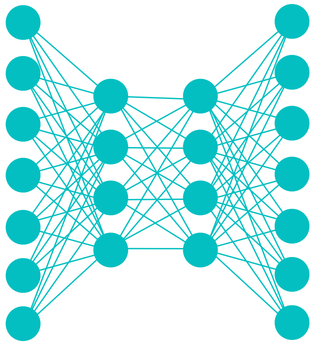
Lecture Notes for

Neural Networks and Machine Learning

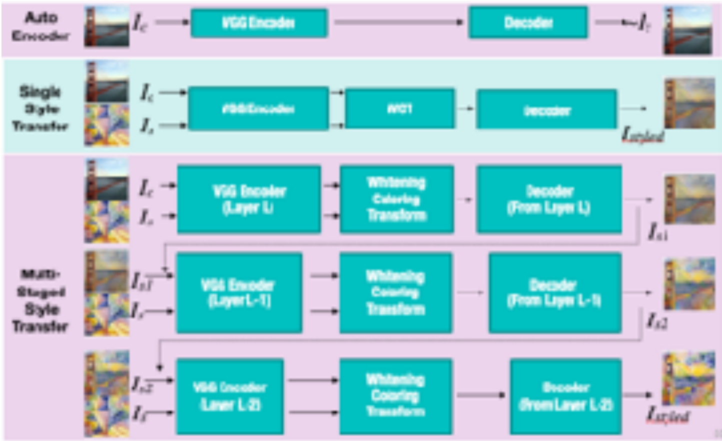
Generative Networks and Auto-Encoding Generators



Logistics and Agenda

- Logistics
  - Lab dates pushed back (see schedule)
  - Next Week: Student paper presentation
- Agenda
  - A historical perspective of generative Neural Networks
  - Variational Auto-Encoding
  - VAE in Keras Demo (if time)
  - Adversarial Auto-Encoders (if time)

Last Time



$$\hat{\mathbf{R}} = (1-\alpha)\boxed{(\mathbf{I} - \alpha\mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}})}^{-1}\mathbf{Y}$$

NPxNP NPxNP identity NPxNP NPxNP NPxNP NPxNP

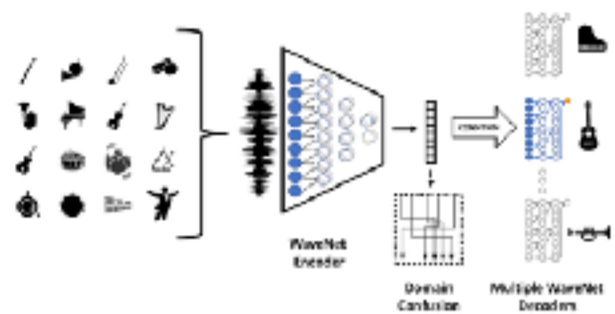
Laplacian of graph

- **D** is diagonal and easily invertible
- **W** is sparse and efficiently inverted after multiplications
- **Y** is the stylized image pixels on a diagonal matrix
- **R** can be converted to an image by returning the diagonal

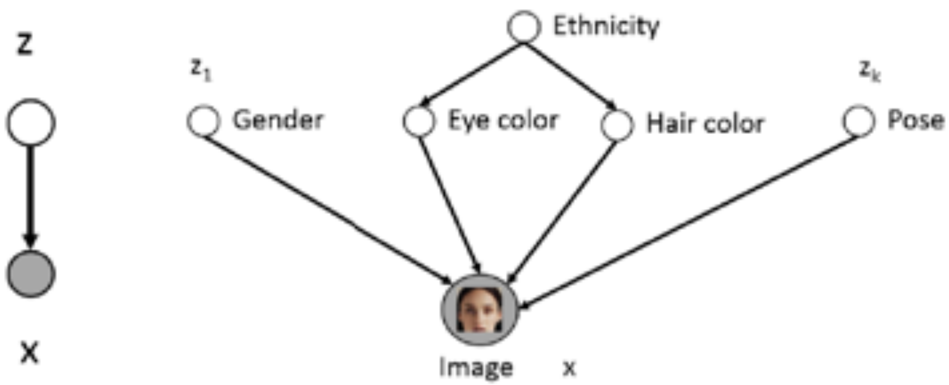
State of the Art in Audio Transfer

- FAIR results are compelling…

- WaveNet is an autoencoder for speech and music, capable of capturing many aspects of music from time domain samples
- FAIR Paper: Train single encoder, multiple decoders

