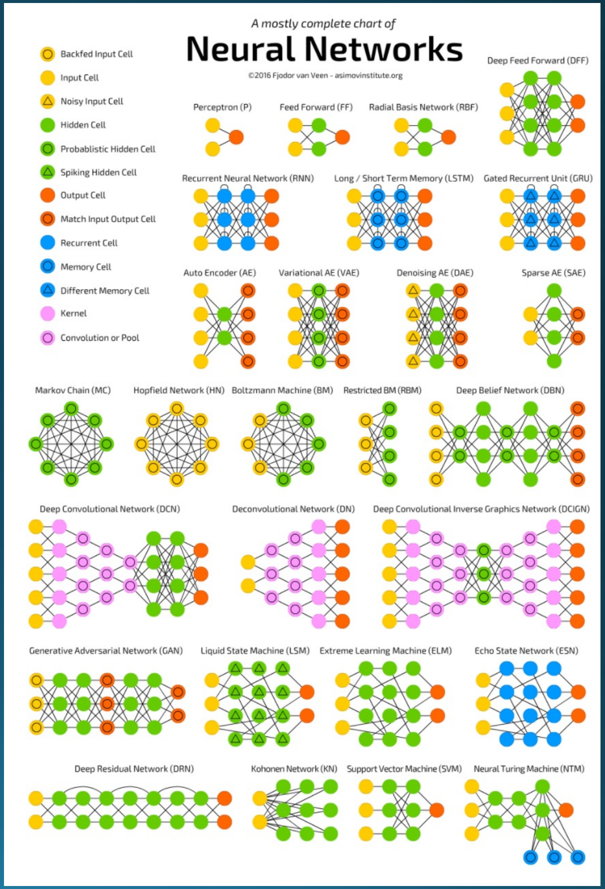
CSCI 4360/6360 Data Science II

Deep Generative Models

The Neural Network Zoo

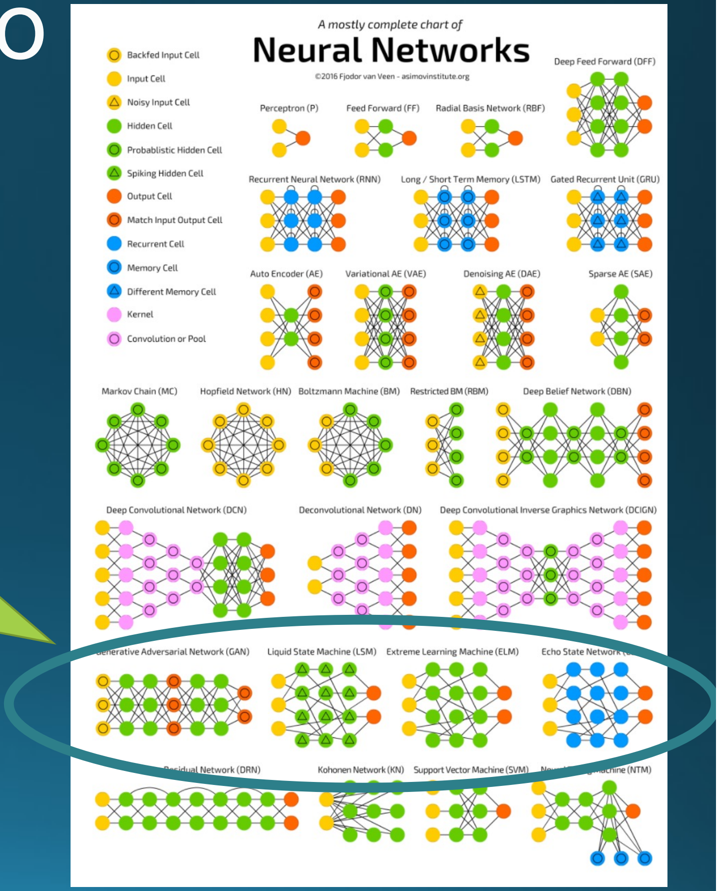
- http://www.asimovinstitute.org/neural-network-zoo/


A mostly complete chart of
Neural Networks
©2016 Fjodor van Veen - asimovinstitute.org
Backfed Input Cell
Input Cell
Noisy Input Cell
Hidden Cell
Probablistic Hidden Cell
Spiking Hidden Cell
Output Cell
Match Input Output Cell
Recurrent Cell
Memory Cell
Different Memory Cell
Kernel
Convolution or Pool
Perceptron (P)
Feed Forward (FF)
Radial Basis Network (RBF)
Deep Feed Forward (DFF)
Recurrent Neural Network (RNN)
Long / Short Term Memory (LSTM)
Gated Recurrent Unit (GRU)
Auto Encoder (AE)
Variational AE (VAE)
Denoising AE (DAE)
Sparse AE (SAE)
Markov Chain (MC)
Hopfield Network (HN)
Boltzmann Machine (BM)
Restricted BM (RBM)
Deep Belief Network (DBN)
Deep Convolutional Network (DCN)
Deconvolutional Network (DN)
Deep Convolutional Inverse Graphics Network (DCIGN)
Generative Adversarial Network (GAN)
Liquid State Machine (LSM)
Extreme Learning Machine (ELM)
Echo State Network (ESN)
Deep Residual Network (DRN)
Kohonen Network (KN)
Support Vector Machine (SVM)
Neural Turing Machine (NTM)

The Neural Network Zoo

- http://www.asimovinstitute.org/neural-network-zoo/

Today



A mostly complete chart of
Neural Networks
©2016 Fjodor van Veen - asimovinstitute.org
Backfed Input Cell
Input Cell
Noisy Input Cell
Hidden Cell
Probablistic Hidden Cell
Spiking Hidden Cell
Output Cell
Match Input Output Cell
Recurrent Cell
Memory Cell
Different Memory Cell
Kernel
Convolution or Pool
Perceptron (P)
Feed Forward (FF)
Radial Basis Network (RBF)
Deep Feed Forward (DFF)
Recurrent Neural Network (RNN)
Long / Short Term Memory (LSTM)
Gated Recurrent Unit (GRU)
Auto Encoder (AE)
Variational AE (VAE)
Denoising AE (DAE)
Sparse AE (SAE)
Markov Chain (MC)
Hopfield Network (HN)
Boltzmann Machine (BM)
Restricted BM (RBM)
Deep Belief Network (DBN)
Deep Convolutional Network (DCN)
Deconvolutional Network (DN)
Deep Convolutional Inverse Graphics Network (DCIGN)
Generative Adversarial Network (GAN)
Liquid State Machine (LSM)
Extreme Learning Machine (ELM)
Echo State Network
Deep Residual Network (DRN)
Kohonen Network (KN)
Support Vector Machine (SVM)
Neural Turing Machine (NTM)

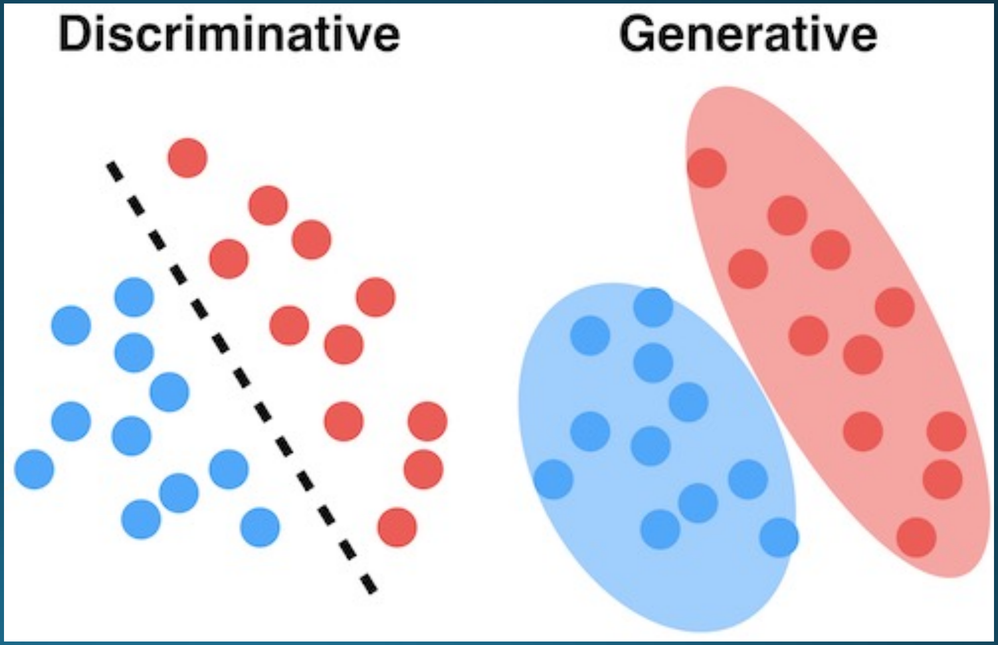
What is a “generative model”?

