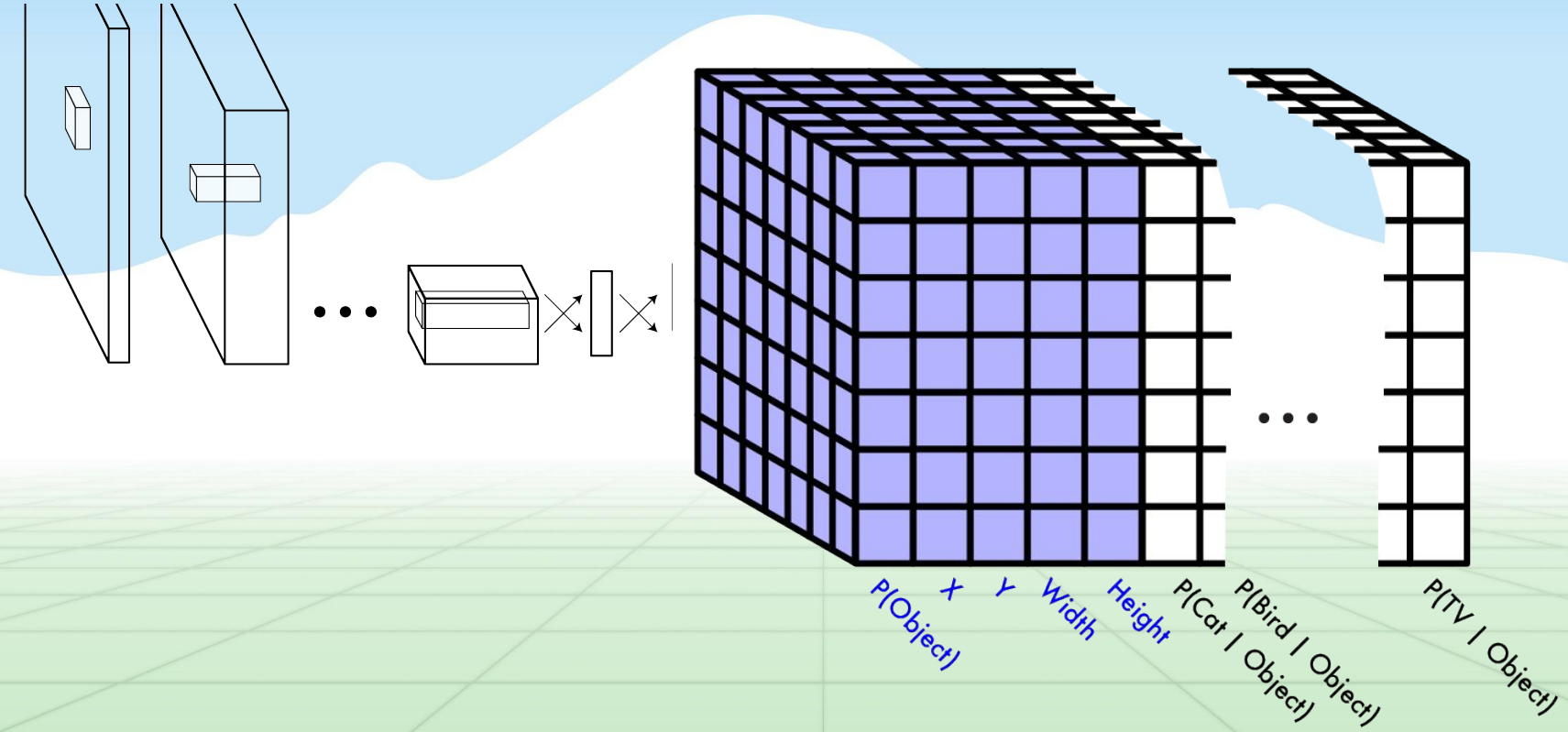

The Ancient
Secrets
Computer Vision

Logistics:

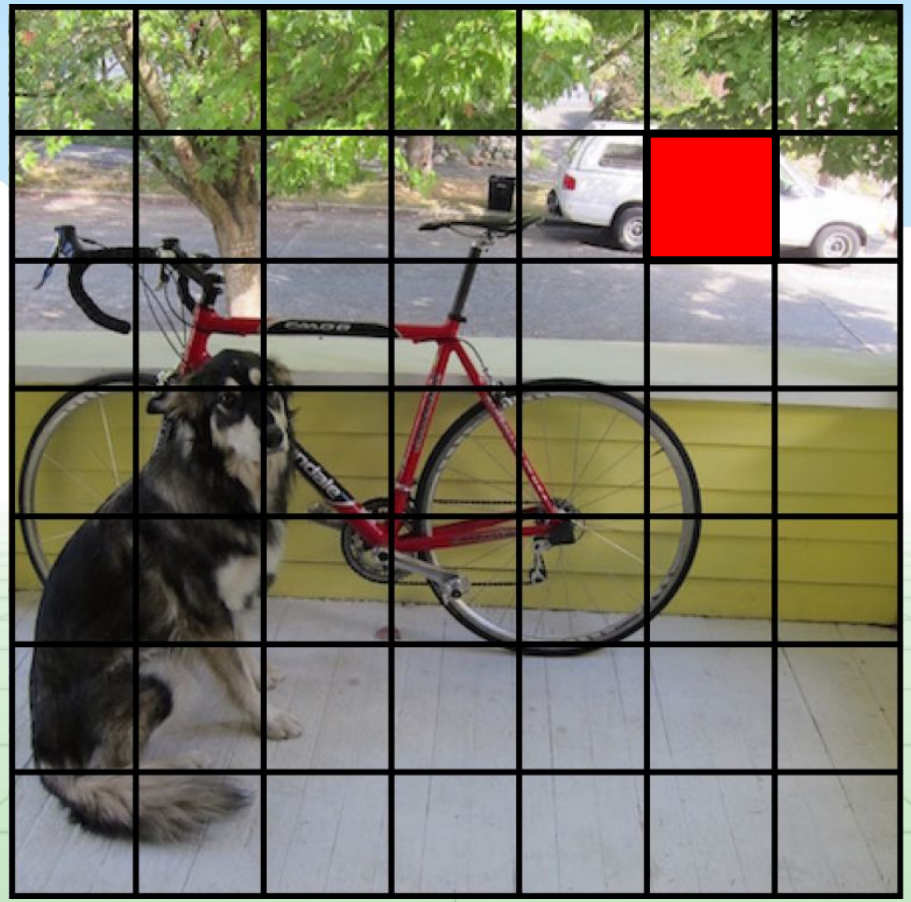
- You are working on your final projects (I hope!)
- Deliverables:
  - Poster session, June 6th, 4:30 - 6:30pm
  - Set up at 4pm, poster board and easels provided
  - You will be graded by at least 3 people (TAs or me or Ali)
  - Rubric is on Canvas
  - Demos are encouraged (live or recorded examples of project)
  - Poster options:
    - Make your own, print it, assemble it on poster board
    - OR
    - Use Kiana's template, send it to cse455-staff@cs.washington.edu Sunday
    - Template is on the website
    - Turn in poster on Canvas as well.


