
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.

Strike

I (and many TAs probably) will still come to poster session, I’m excited to see posters! But I won’t be grading them...

College of Engineering Dean:

coeinfo@uw.edu

(206) 543-0340

Ana Mari Cauce:

pres@uw.edu

(206) 543-5010


Previously
On
the
Ancient Secrets
of Computer 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.

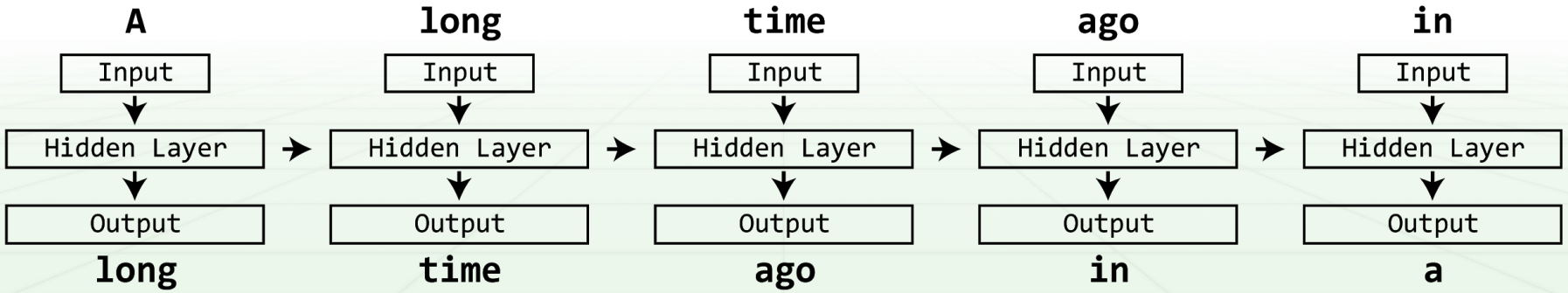
Bunk bed with a narrow shelf sitting underneath it.

Recurrent neural network

Handle sequential data

Idea:

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



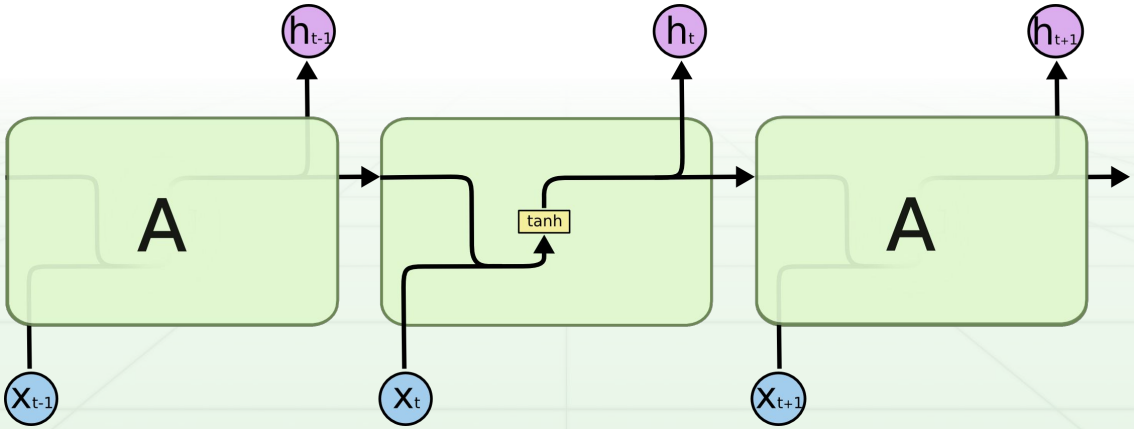
Vanilla RNN

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

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

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

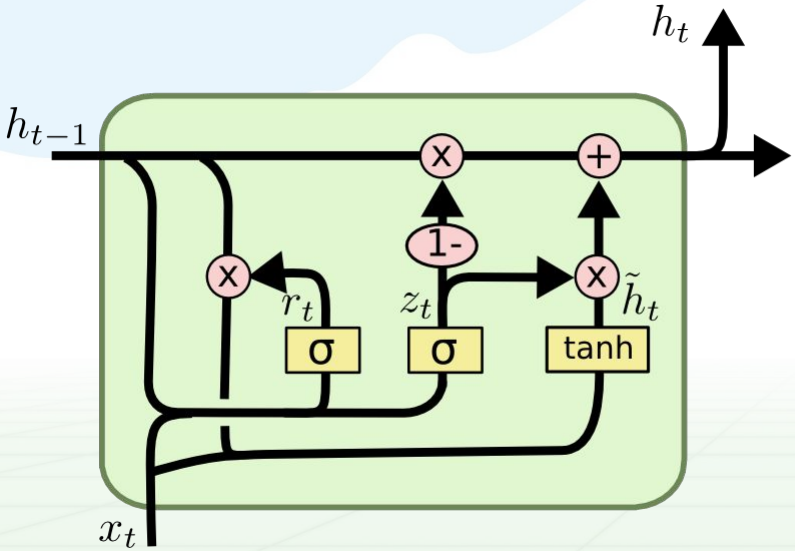
$$y_t = h_t$$



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

GRU: Gated recurrent units

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



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

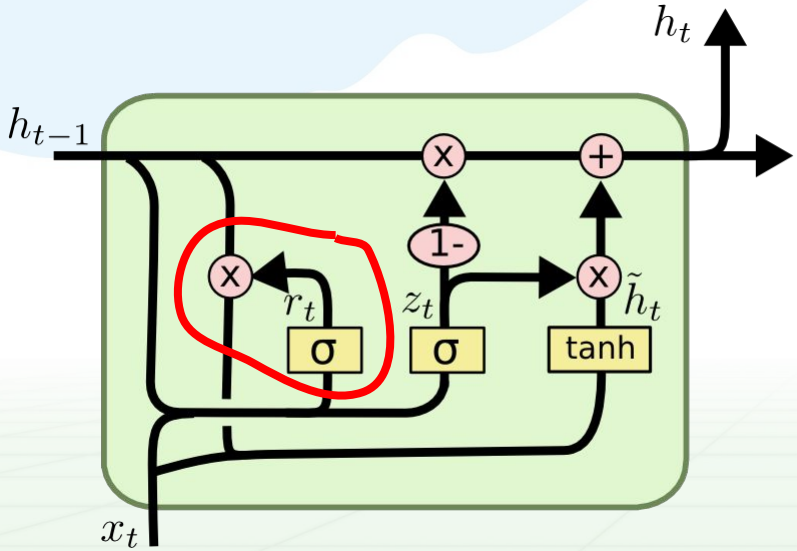
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

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

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

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

Reset gate: ignore some memory



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

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

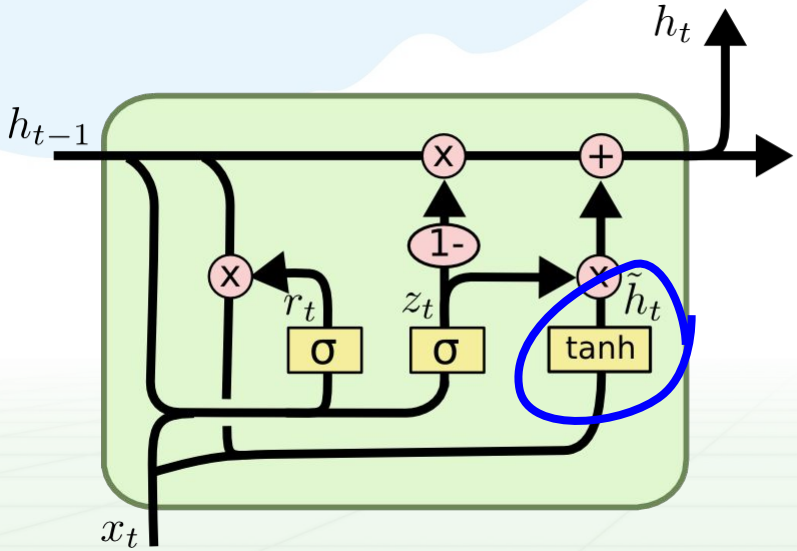
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

