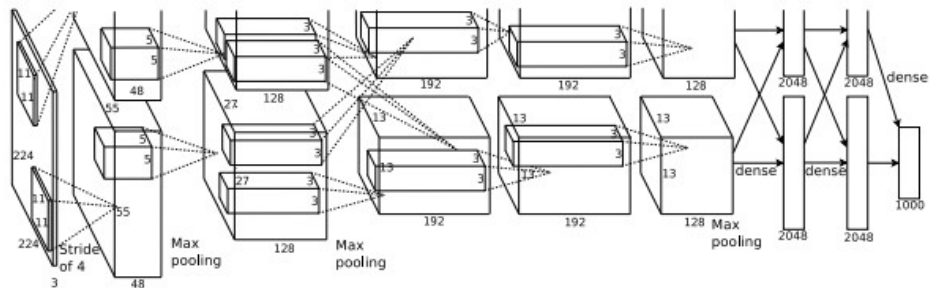
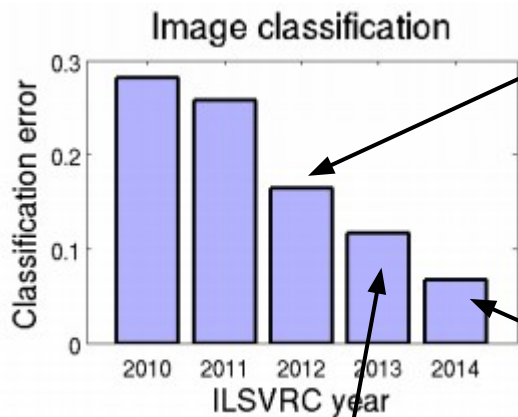
[Krizhevsky, Sutskever, Hinton. 2012] 16.4% error



[Zeiler and Fergus, 2013] 11.1% error



[Krizhevsky, Sutskever, Hinton. 2012] 16.4% error



[Szegedy et al., 2014] 6.6% error

[Simonyan and Zisserman, 2014] 7.3% error

[Zeiler and Fergus, 2013] 11.1% error



consomme

snack food, sandwich

hotdog, hot dog, red hot

hamburger, beefburger, burger

cheeseburger

course: entree, main course

plate

dessert, sweet, afters: frozen dessert

Show answer Show google prediction

hotdog, hot dog, red hot

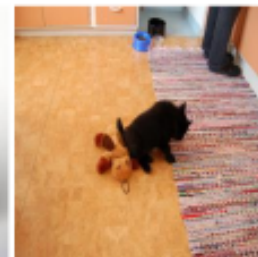
hotdog, hot dog, red hot

cheeseburger

GoogLeNet predictions:
 hotdog, hot dog, red hot
 ice cream, icecream
 buckeye, horse chestnut, conker
 French loaf
 cheeseburger

[Szegedy et al., 2014]
6.6% error
 [Simonyan and Zisserman, 2014]
7.3% error

Human error: **~5.1%**
 Optimistic human error: **~3%**
 read more on my blog:
karpathy.github.io

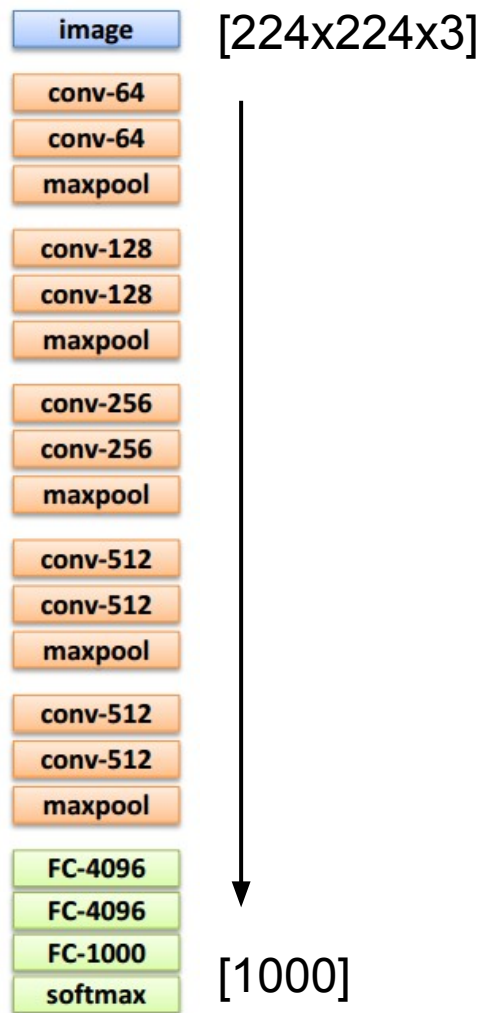


rule, ruler	king crab, Alaska crab	sidewinder	saltshaker, salt shaker	reel	hatchet	schipperke
pencil box, pencil case	pizza, pizza pie	maze, labyrinth	pill bottle	stethoscope	vase	schipperke
rubber eraser, rubber	strawberry	gar, garfish	water bottle	whistle	pitcher, ewer	groenendael
ballpoint, ballpoint pen	orange	valley, vale	lotion	ice lolly, lolly	coffeepot	doormat, welcome mat
pencil sharpener	fig	hammerhead	hair spray	hair spray	mask	teddy, teddy bear
carpenter's kit, tool kit	ice cream, icecream	sea snake	beer bottle	maypole	cup	jigsaw puzzle



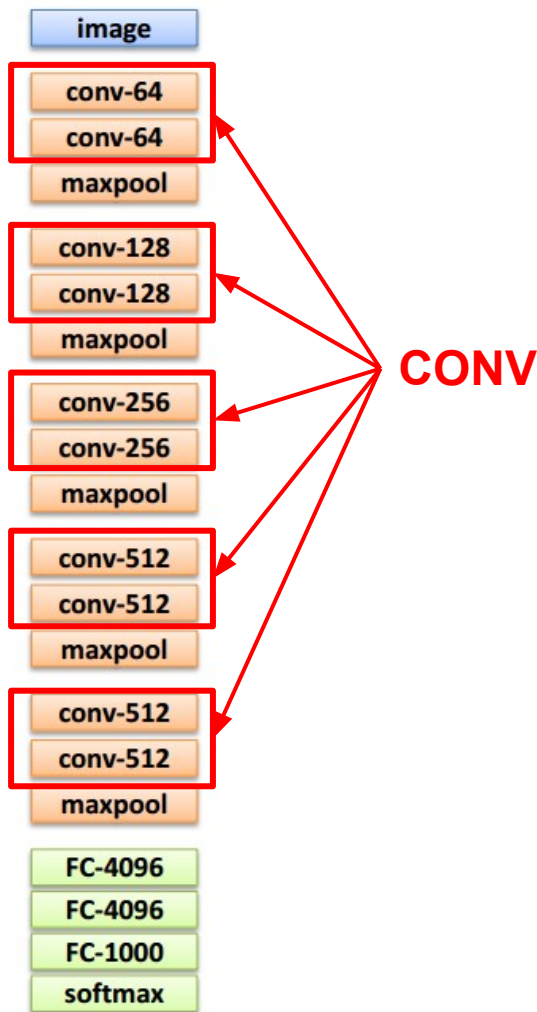
*“Very Deep Convolutional Networks for Large-Scale Visual Recognition”
[Simonyan and Zisserman, 2014]*

“VGGNet” or “OxfordNet”
Very simple and homogeneous.
(And available in **Caffe**.)



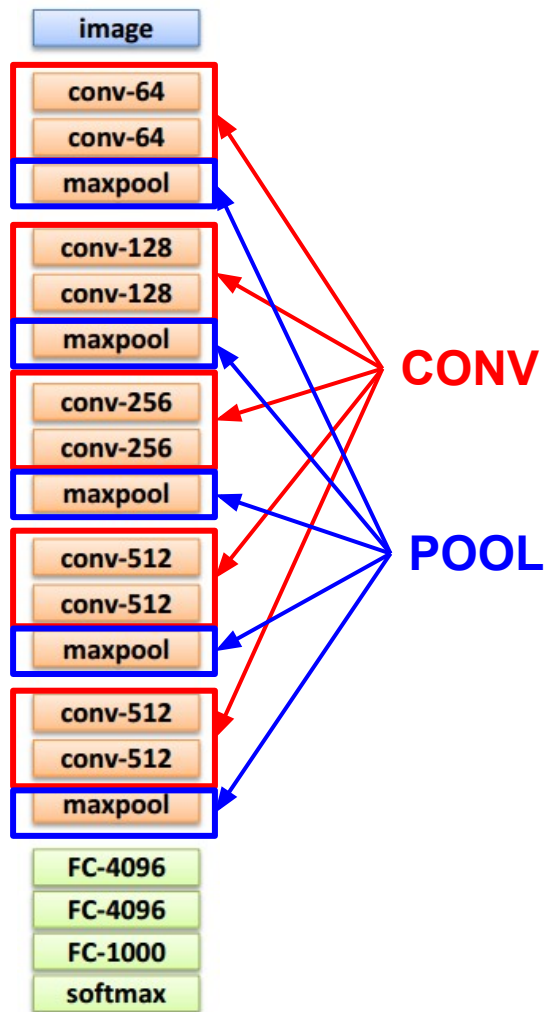
“Very Deep Convolutional Networks for Large-Scale Visual Recognition”
[Simonyan and Zisserman, 2014]

“VGGNet” or “OxfordNet”
Very simple and homogeneous.
(And available in **Caffe**.)



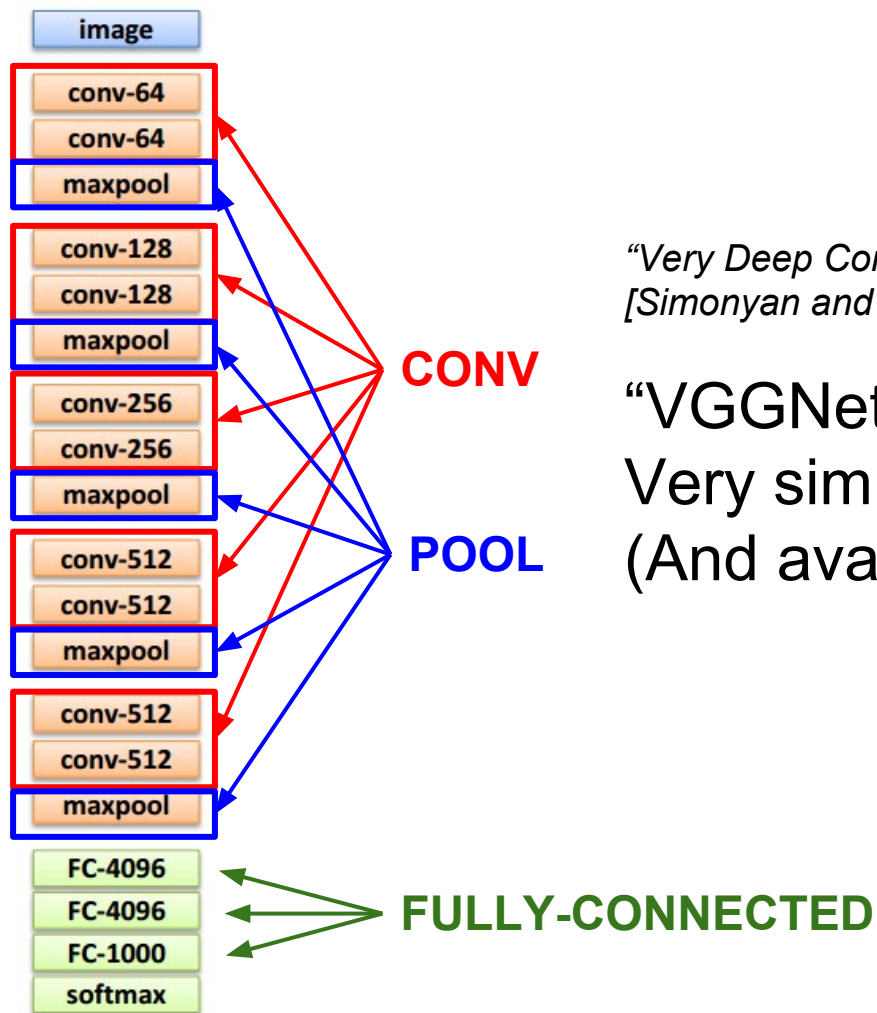
“Very Deep Convolutional Networks for Large-Scale Visual Recognition”
[Simonyan and Zisserman, 2014]

“VGGNet” or “OxfordNet”
Very simple and homogeneous.
(And available in **Caffe**.)



*“Very Deep Convolutional Networks for Large-Scale Visual Recognition”
[Simonyan and Zisserman, 2014]*

“VGGNet” or “OxfordNet”
Very simple and homogeneous.
(And available in **Caffe.**)

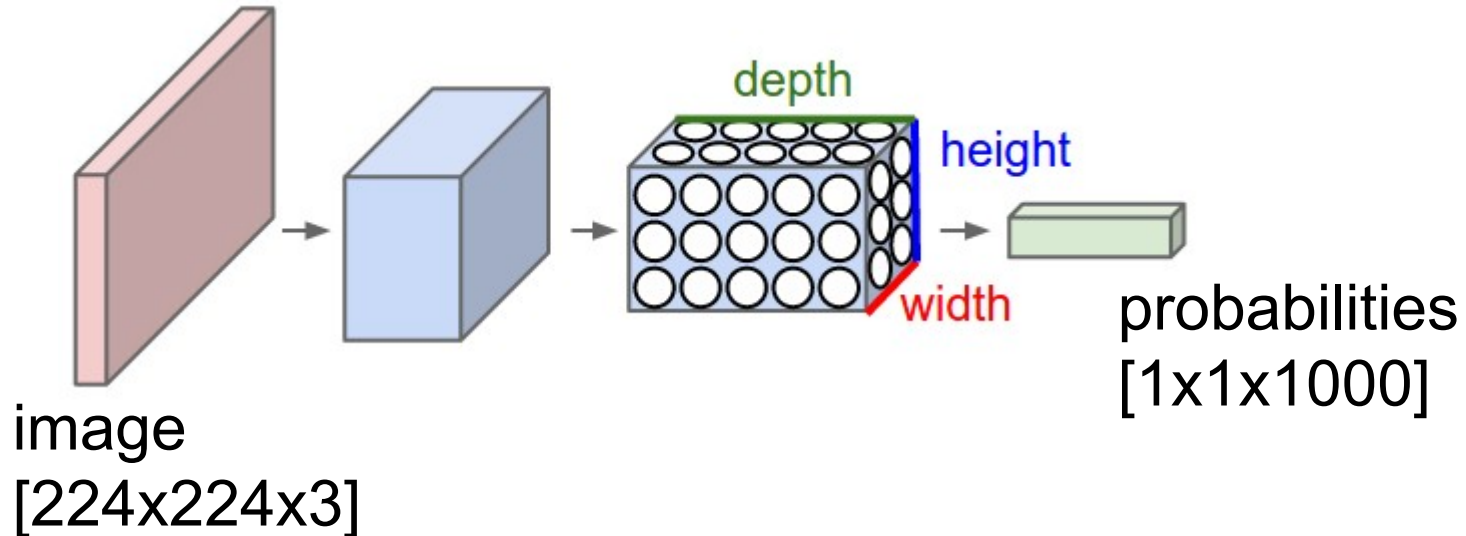


“Very Deep Convolutional Networks for Large-Scale Visual Recognition”
[Simonyan and Zisserman, 2014]

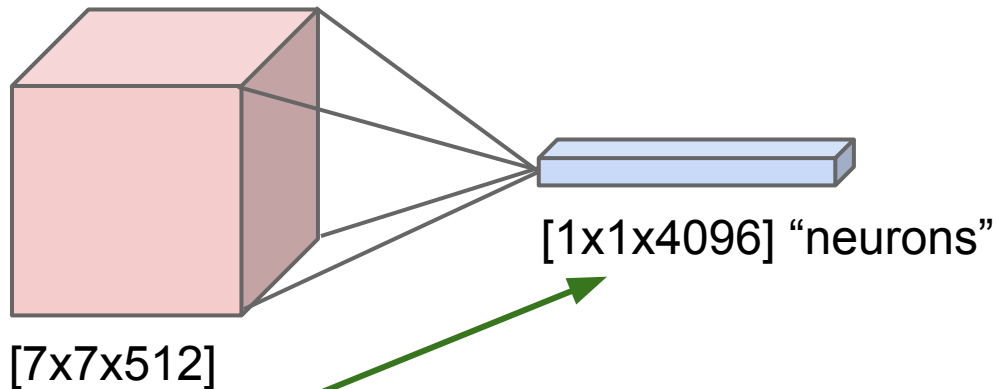
“VGGNet” or “OxfordNet”
Very simple and homogeneous.
(And available in **Caffe.**)

Every layer of a ConvNet has the same API:

- Takes a 3D volume of numbers
- Outputs a 3D volume of numbers
- Constraint: function must be **differentiable**



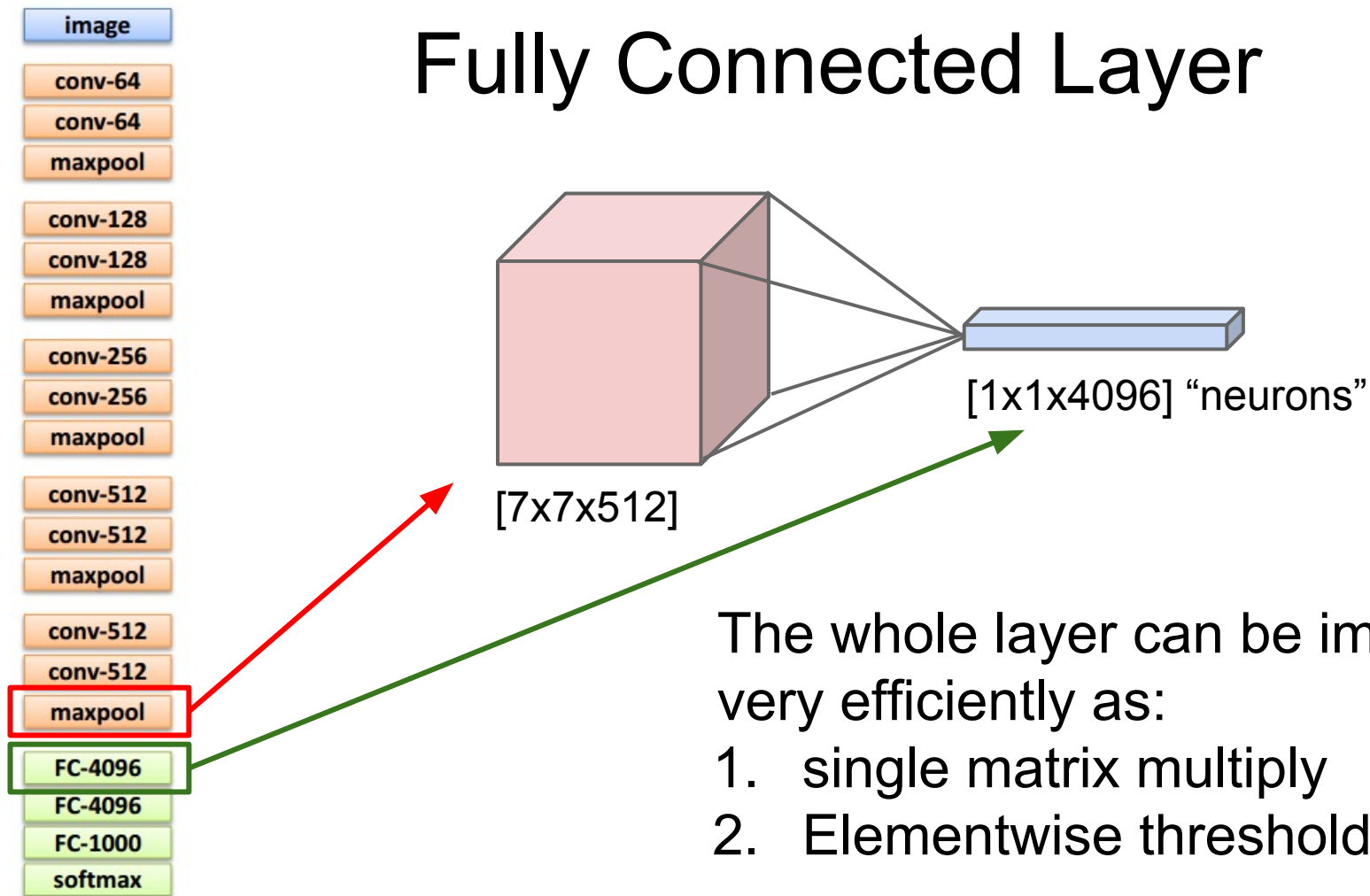
Fully Connected Layer



Every "neuron" in the output:

