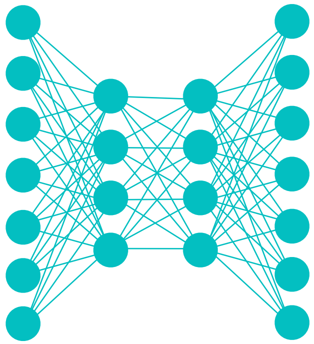
Lecture Notes for

Neural Networks and Machine Learning

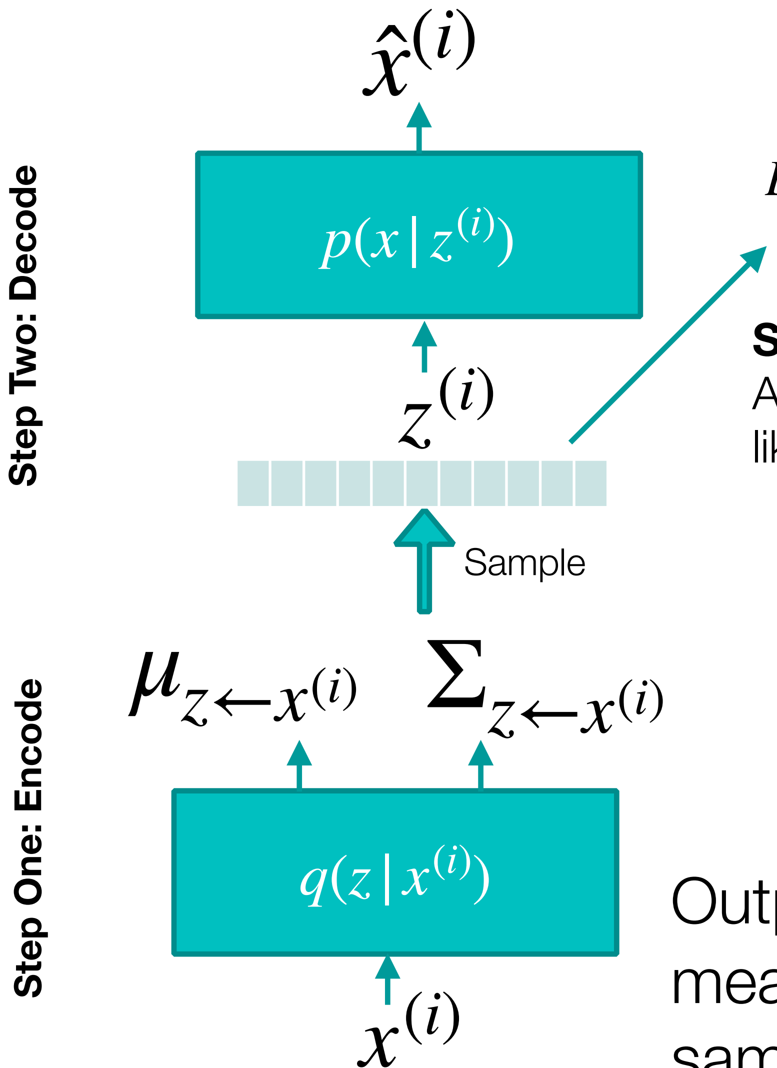
Generative Networks

and

Auto-Encoding Generators



Need a new formulation



Step Three: Make conditional p and q Similar

$$D_{KL}\left[q(z|x^{(i)}) \| p(z|x^{(i)})\right] = \mathbf{E}_{q(z|x)}\left[\log\left(\frac{q(z|x^{(i)})}{p(z|x^{(i)})}\right)\right]$$

Step Four: Use Variational Inference
Assume that a family of distributions can maximize likelihood of observing $x^{(i)}$:

$$\log p(x)_{\forall i} \approx \mathbf{E}_{\mathbf{z} \leftarrow q(z|x^{(i)})}\left[\log p(x^{(i)})\right]$$

Max Log Lik:: maximize probability of observed $x^{(i)}$
given family of distributions q
hope this is a good approximation

Output of network, q, are the mean and covariance for sampling a variable z

Need a new formulation

$$\log p(x)_{\forall i} \approx \mathbf{E}_{\mathbf{z} \leftarrow q(z|x)} \left[\log p(x^{(i)}) \right]$$ Maximize!

