Carnegie Mellon
School of Computer Science

Deep Reinforcement Learning and Control

Handling Uncertainty

CMU 10703

Katerina Fragkiadaki

Curiosity-driven exploration-one way to do it

We will add exploration reward bonuses to the extrinsics (task-related) rewards:

Independent of the task in hand!

$$R^t(s, a, s') = \underbrace{r(s, a, s')}_{extrinsic} + \underbrace{\mathcal{B}^t(s, a, s')}_{intrinsic}$$

We would then be using rewards $R^t(s, a, s')$ in our favorite model free RL method.

Exploration reward bonuses are non stationary: as the agent interacts with the environment, what is now new and novel, becomes old and known.

Computational Curiosity

- “The direct goal of curiosity and boredom is to improve the **world model**. The indirect goal is to ease the learning of new goal-directed action sequences.”
- “The same complex mechanism which is used for ‘normal’ goal-directed learning is used for implementing curiosity and boredom. There is no need for devising a separate system which aims at improving the world model.”
- “Curiosity Unit”: reward is a function of the mismatch between model’s current predictions and actuality. There is positive reinforcement whenever the system fails to correctly predict the environment.
- “Thus the usual credit assignment process ... encourages certain past actions in order to repeat situations similar to the mismatch situation.” (planning to make your (internal) world model to fail)

Jurgen Schmidhuber, 1991, 1991, 1997

Limitation of Prediction Error as Bonus

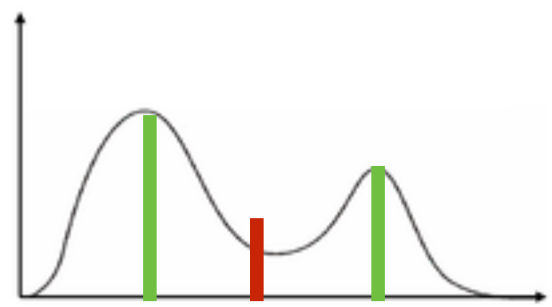
- Agent will be rewarded even though the model cannot improve.
- The agent is attracted forever in the most noisy states, with unpredictable outcomes.
- If we give the agent a TV and a remote, it becomes *a couch potato*!

Note: a standard regression net was used here for model learning

Large-scale study of Curiosity-Driven Learning, Burda et al.

How can we fix this?

- A deterministic regression network, when faced with multimodal outputs, predicts the mean…this is the least squares solution.
- This will always cause out network to have high prediction error, high surprise, high norm of the gradient, but no learning progress…



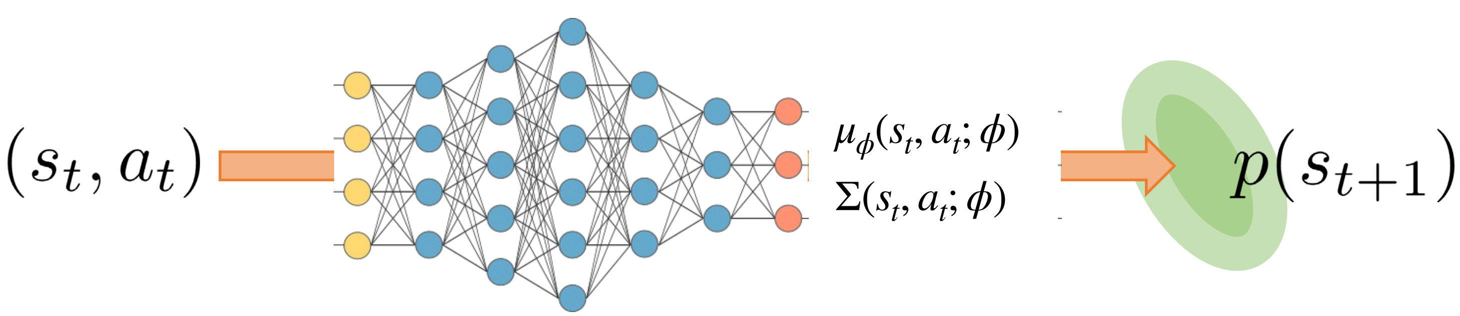
- Q1: How did we call this type of uncertainty caused by stochasticity of the environment?
- A1: Aleatoric
- Q2: How can we handle such uncertainty?
- A2: We need to be predicting a distribution, as opposed to a single solution (in principle..)

Predicting distributions

- We already know how to learn a Gaussian distribution of our output space: we parametrize the mean and the standard deviations with the network

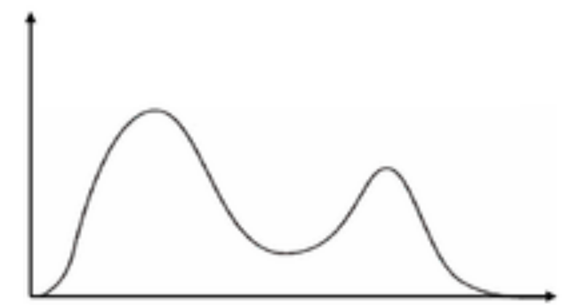
$$p_\phi(s' \mid s, a) = \frac{\exp\left(-\frac{1}{2}(s' - \mu(s,a;\phi))^\top (\boldsymbol{\Sigma}(s,a;\phi))^{-1}(s' - \mu(s,a;\phi)\right)}{\sqrt{(2\pi)^d \det \boldsymbol{\Sigma}(s,a;\phi)}}$$

$$\begin{aligned}\mathscr{L}_\phi &= -\frac{1}{N}\sum_{i=1}^{N} \log p(s_i' \mid s_i, a_i; \phi) \\ &= \left(\frac{1}{2}(s_i' - \mu(s_i, a_i; \phi))^T \boldsymbol{\Sigma}(s_i, a_i; \phi)^{-1}(s_i' - \mu(s_i, a_i; \phi))\right) \\ &+\frac{1}{2}\log(\det \boldsymbol{\Sigma}(s_i, a_i; \phi)) + \text{ const.}\end{aligned}$$



