The power and promise of generative models

Mihaela Rosca
@elaClaudia

DeepMind

Generative models are a core building block of intelligent systems.

What do intelligent systems need?

- Generate new data
- Imagine possible futures & have a model of the world
- Translate between data modalities
- Learn useful representations
- Complete missing data

Generate new data

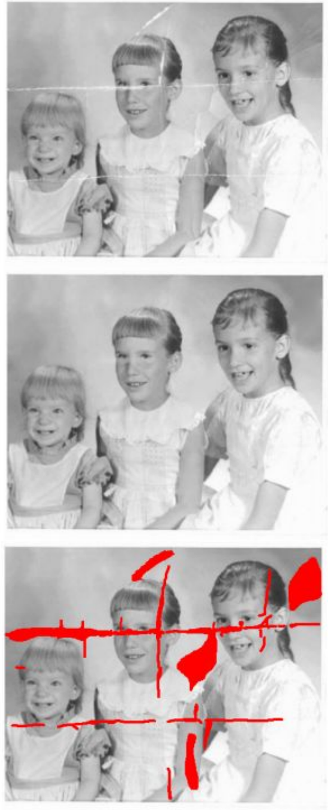
What do intelligent systems need?

Royour was I did too jest of this forget
That I must I should be report of tale
Decost we are bewarved:'d: yet my fearful scope
From whence the duty I may need their course,
Which thou wert sorry for my party was to show
Forthwith Edward for what stout King Richard death!

https://arxiv.org/abs/1710.10196
https://arxiv.org/pdf/1609.03499.pdf
https://arxiv.org/abs/1308.0850

Complete missing data

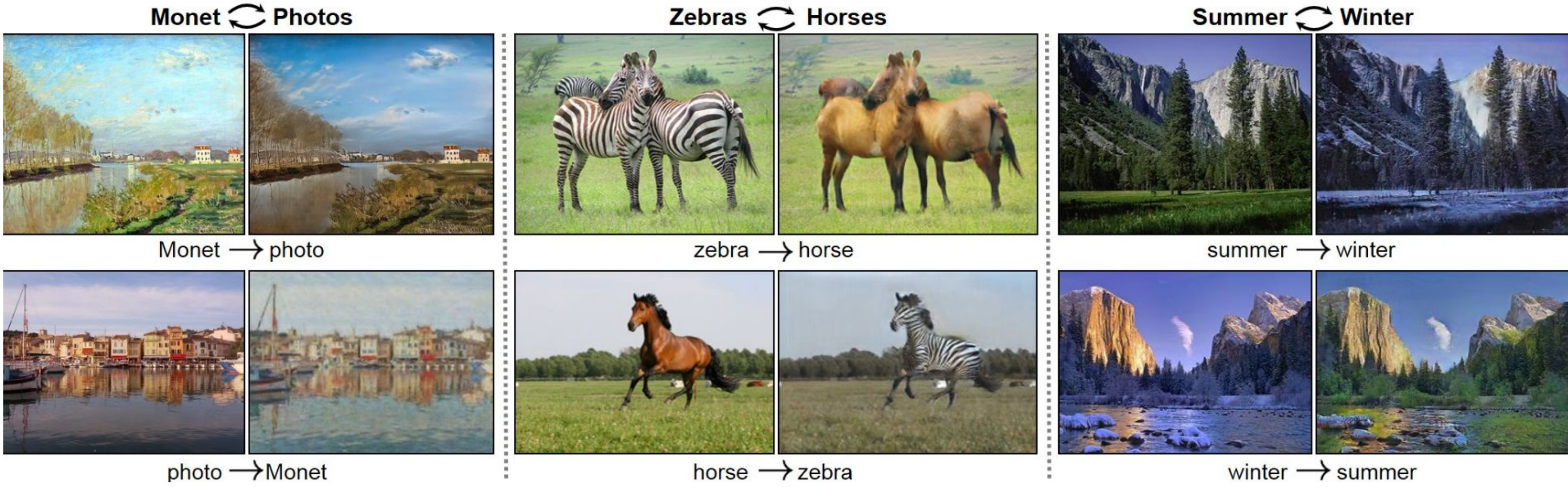
What do intelligent systems need?



http://math.univ-lyon1.fr/homes-www/masnou/fichiers/publications/survey.pdf
http://www.dtic.upf.edu/~mbertalmio/bertalmi.pdf

Translate between data modalities

What do intelligent systems need?