$$= \mathbf{E}_q \left[\log \frac{p(x^{(i)} | z) p(z)}{p(z | x^{(i)})} \frac{q(z | x^{(i)})}{q(z | x^{(i)})} \right]$$

Variational + multiply by one

$p(z | x^{(i)})$ this is still a problem

$$= \mathbf{E}_q \left[\log p(x^{(i)} | z) \right] + \mathbf{E}_q \left[\log \frac{p(z)}{q(z | x^{(i)})} \right] + \mathbf{E}_q \left[\log \frac{q(z | x^{(i)})}{p(z | x^{(i)})} \right]$$

$$= \mathbf{E}_q \left[\log p(x^{(i)} | z) \right] - \mathbf{E}_q \left[\log \frac{q(z | x^{(i)})}{p(z)} \right] + \mathbf{E}_q \left[\log \frac{q(z | x^{(i)})}{p(z | x^{(i)})} \right]$$

$$= \mathbf{E}_q \left[\log p(x^{(i)} | z) \right] - D_{KL} \left[q(z | x^{(i)}) \| p(z) \right] + \cancel{D_{KL} \left[q(z | x^{(i)}) \| p(z | x^{(i)}) \right]}$$

always non-negative

$$\log p(x)_{\forall i} \geq \mathbf{E}_q \left[\log p(x^{(i)} | z) \right] - D_{KL} \left[q(z | x^{(i)}) \| p(z) \right]$$ Will Maximize Lower Bound

Can we motivate this in a different way?

The Loss Function

Maximize through
Error of Reconstruction
Same as minimizing cross entropy

want $p(z)$ to be $\mathcal{N}(\mu = 0, \Sigma = I)$
because it makes nice latent space

$$q(z \mid x^{(i)}) \rightarrow (\mu_{z|x}, \Sigma_{z|x}) \qquad p(z) \rightarrow \mathcal{N}(0,1)$$

$$\geq \mathbf{E}_{q(z|x^{(i)})}\left[\log p(x^{(i)} \mid z)\right] - D_{KL}\left[q(z \mid x^{(i)}) \| p(z)\right]$$

$$D_{KL}\big((\mu, \Sigma) \| \mathcal{N}(0,1)\big) = \frac{1}{2}\left(\text{tr}(\Sigma) + \mu \cdot \mu^T - \underbrace{k}_{|z|} - \log\big(\det(\Sigma)\big)\right)$$

Can get this by manipulating
the KL for normal distribution

Determinant of diagonal
matrix is simple.
Motivates diagonal
covariance…

$$= \frac{1}{2}\left(\sum_k \Sigma_{k,k} + \sum_k \mu_k^2 - \sum_k 1 - \log\left(\prod_k \Sigma_{k,k}\right)\right)$$

$$= \frac{1}{2}\left(\sum_k \Sigma_{k,k} + \sum_k \mu_k^2 - \sum_k 1 - \sum_k \log \Sigma_{k,k}\right)$$

$$= \frac{1}{2}\sum_k \left(\Sigma_{k,k} + \mu_k^2 - 1 - \log \Sigma_{k,k}\right)$$

The Covariance Output

$$\geq \mathbf{E}_{q(z|x^{(i)})}\left[\log p(x^{(i)}\,|\,z)\right] - D_{KL}\left[q(z\,|\,x^{(i)})\|p(z)\right]$$

Maximize through
Error of Reconstruction
Same as minimizing cross entropy

want $p(z)$ to be $\mathcal{N}(\mu = 0, \Sigma = I)$
because it makes nice latent space

$q(z\,|\,x^{(i)}) \rightarrow (\mu_{z|x}, \Sigma_{z|x}) \quad p(z) \rightarrow \mathcal{N}(0,1)$

$$= \frac{1}{2}\sum_k \left(\Sigma_{k,k} + \mu_k^2 - 1 - \log \Sigma_{k,k}\right)$$