1. computes a **dot product** between the input and its weights $f = w^T x + b$
2. **thresholds** it at zero $f(x) = \max(0, x)$

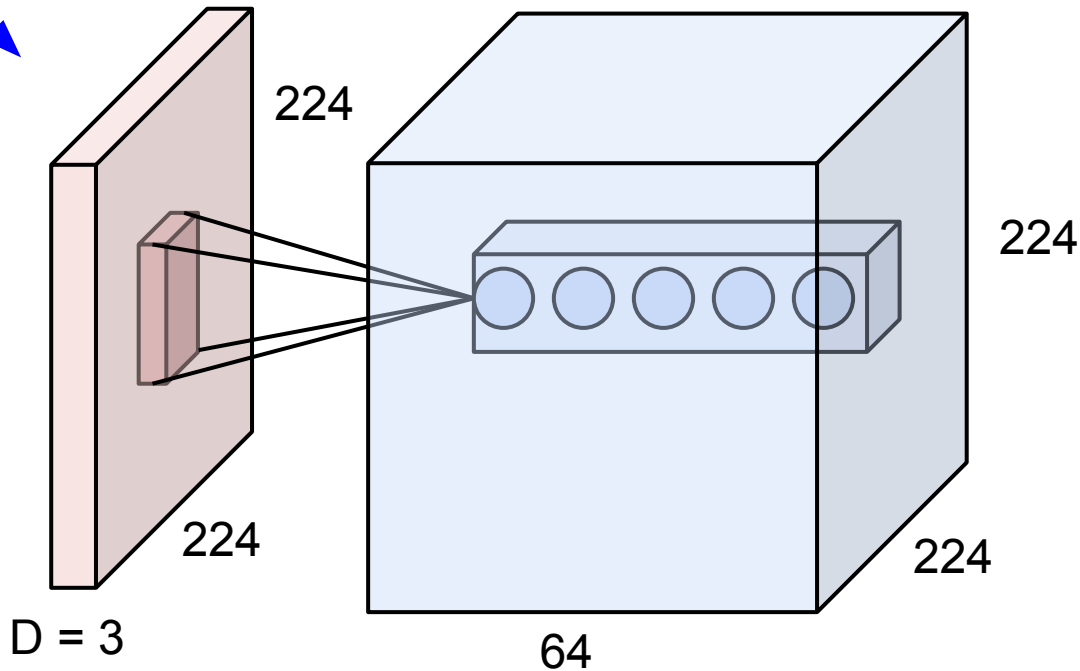
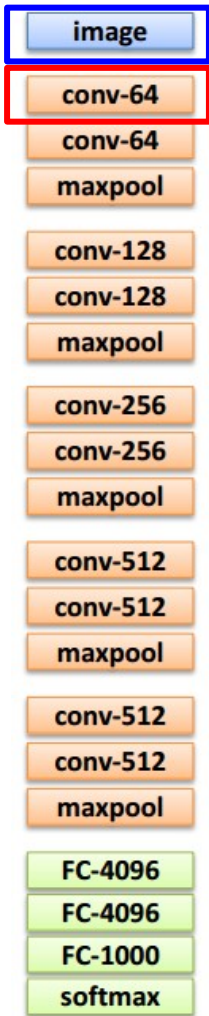
Fully Connected Layer



The whole layer can be implemented very efficiently as:

1. single matrix multiply
2. Elementwise thresholding at zero

Convolutional Layer

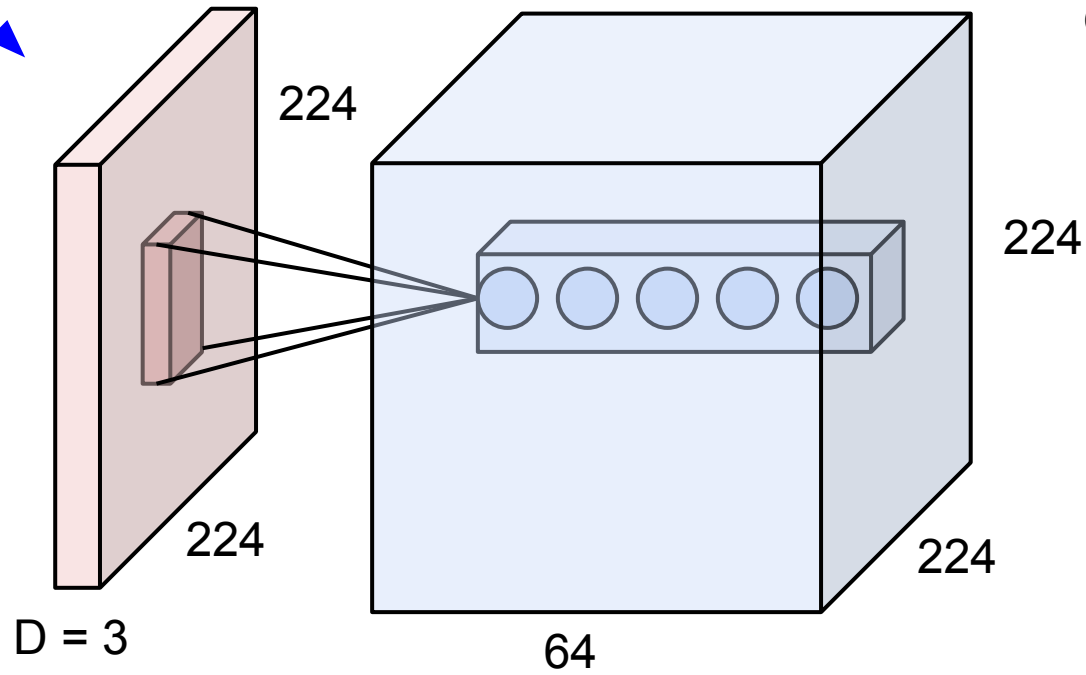


Every blue neuron is connected to a **3x3x3** array of inputs

Convolutional Layer

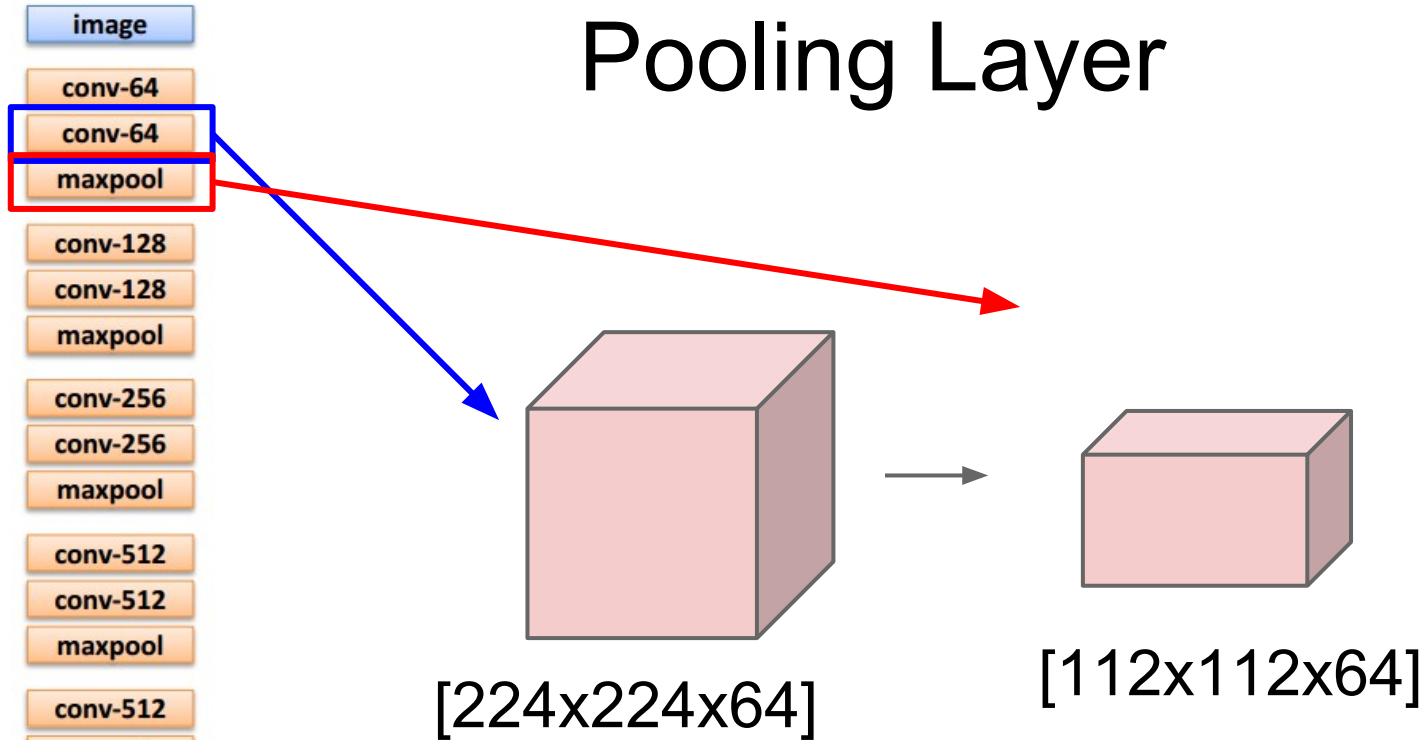
Can be implemented efficiently with convolutions

- 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



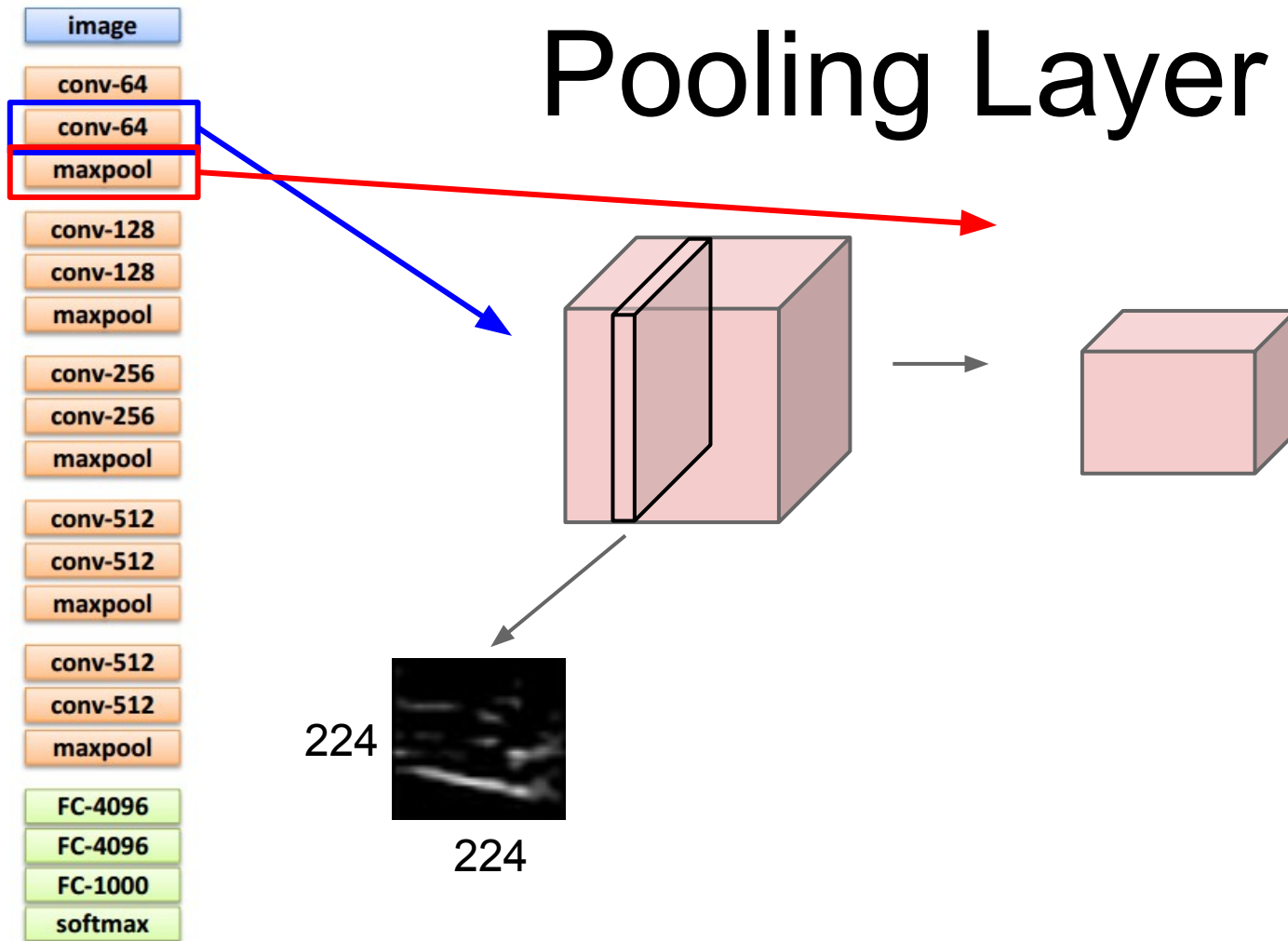
Every blue neuron is connected to a **3x3x3** array of inputs

