CSED703R: Deep Learning for Visual Recognition (2016S)

Lecture 14: Deep Generative Learning

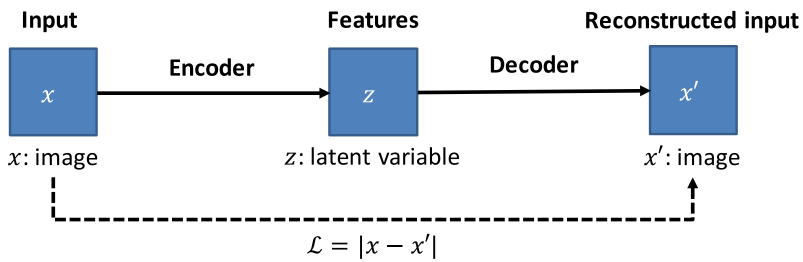
Bohyung Han
Computer Vision Lab.
bhhan@postech.ac.kr

Generative Modeling by Neural Networks

- Variational Auto Encoder (VAE)
  - Neural networks with continuous latent variables
  - Encoder: approximate a posterior distribution
  - Decoder: stochastically reconstruct the data from the latent variables
- Generative Adversarial Networks (GANs)
  - Generator: generating samples using a uniform distribution
  - Discriminator: discriminating between real and generated images
- Deep Boltzmann Machines (DBMs)
- Deep Belief Networks (DBNs)

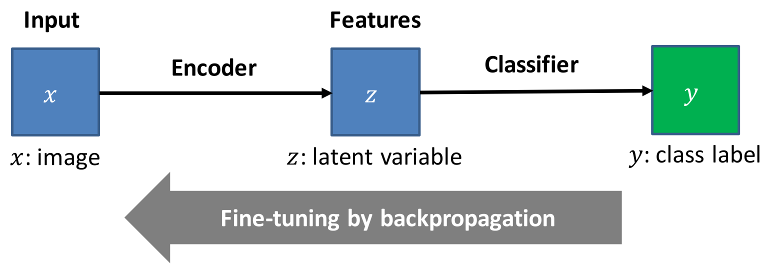
Autoencoder

- Traditional autoencoders
  - Feature learning
  - Learning with the reconstruction error
- Basic pipeline



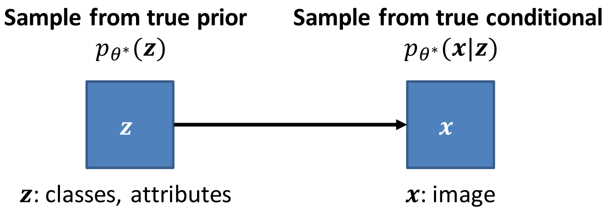
Autoencoder

- Traditional autoencoders
  - Feature learning
  - Learning with the reconstruction error
- Supervised feature learning



Variational Autoencoder

- Characteristics
  - A Bayesian approach on an autoencoder: sample generation
  - Estimating model parameter θ without access to latent variable z
- Problem scenario



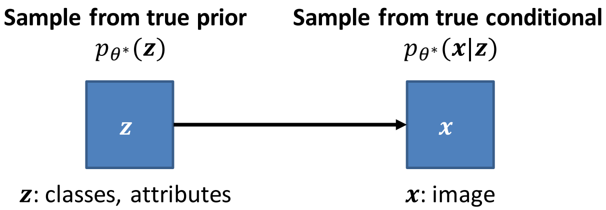
  - A value $\mathbf{z}^{(i)}$ is generated from a prior distribution $p_{\theta^*}(\mathbf{z})$.
  - A value $\mathbf{x}^{(i)}$ is generated from a conditional distribution $p_{\theta^*}(\mathbf{x}|\mathbf{z})$.

True parameters θ^* and the values of the latent variables $\mathbf{z}^{(i)}$ are hidden.