raw covariance is not numerically stable because of underflow

$$\log \Sigma_{k,k} = \widehat{\Sigma_{k,k}}$$

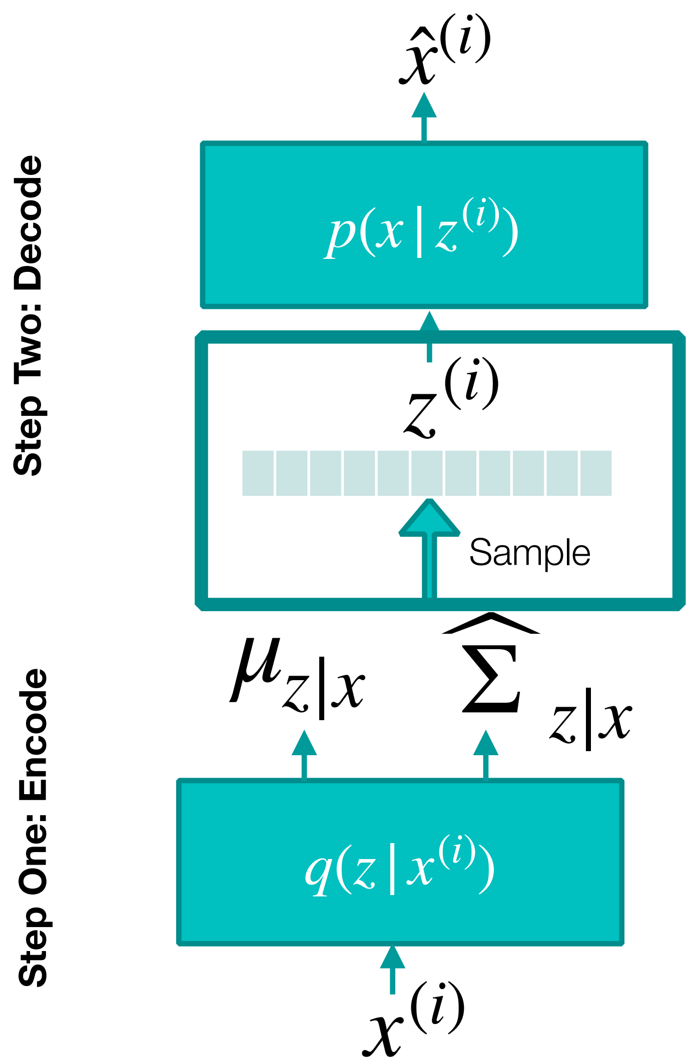
predicted by
$q(z\,|\,x^{(i)})$

$$= \frac{1}{2}\sum_k \left(\exp\left(\widehat{\Sigma_{k,k}}\right) + \mu_k^2 - 1 - \widehat{\Sigma_{k,k}}\right)$$

so we will have the neural network output log variance

Also, remember we assume **diagonal covariance**, so z's are not correlated
This means covariance is only a vector of variances (the diagonal of Σ)

Back Propagating



$$\geq \mathbf{E}_{q(z|x^{(i)})}\left[\log p(x^{(i)} \mid z)\right] - D_{KL}\left[q(z \mid x^{(i)}) \| p(z)\right]$$

This is partially differentiable by chain rule…

$$\mathcal{N}(\mu_{z|x}, \exp(\widehat{\Sigma_{z|x}})) = z$$

$$= \mu(x^{(i)}) + \exp(\widehat{\Sigma(x^{(i)})}) \cdot \mathcal{N}(0,1)$$

To update q,
we need to back propagate
through sampling layer. How?

The Loss Function Implementation

```
# Encode the input into a mean and variance parameter
z_mean, z_log_variance = encoder(input_img)
```

$\mu(x^{(i)})$ $\widehat{\Sigma(x^{(i)})}$ $q(z \mid x^{(i)})$

```
# Draw a latent point using a small random epsilon
z = z_mean + exp(z_log_variance) * epsilon

# Then decode z back to an image
reconstructed_img = decoder(z)
```

$\hat{x}^{(i)} = p(x^{(i)} \mid z)$

```
# Instantiate a model
model = Model(input_img, reconstructed_img)
```

$$z = \mu(x^{(i)}) + \exp(\widehat{\Sigma(x^{(i)})}) \cdot \mathcal{N}(0,1)$$

```
def vae_loss(self, x, z_decoded):
    x = K.flatten(x)
    z_decoded = K.flatten(z_decoded)
    xent_loss = keras.metrics.binary_crossentropy(x, z_decoded)
    kl_loss = -5e-4 * K.mean(
        1 + z_log_var - K.square(z_mean) - K.exp(z_log_var), axis=-1)
    return K.mean(xent_loss + kl_loss)
```

