Deep generative models of natural images

Emily Denton

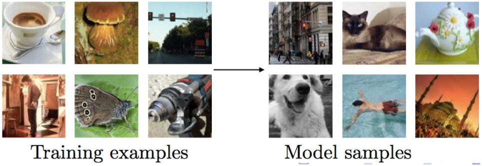
Fall 2016

Outline

- Have access to $x \sim p_{data}(x)$ through training set
- Want to learn a model $x \sim p_{model}(x)$
- Want p_{model} to be similar to p_{data}:

Samples from true data distribution have high likelihood under p_{model}

Samples drawn from p_{model} reflect structure of p_{data}



Why do generative modeling?

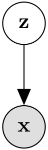
- Unsupervised representation learning
  - Can transfer learned representation so discriminative tasks, retrieval, clustering, etc.
- Train network with both discriminative and generative criterion
  - Utilize unlabeled data, regularize
- Understand data
- Density estimation
- Data augmentation
- ...

Focus of this talk

Generative modeling is a HUGE field...I will focus on (a selected set of) deep directed models of natural images

Outline

Directed graphical models



- We assume data is generated by:

$$z \sim p(z) \qquad x \sim p(x|z)$$

- z is latent/hidden x is observed (image)
- Use θ to denote parameters of the generative model

Parameter estimation

- Given dataset $\{x_1, ..., x_n\}$, maximize likelihood of data under model:

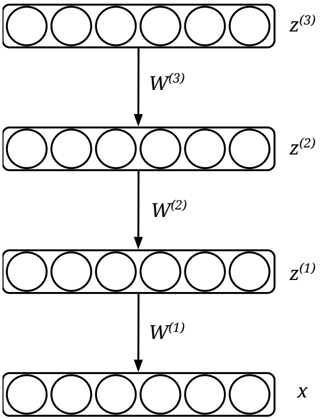
$$\max_\theta \sum_{i=1}^{n} \log p(x_i;\theta) = \max_\theta \sum_{i=1}^{n} \sum_z \log p(x_i, z;\theta)$$

- This quantity often intractable, difficult to optimize directly

- Can be optimized with iterative Expectation Maximization (EM) algorithm
    - Fix parameters and compute log likelihood wrt $p(z|x;\theta^t)$
    - Fix z find parameters $\theta^{(t+1)}$ to maximize log likelihood

Parameter estimation

- Standard EM requires access to posterior $p(z|x)$
- For the deep neural net models we care about this is infeasible
- Solution: introduce *variational* approximation $q(z;\phi)$ to $p(z|x)$
- Will give bound on log likelihood



Bounding the marginal likelihood

Recall Jenson's inequality: When f is concave, $f(\mathbb{E}[x]) \geq \mathbb{E}[f(x)]$

$$
\begin{aligned}
\log p(x) &= \log \int_z p(x, z) \\
&= \log \int_z q(z) \frac{p(x, z)}{q(z)} \\
&\geq \int_z q(z) \log \frac{p(x, z)}{q(z)} = L(x; \theta, \phi) \qquad \text{(by Jensons inequality)} \\
&= \int_z q(z) \log p(x, z) - \int_z q(z) \log q(z) \\
&= \underbrace{\mathbb{E}_{q(z)}[\log p(x, z)]}_{\text{Expectation of joint distribution}} + \underbrace{\mathrm{H}(q(z))}_{\text{Entropy}}
\end{aligned}
$$

Bound is tight when variational approximation matches true posterior:

$$
\begin{aligned}
\log p(x) - L(x;\theta,\phi) &= \log p(x) - \int_z q(z) \log \frac{p(x,z)}{q(z)} \\
&= \int_z q(z) \log p(x) - \int_z q(z) \log \frac{p(x,z)}{q(z)} \\
&= \int_z q(z) \log \frac{q(z)p(x)}{p(x,z)} \\
&= \int_z q(z) \log \frac{q(z)}{p(z|x)} \\
&= D_{KL}(q(z;\phi)||p(z|x))
\end{aligned}
$$

Summary

- Assume existence of $q(z;\phi)$
- Bound $\log p(x;\theta)$ with $L(x;\theta,\phi)$
- Bound is tight when:

$$D_{KL}(q(z;\phi)||p(z|x)) = 0 \iff q(z;\phi) = p(z|x)$$

Learning directed graphical models

- Maximize bound on likelihood of data:

$$\max_{\theta} \sum_{i=1}^{N} \log p(x_i; \theta) \geq \max_{\theta, \phi_1, \ldots, \phi_N} \sum_{i=1}^{N} L(x_i; \theta, \phi_i)$$

- Historically, used different ϕ_i for every data point
  - But we'll move away from this soon..
- Can still use EM style algorithm to iteratively optimize
- For more info see Blei *et al.* (2003)

New method of learning: approximate inference model

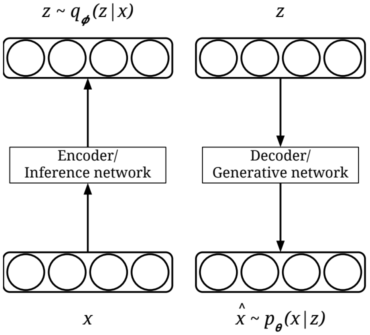
- Instead of having different variational parameters for each data point, fit a conditional parametric function
- The output of this function will be the parameters of the variational distribution $q(z|x)$
- Instead of q(z) we have $q_\phi(z|x)$
- ELBO becomes:

$$L(x;\theta,\phi) = \underbrace{\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x,z)]}_{\text{Expectation of joint distribution}} + \underbrace{\mathrm{H}(q_\phi(z|x))}_{\text{Entropy}}$$