- Discriminative
  - Logistic Regression
  - Support Vector Machines
  - Random Forests

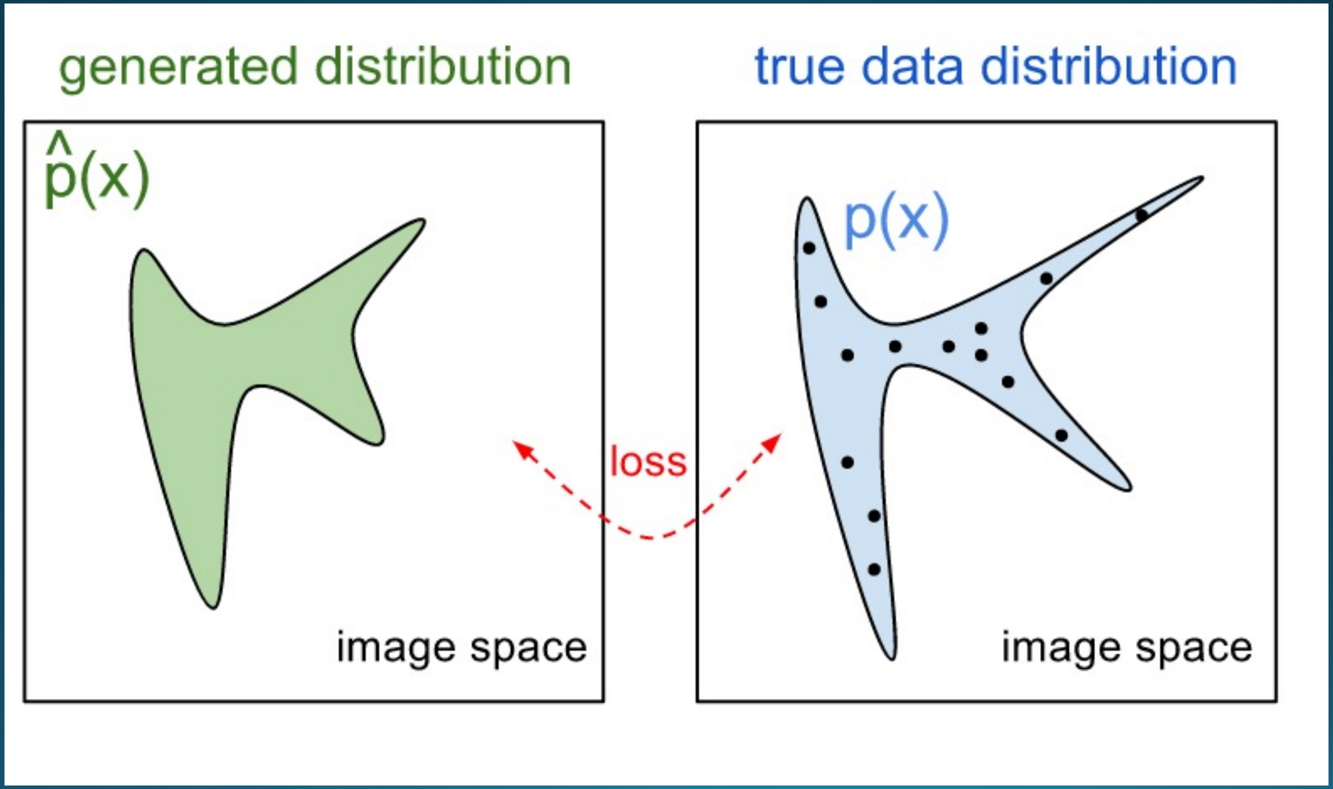
P(Y | X)

- Generative
  - Gaussian Naïve Bayes
  - Variational Autoencoders
  - Adversarial Networks

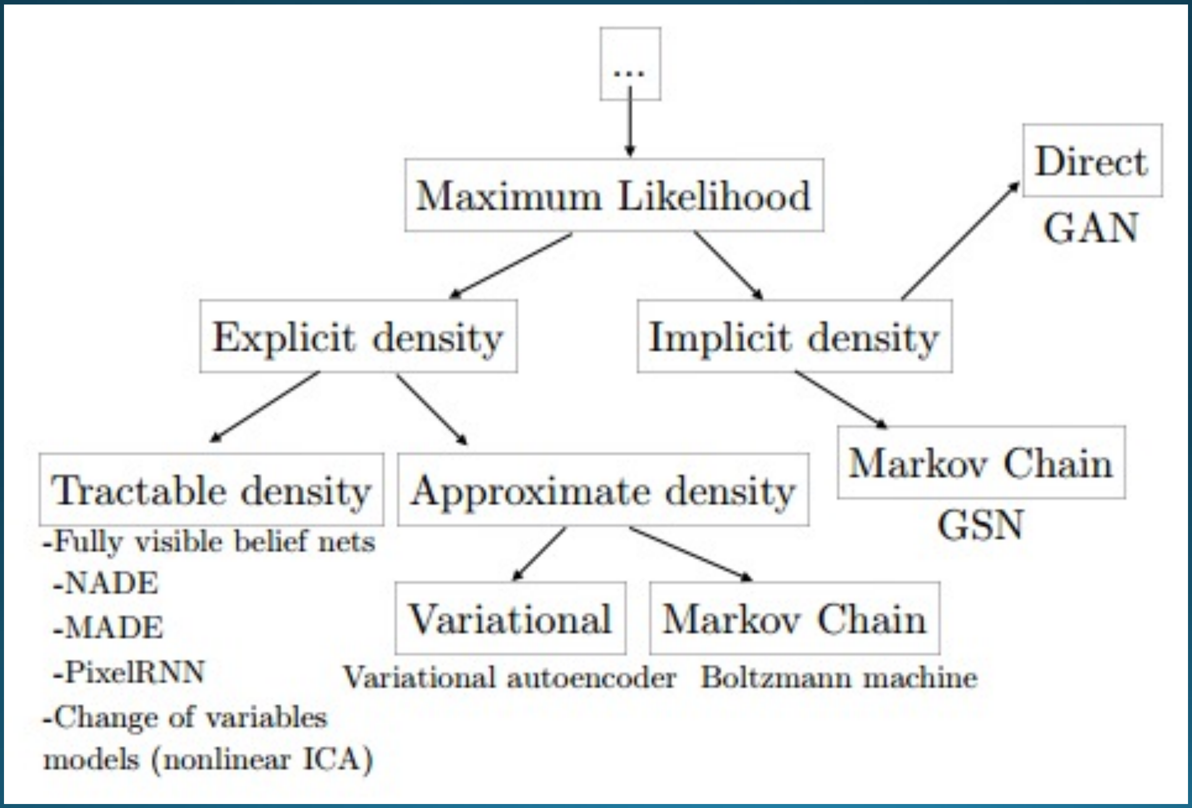
P(X, Y) and P(Y)