$$-\mathbf{E}_{q(z|x^{(i)})}\left[\log p(x^{(i)} \mid z)\right]$$

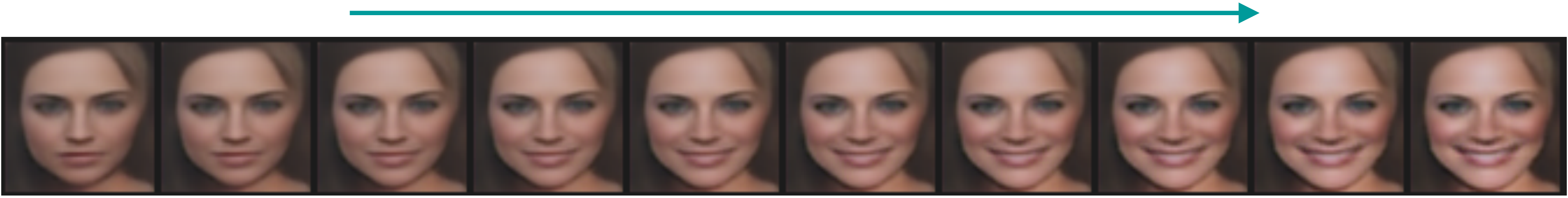
$$-\lambda \sum_k 1 + \widehat{\Sigma(x^{(i)})} - \mu(x^{(i)})^2 - \exp(\widehat{\Sigma(x^{(i)})})$$

Note:
Flipped from maximization to minimization
and added lambda for tradeoff in reconstruction, normal latent space

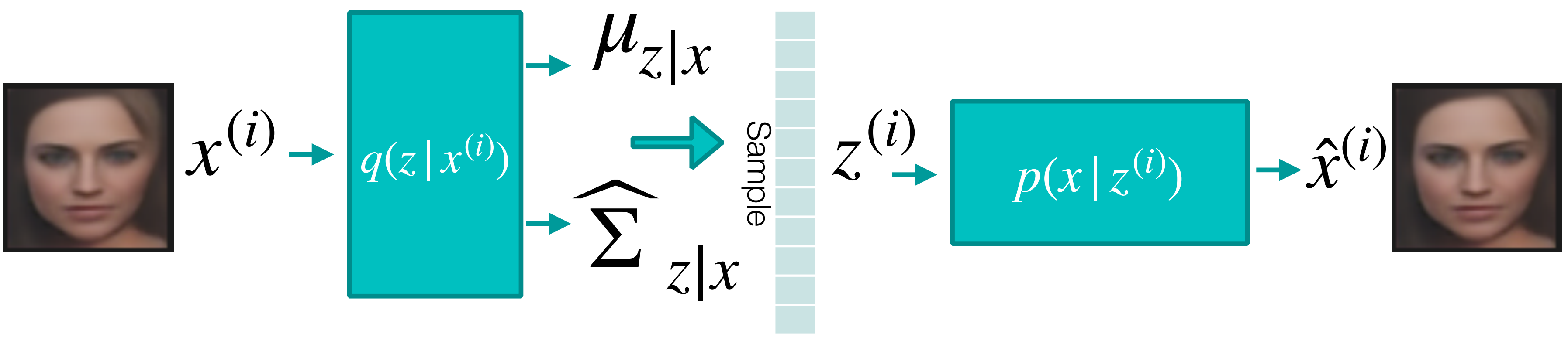
$$= -\mathbf{E}_{q(z|x^{(i)})}\left[\log p(x^{(i)} \mid z)\right] - \lambda \sum_k 1 + \widehat{\Sigma(x^{(i)})} - \mu(x^{(i)})^2 - \exp(\widehat{\Sigma(x^{(i)})})$$

Now that its trained, so what?

Encoding faces, then adjust the “z” that relates to smiling.

