Machine Learning - Other supervision strategies
(69152) Self-supervised & Generative models

Master in Robotics, Graphics and Computer Vision
Ana C. Murillo

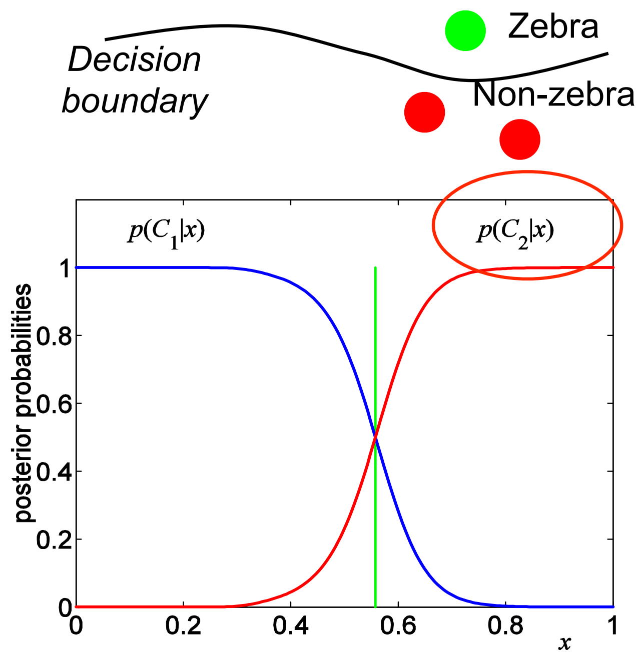
Next

- **Generative models**:
  - Autoencoder
  - Generative Adversarial Networks
  - Diffusion

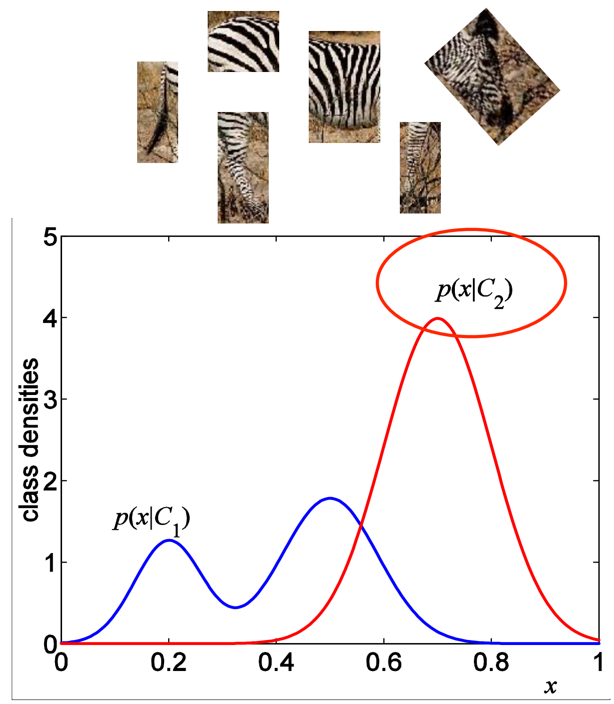
Less human-supervision?

Deep Learning && Generative

- **Discriminative** models:
  - posterior $$P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)}$$

