Variational Autoencoders

Learning generative models with latent representations

Claas Völcker

May 19, 2020

Deep Generative Models - SoSe 2020

Table of contents

Inference through optimization

What if we replaced an inference question with optimization?

- The target:

$$P(X)$$

What if we replaced an inference question with optimization?

- The target:

$$P(X)$$

- The hope: there is a nice

$$z, \text{ so that } P(x) = \int P(z)P(x|z)dz$$

which governs x (latent variable)

What if we replaced an inference question with optimization?

- The target:

$$P(X)$$

- The hope: there is a nice

$$z, \text{ so that } P(x) = \int P(z)P(x|z)dz$$

  which governs x (latent variable)
- I.e. all dogs look similar, if I know something is a dog, certain attributes (tails, legs, snout) are likely

What if we replaced an inference question with optimization?

- The target:

$$P(X)$$

- The hope: there is a nice

$$z, \text{ so that } P(x) = \int P(z)P(x|z)dz$$

  which governs x (latent variable)
- I.e. all dogs look similar, if I know something is a dog, certain attributes (tails, legs, snout) are likely
- Idea: rephrase inference as optimization

Working through the math - 1

$$\text{Maximize: } P(x) \sim \int p_\phi(x|z)p_\phi(z)dz$$

Working through the math - 1

$$\text{Maximize: } P(x) \sim \int p_\phi(x|z)p_\phi(z)dz$$

- A latent variable model

Working through the math - 1

$$\text{Maximize: } P(x) \sim \int p_\phi(x|z)p_\phi(z)dz$$

- A latent variable model
- Assumption: there is some (hopefully small) z which governs data

Working through the math - 1

$$\text{Maximize: } P(x) \sim \int p_\phi(x|z)p_\phi(z)dz$$

- A latent variable model
- Assumption: there is some (hopefully small) z which governs data
- Define $P(z)$ and $P(x|z)$

Working through the math - 1

$$\text{Maximize: } P(x) \sim \int p_\phi(x|z)p_\phi(z)dz$$

- A latent variable model
- Assumption: there is some (hopefully small) z which governs data
- Define $P(z)$ and $P(x|z)$
- Main problem: We don't have that information for our dog

Working through the math - 1

$$\text{Maximize: } P(x) \sim \int p_\phi(x|z)p_\phi(z)dz$$

- A latent variable model
- Assumption: there is some (hopefully small) z which governs data
- Define $P(z)$ and $P(x|z)$
- Main problem: We don't have that information for our dog
- Can we just ignore that problem? :D

Working through the math - 1

$$\text{Maximize: } P(x) \sim \int p_\phi(x|z)p_\phi(z)dz$$

- A latent variable model
- Assumption: there is some (hopefully small) z which governs data
- Define $P(z)$ and $P(x|z)$
- Main problem: We don't have that information for our dog
- Can we just ignore that problem? :D

$$P(x|z) = \mathcal{N}(x|f(z;\phi), \sigma^2 \cdot I)$$

Working through the math - 1

$$\text{Maximize: } P(x) \sim \int p_\phi(x|z)p_\phi(z)dz$$

- A latent variable model
- Assumption: there is some (hopefully small) z which governs data
- Define $P(z)$ and $P(x|z)$
- Main problem: We don't have that information for our dog
- Can we just ignore that problem? :D

$$P(x|z) = \mathcal{N}(x|f(z;\phi), \sigma^2 \cdot I)$$

- f is deterministic, $\mathcal{N}$ enables optimization

Working through the math - 2

- We could maximize by sampling? (akin to MCMC)

Working through the math - 2

- We could maximize by sampling? (akin to MCMC)
- No, too many samples would be needed

Working through the math - 2

- We could maximize by sampling? (akin to MCMC)
- No, too many samples would be needed
- Don't know if a sampled z is actually "meant to" produce a given x

Working through the math - 2

- We could maximize by sampling? (akin to MCMC)
- No, too many samples would be needed
- Don't know if a sampled z is actually "meant to" produce a given x
- It would be great to have $P(z|x)$ for that

Working through the math - 2

- We could maximize by sampling? (akin to MCMC)
- No, too many samples would be needed
- Don't know if a sampled z is actually "meant to" produce a given x
- It would be great to have $P(z|x)$ for that
- That's just our problem in reverse?!

Working through the math - 2

- We could maximize by sampling? (akin to MCMC)
- No, too many samples would be needed
- Don't know if a sampled z is actually "meant to" produce a given x
- It would be great to have $P(z|x)$ for that
- That's just our problem in reverse?!