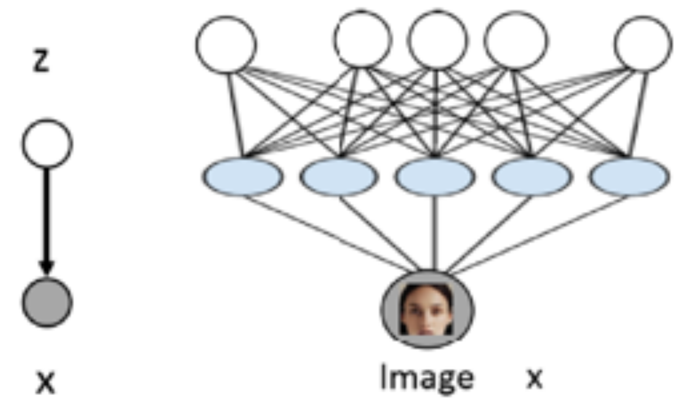


Motivations: Generative Latent Variables



$$p(\mathbf{x} \mid \mathbf{z})$$

Hard: **z** is expertly chosen



$$p(\mathbf{x} \mid \mathbf{z})$$

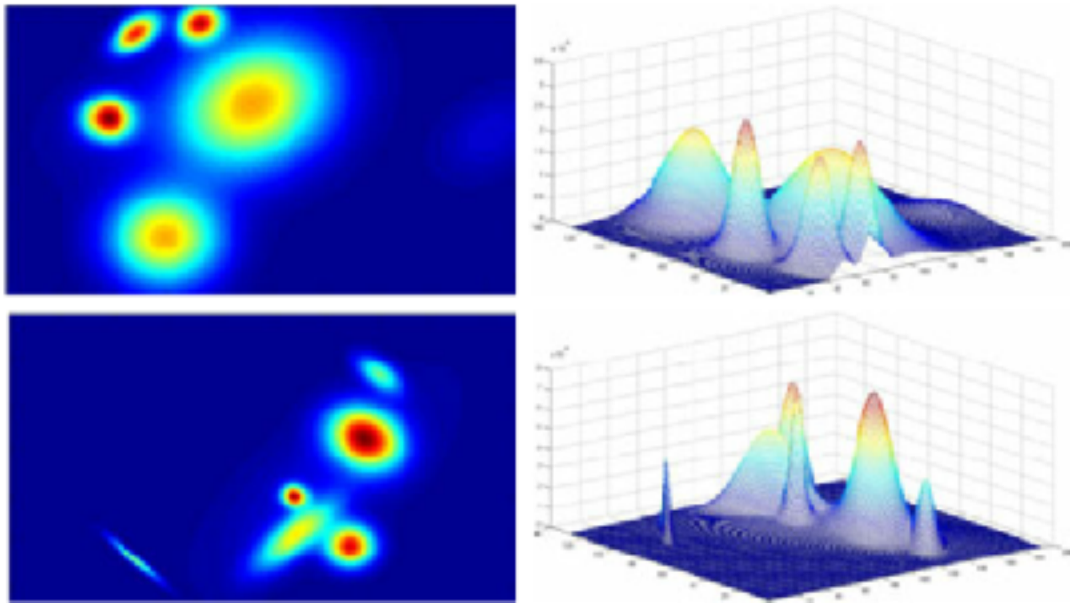
Not as Hard: **z** is trained,
latent variables are uncontrolled

Want: $$p(\mathbf{x}) \approx \sum_{\mathbf{z}} p(\mathbf{z}) p_{\theta}(\mathbf{x} \mid \mathbf{z})$$

http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture13.pdf

Motivation: Mixtures for Simplicity

Want: $$p(\mathbf{x}) \approx \sum_{\mathbf{z}} p(\mathbf{z}) p_\theta(\mathbf{x} \mid \mathbf{z})$$



- Each latent variable is mostly independent of other latent variables
- The sum of various mixtures can approximate most any distribution
- Good choice for conditional is Normal Distribution
- Can parameterize $p(x|z)$ to be a Neural Network

$$p_\theta(\mathbf{x} \mid \mathbf{z} = k) = \mathcal{N}(\mathbf{x}, \mu_k, \Sigma_k)$$