CycleGAN: https://arxiv.org/abs/1703.10593

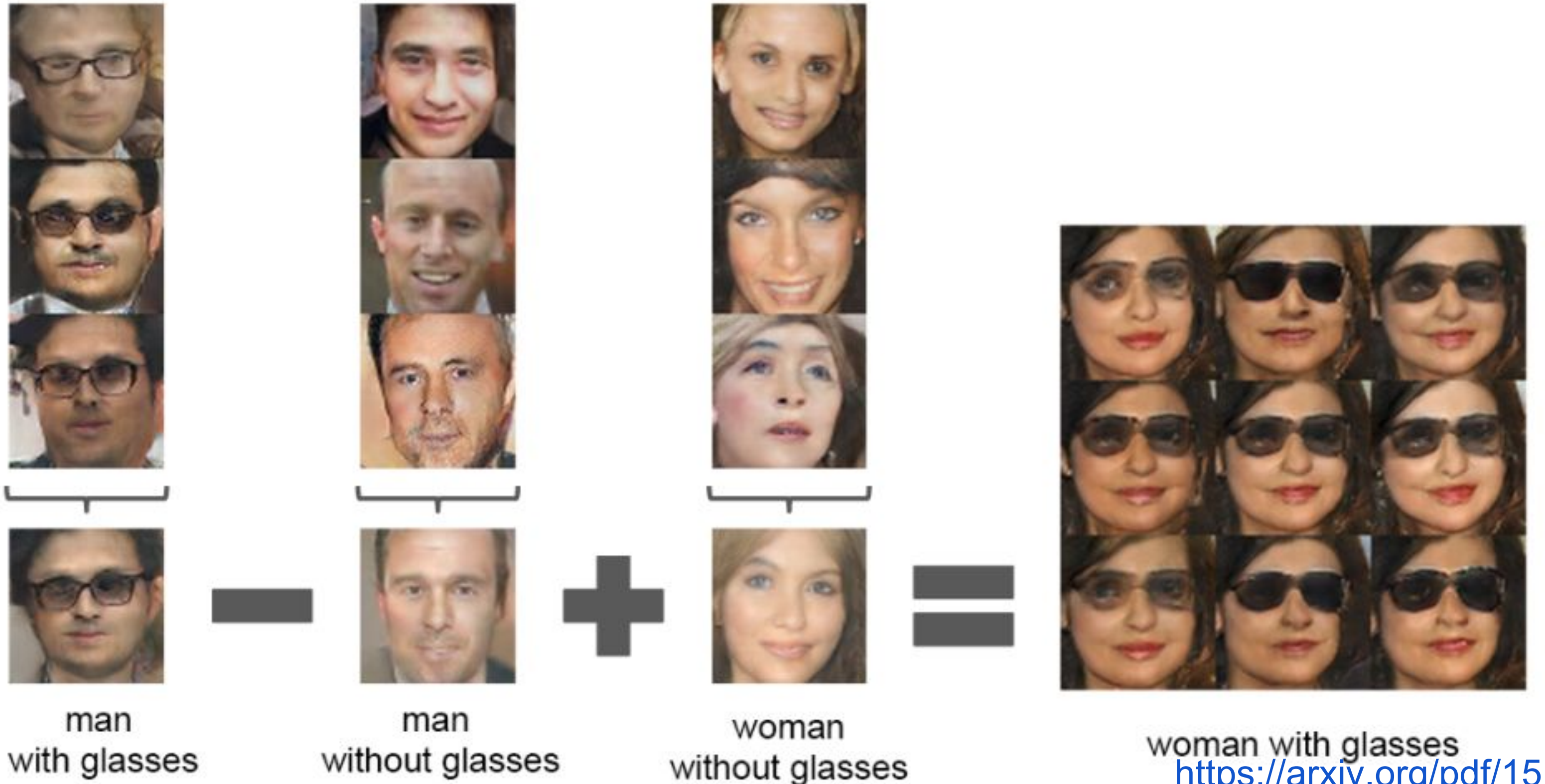
Learn useful representations

What do intelligent systems need?

Leverage learned structure for better classification performance on labelled data.

Learn useful representations

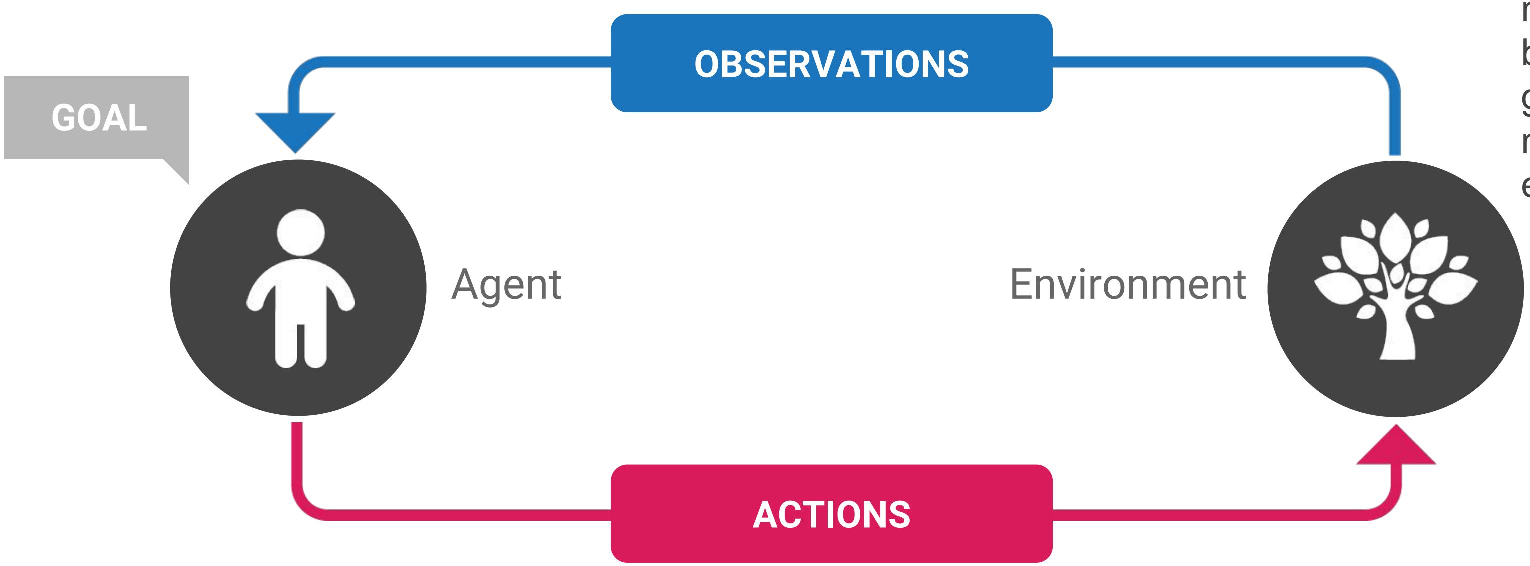
What do intelligent systems need?



https://arxiv.org/pdf/1511.06434.pdf

Have a model of the world

What do intelligent systems need?



Generative models can be used to generate a model of the environment.

How do generative models allow us to build intelligent systems?

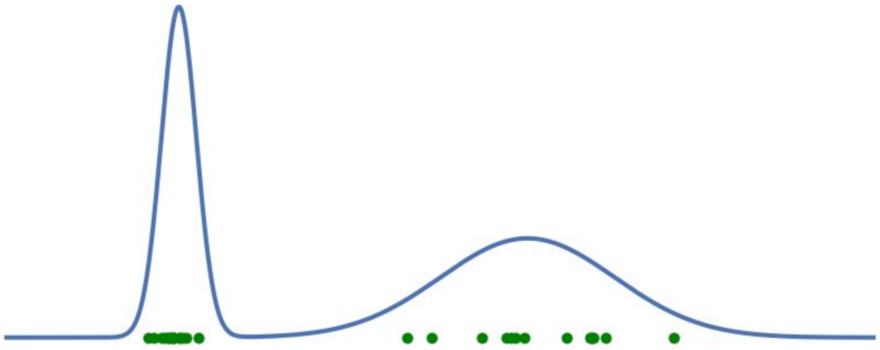
The goal of generative models

Learn a model of the true underlying data distribution $p^*(x)$ from samples

$$x_1, x_2 \dots x_n$$

The goal of generative models

The goal of generative models



The goal of generative models

The goal of generative models

Find p_θ to minimize the distance between p_θ and p^*

Finding p_θ

Choices in generative models

- Model of p_θ
  - you can leverage prior knowledge of the problem
    - what kind of data do you have?
    - what kind of process generated the data?