Outline

Variational autoencoder

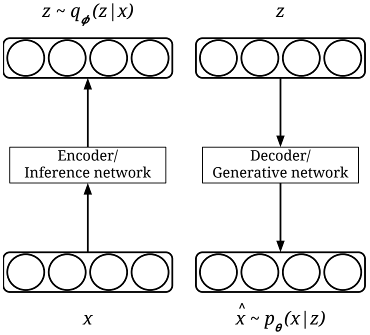
- *Encoder* network maps from image space to latent space
  - Outputs parameters of $q_\phi(z|x)$
- *Decoder* maps from latent space back into image space
  - Outputs parameters of $p_\theta(x|z)$



[Kingma & Welling (2013)]

Example

- *Encoder* network outputs mean and variance of Normal distribution
  - $q_\phi(z|x) = \mathcal{N}(\mu_\phi(x), \sigma_\phi(x))$
- *Decoder* network outputs mean (and optionally variance) of Normal distribution
  - $p_\theta(x|z) = \mathcal{N}(\mu_\theta(z), \mathbf{I})$



[Kingma & Welling (2013)]

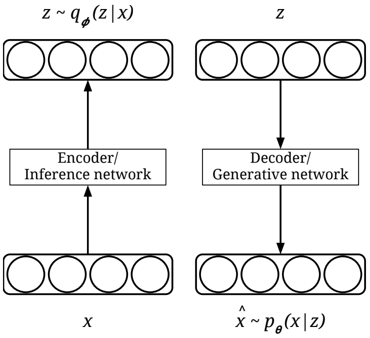
Variational autoencoder

- Rearranging the ELBO:

$$
\begin{aligned}
L(x;\theta,\phi) &= \int_z q(z|x)\log\frac{p(x,z)}{q(z|x)} \\
&= \int_z q(z|x)\log\frac{p(x|z)p(z)}{q(z|x)} \\
&= \int_z q(z|x)\log p(x|z) + \int_z q(z|x)\log\frac{p(z)}{q(z|x)} \\
&= \mathbb{E}_{q(z|x)}\log p(x|z) - \mathbb{E}_{q(z|x)}\log\frac{q(z|x)}{p(z)} \\
&= \underbrace{\mathbb{E}_{q(z|x)}\log p(x|z)}_{\text{Reconstruction term}} - \underbrace{D_{KL}(q(z|x)||p(z))}_{\text{Prior term}}
\end{aligned}
$$

Variational autoencoder

- Inference network outputs parameters of $q_\phi(z|x)$
- Generative network outputs parameters of $p_\theta(x|z)$
- Optimize θ and ϕ jointly by maximizing ELBO:



$$\color{blue}{L(x;\theta,\phi)} = \underbrace{\mathbb{E}_{q(z|x)} \log p(x|z)}_{\text{Reconstruction term}} - \underbrace{D_{KL}(q(z|x)||p(z))}_{\text{Prior term}}$$

Stochastic gradient variation bayes (SGVB) estimator

- Reparameterization trick : re-parameterize $z \sim q_\phi(z|x)$ as

$$z = g_\phi(x, \epsilon) \text{ with } \epsilon \sim p(\epsilon)$$

- For example, with a Gaussian can write $z \sim \mathcal{N}(\mu, \sigma^2)$ as

$$z = \mu + \epsilon\sigma^2 \text{ with } \epsilon \sim \mathcal{N}(0, 1)$$

[Kingma & Welling (2013); Rezende *et al.* (2014)]

Stochastic gradient variation bayes (SGVB) estimator

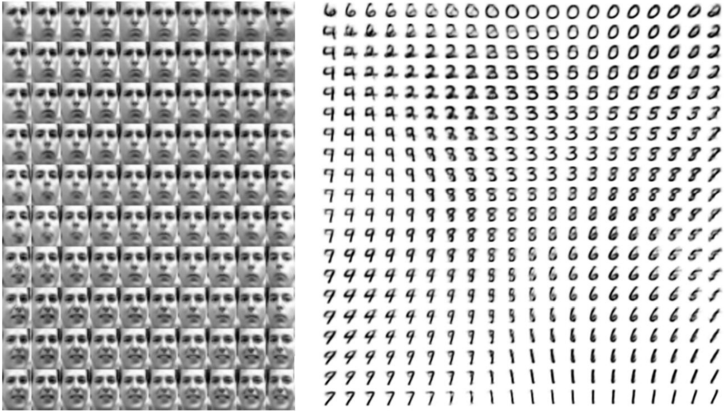
$$L(x;\theta,\phi) = \underbrace{\mathbb{E}_{q(z|x)} \log p(x|z)}_{\text{Reconstruction term}} - \underbrace{D_{KL}(q(z|x)||p(z))}_{\text{Prior term}}$$

- Using reparameterization trick we form Monte Carlo estimate of reconstruction term:

$$\begin{aligned}\mathbb{E}_{q_\phi(z|x)} \log p_\theta(x|z) &= \mathbb{E}_{p(\epsilon)} \log p_\theta(x|g_\phi(x,\epsilon)) \\ &\simeq \frac{1}{L}\sum_{i=1}^{L} \log p_\theta(x|g_\phi(x,\epsilon)) \quad \text{where } \epsilon \sim p(\epsilon)\end{aligned}$$

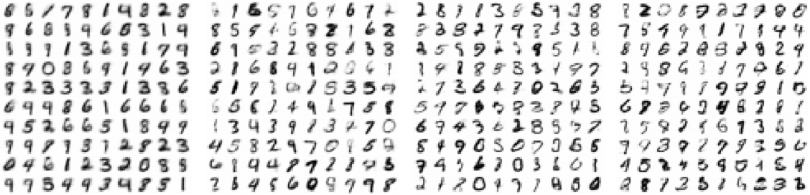
- KL divergence term can often be computed analytically (eg. Gaussian)

VAE learned manifold



[Kingma & Welling (2013)]