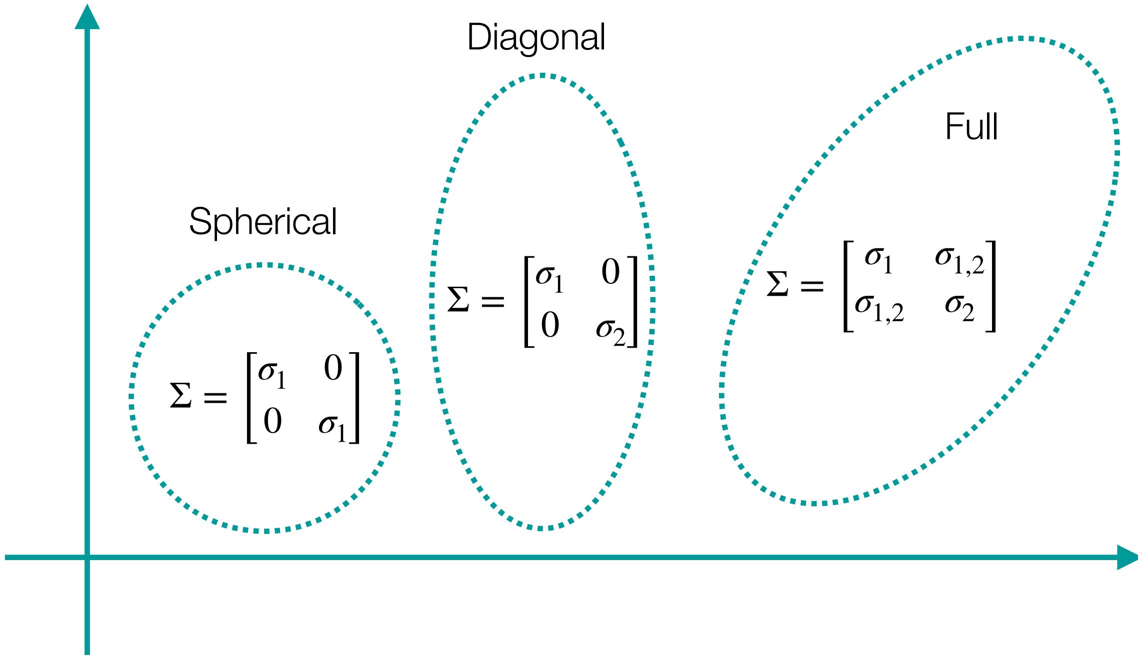
mean and covariance learned

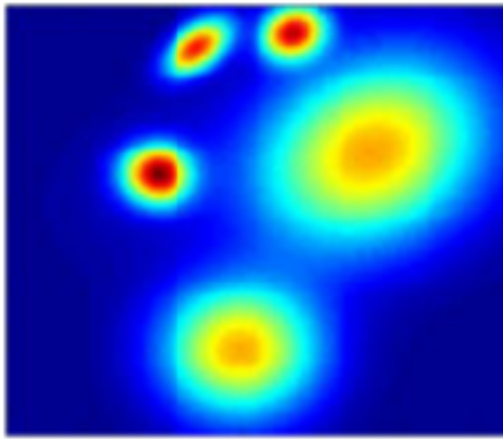
http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture13.pdf

Motivation: Mixtures for Simplicity

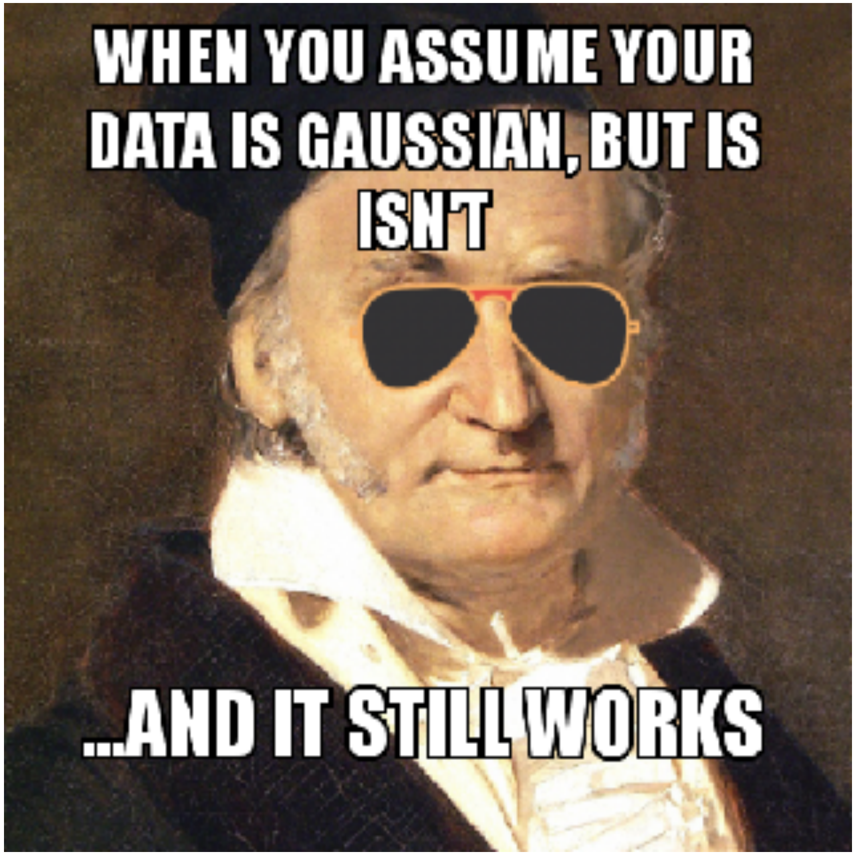
$$= \mathcal{N}(\mathbf{x}, \mu_k, \Sigma_k)$$