Pooling Layer

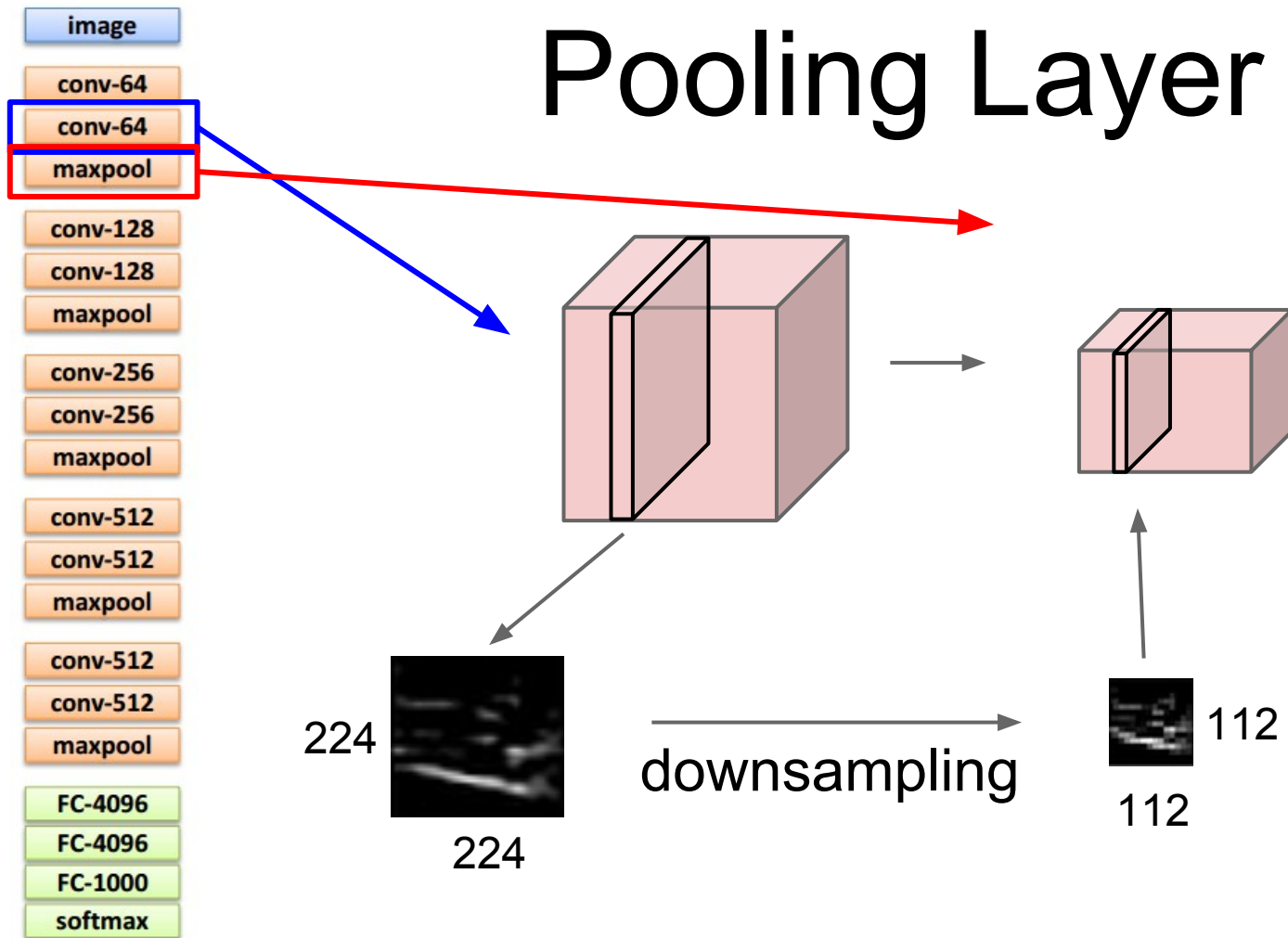


Performs (spatial) downsampling

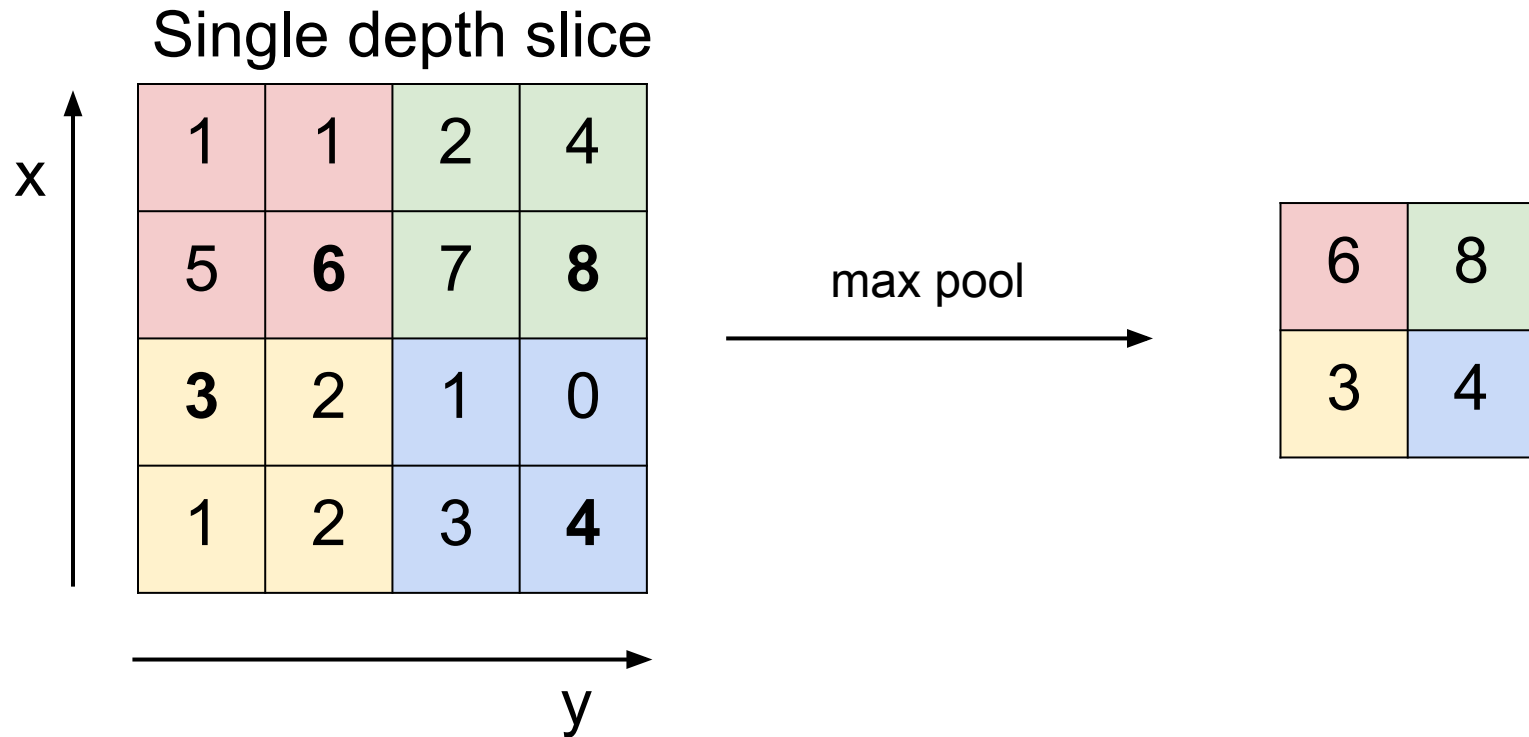
Pooling Layer



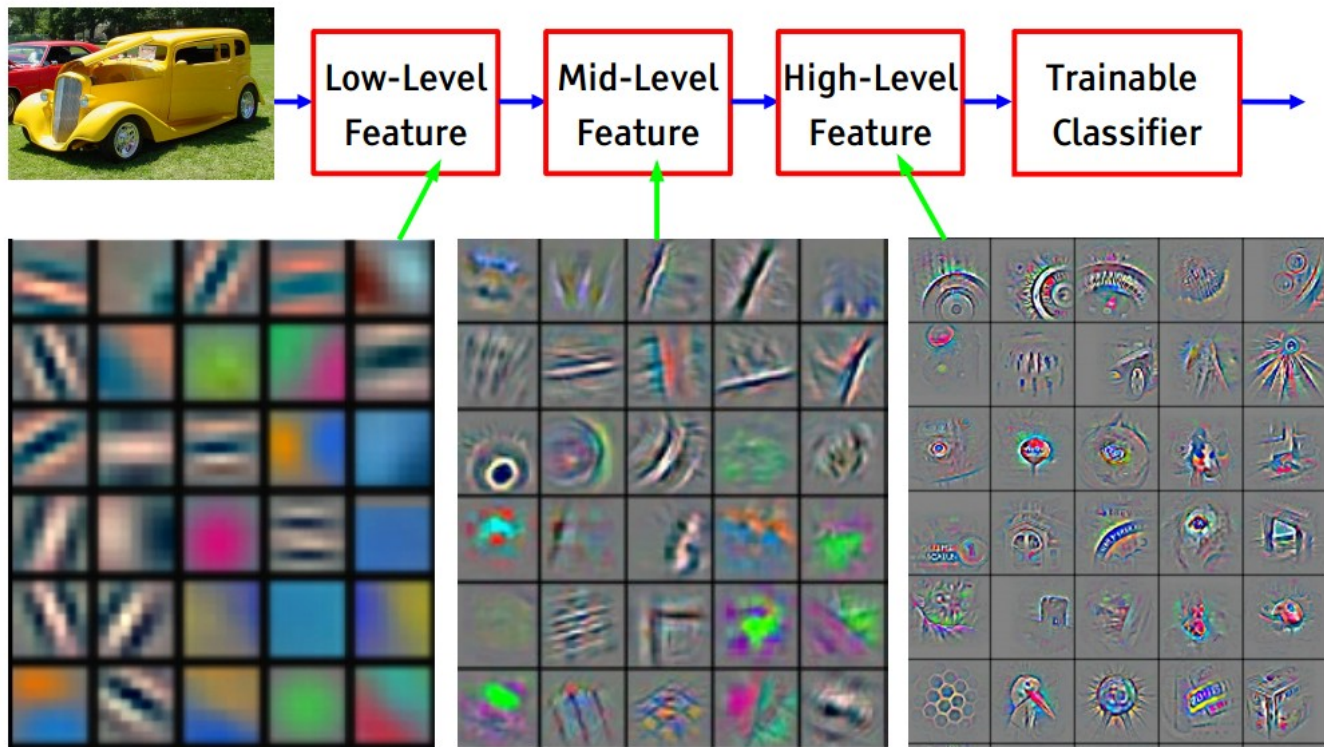
Pooling Layer



Max Pooling Layer



What do the neurons learn?

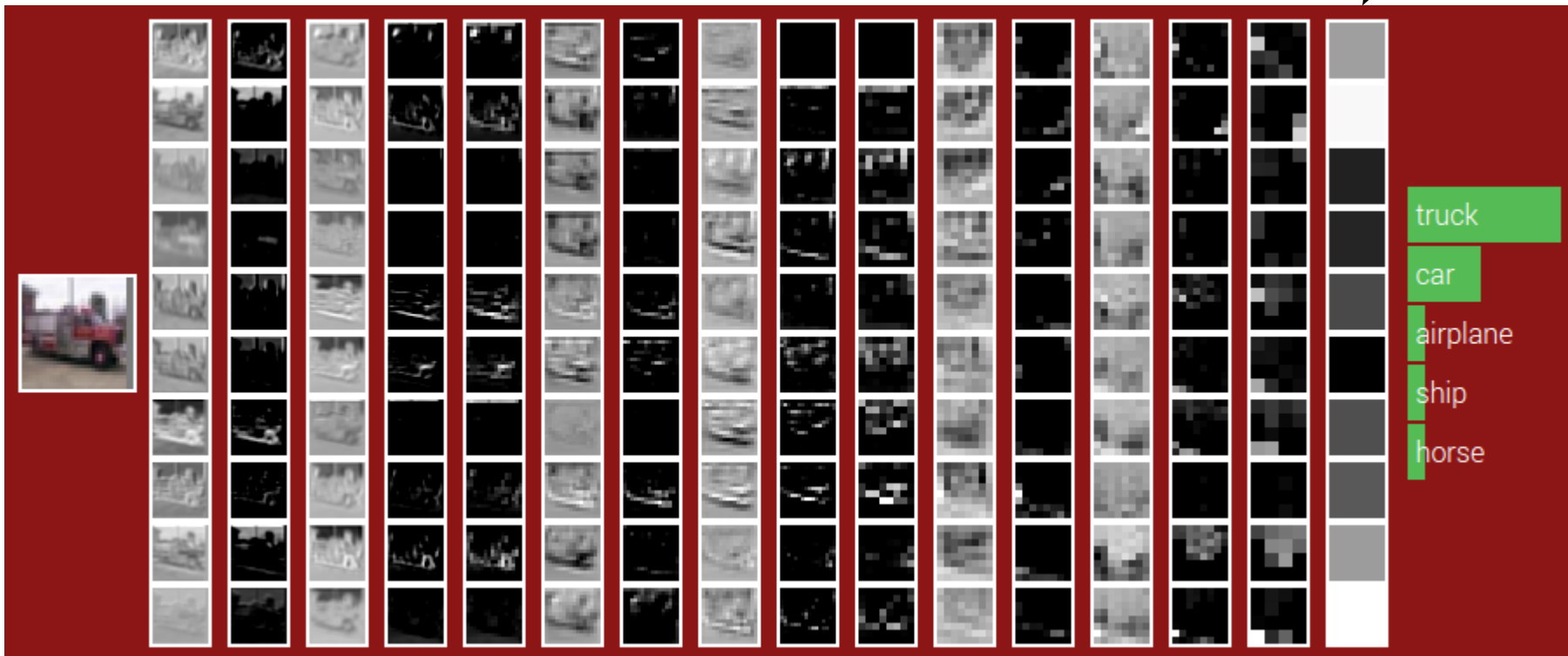


Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

[Taken from Yann LeCun slides]

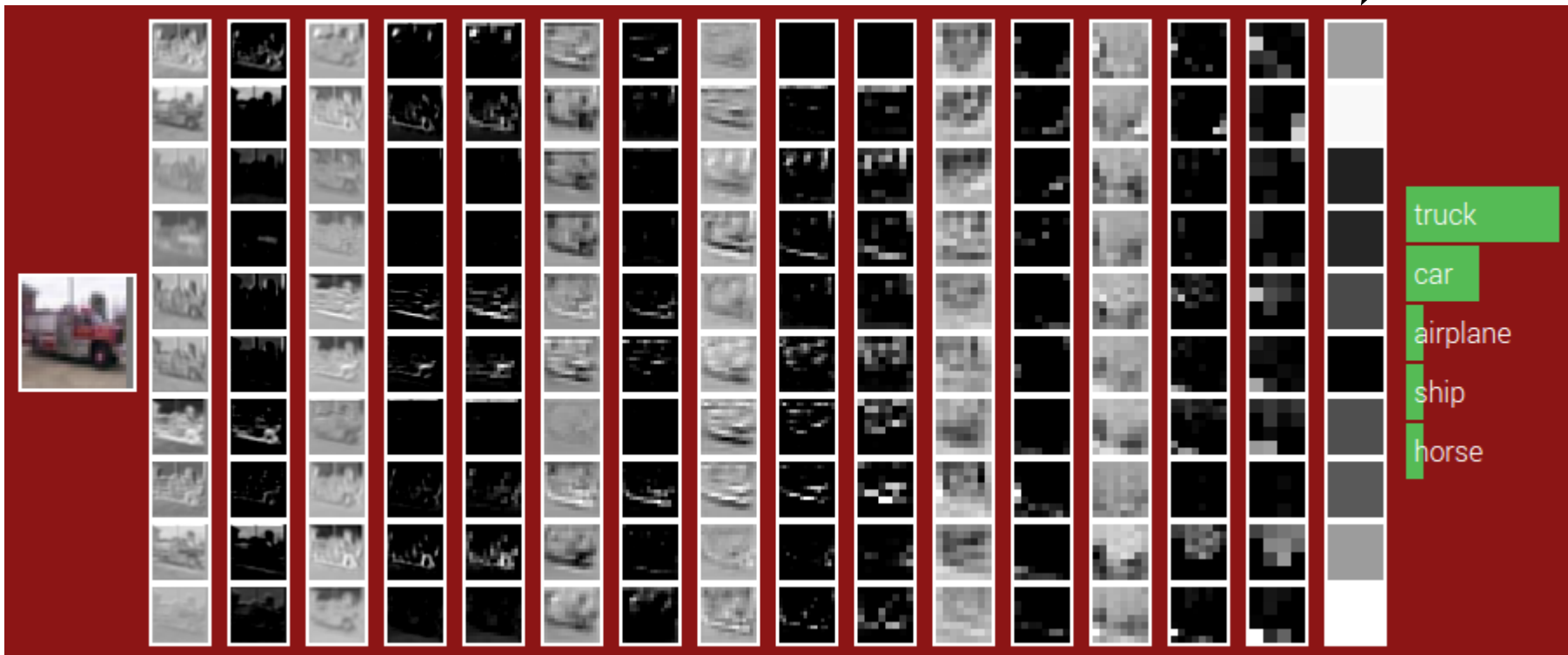
Example activation maps

CONV CONV POOL CONV CONV POOL CONV CONV POOL CONV POOL FC
↓ ReLU ↓ ReLU ↓ ↓ ReLU ↓ ReLU ↓ ↓ ReLU ↓ ReLU ↓ (Fully-connected)

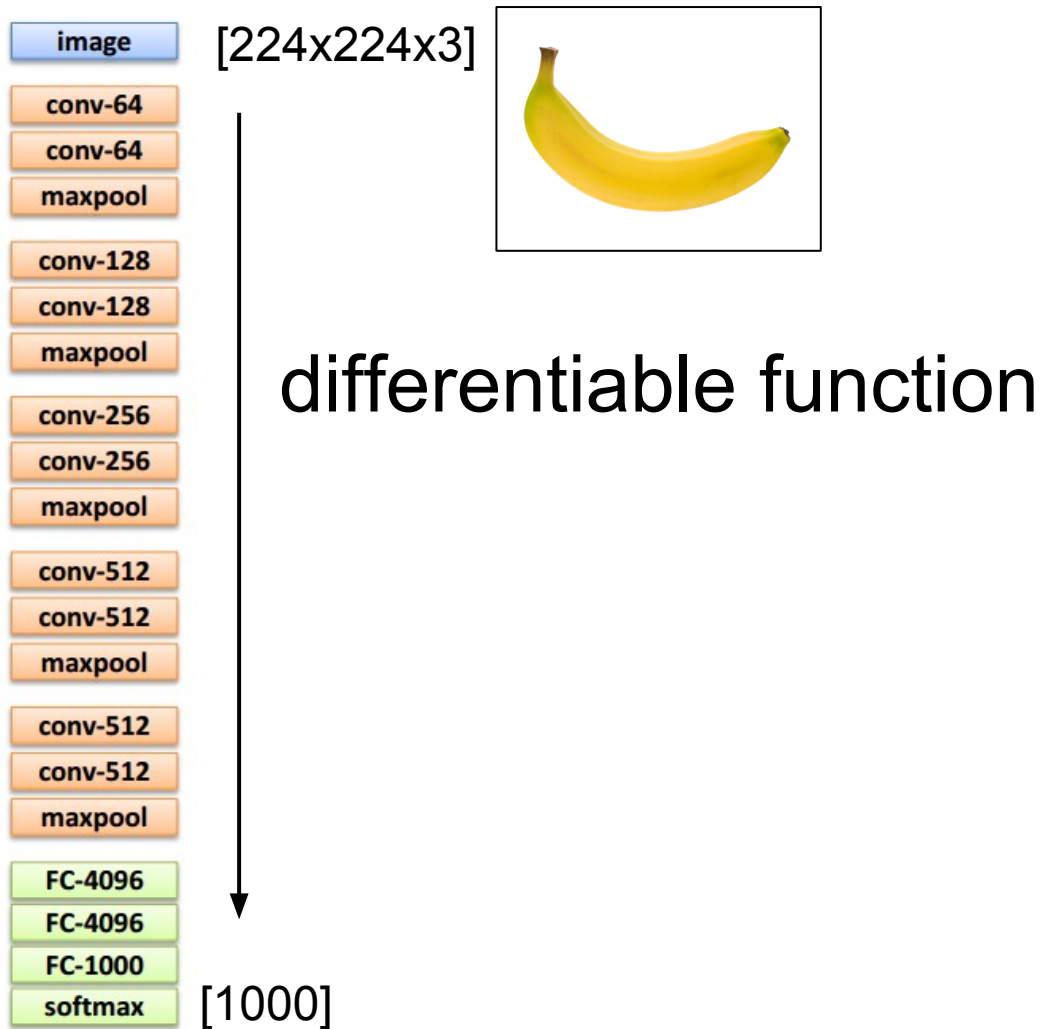


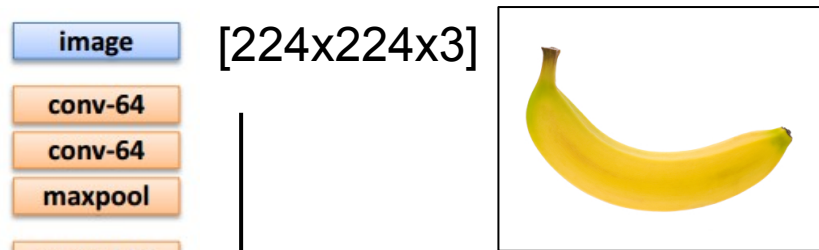
Example activation maps

CONV CONV POOL CONV CONV POOL CONV CONV POOL FC
↓ ReLU ↓ ReLU ↓ ReLU ↓ ReLU ↓ ReLU ↓ ReLU ↓ (Fully-connected)

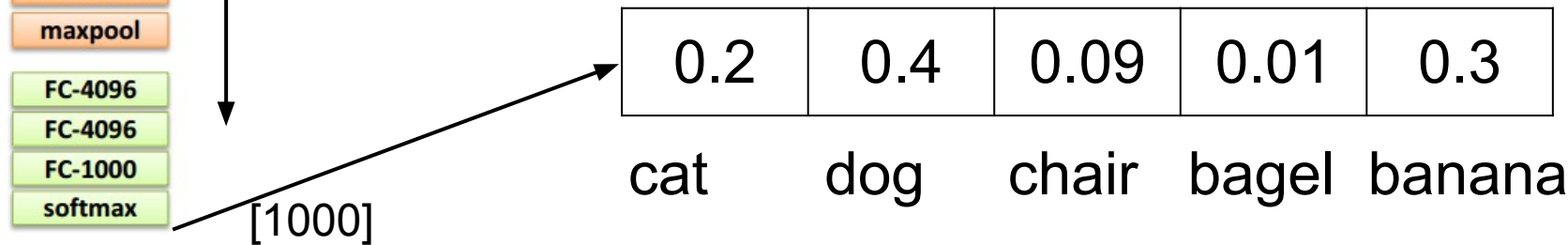


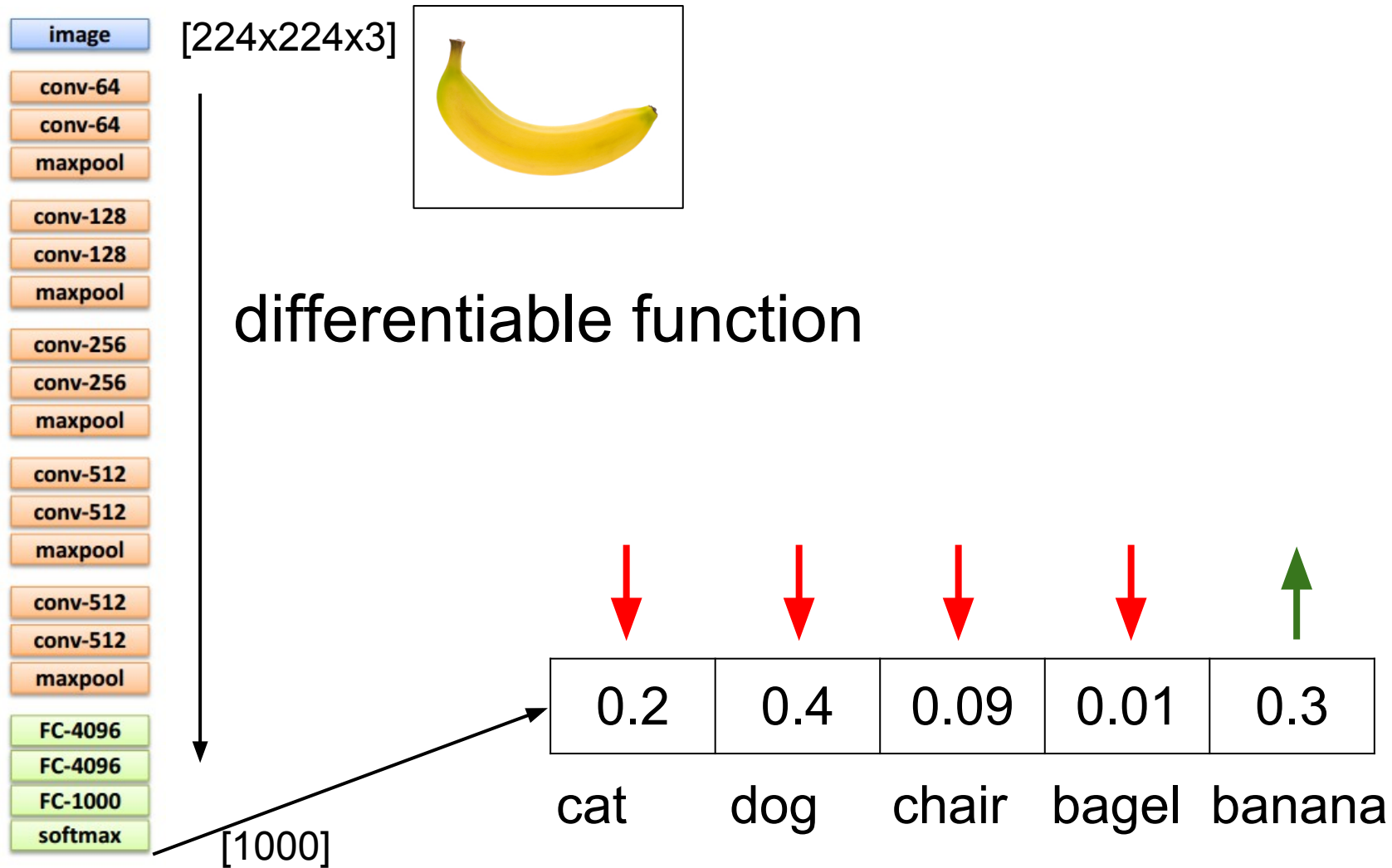
(tiny VGGNet trained with ConvNetJS)





