Automated Image Captioning with ConvNets and Recurrent Nets

Andrej Karpathy, Fei-Fei Li

Summer 2013
Google Research Internship
Large-Scale Supervised Deep Learning for Videos.

2011-
Stanford Computer Science Ph.D. student
Machine Learning (with emphasis on Deep Learning) and Computer Vision. My adviser is Fei-Fei Li.

Summer 2011
Google Research Internship
Large-Scale Unsupervised Deep Learning for Videos.

2009-2011
University of British Columbia: Master's Degree
I worked with Michiel van de Panne on Learning Compositional Controllers for Physically-simulated Articulate Figures.

2005-2009
University of Toronto: Bachelor's Degree
Double major in Computer Science and Physics.

"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

Very hard task

vzntrf bs zr fphon qvivat arkg gb ghegyr

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	52	12	50	77	91	08
49	49	99	40	17	81	18	57	60	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
86	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	99	16	07	97	57	32	16	26	26	79	33	27	98	66
88	36	68	87	57	62	20	72	03	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	23	57	05	54
01	70	54	71	83	51	54	69	16	92	33	48	61	43	52	01	89	19	67	48

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]

IMAGENET

Image classification
Classification error
0.3
0.2
0.1
0
2010
2011
2012
2013
2014
ILSVRC year

[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**

[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

[7x7x512]

[1x1x4096] “neurons”

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

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

Pooling Layer

[224x224x64]

[112x112x64]

Performs (spatial) downsampling

Pooling Layer

Pooling Layer

Max Pooling Layer

Single depth slice

x

1	1	2	4
5	**6**	7	**8**
3	2	1	0
1	2	3	**4**

y

max pool

6	8
3	4

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

Example activation maps

(tiny VGGNet trained with ConvNetJS)

image

[224x224x3]

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

[1000]

differentiable function

[224x224x3]

differentiable function

[1000]

0.2	0.4	0.09	0.01	0.3
cat	dog	chair	bagel	banana

image
[224x224x3]
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
[1000]
differentiable function
0.2
0.4
0.09
0.01
0.3
cat
dog
chair
bagel
banana

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.

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

Recurrent Neural Network

Recurrent Networks are good at modeling sequences...

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 :)

2-layer LSTM

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(next word | previous words)

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

We want to train a **language model**:
P(next word | previous words)

i.e. want these to be high:
P(cat | [<S>])
P(sat | [<S>, cat])
P(on | [<S>, cat, sat])
P(mat | [<S>, cat, sat, on])

“cat sat on mat”

300 (learnable) numbers associated with each word

P(word | [<S>])
P(word | [<S>, cat])
P(word | [<S>, cat, sat])
P(word | [<S>, cat, sat, on])
P(word | [<S>, cat, sat, on, mat])
"cat sat on mat"
y0
y1
y2
y3
y4
h0
h1
h2
h3
h4
x0
<START>
x1
"cat"
x2
"sat"
x3
"on"
x4
"mat"
10,001 numbers (logprobs for 10,000 words in vocabulary and a special <END> token)
y4 = Why * h4
300 (learnable) numbers associated with each word

P(word | [<S>])
P(word | [<S>, cat])
P(word | [<S>, cat, sat])
P(word | [<S>, cat, sat, on])
P(word | [<S>, cat, sat, on, mat])
"cat sat on mat"
y0
y1
y2
y3
y4
h0
h1
h2
h3
h4
x0
<START>
x1
"cat"
x2
"sat"
x3
"on"
x4
"mat"
10,001 numbers (logprobs for 10,000 words in vocabulary and a special <END> token)
y4 = Why * h4
"hidden" representation mediates the contextual information (e.g. 200 numbers)
h4 = max(0, Wxh * x4 + Whh * h3)
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(next word | previous words)

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

P(next word | previous words)

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

P(next word | previous words)

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

P(next word | previous words)

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

P(next word | previous words)

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

P(next word | previous words)

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

P(next word | previous words)

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

P(next word | previous words)

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

P(next word | previous words)

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

P(next word | previous words)

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

P(next word | previous words)

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

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

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
y0
y1
y2
h0
h1
h2
x0 <STARTRT>
x1 "straw"
x2 "hat"
<START>
straw
hat

“straw hat”

training example

before:

h0 = max(0, Wxh * x0)

now:

h0 = max(0, Wxh * x0 + Wih * 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-4096
FC-1000
softmax
"straw hat"
training example
y0
y1
y2
h0
h1
h2
x0 <START>
x1 "straw"
x2 "hat"
<START>
straw
hat
"straw"
"hat"
END
y_t
W_{oh}
W_{hh}
h_t
W_{hx}
x_t
W_{hi}
CNN_{θ_c}
START
"straw"
"hat"

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

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

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

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

test image

sample!
<END> token
=> finish.

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

RNN vs. LSTM

“hidden” representation
(e.g. 200 numbers)
h1 = max(0, Wxh * x1 + Whh * h0)

RNN vs. LSTM

“hidden” representation
(e.g. 200 numbers)
h1 = max(0, Wxh * x1 + Whh * h0)

LSTM changes the form of the equation for **h1** 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

training example

+ Transfer Learning

use weights pretrained from ImageNet

"straw hat"

training example

use word vectors pretrained with word2vec [1]

[1] Mikolov et al., 2013

Summary of the approach

We wanted to describe images with sentences.

1. Define a single function from input -> 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

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)

Ranking model

Ranking model

web demo:

http://bit.ly/rankingdemo

Summary

Neural Networks:

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

Summary

1. image -> sentence
2. sentence -> image

Summary

1. image -> sentence
2. sentence -> image

Summary

1. image -> sentence
2. sentence -> image

natural language

Thank you!