Generative Models



Generative Models



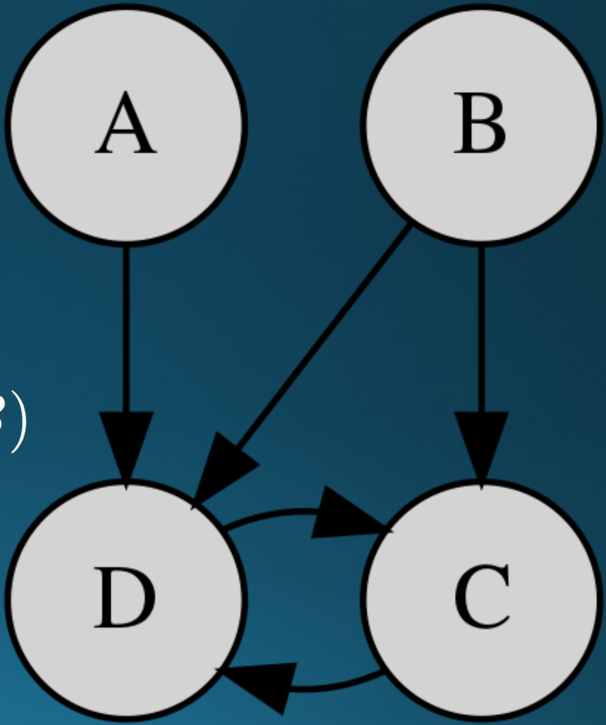
Probabilistic Graphical Models

- Arrows represent conditional dependencies between random variables

$$P(X_1, ..., X_n) = \prod_{i=1}^{n} P(X_i | \text{parents}_i)$$

$$P(A, B, C, D) = P(A)P(B)P(C, D | A, B)$$

- Structure is used in generative models
  - Latent generating distribution (hidden)
  - Observed variables (influenced by latent vars)



Variational Inference

- What is variational inference?
- Good for learning latent variable models (i.e., generating distributions of data)
- For each observation *x* we assign a hidden variable *z*; our model *p* describes the joint distribution between *x* and *z*

$p_\theta(z)$ is very easy 🐣,
$p_\theta(x|z)$ is easy 🐹,
$p_\theta(x, z)$ is easy 🐨,
$p_\theta(x)$ is super-hard 🐍,
$p_\theta(z|x)$ is mega-hard 🐲

Of course these are the things we want to calculate

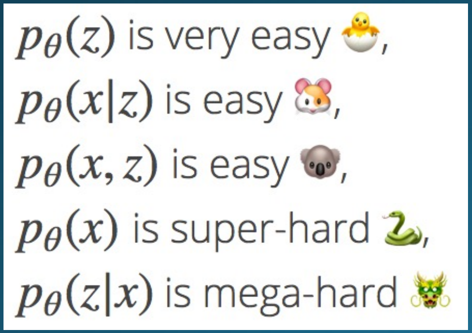
- Inference is p(z|x)
- Learning involves p(x)

Variational Inference

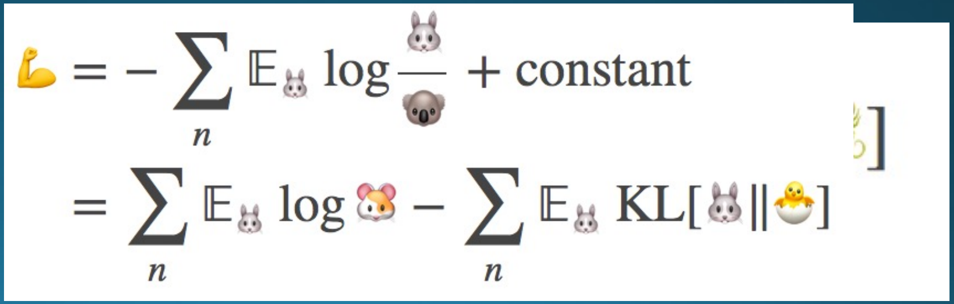
- Rather than learning *p(z|x)* directly, variational inference approximates with *q(z|x)*
- Maximize the evidence lower bound (ELBO)

$$\mathrm{ELBO}(\theta, \psi) = \sum_n \log p(x_n) - \mathrm{KL}\left[q_\psi(z|x_n) || p_\theta(z|x_n)\right]$$

- This can be written in terms of the "friendly" emojis

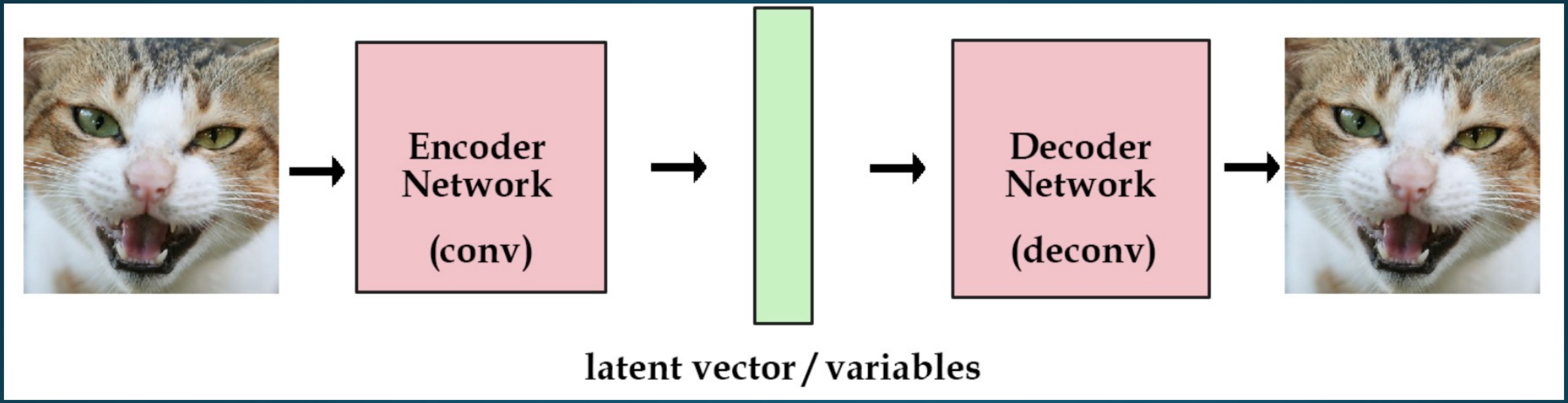


$p_\theta(z)$ is very easy 🐣,
$p_\theta(x|z)$ is easy 🐹,
$p_\theta(x, z)$ is easy 🐨,
$p_\theta(x)$ is super-hard 🐍,
$p_\theta(z|x)$ is mega-hard 🐲



$$\text{💪} = -\sum_n \mathbb{E}_{\text{🐰}} \log \frac{\text{🐰}}{\text{🐨}} + \text{constant}$$
$$= \sum_n \mathbb{E}_{\text{🐰}} \log \text{🐹} - \sum_n \mathbb{E}_{\text{🐰}} \mathrm{KL}[\text{🐰}||\text{🐣}]$$