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

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

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

Calculate update



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

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

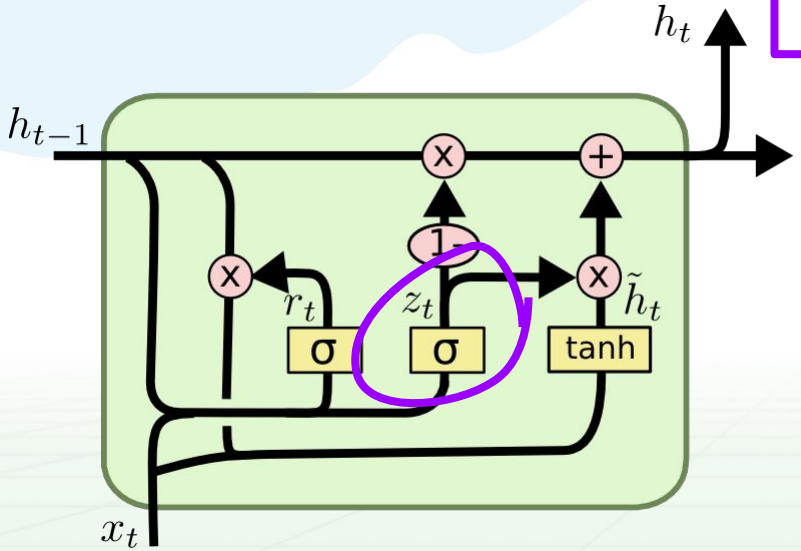
Calculate update using input and the important parts of memory

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

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

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

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

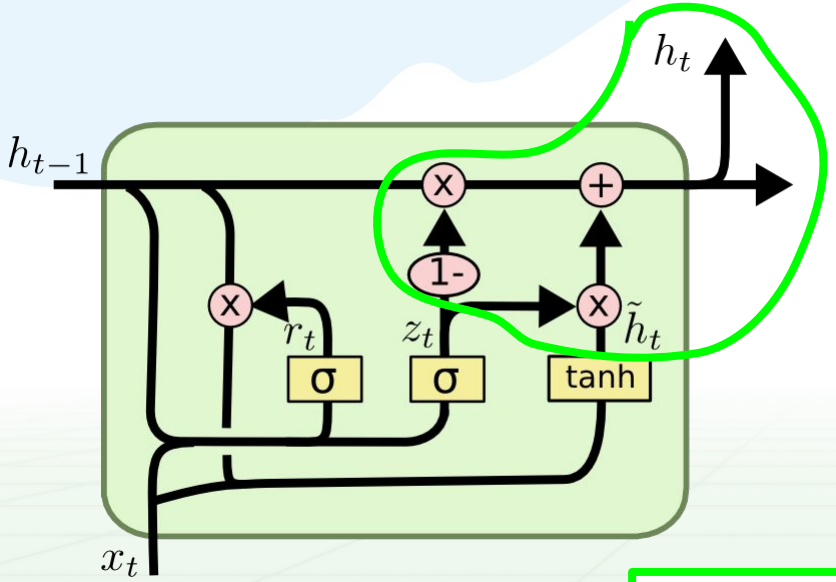
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

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

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

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

Output: weighted sum



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

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

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

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

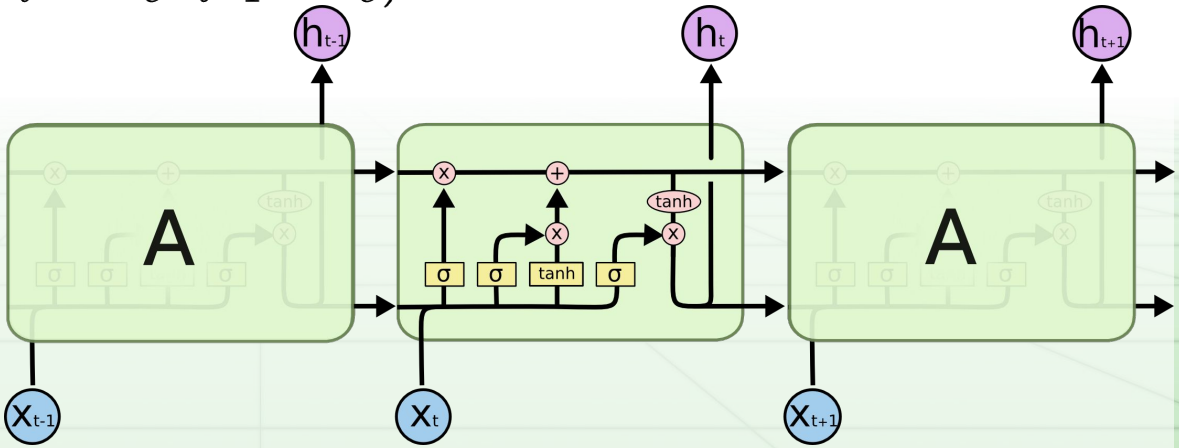
Output is weighted sum of previous output and “new” output

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

LSTM: Long short-term memory

What does that even mean??

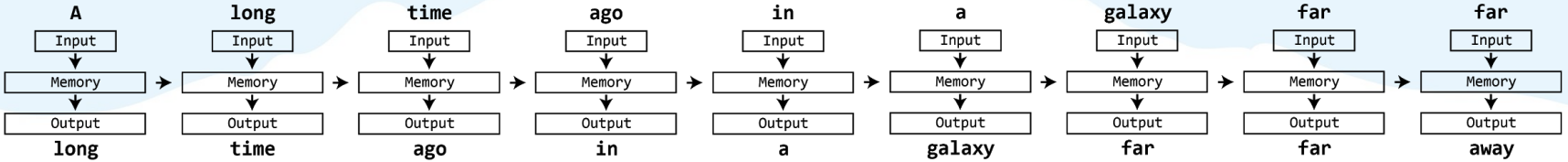
$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f)$$
$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i)$$
$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o)$$
$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c)$$
$$h_t = o_t \circ \sigma_h(c_t)$$



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?