- The learning principle used to minimize the distance between p_θ and p^*

Generative model algorithm

=

learning principle + model

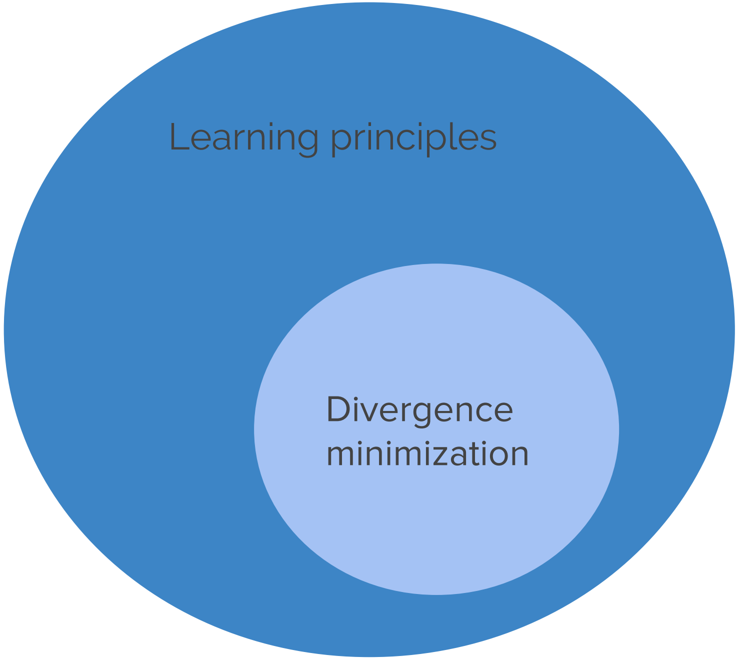
Learning principle

Model of p_θ

	Moment matching	Maximum likelihood	...	Optimal transport
Autoregressive				
Implicit		Algorithm		
...				
Encoder-decoder				

Learning principles

Divergence minimization as a learning principle



Other learning principles:
moment matching
optimal transport

Divergences minimized

Aim: Minimize a divergence between p_θ and p^*

Divergence minimized

Requirements

- Easy to compute
- Needs only samples from p*
- Has an efficient unbiased gradient estimator

Divergences minimized

Common divergence choices

- KL divergence (most common)
    - results in *maximum likelihood* learning
- Reverse KL divergence
- Jensen Shannon

KL divergence

Maximum likelihood

Minimizing the KL divergence between p_θ and p^* ⇒ maximum likelihood learning:

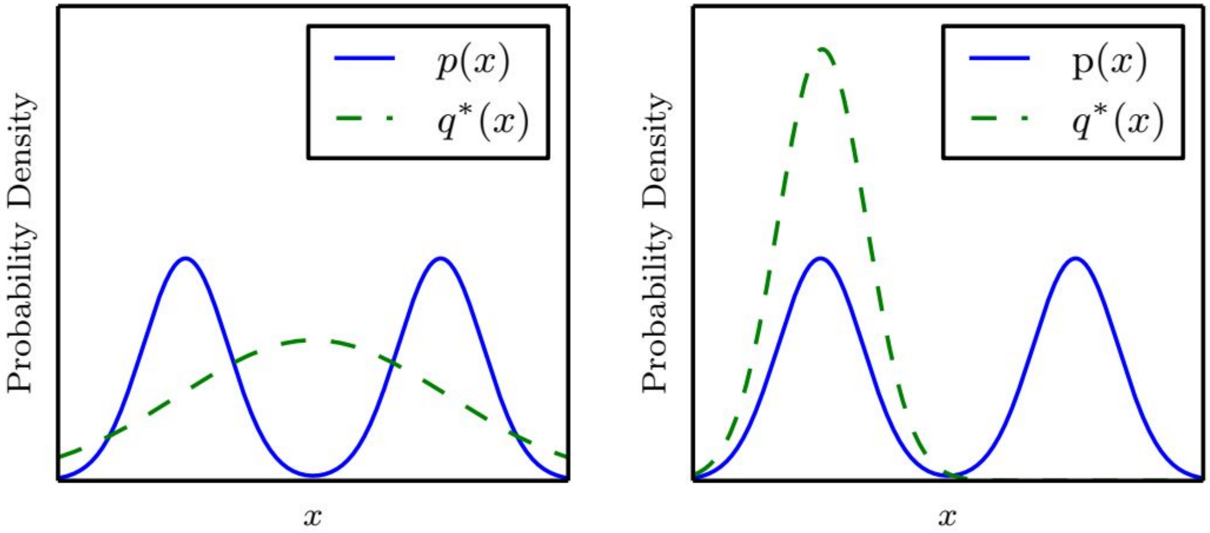
$$\text{argmax}_\theta \, \mathcal{E}_{x \sim p^*} \log p_\theta(x)$$

Intuition: find the model which gives highest likelihood to the data.

Trade-offs for a fixed model choice

KL vs Reverse KL model fit

$q^* = \mathrm{argmin}_q D_{\mathrm{KL}}(p\|q)$ $\qquad$ $q^* = \mathrm{argmin}_q D_{\mathrm{KL}}(q\|p)$



Maximum likelihood $\qquad$ Reverse KL

https://arxiv.org/pdf/1701.00160.pdf

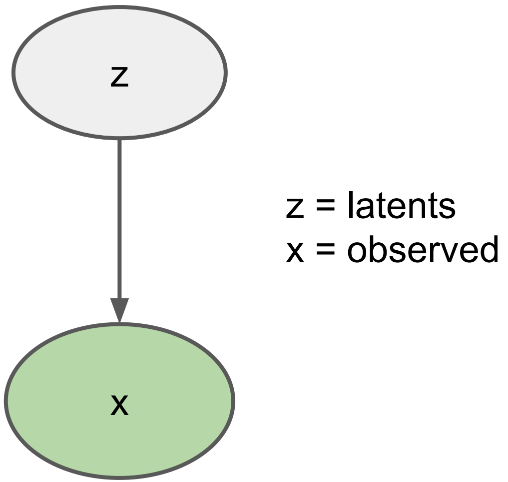
Model choices

Model choices

Graphical structure

Directly model $p_\theta(x)$

Leverage underlying data structure in generative process.