Recall: Autoencoders



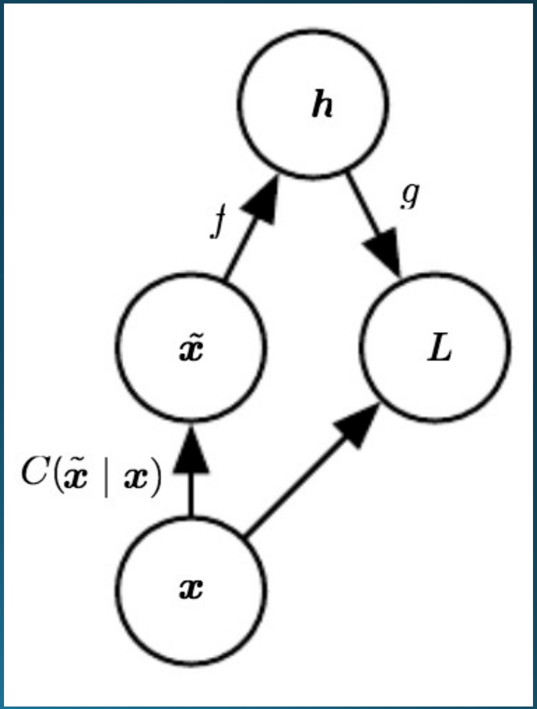
Denoising Autoencoders

- Define a corruption process, *C*

$$C(\tilde{x}|\vec{x})$$

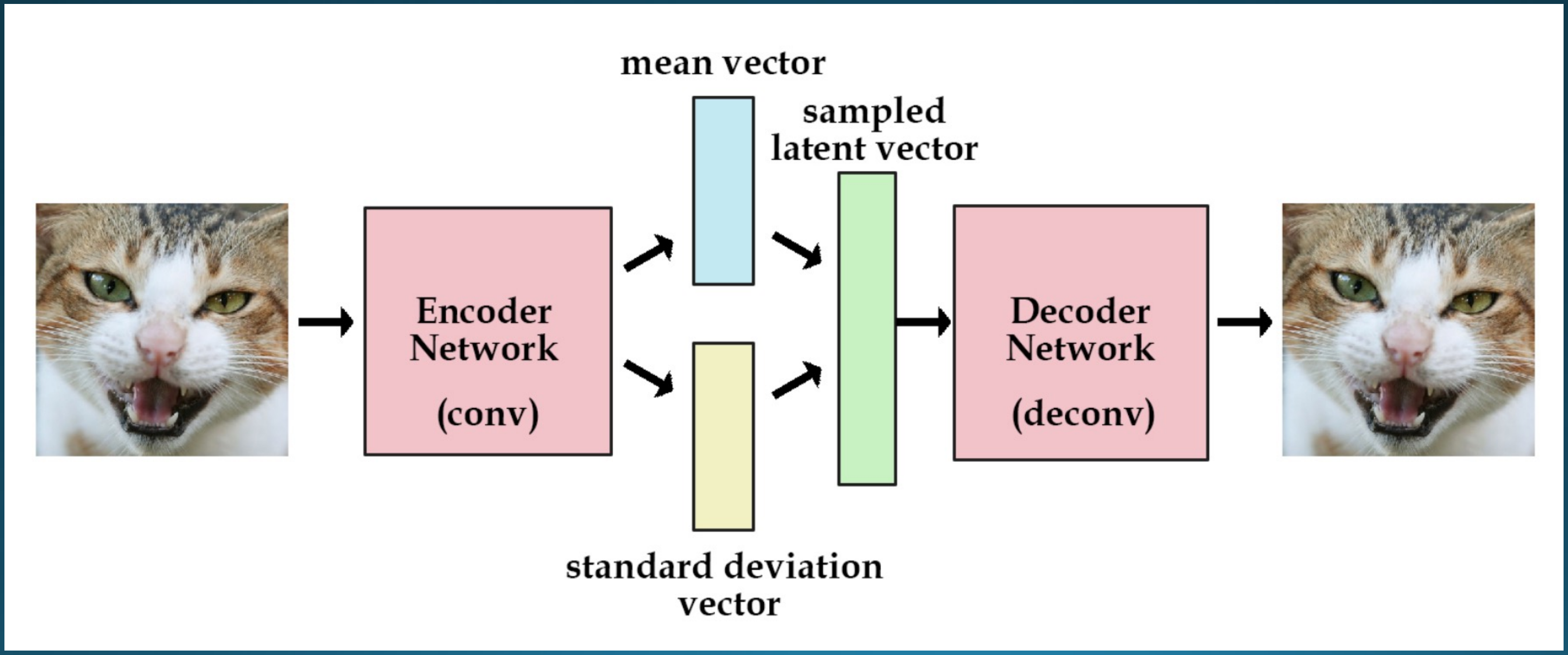
- Autoencoder learns a *reconstruction distribution* $p_{reconstruct}(x\,|\tilde{x})$

1. Sample a training example *x*
2. Sample a corrupted version $\tilde{x}$ from *C*
3. Use $(x, \tilde{x})$ as a training pair



Denoising Autoencoders

- De-corruption process results in learning a *distribution*

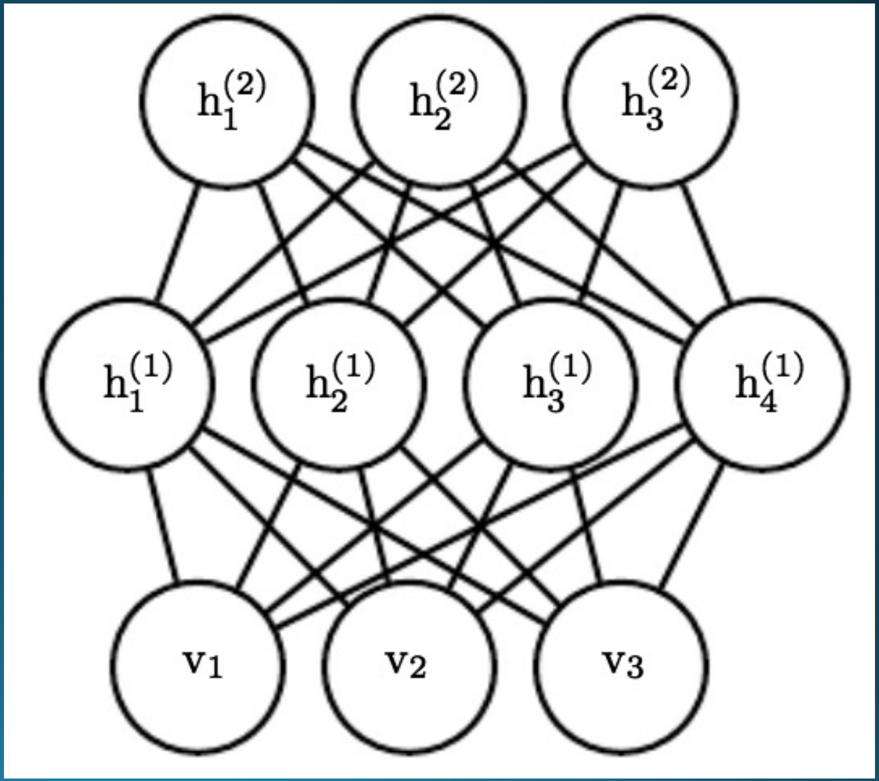


Restricted Boltzmann Machines (RBMs)

- Wholly undirected deep network
  - Implementation of a probabilistic graphical model
  - Each variable conditionally independent given neighboring nodes
- Parameterized by energy function