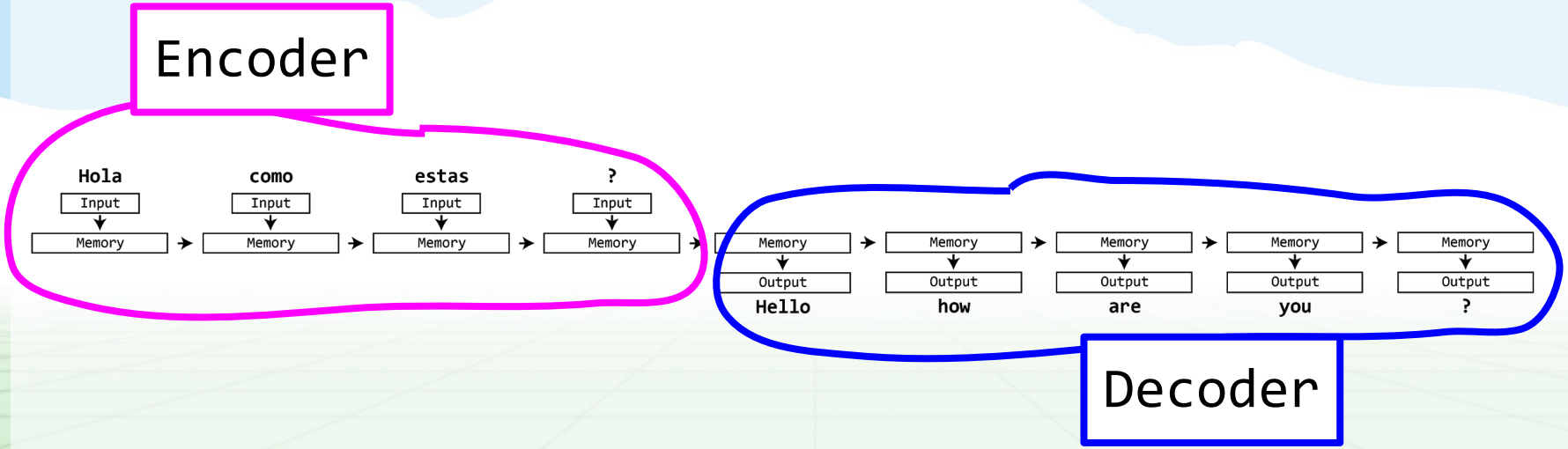
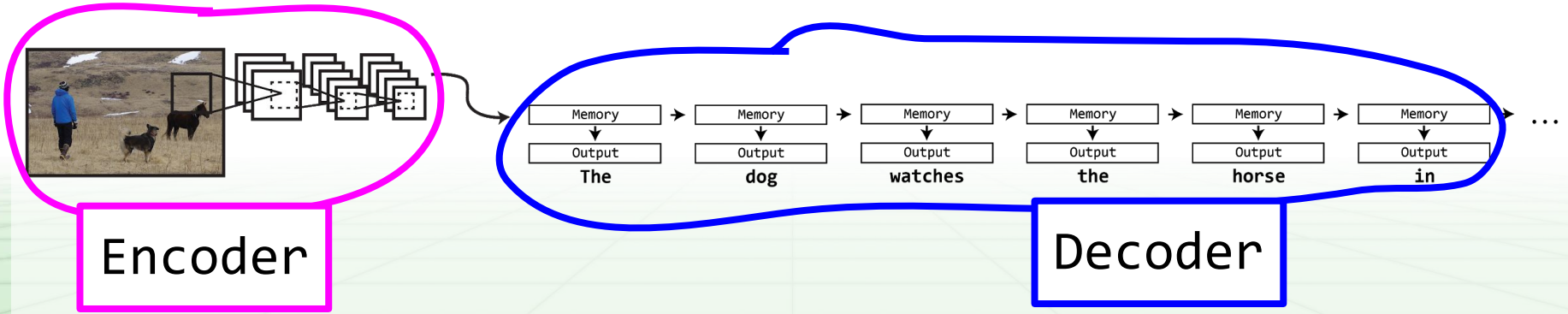


Image captioning

Given an image:

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



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

Situation Recognition

Images often have one main thing going on, one verb

Recognize that verb and what sense it's being used in, fill in the other important objects and how they relate in a linguistic *frame*

CLIPPING

ROLE	VALUE
AGENT	MAN
SOURCE	SHEEP
TOOL	SHEARS
ITEM	WOOL
PLACE	FIELD

ROLE	VALUE
AGENT	VET
SOURCE	DOG
TOOL	CLIPPER
ITEM	CLAW
PLACE	ROOM

JUMPING

ROLE	VALUE
AGENT	BOY
SOURCE	CLIFF
OBSTACLE	-
DESTINATION	WATER
PLACE	LAKE

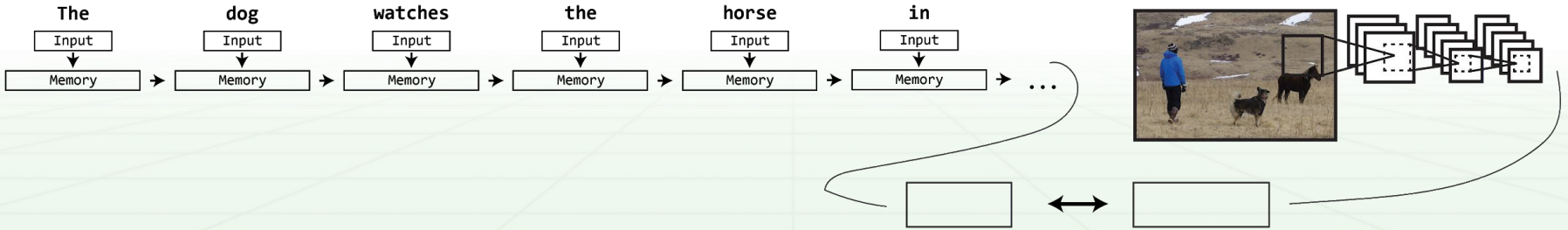
ROLE	VALUE
AGENT	BEAR
SOURCE	ICEBERG
OBSTACLE	WATER
DESTINATION	ICEBERG
PLACE	OUTDOOR

http://imsitu.org/

Image retrieval

Given a sentence:

- Extract representation using RNN
- Find matching representations from images processed with CNN




Chapter
Twenty
Generative Adversarial
Networks

Previously on neural networks...

We've covered a lot of techniques for solving vision problems with neural networks.