Model choices

Distribution choice

$p_\theta(x)$

- Categorical
- Gaussian
- Bernoulli
- do not directly model $p_\theta(x)$

Model choices

Embedding priors in your model

Leverage data knowledge:

- convolutional models for images
- recurrent models for text and sound
- appropriate priors for latent variables

LDA

Embedding priors in your model

"Arts" "Budgets" "Children" "Education"

The William Randolph Hearst Foundation will give $1.25 million to Lincoln Center, Metropolitan Opera Co., New York Philharmonic and Juilliard School. "Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants an act every bit as important as our traditional areas of support in health, medical research, education and the social services," Hearst Foundation President Randolph A. Hearst said Monday in announcing the grants. Lincoln Center's share will be $200,000 for its new building, which will house young artists and provide new public facilities. The Metropolitan Opera Co. and New York Philharmonic will receive $400,000 each. The Juilliard School, where music and the performing arts are taught, will get $250,000. The Hearst Foundation, a leading supporter of the Lincoln Center Consolidated Corporate Fund, will make its usual annual $100,000 donation, too.

Algorithms

Autoregressive maximum likelihood models

PixelCNN, PixelRNN, WaveNet

Autoregressive maximum likelihood models

Principle: maximum likelihood

Model: Autoregressive model with no latent variables

$$p_\theta(x) = \prod_i p_\theta(x_i | x_1, x_2 \ldots x_{i-1})$$

PixelCNN, PixelRNN, WaveNet

Autoregressive maximum likelihood models

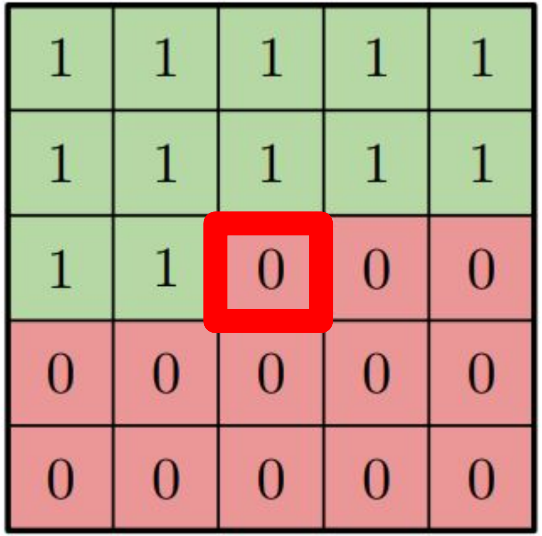
$$p_\theta(x) = \Pi_i \underbrace{p_\theta(x_i | x_1, x_2 \ldots x_{i-1})}$$

Modelled with a convolutional or recurrent network

https://arxiv.org/pdf/1601.06759.pdf
https://arxiv.org/abs/1606.05328
https://deepmind.com/blog/wavenet-generative-model-raw-audio/

PixelCNN, PixelRNN, WaveNet

Sampling from autoregressive maximum likelihood models



O(data_dim) sampling cost.

https://arxiv.org/pdf/1601.06759.pdf
https://arxiv.org/abs/1606.05328
https://deepmind.com/blog/wavenet-generative-model-raw-audio/

Autoregressive maximum likelihood models

Pros:

- powerful - state of the art for many applications
- explicit, exact density models

Cons:

- High sampling cost

Mihaela Rosca

Variational autoencoders

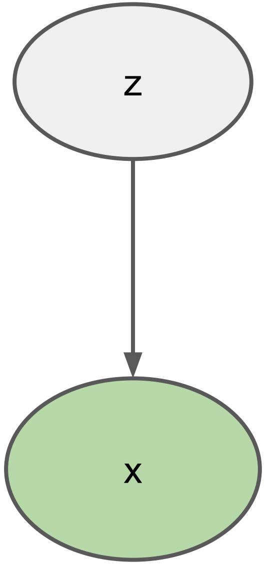
Variational autoencoders

Maximum likelihood for latent variable models

Principle: maximum likelihood

Model: Encoder-decoder model with latent variables

$$\mathbb{E}_{p^*(\mathbf{x})} \log p_{\boldsymbol{\theta}}(\mathbf{x})$$

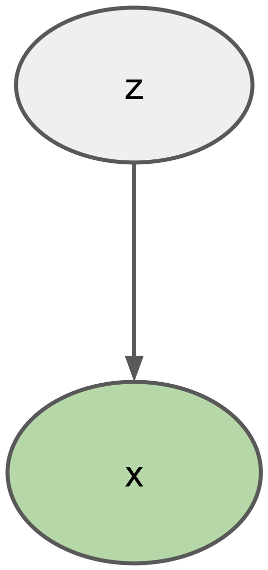


Variational autoencoders

Maximum likelihood for latent variable models

Latent variables introduce an intractable integral:

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) = \log \int p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$$



Variational autoencoders

Maximum likelihood for latent variable models

A solution is to introduce a variational distribution *q*:

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) = \log \int p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z} \geq \mathbb{E}_{q_\eta(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})] - \mathbf{KL}[q_\eta(\mathbf{z}|\mathbf{x})||p(\mathbf{z})]$$

Variational autoencoders

Maximum likelihood for latent variable models

A solution is to introduce a variational distribution *q*:

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) = \log \int p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z} \geq \underbrace{\mathbb{E}_{q_\eta(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})]}_{\text{Encode information about x - make sampling efficient}} - \underbrace{\mathrm{KL}[q_\eta(\mathbf{z}|\mathbf{x})||p(\mathbf{z})]}_{\text{Stay close to the prior}}$$