Previously
On
the
Ancient Secrets
of Computer Vision

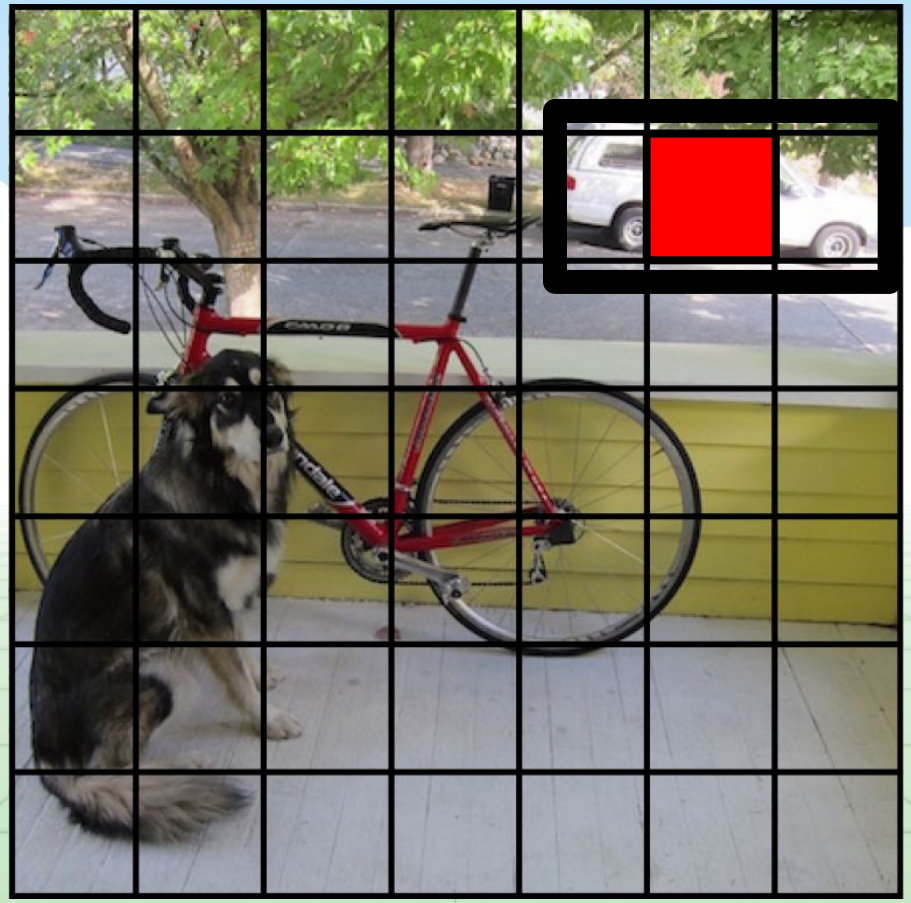
Tensor encoding detection



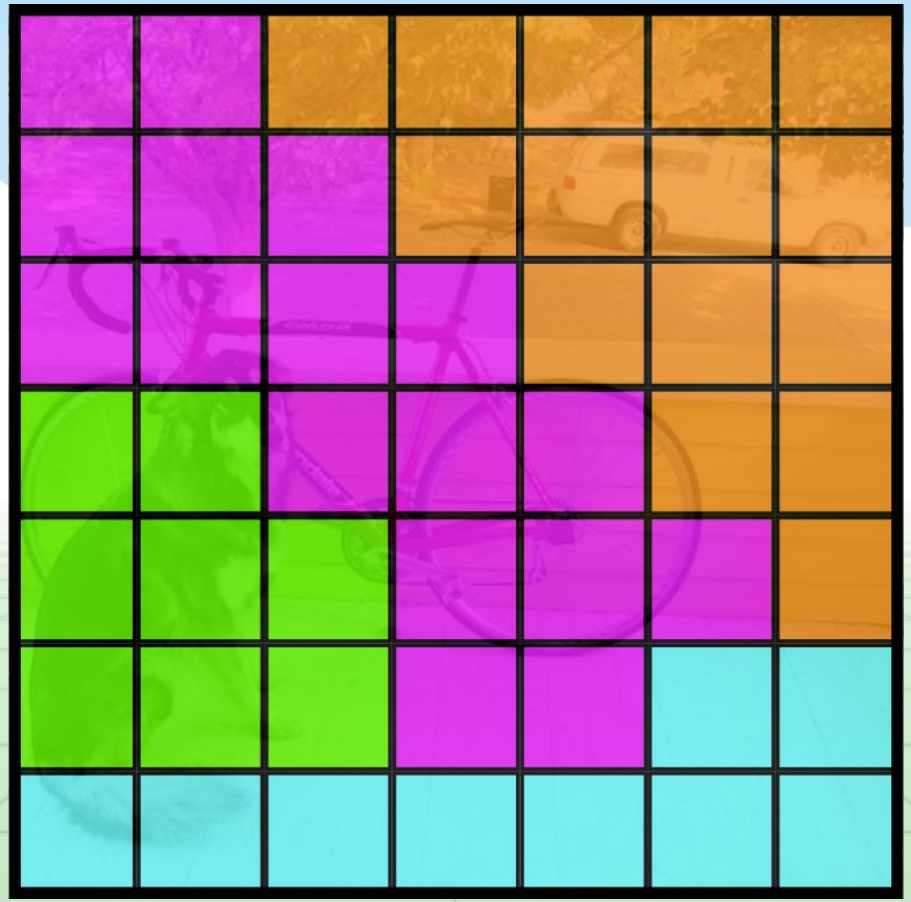
For each cell predicts P(obj)



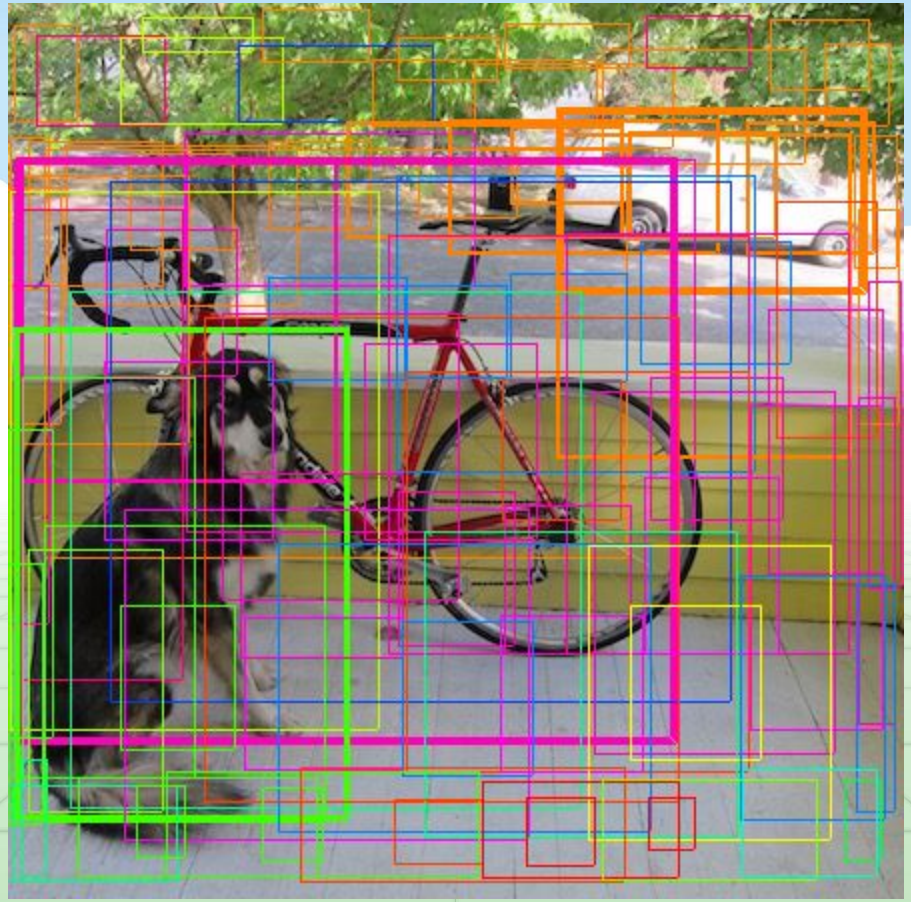
Also predict a bounding box



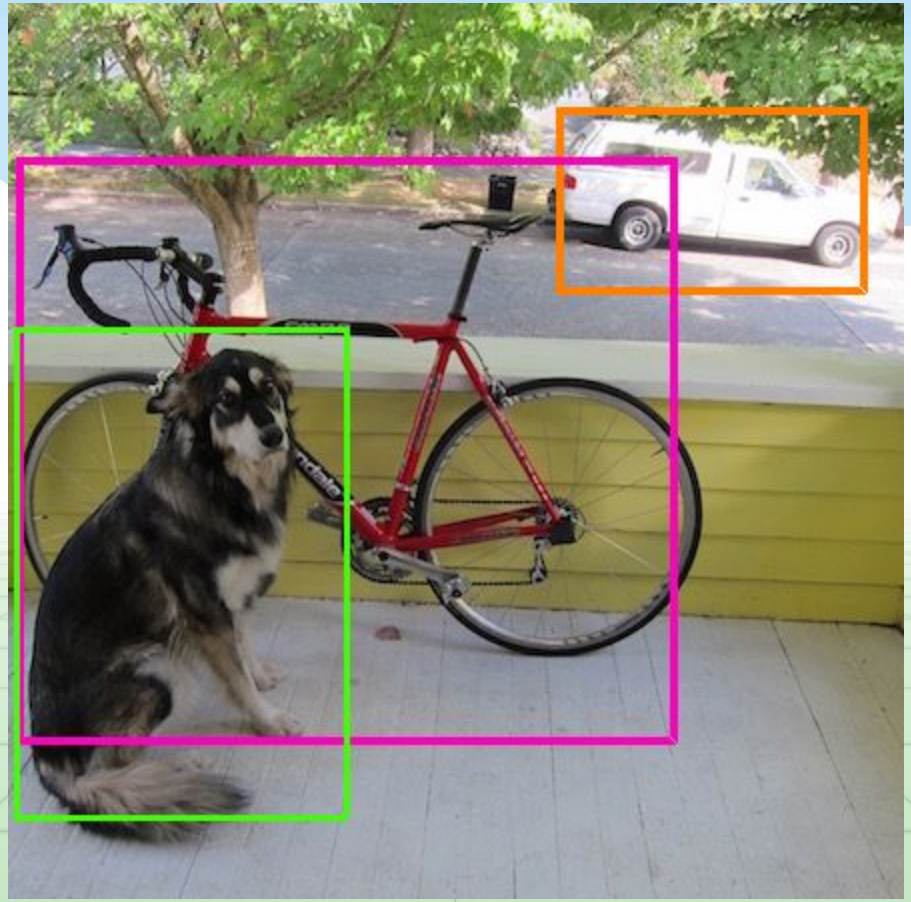
Also class probabilities



Also class probabilities



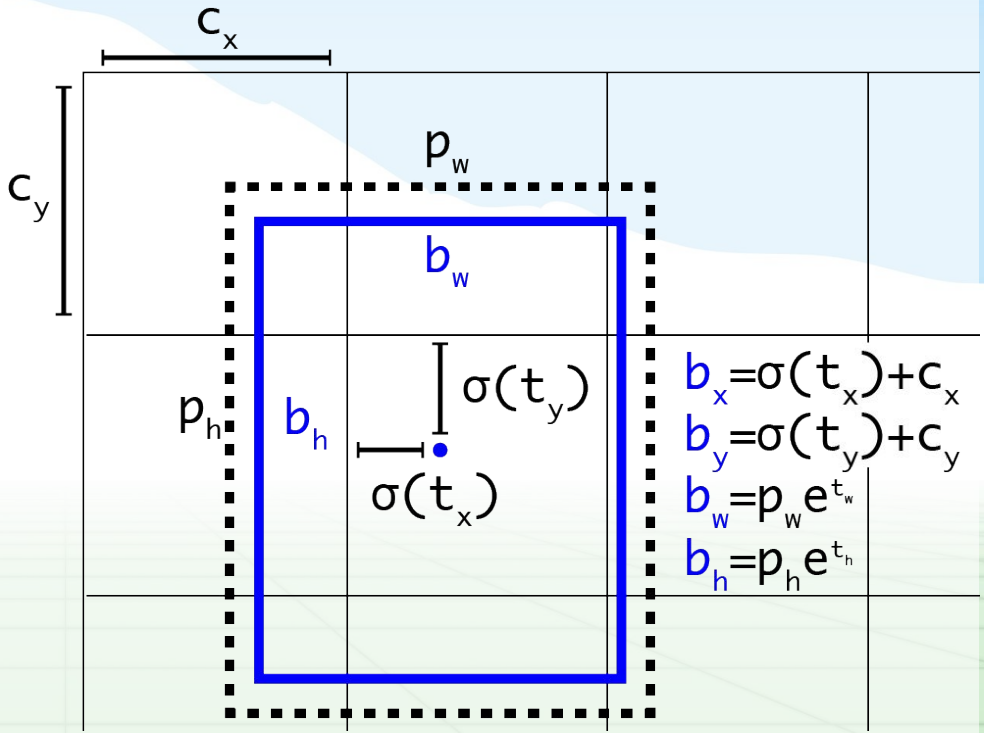
Threshold and non-max suppression



Bounding box encoding (YOLOv2 and v3)

Given cell offset c_x and c_y
Prior box p_w and p_h

Our predicted bounding box is given by:



YOLO loss function

$L(YOLO) = \alpha_1 L(confidence) + \alpha_2 L(localization) + \alpha_3 L(classification)$