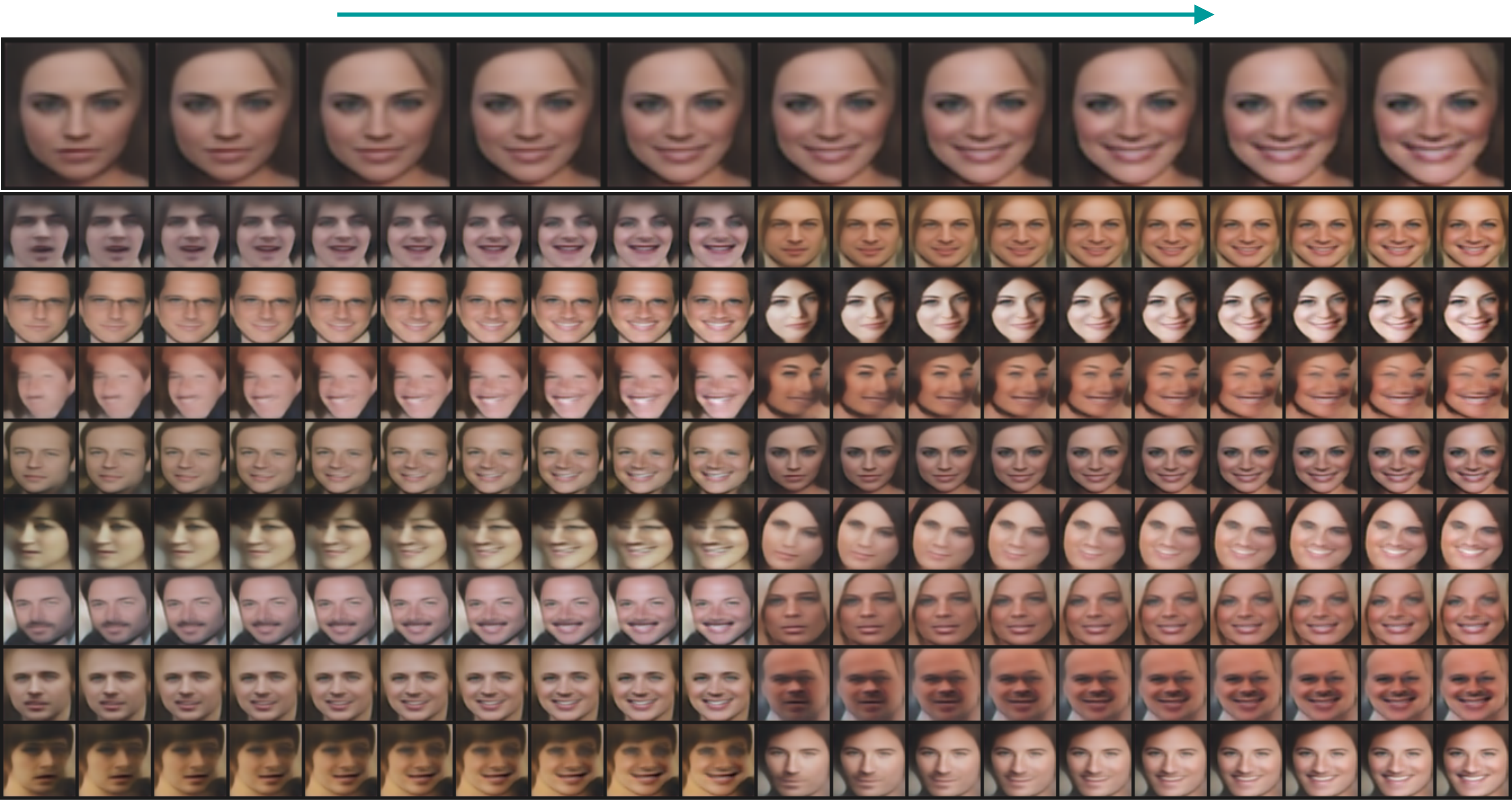


Investigate what happens by moving around each z_i



VAE Examples

Encoding faces, then adjust the “z” that relates to smiling.



VAE Examples

Different, automatically found z, latent variables

VAEs in Keras

Sampling from variational auto encoder using MNIST

Demo by Francois Chollet

In Master Repo: 07a VAEs in Keras.ipynb

Follow Along: https://github.com/fchollet/deep-learning-with-python-notebooks/blob/master/8.4-generating-images-with-vaes.ipynb