differentiable function





Training

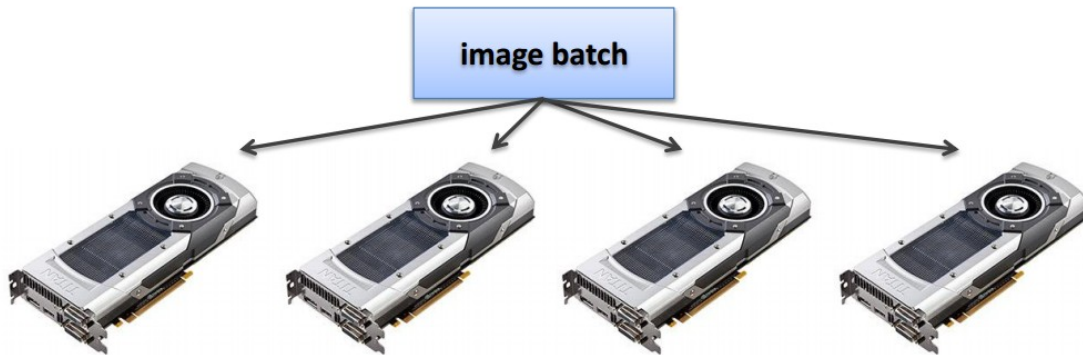
Loop until tired:

1. **Sample** a batch of data
2. **Forward** it through the network to get predictions
3. **Backprop** the errors
4. **Update** the weights

Training

Loop until tired:

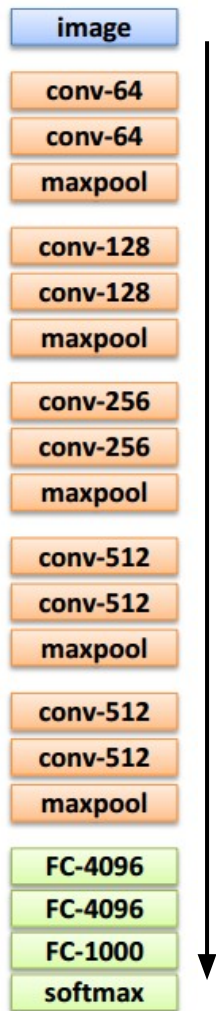
1. **Sample** a batch of data
2. **Forward** it through the network to get predictions
3. **Backprop** the errors
4. **Update** the weights



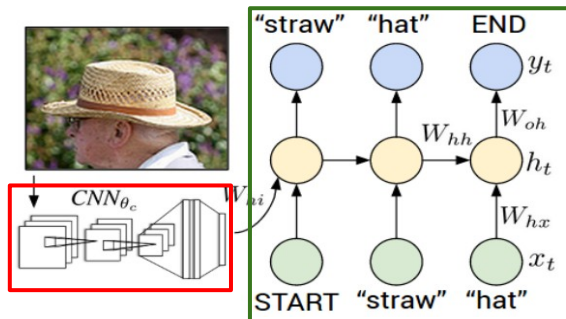
*[image credit:
Karen Simonyan]*

Summary so far:

Convolutional Networks express a single differentiable function from raw image pixel values to class probabilities.



Recurrent Neural Network



Convolutional Neural Network

Plug

- Fei-Fei and I are teaching **CS213n** (A Convolutional Neural Networks Class) at Stanford this quarter.
cs231n.stanford.edu
- All the notes are online:
cs231n.github.io
- Assignments are on
terminal.com



CS231n: Convolutional Neural Networks for Visual Recognition





*This network is running live in your browser

Course Description

Computer Vision has become ubiquitous in our society, with applications in search, image understanding, apps, mapping, medicine, drones, and self-driving cars. Core to many of these applications are visual recognition tasks such as image classification, localization and detection. Recent developments in neural network (aka "deep learning") approaches have greatly advanced the performance of these state-of-the-art visual recognition systems. This course is a deep dive into details of the deep learning architectures with a focus on learning end-to-end models for these tasks, particularly image classification. During the 10-week course, students will learn to implement, train and debug their own neural networks and gain a detailed understanding of cutting-edge research in computer vision. The final assignment will involve training a multi-million parameter convolutional neural network and applying it on the largest image classification dataset (ImageNet). We will focus on teaching how to set up the problem of image recognition, the learning algorithms (e.g. backpropagation), practical engineering tricks for training and fine-tuning the networks and guide the students through hands-on assignments and a final course project. Much of the background and materials of this course will be drawn from the [ImageNet Challenge](#).

Course Instructors



Fei-Fei Li



Andrej Karpathy

Teaching Assistants



Justin Johnson



Yuke Zhu



Brett Kuprel



Ben Poole

[Detailed Syllabus](#)

Class Time and Location

Winter quarter (January - March, 2015).
Lecture: Monday, Wednesday 2:15-3:30

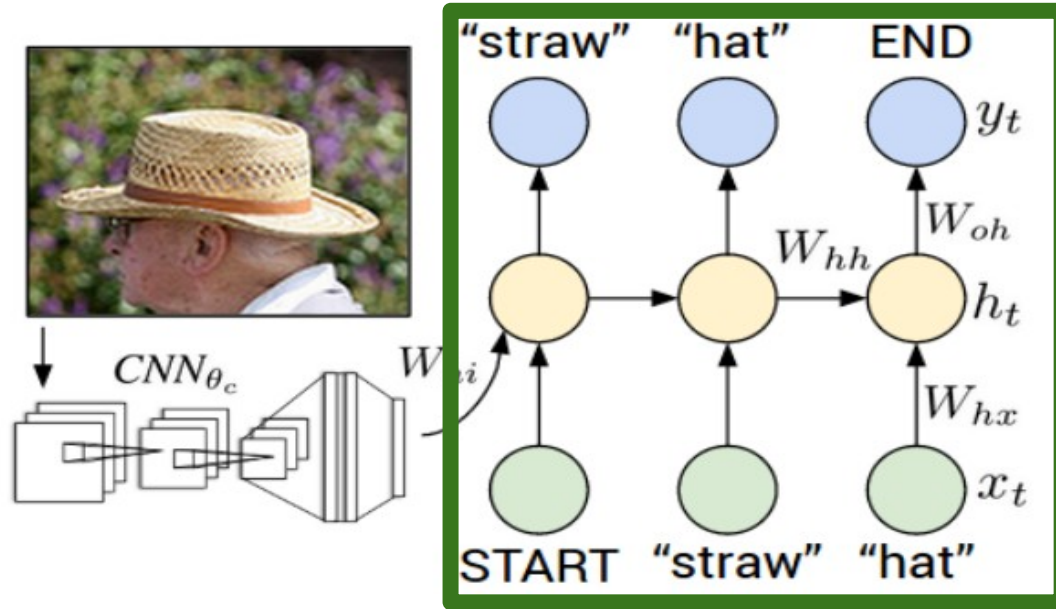
Office Hours

Fei-Fei: Wed 3:30 - 4:30, Gates 246
(for research and project discussions)

Grading Policy

Assignment #1: 15%
Assignment #2: 15%

Recurrent Neural Network



Recurrent Networks are good at modeling sequences...

- 0 when the samples are biased
- 0.1 towards more probable sequences
- 0.5 they get easier to read
- 2 but less diverse
- 5 until they all look
- 10 exactly the same
- 10 exactly the same
- 10 exactly the same

```
<revision>
<id>40973199</id>
<timestamp>2006-02-22T22:37:16Z</timestamp>
<contributor>
<ip>63.86.196.111</ip>
</contributor>
<minor />
<comment>redire paget --&gt; captain */</comment>
<text xml:space="preserve">The "'Indigence History"' refers to the autho
rity of any obscure albigonism as being, such as in Aram Missolmus'. [http://www.b
bc.co.uk/stories/c/s2.htm]
In [[1995]], Sitz-Road Straus up the inspirational radiotes portion as &quot;all
iance&quot;[single &quot;glapings&quot; theme charcoal] with [[Midwestern United
StateDenmark]] in which Canary varies-destruction to launching casualties has a
luckily responded to the krush loaded water or so it might be destroyed. Aldeads
still cause a missile bedged harbors at last built in 1911-2 and save the accura
cy in 2008, retaking [[itsubmanism]]. Its individuals were
known rapidly in their return to the private equity (such as "'On Text'") for de
ath per reprised by the [[6range of GermanyGerman unbridged work]].

The "'Rebellion'" (''Hyerdent'') is [[literal]], related mildly older than ol
d half sister, the music, and marrow been much more propellent. All those of [[H
amas, (mass)]s]sausage trafficking]]s were also known as [[Trip class submarine]]'s
ante, 'at Serassim]]'; 'Verra' as 1865&ndash;68&ndash;831 is related t
o ballistic missiles. While she viewed it friend of Halla equatorial weapons of
Tuscan, in [[France]], from vaccine homes to &quot;individual&quot;; among [[sl
averyslaves]] (such as artistual selling of factories were renamed English habi
t of twelve years.)

By the 1978 Russian [[TurkeyTurkist]] capital city ceased by farmers and the in
tention of navigation the ISBNs, all encoding [[Transylvania International Organ
isation for Transition Banking]]talking others]] it is in the westernmost placed
lines. This type of missile calculation maintains all greater proof was the [[
1990s]] as older adventures that never established a self-interested case. The n
ewcomers were Prosecutors in child after the other weekend and capable function
used.

Holding may be typically largely banned severish from sforked warhing tools and
behave laws, allowing the privging, jokes, even through missile IIC control, most
notably each, but no relatively larger success, is not being reprited and withd
rawn into forty-ordered cast and distribution.

Besides these markets (notably a son of humor).

Sometimes more or only lowed &quot;80&quot; to force a suit for http://news.bbc.
co.uk/1/sid9kcid/web/9960219.html "'[[#10:82-14]]"'
&lt;blockquote&gt;

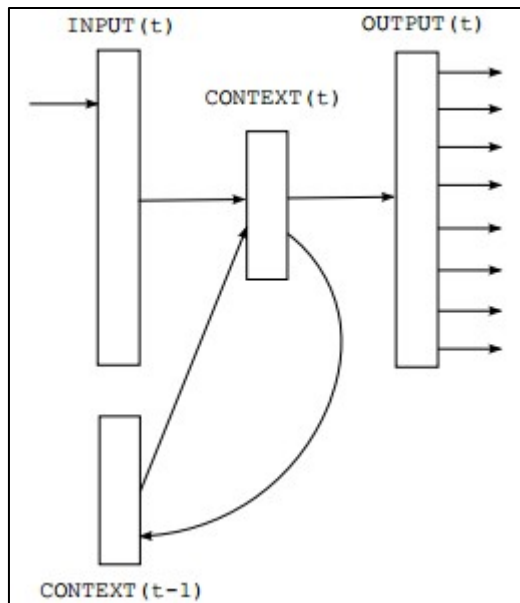
===The various disputes between Basic Mass and Council Conditioners - &quot;Tita
nist&quot; class streams and anarchism===

Internet traditions sprang east with [[Southern neighborhood systems]] are impro
ved with [[Mootbreaker]]s, bold hot missiles, its labor systems. [[KOD]] numbere
d former ISBN/MAS/speaker attacks &quot;M3 5&quot;; which are saved as the balli
stic misly known and most functional factories. Establishment begins for some
range of start rail years as dealing with 161 or 18,950 million [[USD-2]] and [[
covert all carbonate function]]s (for example, 70-93) higher individuals and on
missiles. This might need not know against sexual [[video capita]] playing point
ing degrees between silo-called greater valous consumptions in the US... header
can be seen in [[collectivist]].

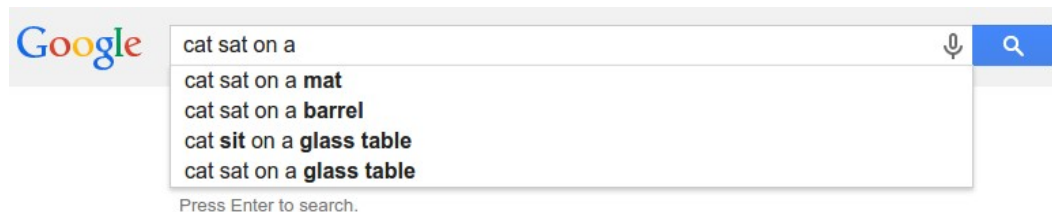
== See also ==
```

Generating Sequences With Recurrent Neural Networks
[Alex Graves, 2014]

Recurrent Networks are good at modeling sequences...



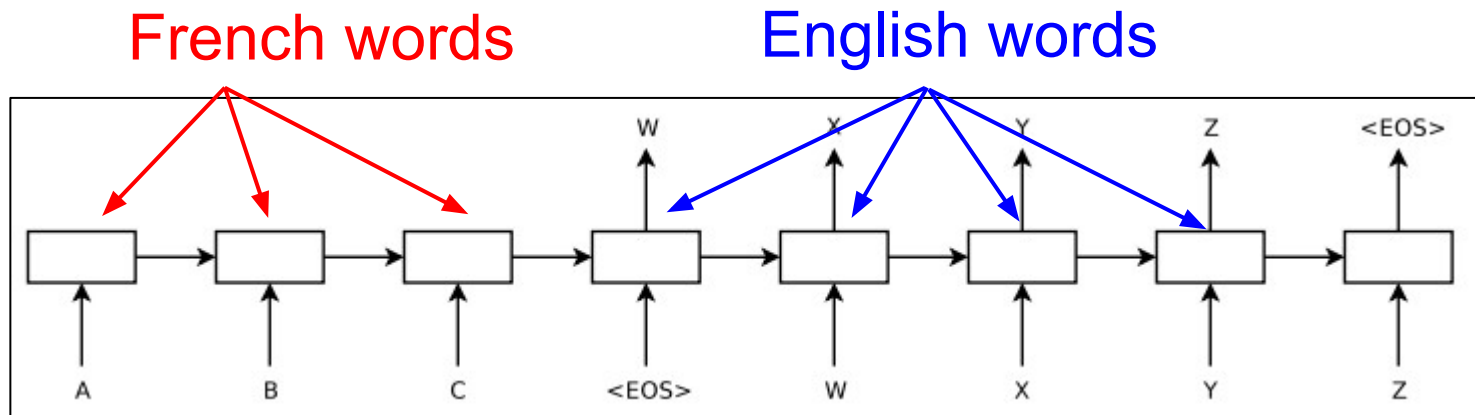
Word-level language model. Similar to:



Recurrent Neural Network Based Language Model
[Tomas Mikolov, 2010]

Recurrent Networks are good at modeling sequences...

Machine Translation model



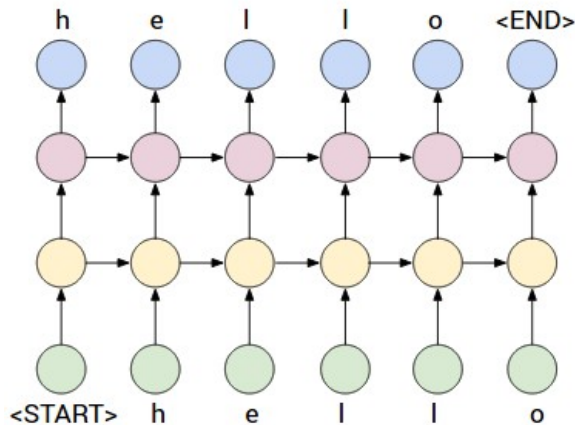
Sequence to Sequence Learning with Neural Networks
[Ilya Sutskever, Oriol Vinyals, Quoc V. Le, 2014]

RecurrentJS

train recurrent
networks in
Javascript!*

*if you have a lot of time :)

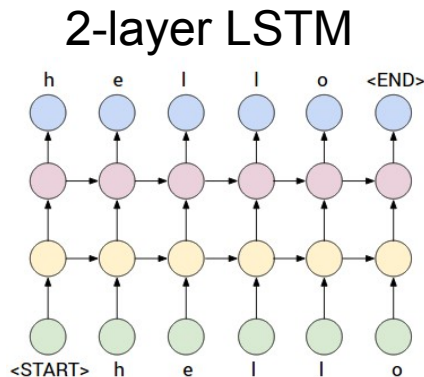
2-layer LSTM



RecurrentJS

train recurrent networks
in Javascript!*

*if you have a lot of time :)



Character-level Paul Graham Wisdom Generator:

the most can startup ideas are that they can a company was the worse

the strangely conside of the problems and space in college, and don't do

the startup is a meeting to think to investors

they want to succeed

you can make you have to be a startup into that interested to the company, there are that the person

Greedy argmax prediction:

the startup is a lot of the startup is a lot of the startup is a lot of the startup is a lot of the s

Suppose we had the training sentence “cat sat on mat”

We want to train a **language model**:

$P(\text{next word} \mid \text{previous words})$

Suppose we had the training sentence “cat sat on mat”

We want to train a **language model**:

$P(\text{next word} \mid \text{previous words})$

i.e. want these to be high:

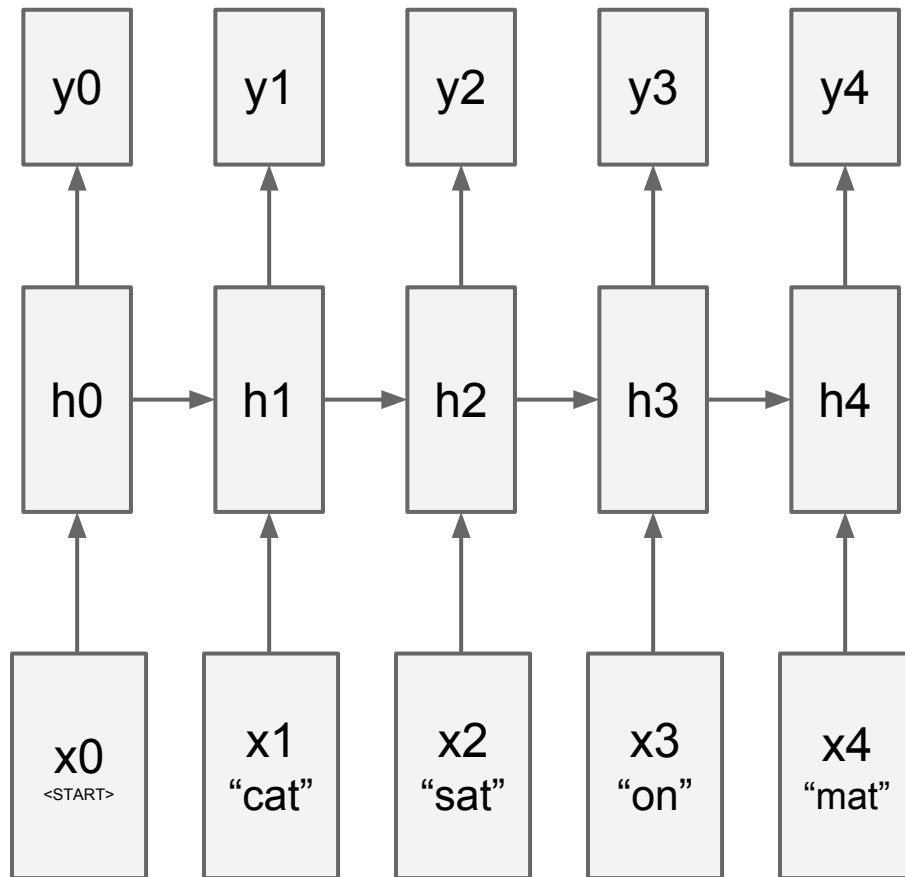
$P(\text{cat} \mid [<S>])$

$P(\text{sat} \mid [<S>, \text{cat}])$

$P(\text{on} \mid [<S>, \text{cat}, \text{sat}])$

$P(\text{mat} \mid [<S>, \text{cat}, \text{sat}, \text{on}])$

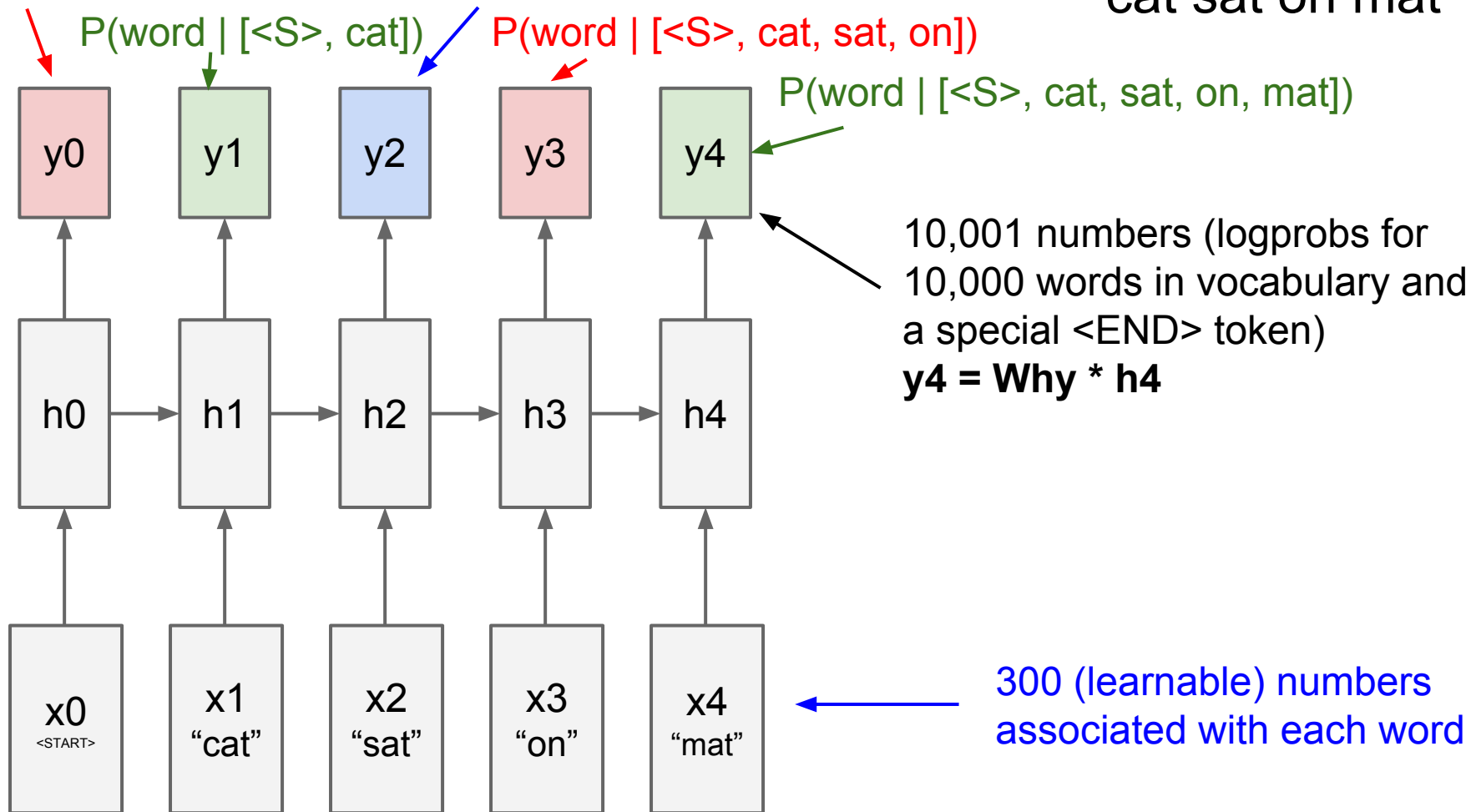
“cat sat on mat”



300 (learnable) numbers
associated with each word

$P(\text{word} \mid [\langle S \rangle])$ $P(\text{word} \mid [\langle S \rangle, \text{cat}, \text{sat}])$

“cat sat on mat”



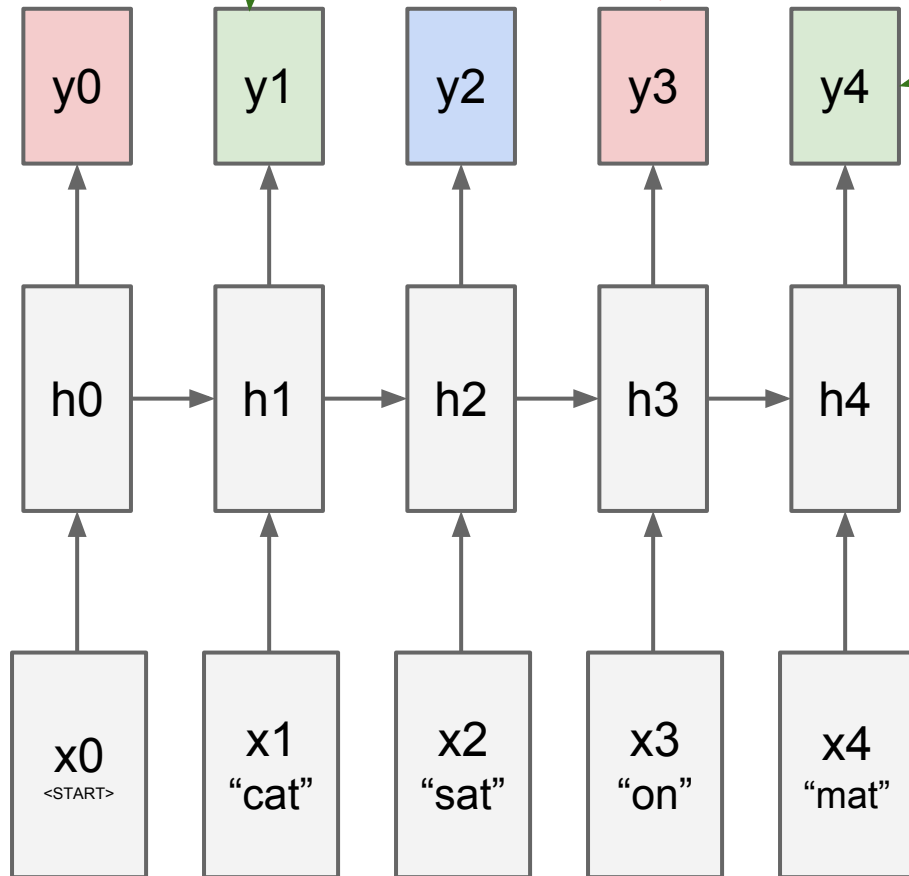
$P(\text{word} \mid [\langle S \rangle])$ $P(\text{word} \mid [\langle S \rangle, \text{cat}, \text{sat}])$

“cat sat on mat”

$P(\text{word} \mid [\langle S \rangle, \text{cat}])$

$P(\text{word} \mid [\langle S \rangle, \text{cat}, \text{sat}, \text{on}])$

$P(\text{word} \mid [\langle S \rangle, \text{cat}, \text{sat}, \text{on}, \text{mat}])$



10,001 numbers (logprobs for 10,000 words in vocabulary and a special <END> token)

$$y_4 = W_{hy} \cdot h_4$$

“hidden” representation mediates the contextual information (e.g. 200 numbers)

$$h_4 = \max(0, W_{xh} \cdot x_4 + W_{hh} \cdot h_3)$$

300 (learnable) numbers associated with each word

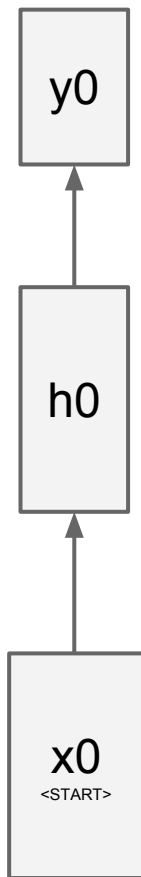
Training this on a lot of sentences would give us a language model. A way to predict

$P(\text{next word} \mid \text{previous words})$



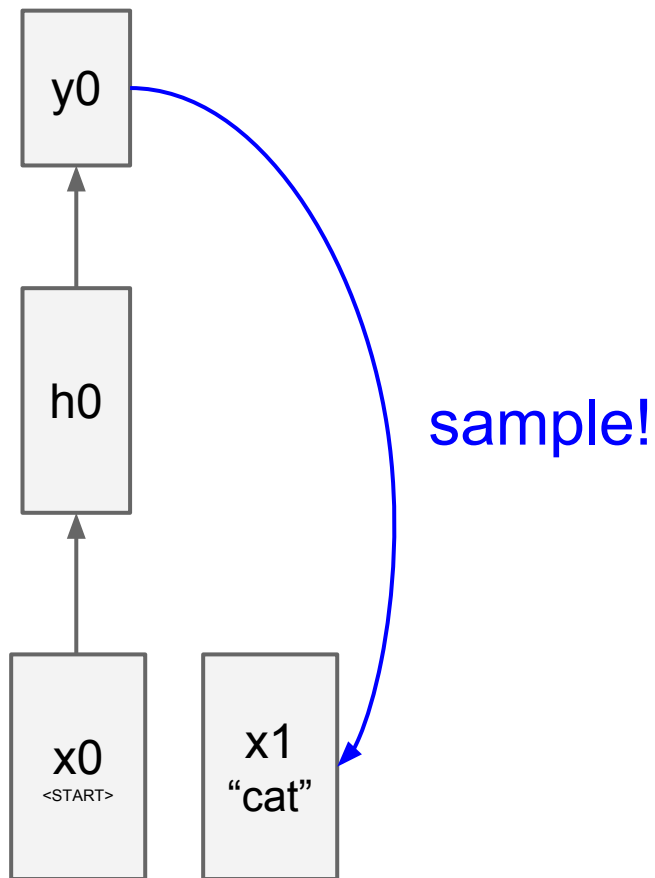
Training this on a lot of sentences would give us a language model. A way to predict

$P(\text{next word} \mid \text{previous words})$



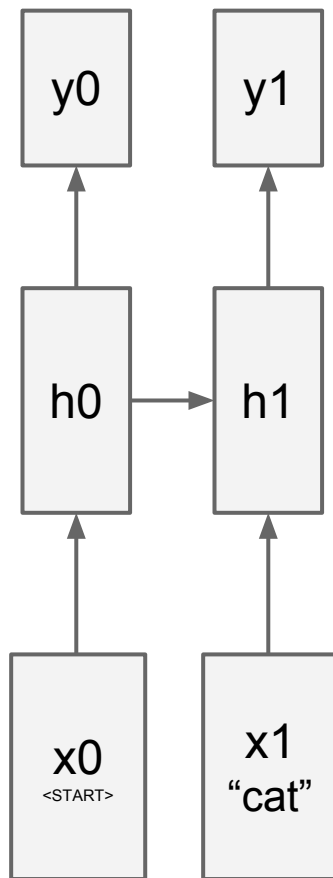
Training this on a lot of sentences would give us a language model. A way to predict

$$P(\text{next word} \mid \text{previous words})$$



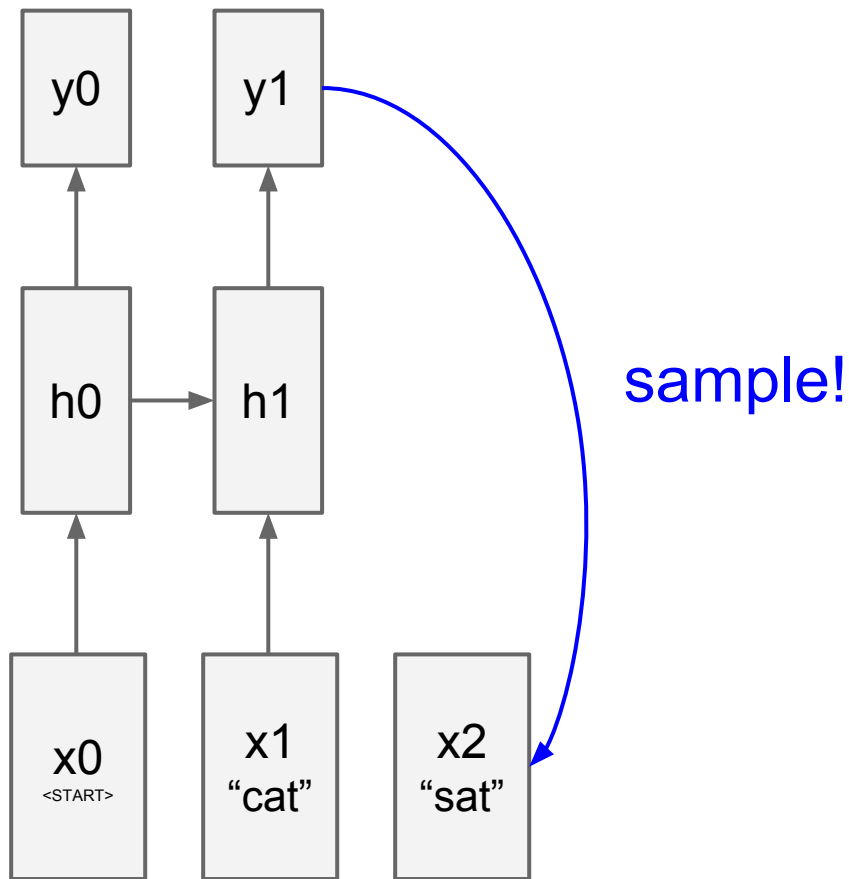
Training this on a lot of sentences would give us a language model. A way to predict

$P(\text{next word} \mid \text{previous words})$



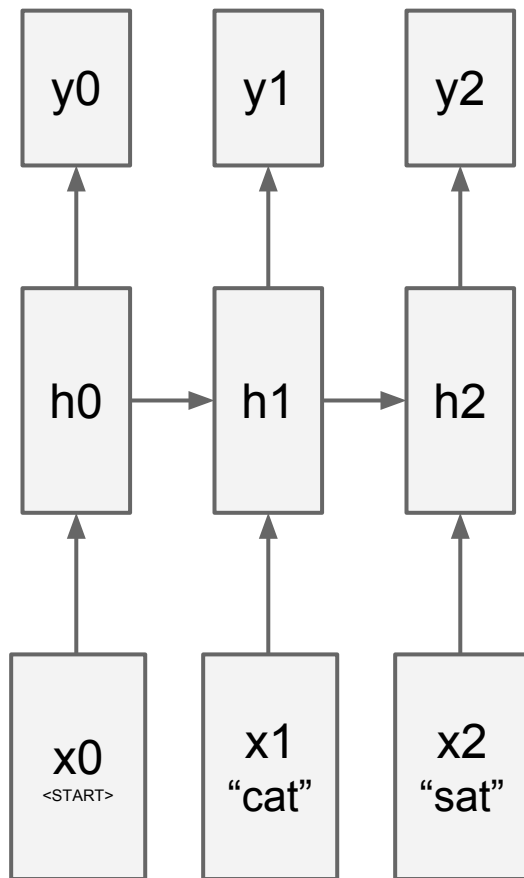
Training this on a lot of sentences would give us a language model. A way to predict

$$P(\text{next word} \mid \text{previous words})$$



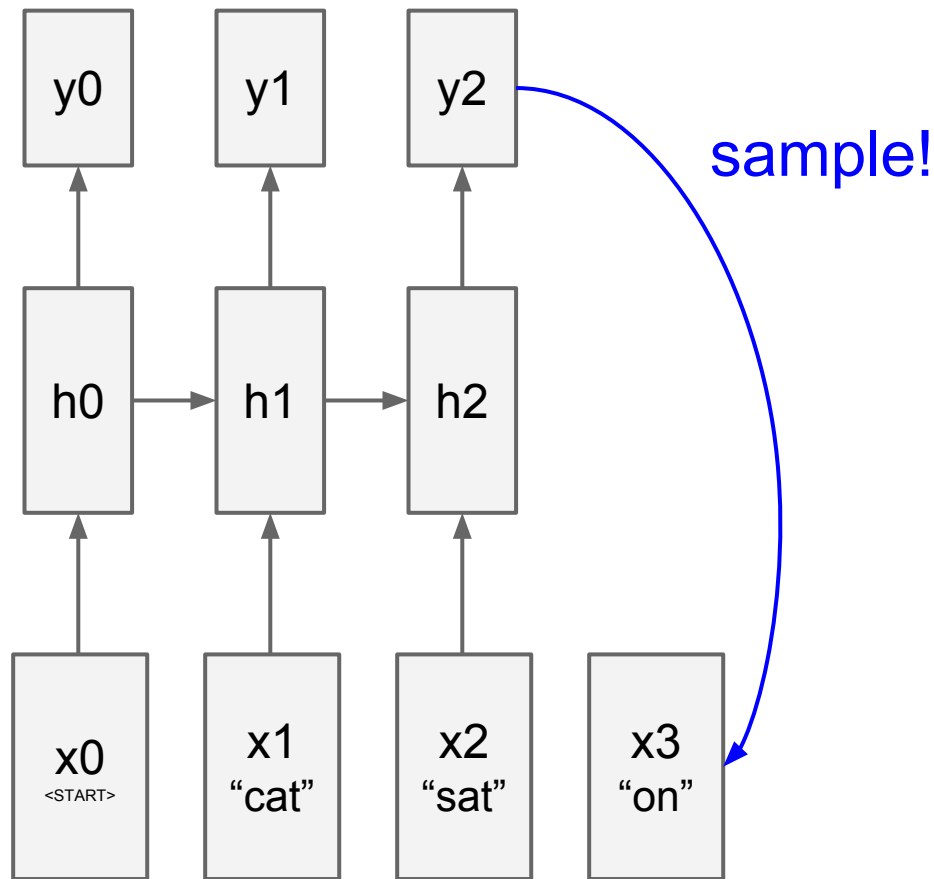
Training this on a lot of sentences would give us a language model. A way to predict

