Variational Autoencoders

Learning generative models with latent representations

Claas Völcker

April 30, 2019

Deep Generative Models - SoSe 2019

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

$$\text{Maximize: } P(X) \sim \int p_\phi(X|z)p_\phi(z)dz \tag{1}$$

Working through the math - 1

$$\text{Maximize: } P(X) \sim \int p_\phi(X|z)p_\phi(z)dz \tag{1}$$

- a latent variable model

Working through the math - 1

$$\text{Maximize: } P(X) \sim \int p_\phi(X|z)p_\phi(z)dz \tag{1}$$

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

Working through the math - 1

$$\text{Maximize: } P(X) \sim \int p_\phi(X|z)p_\phi(z)dz \tag{1}$$

- a latent variable model
- Assumption: there is some (hopefully small) z which governs data
- define $P(z)$, $P(X|z)$ and $\int$?

Working through the math - 1

$$\text{Maximize: } P(X) \sim \int p_\phi(X|z)p_\phi(z)dz \tag{1}$$

- a latent variable model
- Assumption: there is some (hopefully small) z which governs data
- define $P(z)$, $P(X|z)$ and $\int$?
- first intuition: choose "easy" $P(z)$ (like Gaussian), learn $P(X|z)$ from the data

Working through the math - 1

$$\text{Maximize: } P(X) \sim \int p_\phi(X|z)p_\phi(z)dz \tag{1}$$

- a latent variable model
- Assumption: there is some (hopefully small) z which governs data
- define $P(z)$, $P(X|z)$ and $\int$?
- first intuition: choose "easy" $P(z)$ (like Gaussian), learn $P(X|z)$ from the data

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

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

Working through the math - 2

- We could maximize by sampling?

Working through the math - 2

- We could maximize by sampling?
- No, too many samples would be needed

Working through the math - 2

- We could maximize by sampling?
- No, too many samples would be needed
- A given z is never guaranteed to produce a likely X

Working through the math - 2

- We could maximize by sampling?
- No, too many samples would be needed
- A given z is never guaranteed to produce a likely X
- We are sampling far from the useful mainfold

Working through the math - 2

- We could maximize by sampling?
- No, too many samples would be needed
- A given z is never guaranteed to produce a likely X
- We are sampling far from the useful mainfold
- $P(Z|X)$ is intractable, it is the reverse of what we are looking for

Working through the math - 2

- We could maximize by sampling?
- No, too many samples would be needed
- A given z is never guaranteed to produce a likely X
- We are sampling far from the useful mainfold
- $P(Z|X)$ is intractable, it is the reverse of what we are looking for
- We need to learn a surrogate probability density $Q(Z)$

Working through the math - 2

- We could maximize by sampling?
- No, too many samples would be needed
- A given z is never guaranteed to produce a likely X
- We are sampling far from the useful mainfold
- $P(Z|X)$ is intractable, it is the reverse of what we are looking for
- We need to learn a surrogate probability density $Q(Z)$
- $Q(Z)$ should be close to $P(Z|X)$

Working through the math - 3

$$\mathcal{KL}[Q(Z)||P(Z|X)] = \mathbb{E}_{z\sim Q}[\log Q(Z) - \log P(Z|X)] \tag{3}$$

Working through the math - 3

$$\mathcal{KL}[Q(Z)||P(Z|X)] = \mathbb{E}_{z\sim Q}[\log Q(Z) - \log P(Z|X)] \tag{3}$$

- Using Bayes rule:

Working through the math - 3

$$\mathcal{KL}[Q(Z)||P(Z|X)] = \mathbb{E}_{z\sim Q}[\log Q(Z) - \log P(Z|X)] \tag{3}$$

- Using Bayes rule:

$$\mathcal{KL}[Q(Z)||P(Z|X)] = \mathbb{E}_{Z\sim Q}[\log Q(Z) - \log \frac{P(X|Z)P(Z)}{P(X)}] \tag{4}$$

$$= \mathbb{E}_{Z\sim Q}[\log Q(Z) - (\log P(X|Z) + \log P(Z) - \log P(X))] \tag{5}$$

$$= \mathbb{E}_{Z\sim Q}[\log Q(Z) - \log P(X|Z) - \log P(Z)] + \log P(X) \tag{6}$$

Working through the math - 4

$$\mathcal{KL}[Q(Z)||P(Z|X)] = \mathbb{E}_{Z\sim Q}[\log Q(Z) - \log P(X|Z) - \log P(Z)] + \log P(X) \tag{7}$$

$$\log P(X) - \mathcal{KL}[Q(Z)||P(Z|X)] = \mathbb{E}_{Z\sim Q}[\log P(X|Z)] - \mathcal{KL}[Q(Z)||P(Z)] \tag{8}$$