Predicting distributions

- We already know how to learn a Gaussian distribution of our output space: we parametrize the mean and the standard deviations with the network
- How can we predict more general (multimodal) distributions?

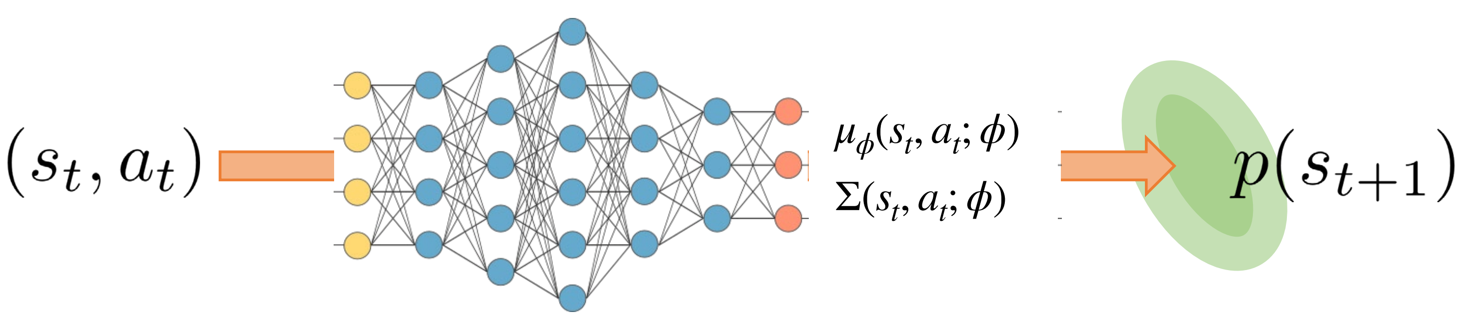


$$p_\phi(s'|s,a) = \frac{\exp\left(-\frac{1}{2}(s'-\mu(s,a;\phi)^\top(\mathbf{\Sigma}(s,a;\phi))^{-1}(s'-\mu(s,a;\phi)\right)}{\sqrt{(2\pi)^d \det \mathbf{\Sigma}(s,a;\phi)}}$$

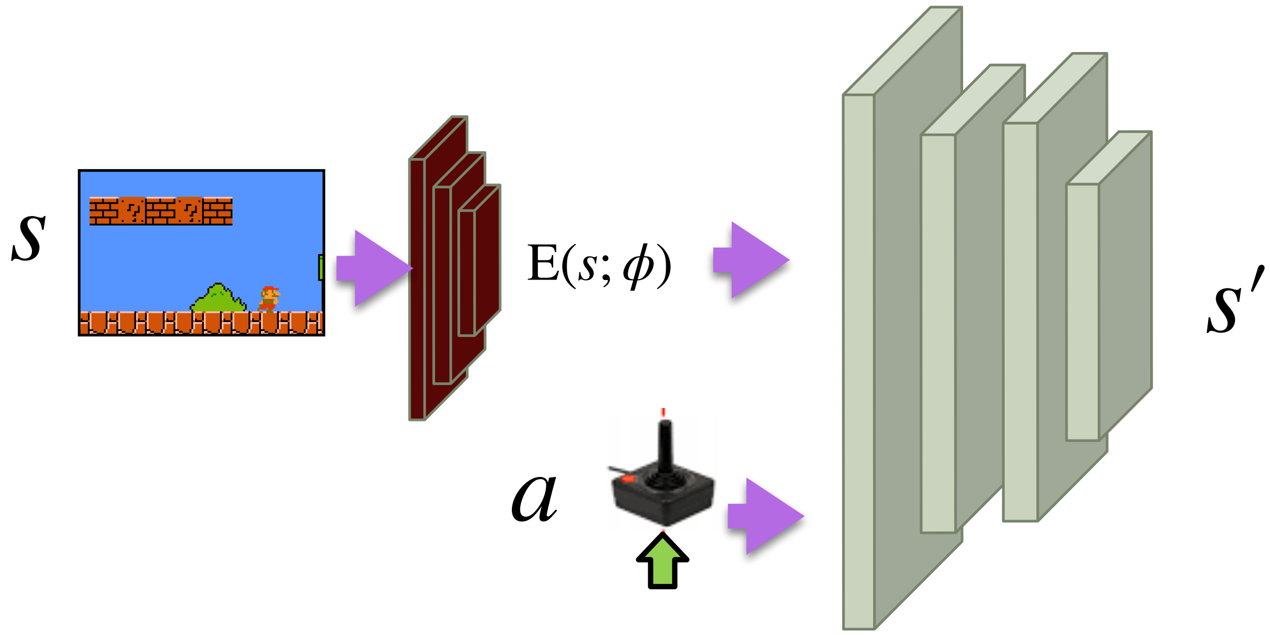
$$\mathscr{L}_\phi = -\frac{1}{N}\sum_{i=1}^{N} \log p(s_i'|s_i,a_i;\phi)$$