Adversarial Auto Encoding

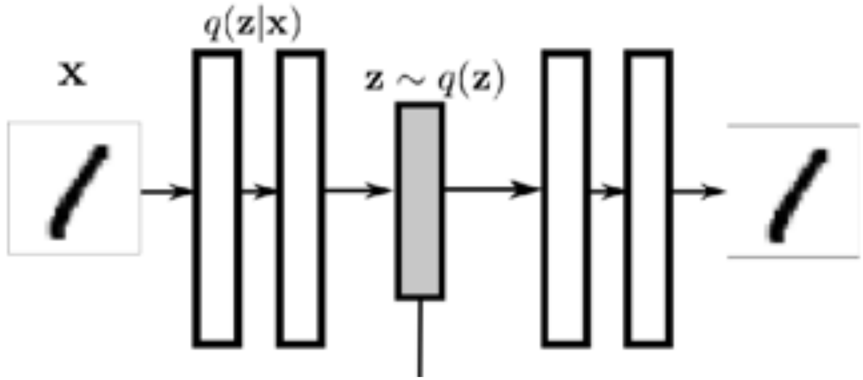
Do we need something more than VAE?

- Arguments for Yes:
  - ELBO is not global optimum! But… provides theory
  - Assumption of Normal distributions to q(z) is limiting
  - Training tends to be slower (…*so do GANs…*)
  - Manifold of distributions do not cover the latent space completely (not guaranteed)
  - We can't incorporate distributions separately for different classes without reformulating loss function
- Arguments for No:
  - It seems hard, how can we research methods that aren't low hanging fruit? Plus the VAE math was like really hard for me to understand so this is not going to be very fun, guaranteed. Ah, fine lets look at it.

The Main Idea

- How can we enforce constraints on the latent space with a pair of networks?

Encoder-Decoder
Network



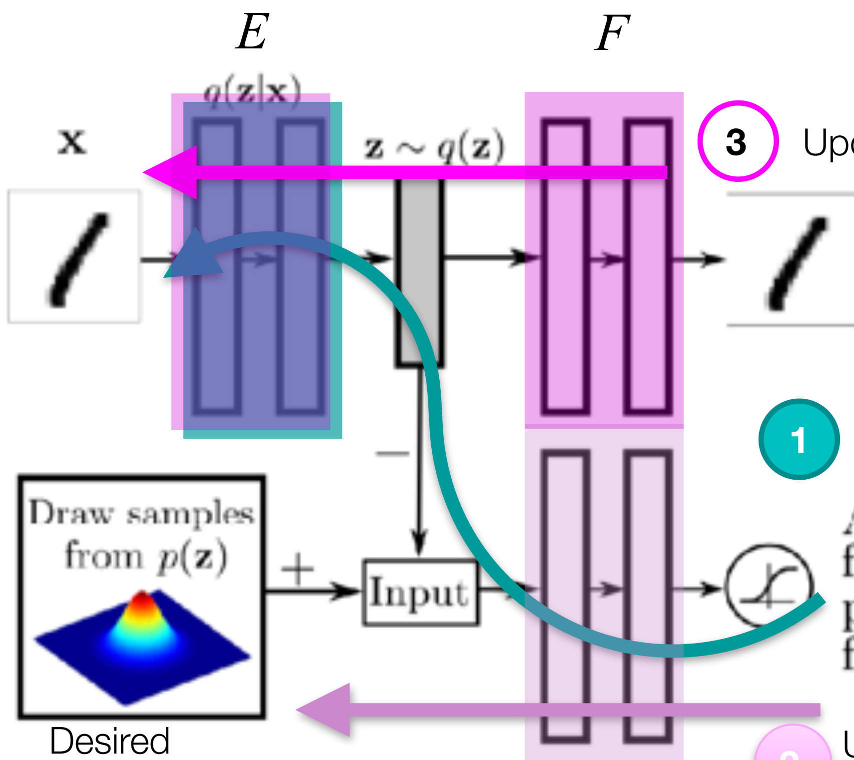
Makhzani, Alireza, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. "Adversarial autoencoders." arXiv preprint arXiv:1511.05644 (2015).

The Main Idea

- How can we enforce constraints on the latent space with a pair of networks?