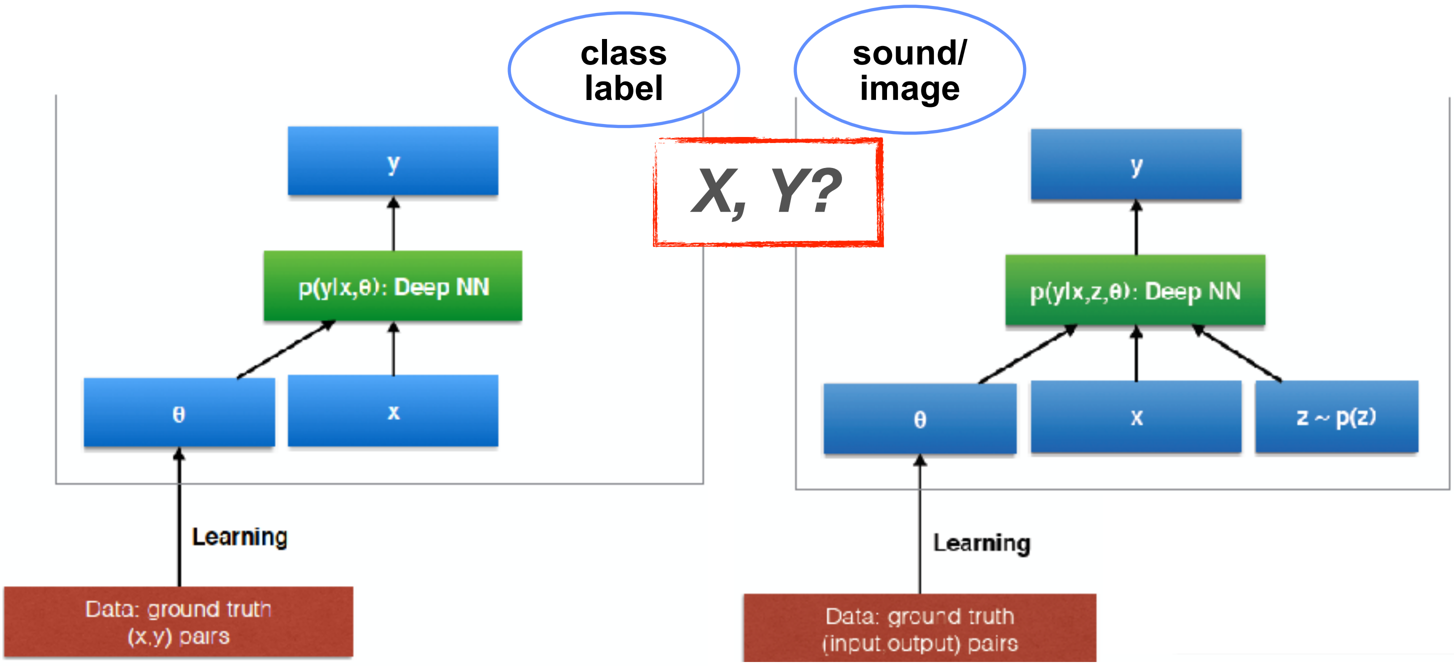


- **Generative** models:
  - likelihood and priors



Deep Learning && Generative

- **Discriminative** models:
- **Generative** models:



Durk Kingma
Berkeley Lecture slides - 2016

Deep Learning && Generative

Assuming data comes from a data distribution that is unknown:

- goal: train networks to learn this distribution from its samples
- … to generate new data by sampling from the learned model

Many **generative models**, with <u>different types of supervision</u>:

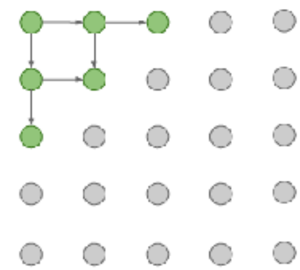
- Pixel RNN, Autoencoders, Generative Adversarial Networks (GAN), Diffusion-based, etc …

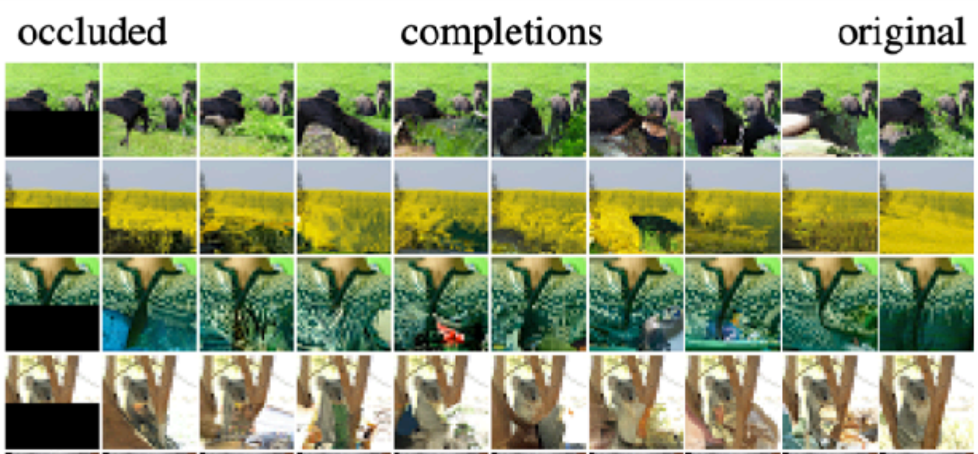
Deep Learning && Generative

- **Pixel RNN / CNN**
  - Start generating one image pixel (e.g. 0,0)
  - Probability of each pixel given the previous

$$p(\mathbf{x}) = \prod_{i=1}^{n^2} p(x_i | x_1, ..., x_{i-1})$$

  - Dependencies on previous pixels
    - modelled with a recursive net (**RNN**)*
    - modelled with a CNN over context region