D. Kingma, M. Welling: **Auto-Encoding Variational Bayes**. ICLR 2014

Variational Autoencoder

- Characteristics
  - A Bayesian approach on an autoencoder: sample generation
  - Estimating model parameter θ without access to latent variable z
- Limitation

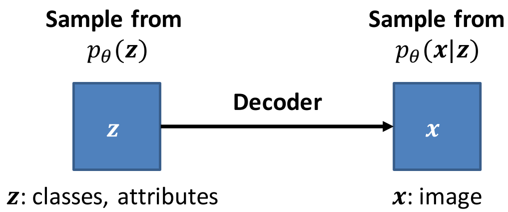


  - It is difficult to estimate $p_{\theta^*}(\mathbf{z})$ and $p_{\theta^*}(\mathbf{x}|\mathbf{z})$ in practice.
  - We need to approximate these distributions.

D. Kingma, M. Welling: **Auto-Encoding Variational Bayes**. ICLR 2014

Variational Autoencoder

- Characteristics
  - A Bayesian approach on an autoencoder: sample generation
  - Estimating model parameter θ without access to latent variable z
- Main idea

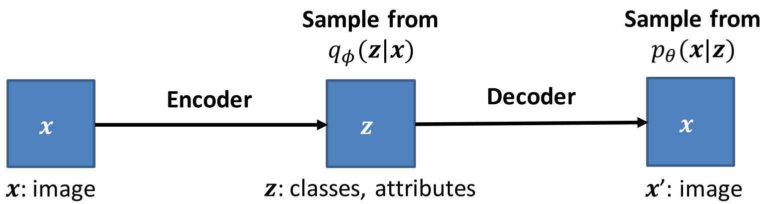


  - Assume the prior $p_{\theta}(\mathbf{z})$ is a unit Gaussian.
  - Assume the likelihood $p_{\theta}(\mathbf{x}|\mathbf{z})$ is a diagonal Gaussian.
  - Decoder estimates mean and variance of $p_{\theta}(\mathbf{x}|\mathbf{z})$.

D. Kingma, M. Welling: **Auto-Encoding Variational Bayes**. ICLR 2014

Variational Autoencoder

- Characteristics
  - A Bayesian approach on an autoencoder: sample generation
  - Estimating model parameter θ without access to latent variable z
- Main idea



  - Encoder estimates mean and variance of $q_{\phi}(\mathbf{z}|\mathbf{x})$.
  - Decoder estimates mean and variance of $p_{\theta}(\mathbf{x}|\mathbf{z})$.

Maximize the lower bound of the marginal likelihood.

D. Kingma, M. Welling: **Auto-Encoding Variational Bayes**. ICLR 2014

Optimization

- Variational lower bound of marginal likelihood

$$\log p_\theta(\boldsymbol{x}^{(i)}) = E_{q_\phi(z|x)}[\log p_\theta(\boldsymbol{x}^{(i)})] = D_{KL}\left(q_\phi(\boldsymbol{z}|\boldsymbol{x}^{(i)})||p_\theta(\boldsymbol{z}|\boldsymbol{x}^{(i)})\right) + \mathcal{L}(\theta,\phi;\boldsymbol{x}^{(i)})$$

KL divergence of the approximate from the true posterior (positive)

Variational lower bound on the marginal likelihood

- Maximization of variational lower bound

$$\mathcal{L}(\theta,\phi;\boldsymbol{x}) = E_{q_\phi(z|x)}[-\log q_\phi(\boldsymbol{z}|\boldsymbol{x}) + \log p_\theta(\boldsymbol{x},\boldsymbol{z})]$$

$$= E_{q_\phi(z|x)}[-\log q_\phi(\boldsymbol{z}|\boldsymbol{x}) + \log p(\boldsymbol{z}) + \log p_\theta(\boldsymbol{x}|\boldsymbol{z})]$$

$$= -D_{KL}\left(q_\phi(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z})\right) + E_{q_\phi(z|x)}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})]$$