Encoder-Decoder Network

Adversarial Network



Desired Latent Space

Replace VAE KL loss with a discriminator

D

$$\mathcal{L}^{E,F}_{error} = BCE(x, F \circ E(x))$$

3 Update Encoder and Decoder Pair

Impose Reconstruction Loss

$$\mathcal{L}^{E}_{error} = BCE(1,D \circ E(x))$$

1 Update Encoder to fool Discriminator
Do **not** update Discriminator

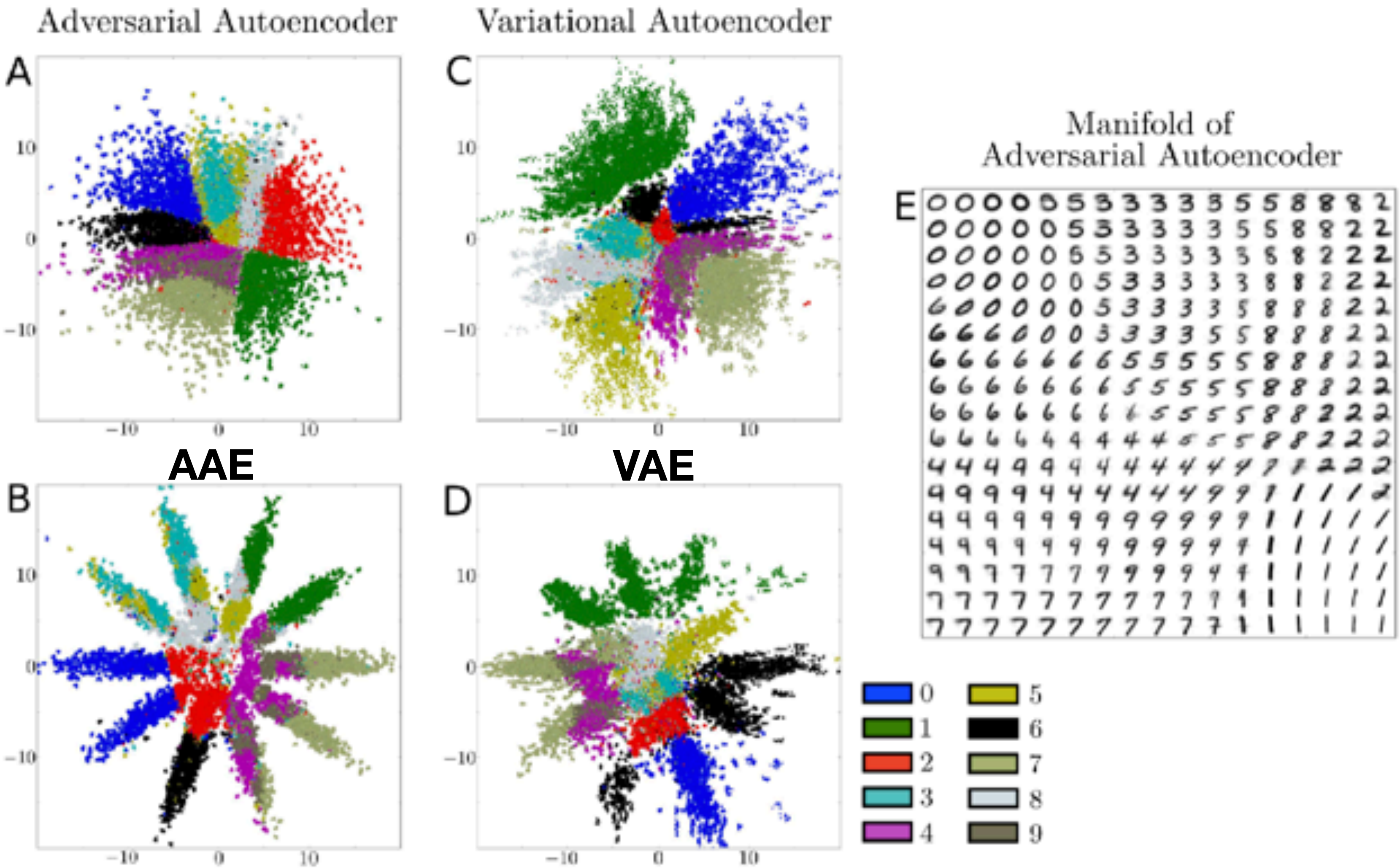
Adversarial cost for distinguishing positive samples $p(\mathbf{z})$ from negative samples $q(\mathbf{z})$

2 Update Discriminator to Detect Fake z
Do **not** update Encoder

$$\mathcal{L}^{D}_{error} = BCE(0,D \circ E(x)) + BCE(1,D(z))$$

Makhzani, Alireza, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. "Adversarial autoencoders." arXiv preprint arXiv:1511.05644 (2015).

Arbitrary Prior Distributions



Makhzani, Alireza, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. "Adversarial autoencoders." arXiv preprint arXiv:1511.05644 (2015).