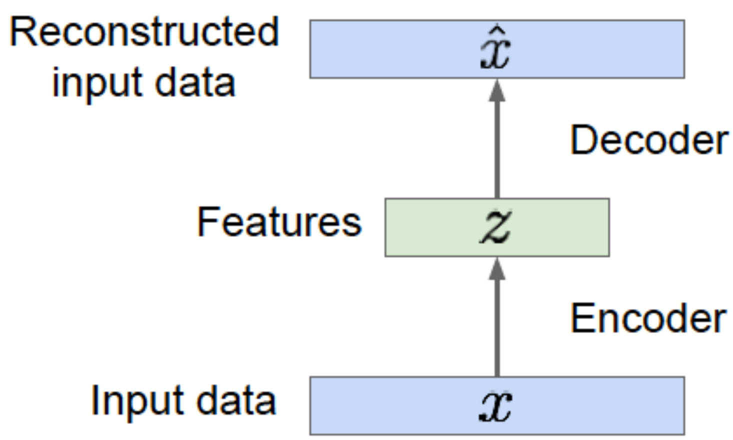


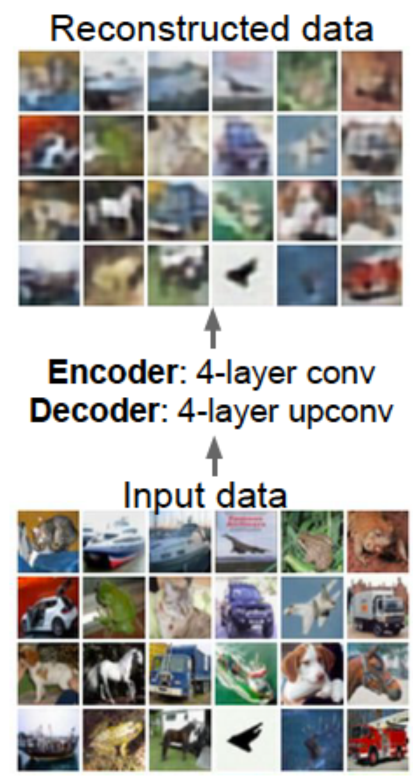
PixelRNN, PixelCNN, van der Oord et al. 2016.

Deep Learning && Generative

- **Auto encoders**
  - Unsupervised learning
  - Learn features that can reconstruct original images (*auto-encoding*)

How to learn this?



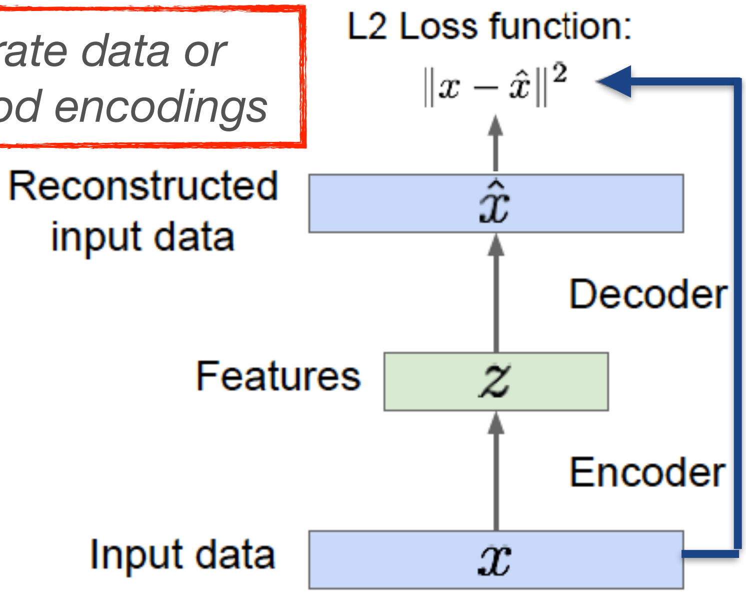


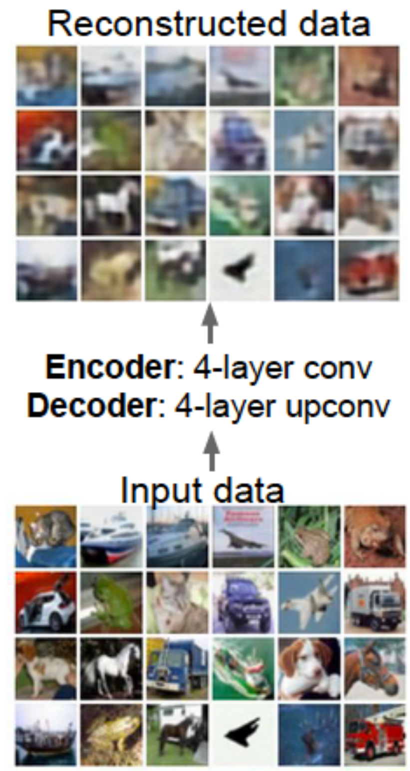
Adapted from Li, Johnson, Yeung. http://cs231n.stanford.edu 2017

Deep Learning && Generative

- **Auto encoders**
  - Unsupervised learning
  - Learn features that can reconstruct original images (*auto-encoding*)