Can tune all the alphas

L(confidence): binary cross-entropy

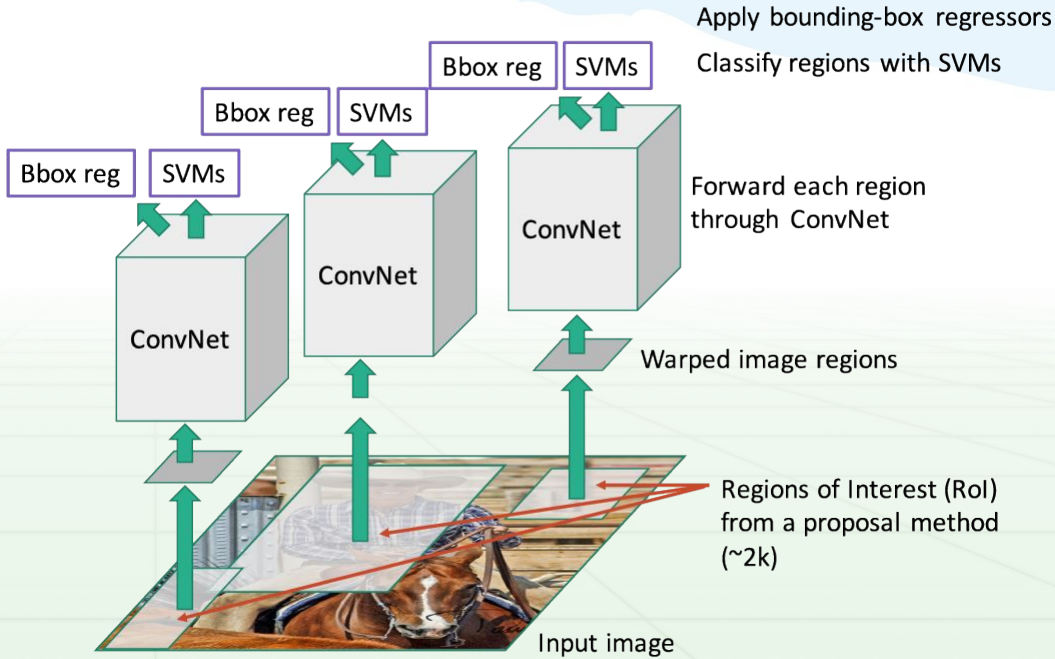
L(localization): RMSE

L(classification): multi-class cross-entropy or
1 v all binary cross-entropy

	Pascal 2007 mAP	Speed	
DPM v5	33.7	.07 FPS	14 s/img
R-CNN	66.0	.05 FPS	20 s/img
YOLO	63.4	45 FPS	22 ms/img
YOLOv2	78.6	40 FPS	25 ms/img
YOLOv3	83.1	30 FPS	33 ms/img
Fast R-CNN	70.0	.5 FPS	2 s/img
Faster R-CNN	73.2	7 FPS	140 ms/img
Resnet FRCNN	76.4	??	??
ResNet + COCO data	83.8	??	??
R-FCN	83.6	6 FPS	170 ms/img

R-CNN is slow

Run convnet independently over every ROI



https://dl.dropboxusercontent.com/s/vlyrkgd8nz8gy5l/fast-rcnn.pdf?dl=0

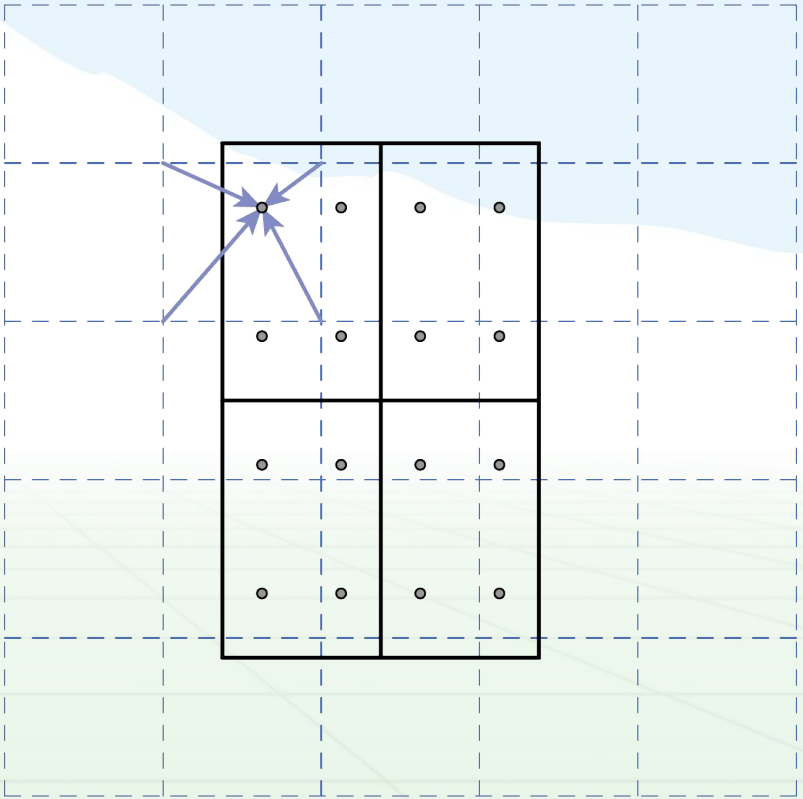
ROI Align

Better than ROI Pool so we'll talk about it instead

Split ROI into fixed size (say 2x2)

Sample image at multiple points for each cell in ROI (bilinear interp.)

Pool over these samples (max, avg…)



https://arxiv.org/pdf/1703.06870.pdf

Fast R-CNN

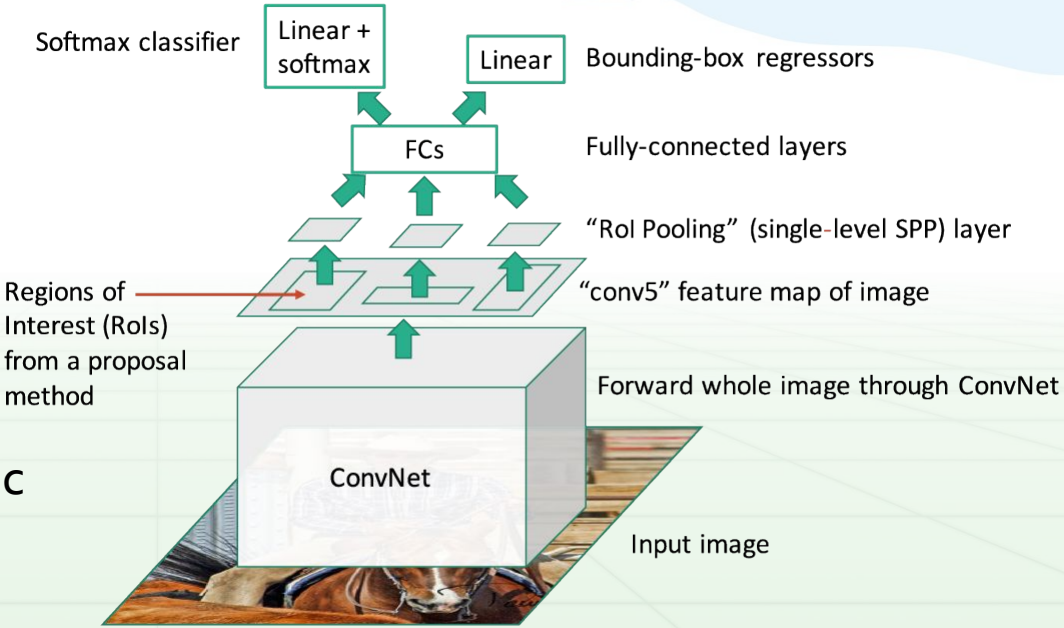
Run convnet once, extract features using ROI pooling

ROI Pool:

Convert variable sized ROI to fixed size output

Much faster, no independent network evals (except last linear layer)

Still slow region proposer, selective search takes ~2 sec



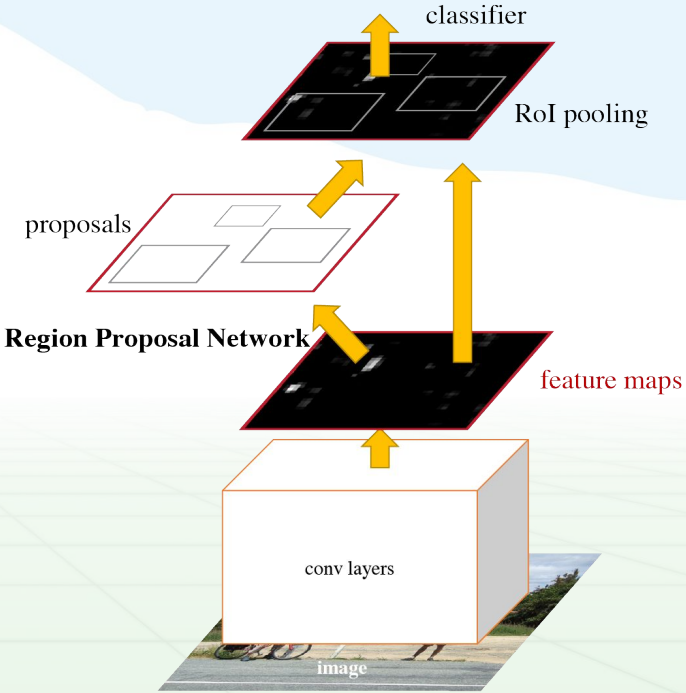
https://dl.dropboxusercontent.com/s/vlyrkgd8nz8gy5l/fast-rcnn.pdf?dl=0

Faster R-CNN

Use Convnet to propose regions *and* generate features

ROI Pool to fix size of ROI features

Additional layers to classify and predict bbox for ROIs



https://dl.dropboxusercontent.com/s/vlyrkgd8nz8gy5l/fast-rcnn.pdf?dl=0

Common Objects in COntext (COCO)

80 objects

117,261 train/val images

902,435 object instances

New detection metric, mAP averaged over IOU [.5 - .95]