Variational autoencoders

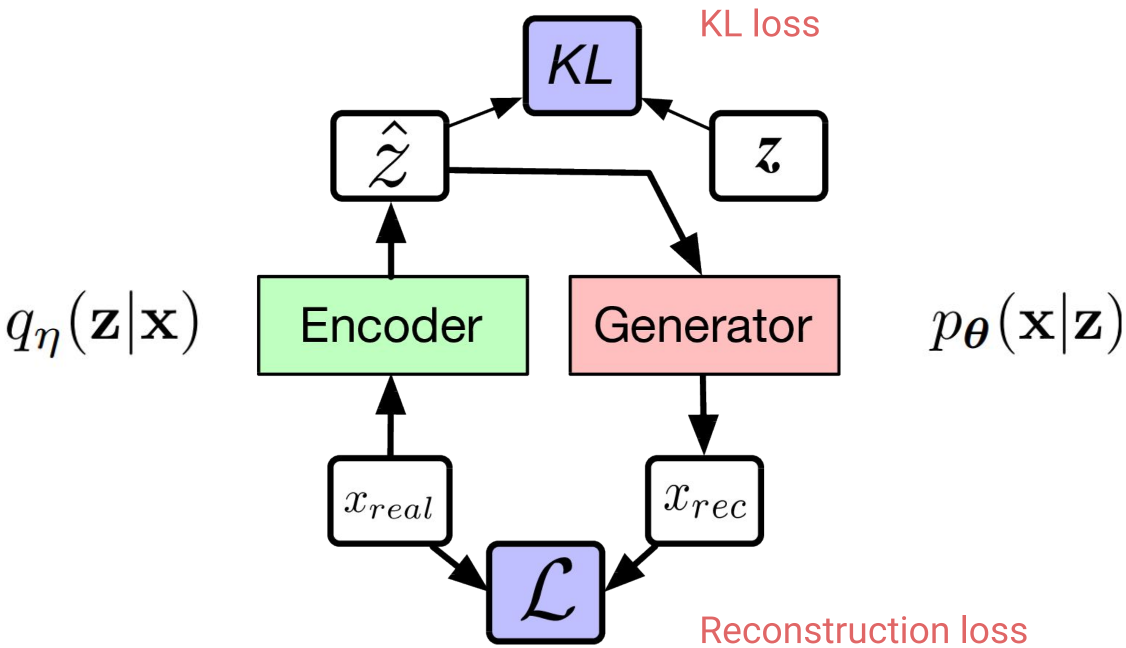
Maximum likelihood for latent variable models

A solution is to introduce a variational distribution *q*:

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) = \log \int p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z} \geq \underbrace{\mathbb{E}_{q_\eta(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})]}_{\text{reconstruction loss}} - \underbrace{\mathbf{KL}[q_\eta(\mathbf{z}|\mathbf{x})||p(\mathbf{z})]}_{\text{KL loss}}$$

Variational autoencoders

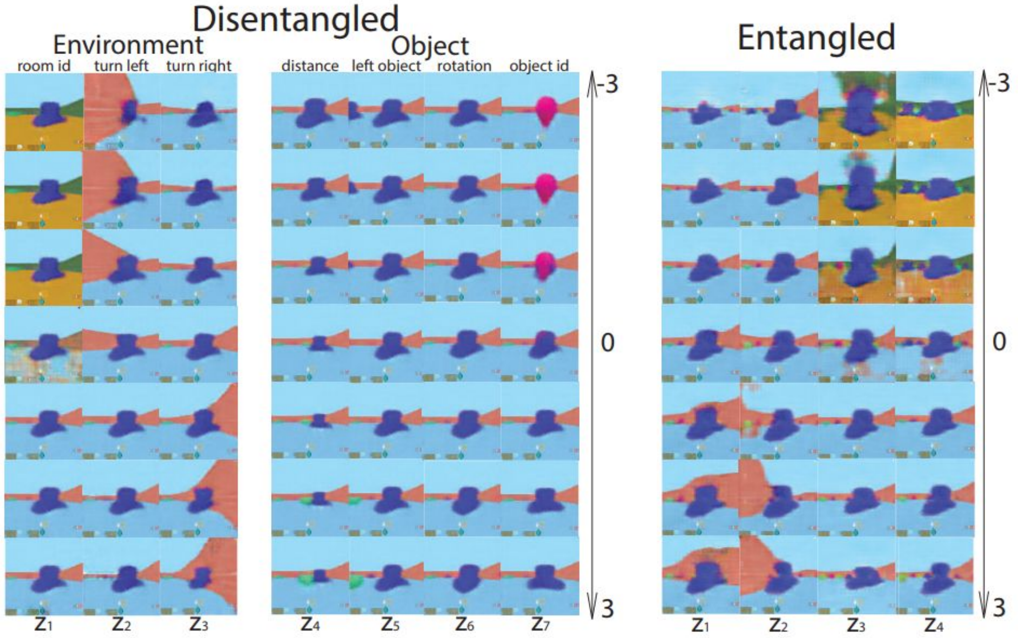
Latent approximate maximum likelihood models



Variational autoencoders

Pros:

- inference

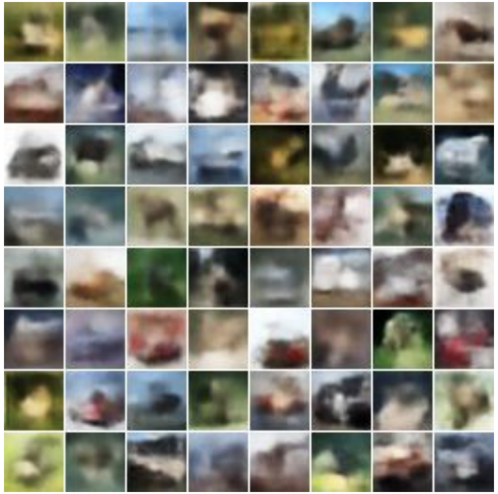


https://openreview.net/forum?id=Sy2fzU9gl

Variational autoencoders

Cons:

- Approximate density estimation
- Sensitive to choice of posterior distribution
- Low quality samples