Regularization term

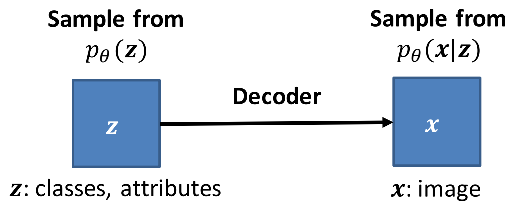
Reconstruction term

Train with error backpropagation with reprametrization

D. Kingma, M. Welling: **Auto-Encoding Variational Bayes**. ICLR 2014

Variational Autoencoder

- Characteristics
  - A Bayesian approach on an autoencoder: generate samples
  - Estimating model parameter θ without access to latent variable z
- Inference

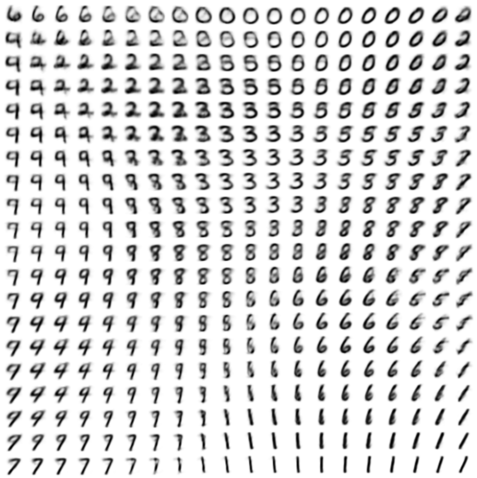


  - Decoder samples mean and variance from $p_\theta(\boldsymbol{z})$.
  - Decoder samples $\boldsymbol{x}$ from the mean and variance.

D. Kingma, M. Welling: **Auto-Encoding Variational Bayes**. ICLR 2014

Learned Manifold

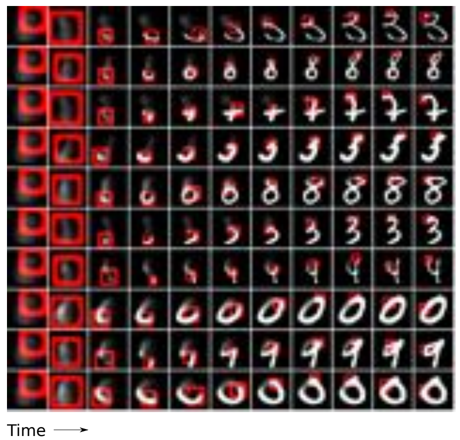
Visualizations of learned manifold for generative models with 2D latent space



D. Kingma, M. Welling: **Auto-Encoding Variational Bayes**. ICLR 2014

DRAW

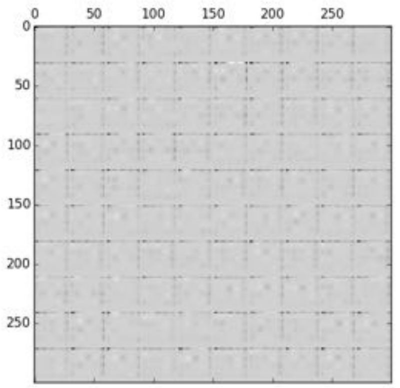
- Deep Recurrent Attentive Writer (DRAW)
  - Combines a spatial attention mechanism with a sequential variational auto-encoding framework
  - Allows for the iterative construction of complex images



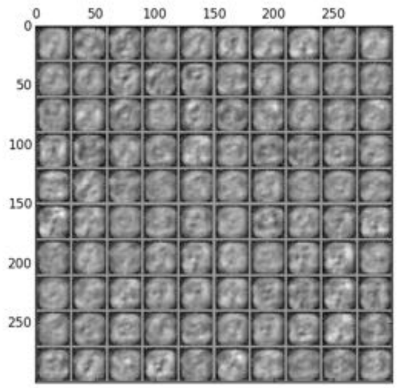
[Gregor15] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, D. Wierstra, **DRAW: A Recurrent Neural Network For Image Generation**. ICML 2015