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

Intermission: Kullback Leibler Divergence

- Characterizes the ”distance” between distributions

[1] In a mathematical, strict sense, for practical purposes

$$KL(P||Q) = 0 \text{ , iff } P = Q$$

Intermission: Kullback Leibler Divergence

- Characterizes the ”distance” between distributions
- Positive, 0 only if two distributions are equal (almost everywhere)[1]

[1] In a mathematical, strict sense, for practical purposes

$$KL(P||Q) = 0 \text{ , iff } P = Q$$

Intermission: Kullback Leibler Divergence

- Characterizes the ”distance” between distributions
- Positive, 0 only if two distributions are equal (almost everywhere)[1]
- Not a metric, since it is asymetric, but still useful

[1] In a mathematical, strict sense, for practical purposes

$$KL(P||Q) = 0 \text{ , iff } P = Q$$

Intermission: Kullback Leibler Divergence

- Characterizes the ”distance” between distributions
- Positive, 0 only if two distributions are equal (almost everywhere)[1]
- Not a metric, since it is asymetric, but still useful

$$\mathcal{KL}(P||Q) = \int p(x) \log \frac{p(x)}{q(x)} dx$$

[1] In a mathematical, strict sense, for practical purposes

$$KL(P||Q) = 0 \text{ , iff } P = Q$$

Intermission: Kullback Leibler Divergence

- Characterizes the ”distance” between distributions
- Positive, 0 only if two distributions are equal (almost everywhere)[1]
- Not a metric, since it is asymetric, but still useful

$$\mathcal{KL}(P||Q) = \int p(x) \log \frac{p(x)}{q(x)} dx$$
$$= \mathbb{E}_{x\sim p}\left[\log \frac{p(x)}{q(x)}\right] = \mathbb{E}_{x\sim p}[\log(p(x)) - \log(q(x))]$$

[1] In a mathematical, strict sense, for practical purposes

$$KL(P||Q) = 0 \text{ , iff } P = Q$$

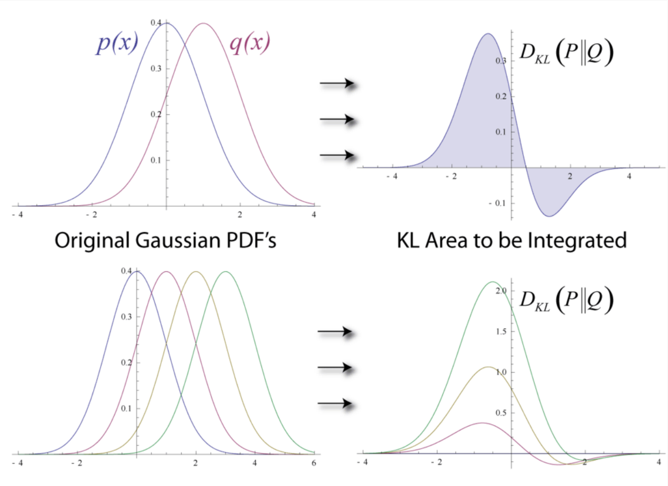


Figure 1: Taken from "Kullback–Leibler divergence", Wikipedia, CC BY-SA 3.0

Working through the math - 3

- Ignoring problems seemed like a good way to go :D

Working through the math - 3

- Ignoring problems seemed like a good way to go :D
- Let's continue with that idea, parametrize a function Q and optimize it

Working through the math - 3

- Ignoring problems seemed like a good way to go :D
- Let's continue with that idea, parametrize a function Q and optimize it

$$\mathcal{KL}[Q(z)||P(z|x)] = \int Q(z) \log\left(\frac{Q(z)}{P(z|x)}\right)$$

Working through the math - 3

- Ignoring problems seemed like a good way to go :D
- Let's continue with that idea, parametrize a function Q and optimize it

$$\mathcal{KL}[Q(z)||P(z|x)] = \int Q(z) \log \left(\frac{Q(z)}{P(z|x)} \right)$$

$$\mathbb{E}_{z \sim Q}[\log Q(z) - \log P(z|x)]$$

Working through the math - 3

- Ignoring problems seemed like a good way to go :D
- Let's continue with that idea, parametrize a function Q and optimize it

$$\mathcal{KL}[Q(z)||P(z|x)] = \int Q(z) \log \left(\frac{Q(z)}{P(z|x)} \right)$$

$$\mathbb{E}_{z \sim Q}[\log Q(z) - \log P(z|x)]$$

- Using Bayes rule:

Working through the math - 3

- Ignoring problems seemed like a good way to go :D
- Let's continue with that idea, parametrize a function Q and optimize it

$$\mathcal{KL}[Q(z)||P(z|x)] = \int Q(z) \log \left(\frac{Q(z)}{P(z|x)} \right)$$

$$\mathbb{E}_{z \sim Q}[\log Q(z) - \log P(z|x)]$$

- Using Bayes rule:

$$\mathcal{KL}[Q(z)||P(z|x)] = \mathbb{E}_{z \sim Q}[\log Q(z) - \log \frac{P(x|z)P(z)}{P(x)}]$$

Working through the math - 3

- Ignoring problems seemed like a good way to go :D
- Let's continue with that idea, parametrize a function Q and optimize it

$$\mathcal{KL}[Q(z)||P(z|x)] = \int Q(z) \log \left(\frac{Q(z)}{P(z|x)} \right)$$

$$\mathbb{E}_{z \sim Q}[\log Q(z) - \log P(z|x)]$$

- Using Bayes rule:

$$\mathcal{KL}[Q(z)||P(z|x)] = \mathbb{E}_{z \sim Q}[\log Q(z) - \log \frac{P(x|z)P(z)}{P(x)}]$$

$$= \mathbb{E}_{z \sim Q}[\log Q(z) - (\log P(x|z) + \log P(z) - \log P(x))]$$

Working through the math - 3

- Ignoring problems seemed like a good way to go :D
- Let's continue with that idea, parametrize a function Q and optimize it

$$\mathcal{KL}[Q(z)||P(z|x)] = \int Q(z) \log \left(\frac{Q(z)}{P(z|x)} \right)$$

$$\mathbb{E}_{z \sim Q}[\log Q(z) - \log P(z|x)]$$

- Using Bayes rule:

$$\mathcal{KL}[Q(z)||P(z|x)] = \mathbb{E}_{z \sim Q}[\log Q(z) - \log \frac{P(x|z)P(z)}{P(x)}]$$

$$= \mathbb{E}_{z \sim Q}[\log Q(z) - (\log P(x|z) + \log P(z) - \log P(x))]$$

$$= \mathbb{E}_{z \sim Q}[\log Q(z) - \log P(x|z) - \log P(z)] + \log P(x)$$

Working through the math - 4

$$\mathcal{KL}[Q(z)||P(z|x)] = \mathbb{E}_{z\sim Q}[\log Q(z) - \log P(x|z) - \log P(z)] + \log P(x)$$

Working through the math - 4

$$\mathcal{KL}[Q(z)||P(z|x)] = \mathbb{E}_{z\sim Q}[\log Q(z) - \log P(x|z) - \log P(z)] + \log P(x)$$
$$\log P(x) - \mathcal{KL}[Q(z)||P(z|x)] = \mathbb{E}_{z\sim Q}[\log P(x|z)] - \mathcal{KL}[Q(z)||P(z)]$$

Working through the math - 4

$$\mathcal{KL}[Q(z)||P(z|x)] = \mathbb{E}_{z\sim Q}[\log Q(z) - \log P(x|z) - \log P(z)] + \log P(x)$$
$$\log P(x) - \mathcal{KL}[Q(z)||P(z|x)] = \mathbb{E}_{z\sim Q}[\log P(x|z)] - \mathcal{KL}[Q(z)||P(z)]$$

What we have now:

Working through the math - 4

$$\mathcal{KL}[Q(z)||P(z|x)] = \mathbb{E}_{z\sim Q}[\log Q(z) - \log P(x|z) - \log P(z)] + \log P(x)$$
$$\log P(x) - \mathcal{KL}[Q(z)||P(z|x)] = \mathbb{E}_{z\sim Q}[\log P(x|z)] - \mathcal{KL}[Q(z)||P(z)]$$

What we have now:

- $\log P(x)$: maximization goal

Working through the math - 4

$$\mathcal{KL}[Q(z)||P(z|x)] = \mathbb{E}_{z\sim Q}[\log Q(z) - \log P(x|z) - \log P(z)] + \log P(x)$$
$$\log P(x) - \mathcal{KL}[Q(z)||P(z|x)] = \mathbb{E}_{z\sim Q}[\log P(x|z)] - \mathcal{KL}[Q(z)||P(z)]$$

What we have now:

- $\log P(x)$: maximization goal
- $\mathcal{KL}[Q(z)||P(z|x)]$: “closeness” of Q to P

Working through the math - 4

$$\mathcal{KL}[Q(z)||P(z|x)] = \mathbb{E}_{z\sim Q}[\log Q(z) - \log P(x|z) - \log P(z)] + \log P(x)$$
$$\log P(x) - \mathcal{KL}[Q(z)||P(z|x)] = \mathbb{E}_{z\sim Q}[\log P(x|z)] - \mathcal{KL}[Q(z)||P(z)]$$

What we have now:

- $\log P(x)$: maximization goal
- $\mathcal{KL}[Q(z)||P(z|x)$: “closeness” of Q to P
- $\mathbb{E}_{z\sim Q}[\log P(x|z)]$: maximization of $P(x|z)$ with regards to Q

Working through the math - 4

$$\mathcal{KL}[Q(z)||P(z|x)] = \mathbb{E}_{z\sim Q}[\log Q(z) - \log P(x|z) - \log P(z)] + \log P(x)$$
$$\log P(x) - \mathcal{KL}[Q(z)||P(z|x)] = \mathbb{E}_{z\sim Q}[\log P(x|z)] - \mathcal{KL}[Q(z)||P(z)]$$

What we have now:

- $\log P(x)$: maximization goal
- $\mathcal{KL}[Q(z)||P(z|x)$: “closeness” of Q to P
- $\mathbb{E}_{z\sim Q}[\log P(x|z)]$: maximization of $P(x|z)$ with regards to Q
- $\mathcal{KL}[Q(z)||P(z)]$: regularization of Q on prior $P(z)$

Working through the math - 4

$$\mathcal{KL}[Q(z)||P(z|x)] = \mathbb{E}_{z\sim Q}[\log Q(z) - \log P(x|z) - \log P(z)] + \log P(x)$$
$$\log P(x) - \mathcal{KL}[Q(z)||P(z|x)] = \mathbb{E}_{z\sim Q}[\log P(x|z)] - \mathcal{KL}[Q(z)||P(z)]$$

What we have now:

- $\log P(x)$: maximization goal
- $\mathcal{KL}[Q(z)||P(z|x)$: “closeness” of Q to P
- $\mathbb{E}_{z\sim Q}[\log P(x|z)]$: maximization of $P(x|z)$ with regards to Q
- $\mathcal{KL}[Q(z)||P(z)]$: regularization of Q on prior $P(z)$

- We can choose Q arbitrarily…

Working through the math - 4

$$\mathcal{KL}[Q(z)||P(z|x)] = \mathbb{E}_{z\sim Q}[\log Q(z) - \log P(x|z) - \log P(z)] + \log P(x)$$
$$\log P(x) - \mathcal{KL}[Q(z)||P(z|x)] = \mathbb{E}_{z\sim Q}[\log P(x|z)] - \mathcal{KL}[Q(z)||P(z)]$$

What we have now:

- $\log P(x)$: maximization goal
- $\mathcal{KL}[Q(z)||P(z|x)$: “closeness” of Q to P
- $\mathbb{E}_{z\sim Q}[\log P(x|z)]$: maximization of $P(x|z)$ with regards to Q
- $\mathcal{KL}[Q(z)||P(z)]$: regularization of Q on prior $P(z)$

- We can choose Q arbitrarily...
- ... so we can choose an entry which depends on x

Working through the math - 5

$$\log P(x) - D[Q(z|x)||P(z|x)] = \mathbb{E}_{z\sim Q}[\log P(x|z)] - D[Q(z|x)||P(z)]$$

Working through the math - 5

$$\log P(x) - D[Q(z|x)||P(z|x)] = \mathbb{E}_{z\sim Q}[\log P(x|z)] - D[Q(z|x)||P(z)]$$

- Right hand side: ELBO (Evidence Lower BOund): maximization target

Working through the math - 5

$$\log P(x) - D[Q(z|x)||P(z|x)] = \mathbb{E}_{z\sim Q}[\log P(x|z)] - D[Q(z|x)||P(z)]$$

- Right hand side: ELBO (Evidence Lower BOund): maximization target
- $P(x|z)$ and $Q(z|x)$: decoder and encoder learned from the data

Working through the math - 5

$$\log P(x) - D[Q(z|x)||P(z|x)] = \mathbb{E}_{z\sim Q}[\log P(x|z)] - D[Q(z|x)||P(z)]$$