Image classification

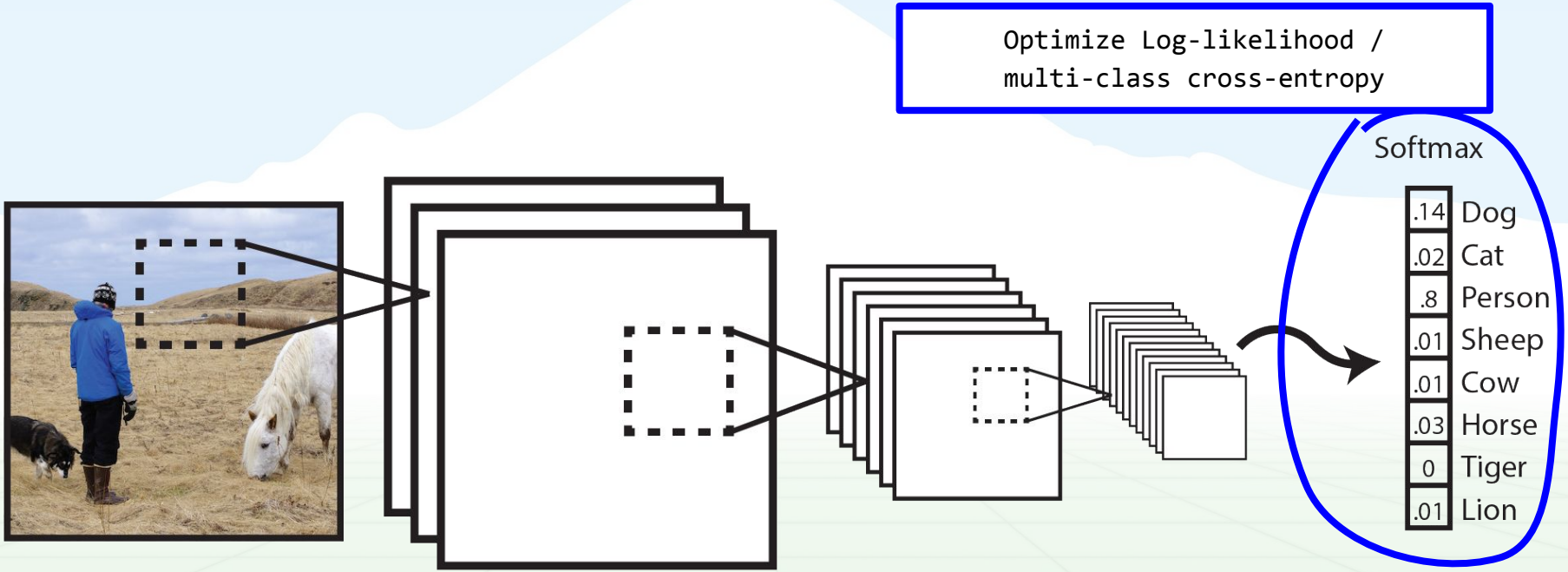
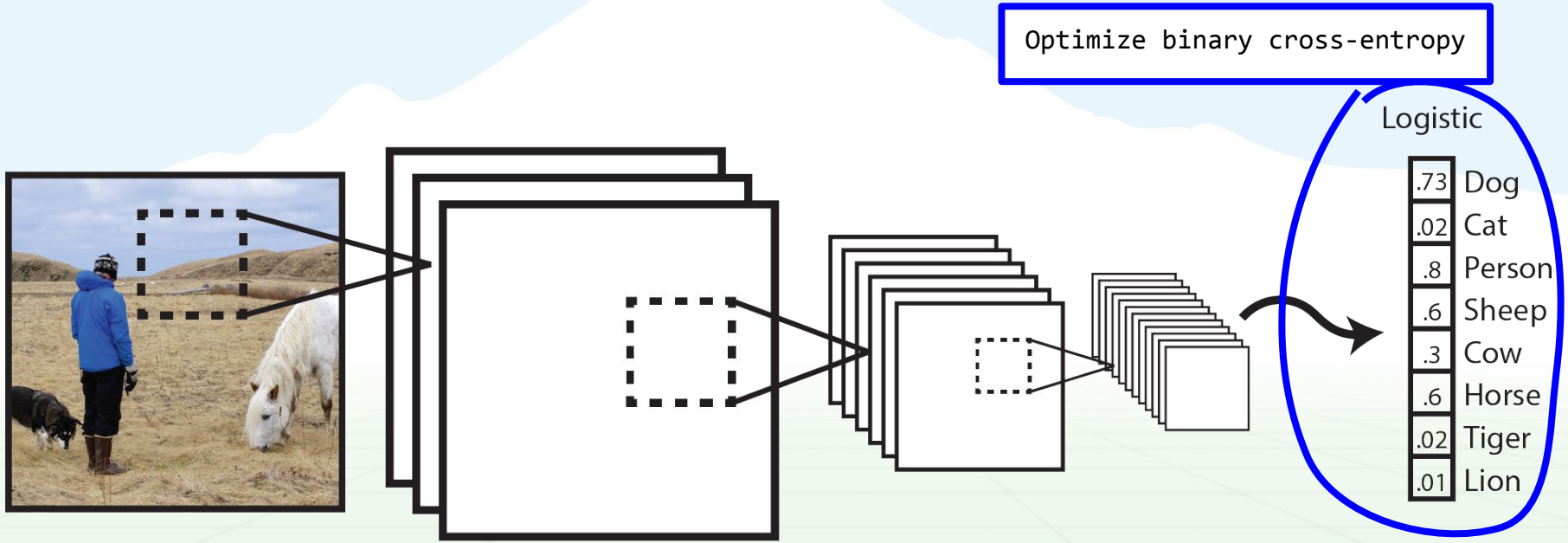
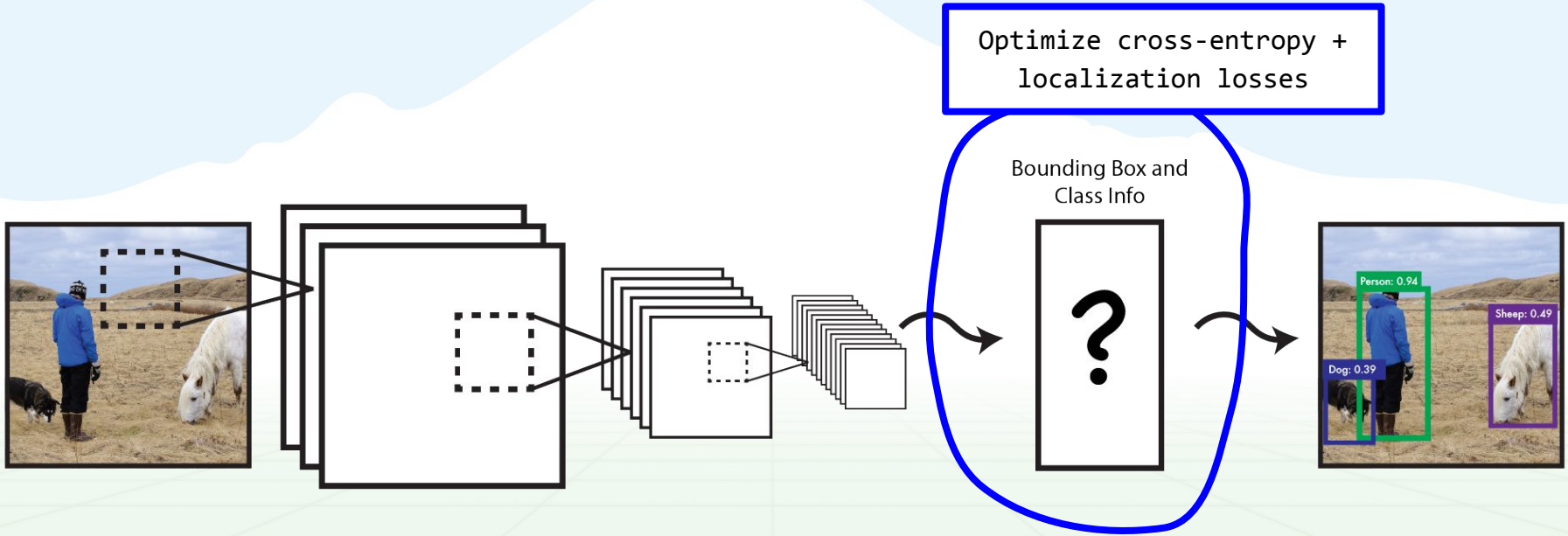


Image tagging



Object detection



Semantic segmentation

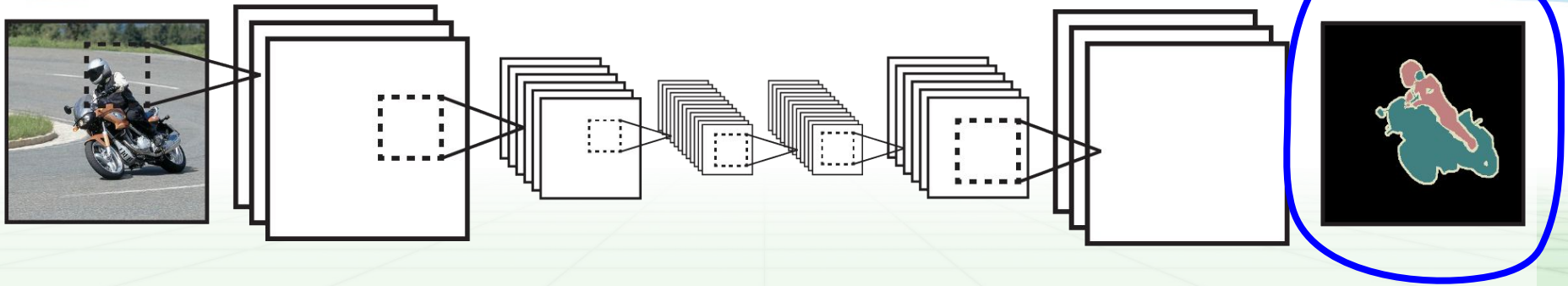
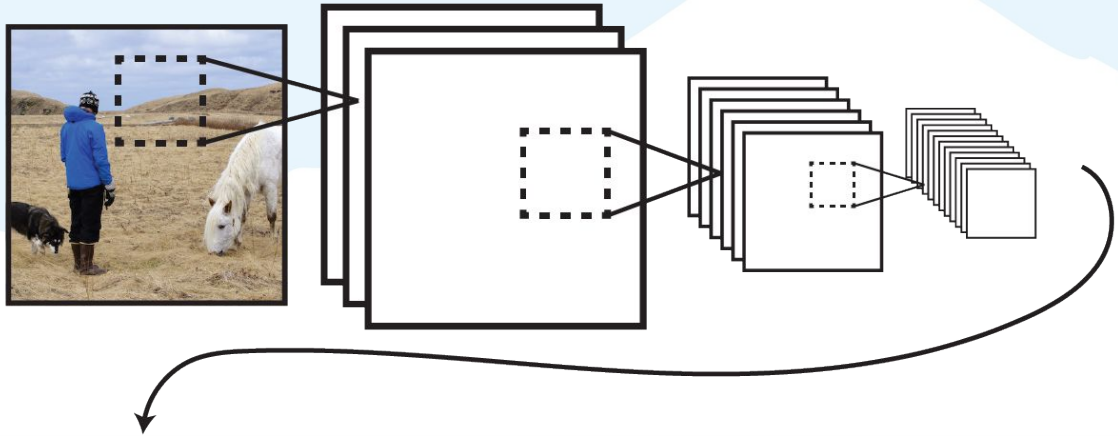
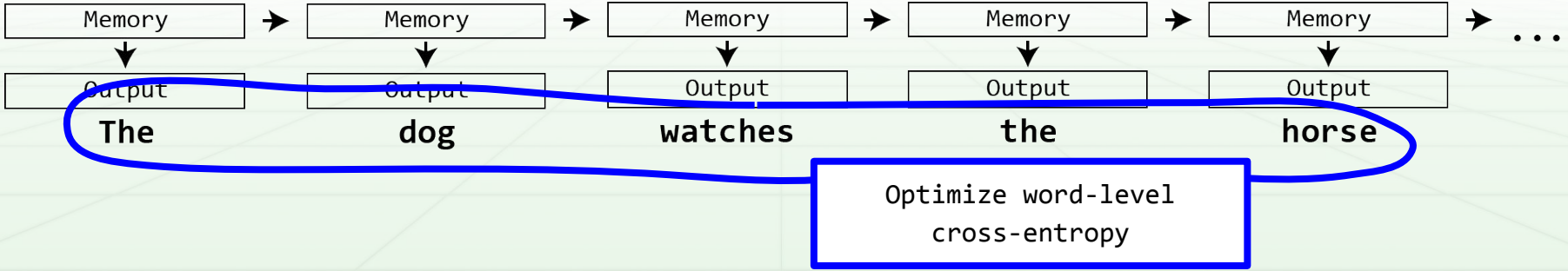