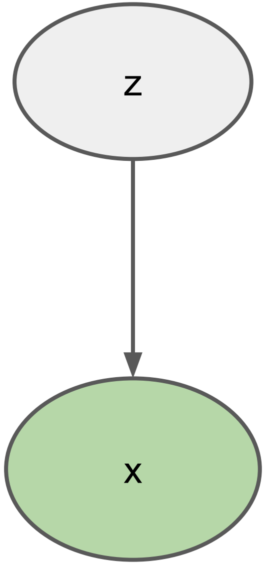


Generative adversarial networks

Generative adversarial networks

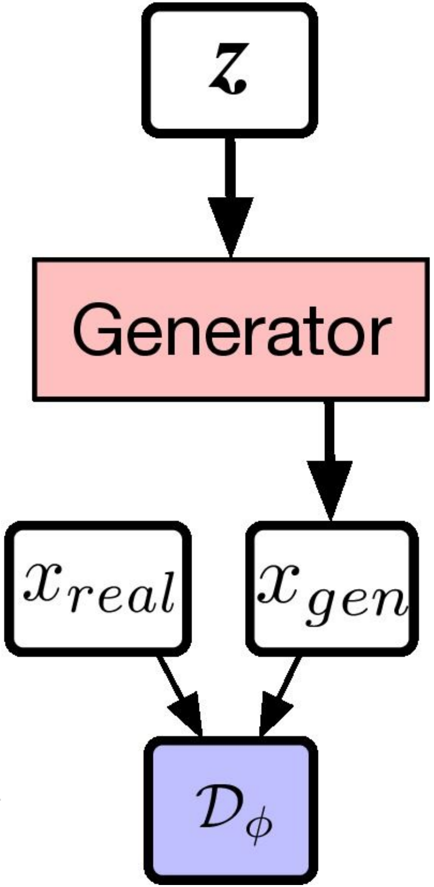
Latent variable models without maximum likelihood

- Minimize distance between p_θ and p^*
  - provided by another model "discriminator"
  - connections to Jensen Shannon and Earth Mover's
- How they model p_θ
  - model the generative process: sampling
  - no direct access to p_θ



https://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

Generative adversarial networks



https://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

Generative adversarial networks

The model objective

D is a classifier trained with cross entropy loss.

$$J^{(D)}(\boldsymbol{\theta}^{(D)}, \boldsymbol{\theta}^{(G)}) = \underbrace{-\frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}} \log D(\boldsymbol{x})}_{\text{Maximize probability that the data is real}} \underbrace{- \frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log\left(1 - D\left(G(\boldsymbol{z})\right)\right)}_{\text{Maximize probability that the samples are fake}}.$$

Generative adversarial networks

Connection to Jensen Shannon divergence

If D is an optimal discriminator:

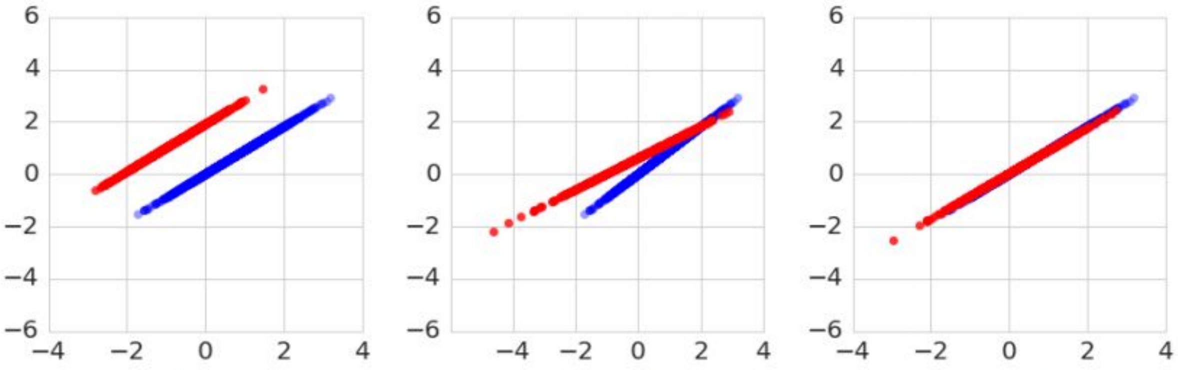
G is minimizing JSD(p_θ,p^*)

Generative adversarial networks

Connection to Jensen Shannon divergence

In practice:

- simultaneous gradient descent
- finite data



(a) Step 0 (b) Step 5000 (c) Step 12500

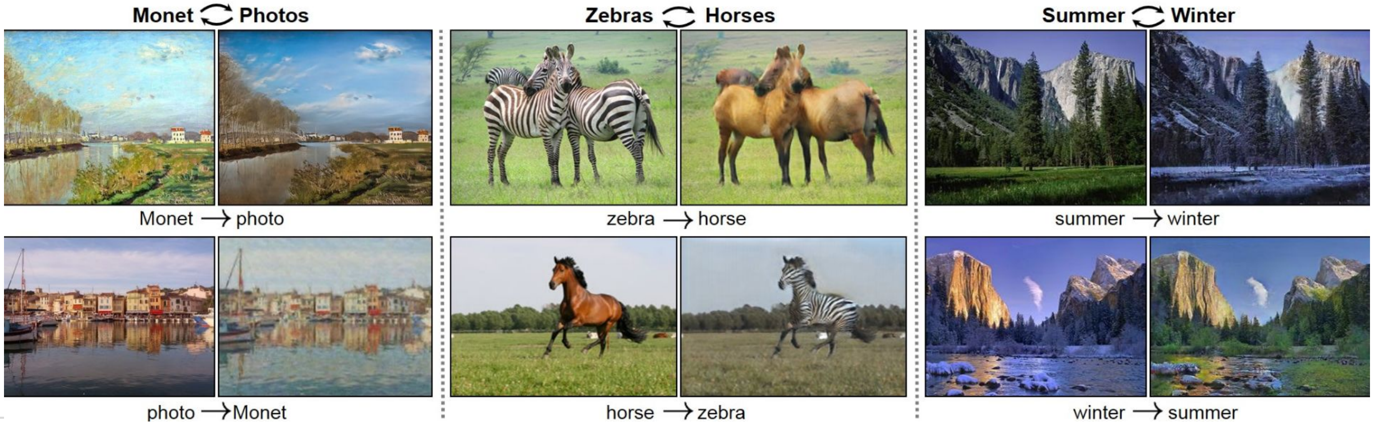
https://arxiv.org/abs/1710.08446

DeepMind

Generative adversarial networks

Pros:

- Generate compelling samples
- Enable learning from unpaired data



Generative adversarial networks

Cons:

- Instability in training
- No explicit density
- No inference

Hybrids

VAE-GAN hybrids

Why: Combine the pros for VAEs and GANs.