VAE samples

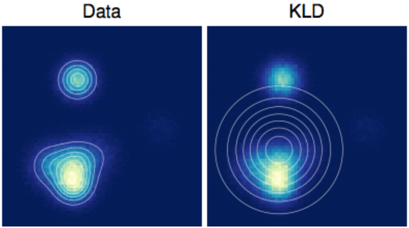


(a) 2-D latent space (b) 5-D latent space (c) 10-D latent space (d) 20-D latent space

[Kingma & Welling (2013)]

VAE tradeoffs

- Pros:
  - Theoretically pleasing
  - Optimizes bound on likelihood
  - Easy to implement
- Cons:
  - Samples tend to be blurry
    - Maximum likelihood minimizes $D_{KL}(p_{data}||p_{model})$



[Theis *et al.* (2016)]

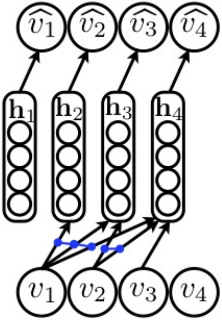
Autoregressive models

- Tractably model a joint distribution of the pixels in the image
- Learn to predict the next pixel given all the previously generated pixels
- Joint distribution of all pixels just product of conditionals:

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

Autoregressive models: NADE

- Recently gained popularity with Neural Autoregressive Density Estimator (NADE)
- Basic idea: use neural network to implement conditional probability functions



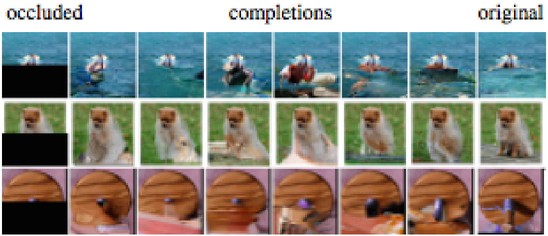
[Larochelle & Murray (2011)]

NADE samples

[Larochelle & Murray (2011)]

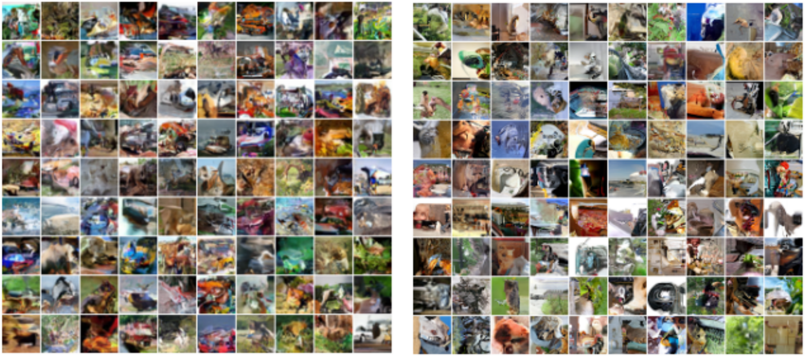
Autoregressive models: PixelRNN

- Use 2 dimensional RNN to model conditional probabilities
- More powerful model, still easy to train



[van den Oord *et al.* (2016)]

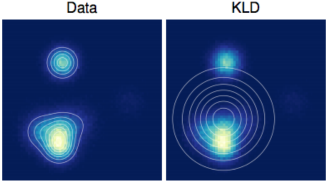
PixelRNN samples



[van den Oord *et al.* (2016)]

Autoregressive tradeoffs

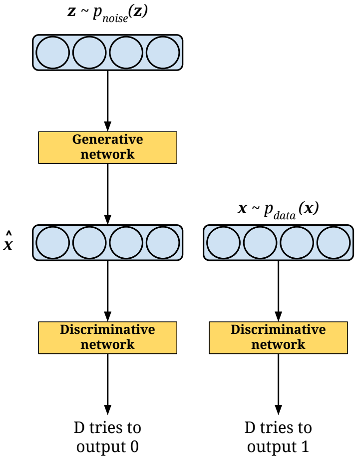
- Pros:
  - Tractable and exact likelihood
  - Simple maximum likelihood training
- Cons:
  - Inefficient sampling
  - No obvious way to get latent representation of image
  - Same issue of blurry samples due to optimizing log likelihood objective



[Theis *et al.* (2016)]

Generative adversarial networks

- Don't focus on optimizing $p(x)$, just learn to sample
- Two networks pitted against one another:
  - Generative model G captures data distribution
  - Discriminative model D distinguishes between real and fake samples



[Goodfellow *et al.* (2014)]

Generative adversarial networks

- D is trained to estimate the probability that a sample came from data distribution rather than G
- G is trained to maximize the probability of D making a mistake

$$\min_G \max_D \mathbb{E}_{x\sim p_{data(x)}} \log D(x) + \mathbb{E}_{z\sim p_{noise(z)}} \log(1 - D(G(z)))$$

Generative adversarial networks

$$\min_G \max_D \mathbb{E}_{x \sim p_{data(x)}} \log D(x) + \mathbb{E}_{z \sim p_{noise(z)}} \log(1 - D(G(z)))$$

- Resembles Jensen-Shannon divergence
- Alternating optimization procedure
- Training can (and often is) very unstable
- No obvious objective criterion to track during training

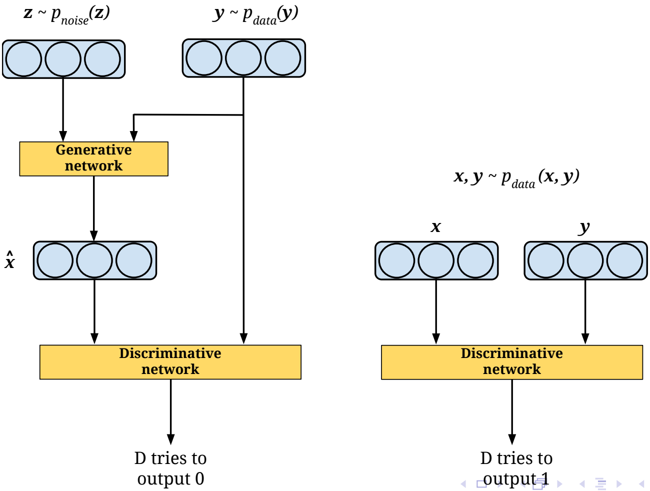
Conditional generative adversarial networks