generate data or find good encodings

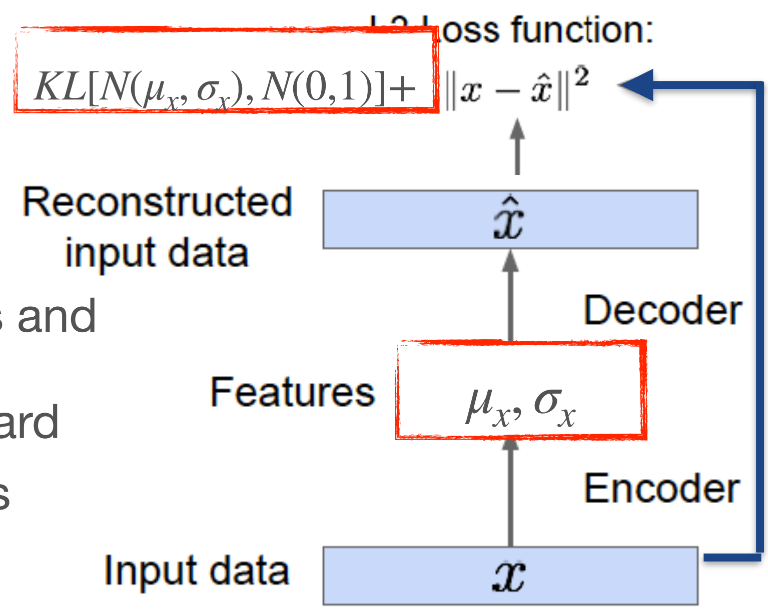




Adapted from Li, Johnson, Yeung. http://cs231n.stanford.edu 2017

Deep Learning && Generative

- Auto-encoder:
  - **deterministic** mapping: input —> **latent vector**
  - only the outputs of the encoder can be decoded … decoding a random latent might give garbage
- **Variational Auto encoders**
  - mapping: input —> **distribution of latent vectors**, i.e., mean and covariance of distribution of latent vectors
  - Train with a reconstruction loss and **regularizing term** to force **Gaussian** to be close to standard
  - probabilistic encoding enforces **structured latents** —> better results for random latent



Deep Learning && Generative

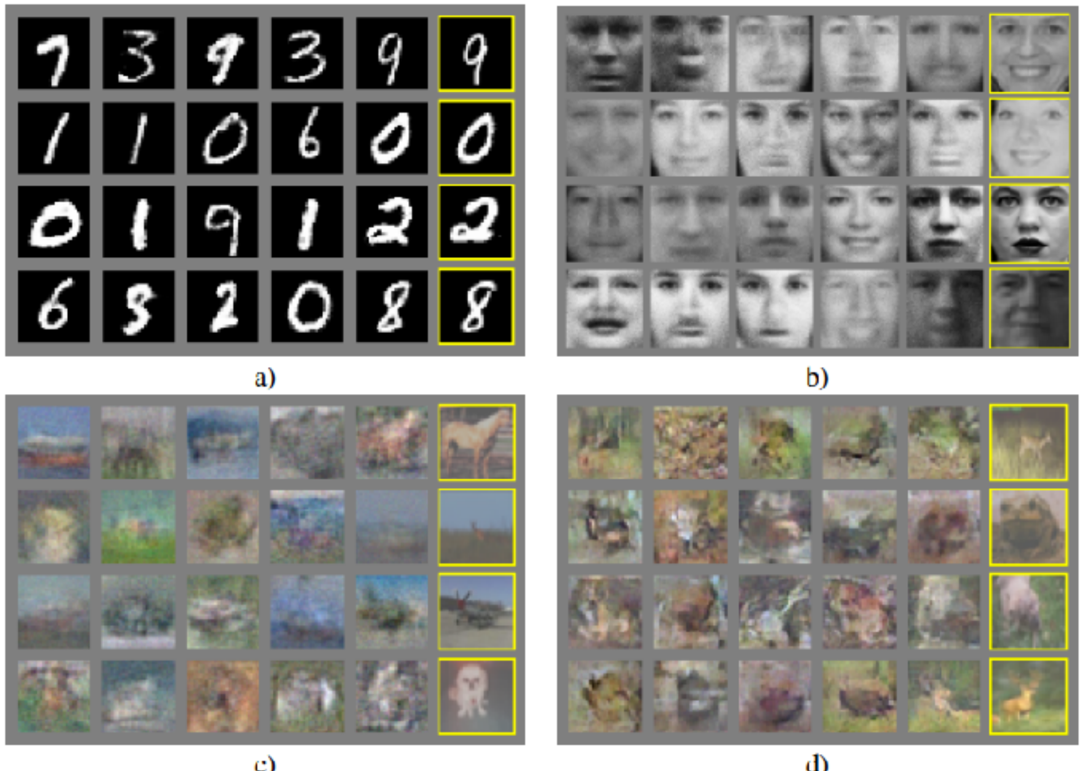
- **Goal**: realistic results from image decoders
- Can I **automatically evaluate** how “realistic” my decoder is?

Generative Adversarial Networks (GAN)

Deep Learning && Generative

- **Generative Adversarial Networks (GAN)**

Train 2 networks simultaneously to generate realistic data



samples from the generator networks

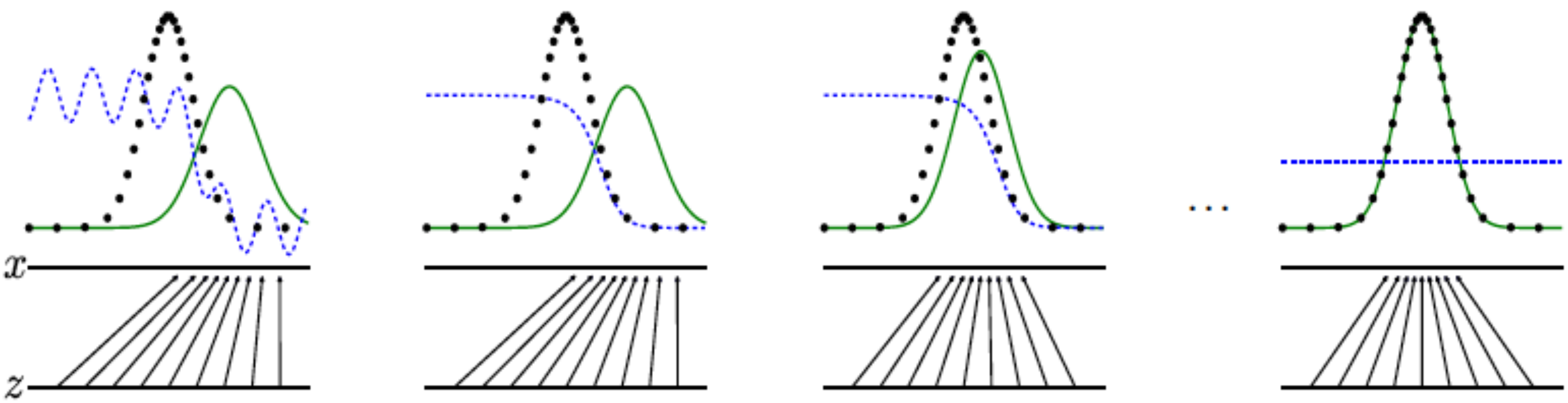
Goodfellow, Ian, et al. "Generative adversarial nets." NIPS 2014.