$P(\text{next word} \mid \text{previous words})$



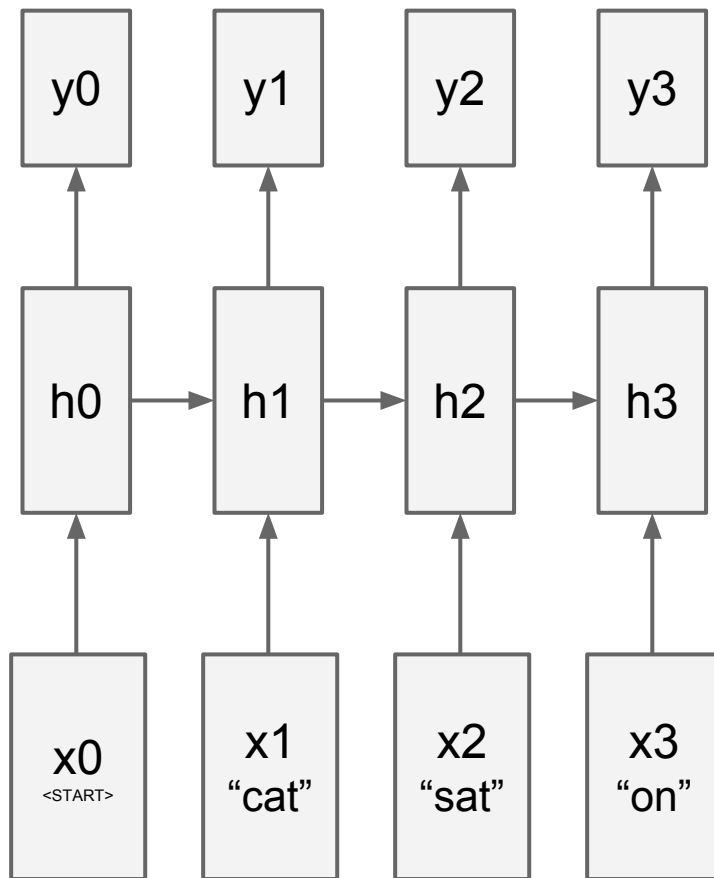
Training this on a lot of sentences would give us a language model. A way to predict

$P(\text{next word} \mid \text{previous words})$



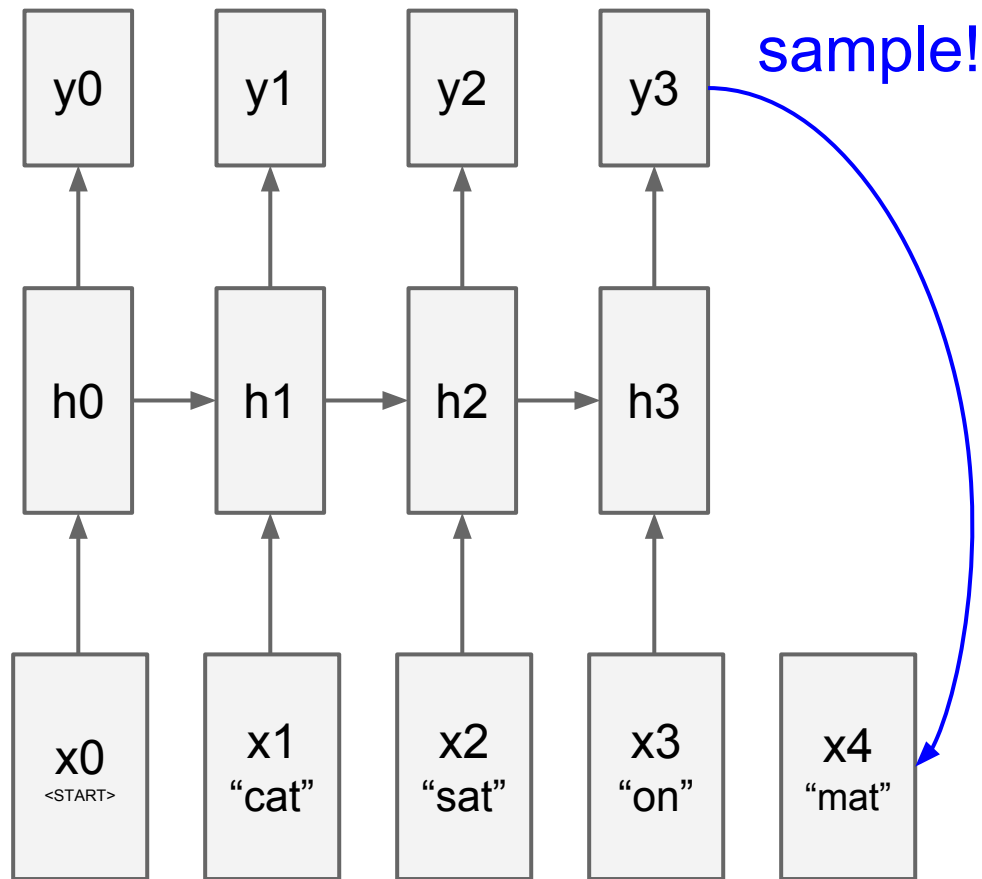
Training this on a lot of sentences would give us a language model. A way to predict

$P(\text{next word} \mid \text{previous words})$



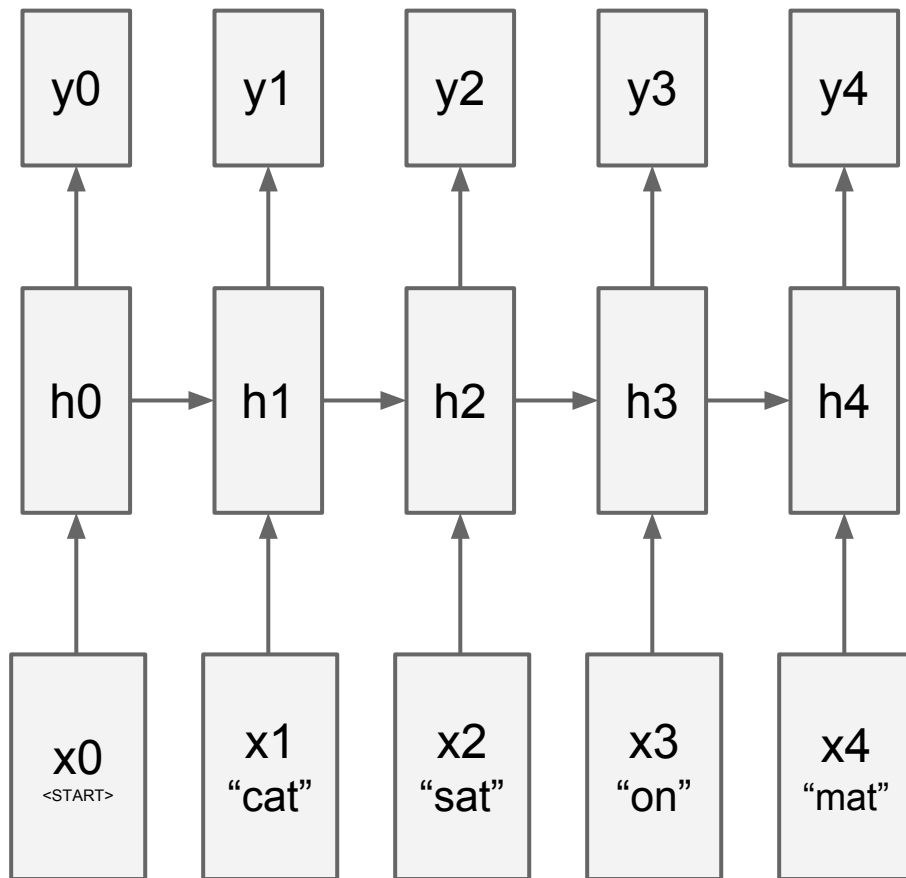
Training this on a lot of sentences would give us a language model. A way to predict

$$P(\text{next word} \mid \text{previous words})$$



Training this on a lot of sentences would give us a language model. A way to predict

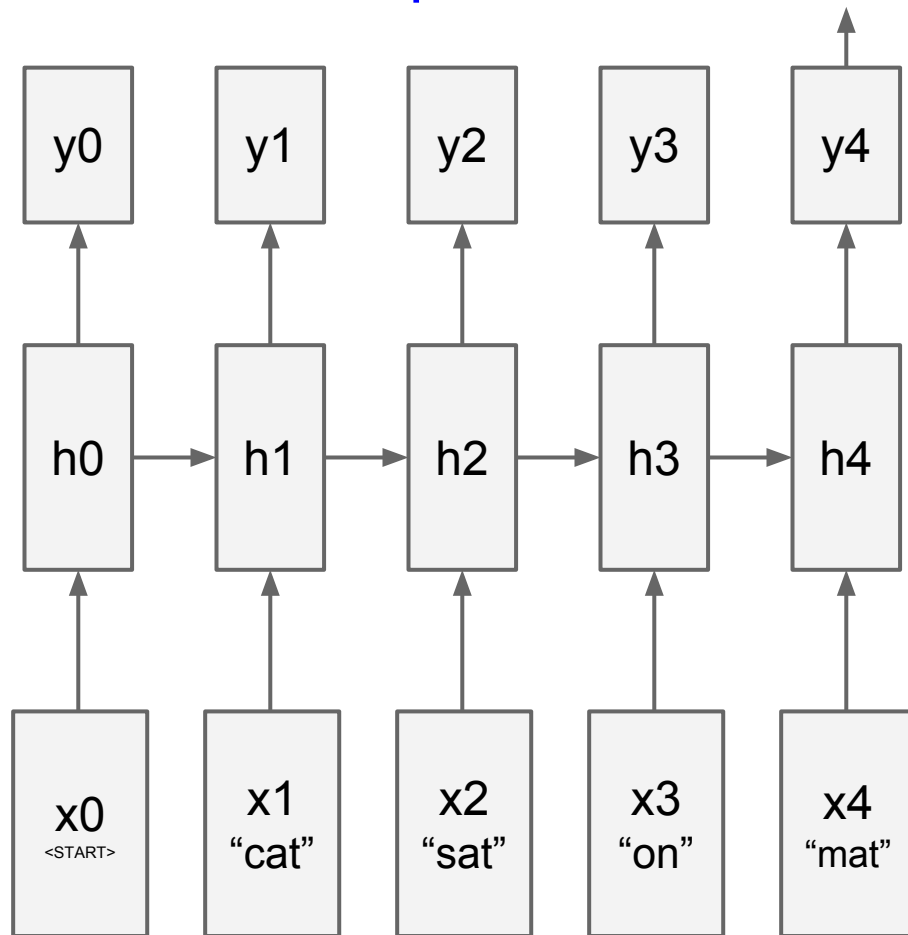
$P(\text{next word} \mid \text{previous words})$



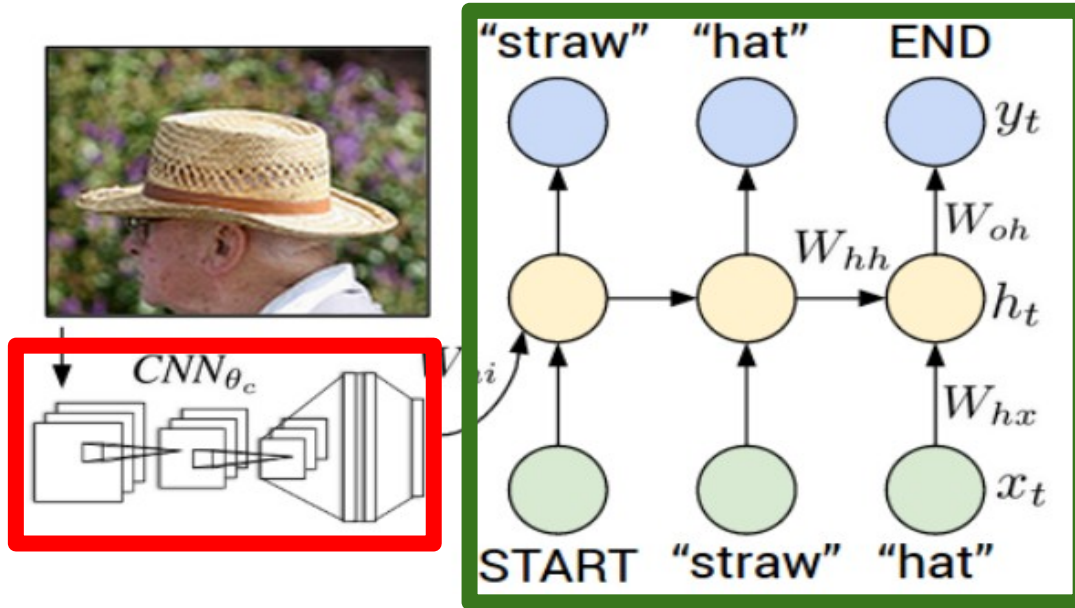
Training this on a lot of sentences would give us a language model. A way to predict

$$P(\text{next word} \mid \text{previous words})$$

samples <END>? done.



Recurrent Neural Network



Convolutional Neural Network



“straw hat”

training example

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

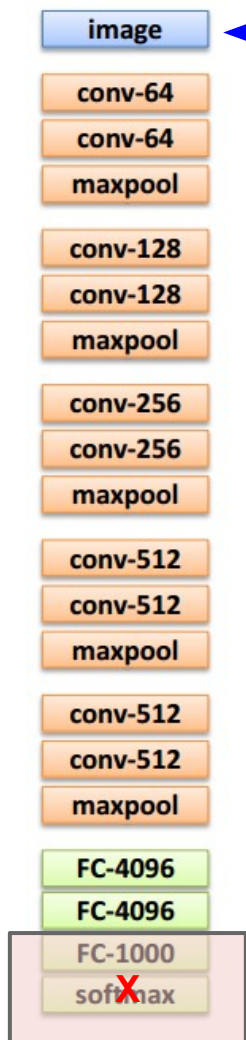
FC-1000

softmax



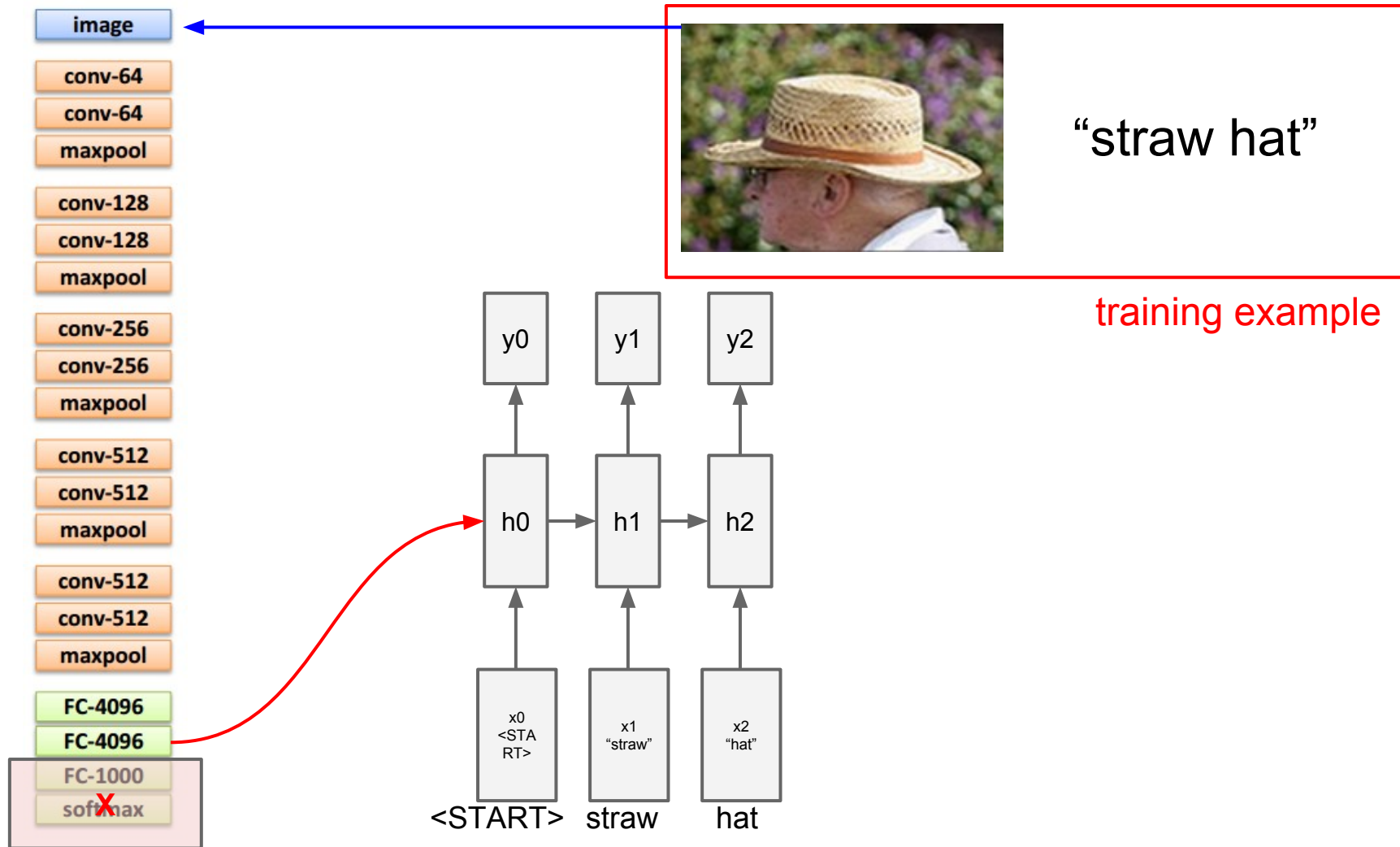
“straw hat”

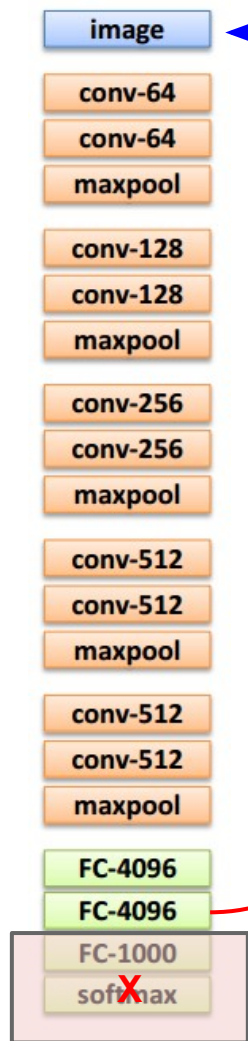
training example



“straw hat”

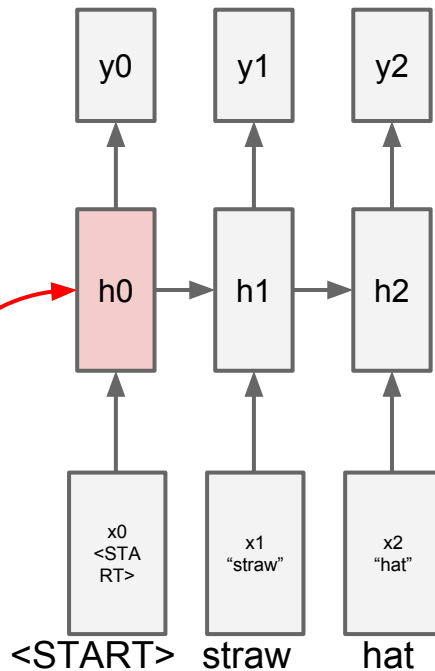
training example





“straw hat”

training example

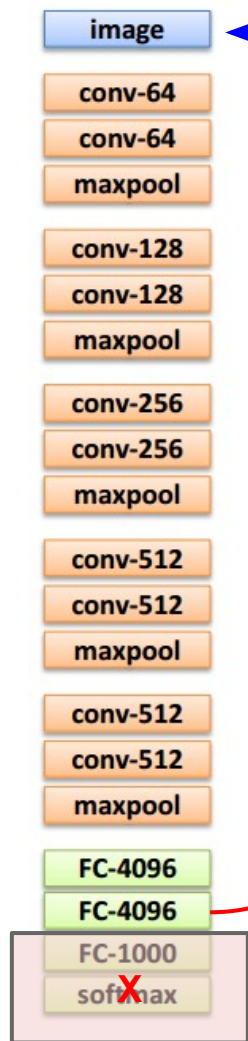


before:

$$h0 = \max(0, W_{xh} * x0)$$

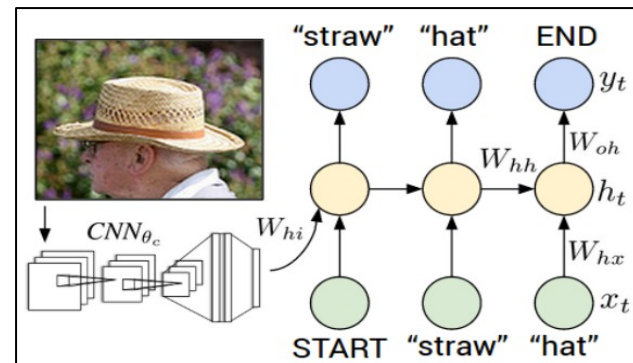
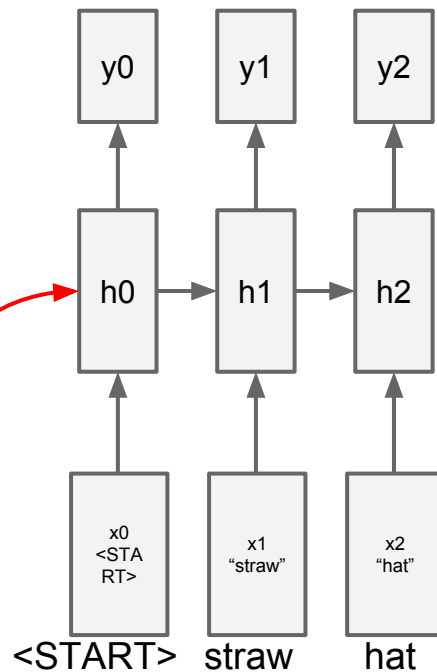
now:

$$h0 = \max(0, W_{xh} * x0 + W_{ih} * v)$$



“straw hat”

training example





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-4096



test image

x0
<STA
RT>

<START>

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>

<START>

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
<START
RT>

x1

sample!