Working through the math - 4

$$\mathcal{KL}[Q(Z)||P(Z|X)] = \mathbb{E}_{Z\sim Q}[\log Q(Z) - \log P(X|Z) - \log P(Z)] + \log P(X) \tag{7}$$

$$\log P(X) - \mathcal{KL}[Q(Z)||P(Z|X)] = \mathbb{E}_{Z\sim Q}[\log P(X|Z)] - \mathcal{KL}[Q(Z)||P(Z)] \tag{8}$$

What we have now:

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

Working through the math - 4

$$\mathcal{KL}[Q(Z)||P(Z|X)] = \mathbb{E}_{Z\sim Q}[\log Q(Z) - \log P(X|Z) - \log P(Z)] + \log P(X) \tag{7}$$

$$\log P(X) - \mathcal{KL}[Q(Z)||P(Z|X)] = \mathbb{E}_{Z\sim Q}[\log P(X|Z)] - \mathcal{KL}[Q(Z)||P(Z)] \tag{8}$$

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

$$\mathcal{KL}[Q(Z)||P(Z|X)] = \mathbb{E}_{Z\sim Q}[\log Q(Z) - \log P(X|Z) - \log P(Z)] + \log P(X) \tag{7}$$

$$\log P(X) - \mathcal{KL}[Q(Z)||P(Z|X)] = \mathbb{E}_{Z\sim Q}[\log P(X|Z)] - \mathcal{KL}[Q(Z)||P(Z)] \tag{8}$$

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

$$\mathcal{KL}[Q(Z)||P(Z|X)] = \mathbb{E}_{Z\sim Q}[\log Q(Z) - \log P(X|Z) - \log P(Z)] + \log P(X) \tag{7}$$

$$\log P(X) - \mathcal{KL}[Q(Z)||P(Z|X)] = \mathbb{E}_{Z\sim Q}[\log P(X|Z)] - \mathcal{KL}[Q(Z)||P(Z)] \tag{8}$$

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

$$\mathcal{KL}[Q(Z)||P(Z|X)] = \mathbb{E}_{Z\sim Q}[\log Q(Z) - \log P(X|Z) - \log P(Z)] + \log P(X) \tag{7}$$

$$\log P(X) - \mathcal{KL}[Q(Z)||P(Z|X)] = \mathbb{E}_{Z\sim Q}[\log P(X|Z)] - \mathcal{KL}[Q(Z)||P(Z)] \tag{8}$$

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

$$\mathcal{KL}[Q(Z)||P(Z|X)] = \mathbb{E}_{Z\sim Q}[\log Q(Z) - \log P(X|Z) - \log P(Z)] + \log P(X) \tag{7}$$

$$\log P(X) - \mathcal{KL}[Q(Z)||P(Z|X)] = \mathbb{E}_{Z\sim Q}[\log P(X|Z)] - \mathcal{KL}[Q(Z)||P(Z)] \tag{8}$$

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

$$\log P(X) - D[Q(Z|X)||P(Z|X)] = \mathbb{E}_{Z\sim Q}[\log P(X|z)] - D[Q(z|X)||P(z)] \tag{9}$$

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

Working through the math - 5

$$\log P(X) - D[Q(Z|X)||P(Z|X)] = \mathbb{E}_{Z\sim Q}[\log P(X|z)] - D[Q(z|X)||P(z)] \tag{9}$$

- 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

$$\log P(X) - D[Q(Z|X)||P(Z|X)] = \mathbb{E}_{Z\sim Q}[\log P(X|z)] - D[Q(z|X)||P(z)] \tag{9}$$

- 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

$$\log P(X) - D[Q(Z|X)||P(Z|X)] = \mathbb{E}_{Z\sim Q}[\log P(X|z)] - D[Q(z|X)||P(z)] \tag{9}$$

- 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

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

$$Q(z|X) = \mathcal{N}(z|\mu(X;\theta), \Sigma(x;\theta)) \tag{10}$$

- $Q(Z|X)$ 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

$$Q(z|X) = \mathcal{N}(z|\mu(X;\theta), \Sigma(x;\theta)) \tag{10}$$

- $Q(Z|X)$ 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?

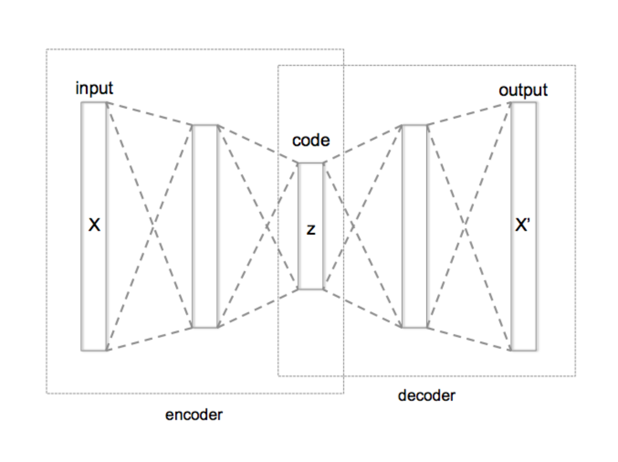


Figure 1: Taken from `https://commons.wikimedia.org/wiki/File:Autoencoder_structure.png`, (CC BY-SA 4.0)

What is a variational autoencoder?