- Can extend to case where both networks receive additional vector y (e.g. class label) of information:
- D now has to determine (i) if sample is real and (ii) correspondence

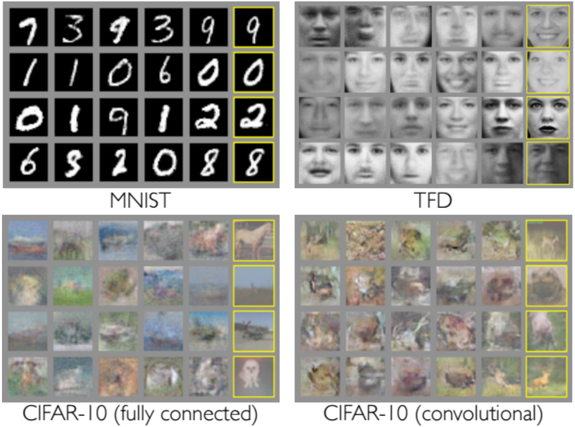
$$\min_G \max_D \quad \mathbb{E}_{x,y \sim p_{data(x,y)}} \log D(x,y) \quad + \\ \mathbb{E}_{z \sim p_{noise(z)}, y \sim p_{noise(y)}} \log(1 - D(G(z,y),y))$$

[Mirza & Osindero, 2014; Gauthier, 2014]

Conditional generative adversarial networks



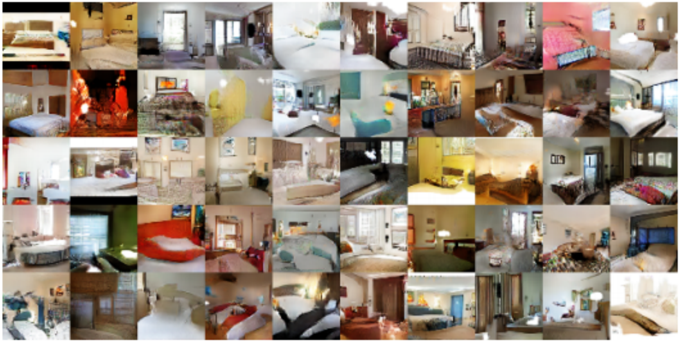
GAN samples (original paper)



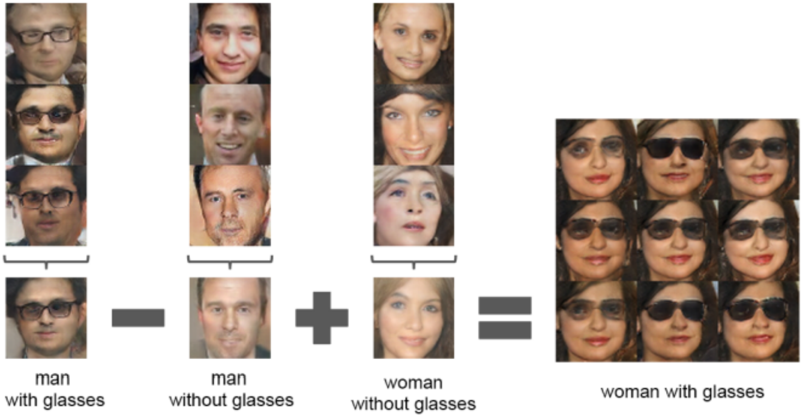
[Goodfellow *et al.* (2014)]

Deep convolutional generative adversarial networks (DCGAN)

- Radford *et al.* (2016) propose several tricks to make GAN training more stable

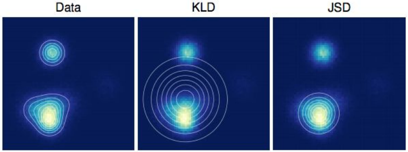


DCGAN vector arithmetic



GAN tradeoffs

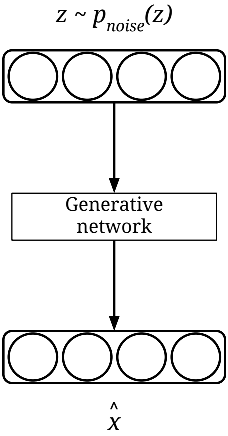
- Pros:
  - Very powerful model
  - High quality samples
- Cons:
  - Tricky to train (see: https://github.com/soumith/ganhacks)
  - Can ignore large parts of image space



[Theis *et al.* (2016)]

Generative moment matching networks

- Same idea as GANs, but different optimization method
- Match moments of data and generative distributions
- Maximum mean discrepancy
  - Estimator for answering whether two samples come from same distribution
- Evaluate MMD on generated samples



[Li *et al.* (2015); Dziugaite *et al.* (2015)]

Generative moment matching networks

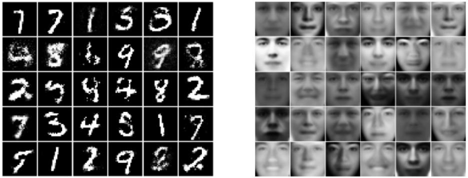
$$\mathcal{L}_{MMD^2} = \| \frac{1}{N} \sum_{i=1}^{N} \phi(x_i) - \frac{1}{M} \sum_{j=1}^{M} \phi(x_j) \|^2$$

$$= \frac{1}{N^2} \sum_{i=1}^{N} \sum_{i'=1}^{N} \phi(x_i)^\top \phi(x_{i'}) - \frac{1}{M^2} \sum_{j=1}^{M} \sum_{j'=1}^{M} \phi(x_j)^\top \phi(x_{j'})$$

$$- \frac{2}{NM} \sum_{i=1}^{N} \sum j = 1^M \phi(x_i)^\top \phi(x_j)$$

- Can make use of kernel trick
- If ϕ is identity, then matching means
- Complex ϕ can match higher order moments