$$= \left(\frac{1}{2}(s_i'-\mu(s_i,a_i;\phi))^T \mathbf{\Sigma}(s_i,a_i;\phi)^{-1}(s_i'-\mu(s_i,a_i;\phi))\right)$$

$$+\frac{1}{2}\log(\det \mathbf{\Sigma}(s_i,a_i;\phi)) + \text{ const.}$$

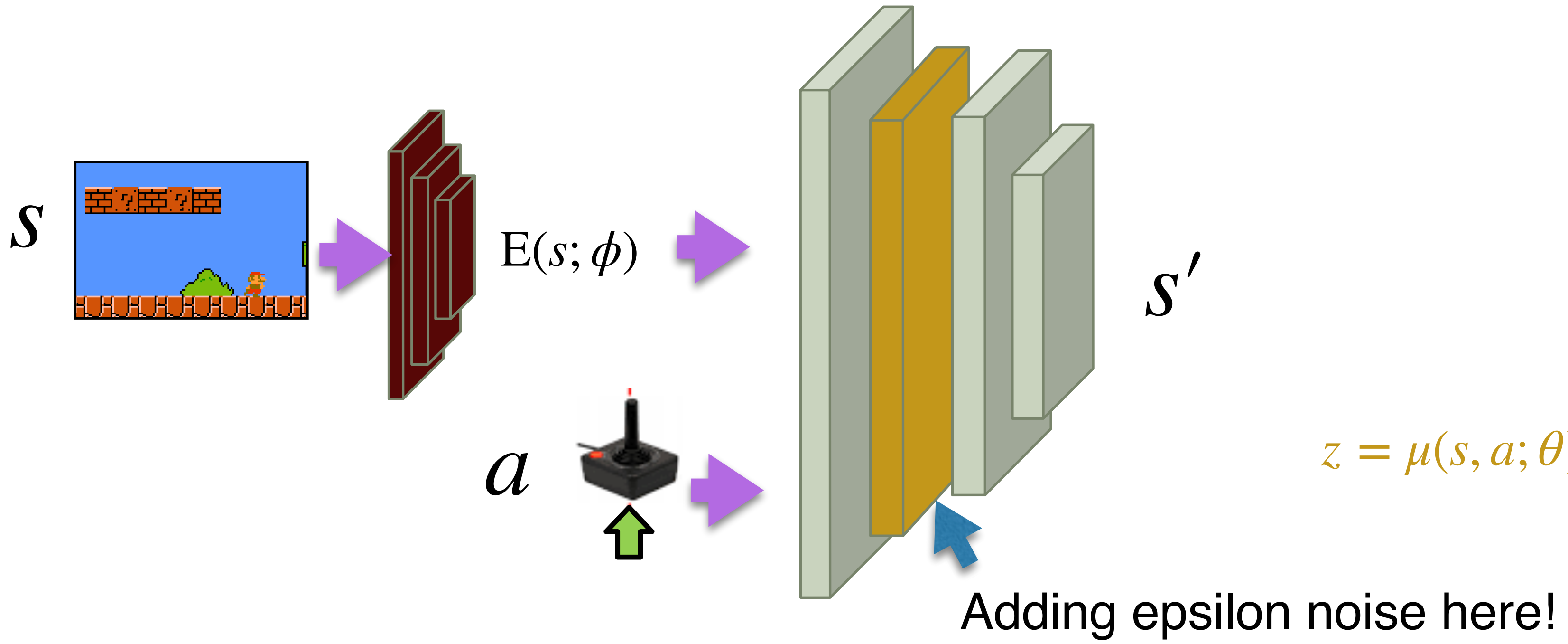


Predicting distributions



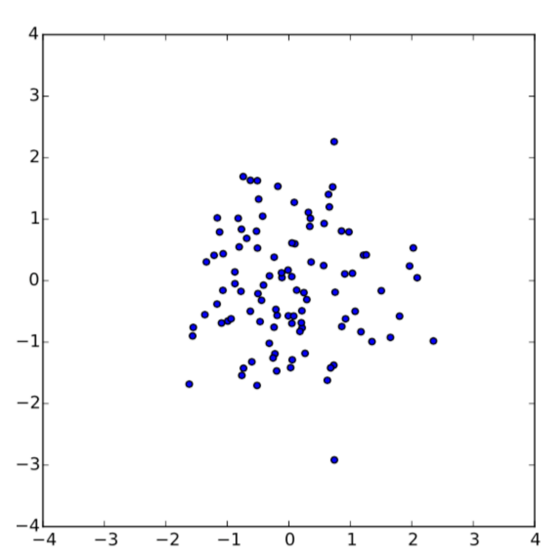
- Goal: I want to learn a complex multimodal distribution over next states s'