Deep Learning && Generative

- **Generative Adversarial Networks (GAN)**

Train **2 networks, *G* and *D***: *How?*

  - **Generative** model (captures data distribution) —> maximize probability of D making a mistake
  - **Discriminative** model—> discriminate "actual" data from "generated" data by G, i.e., probability of a sample coming from the training data or G.



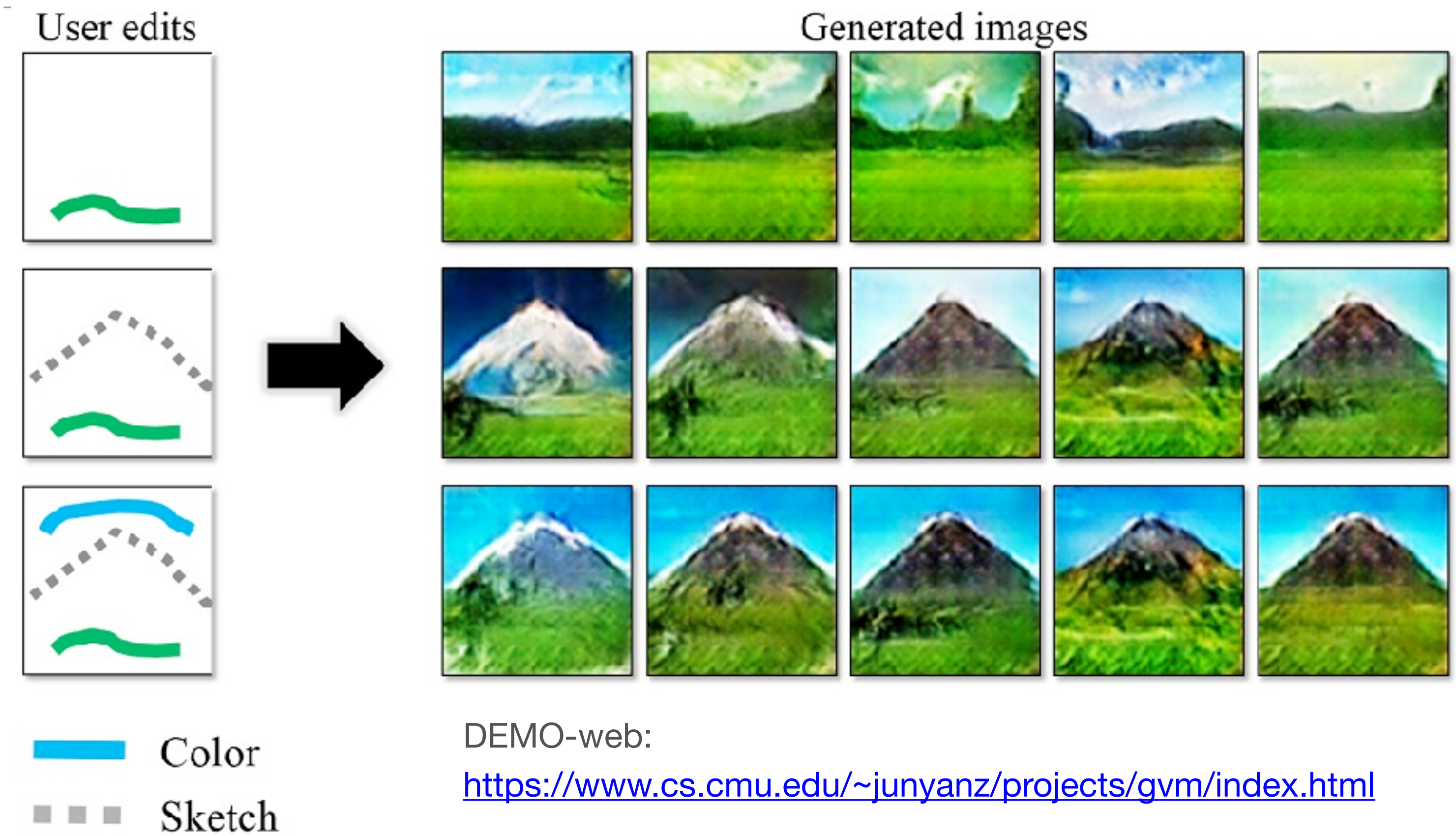
DEMO-web: https://cs.stanford.edu/people/karpathy/gan

Goodfellow, Ian, et al. "Generative adversarial nets" NIPS 2014.