mean and covariance learned





A History of Generative Networks



Aside: Notation

some function

could be neural networks

$$\mathbf{E}_{s \leftarrow q(s|x)}[f(\,\cdot\,)] = \int q(s\,|\,x) \cdot f(x)\; dx \approx \sum_{\forall i} q(s\,|\,x^{(i)}) \cdot f(x^{(i)})$$

Expected value of f under conditional distribution, q

s is latent variable, $x^{(i)}$ is an observation

$$\mathbf{E}_{s \leftarrow q(s|x)}[\log f(\,\cdot\,)] = \sum_{\forall i} q(s\,|\,x^{(i)}) \cdot \log\left(f(x^{(i)})\right)$$

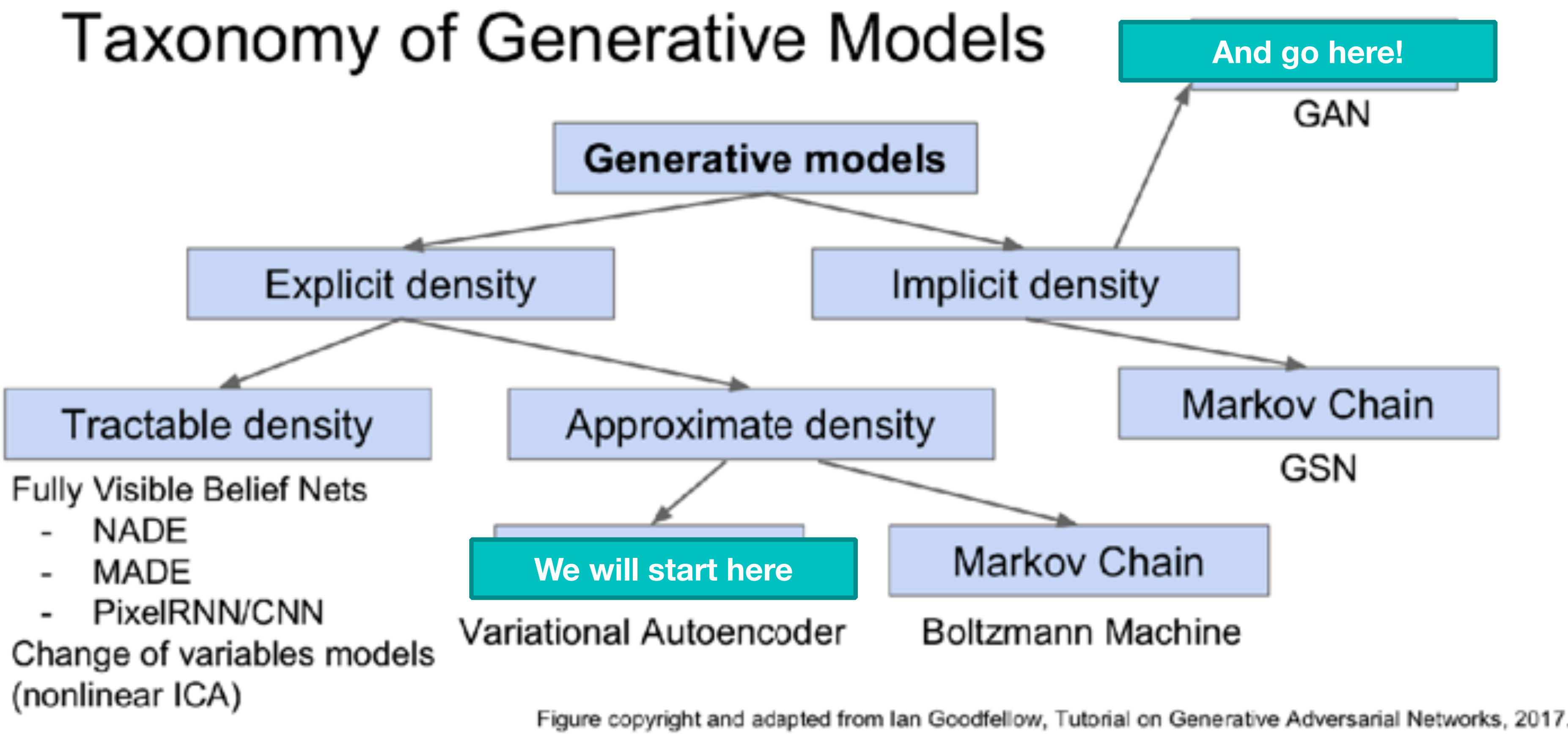
If function is a probability, this is just the negative of cross entropy of distributions:

$$H(q,p) = -\sum_{x} q(x) \cdot \log(p(x))$$

Recall that KL divergence is a measure of difference in two distribution, and is just:

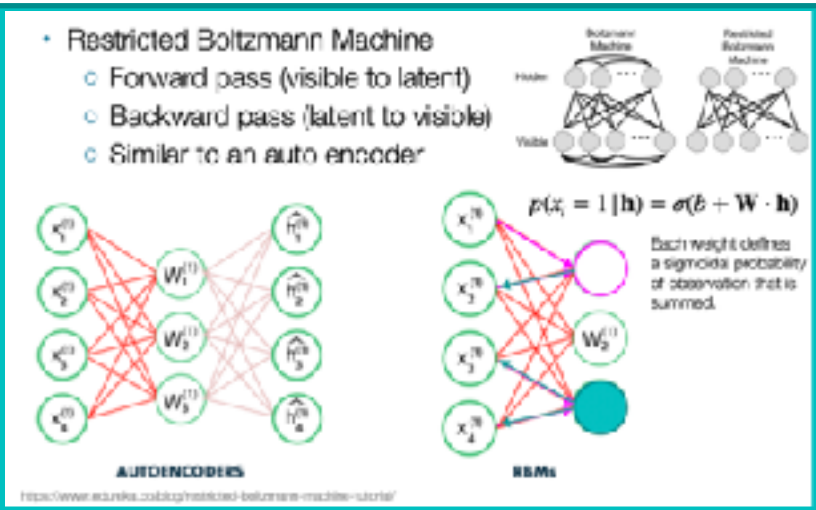
$$D(p||q) = \sum_{x} p(x) \cdot \log\left(\frac{p(x)}{q(x)}\right) = \mathbf{E}_p\left[\log\left(\frac{p(x)}{q(x)}\right)\right]$$