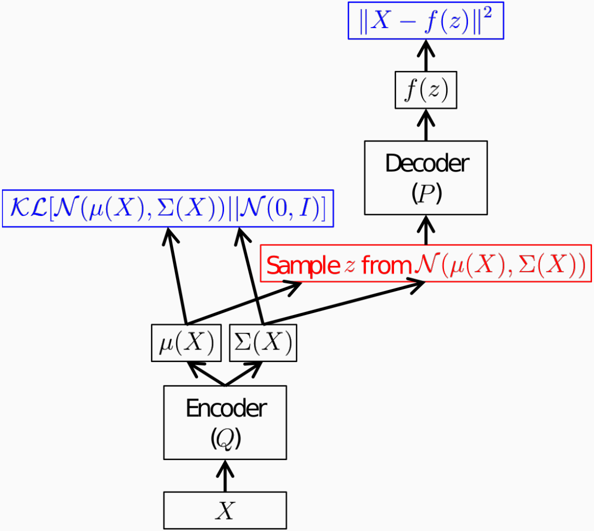
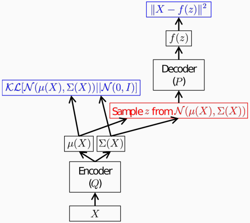
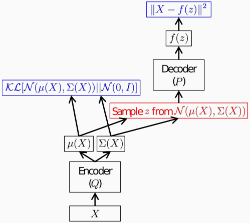


Figure 2: Taken from "Auto-Encoding Variational Bayes", Kingma & Welling, 2014

Where is the trick?

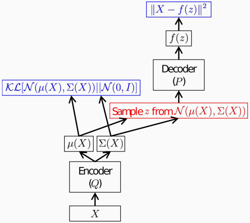


Where is the trick?



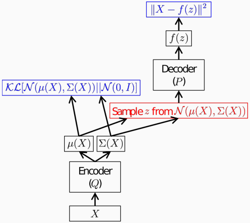
- ELBO is captured in the optimization by loss

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

All the parts in detail

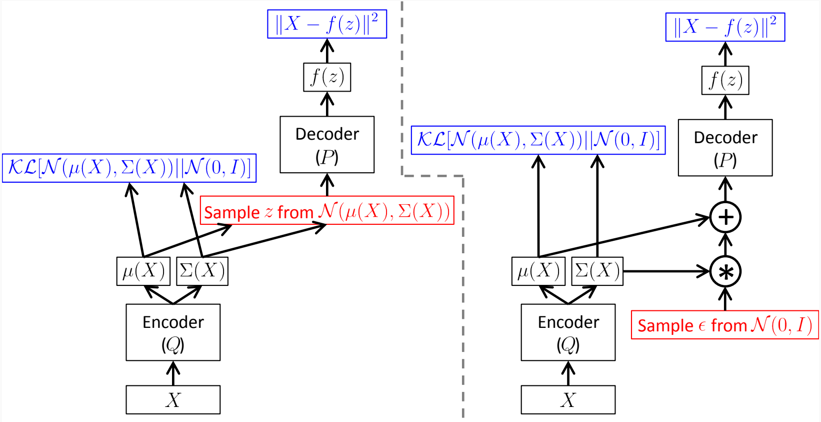


Figure 3: Taken from "Tutorial on Variational Autoencoders", Doersch, 2016

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?

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?

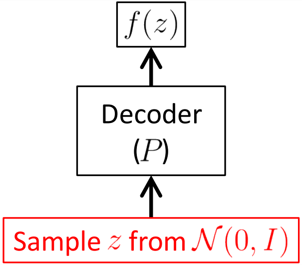


Figure 4: Taken from "Tutorial on Variational Autoencoders", Doersch, 2016

Code!

Code presentation

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?

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?