- Right hand side: ELBO (Evidence Lower BOund): maximization target
- $P(x|z)$ and $Q(z|x)$: decoder and encoder learned from the data
- $P(z)$: prior on latent variables

Working through the math - 5

$$\log P(x) - D[Q(z|x)||P(z|x)] = \mathbb{E}_{z\sim Q}[\log P(x|z)] - D[Q(z|x)||P(z)]$$

- Right hand side: ELBO (Evidence Lower BOund): maximization target
- $P(x|z)$ and $Q(z|x)$: decoder and encoder learned from the data
- $P(z)$: prior on latent variables
- $P(z|x)$ will hopefully be approximated well by $Q(z|x)$. (discussion later)

Working through the math - 6

- Can we optimize now?

Working through the math - 6

- Can we optimize now?
- No, not yet: we still need to choose $Q(z|x)$ and work out some math

Working through the math - 6

- Can we optimize now?
- No, not yet: we still need to choose $Q(z|x)$ and work out some math

$$Q(z|x) = \mathcal{N}(z|\mu(x;\theta), \Sigma(x;\theta))$$

Working through the math - 6

- Can we optimize now?
- No, not yet: we still need to choose $Q(z|x)$ and work out some math

$$Q(z|x) = \mathcal{N}(z|\mu(x;\theta), \Sigma(x;\theta))$$

- $Q(z|x)$ can now approximate arbitrary PDF via μ

Working through the math - 6

- Can we optimize now?
- No, not yet: we still need to choose $Q(z|x)$ and work out some math

$$Q(z|x) = \mathcal{N}(z|\mu(x;\theta), \Sigma(x;\theta))$$

- $Q(z|x)$ can now approximate arbitrary PDF via μ
- Σ represents noise

Final math slide!

- With choices, $(P(x|z) \sim \mathcal{N}$ and $Q(z|x) \sim \mathcal{N})$, KL has closed form solution

Final math slide!

- With choices, ($P(x|z) \sim \mathcal{N}$ and $Q(z|x) \sim \mathcal{N}$), KL has closed form solution
    - possible for other PDFs too

Final math slide!

- With choices, ($P(x|z) \sim \mathcal{N}$ and $Q(z|x) \sim \mathcal{N}$), KL has closed form solution
  - possible for other PDFs too
- KL depends on learnable functions f and μ

Final math slide!

- With choices, $(P(x|z) \sim \mathcal{N}$ and $Q(z|x) \sim \mathcal{N})$, KL has closed form solution
  - possible for other PDFs too
- KL depends on learnable functions f and μ
- We can use neural networks to approximate those!

Final math slide!

- With choices, ($P(x|z) \sim \mathcal{N}$ and $Q(z|x) \sim \mathcal{N}$), KL has closed form solution
  - possible for other PDFs too
- KL depends on learnable functions f and μ
- We can use neural networks to approximate those!
- Are we finished now?

Variational Autoencoders

What is an autoencoder?

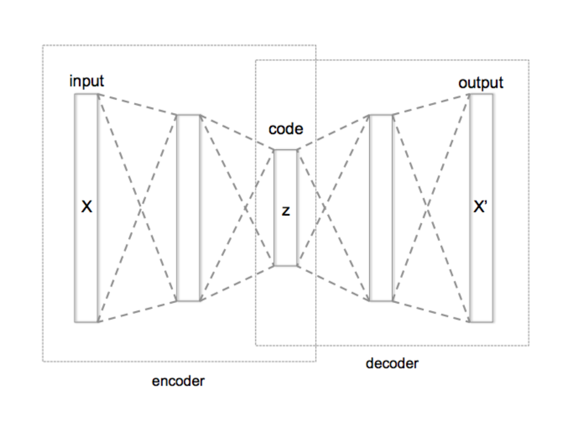


Figure 2: Taken from `https://commons.wikimedia.org/wiki/File:Autoencoder_structure.png`, (CC BY-SA 4.0)

What is a variational autoencoder?