GMMN samples



(a) GMMN MNIST samples

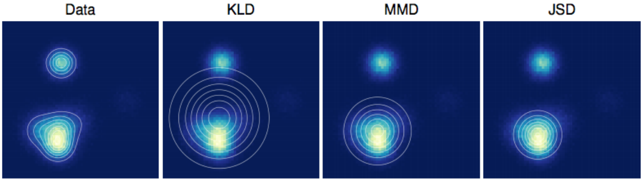
(b) GMMN TFD samples

(c) GMMN+AE MNIST samples

(d) GMMN+AE TFD samples

GMMN tradeoffs

- Pros:
  - Theoretically pleasing
- Cons:
  - Batch size very important
  - Samples aren't great (get better when combined with autoencoder)



[Theis *et al.* (2016)]

Summary

- Models that optimize log likelihood (VAEs, autoregressive) tend to put density where there is none
  - Results in blurry samples
- Models that optimize JS divergence (GANs) or MMD (GMMNs) have mode seeking tendencies
  - Results in crisp samples at expense of missing some of data space
- GANs currently produce best visual samples, but difficult to train

Outline

Evaluating a generative model: log likelihood

- Has generally been default criterion
- Makes sense when goal is density estimation
- Many approaches don't have tractable likelihood or it isn't explicitly represented
    - Have to resort to Parzen window estimates (Breuleux *et al.* , 2009) ... can be meaningless in high dimensional spaces
- Model can have poor log-likelihood and good samples (and vice versa)

Evaluating a generative model: sample quality

- If goal is image synthesis, this makes more sense
- How how to get objective measure of perceptual quality?
  - Human experiments
  - Look at responses from pretrained imagenet network (Salimans *et al.* , 2016; Augustus Odena, 2016)
  - Measure diversity in samples
- But a lookup table of training images will succeed here, need to be careful about memorizing

How to evaluate a generative model?

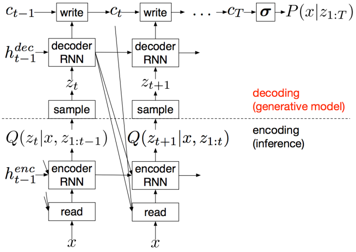
- Log likelihood on held out data
- Quality of samples
- Best: evaluate in context of particular application
- See Theis *et al.* (2016) for more details

Outline

DRAW: Deep Recurrent Attentive Writer

Basic idea:

- Iteratively construct image
- Observe image through sequence of glimpses



[Gregor *et al.* (2015)]

- Recurrent encoder and decoder
- Optimizes variational bound
- Attention mechanism determines:
  - Input region observed by encoder
  - Output region modified by decoder

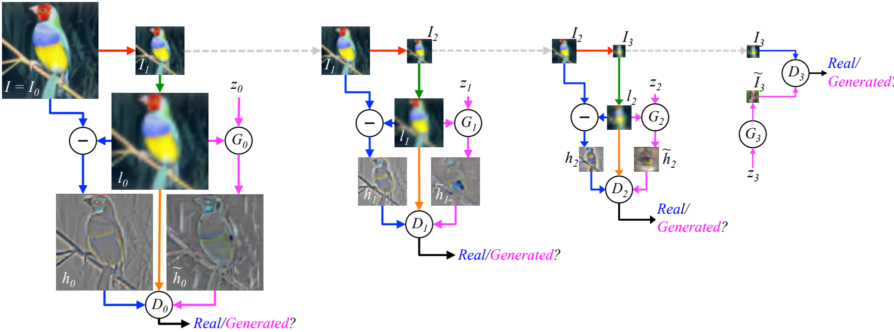
DRAW samples

Laplacian pyramid of generative adversarial networks

- Generate images in coarse-to-fine fashion
- Train conditional GAN on each scale, afterwards chain together

[Denton *et al.* (2015)]

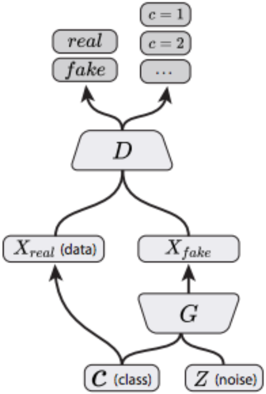
Laplacian pyramid of generative adversarial networks



Laplacian pyramid of generative adversarial networks

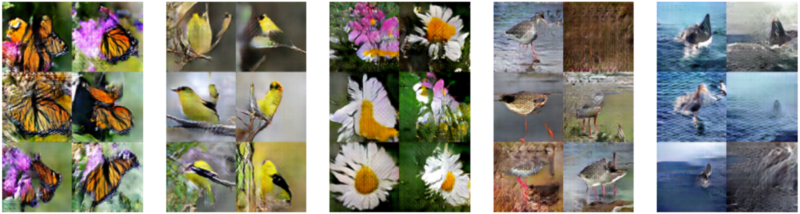
Conditional image synthesis with auxiliary classifier GANs

- Have discriminator not only predict real/fake but also classify images form 1 of k classes
- Condition generator on 1-hot encoding of class
- On Imagenet, train 100 different models each on 10 classes

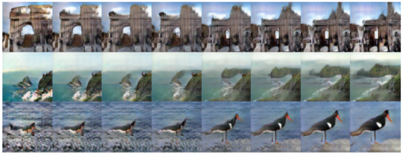


[Augustus Odena (2016)]