Predicting distributions

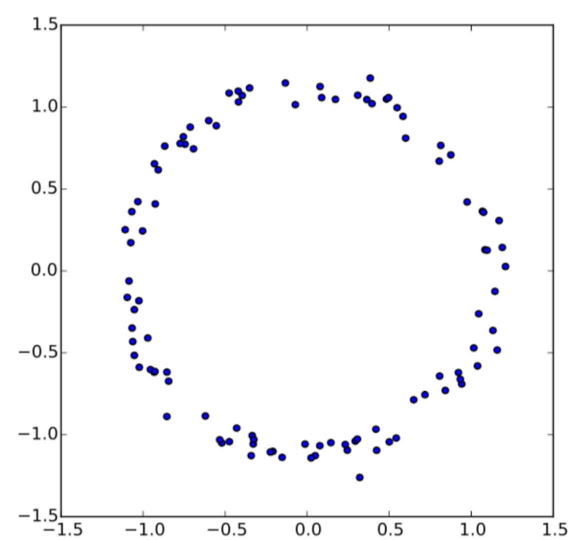


$$z = \mu(s, a; \theta) + \Sigma(\mu, a; \theta)^{1/2}\epsilon, \quad \epsilon \sim \mathcal{N}(\mathbf{0}, I)$$

- Why simple gaussian noise suffices to create complex outputs?
- The neural net will transform it to an arbitrarily complex distribution!