Taxonomy of Generative Models

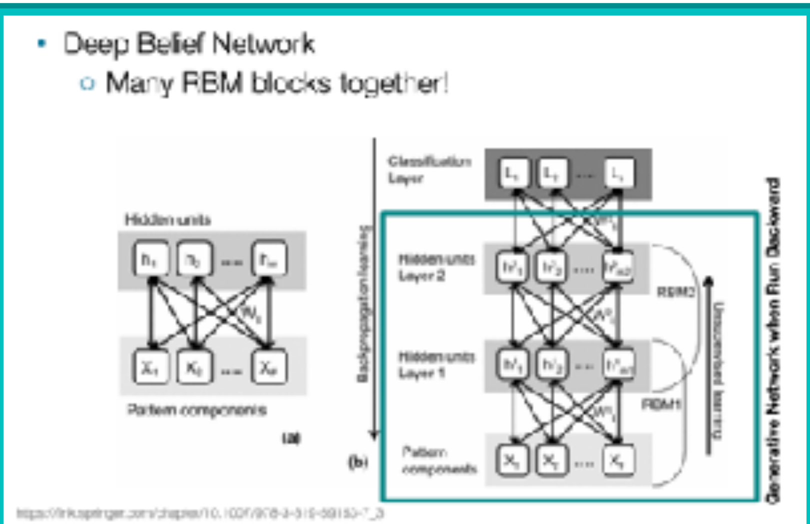


http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture13.pdf

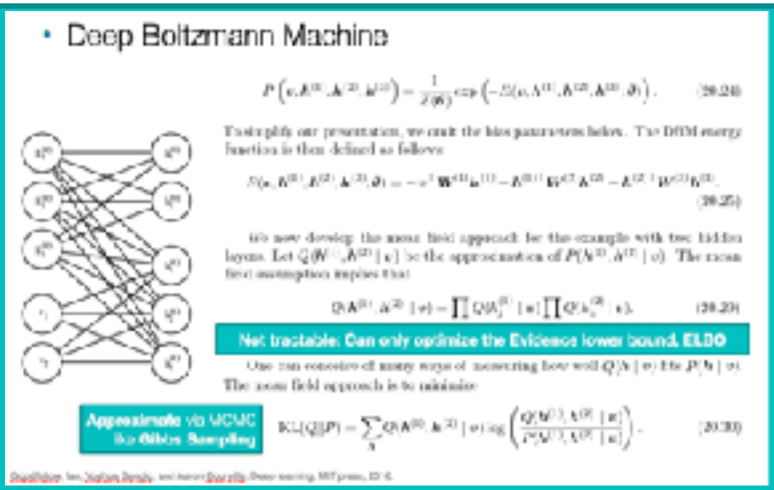
Abridged History of Generative Networks



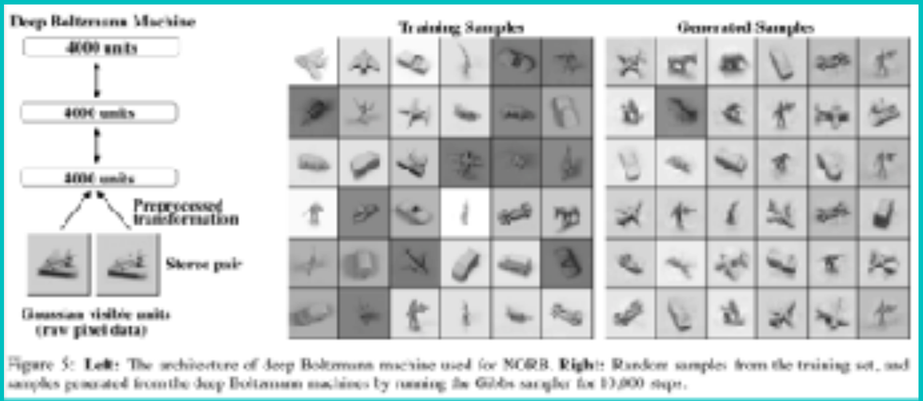
2006 Restricted Boltzmann Machine



2007, Deep Belief Networks
RBMs with many layers



2009 Deep Boltzmann Machine
Goodfellow, Bengio, Courville

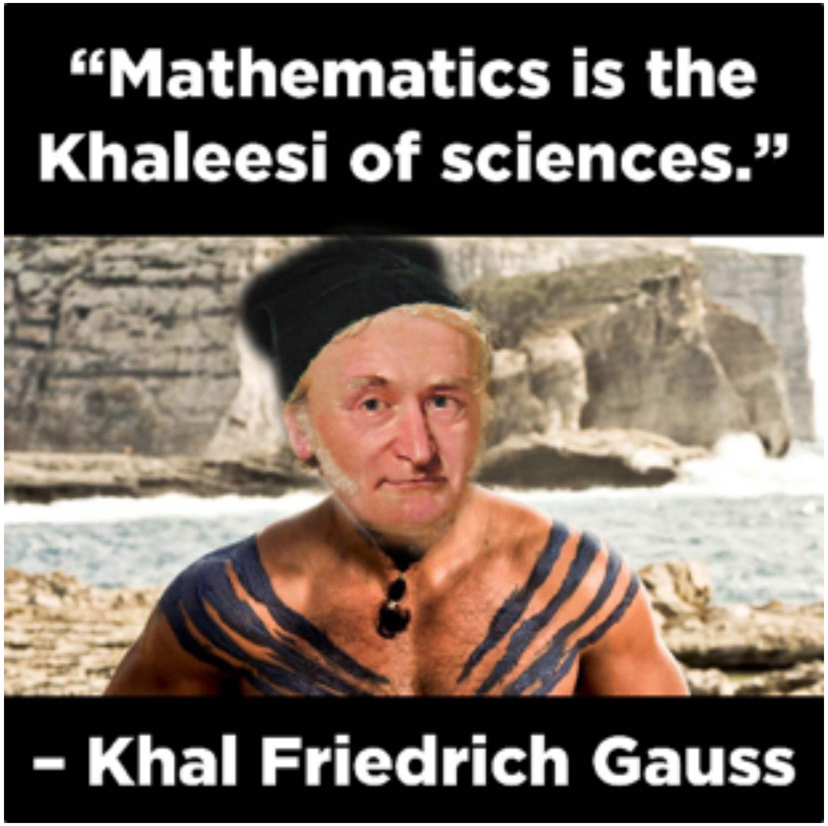


2009, Practical Examples
Salakhutdinov and Hinton

Contemporary Modeling

- DBNs and DBMs did not become very popular
  - Mathematics detracts from popular understanding
  - Often methods using sampling are not scalable
  - Cannot directly use Gradients (no Back Prop) 😢
- Popular method for calculating generative networks with Evidence Lower Bound (ELBO) approximation:
  - Variational Auto Encoding
    - No guarantees about global minimum
    - But scalable and will converge in finite time

Variational Auto Encoding



Can Auto Encoding Generate Samples?