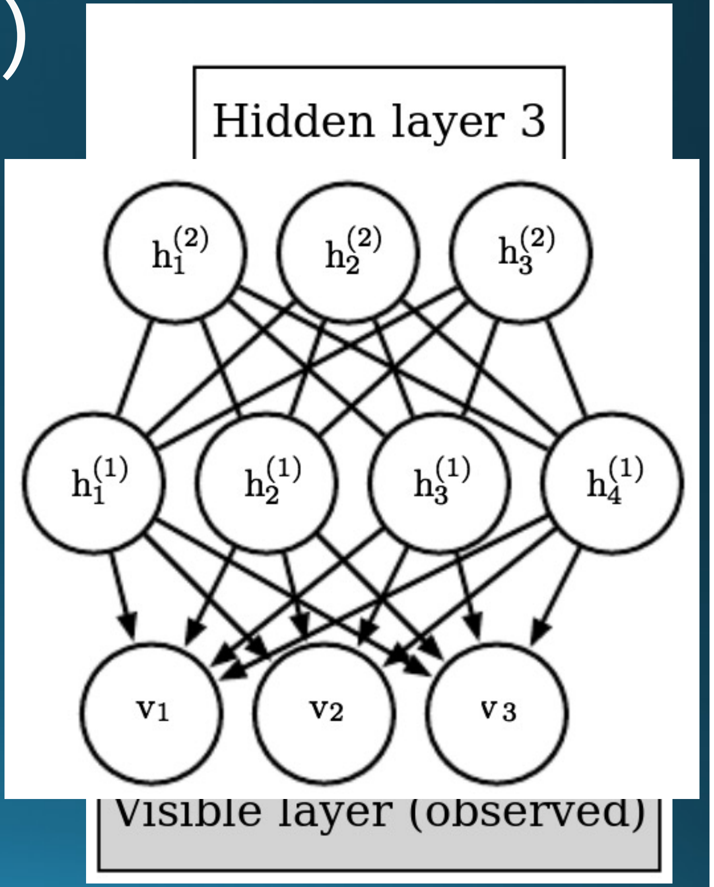
$$P\left(\boldsymbol{v}, \boldsymbol{h}^{(1)}, \boldsymbol{h}^{(2)}, \boldsymbol{h}^{(3)}\right) = \frac{1}{Z(\boldsymbol{\theta})} \exp\left(-E(\boldsymbol{v}, \boldsymbol{h}^{(1)}, \boldsymbol{h}^{(2)}, \boldsymbol{h}^{(3)}; \boldsymbol{\theta})\right)$$

- Sampling from deep RBMs is hard, but training is paradoxically easy



Deep Belief Nets (DBNs)

- Connections *between* layers, but not units *within* a layer
- Arguably one of the first successful applications of modern deep learning
    - Hinton 2006 and 2007
- Often built from an RBM template
- Training is nearly intractable
    - Posterior has to be approximated through annealed importance sampling (AIS)



Generative Adversarial Networks (GANs)

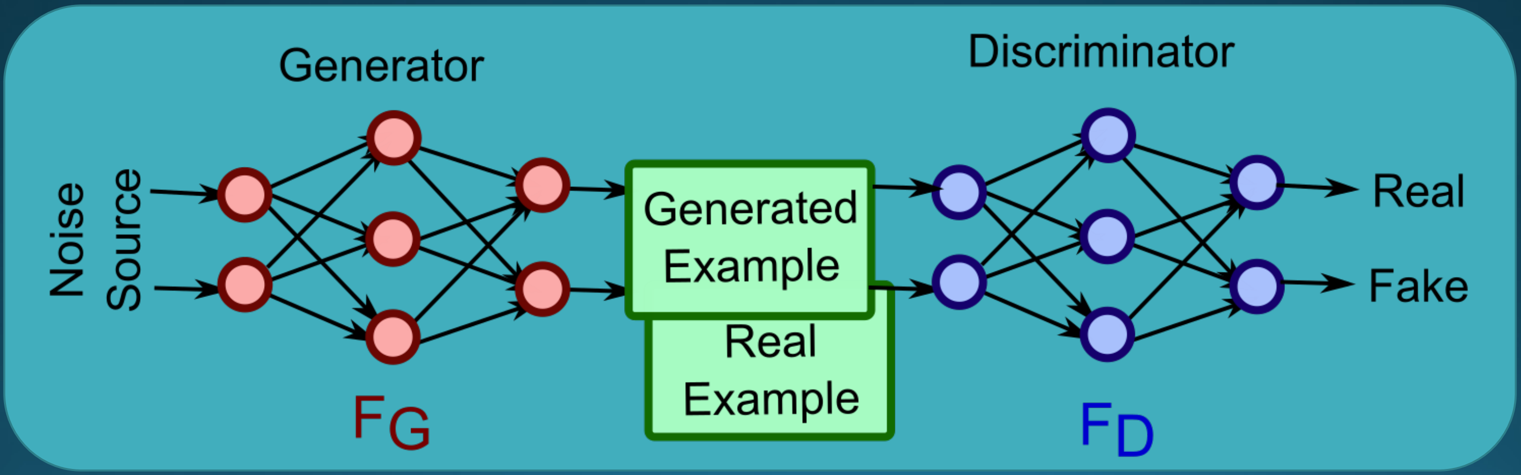
“

There are many interesting recent development in deep learning...The most important one, in my opinion, is adversarial training (also called GAN for Generative Adversarial Networks). This, and the variations that are now being proposed, is the most interesting idea in the last 10 years in ML.

Yann LeCun

GANs

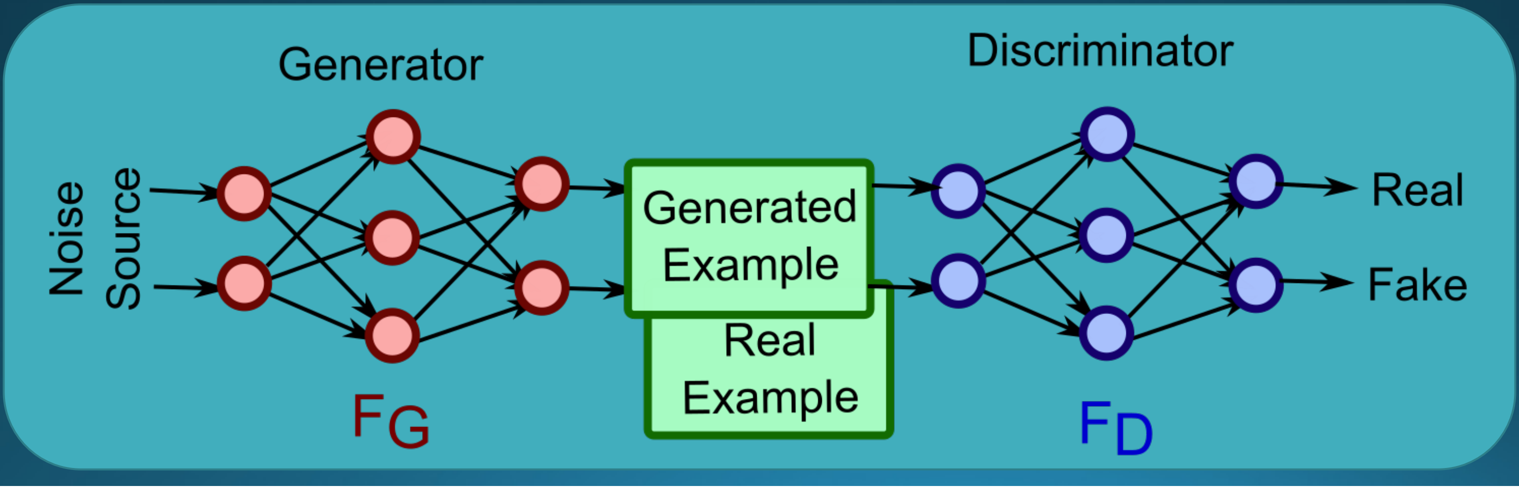
- Game-theoretic approach to generative modeling
- Two deep networks: a **generator** (G) and **discriminator** (D)



GANs

- **Generator**
- Input: a random vector *z*
- Output: something as close to a "real" data point as possible

- **Discriminator**
- Input: a "real" data point OR a synthetic example from *G*
- Output: 1 or 0 (real or fake)