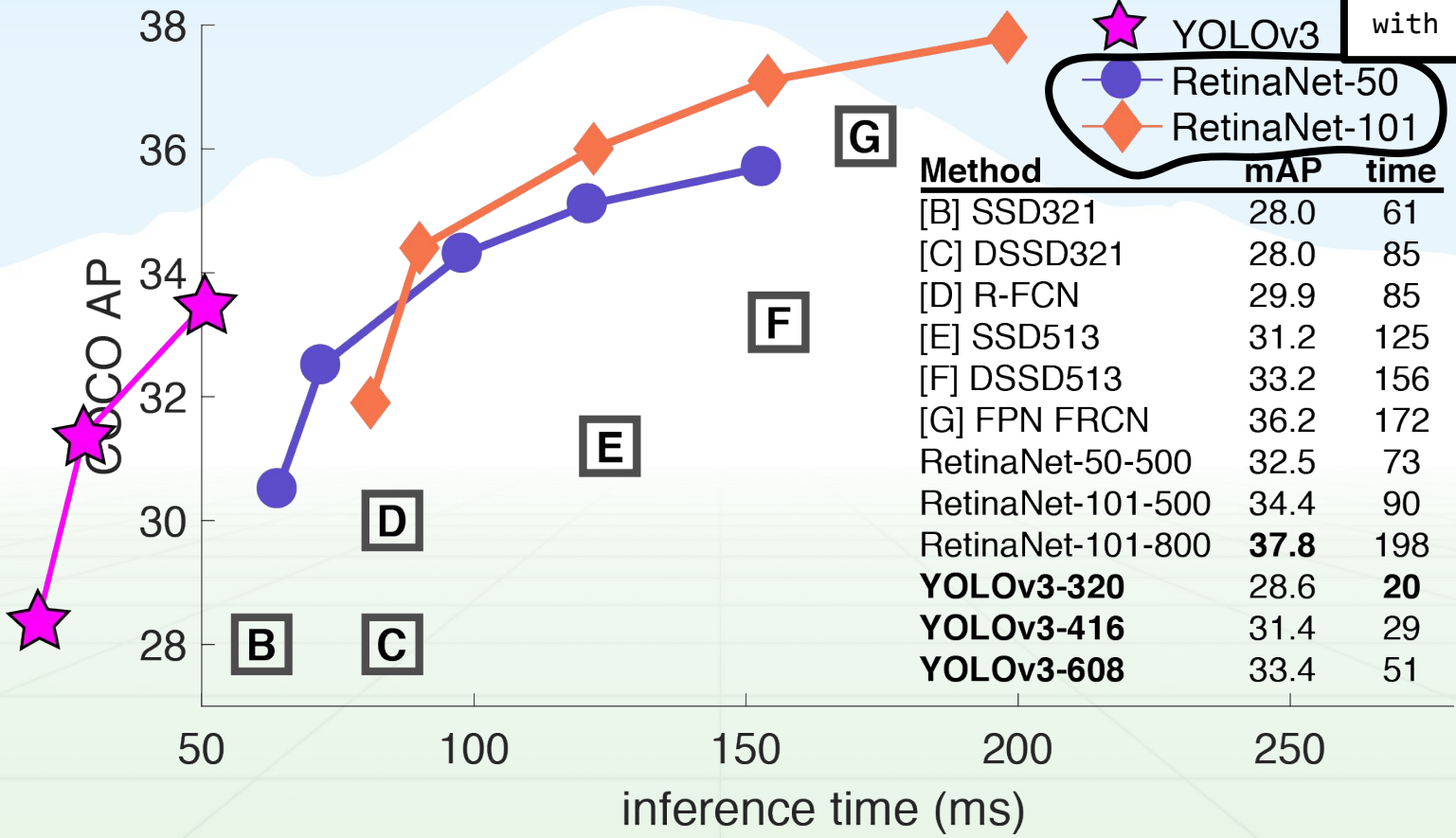
Segmentation masks for each instance

Originally by Microsoft but they were scared of copyright something something so they spun it off

http://cocodataset.org/#home

Performance on COCO

These are Faster R-CNN with new loss

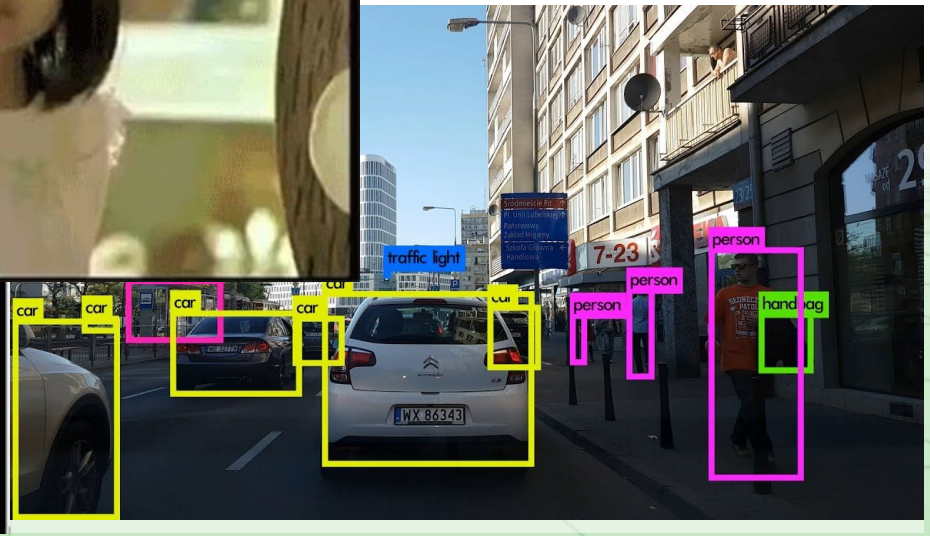


Method	mAP	time
[B] SSD321	28.0	61
[C] DSSD321	28.0	85
[D] R-FCN	29.9	85
[E] SSD513	31.2	125
[F] DSSD513	33.2	156
[G] FPN FRCN	36.2	172
RetinaNet-50-500	32.5	73
RetinaNet-101-500	34.4	90
RetinaNet-101-800	**37.8**	198
YOLOv3-320	28.6	**20**
YOLOv3-416	31.4	29
YOLOv3-608	33.4	51

Segmentation vs Detection

Pixel-level labels
Category only

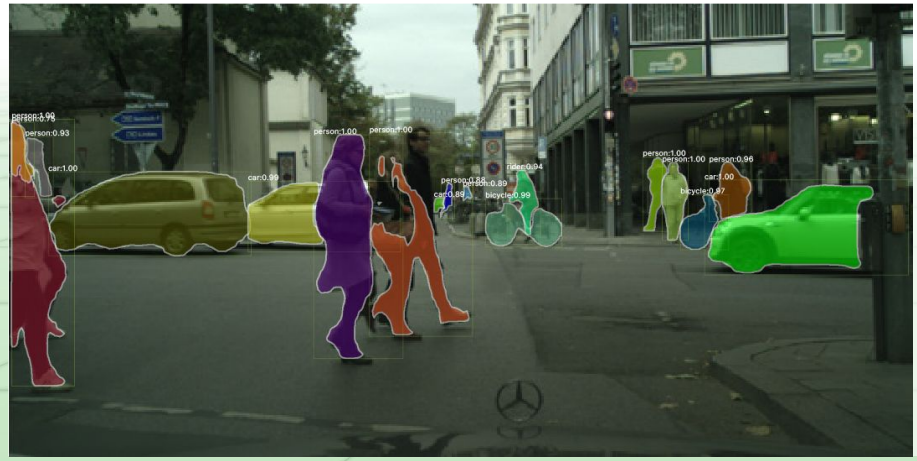
Bounding box labels
Category + instance

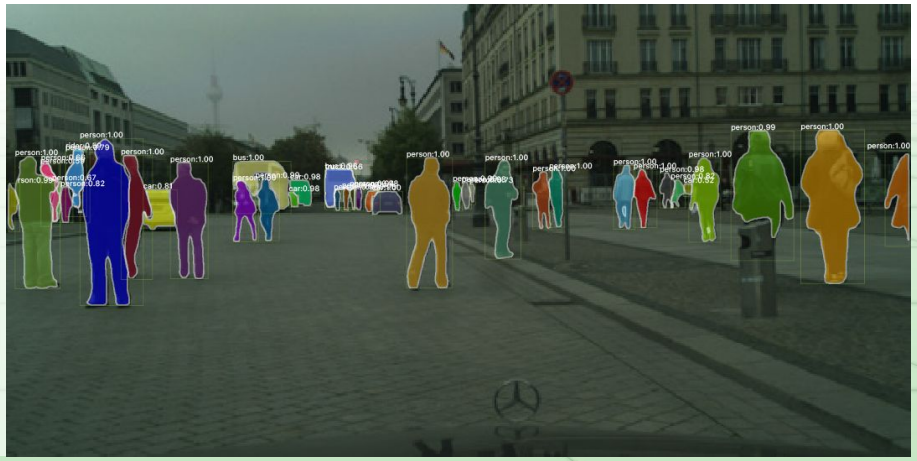


Instance Segmentation

Given an image produce instance-level segmentation

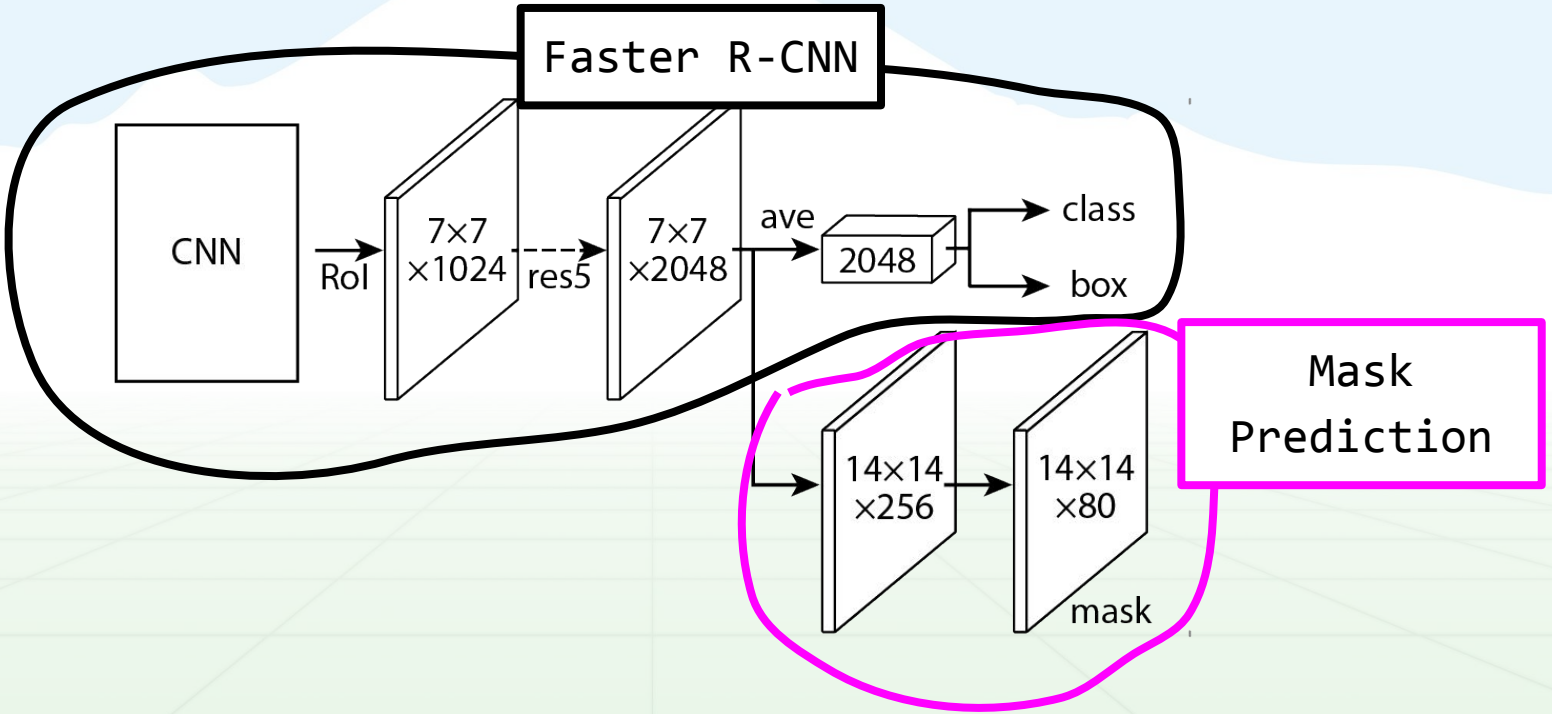
- Which class does each pixel belong to
- Also which instance