<START>

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

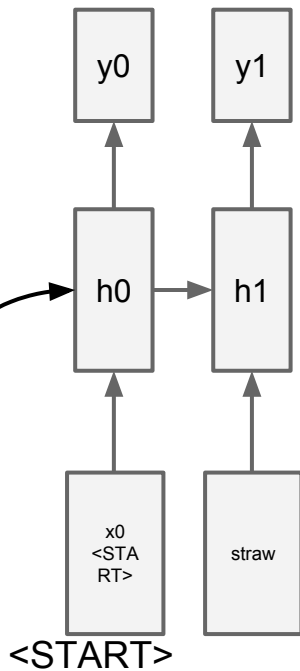
maxpool

FC-4096

FC-4096



test image



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
<START
RT>

straw

hat

sample!

<START>

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

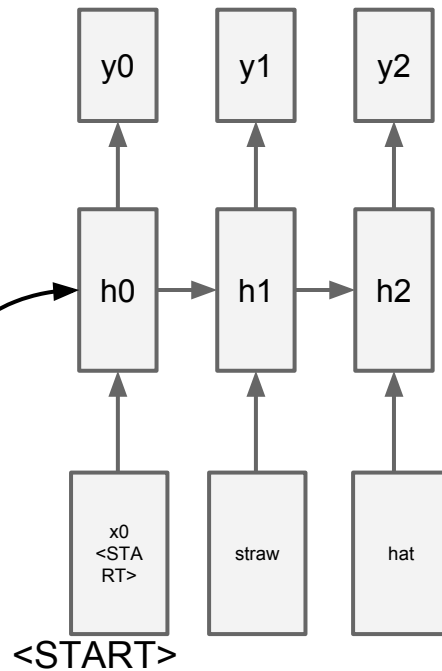
maxpool

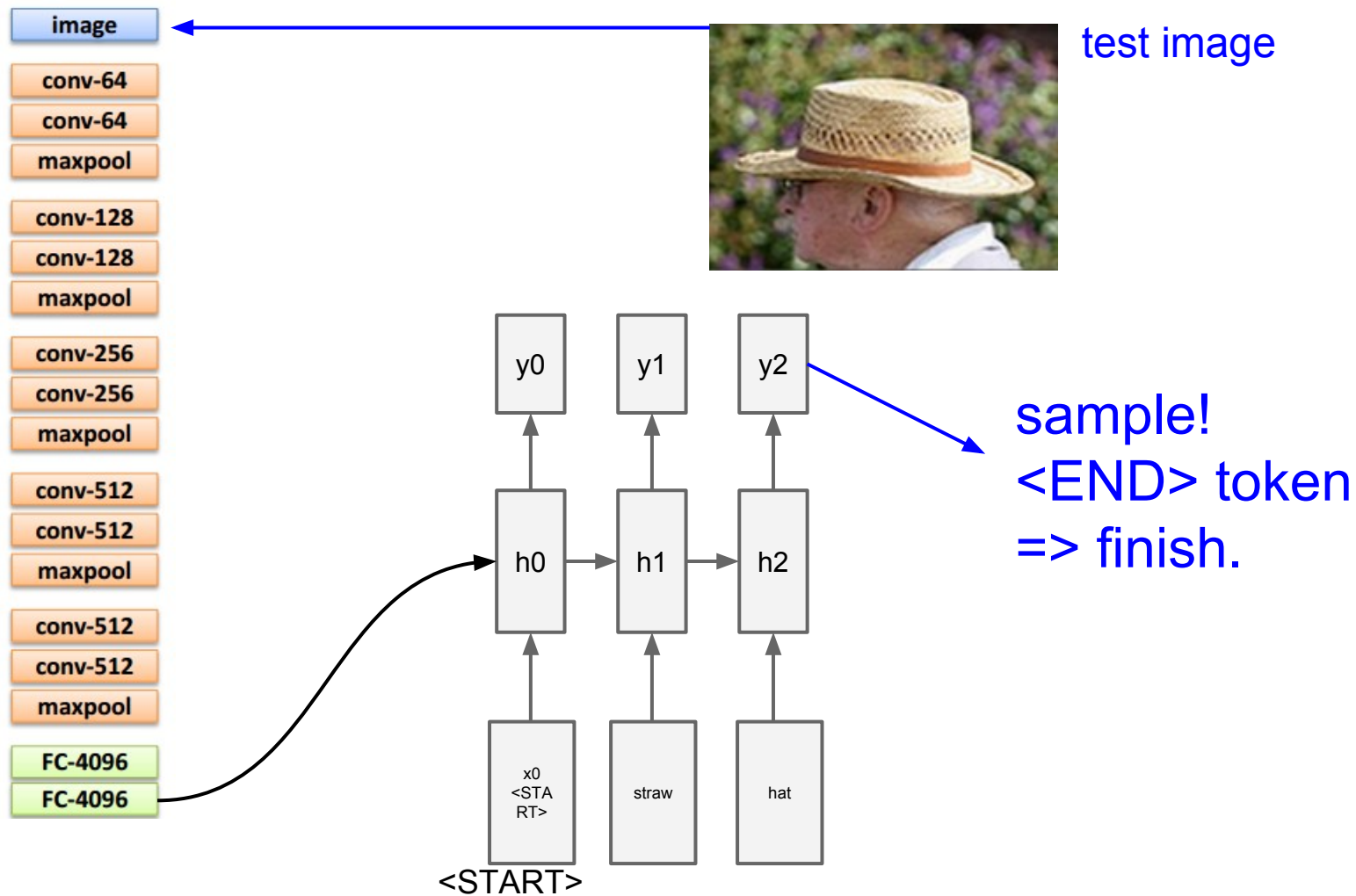
FC-4096

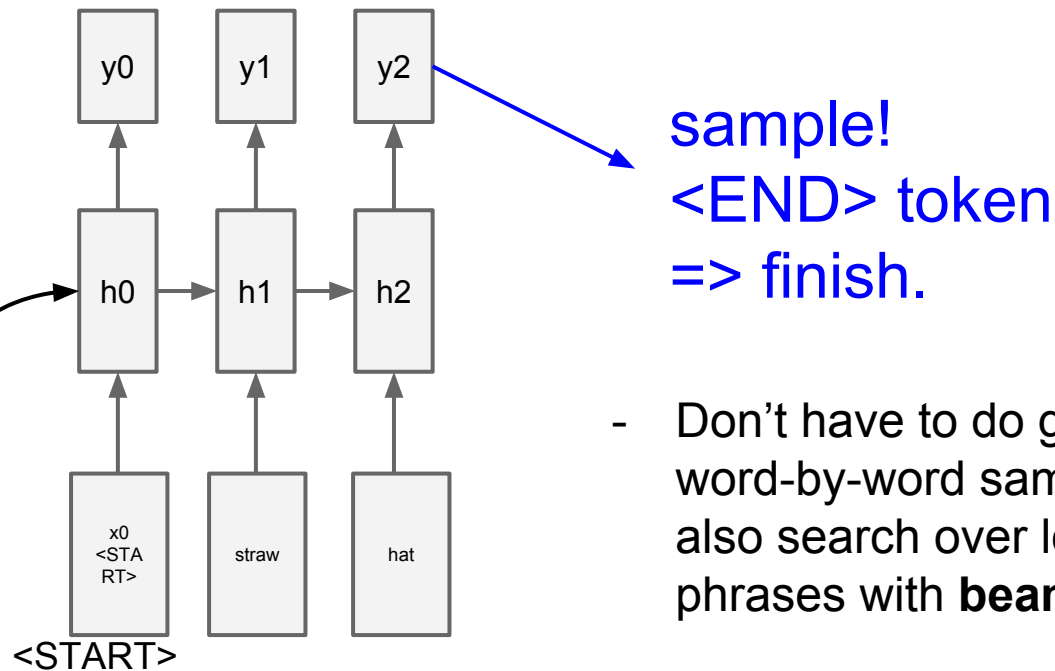
FC-4096



test image

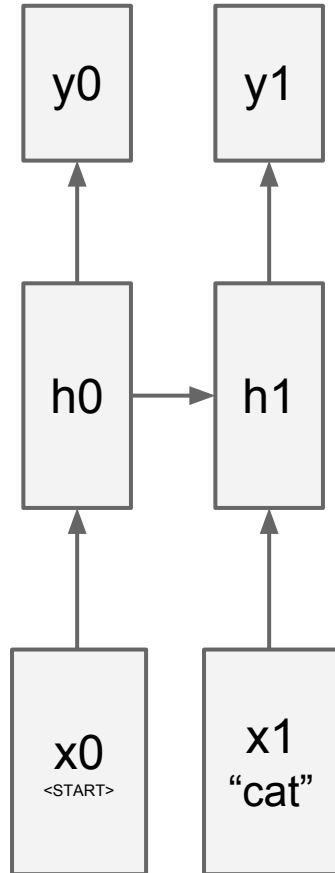






- Don't have to do greedy word-by-word sampling, can also search over longer phrases with **beam search**

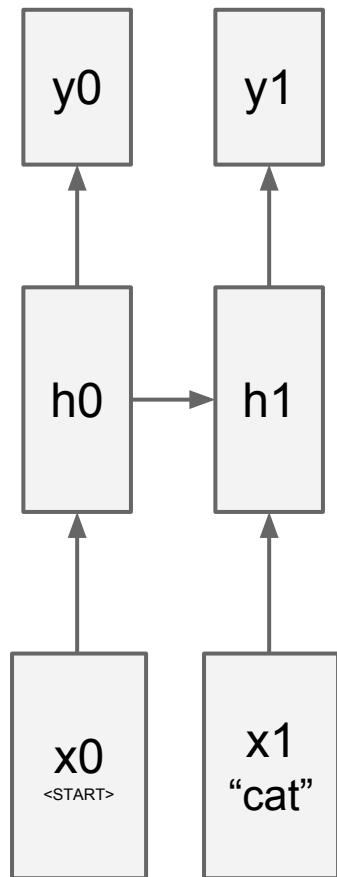
RNN vs. LSTM



"hidden" representation
(e.g. 200 numbers)

$$h_1 = \max(0, W_{xh} * x_1 + W_{hh} * h_0)$$

RNN vs. LSTM



"hidden" representation
(e.g. 200 numbers)

$$h_1 = \max(0, W_{xh} * x_1 + W_{hh} * h_0)$$

LSTM changes the form of the equation for h_1 such that:

1. more expressive multiplicative interactions
2. gradients flow nicer
3. network can explicitly decide to reset the hidden state

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:

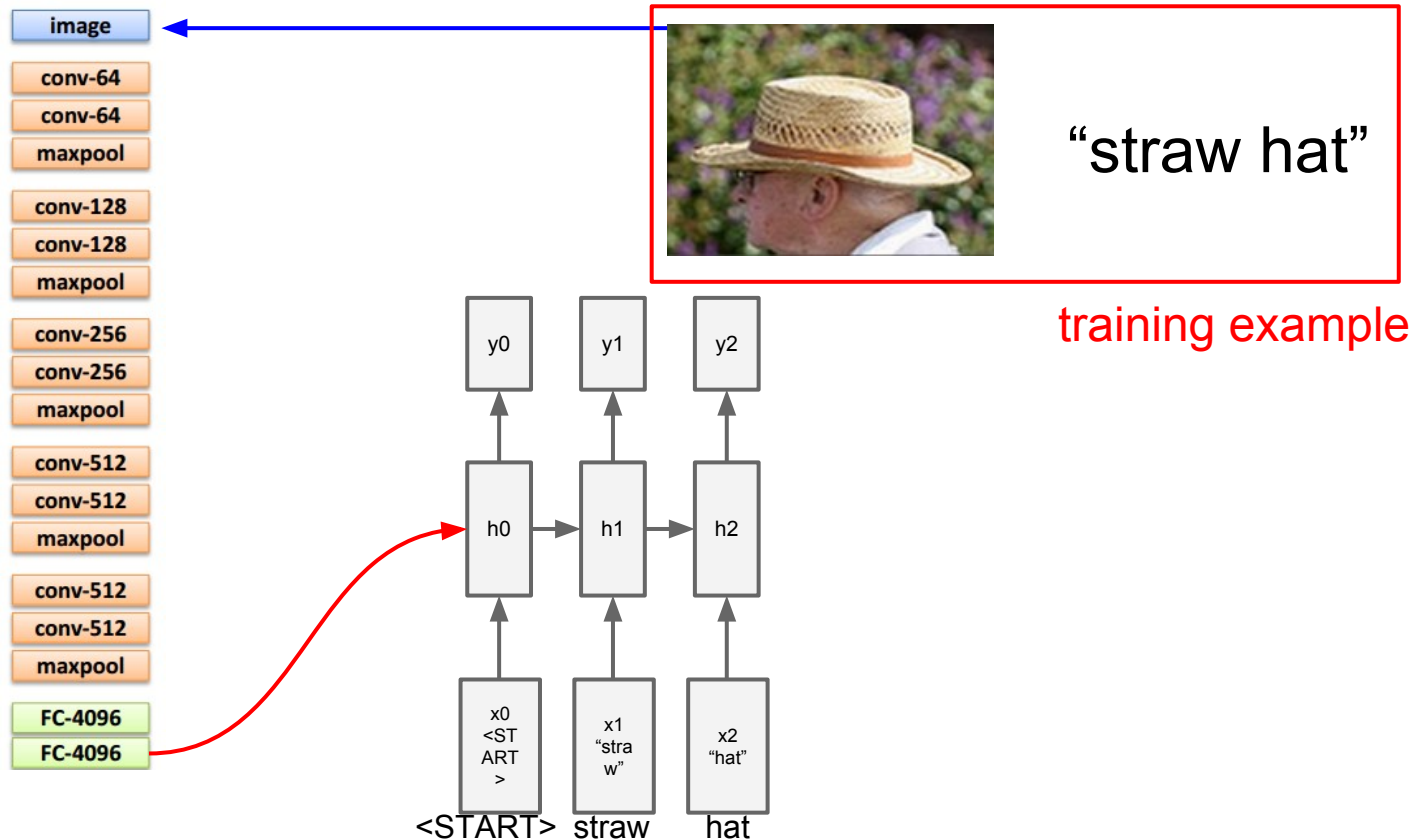
~120K images

~5 sentences each

Training an RNN/LSTM...