$\hat{x}^{(i)}$

Decoder

$z^{(i)}$

Encoder

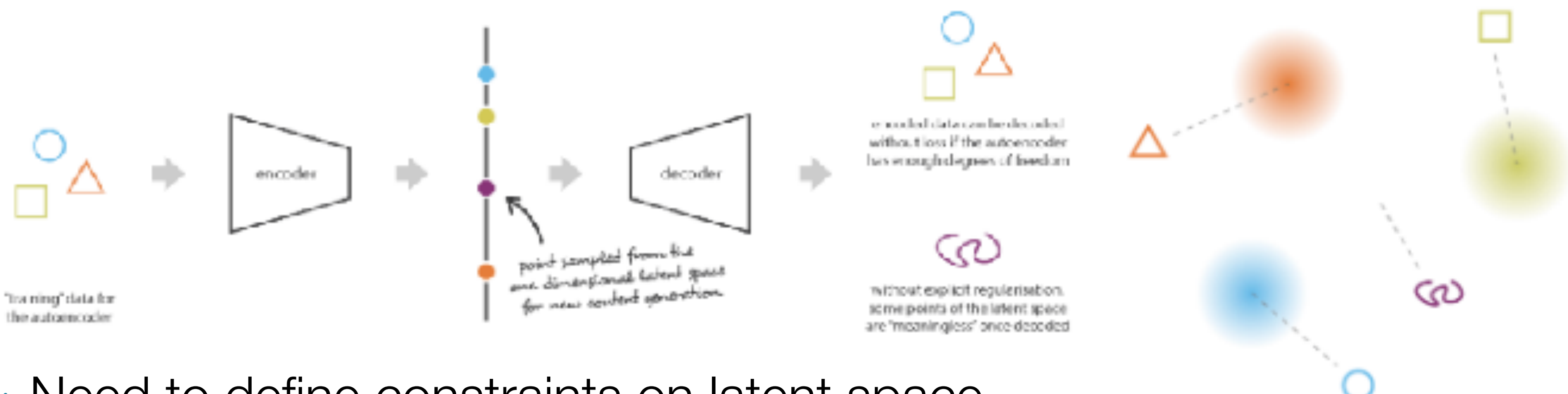
$x^{(i)}$

Once trained, is it possible to generate data?

$\hat{x}^{(i)}$

Decoder

z Sampled in Latent Space



- Need to define constraints on latent space…

http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture13.pdf

https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

Reasonable constraints for p(z)?