Mask R-CNN

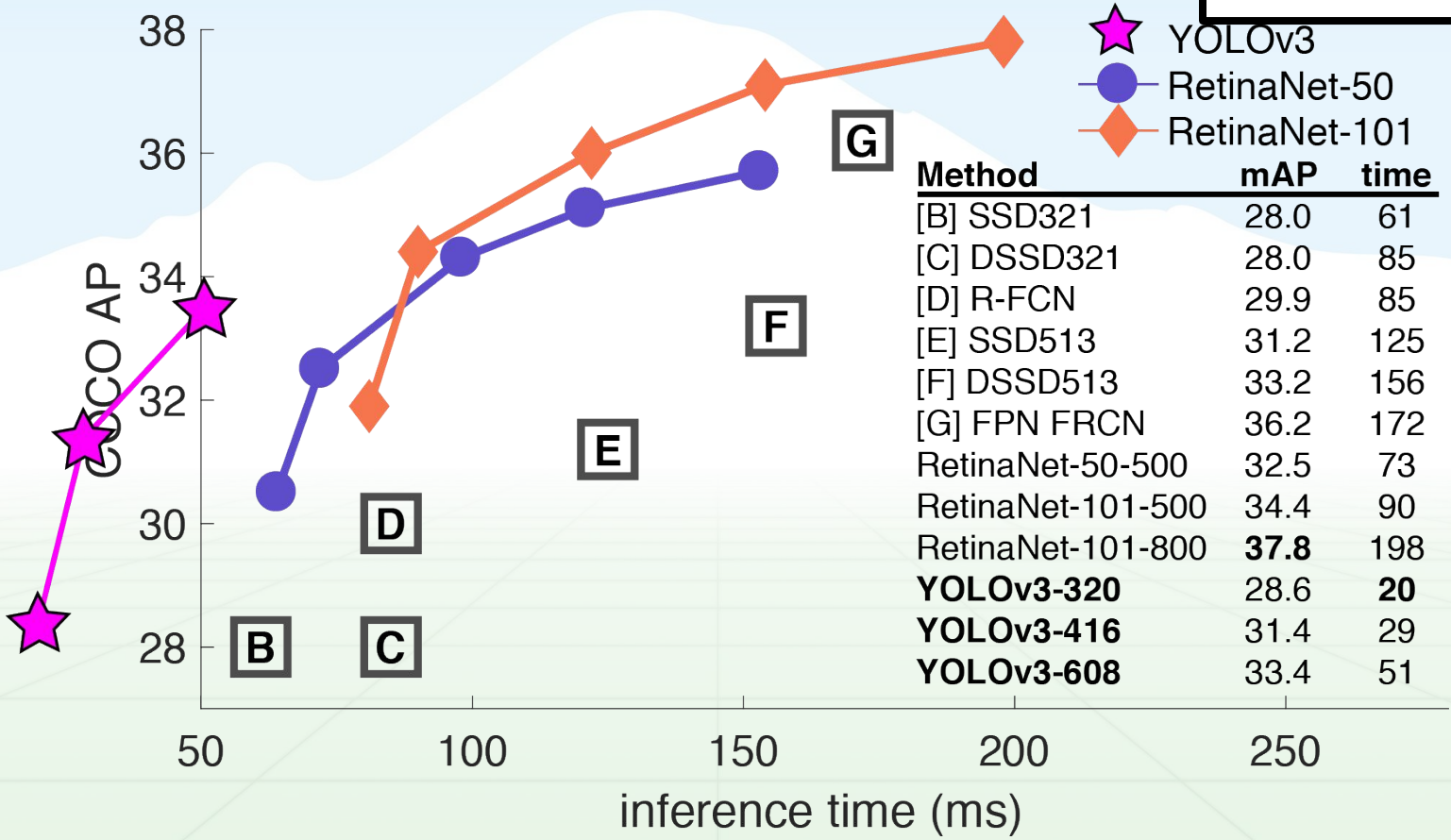
Similar to R-CNN but predict mask *as well as* box



https://arxiv.org/pdf/1703.06870.pdf

Mask R-CNN on COCO: 41 mAP

Mask R-CNN is over here somewhere



Method	mAP	time
[B] SSD321	28.0	61
[C] DSSD321	28.0	85
[D] R-FCN	29.9	85
[E] SSD513	31.2	125
[F] DSSD513	33.2	156
[G] FPN FRCN	36.2	172
RetinaNet-50-500	32.5	73
RetinaNet-101-500	34.4	90
RetinaNet-101-800	**37.8**	198
YOLOv3-320	28.6	**20**
YOLOv3-416	31.4	29
YOLOv3-608	33.4	51


Chapter
Nineteen
Language and Vision

COCO dataset also has captions!

5 captions per image

Detection/segmentation is (maybe) just pattern matching

To caption an image maybe you really have to *understand* it

Need to model both visual information and *language*

The man at bat readies to swing at the pitch while the umpire looks on.

A large bus sitting next to a very tall building.

A horse carrying a large load of hay and two people sitting on it.

Bunk bed with a narrow shelf sitting underneath it.

Natural Language Processing using neural networks

Images are static, language has a component of time

Characters/words appear in sequence, need to read previous words to understand subsequent ones (kind of like frames in a video)

How to we process time-series data using neural networks?

A long time ago in a _____ far, far away....

A long time ago in a galaxy far, far away....

Neural networks + language

Naive approach

Input: string
Output: string

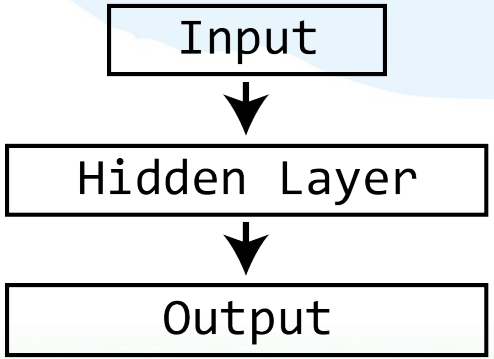
How would we encode?
    ~3000 common words

1-hot encoding of all 6-word
    strings is 18,000 input vector

    What about 7-word or 8-word strings?

    Different networks? What about longer output?

A long time ago in a



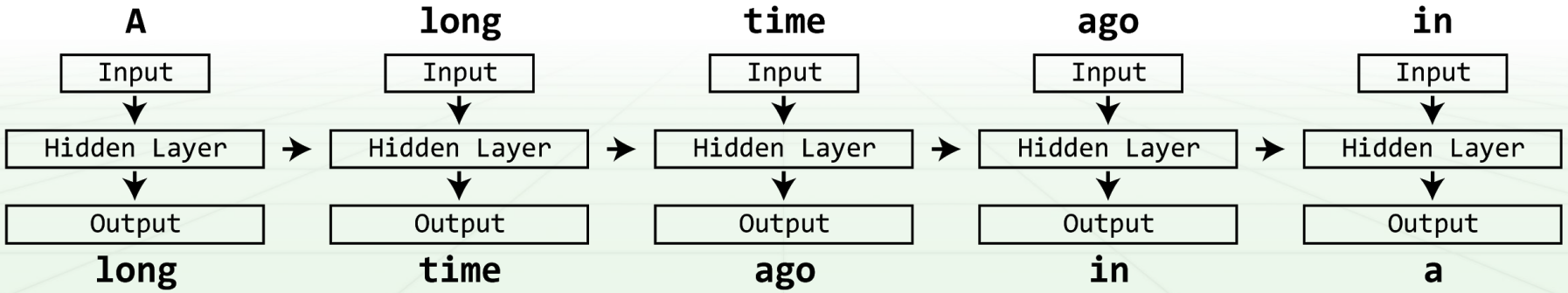
galaxy

Recurrent neural network

Handle sequential data

Idea:

- Read one token (word, character, etc) at a time.
- Produce output
- Also update internal memory

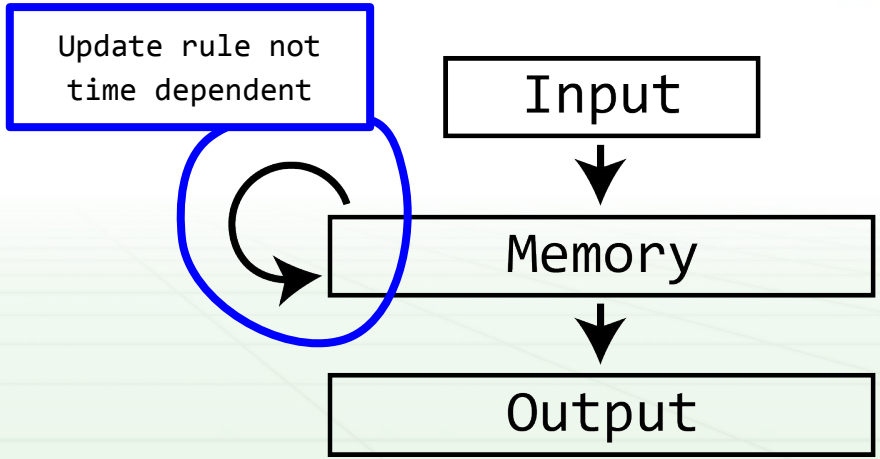


Recurrent neural network

Handle sequential data

Idea:

- Read one token (word, character, etc) at a time.
- Produce output
- Also update internal memory

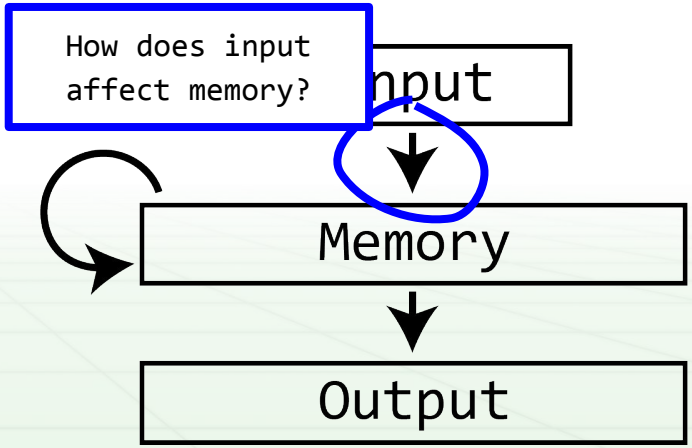


Recurrent neural network

Handle sequential data

Idea:

- Read one token (word, character, etc) at a time.
- Produce output
- Also update internal memory

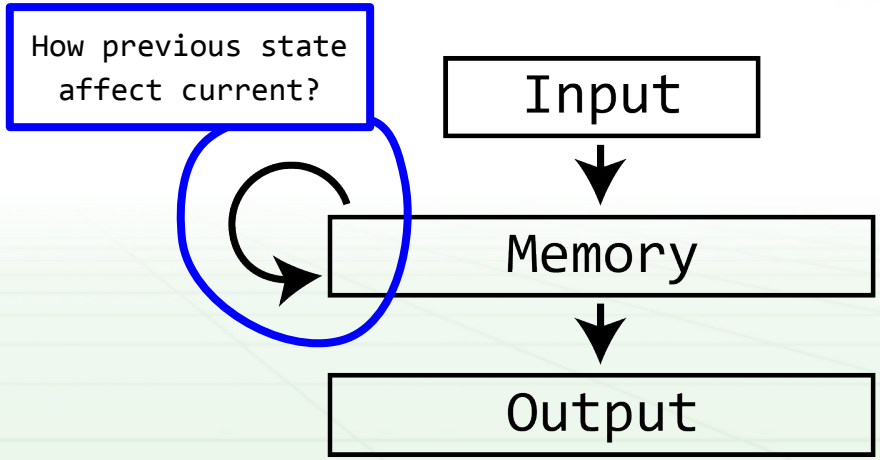


Recurrent neural network

Handle sequential data

Idea:

- Read one token (word, character, etc) at a time.
- Produce output
- Also update internal memory

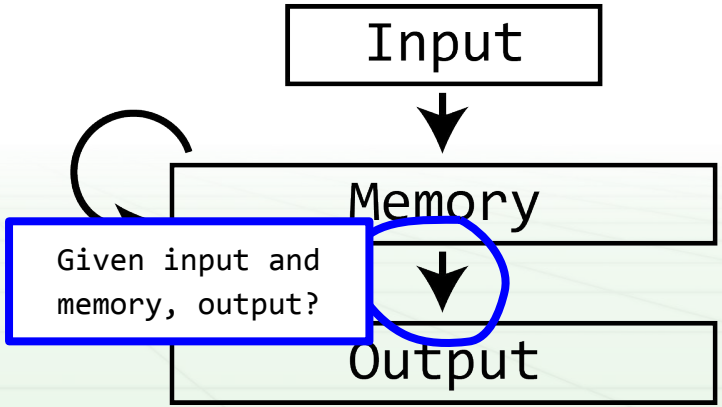


Recurrent neural network

Handle sequential data

Idea:

- Read one token (word, character, etc) at a time.
- Produce output
- Also update internal memory

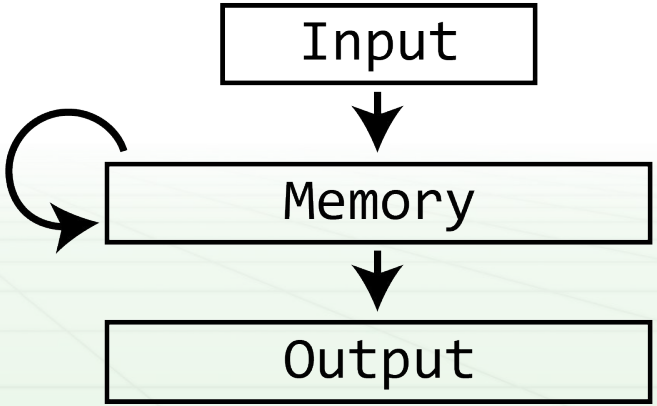


Vanilla RNN

Given input x_t, previous memory h_{t-1}, produce output y_t

$$h_t = \varphi(wx_t + vh_{t-1})$$

$$y_t = h_t$$



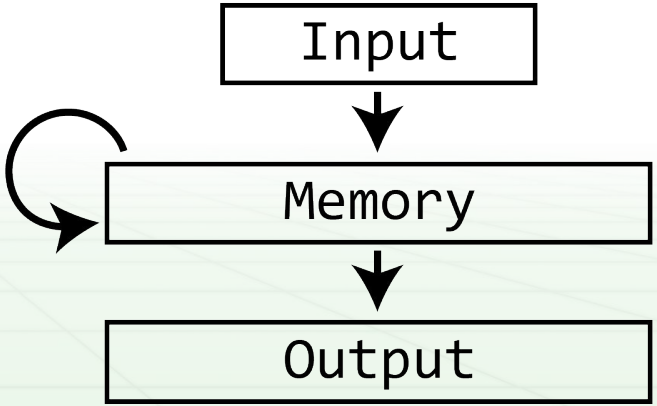
Vanilla RNN

Given input x_t, previous memory h_{t-1}, produce output y_t

$$h_t = \varphi(wx_t + vh_{t-1})$$

$$y_t = h_t$$

Output is same as our current memory



Vanilla RNN

Given input x_t, previous memory h_{t-1}, produce output y_t

$h_t = \varphi(wx_t + vh_{t-1})$

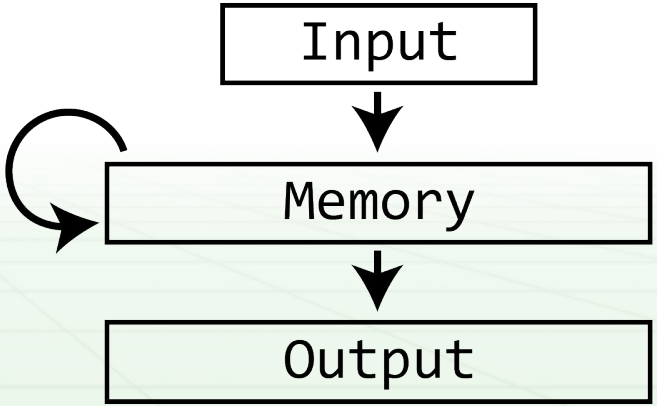
$y_t = h_t$

Output is same as our current memory

w takes input and updates memory

v takes previous memory and updates current

φ is some activation function



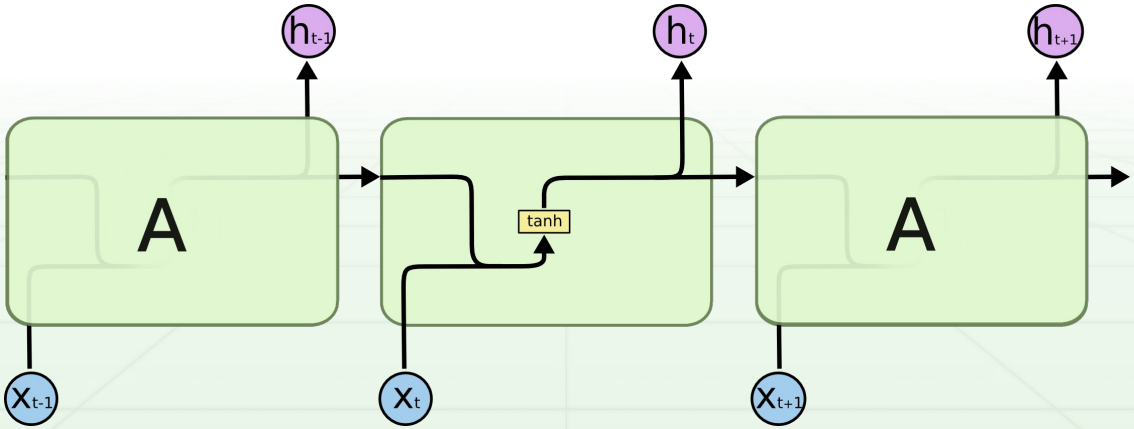
Vanilla RNN

Given input x_t, previous memory h_{t-1}, produce output y_t

In practice, append x_t to h_{t-1} and use one set of weights

$$h_t = \varphi(w \cdot [x_t, h_{t-1}])$$

$$y_t = h_t$$



http://colah.github.io/posts/2015-08-Understanding-LSTMs

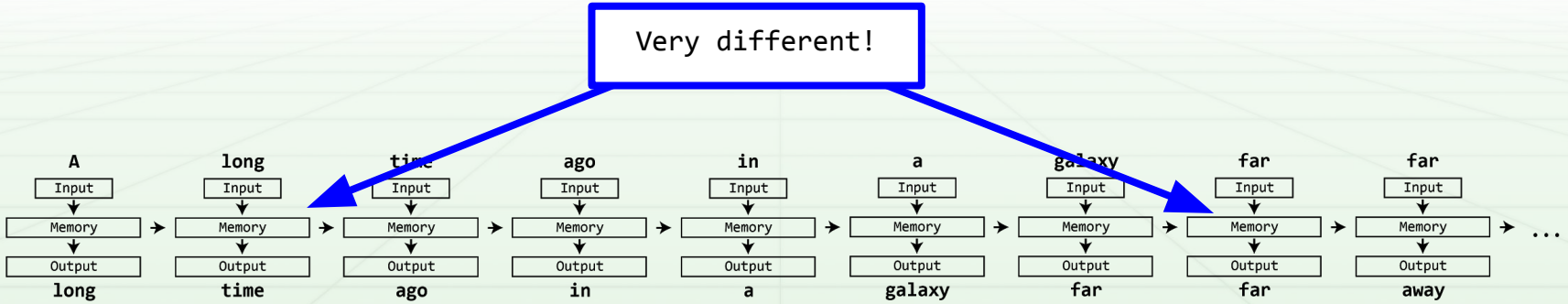
Vanilla RNN

Given input x_t, previous memory h_{t-1}, produce output y_t

$$h_t = \varphi(w \cdot [x_t, h_{t-1}])$$

$$y_t = h_t$$

Problem: long-term dependence, memory computed from scratch every round, hard to remember things for a long time.