Conditional image synthesis with auxiliary classifier GANs

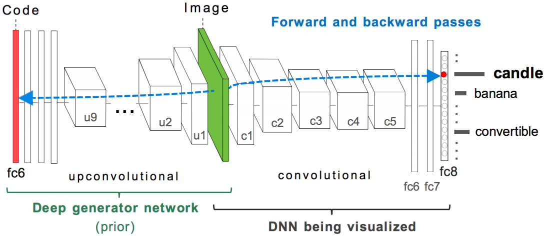


monarch butterfly | goldfinch | daisy | redshank | grey whale



Synthesizing preferred inputs for neurons in n.n via deep generator nets

- Generator G maps pre-trained classifier features to images (Dosovitskiy & Brox, 2016)
  - Combined pixel mse, feature mse and adversarial loss
- Optimize latent code to find image that highly activates specified class
- Not generative model (no prior over latent space, no implicit density model, no sampling procedure)



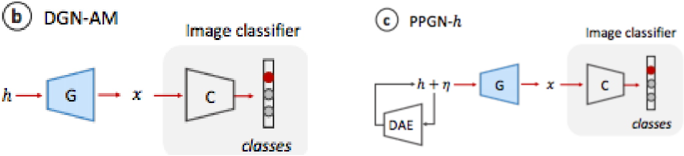
[Nguyen *et al.* (2016b)]

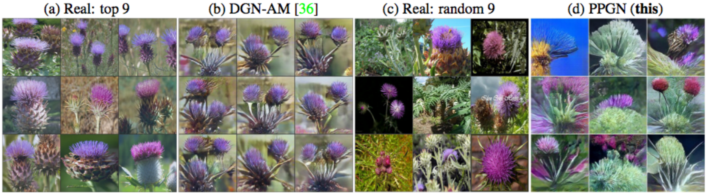
Synthesizing preferred inputs for neurons in n.n via deep generator nets

[Nguyen *et al.* (2016b)]

Plug & Play Generative Networks

- Turn activation maximization model into generative model
- Synthesized images are diverse and high quality

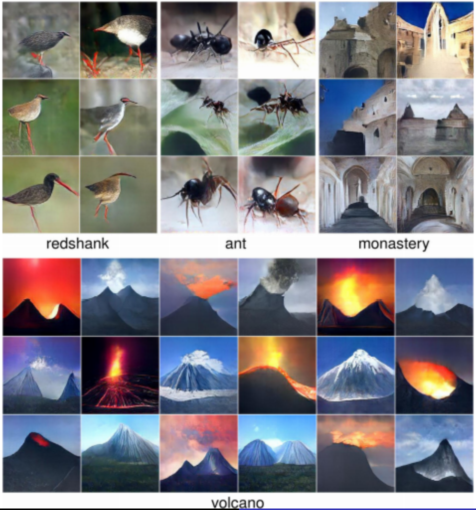




cardoon

[Nguyen *et al.* (2016a)]

Plug & Play Generative Networks



Plug & Play Generative Networks

- Can plug in image captioning network instead of classifier (left)
- Can use different classifier from one generator was trained on (right)

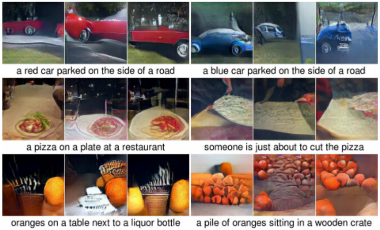
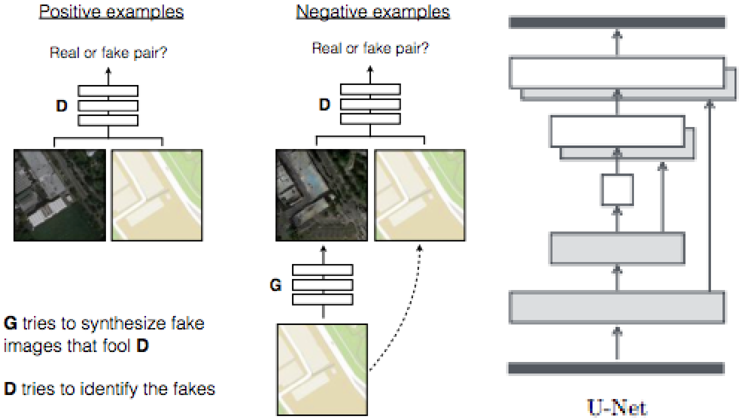


Figure 4: Images synthesized conditioned on MIT Places [62] classes instead of ImageNet classes.

Image-to-Image translation with conditional adversarial nets



[Isola *et al.* (2016)]

Image-to-Image translation with conditional GANs

Facade labels → photos:

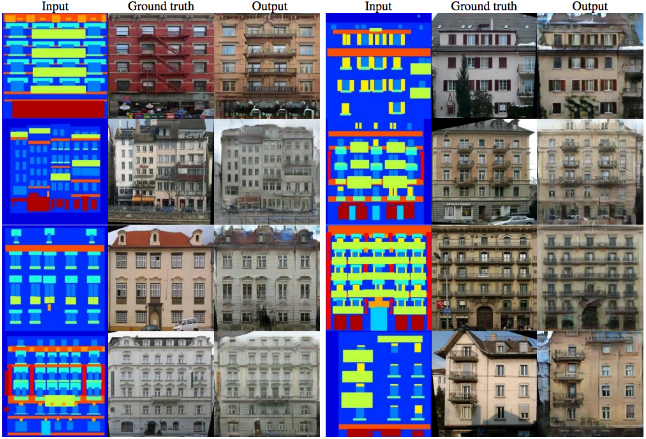


Image-to-Image translation with conditional GANs

Day $\rightarrow$ night:

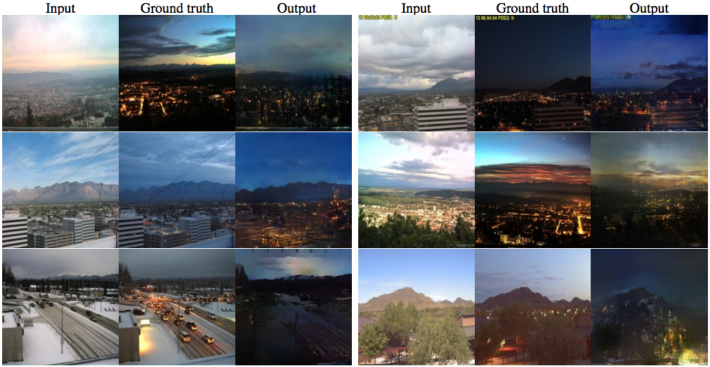
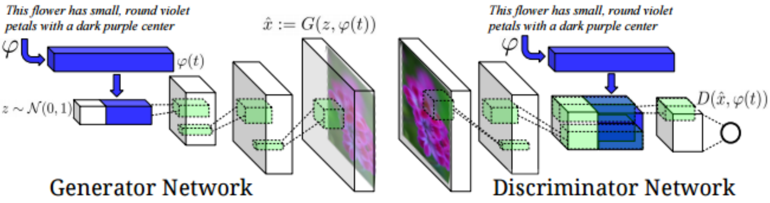


Image-to-Image translation with conditional GANs

Edges → handbags:

Generative adversarial text to image synthesis

- Conditional GAN model
- Generation conditioned on text features encoded by a hybrid character-level recurrent convnet neural network.



[Reed *et al.* (2016a)]

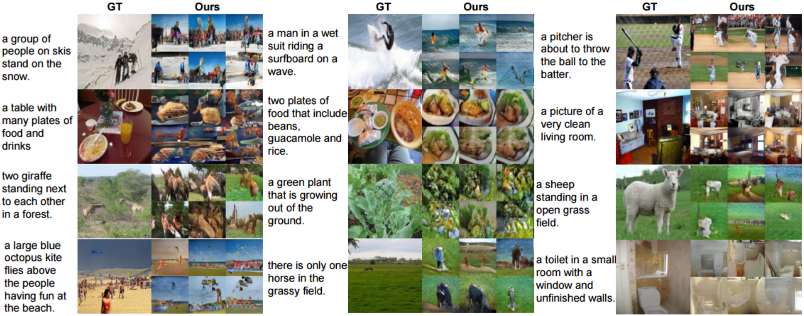
Generative adversarial text to image synthesis

Caption	Image
a pitcher is about to throw the ball to the batter	
a group of people on skis stand in the snow	
a man in a wet suit riding a surfboard on a wave	

Generative adversarial text to image synthesis

Caption	**Image**
this vibrant red bird has a pointed black beak	
this bird is yellowish orange with black wings	
the bright blue bird has a white colored belly	

Generative adversarial text to image synthesis



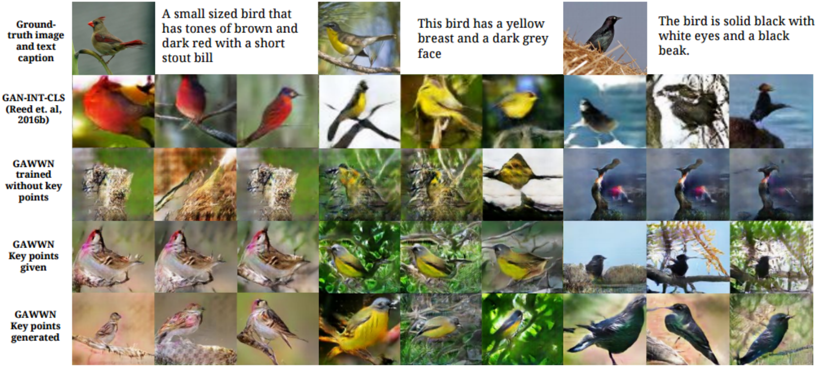
Learning what and where to draw

- Generative Adversarial What-Where Network (GAWWN)
- Condition on content (text) and location (keypoints or bounding box)

[Reed *et al.* (2016b)]

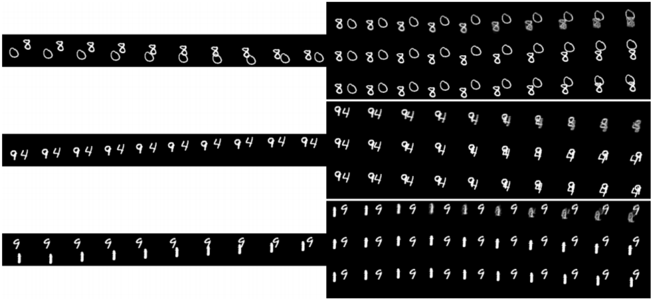
Learning what and where to draw

Conditioning on location improves image quality



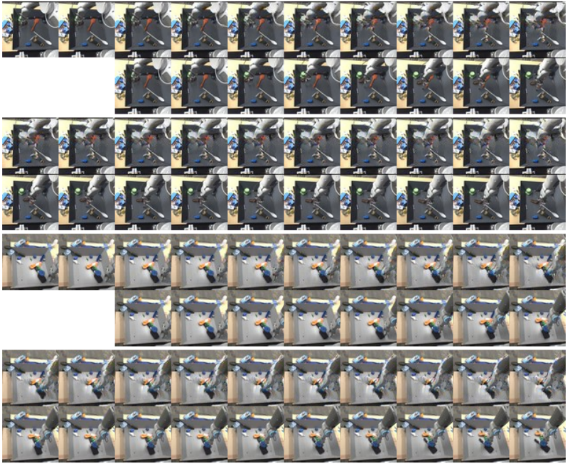
Video pixel networks

- Extension of autoregressive model to video



[Kalchbrenner *et al.* (2016)]