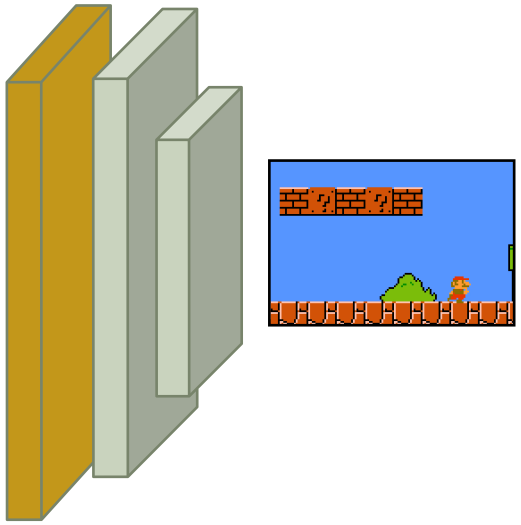
$$z \sim \mathcal{N}(\mathbf{0}, I)$$



$$f(z) = \frac{z}{10} + \frac{z}{\|z\|}$$

Tutotial on variational Autoencoders, Doersch

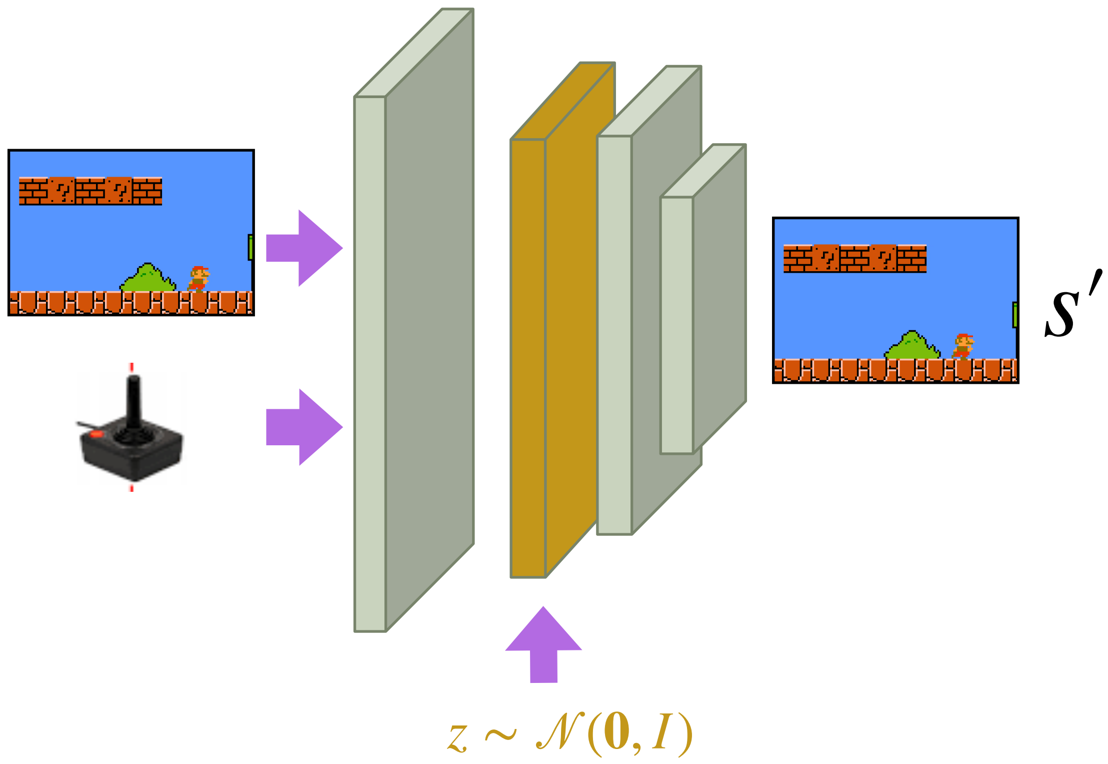
Predicting unconditional distributions



$z \sim \mathcal{N}(\mathbf{0}, I)$

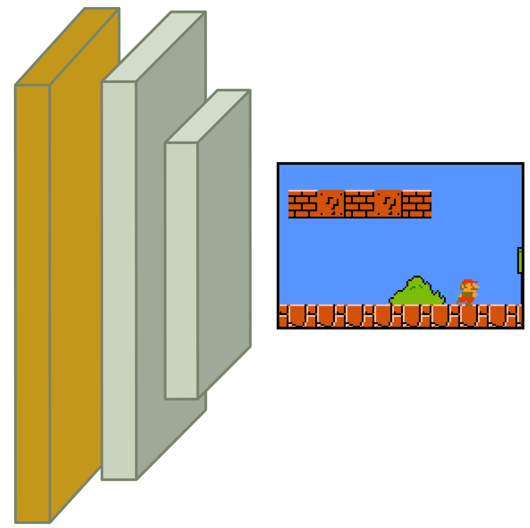
- Let's forget for now the conditioning part, and imagine we want to learn to draw samples from a Mario Bros image distribution.
- Each sample z (which I already know how to draw) should map to a realistic image once it passes through the neural network

Predicting unconditional distributions



- Let's forget for now the conditioning part, and imagine we want to learn to draw samples from a Mario Bros image distribution.
- Each sample z (which I already know how to draw) should map to a realistic image once it passes through the neural network
- We will soon return to predicting conditional distributions of such images, e.g., next state image distributions

Predicting unconditional distributions



$z \sim \mathcal{N}(\mathbf{0}, I)$

We want to learn a mapping from z to the **output image X**, usually we assume a Gaussian distribution to sample every pixel from, so f(z,\theta) predicts the mean of this:

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

Let's maximize data likelihood. This requires an intractable integral, too many zs..

$$\max_{\theta} . \quad P(X) = \int P(X \mid z; \theta) P(z) dz$$

This is called the *evidence*

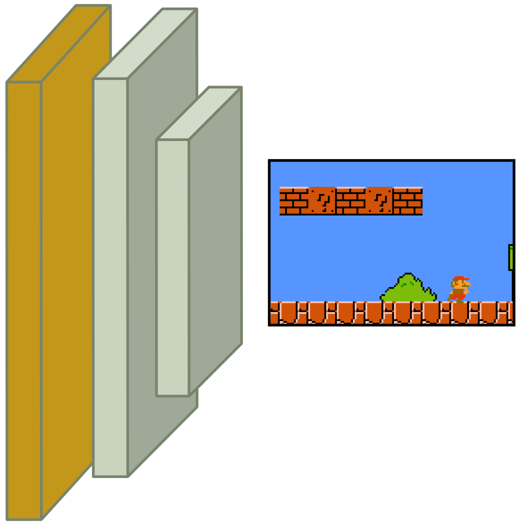
(What if we forget that it is intractable and approximate it with few samples from the prior P(z)?

$$\min_{\theta} . \quad \sum_j -\log P(X_j) = -\sum_j \sum_{z_i \sim \mathcal{N}(\mathbf{0}, I)} \log P(X_j \mid z; \theta) = -\sum_j \sum_{z_i \sim \mathcal{N}(\mathbf{0}, I)} \|f(z_i; \theta) - X_j\|^2 + \text{const.}$$

This is a bad approximation, except if we have a very large number of zs. Only few zs would produce after training reasonable X. How will we find the zs that produce good X?

Tutotial on variational Autoencoders, Doersch

Predicting unconditional distributions



$z \sim \mathcal{N}(\mathbf{0}, I)$

We want to learn a mapping from z to the **output image X**, usually we assume a Gaussian distribution to sample every pixel from, so f(z,\theta) predicts the mean of this:

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

Let's maximize data likelihood. This requires an intractable integral:

$$\max_{\theta} . \quad P(X) = \int P(X \mid z; \theta) P(z) dz$$

Tutotial on variational Autoencoders, Doersch

Variational inference

The goal of variational inference is to approximate a conditional density of latent variables given observed variables. The key idea is to solve this problem with optimization. We use a family of densities over the latent variables, parameterized by free “variational parameters.”

The optimization finds the member of this family, i.e., the setting of the parameters, that is closest in KL divergence to the conditional of interest. The fitted variational density then serves as a proxy for the exact conditional density

Deep Variational Inference

Idea behind variational inference methods:
Let's consider sampling zs from an alternative distribution Q(z) and try to minimize the KL between this (variational approximation) and the true posterior P(z|X) to find the best possible approximation. And because I can pick any distribution Q I like, I will also condition it on X (the image I am trying to geenrate) to help guide the sampling.

https://en.wikipedia.org/wiki/Variational_Bayesian_methods

Deep Variational Inference

$$D_{KL}(Q(z|X)||P(z|X)) = \int Q(z|X) \log \frac{Q(z|X)}{P(z|X)} dz$$

https://en.wikipedia.org/wiki/Variational_Bayesian_methods

Deep Variational Inference

$$D_{KL}(Q(z|X)||P(z|X)) = \int Q(z|X) \log \frac{Q(z|X)}{P(z|X)} dz$$

$$= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log P(z|X)$$

Deep Variational Inference

$$
\begin{aligned}
D_{KL}(Q(z|X)||P(z|X)) &= \int Q(z|X)\log\frac{Q(z|X)}{P(z|X)}dz \\
&= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log P(z|X) \\
&= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log \frac{P(X|z)P(z)}{P(X)}
\end{aligned}
$$