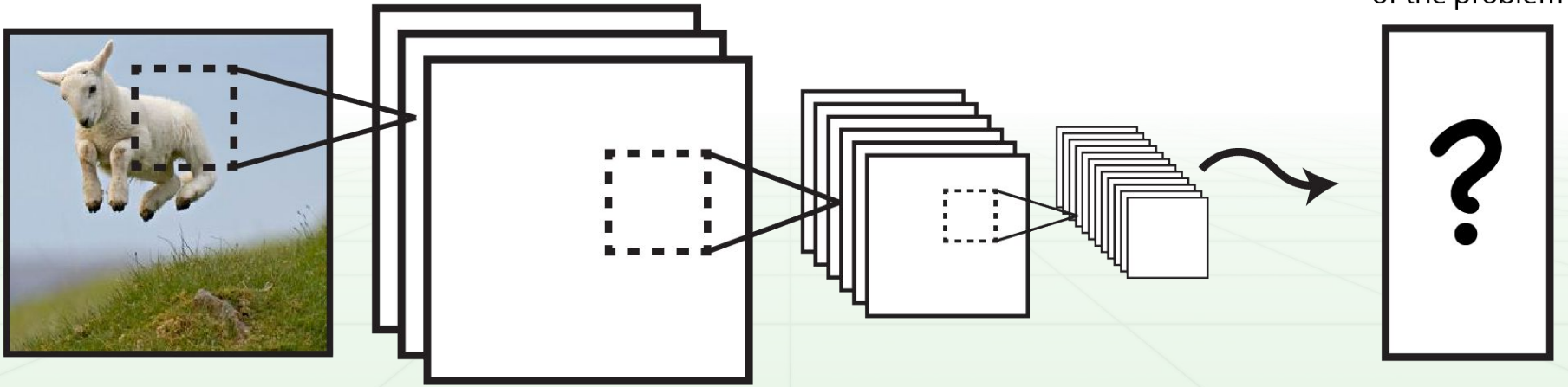
Image captioning





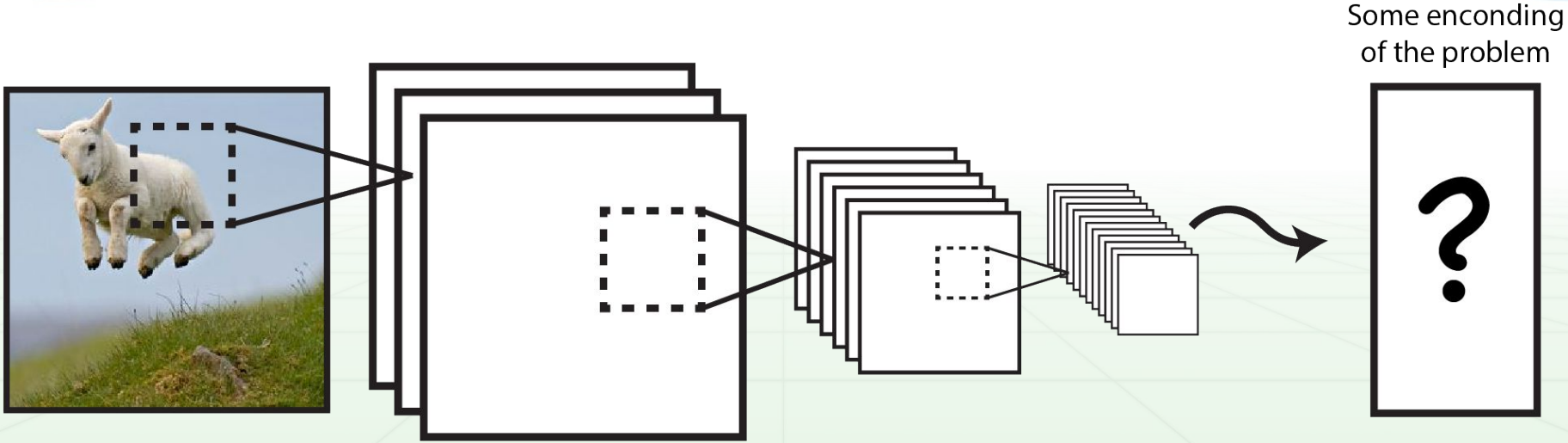
How to solve any vision problem

1. Pick a neural network architecture
2. Design an encoding of the expected output
3. Pick a loss function for that encoding (squared error? Log-likelihood?)
4. Gather a bunch of training data (and label it)
5. Train your network with backpropagation for a long time



How to solve any vision problem

1. Pick a neural network architecture
2. Design an encoding of the expected output
3. Pick a loss function for that encoding (squared error? Log-likelihood?)
4. Gather a bunch of training data (and label it)
5. Train your network with backpropagation for a long time



How to solve any vision problem

1. Pick a neural network architecture
2. Design an encoding of the expected output
3. Pick a loss function for that encoding (squared error? Log-likelihood?)
4. Gather a bunch of training data (and label it)
5. Train your network with backpropagation for a long time