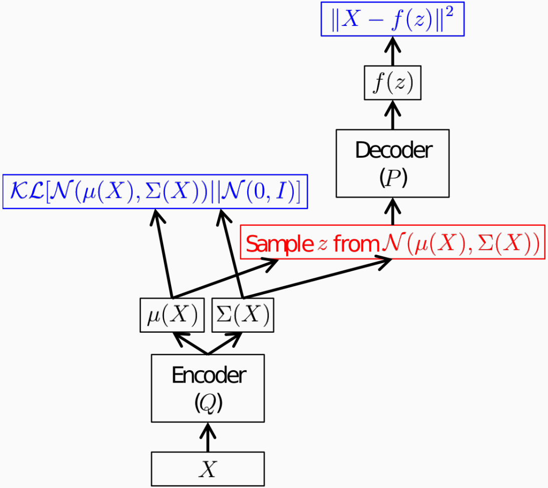
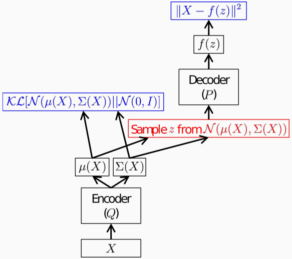
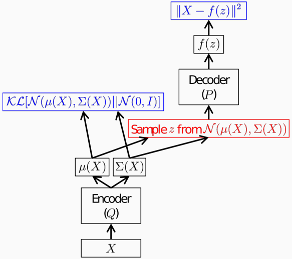


Figure 3: Taken from "Tutorial on Variational Autoencoders", Doersch, 2016

Where is the trick?

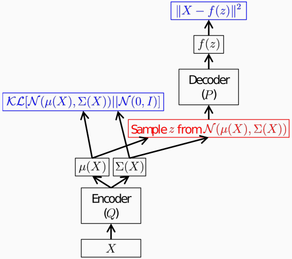


Where is the trick?



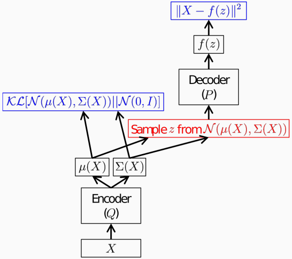
- ELBO is captured in the optimization by loss

Where is the trick?



- ELBO is captured in the optimization by loss
- The problem is in backpropagation

Where is the trick?



- ELBO is captured in the optimization by loss
- The problem is in backpropagation
- You can't propagate through a sampling layer

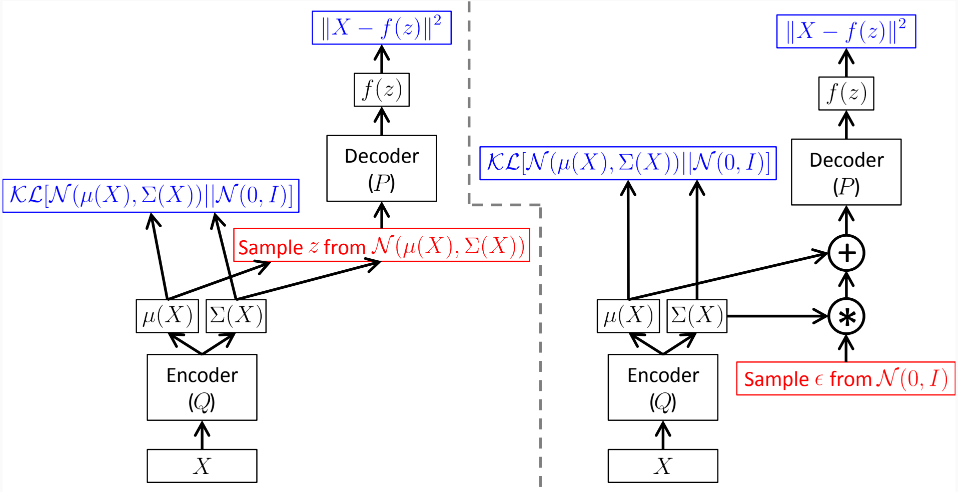


Figure 4: Taken from "Tutorial on Variational Autoencoders", Doersch, 2016

Why the sampling at all?

- We are trying to learn a distribution, not an encoder (per se)

Why the sampling at all?

- We are trying to learn a distribution, not an encoder (per se)
- Encoder - decoder relations are deterministic

Why the sampling at all?

- We are trying to learn a distribution, not an encoder (per se)
- Encoder - decoder relations are deterministic
- How would we sample at inference time?

Why the sampling at all?

- We are trying to learn a distribution, not an encoder (per se)
- Encoder - decoder relations are deterministic
- How would we sample at inference time?

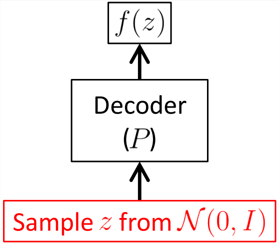


Figure 5: Taken from ”Tutorial on Variational Autoencoders”, Doersch, 2016

Why use a variational autoencoder?