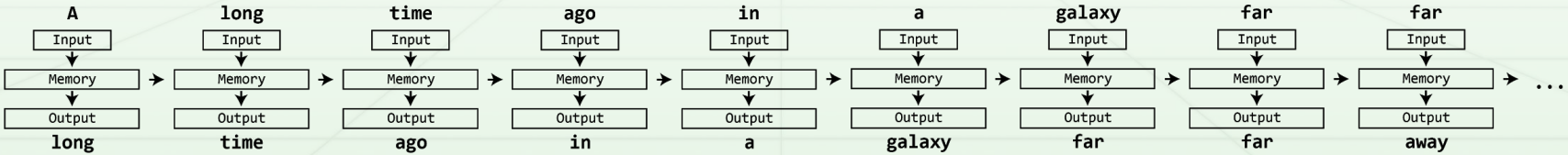
Idea: incremental change to memory

Instead of completely re-doing memory every round, write to and read from memory like a computer (sort of)

Calculate a gating function to decide what parts of memory to keep and what to change

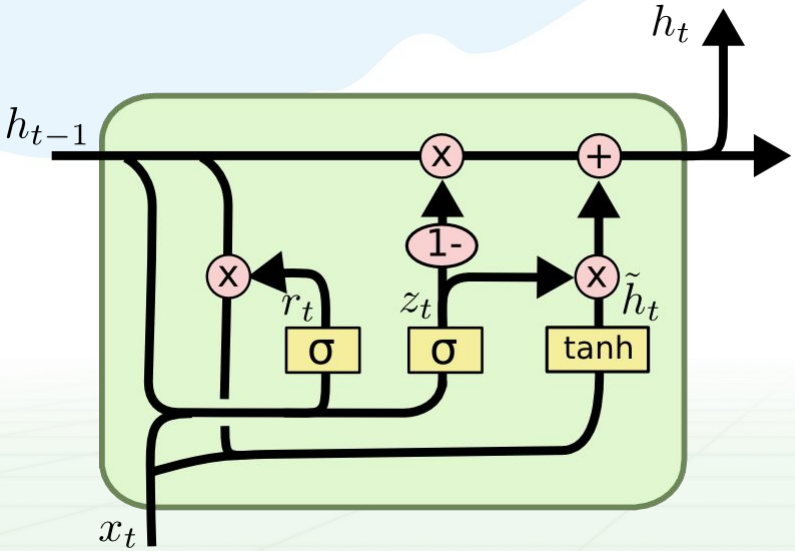
This allows us to make changes but also remember very important things for a while. Different options:

GRU, LSTM, ...



GRU: Gated recurrent units

OK there's a WHOLE lot going on here...



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

https://arxiv.org/pdf/1406.1078.pdf, http://colah.github.io/posts/2015-08-Understanding-LSTMs/

GRU: Gated recurrent units

Steps in a GRU

Look at memory and input, decide what part of memory is important

Maybe you weren't certain which way a sentence would go but now you know, can forget other options

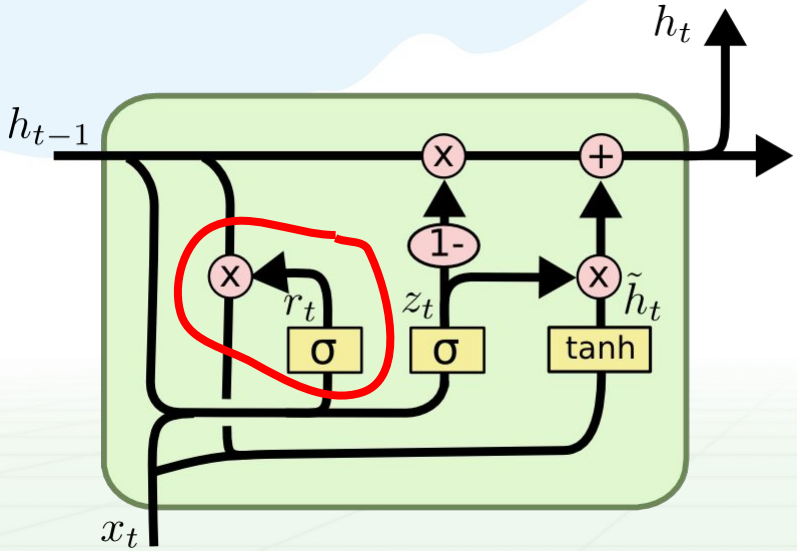
Calculate an update to memory

Decide what parts of memory to keep and what to update

Output is old memory you kept + the update you calculated

https://arxiv.org/pdf/1406.1078.pdf, http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Reset gate: ignore some memory



$$z_t = \sigma\left(W \right.$$

Figure out what parts of memory to pay attention to, what to ignore

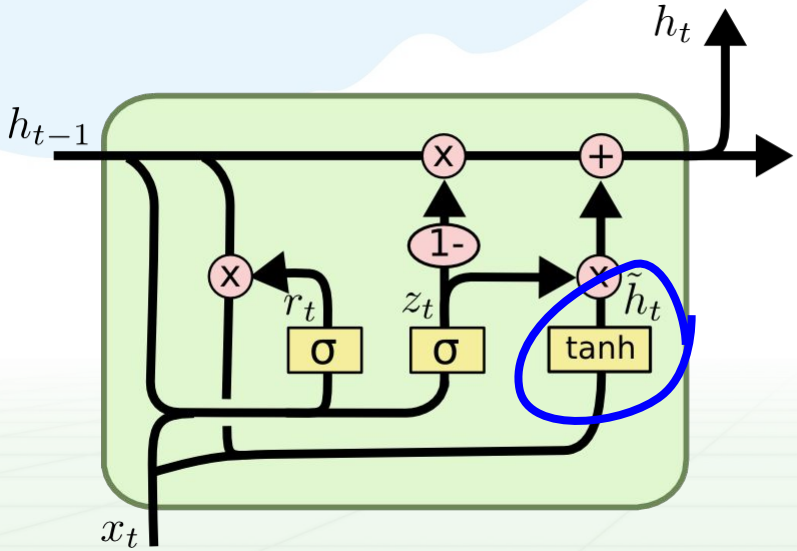
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

https://arxiv.org/pdf/1406.1078.pdf, http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Calculate update



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W\right.$$

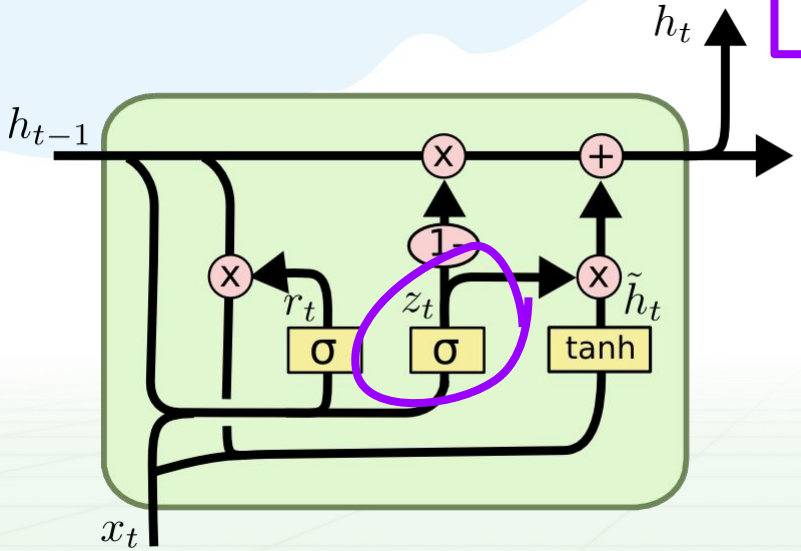
Calculate update using input and the important parts of memory

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

https://arxiv.org/pdf/1406.1078.pdf, http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Update gate: what to save/update



Calculate what parts of memory to save, what to replace

$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

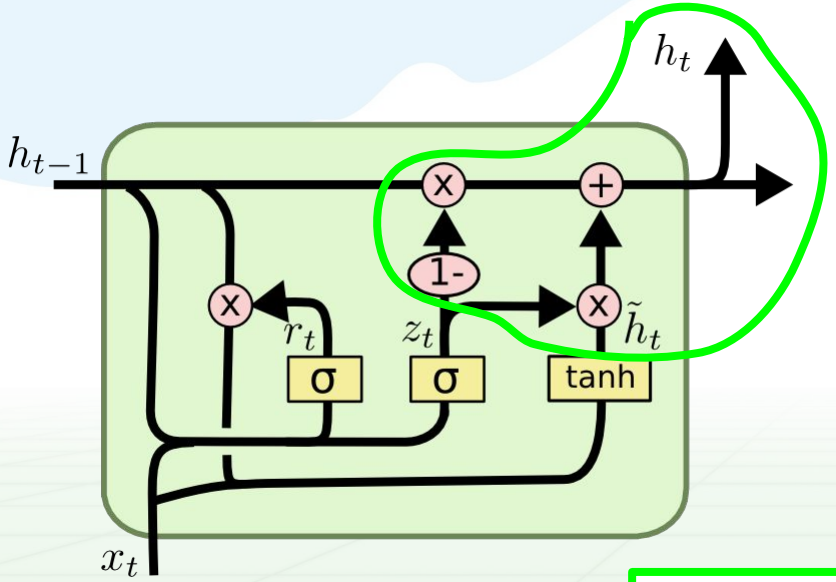
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

https://arxiv.org/pdf/1406.1078.pdf, http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Output: weighted sum



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

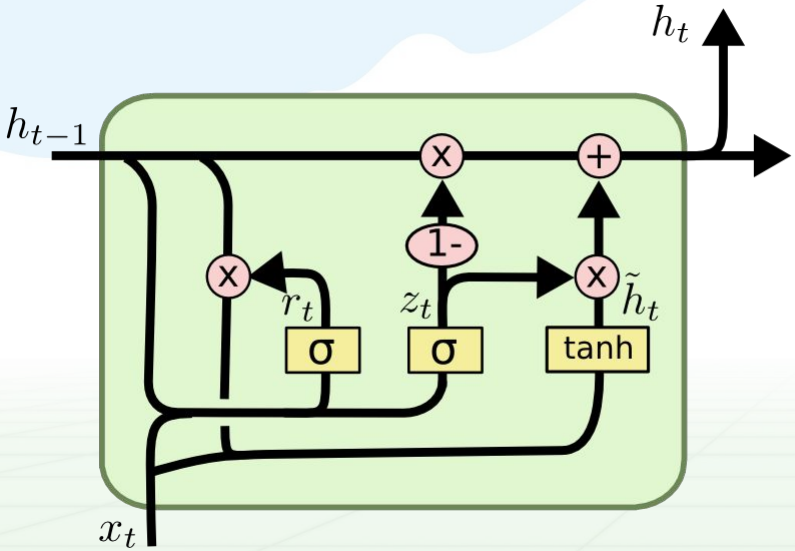
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Output is weighted sum of previous output and “new” output

https://arxiv.org/pdf/1406.1078.pdf, http://colah.github.io/posts/2015-08-Understanding-LSTMs/

GRU: Gated recurrent units

OK there's a WHOLE lot going on here...