Deep Learning && Generative

- **Generative Adversarial Networks** (GAN)
    - Supervised
      … but "weakly"
    - Known to be **very hard to train**
        - Generator stops learning (e.g. If the discriminator is too good, gradients to the generator can vanish)
        - Generator might slip up and get caught in a "mode collapse" (e.g. finds a few outputs that perform well at fooling the discriminator, and outputs only those for all inputs)
        - Training procedure issues (The vanilla GAN setup doesn't converge, wild oscillations during training)
    - Many strategies proposed to improve these issues

Deep Learning && Generative

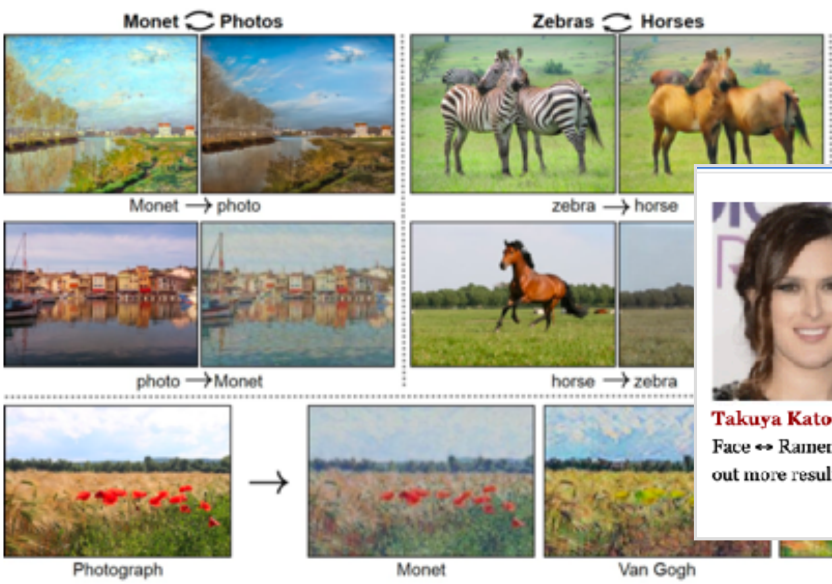


DEMO-web:
https://www.cs.cmu.edu/~junyanz/projects/gvm/index.html

Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman and Alexei A. Efros.
"Generative Visual Manipulation on the Natural Image Manifold", ECCV 2016.

Deep Learning && Generative

- CycleGAN: Gan losses beyond raw image generation… used to ensure "quality" and "realism" in different tasks
  https://junyanz.github.io/CycleGAN/



J. Zhu, T. Park, P. Isola. A. Efros,
"Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks," ICCV 2017.

Deep Learning && Generative

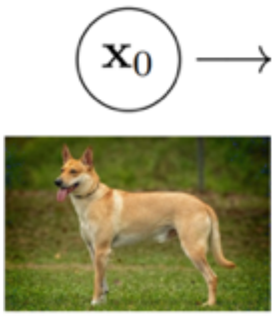
Gatys, Leon A., Alexander S. Ecker, and Matthias Bethge. "Image style transfer using convolutional neural networks." CVPR. 2016.
Johnson, Justin, Alexandre Alahi, and Li Fei-Fei. "Perceptual losses for real-time style transfer and super-resolution" ECCV 2016.

- DEMO-web: https://tenso.rs/demos/fast-neural-style/
- A lot of interactive demos: https://www.nvidia.com/en-us/research/ai-playground/
- Of course … deep-fakes: https://deepfakedetectionchallenge.ai/
- Although lately GANs are losing impact regarding image generation with respect to new text2im approaches using diffusion models …

Deep Learning && Generative

Diffusion models - Forward diffusion process:

- Gradually add very small amounts of Gaussian noise to input data x_0
- Repeat this process for T steps until you get something resembling pure noise