Deep Variational Inference

$$D_{KL}(Q(z|X)||P(z|X)) = \int Q(z|X) \log \frac{Q(z|X)}{P(z|X)} dz$$

$$= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log P(z|X)$$

$$= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log \frac{P(X|z)P(z)}{P(X)}$$

$$= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log \frac{P(X|z)P(z)}{P(X)}$$

Deep Variational Inference

$$
\begin{aligned}
D_{KL}(Q(z|X)||P(z|X)) &= \int Q(z|X)\log\frac{Q(z|X)}{P(z|X)}dz \\
&= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log P(z|X) \\
&= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log \frac{P(X|z)P(z)}{P(X)} \\
&= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log \frac{P(X|z)P(z)}{P(X)} \\
&= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log P(X|z) - \mathbb{E}_Q \log P(z) + \mathbb{E}_Q \log P(X)
\end{aligned}
$$

Deep Variational Inference

$$
\begin{aligned}
D_{KL}(Q(z|X)||P(z|X)) &= \int Q(z|X)\log\frac{Q(z|X)}{P(z|X)}dz \\
&= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log P(z|X) \\
&= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log \frac{P(X|z)P(z)}{P(X)} \\
&= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log \frac{P(X|z)P(z)}{P(X)} \\
&= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log P(X|z) - \mathbb{E}_Q \log P(z) + \mathbb{E}_Q \log P(X) \\
&= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log P(X|z) - \mathbb{E}_Q \log P(z) + \log P(X)
\end{aligned}
$$

Deep Variational Inference

$$
\begin{aligned}
D_{KL}(Q(z|X)||P(z|X)) &= \int Q(z|X)\log\frac{Q(z|X)}{P(z|X)}dz \\
&= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log P(z|X) \\
&= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log \frac{P(X|z)P(z)}{P(X)} \\
&= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log \frac{P(X|z)P(z)}{P(X)} \\
&= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log P(X|z) - \mathbb{E}_Q \log P(z) + \mathbb{E}_Q \log P(X) \\
&= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log P(X|z) - \mathbb{E}_Q \log P(z) + \log P(X) \\
&= D_{KL}(Q(z|X)|P(z)) - \mathbb{E}_Q \log P(X|z) + \log P(X)
\end{aligned}
$$

Deep Variational Inference

$$
\begin{aligned}
D_{KL}(Q(z|X)||P(z|X)) &= \int Q(z|X)\log\frac{Q(z|X)}{P(z|X)}dz \\
&= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log P(z|X) \\
&= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log \frac{P(X|z)P(z)}{P(X)} \\
&= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log \frac{P(X|z)P(z)}{P(X)} \\
&= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log P(X|z) - \mathbb{E}_Q \log P(z) + \mathbb{E}_Q \log P(X) \\
&= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log P(X|z) - \mathbb{E}_Q \log P(z) + \log P(X) \\
&= D_{KL}(Q(z|X)|P(z)) - \mathbb{E}_Q \log P(X|z) + \log P(X)
\end{aligned}
$$

Rearranging: $\log P(X) - D_{KL}(Q(z|X)||P(z|X)) = \mathbb{E}_Q \log P(X|z) - D_{KL}(Q(z|X)|P(z))$

Deep Variational Inference

$$
\begin{aligned}
D_{KL}(Q(z|X)||P(z|X)) &= \int Q(z|X)\log\frac{Q(z|X)}{P(z|X)}dz \\
&= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log P(z|X) \\
&= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log \frac{P(X|z)P(z)}{P(X)} \\
&= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log \frac{P(X|z)P(z)}{P(X)} \\
&= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log P(X|z) - \mathbb{E}_Q \log P(z) + \mathbb{E}_Q \log P(X) \\
&= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log P(X|z) - \mathbb{E}_Q \log P(z) + \log P(X) \\
&= D_{KL}(Q(z|X)|P(z)) - \mathbb{E}_Q \log P(X|z) + \log P(X)
\end{aligned}
$$

Rearranging:

$$
\log P(X) - D_{KL}(Q(z|X)||P(z|X)) = \mathbb{E}_Q \log P(X|z) - D_{KL}(Q(z|X)|P(z))
$$

What we are trying to maximize

Deep Variational Inference

$$
\begin{aligned}
D_{KL}(Q(z|X)||P(z|X)) &= \int Q(z|X)\log\frac{Q(z|X)}{P(z|X)}dz \\
&= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log P(z|X) \\
&= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log \frac{P(X|z)P(z)}{P(X)} \\
&= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log \frac{P(X|z)P(z)}{P(X)} \\
&= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log P(X|z) - \mathbb{E}_Q \log P(z) + \mathbb{E}_Q \log P(X) \\
&= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log P(X|z) - \mathbb{E}_Q \log P(z) + \log P(X) \\
&= D_{KL}(Q(z|X)|P(z)) - \mathbb{E}_Q \log P(X|z) + \log P(X)
\end{aligned}
$$