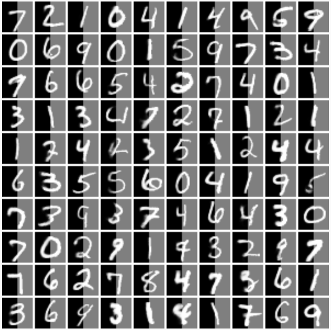
	VAE	GAN
Loss	L2 loss	adversarial loss
Latent	structured	unstructured
Results	blurry	sharp
Goal	generate good latent	generate good pictures

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

Applications of VAE

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

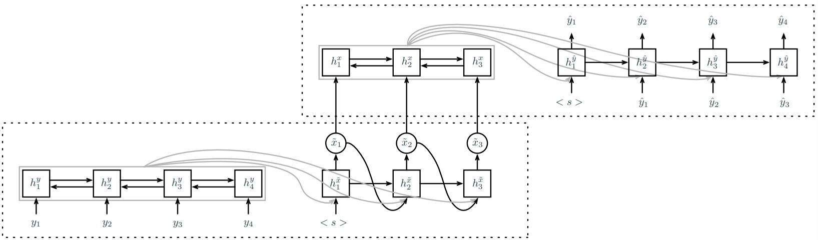


Figure 6: Taken from "Semantic Parsing with Semi-Supervised Sequential Autoencoders", Kocisky et al, 2016

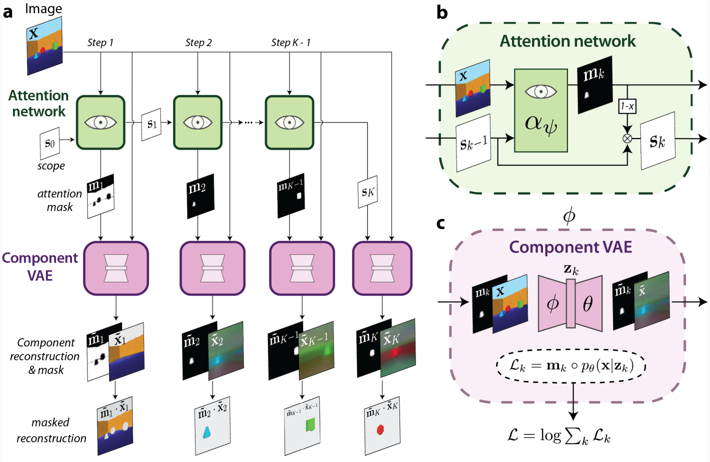


Figure 7: Taken from "MONet: Unsupervised Scene Decomposition and Representation", Burgess et al., 2019

More complex VAE usage

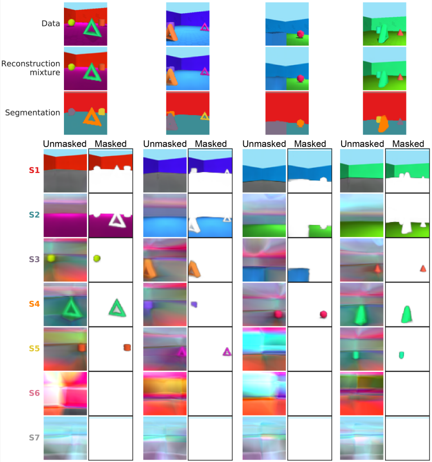


Figure 8: Taken from "MONet: Unsupervised Scene Decomposition and Representation", Burgess et al., 2019

Further reading

- "Stochastic Backpropagation and Approximate Inferencein 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?

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

$$KL(P||Q) = \int p(x) \log \frac{p(x)}{q(x)} dx \tag{11}$$

$$= \mathbb{E}_{x \sim p}[\log \frac{p(x)}{q(x)}] = \mathbb{E}_{x \sim p}[\log(p(x)) - \log(q(x))] \tag{12}$$

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

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

Kullback Leibler Divergence

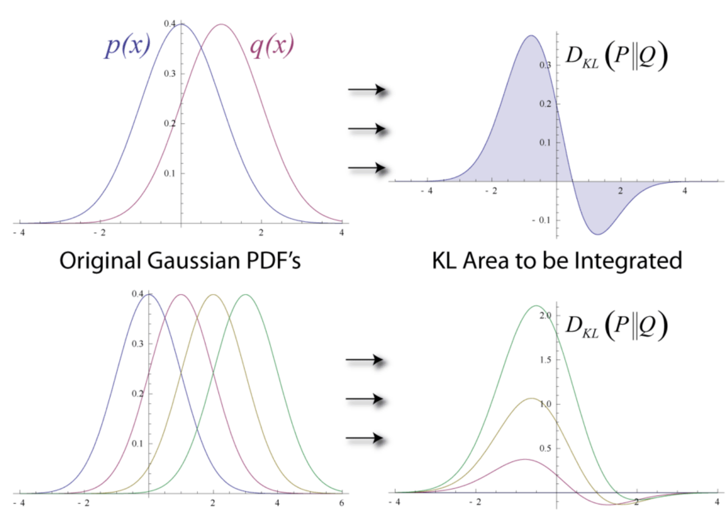


Figure 9: Taken from "Kullback–Leibler divergence", Wikipedia, CC BY-SA 3.0