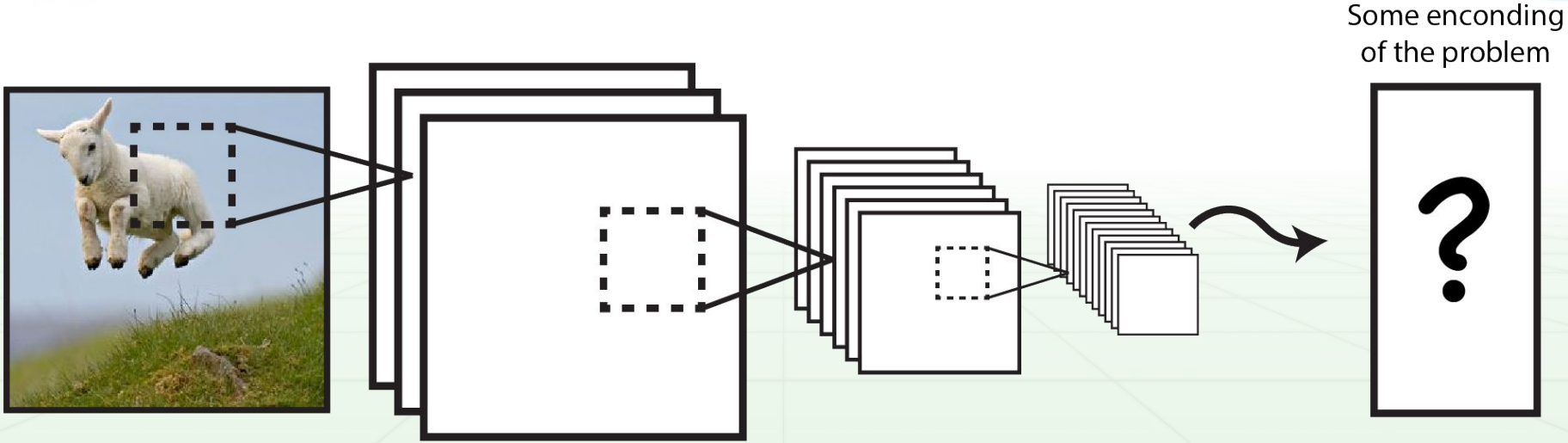


Image colorization

Training is easy, grayscale a bunch of images, try to predict the original one!

But there’s a problem...

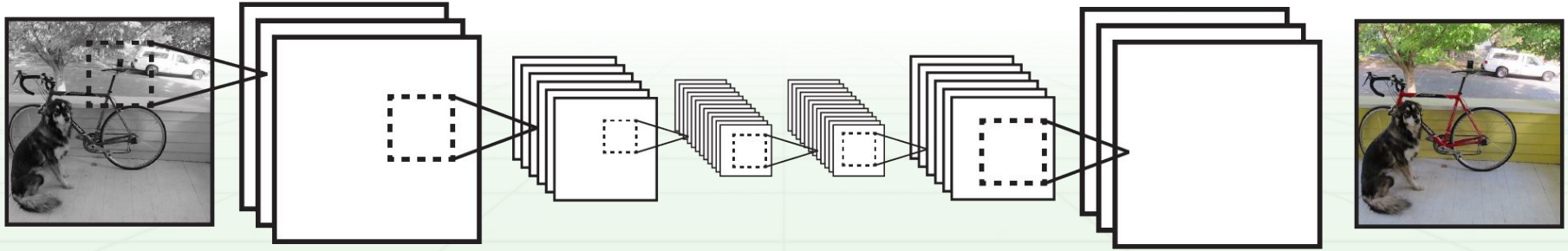


Image colorization

Training is easy, grayscale a bunch of images, try to predict the original one!

But there’s a problem...

Image colorization

Training is easy, grayscale a bunch of images, try to predict the original one!

But there’s a problem...

Image colorization

Training is easy, grayscale a bunch of images, try to predict the original one!

But there's a problem…

Big difference between green and red according to L_2 loss, perceptually we don't really care

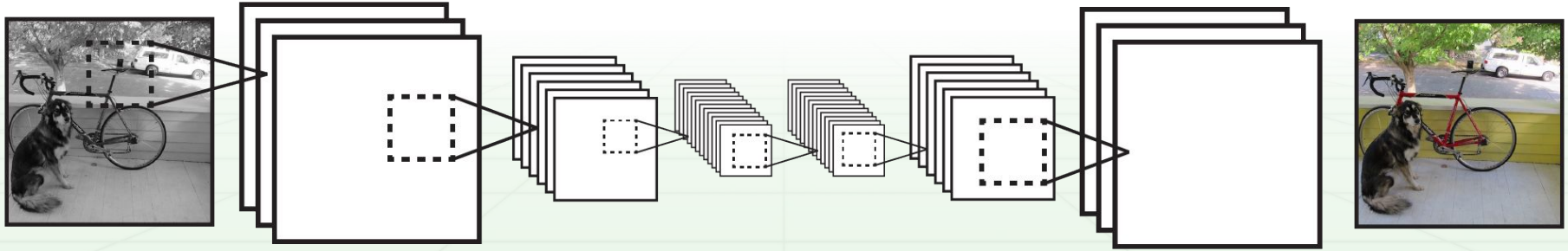
Don't want prediction to be "right" we just want it to look good!

We want a perceptual loss function

What we really want is… does the image look good?

And if not, how do we change it to look better!

Well, we just spent a long time learning how to train neural networks to do stuff...



Course Evaluation

Please fill out!

https://uw.iasystem.org/survey/194105

Shortened:

https://goo.gl/1RTteX

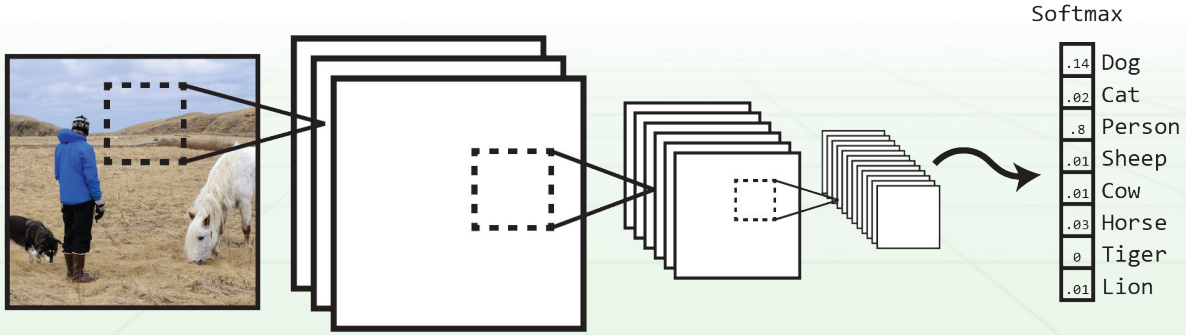
Also on website!

We want a perceptual loss function

What we really want is… does the image look good?

And if not, how do we change it to look better!

Well, we just spent a long time learning how to train neural networks to do stuff...

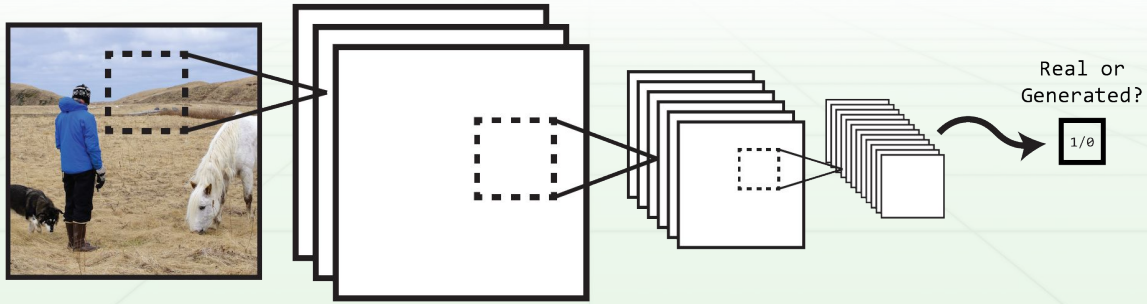


We want a perceptual loss function

What we really want is… does the image look good?

And if not, how do we change it to look better!

Well, we just spent a long time learning how to train neural networks to do stuff...