Rearranging:

$$
\log P(X) - D_{KL}(Q(z|X)||P(z|X)) = \mathbb{E}_Q \log P(X|z) - D_{KL}(Q(z|X)|P(z))
$$

What we are trying to maximize

A positive quantity

Deep Variational Inference

$$
\begin{aligned}
D_{KL}(Q(z|X)||P(z|X)) &= \int Q(z|X)\log\frac{Q(z|X)}{P(z|X)}dz \\
&= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log P(z|X) \\
&= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log \frac{P(X|z)P(z)}{P(X)} \\
&= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log \frac{P(X|z)P(z)}{P(X)} \\
&= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log P(X|z) - \mathbb{E}_Q \log P(z) + \mathbb{E}_Q \log P(X) \\
&= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log P(X|z) - \mathbb{E}_Q \log P(z) + \log P(X) \\
&= D_{KL}(Q(z|X)|P(z)) - \mathbb{E}_Q \log P(X|z) + \log P(X)
\end{aligned}
$$

Rearranging:

$$
\log P(X) \geq \mathbb{E}_Q \log P(X|z) - D_{KL}(Q(z|X)|P(z))
$$

What we are trying to maximize A lower bound of the evidence (ELBO)

Deep Variational Inference

$$
\begin{aligned}
D_{KL}(Q(z|X)||P(z|X)) &= \int Q(z|X) \log \frac{Q(z|X)}{P(z|X)} dz \\
&= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log P(z|X) \\
&= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log \frac{P(X|z)P(z)}{P(X)} \\
&= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log \frac{P(X|z)P(z)}{P(X)} \\
&= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log P(X|z) - \mathbb{E}_Q \log P(z) + \mathbb{E}_Q \log P(X) \\
&= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log P(X|z) - \mathbb{E}_Q \log P(z) + \log P(X) \\
&= D_{KL}(Q(z|X)|P(z)) - \mathbb{E}_Q \log P(X|z) + \log P(X)
\end{aligned}
$$

Rearranging:

$$\log P(X) \geq \mathbb{E}_Q \log P(X|z) - D_{KL}(Q(z|X)|P(z))$$

We will maximize the ELBO instead to estimates parameters of both P(XIz) and Q(zIX)

Deep Variational Inference

$$D_{KL}(Q(z|X)||P(z|X)) = \int Q(z|X)\log\frac{Q(z|X)}{P(z|X)}dz$$

$$= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log P(z|X)$$

$$= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log \frac{P(X|z)P(z)}{P(X)}$$

$$= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log \frac{P(X|z)P(z)}{P(X)}$$

$$= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log P(X|z) - \mathbb{E}_Q \log P(z) + \mathbb{E}_Q \log P(X)$$

$$= \mathbb{E}_Q \log Q(z|X) - \mathbb{E}_Q \log P(X|z) - \mathbb{E}_Q \log P(z) + \log P(X)$$

$$= D_{KL}(Q(z|X)|P(z)) - \mathbb{E}_Q \log P(X|z) + \log P(X)$$

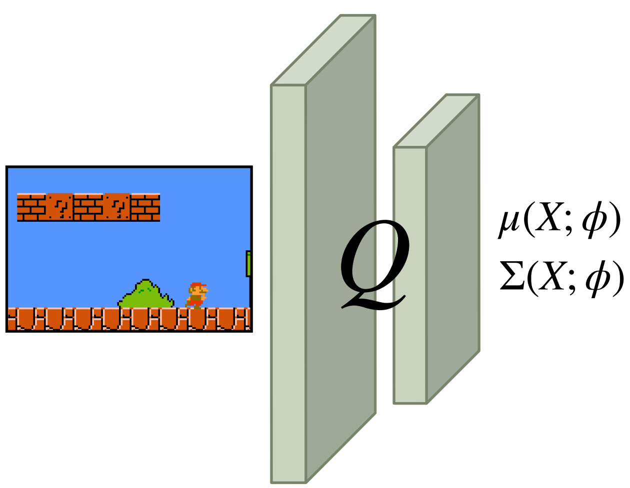
$$\min_{\phi,\theta}. \quad D_{KL}(Q(z|X;\phi)||P(z)) - \mathbb{E}_Q \log P(X|z;\theta)$$

Deep Variational Inference

$$\min_{\phi,\theta} . \quad D_{KL}(Q(z\,|\,X;\phi)\,||\,P(z)) - \mathbb{E}_Q \log P(X\,|\,z;\theta)$$

What will be our $Q(z\,|\,X;\phi)$?