$$\hat{x}^{(i)}$$

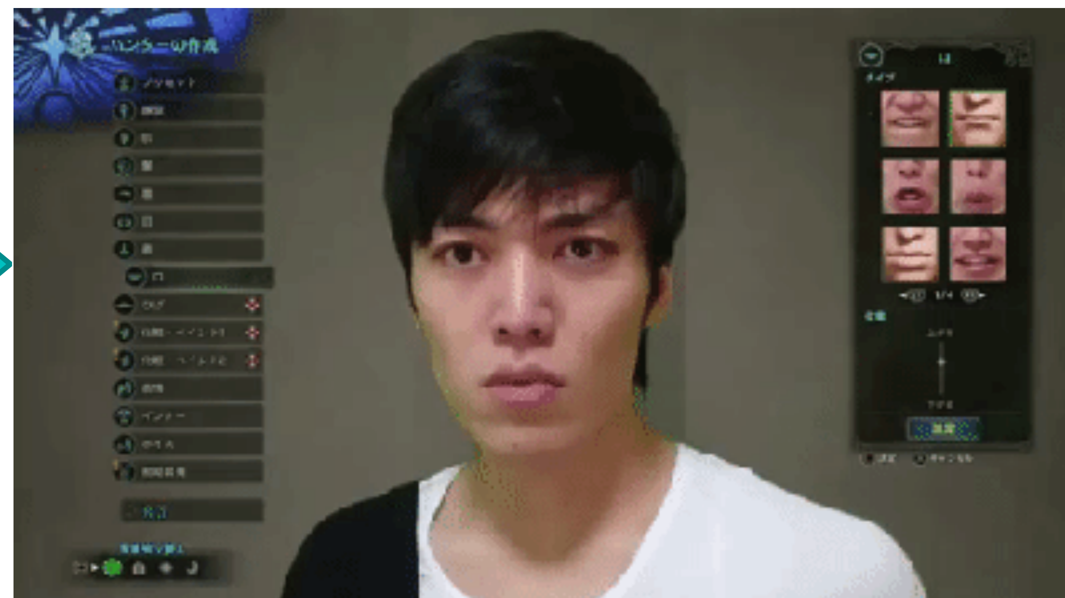
Decoder

z Sampled in Latent Space

- Should be simple, easy to sample from: **Normal**
- Each component should be i.i.d.: **Diag. Covariance**
    - Encourages features that may be semantic, like expert might select

Ideal:
Smiling
Hair Color
Skin Color
Eye Color
...
Nose Position

z → Transform with Decoder → $\hat{x}^{(i)}$



Optimizing

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)}$$

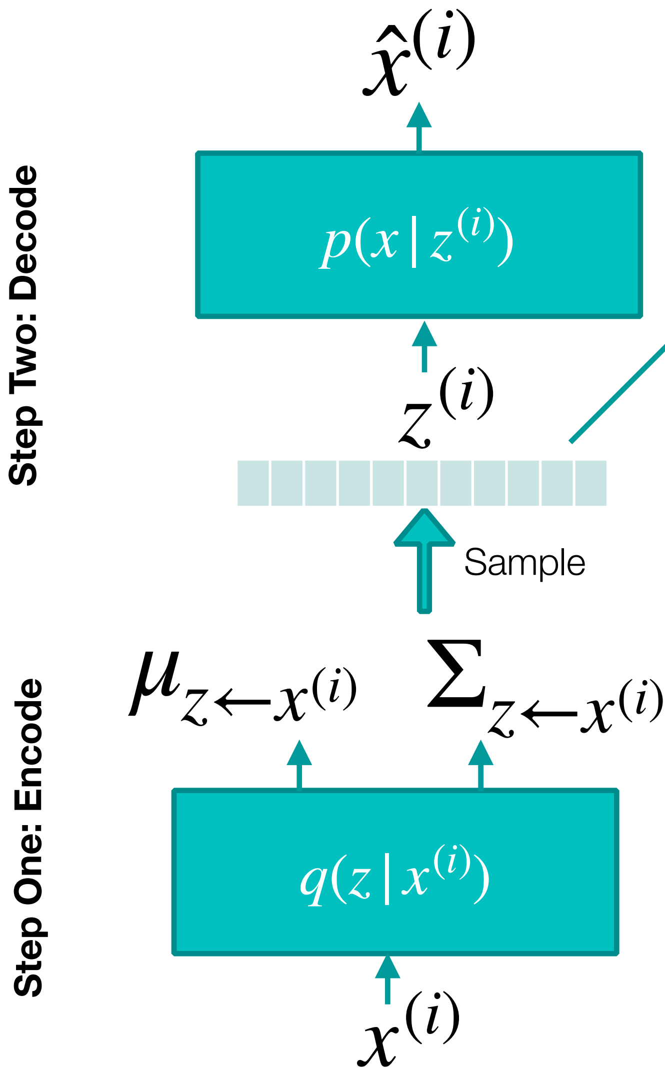
We need this inference in order to compute latent variable

$$p(x) = \int p(x|z)p(z)dz$$

Denominator is of this form

- We can't compute! **Intractable computation** for all "z"
- So let's define this with **variational inference**:
  - AKA: Find the best approximation of desired distribution using a parametrized set of distributions (usually normal distributions)
  - Only needs to work for z **with observed** $x^{(i)}$
  - 1. **Encode** observed $x^{(i)}$ as Gaussian distribution via network $q(z \mid x^{(i)})$,
  - 2. Use $q(z \mid x^{(i)})$ to sample z appropriately, then **decode** with another neural network, $p(x^{(i)} \mid z^{(i)})$
  - 3. Make $q(z \mid x^{(i)})$ largest probability possible via Gaussian Distributions

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 $x^{(i)}$:

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

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

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

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

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

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

Can get this by manipulating the KL for normal distribution

Determinant of diagonal matrix is simple. Motivates diagonal covariance…

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

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

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

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

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)