GANs

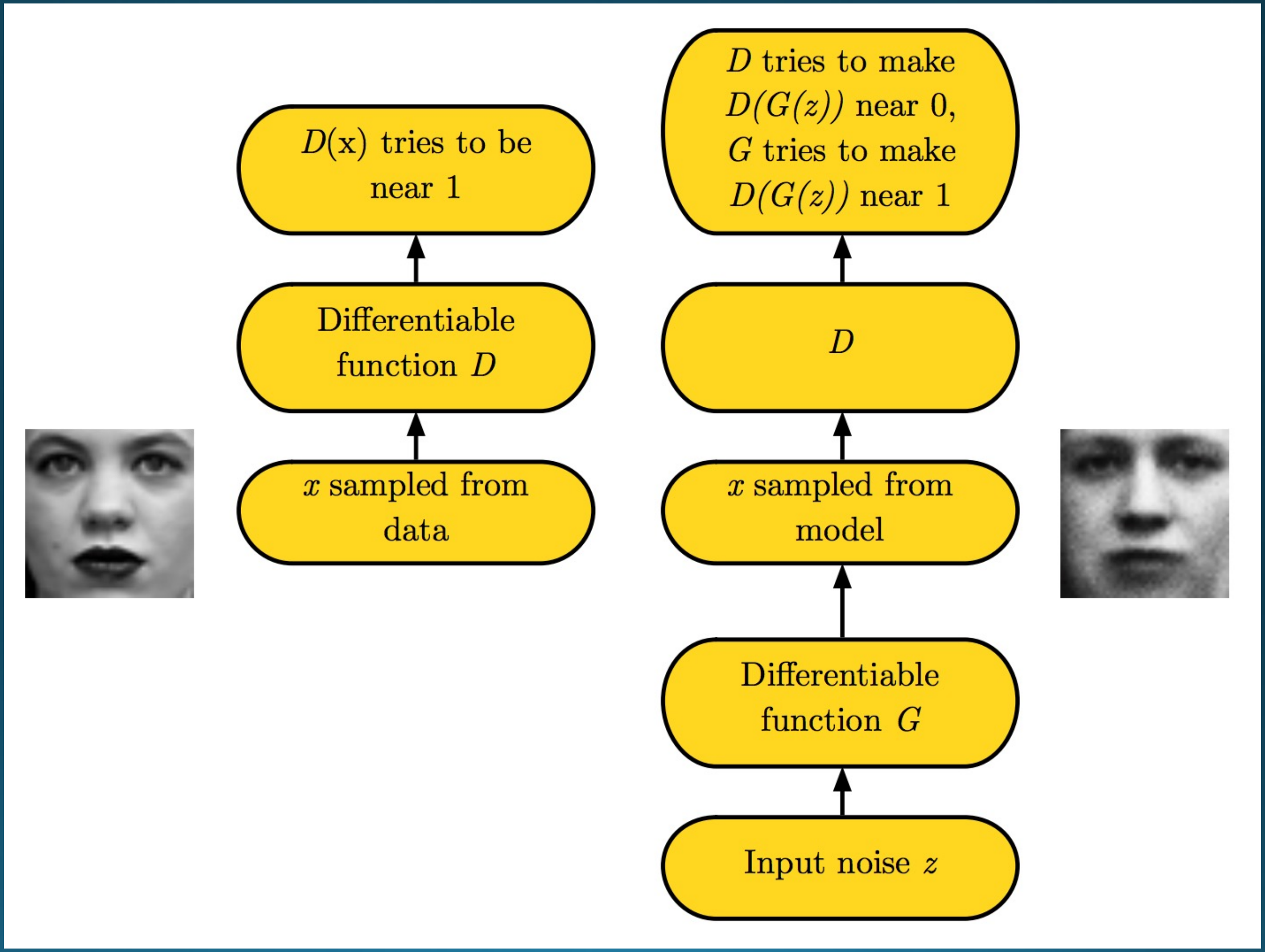
- Minimax "game"
  - Generator and Discriminator have competing objectives
  - Goal is to find an equilibrium point

$$\min_G \max_D \mathbb{E}_{x \sim P_{real}} \log D(x) - \mathbb{E}_z \log(1 - D(G(z)))$$

Maximize the Discriminator's likelihood of identifying a real data example

Minimize the Discriminator's ability to differentiate real data from Generator examplars

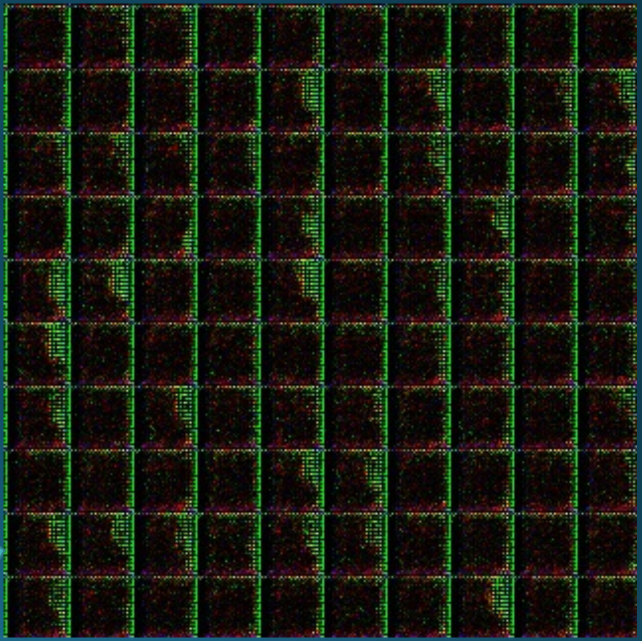
GANs


$D(\mathrm{x})$ tries to be near 1
Differentiable function D
x sampled from data
D tries to make $D(G(z))$ near 0, G tries to make $D(G(z))$ near 1
D
x sampled from model
Differentiable function G
Input noise z

VAEs versus GANs

VAEs
Expectation over learned distribution results in blurring

GANs
Samples from learned distribution, resulting in sharper images



GAN Advances

- Progressively grow the GAN subspace over training

GAN Advances

- Wasserstein objective function
  - "Earth-mover" distance *W(q, p)*
  - Minimum cost of transporting mass in order to transform distribution *q* into the distribution *p* (where cost is mass x distance)