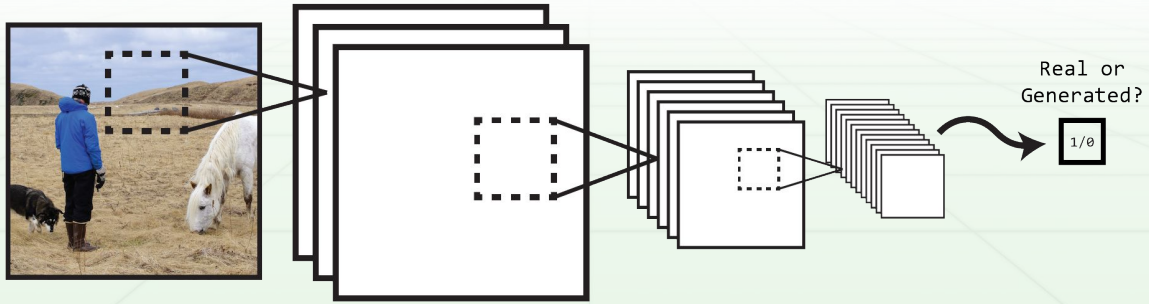


Discriminator network

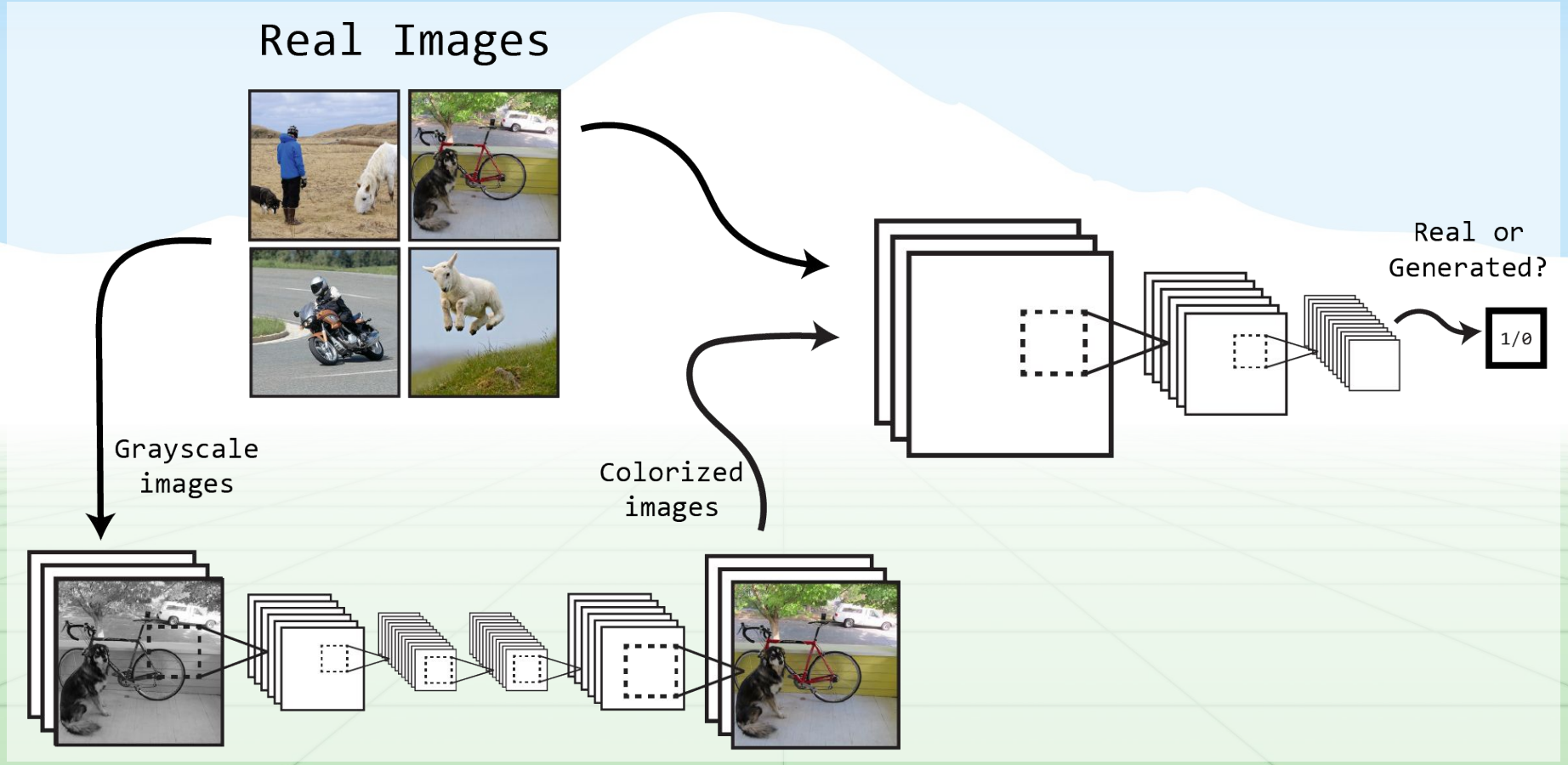
Trained to tell apart real and generated images

How do we make an image more “real”?

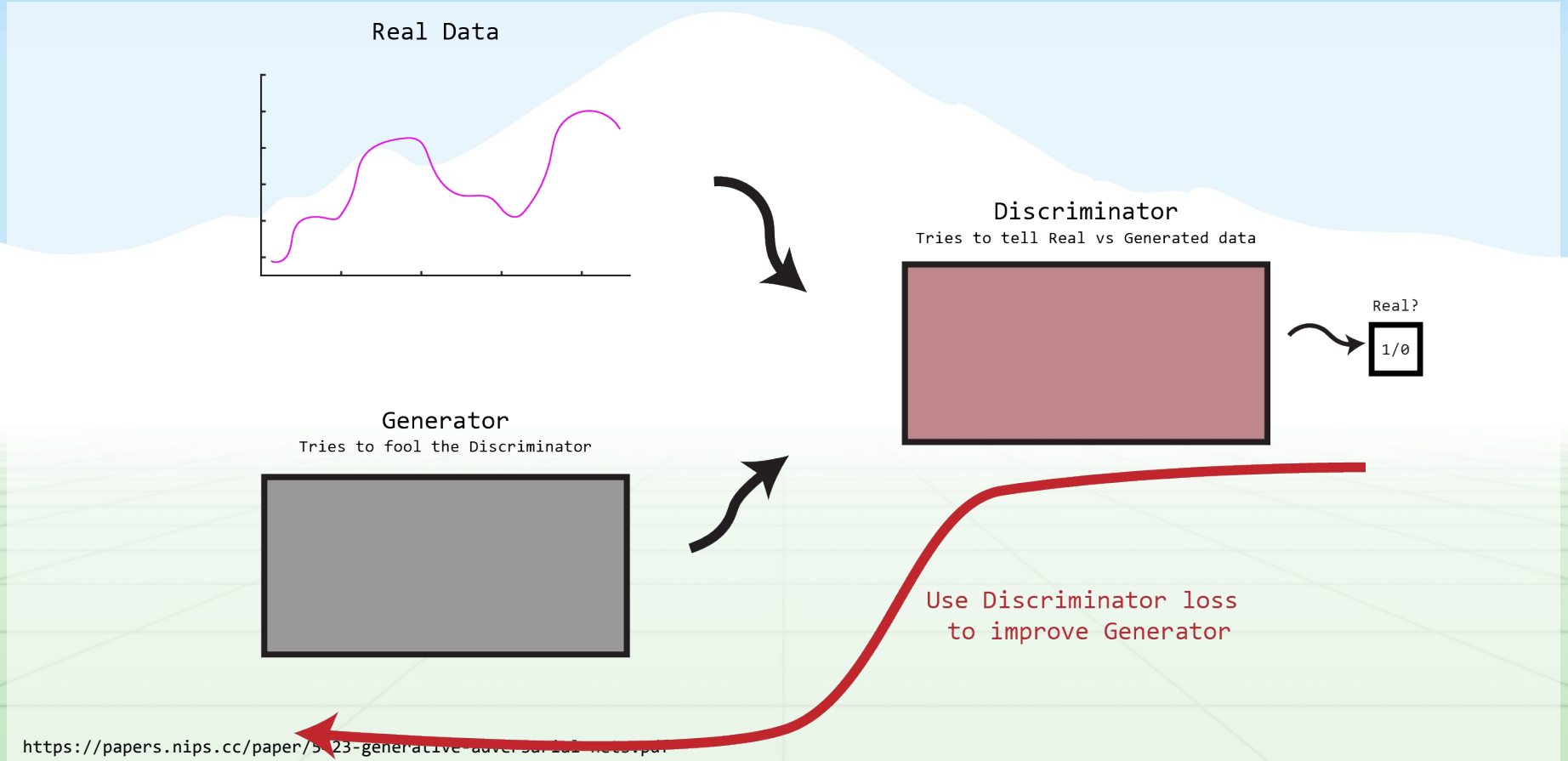
- Run image through
- Calculate loss with real = 1
- Backpropagate loss through the network to the image itself
- We get “error” for the image that would make it more real!