DRAW

- Deep Recurrent Attentive Writer (DRAW)



DRAW with attention



DRAW without attention

[Gregor15] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, D. Wierstra, **DRAW: A Recurrent Neural Network For Image Generation**. ICML 2015

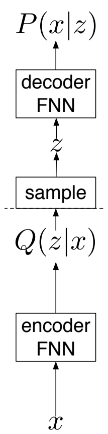
Draw

- Architecture and training
  - Encoder: compressing the real images presented during training
  - Decoder: reconstituting images after receiving codes
  - Trained end-to-end with SGD
  - Loss function: variational upper bound on the log-likelihood of data
  - It belongs to the family of *variational auto-encoders*.
- Characteristics
  - Progressive refinement: implemented by RNN (LSTM)
  - Spatial attention

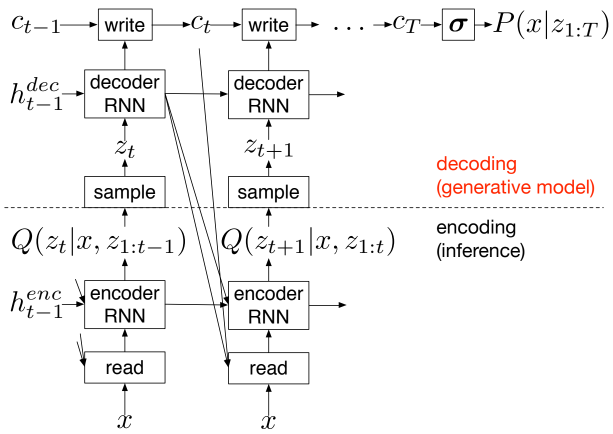
[Gregor15] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, D. Wierstra, **DRAW: A Recurrent Neural Network For Image Generation**. ICML 2015

Architecture

Conventional VAE

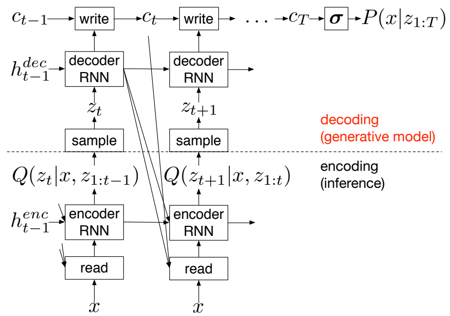


DRAW network



Operations in Network

DRAW network



Error image

$$\hat{x}_t = x - \sigma(c_{t-1})$$

$$r_t = \text{read}(x_t, \hat{x}_t, h^{\text{dec}}_{t-1})$$

VAE

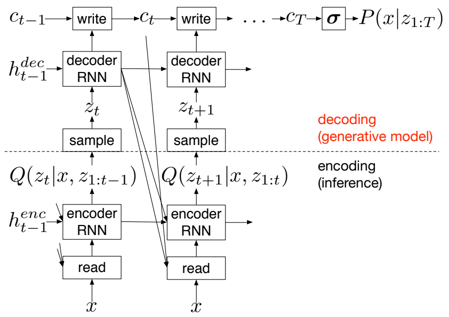
$$\boxed{\begin{aligned} h^{\text{enc}}_t &= \text{RNN}^{\text{enc}}(h^{\text{enc}}_{t-1}, [r_t, h^{\text{dec}}_{t-1}]) \\ z_t &\sim Q(Z_t|h^{\text{enc}}_t) = Q(Z_t|x, z_{1:t}) \\ h^{\text{dec}}_t &= \text{RNN}^{\text{dec}}(h^{\text{dec}}_{t-1}, z_t) \end{aligned}}$$

$$c_t = c_{t-1} + \text{write}(h^{\text{dec}}_t)$$

- VAE
  - It is extremely difficult to sample images in a pixel space.
  - We hope Q looks like a true distribution.