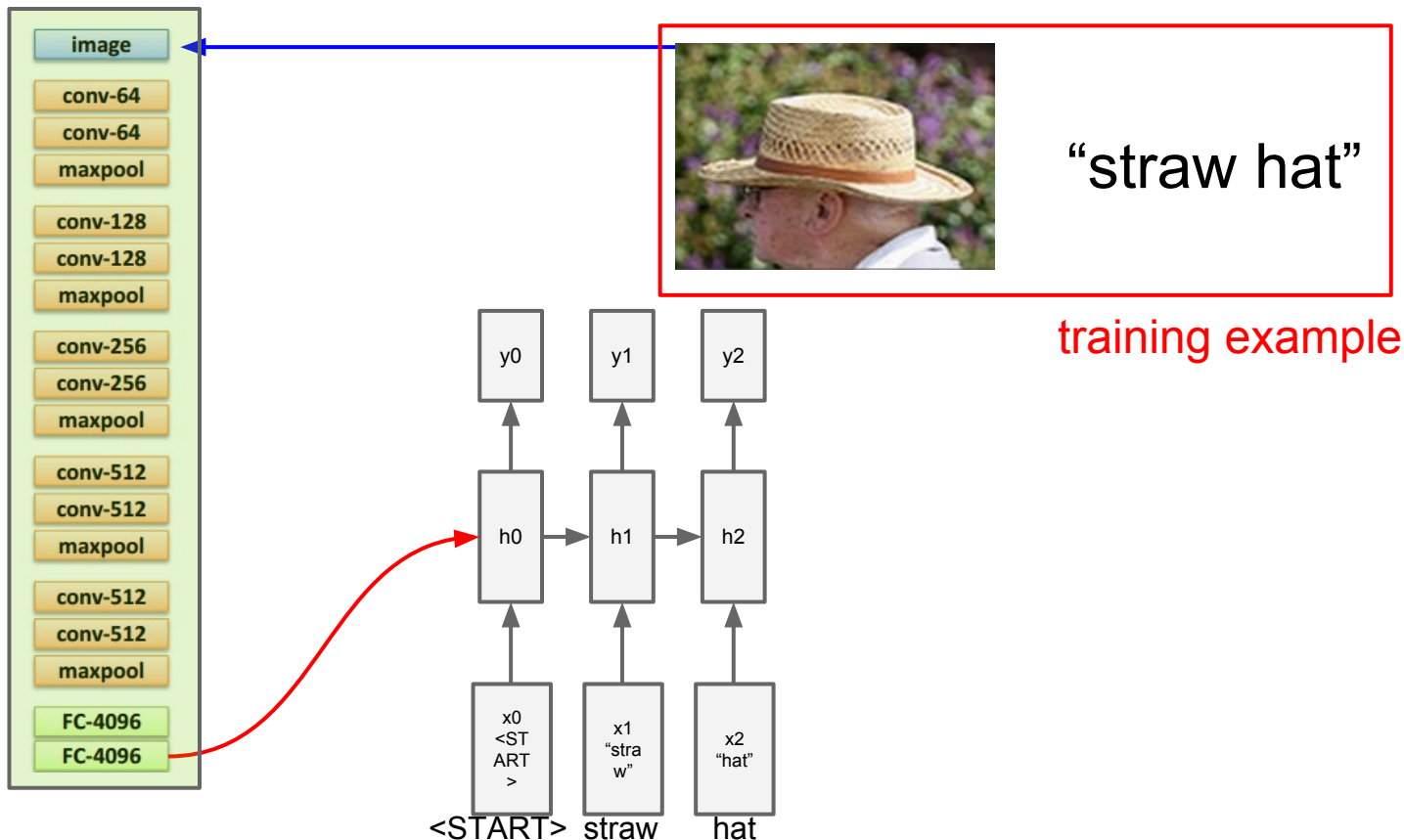
- **Clip** the gradients (important!). 5 worked ok
- **RMSprop** adaptive learning rate worked nice
- Initialize softmax **biases** with log word frequency distribution
- Train for **long time**

+ Transfer Learning



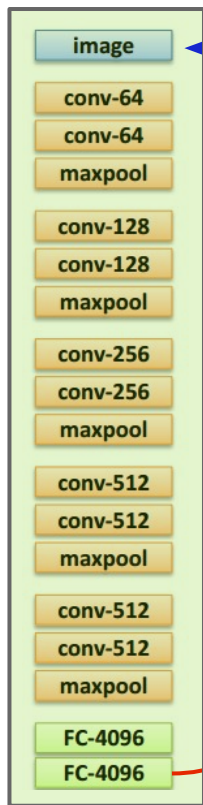
+ Transfer Learning

use weights
pretrained from
ImageNet



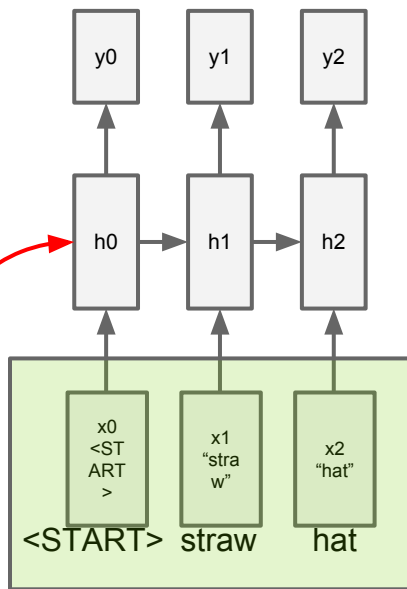
+ Transfer Learning

use weights
pretrained from
ImageNet



“straw hat”

training example



use word vectors
pretrained with
word2vec [1]

Summary of the approach

We wanted to describe images with sentences.

1. Define a single function from input $\rightarrow$ output
2. Initialize parts of net from elsewhere if possible
3. Get some data
4. Train with SGD

Wow I can't believe that worked



a group of people standing
around a room with
remotes
logprob: -9.17



a young boy is holding a
baseball bat
logprob: -7.61



a cow is standing in the middle of a street
logprob: -8.84

Wow I can't believe that worked



a cat is sitting on a toilet seat
logprob: -7.79



a display case filled with lots of different types of donuts
logprob: -7.78



a group of people sitting at a table with wine glasses
logprob: -6.71

Well, I can kind of see it



a man standing next to a clock on a wall
logprob: -10.08



a young boy is holding a
baseball bat
logprob: -7.65

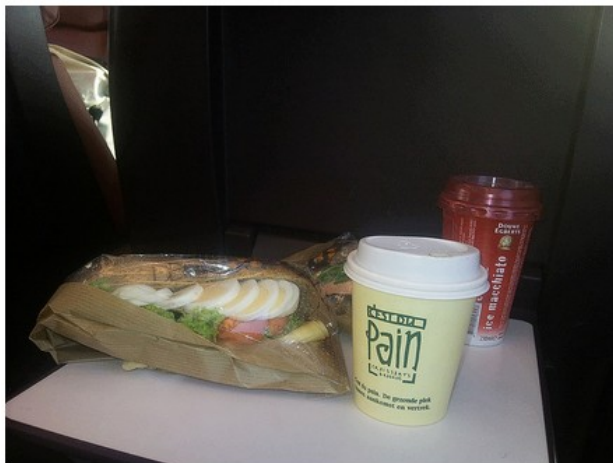


a cat is sitting on a couch with a remote control
logprob: -12.45

Well, I can kind of see it



a baby laying on a bed with a stuffed bear
logprob: -8.66

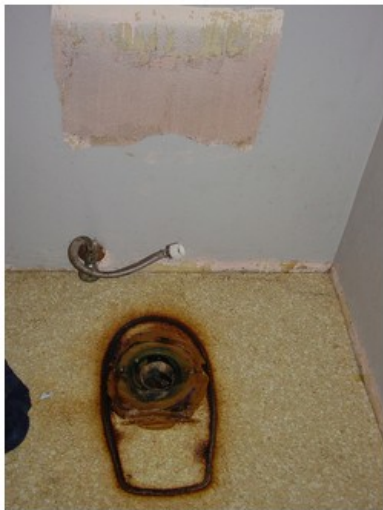


a table with a plate of food and a cup of coffee
logprob: -9.93



a young boy is playing frisbee in the park
logprob: -9.52

Not sure what happened there...



a toilet with a seat up in a bathroom
logprob: -13.44



a woman holding a teddy bear in front of a mirror
logprob: -9.65



a horse is standing in the middle of a road
logprob: -10.34

See predictions on
1000 COCO images:
<http://bit.ly/neuraltalkdemo>

NeuralTalk Sentence Generation Results

Showing results for coco on 1000 images

Eval params were: ("result_struct_filename": "result_struct.json", "beam_size": 1, "checkpoint_path": "cv/model_checkpoint_coco_visionlab43.stanford.edu_lstm_11.14.p", "dump_folder": "out1", "max_images": 1000)
Final average perplexity of ground truth words: 11.56



a group of people walking down a street
logprob: ~-3.13



a man is standing on a beach with a surfboard
logprob: ~-10.33



a woman is holding a black umbrella in the grass
logprob: ~-10.59



a plate of food with a sandwich and a salad
logprob: ~-9.69



a woman sitting on a bench with a laptop
logprob: ~-8.55



a traffic light with a red light on top of it
logprob: ~-10.33



a cat laying on a bed with a laptop
logprob: ~-10.67



a group of people sitting at a table with wine glasses
logprob: ~-6.71



a bus is parked in a parking lot
logprob: ~-8.06



a pizza with toppings on a white plate
logprob: ~-7.40



a herd of sheep standing on top of a lush green field
logprob: ~-6.22



a giraffe standing in a field of grass
logprob: ~-7.43



a pizza with toppings on a white plate
logprob: ~-7.09



a large clock tower with a clock on top
logprob: ~-8.21



a woman is holding a pink umbrella in her hand
logprob: ~-11.47



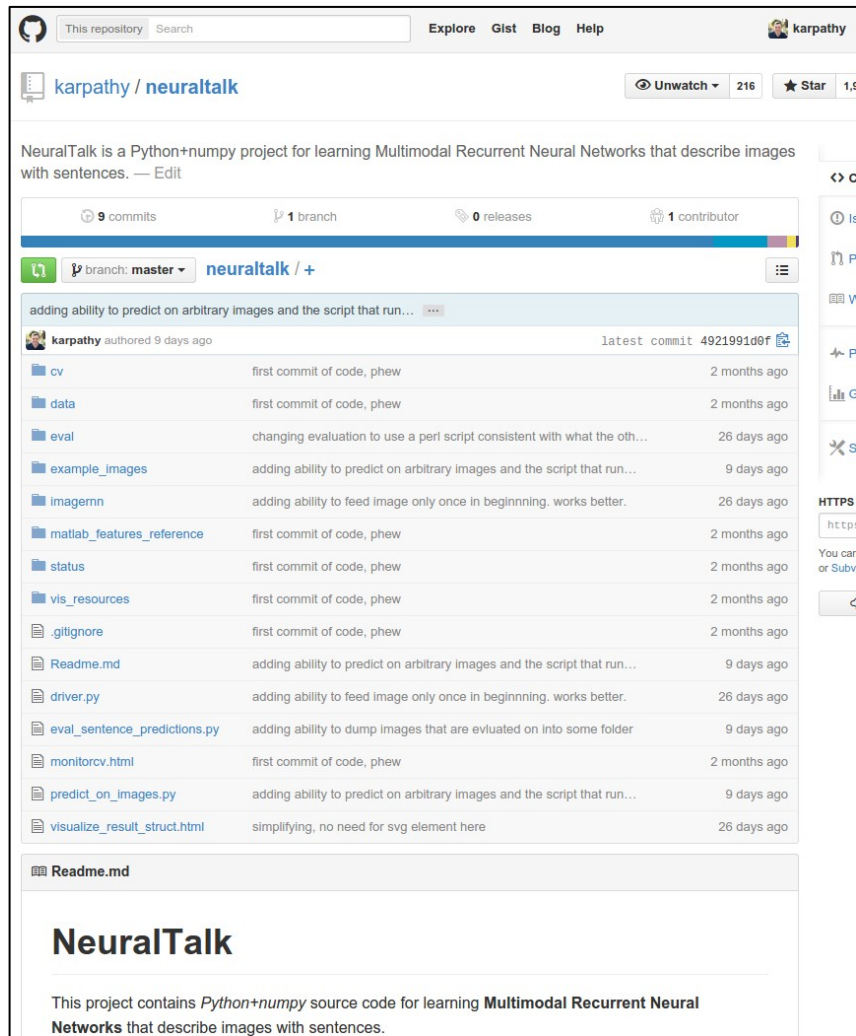
a plate of food with a fork and a glass of wine
logprob: ~-11.50

What this approach Doesn't do:

- There is no *reasoning*
- A single glance is taken at the image, no objects are detected, etc.
- We can't just describe any image

NeuralTalk

- Code on Github
- Both RNN/LSTM
- Python+numpy (CPU)
- Matlab+Caffe if you want to run on new images (for now)



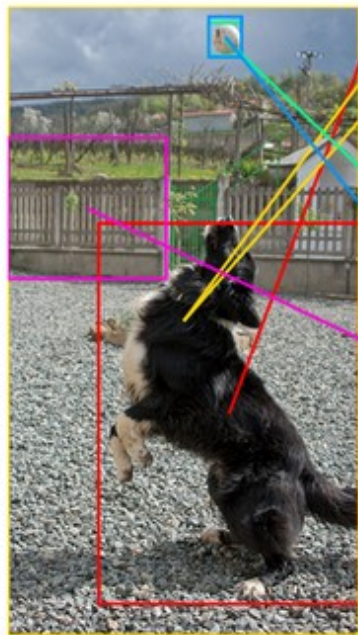
The screenshot shows the GitHub repository page for `karpathy / neuraltalk`. The repository is a Python+numpy project for learning Multimodal Recurrent Neural Networks that describe images with sentences. It has 9 commits, 1 branch, 0 releases, and 1 contributor.

The commit history table is as follows:

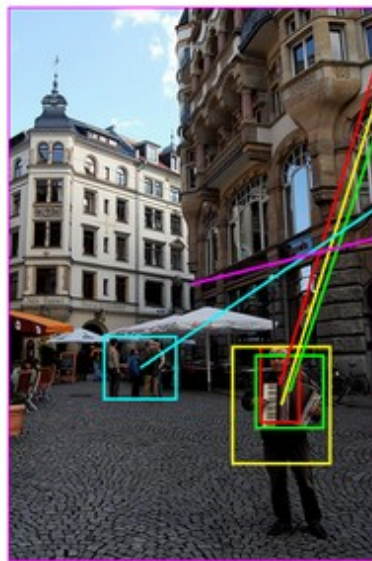
File	Commit Message	Time Ago
cv	first commit of code, phew	2 months ago
data	first commit of code, phew	2 months ago
eval	changing evaluation to use a perl script consistent with what the oth...	26 days ago
example_images	adding ability to predict on arbitrary images and the script that run...	9 days ago
imagernn	adding ability to feed image only once in beginning, works better.	26 days ago
matlab_features_reference	first commit of code, phew	2 months ago
status	first commit of code, phew	2 months ago
vis_resources	first commit of code, phew	2 months ago
.gitignore	first commit of code, phew	2 months ago
Readme.md	adding ability to predict on arbitrary images and the script that run...	9 days ago
driver.py	adding ability to feed image only once in beginning, works better.	26 days ago
eval_sentence_predictions.py	adding ability to dump images that are evaluated on into some folder	9 days ago
monitorcv.html	first commit of code, phew	2 months ago
predict_on_images.py	adding ability to predict on arbitrary images and the script that run...	9 days ago
visualize_result_struct.html	simplifying, no need for svg element here	26 days ago

The Readme.md section is titled **NeuralTalk** and states: "This project contains *Python+numpy* source code for learning **Multimodal Recurrent Neural Networks** that describe images with sentences."

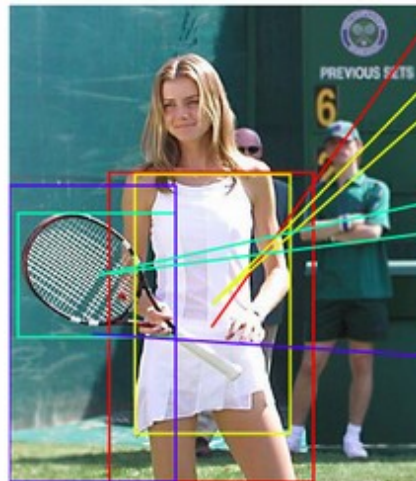
Ranking model



1.31 dog
0.31 plays
0.45 catch
-0.02 with
0.25 white
1.62 ball
-0.10 near
-0.07 wooden
0.22 fence



0.26 man
0.31 playing
1.51 accordion
-0.07 among
-0.08 in
0.42 public
0.30 area

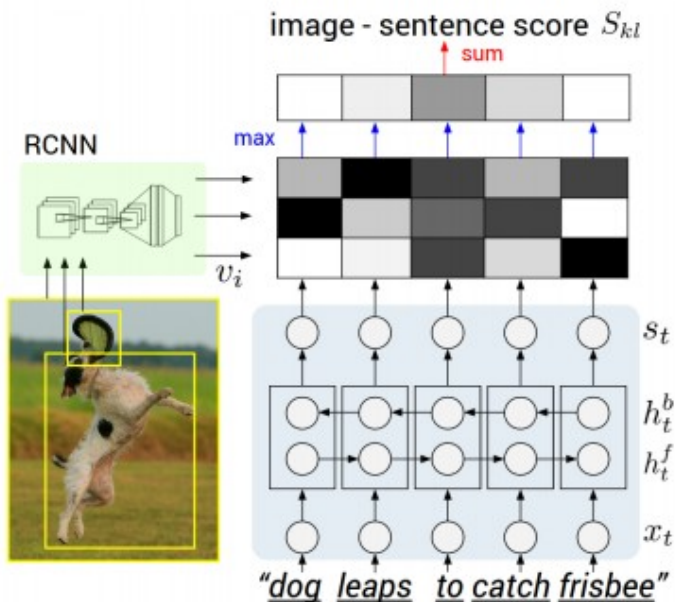


1.12 woman
-0.28 in
1.23 white
1.45 dress
0.06 standing
-0.13 with
3.58 tennis
1.81 racket
0.06 two
0.05 people
-0.14 in
0.30 green
-0.09 behind
-0.14 her

Ranking model

web demo:

<http://bit.ly/rankingdemo>



Grounded Image Sentence Retrieval

For every test set sentence below we retrieve the top images (from set of 1000). Yellow number top left of each image = score. Clicking on each image reveals the precise inferred grounding. Red border = incorrect retrieval, green border = correct retrieval. Yellow border = ground truth image that wasn't retrieved among top 5 predictions.

Test set of size: 1000 images

regex filter sentences

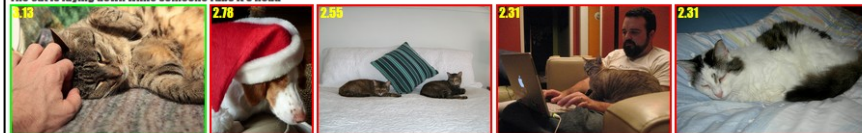
A cop riding on the back of a black horse.



top sentences for this image:

- (4.82) a horse cop sitting on a horse on a public street.
- (3.65) The drawing depicts a man with a horse and two people on the porch.
- (3.60) An older man is riding a horse down the street.
- (3.59) A man riding a horse in front of a fence.
- (3.56) A man on a horse walking down the middle of a street.

The cat is laying down while someone rubs it's head



The man in the dark suit stands in the dark room.

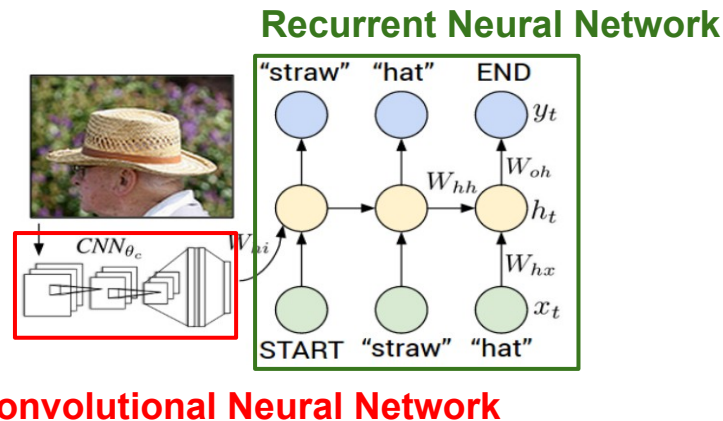


A very large farmers market with customers waiting for samples.

Summary

Neural Networks:

- input->output **end-to-end** optimization
- stackable / composable like Lego
- easily support Transfer Learning
- work very well.



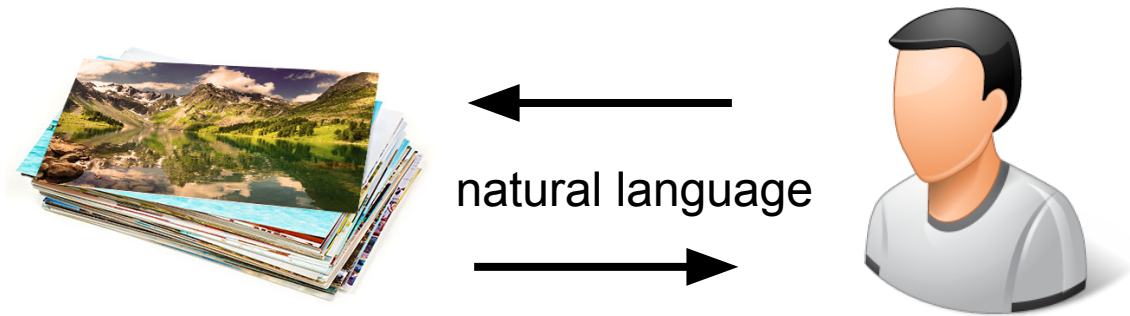
Summary

1. image $\rightarrow$ sentence
2. sentence $\rightarrow$ image

Summary

1. image $\rightarrow$ sentence

2. sentence $\rightarrow$ image



Summary

1. image $\rightarrow$ sentence
2. sentence $\rightarrow$ image



Thank you!