- Image generation is often done via GAN

Why use a variational autoencoder?

- Image generation is often done via GAN
- Is VAE obsolete?

Why use a variational autoencoder?

- Image generation is often done via GAN
- Is VAE obsolete?

	VAE	GAN

Why use a variational autoencoder?

- Image generation is often done via GAN
- Is VAE obsolete?

	VAE	GAN
Loss	L2 loss	adversarial loss

Why use a variational autoencoder?

- Image generation is often done via GAN
- Is VAE obsolete?

	VAE	GAN
Loss	L2 loss	adversarial loss
Latent	structured	unstructured

Why use a variational autoencoder?

- Image generation is often done via GAN
- Is VAE obsolete?

	VAE	GAN
Loss	L2 loss	adversarial loss
Latent	structured	unstructured
Results	blurry	sharp

Why use a variational autoencoder?

- Image generation is often done via GAN
- Is VAE obsolete?

	VAE	GAN
Loss	L2 loss	adversarial loss
Latent	structured	unstructured
Results	blurry	sharp
Goal	generate good latent	generate good pictures

Why use a variational autoencoder?

- Image generation is often done via GAN
- Is VAE obsolete?

	VAE	GAN
Loss	L2 loss	adversarial loss
Latent	structured	unstructured
Results	blurry	sharp
Goal	generate good latent	generate good pictures

- VAEs are density models, GAN not so much

Why use a variational autoencoder?

- Image generation is often done via GAN
- Is VAE obsolete?

	VAE	GAN
Loss	L2 loss	adversarial loss
Latent	structured	unstructured
Results	blurry	sharp
Goal	generate good latent	generate good pictures

- VAEs are density models, GAN not so much
- But the density is still intractable

Why use a variational autoencoder?

- Image generation is often done via GAN
- Is VAE obsolete?

	VAE	GAN
Loss	L2 loss	adversarial loss
Latent	structured	unstructured
Results	blurry	sharp
Goal	generate good latent	generate good pictures

- VAEs are density models, GAN not so much
- But the density is still intractable
- There is no mathematical guarantee for choosing a ”good” latent

Why use a variational autoencoder?

- Image generation is often done via GAN
- Is VAE obsolete?

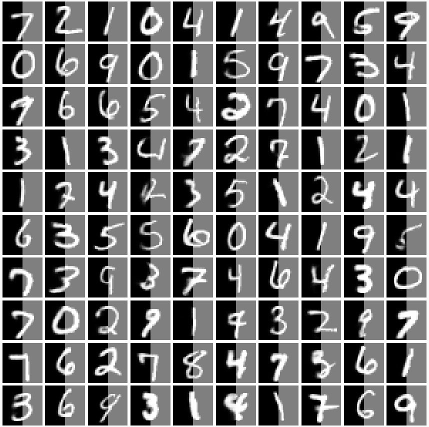
	VAE	GAN
Loss	L2 loss	adversarial loss
Latent	structured	unstructured
Results	blurry	sharp
Goal	generate good latent	generate good pictures

- VAEs are density models, GAN not so much
- But the density is still intractable
- There is no mathematical guarantee for choosing a ”good” latent
- L2 loss is computationally expensive, also does not capture ”visual closeness” well

Applications of VAE/VI

VAE for generating images

(a) Taken from "Tutorial on Variational Autoencoders", Doersch, 2016



(b) Taken from GitHub `https://github.com/yzwxx/vae-celebA`

VAE for translations

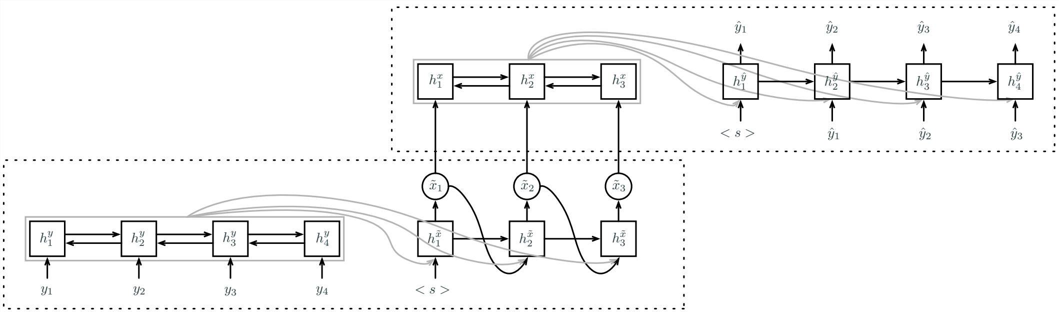


Figure 7: Taken from ”Semantic Parsing with Semi-Supervised Sequential Autoencoders”, Kocisky et al, 2016