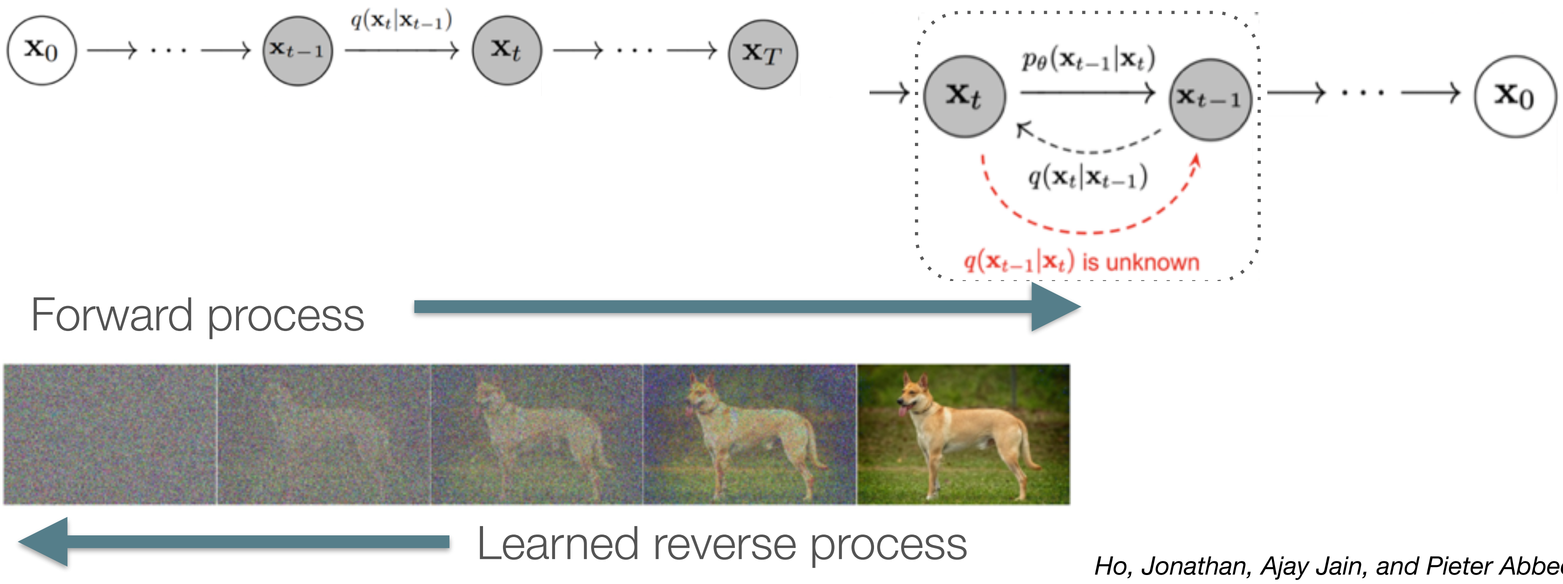


Ho, Jonathan, Ajay Jain, and Pieter Abbeel.
“Denoising diffusion probabilistic models." NeurIPS 2020.

Deep Learning && Generative

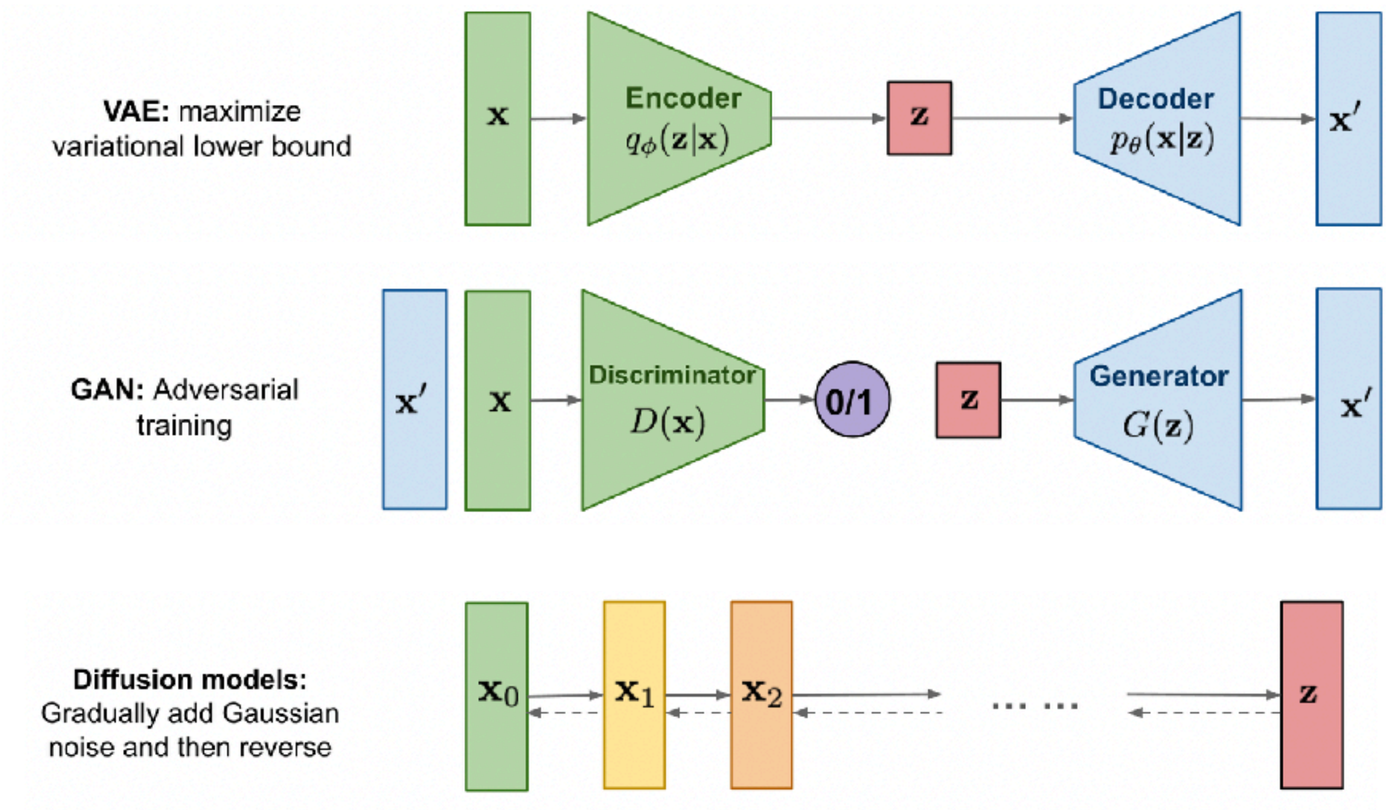
Diffusion models - reverse step

- Goal: learn reverse denoising process to iteratively undo the forward pass
- The reverse process seems to be generating new data from random noise



Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." NeurIPS 2020.