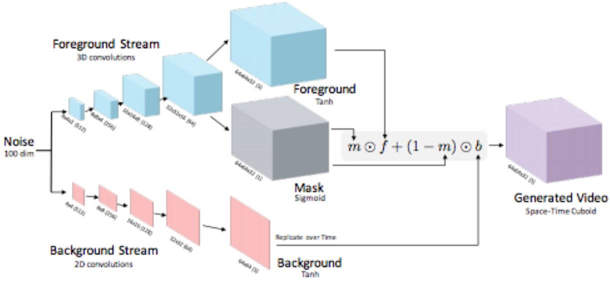
Video pixel networks



[Kalchbrenner *et al.* (2016)]

Generating videos with scene dynamics

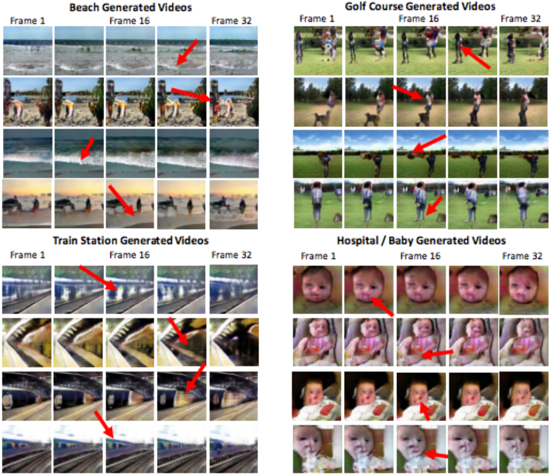
- Two-stream generative adversarial network model
- Foreground and background modeled separately



[Vondrick *et al.* (2016)]

Generating videos with scene dynamics

Generating videos with scene dynamics



References I

Augustus Odena, Christopher Olah, Jonathon Shlens. 2016. Conditional Image Synthesis With Auxiliary Classifier GANs. *arXiv preprint:1610.09585*.

Blei, David M., Ng, Andrew Y., & Jordan, Michael I. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*.

Breuleux, O., Bengio, Y., & Vincent, P. 2009. Unlearning for better mixing. *Technical report, Universite de Montreal*.

Denton, Emily, Chintala, Soumith, Szlam, Arthur, & Fergus, Rob. 2015. Deep generative image models using a laplacian pyramid of adversarial networks. *In: NIPS*.

References II

Dosovitskiy, Alexey, & Brox, Thomas. 2016. Generating Images with Perceptual Similarity Metrics based on Deep Networks.

Dziugaite, Gintare Karolina, Roy, Daniel M., & Ghahramani, Zoubin. 2015. Training generative neural networks via Maximum Mean Discrepancy optimization. *In: UAI*.

Goodfellow, Ian J., Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron C., & Bengio, Yoshua. 2014. Generative adversarial networks. *In: NIPS*.

Gregor, Karol, Danihelka, Ivo, Graves, Alex, & Wierstra, Daan. 2015. DRAW: A Recurrent Neural Network For Image Generation. *arXiv preprint:1502.04623*.

References III

Isola, Phillip, Zhu, Jun-Yan, Zhou, Tinghui, & Efros, Alexei A. 2016. Image-to-Image Translation with Conditional Adversarial Nets. *arXiv preprint:1611.07004*.

Kalchbrenner, Nal, van den Oord, Aaaron, Danihelka, Karen Simonyan Ivo, Vinyals, Oriol, Graves, Alex, & Kavukcuoglu, Koray. 2016. Video Pixel Networks. *arXiv preprint:1610.00527*.

Kingma, Diederik P, & Welling, Max. 2013. Auto-encoding variational bayes. *arXiv preprint:1312.6114*.

Larochelle, Hugo, & Murray, Iain. 2011. The Neural Autoregressive Distribution Estimator. *In: Artificial Intelligence and Statistics*.

References IV

Li, Yujia, Swersky, Kevin, & Zemel, Richard. 2015. Generative Moment Matching Networks. *In: ICML.*

Nguyen, Anh, Yosinski, Jason, & Bengio, Yoshua. 2016a. Plug & Play Generative Networks: Conditional Iterative Generation of Images in Latent Space. *arXiv preprint:1612.00005.*

Nguyen, Anh, Dosovitskiy, Alexey, Yosinski, Jason, Brox, Thomas, & Clune, Jeff. 2016b. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks.

Radford, Alec, Metz, Luke, & Chintala, Soumith. 2016. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *In: ICLR.*

References V

Reed, Scott, Akata, Zeynep, Yan, Xinchen, Logeswaran, Lajanugen, Schiele, Bernt, & Lee, Honglak. 2016a. Generative Adversarial Text to Image Synthesis. *In: ICML*.

Reed, Scott, Akata, Zeynep, Mohan, Santosh, Tenka, Samuel, Schiele, Bernt, & Lee, Honglak. 2016b. Learning What and Where to Draw. *In: NIPS*.

Rezende, Danilo Jimenez, Mohamed, Shakir, & Wierstra, Daan. 2014. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint:1401.4082*.

Salimans, Tim, Goodfellow, Ian, Zaremba, Wojciech, Cheung, Vicki, Radford, Alec, & Chen, Xi. 2016. Improved Techniques for Training GANs. *In: NIPS*.

References VI

Theis, Lucas, van den Oord, Aaron, & Bethge, Matthias. 2016. A note on the evaluation of generative models. *In: ICLR*.

van den Oord, Aaron, Kalchbrenner, Nal, & Kavukcuoglu, Koray. 2016. Pixel Recurrent Neural Networks. *In: ICML*.

Vondrick, Carl, Pirsiavash, Hamed, & Torralba, Antonio. 2016. Generating Videos with Scene Dynamics. *In: NIPS*.