Operations in Network

DRAW network



$$\hat{x}_t = x - \sigma(c_{t-1})$$
$$r_t = \mathrm{read}(x_t, \hat{x}_t, h_{t-1}^{\mathrm{dec}})$$
$$h_t^{\mathrm{enc}} = \mathrm{RNN}^{\mathrm{enc}}(h_{t-1}^{\mathrm{enc}}, [r_t, h_{t-1}^{\mathrm{dec}}])$$
$$z_t \sim Q(Z_t | h_t^{\mathrm{enc}}) = Q(Z_t | x, z_{1:t})$$
$$h_t^{\mathrm{dec}} = \mathrm{RNN}^{\mathrm{dec}}(h_{t-1}^{\mathrm{dec}}, z_t)$$
$$c_t = c_{t-1} + \mathrm{write}(h_t^{\mathrm{dec}})$$

Without attention:

$$\mathrm{read}(x_t, \hat{x}_t, h_{t-1}^{\mathrm{dec}}) = [x, \hat{x}]$$
$$\mathrm{write}(h_t^{\mathrm{dec}}) = W(h_t^{\mathrm{dec}})$$

With attention:

$$\mathrm{read}(x_t, \hat{x}_t, h_{t-1}^{\mathrm{dec}}) = \gamma[F_Y x F_X^T, F_Y \hat{x} F_X^T]$$
$$\mathrm{write}(h_t^{\mathrm{dec}}) = \frac{1}{\hat{\gamma}} \hat{F}_Y w_t \hat{F}_X^T = \frac{1}{\hat{\gamma}} \hat{F}_Y W(h_t^{\mathrm{dec}}) \hat{F}_X^T$$

Training

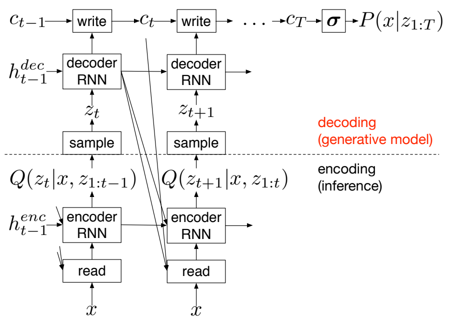
- Loss function
  - the expectation of the sum of the reconstruction and latent lose

$$\mathcal{L} = \langle \mathcal{L}^x + \mathcal{L}^z \rangle_{z\sim Q}$$

  - Reconstruction loss: $\mathcal{L}^x = -\log D(x|c_T)$
  - Latent loss: $\mathcal{L}^z = \sum_{t=1}^{T} KL(Q(Z_t|h_t^{\mathrm{enc}}) \,||\, P(Z_t))$
- Backpropagation
  - Stochastic gradient descent
  - Optimizing using a single sample of z for each step

Image Generation

DRAW network



$$\hat{z}_t \sim P(Z_t)$$
$$\tilde{h}_t^{\mathrm{dec}} = \mathrm{RNN}^{\mathrm{dec}}(\tilde{h}_{t-1}^{\mathrm{enc}}, \hat{z}_t)$$
$$\tilde{c}_t = \tilde{c}_{t-1} + \mathrm{write}(\tilde{h}_t^{\mathrm{dec}})$$
$$\hat{x}_t \sim D(X|\tilde{c}_T)$$

Results

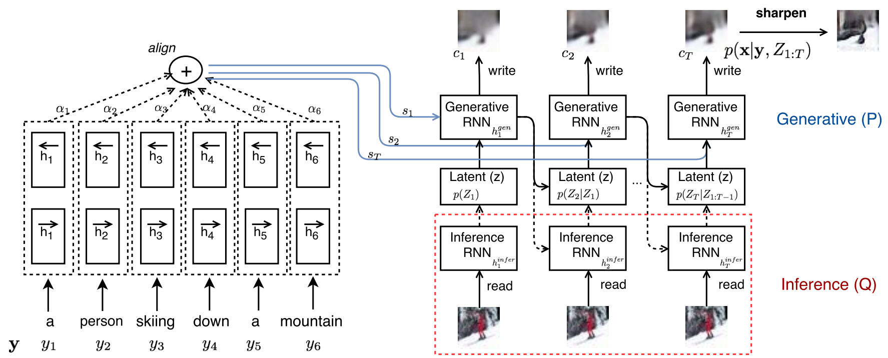
alignDRAW

- Image generation from caption
  - Extension of DRAW
  - A generative model of images from captions using a soft attention mechanism
  - Aligning between the input captions and generating canvas
  - Iteratively draws patches on a canvas, while attending to the relevant words in the description