Deep Learning && Generative

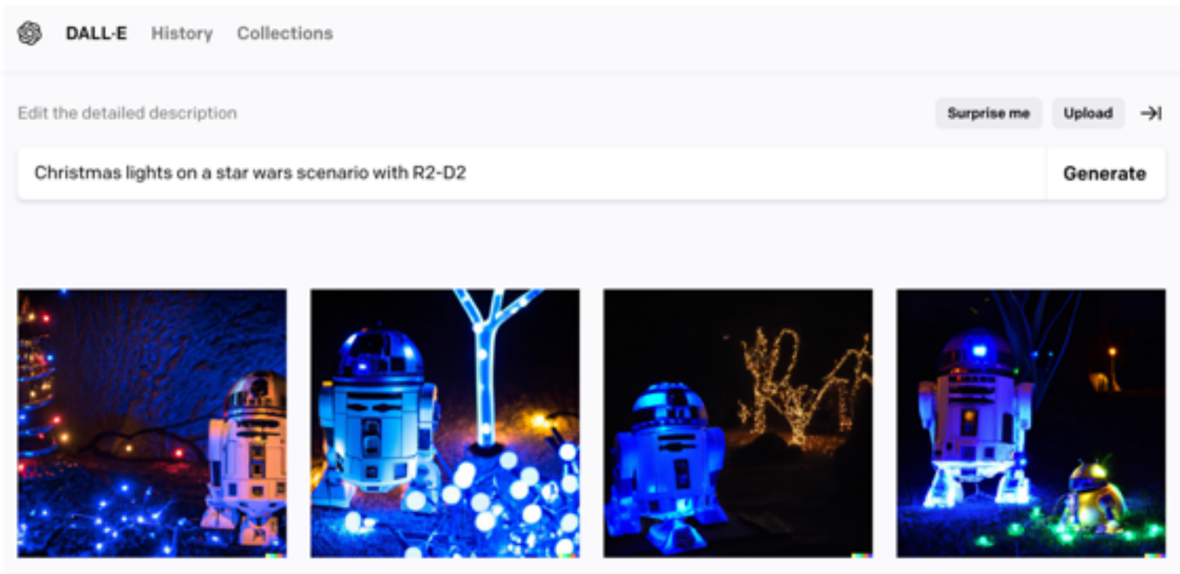


CS 198-126. Deep learning for visual data. UC Berkeley

Deep Learning && Generative

Diffusion models

- Text to image: Dall-E https://labs.openai.com/
- Image to image: https://imagic-editing.github.io/
- Text to video: https://imagen.research.google/video/
- Many more …



Next …

- Recurrent and other *temporal* models
- LAB 5
  - Unsupervised: TSNE, auto-encoders
  - DRL
  - OPTIONAL: Generating Images:
    - Analyzing image gradients & GANs

Prior work for Lab5

- **Extract features from your toy-data** set from Lab2 (decrease set to around 300 images if you have time restrictions)
  - Extract the output of the layer before the last of your fine-turned model (or default mobileNet if you haven't finished your Lab2 or have trouble with it)
  - **Store all the features (one per row) into a numpy matrix (use numpy.save and load)**
  - You can additionally extract any layer output (make sure you flatten the output if necessary)

```
from tensorflow.keras.applications.vgg19 import VGG19
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.vgg19 import preprocess_input
from tensorflow.keras.models import Model
import numpy as np

base_model = VGG19(weights='imagenet')
model = Model(inputs=base_model.input, outputs=base_model.get_layer('block4_pool').output)

img_path = 'elephant.jpg'
img = image.load_img(img_path, target_size=(224, 224))
x = image.img_to_array(img)
x = np.expand_dims(x, axis=0)
x = preprocess_input(x)

block4_pool_features = model.predict(x)
```

Lab 5 template for task 1:
0. Get your deep features

```
from tensorflow.keras.applications.vgg16 import VGG16
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.vgg16 import preprocess_input
import numpy as np

model = VGG16(weights='imagenet', include_top=False)

img_path = 'elephant.jpg'
img = image.load_img(img_path, target_size=(224, 224))
x = image.img_to_array(img)
x = np.expand_dims(x, axis=0)
x = preprocess_input(x)

features = model.predict(x)
```

?

Main resources for the materials in this class

- CS294-129 Designing, Visualizing and Understanding Deep Neural Networks. UC Berkeley.
- CS 198-126. Deep learning for visual data. UC Berkeley.
- Stanford classes on deep learning for Computer Vision http://cs231n.stanford.edu
- *Computer Vision: Algorithms and Applications*. 2nd Edition. Richard Szeliski. https://szeliski.org/Book/