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

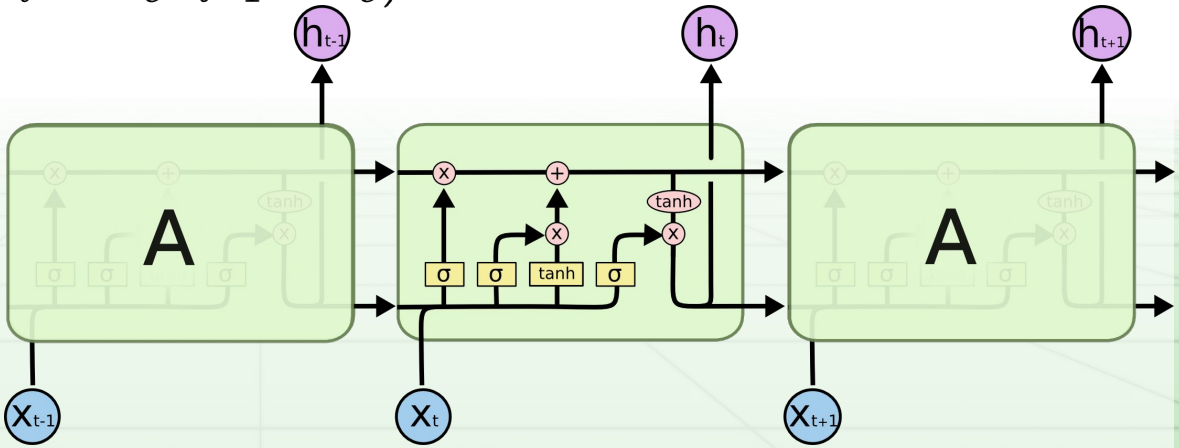
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

https://arxiv.org/pdf/1406.1078.pdf, http://colah.github.io/posts/2015-08-Understanding-LSTMs/

LSTM: Long short-term memory

What does that even mean??

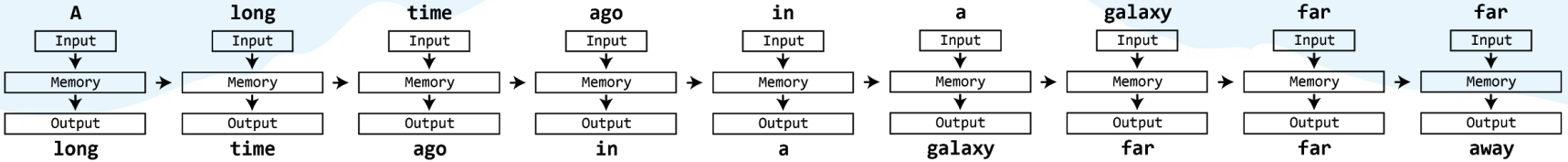
$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f)$$
$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i)$$
$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o)$$
$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c)$$
$$h_t = o_t \circ \sigma_h(c_t)$$



http://www.bioinf.jku.at/publications/older/2604.pdf, http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Language modeling: what's next

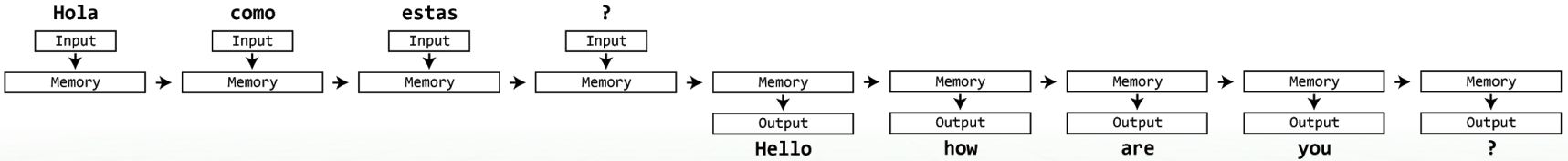
Given a string of words/characters, what's the next one



Demo!

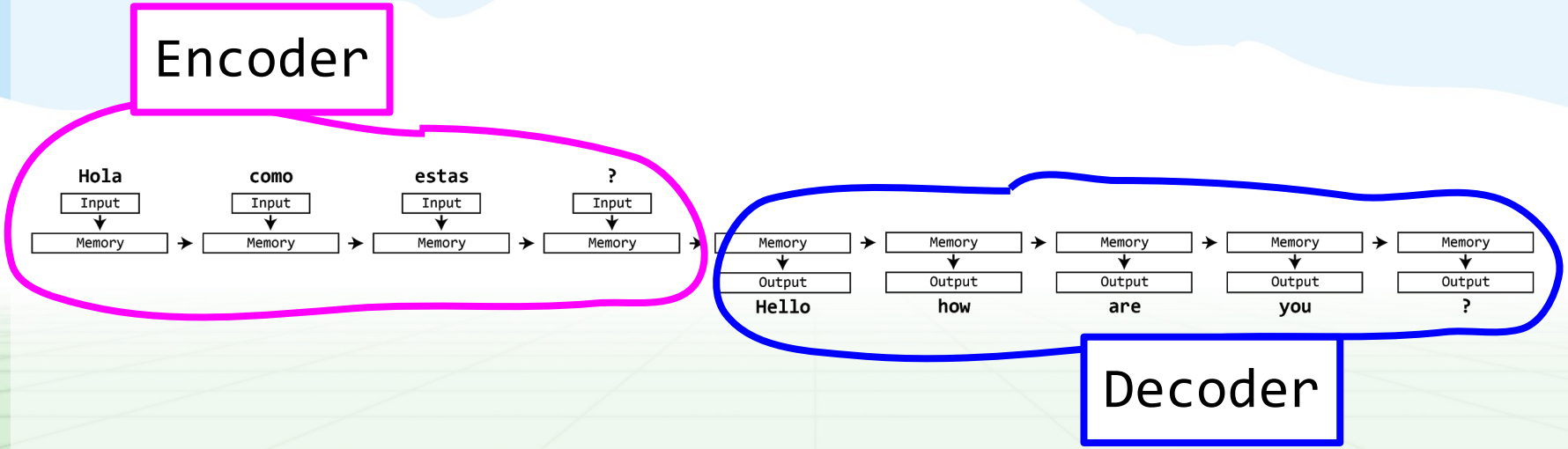
Translation and Q/A

Given a string of words/characters, what's the response?



Translation and Q/A

Given a string of words/characters, what's the response?



Translation and Q/A

Given a string of words/characters, what's the response?

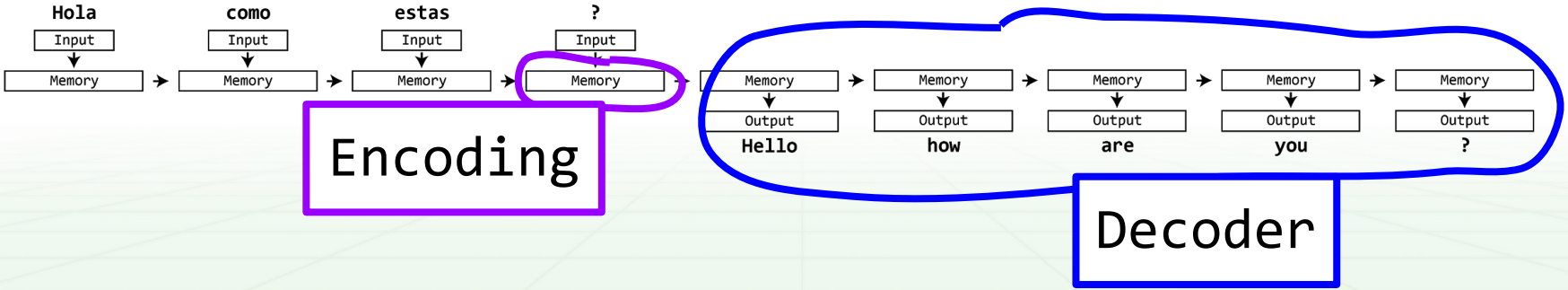
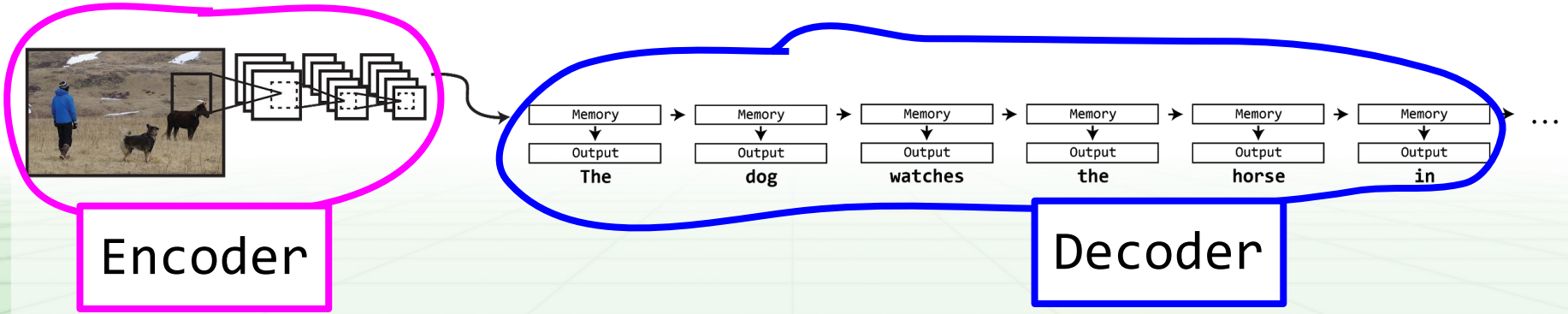


Image captioning

Given an image:

- Extract features using CNN
- Feed into RNN
- Generate sentences!



The Good

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

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

https://cs.stanford.edu/peopl

The Bad

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

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

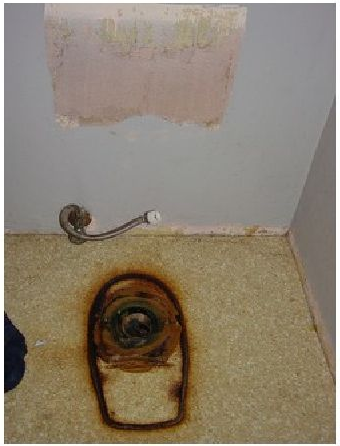
a cat is sitting on a couch with a remote control
logprob: -12.45

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

The Huh?



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

https://cs.stanford.edu/people/karpathy/sfmltalk.pdf

Not all it’s cracked up to be

Many methods seem to be glorified nearest-neighbor

Dataset is really large so often performs well

But there’s another problem, how do we know what a “good” caption even is?? Not like scoring detection or classification!

Automated scoring, BLEU, METEOR, etc.

“Demo” scoring methods

https://arxiv.org/pdf/1505.04467.pdf

Visual Question Answering

Given image and question...

Answer it!

Harder than captioning?

Requires more understanding?

Easier to evaluate!

More demo!

What is the mustache made of?

AI System

bananas

http://www.visualqa.org/challenge.html