[Mansimov16] E. Mansimov, E. Parisotto, J. L. Ba, R. Salakhutdinov: **Generating Images from Captions with Attention**, ICLR 2016

alignDRAW

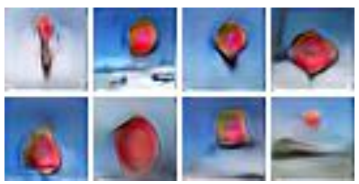
- Image generation from caption



[Mansimov16] E. Mansimov, E. Parisotto, J. L. Ba, R. Salakhutdinov: **Generating Images from Captions with Attention**, ICLR 2016

alignDraw

- Generating unrealistic images
  - But conforming to input captions



A stop sign is flying in blue skies.

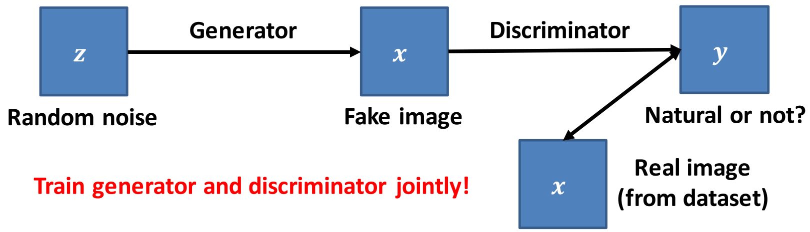
A herd of elephants flying in the blue skies.

A toilet seat sits open in the grass field.

A person skiing on sand clad vast desert.

Generative Adversarial Networks

- Two models
  - A generative model G that captures the data distribution
  - A discriminative model D that estimates the probability that a sample came from the training data rather than G
- Objective
  - Maximize the probability of D making a mistake
  - Trained with backpropagation



[Goodfellow14] I. J. Goodfellow, et al.: **Generative Adversarial Nets**, NIPS 2014

Optimization

- Objective

$$\min_G \max_D V(D,G) = E_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x};\theta_d)] + E_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\left[\log\left(1 - D\left(G\left(\boldsymbol{z};\theta_g\right);\theta_d\right)\right)\right]$$

D and G play the two-player minimax game with value function $V(D,G)$

Simultaneously training G to minimize $\log\left(1 - D\left(G\left(\boldsymbol{z};\theta_g\right);\theta_d\right)\right)$

- $p_{\boldsymbol{z}}(\boldsymbol{z})$: prior on input noise variables
- $G(\boldsymbol{z};\theta_g)$
  - Representing a mapping from $\boldsymbol{z}$ to data space
  - A differentiable function represented by a multilayer perceptron with parameters θ_g
- $D(\boldsymbol{x};\theta_d)$:
  - Representing the probability that $\boldsymbol{x}$ came from data rather than generator distribution p_g

Optimization

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**
  for k steps **do**
  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)},\ldots,\boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Sample minibatch of m examples $\{\boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
  - Update the discriminator by ascending its stochastic gradient:

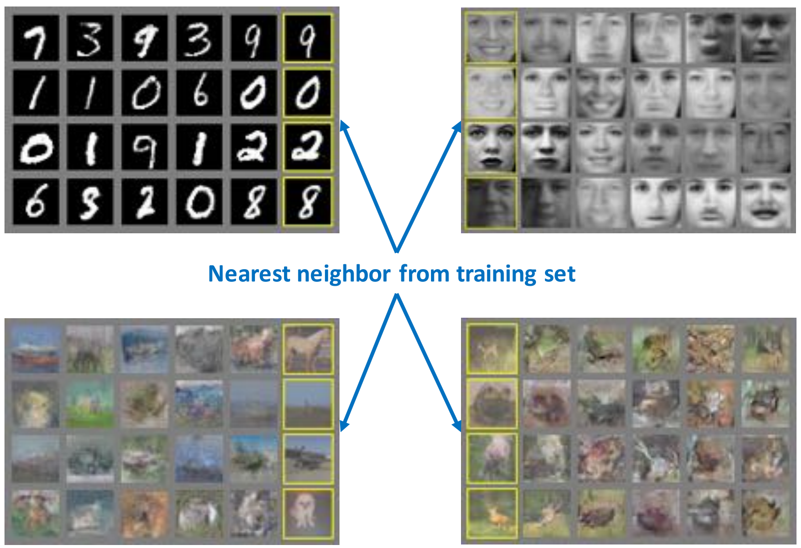
$$\nabla_{\theta_d}\frac{1}{m}\sum_{i=1}^{m}\left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)\right].$$

  end for
  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)},\ldots,\boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g}\frac{1}{m}\sum_{i=1}^{m}\log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for
The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

Generated Examples

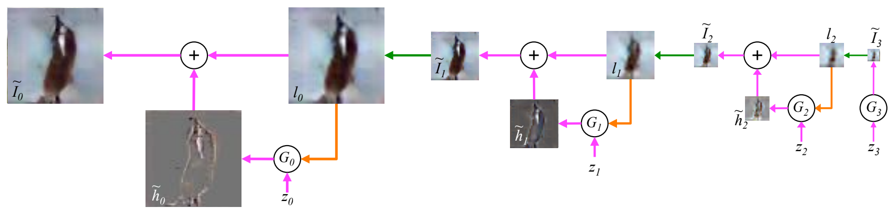


Laplacian Generative Adversarial Networks

- Generating high-resolution images

$$\hat{I}_k = u(I_{k+1}) + \hat{h}_k(I_{k+1}) = u(I_{k+1}) + G_k\left(z_k, u(I_{k+1})\right)$$

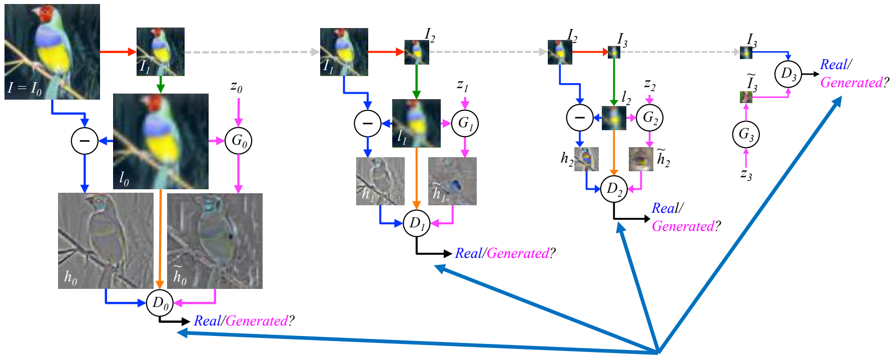
Upsampling **Generative model**



[Denton15] E. Denton, S. Chintalal, A. Szlam, R. Fergus: **Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks**, NIPS 2015

Laplacian Generative Adversarial Networks

- Training
  - The opposite direction



[Denton15] E. Denton, S. Chintalal, A. Szlam, R. Fergus: **Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks**, NIPS 2015

Results

Nearest neighbor from training set

VAE vs. GAN

- Variational Auto Encoders (VAEs)
  - Require differentiation through the hidden units
  - Cannot have discrete latent variables
  - MCMC-based inference
- Generative Adversarial Networks (GANs)
  - Requires differentiation through the visible units
  - Cannot model discrete data
  - Learned approximate inference