More complex VAE usage

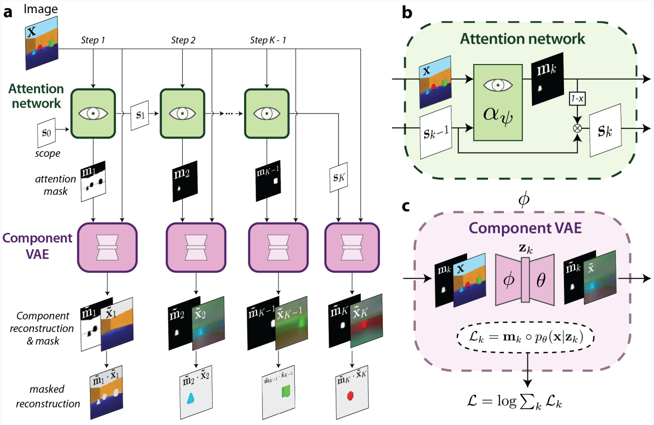


Figure 8: Taken from "MONet: Unsupervised Scene Decomposition and Representation", Burgess et al., 2019

More complex VAE usage

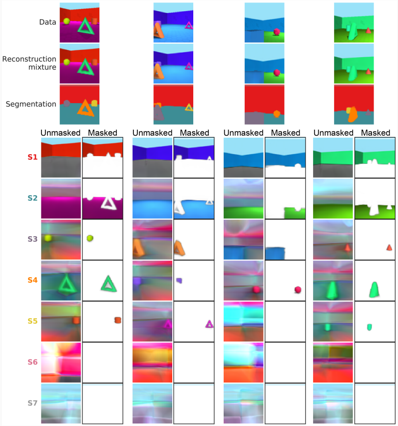
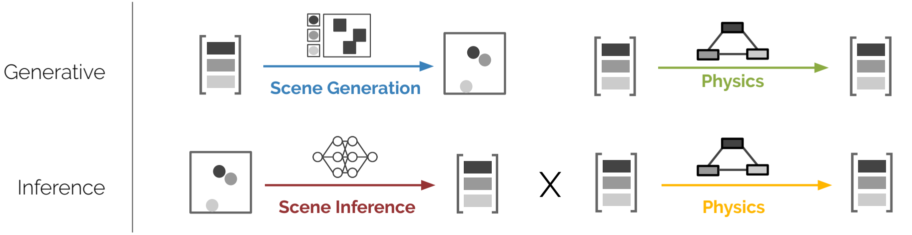


Figure 9: Taken from ”MONet: Unsupervised Scene Decomposition and Representation”, Burgess et al., 2019

The *STOVE* Model

Maximising the ELBO

$$\log p(x_{1:T}) \geq \mathbb{E}_{q(z_{1:T} \mid x_{1:T})}\left[\log \frac{p(x_{1:T}, z_{1:T})}{q(z_{1:T} \mid x_{1:T})}\right] \propto \mathbb{E}_{q(z_{1:T} \mid x_{1:T})}\left[\sum_{t=1}^{T} \log \left\{ \frac{p(x_t \mid z_t)\, p(z_t \mid z_{t-1})}{q(z_t \mid x_t)\, q(z_t \mid z_{t-1})} \right\}\right]$$



Kossen et al., Structured Object-Aware Physics Prediction for Video Modeling and Planning

Further reading

- ”Stochastic Backpropagation and Approximate Inference in Deep Generative Models” by Danilo Rezende et al., 2014
- ”Auto-Encoding Variational Bayes” by Durk Kingma and Max Welling, 2014
- ”Tutorial on Variational Autoencoders” by Carl Doersch, 2016
- ”Variational Inference: A Review for Statisticians” by David Blei et al., 2018
- ”Improving Variational Inference with Inverse Autoregressive Flow” by Durk Kingma, 2016
- ”Attend, Infer, Repeat: Fast Scene Understanding with Generative Models” by Ali Eslami et al., 2016
- ”The Dreaming Variational Autoencoder for Reinforcement Learning Environments” by Per-Arne Andersen, 2018

Questions?

Any remaining questions?