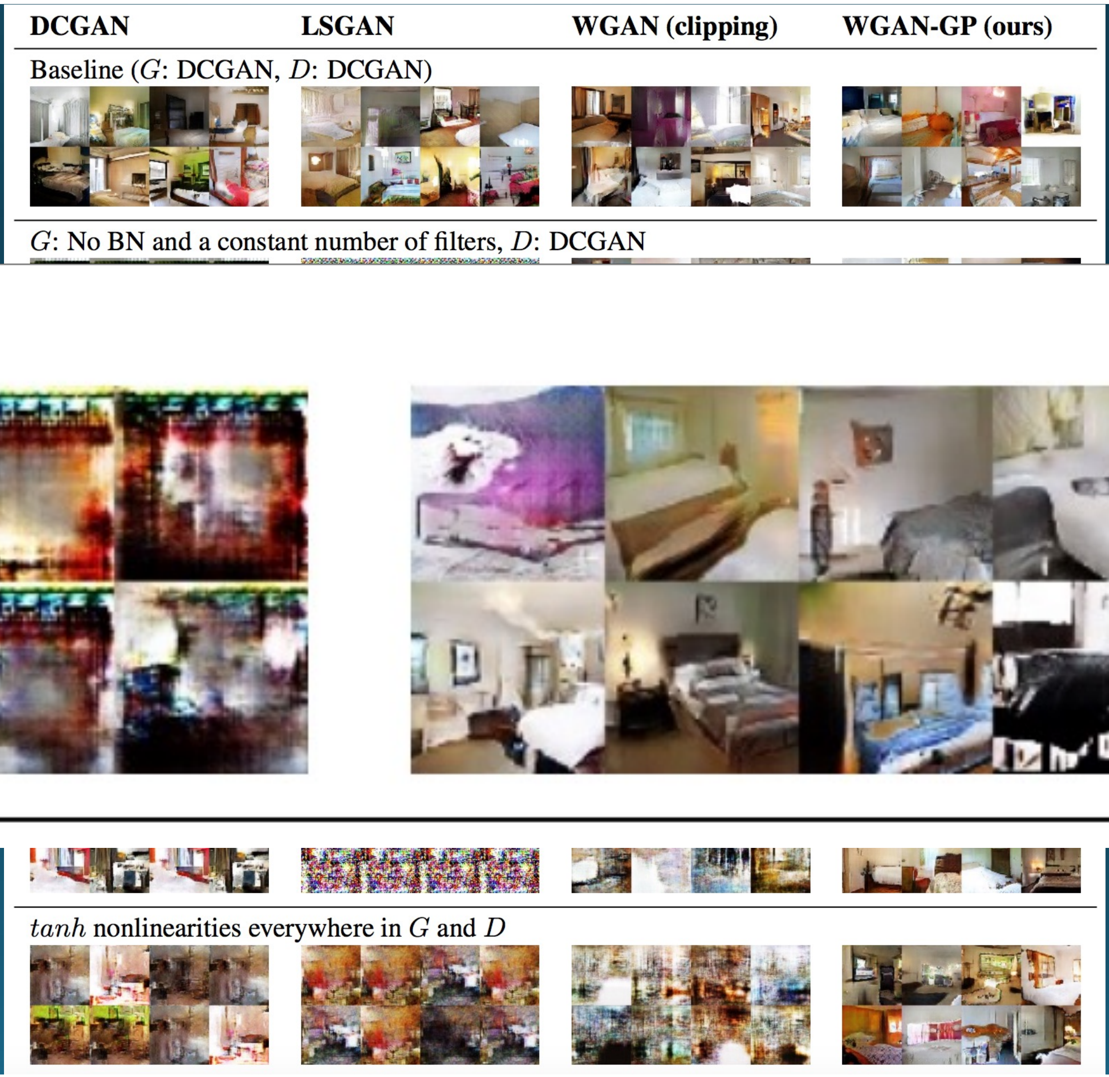
$$\min_G \max_D \mathbb{E}_{x \sim P_{real}} \log D(x) - \mathbb{E}_z \log D(z)$$

- Gradient is much better behaved than Jenson-Shannon objective (KL-divergence based)
- Weights are clipped at *[-c, c]*
- Takes a lot longer to train on average

GAN Advances

- Improved Wasserstein
  - Introduces a gradient penalty on the discriminator output with respect to its input
  - Instead of hard clipping gradient weights, soft[max] penalties are used
  - P_r is the distribution of real data, P_g is from the generator, and P_x is defined from sampling uniformly along straight lines between pairs of points sampled from P_r and P_g

$$\min_G \max_D \underbrace{\mathbb{E}_{\tilde{x}\sim P_g}[D(\tilde{x})] - \mathbb{E}_{x\sim P_r}[D(x)]}_{\text{Original** WGAN objective}} + \underbrace{\lambda \mathbb{E}_{\hat{x}\sim P_x}\left[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2\right]}_{\text{Two-sided gradient penalty on Discriminator}}$$

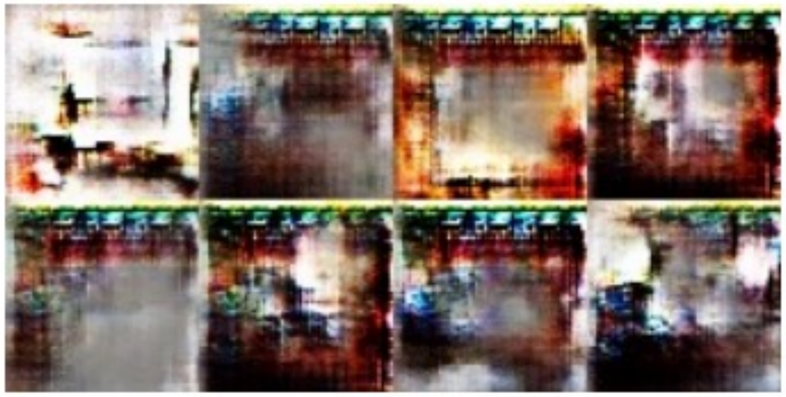


DCGAN	LSGAN	WGAN (clipping)	WGAN-GP (ours)

Baseline (G: DCGAN, D: DCGAN)

G: No BN and a constant number of filters, D: DCGAN

$tanh$ nonlinearities everywhere in G and D

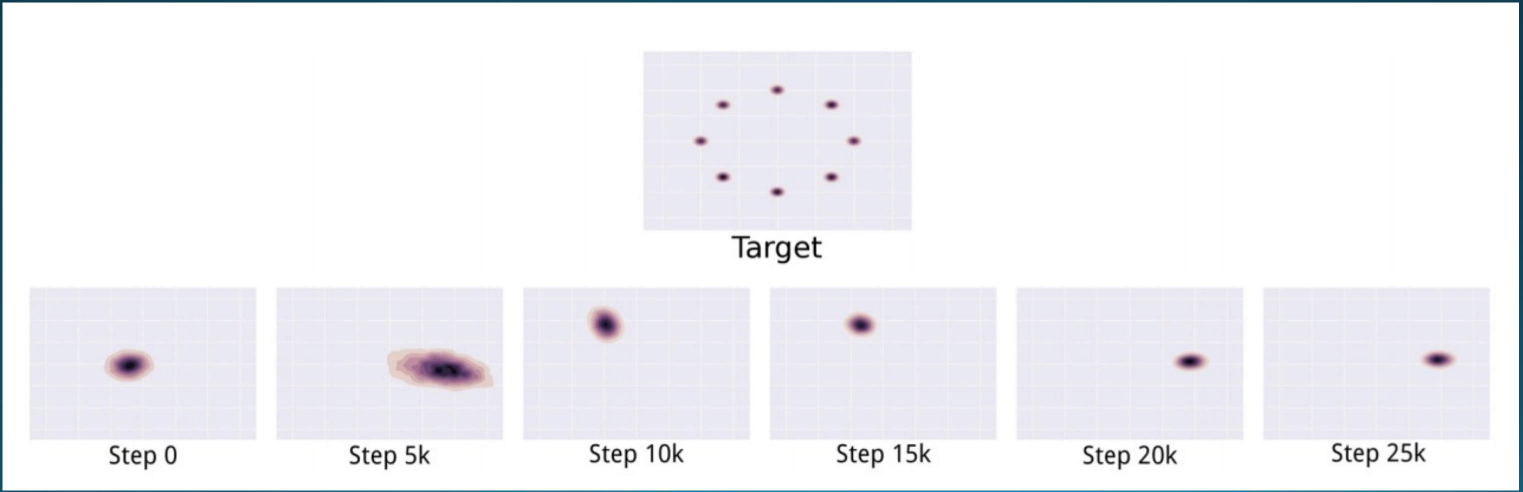


Open Questions with GANs

$$G^* = \min_G \max_D V(G, D),$$

$$G^* = \max_D \min_G V(G, D),$$

- **Mode collapse**
  - The "Helvetica Scenario"
  - Maps several different inputs *z* to the same output
  - Full collapse is rare, but partial collapse is common



Open Questions with GANs

- **Evaluation of GANs**

- (not specific to GANs *per se*, but generative models)

- Models that obtain good likelihoods can generate bad samples
- Models that generate good samples can have poor likelihoods
- Also difficult to evaluate likelihood with GANs