What: variational inference and implicit models.

https://arxiv.org/abs/1511.05644
https://arxiv.org/abs/1706.04987
https://arxiv.org/abs/1705.07761

Hybrids

VAE-GAN hybrids

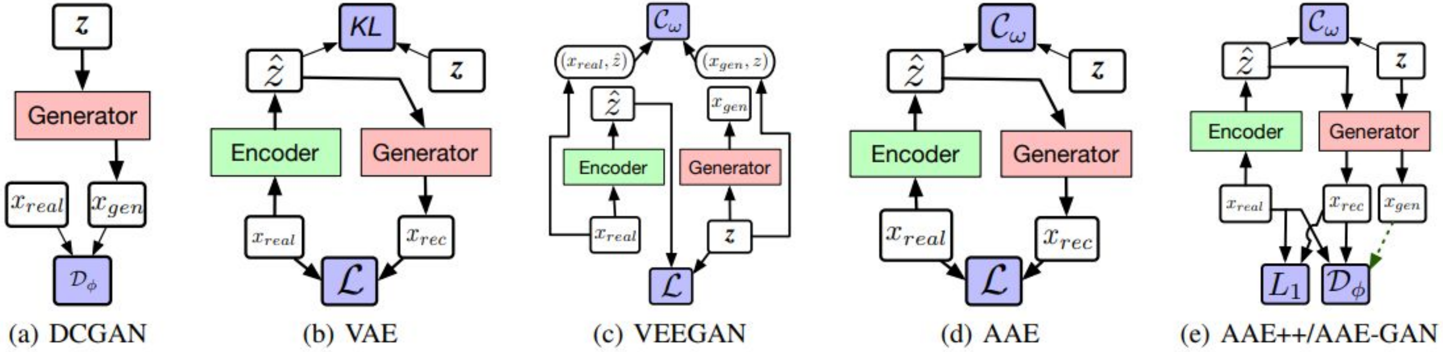
$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) = \log \int p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z} \geq \underbrace{\mathbb{E}_{q_\eta(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})]}_{\text{use a discriminator to estimate it}} - \underbrace{\mathbf{KL}[q_\eta(\mathbf{z}|\mathbf{x})||p(\mathbf{z})]}_{\text{use a discriminator to estimate this}}$$

Hybrids

VAE-GAN hybrids

We performed an extensive study and concluded:

- Use VAEs for inference
- GANs for generation

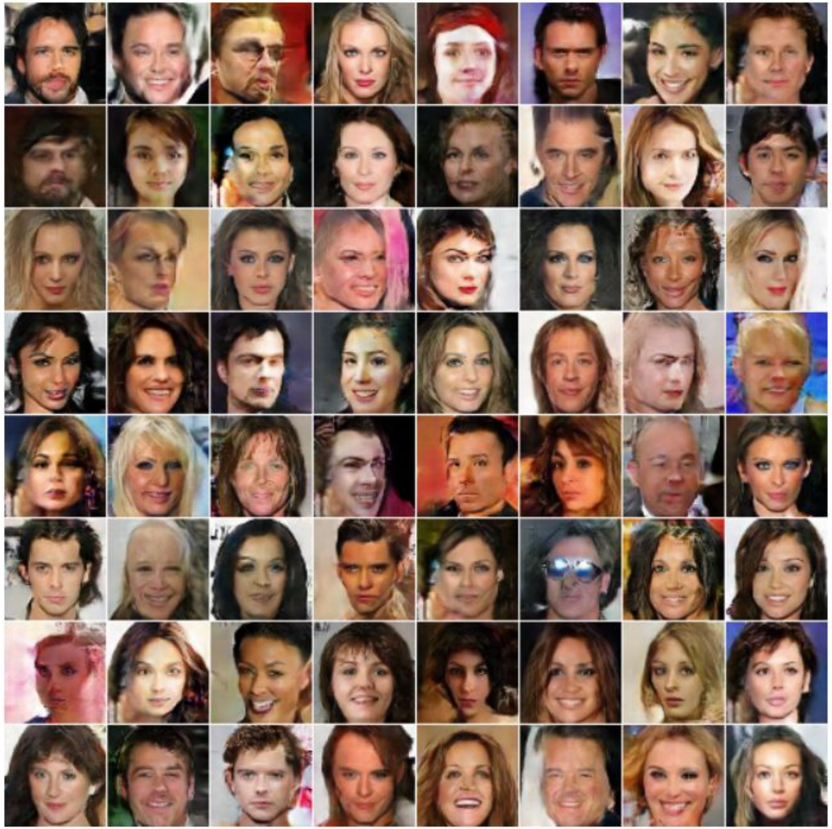


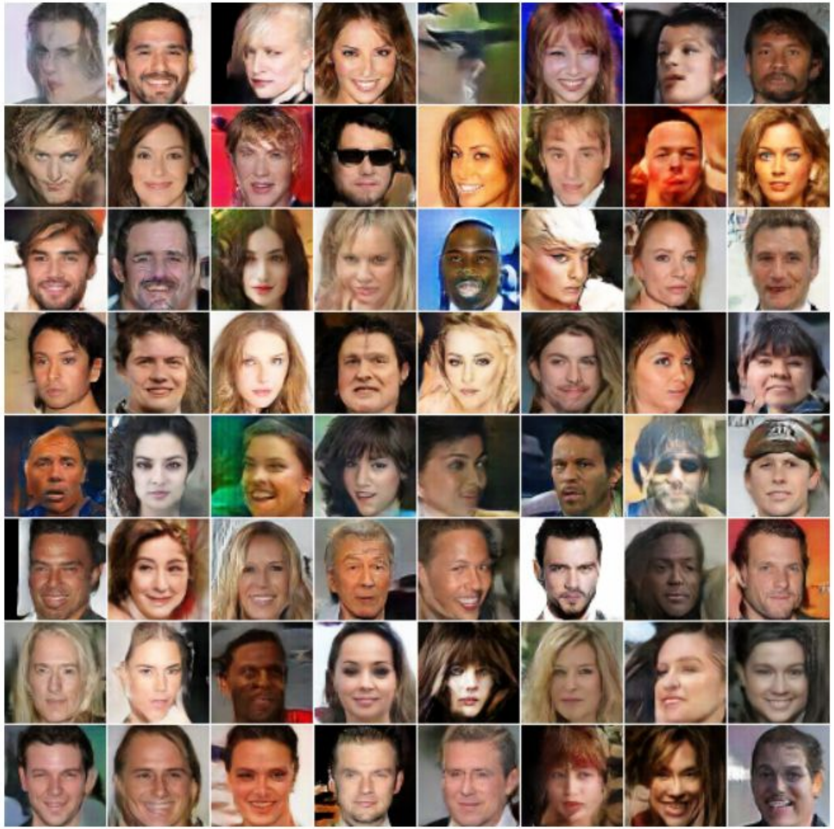
https://arxiv.org/abs/1802.06847

Evaluating generative models

Evaluation of generative models

How can we compare generative models?





Evaluation of generative models

No evaluation metric is able to capture all desired properties.

- sample quality
- generalization
- representation learning

Evaluation of generative models

Use application specific metrics

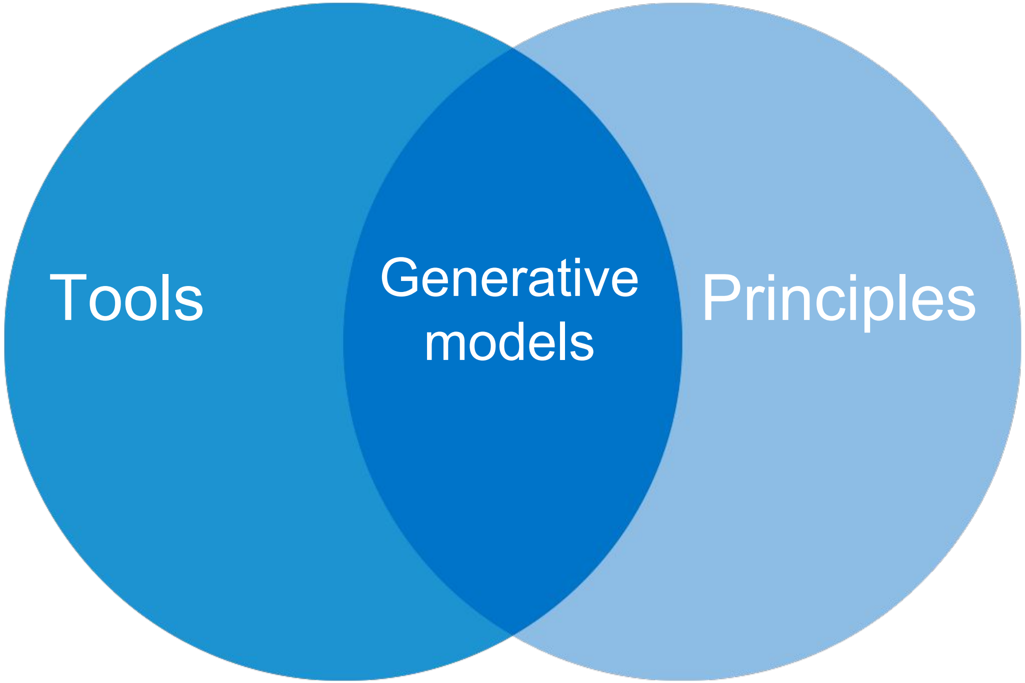
No evaluation metric is able to capture all desired properties.

Evaluate performance based on the end goal:

- semi supervised learning: classification accuracy
- reinforcement learning: total agent reward
- data generation (eg: text to speech): human (user) evaluation

I do not work on generative models

- why should I care?


Tools
Generative models
Principles

Generative models as tool

- learning with scarce labelled data
- estimating uncertainty
- eliminating outliers
- completing missing data
- generating data
- building useful (disentangled) representations

Generative models as a learning principles

Modelling probability distributions is at the core of machine learning.

Classification

Maximum likelihood

$$\mathbb{E}_{x,y \sim p^*} \; p_\theta(x, y)$$

$$= \mathbb{E}_{x,y \sim p^*} \; p_\theta(y|x)p(x)$$

$$\sim \mathbb{E}_{x,y \sim p^*} \; p_\theta(y|x)$$

Reinforcement learning

As inference

$$\log p(R) = \log \int p(R, \tau) d\tau \geq \mathbb{E}_{q(\tau)} \log p(R|\tau) + KL(q(\tau)||p(\tau))$$

$q(\tau)$ is the variational distribution and encodes the policy.

Reinforcement learning

As inference

$$\log p(R) = \log \int p(R, \tau) d\tau \geq \mathbb{E}_{q(\tau)} \log p(R|\tau) + KL(q(\tau)||p(\tau))$$

This is entropy regularized policy gradient.

Compression

Choose the model that gives the shortest description of data

Goal: Find a code with which the receiver can reconstruct the original data

http://www.helsinki.fi/~ahonkela/papers/infview.pdf

Compression

$$
\begin{aligned}
L_{q(\boldsymbol{\theta})}(\boldsymbol{X}) &= \sum_{\boldsymbol{\theta}} q(\boldsymbol{\theta}) \log \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{X}, \boldsymbol{\theta} | \mathcal{H})} \\
&= \sum_{\boldsymbol{\theta}} q(\boldsymbol{\theta}) \log \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta} | \boldsymbol{X}, \mathcal{H})} - \log p(\boldsymbol{X} | \mathcal{H}) \\
&= D\left(q(\boldsymbol{\theta}) || p(\boldsymbol{\theta} | \boldsymbol{X}, \mathcal{H})\right) - \log p(\boldsymbol{X} | \mathcal{H})
\end{aligned}
$$

http://www.helsinki.fi/~ahonkela/papers/infview.pdf

The principles behind generative models can be applied everywhere in ML.

THANK YOU

Credits

Shakir Mohamed, Balaji Lakshminarayanan

Additional Credits