Generative adversarial network

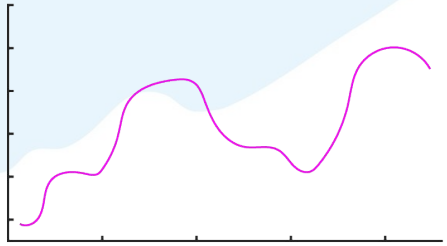


Generative adversarial network



https://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

Generative adversarial network



Art Dealer

Discriminator

Tries to tell Real vs Generated data

Real?

1/0

Generator

Tries to fool the Discriminator

Art Forger

“I knew it was fake because these brush strokes are wrong”

https://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

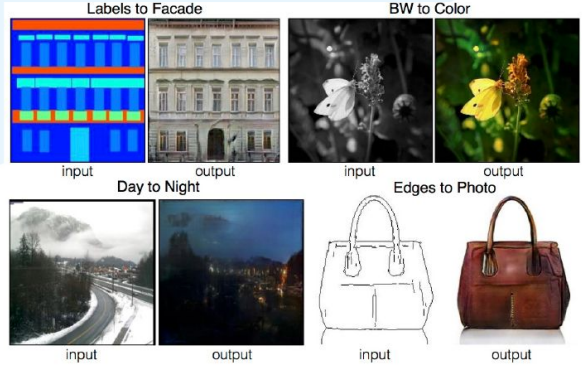
Why are GANs so cool?

Cause we are learning our loss function!

https://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

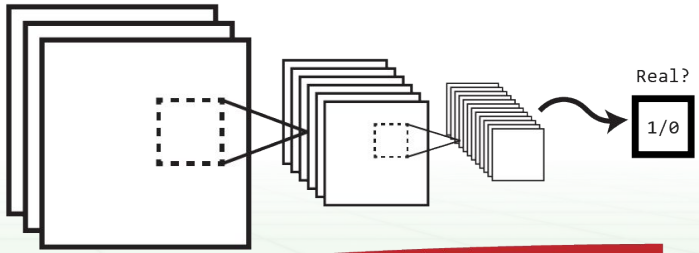
pix2pix: paired image modification

Real Image Pairs



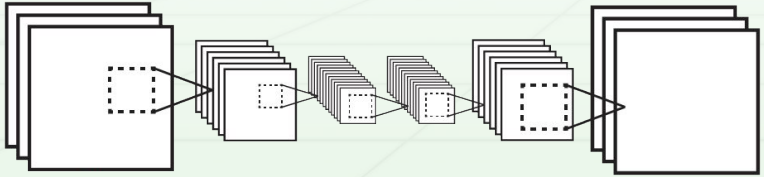
Discriminator

Tries to tell Real Pairs vs Generated Pairs



Generator

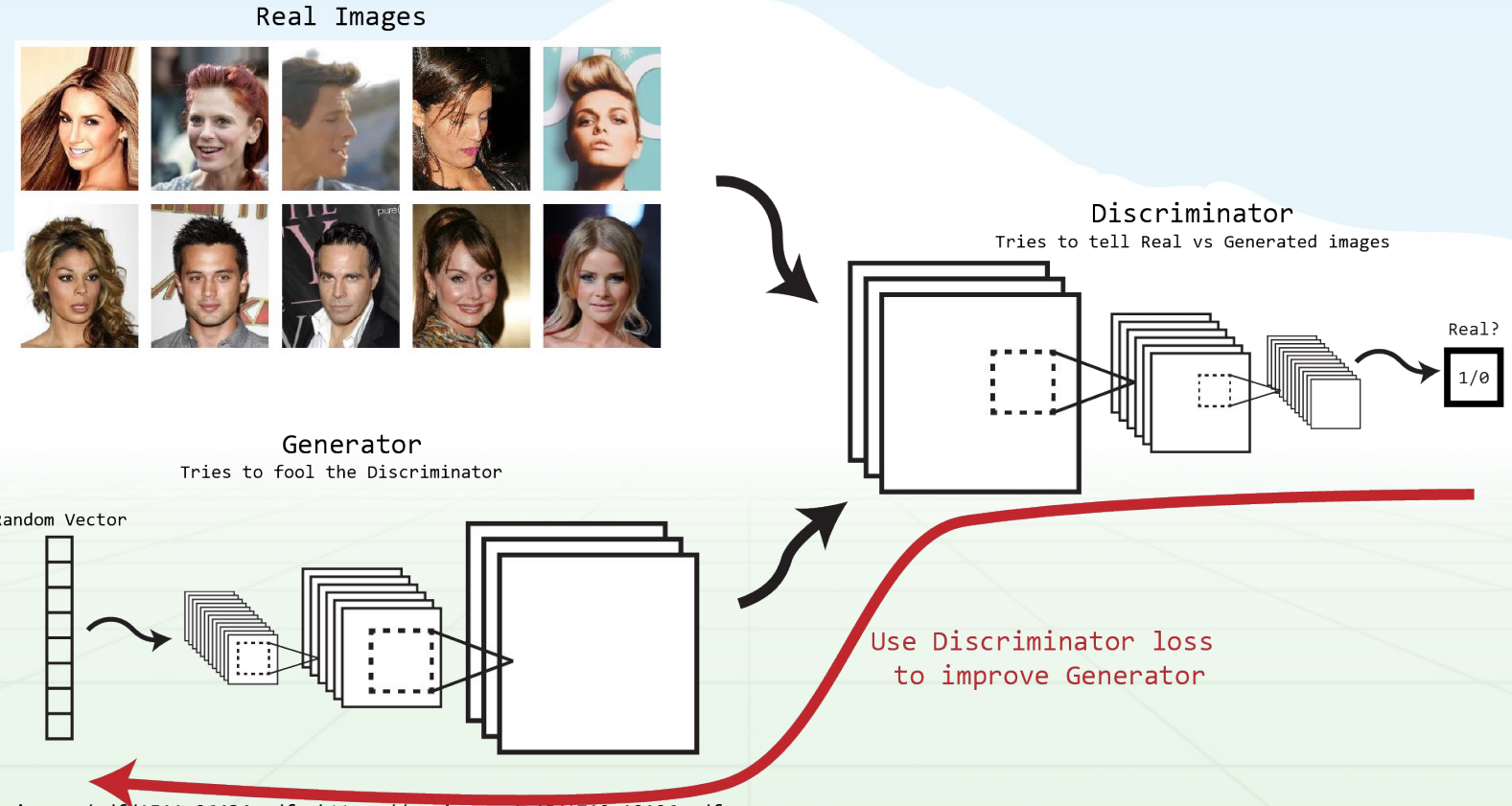
Tries to fool the Discriminator



Use Discriminator loss
to improve Generator

https://affinelayer.com/pixsrv/, https://github.com/phillipi/pix2pix

DCGAN (deep convolutional GANs)



https://arxiv.org/pdf/1511.06434.pdf, https://arxiv.org/pdf/1710.10196.pdf

Progressive growing of GANs

https://arxiv.org/pdf/1710.10196.pdf

ACM Code of Ethics

"An essential aim of computing professionals is to **minimize negative consequences** of computing systems, including threats to health and safety. When designing or implementing systems, computing professionals **must attempt to ensure** that the products of **their efforts will be used in socially responsible ways, will meet social needs, and will avoid harmful effects to health and welfare.**"

https://www.acm.org/about-acm/acm-code-of-ethics-and-professional-conduct


The
End