Open Questions with GANs

- Goal is to find *Nash equilibrium*

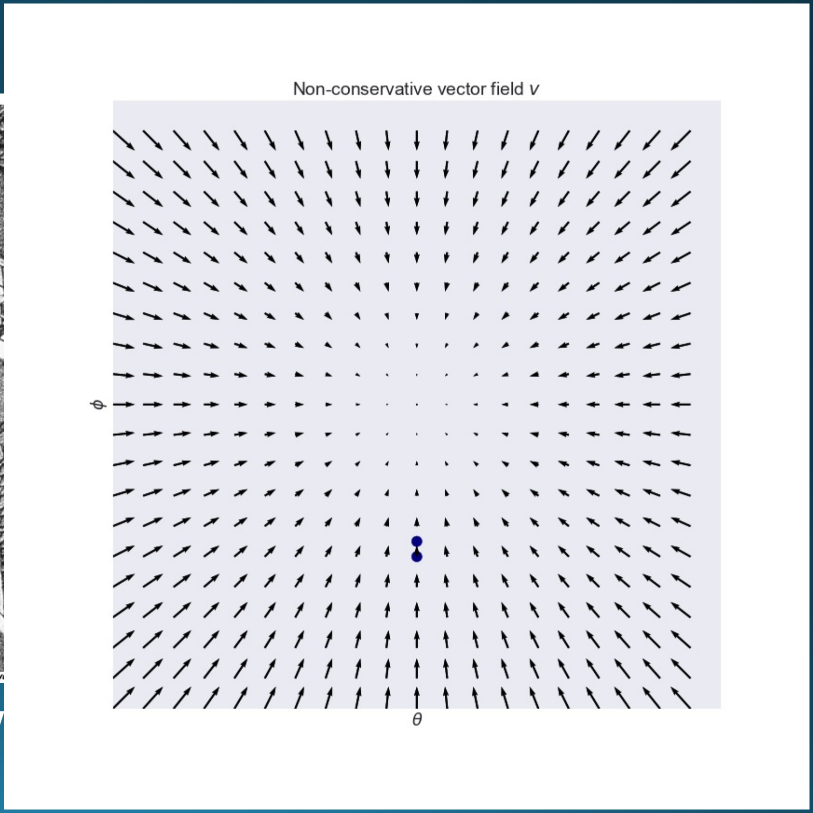
- **Problem 1: Does it exist?**
  - No conclusive way to show this
- **Problem 2: If it exists, can we find it?**
  - Inability to find equilibrium can be cause of oscillatory behavior in training
  - …or a sign that equilibrium doesn't exist?
- **Problem 3: More than finding equilibrium, can generator *win?***
  - Intuitively: to learn as representative a generator as possible, discriminator should be *utterly unable to differentiate between real and fake*

Training a GAN

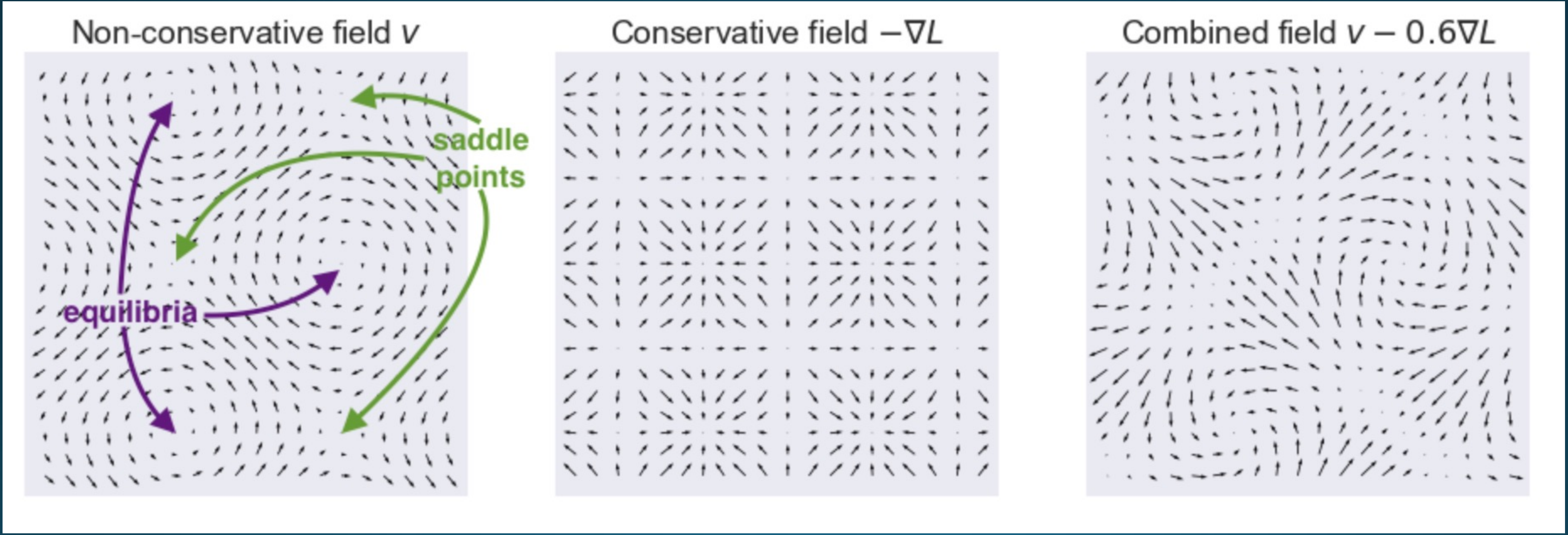
- GANs use a variant of SGD called *simultaneous gradient descent*
  - Key difference: the latter gives rise to *non-conservative vector fields*
  - Like Escher's staircase
- Solution: convert to conservative vector field

$$-\nabla L(x) := -\frac{\partial}{\partial x}||v(x)||_2^2$$

- **New problem: can't differentiate between saddle points or equilibria, or negative or positive equilibria**



Training a GAN



Non-conservative field v
saddle points
equilibria
Conservative field $-\nabla L$
Combined field $v - 0.6\nabla L$

Open Questions with GANs

- **Exploration of the learned manifold**

- Advantage of GANs: no *a priori* assumptions on the underlying form of the generating distribution
- Disadvantage of GANs: no way to meaningfully interpret the resulting learned generating distribution

- Manifold walking, interpolation, image algebra, INFO-GANs

Conclusions

- Generative modeling
  - Learn a *distribution* instead of a decision boundary
  - Can still be used for classification
  - Usually requires more data than discriminative models
- Deep generative modeling
  - Denoising & Variational Autoencoders
  - Deep Belief Networks
  - Restricted Boltzmann Machines
- Generative Adversarial Networks (GANs)
  - Game-theoretic generative modeling through dueling deep networks
  - Attempt to find Nash equilibrium
  - Many numerical and algorithmic limitations but results are impressive

References

- "Generative Learning algorithms", CS 229 notes by Andrew Ng http://cs229.stanford.edu/notes/cs229-notes2.pdf
- Andrew Ng and Michael I. Jordan (!!!), "On Discriminative vs. Generative classifiers: A comparison of logistic regression and naive Bayes" (NIPS 2002) http://papers.nips.cc/paper/2020-on-discriminative-vs-generative-classifiers-a-comparison-of-logistic-regression-and-naive-bayes.pdf
- "Generative versus Discriminative", Cross-Validated https://stats.stackexchange.com/questions/12421/generative-vs-discriminative/223850
- "Generative Adversarial Networks", NIPS 2016 Tutorial https://arxiv.org/pdf/1701.00160.pdf
- Variational Inference http://www.inference.vc/choice-of-recognition-models-in-vaes-a-regularisation-view/
- Tutorial on Variational Autoencoders https://arxiv.org/abs/1606.05908
- "Progressive Growing of GANs for Improved Quality, Stability, and Variation", https://arxiv.org/abs/1710.10196
- "Generative Adversarial Networks (GANs), Some Open Questions", http://www.offconvex.org/2017/03/15/GANs/
- "GANs are Broken in More than One Way: The Numerics of GANs", http://www.inference.vc/my-notes-on-the-numerics-of-gans/