Usually a Gaussian distribution:

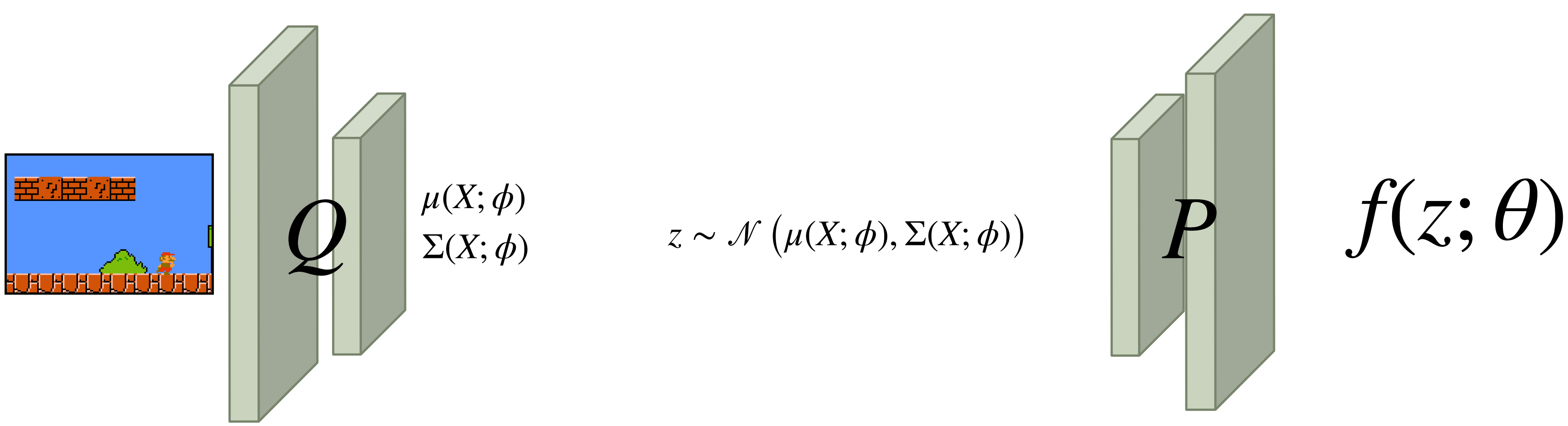


$$z \sim \mathcal{N}\left(\mu(X;\phi), \Sigma(X;\phi)\right)$$

https://en.wikipedia.org/wiki/Variational_Bayesian_methods

Deep Variational Inference

$$\min_{\phi,\theta} . \quad D_{KL}(Q(z\,|\,X;\phi)\,||\,P(z)) - \mathbb{E}_Q \log P(X\,|\,z;\theta)$$



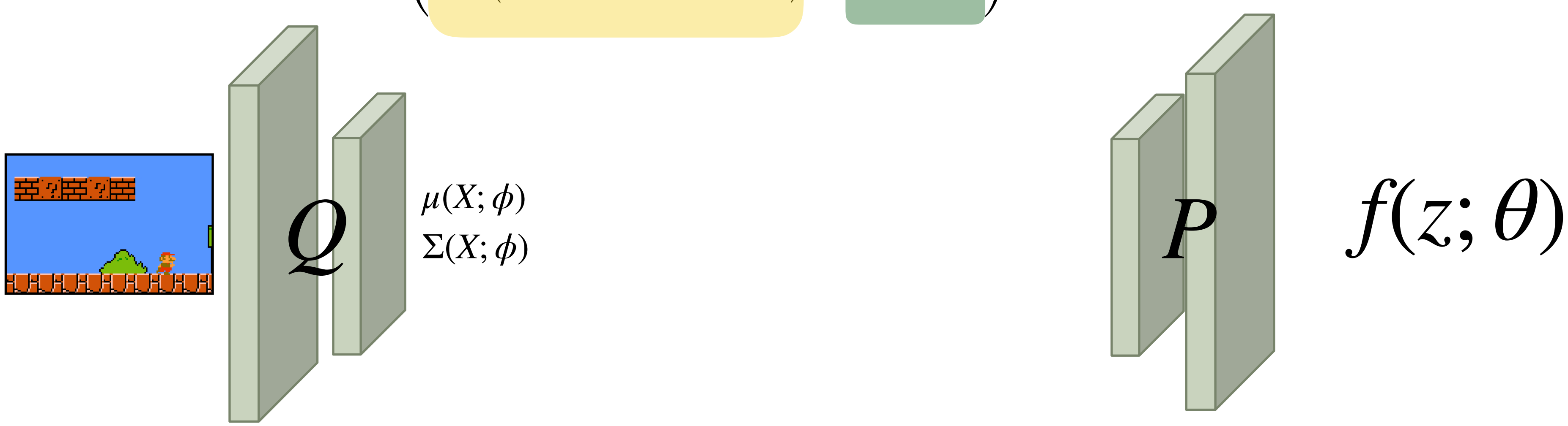
https://en.wikipedia.org/wiki/Variational_Bayesian_methods

Deep Variational Inference

$$\min_{\phi,\theta} . \quad D_{KL}(Q(z \mid X;\phi) \,||\, P(z)) - \mathbb{E}_Q \log P(X \mid z;\theta)$$

Adding the losses!

$$D_{KL}\left(\mathcal{N}\left(\mu(X;\phi), \Sigma(X;\phi)\right) \,||\, \mathcal{N}(0,I)\right)$$



KL between two Gaussian distributions:

$$D_{KL}\left(\mathcal{N}\left(\mu_0, \Sigma_0\right) \,||\, \mathcal{N}(\mu_1, \Sigma_1)\right) = \tfrac{1}{2}\left(\mathrm{tr}\left(\Sigma_1^{-1}\Sigma_0\right) + (\mu_1 - \mu_0)^{\top}\Sigma_1^{-1}(\mu_1 - \mu_0) - k + \log\left(\frac{\det \Sigma_1}{\det \Sigma_0}\right)\right)$$

KL for our special case:

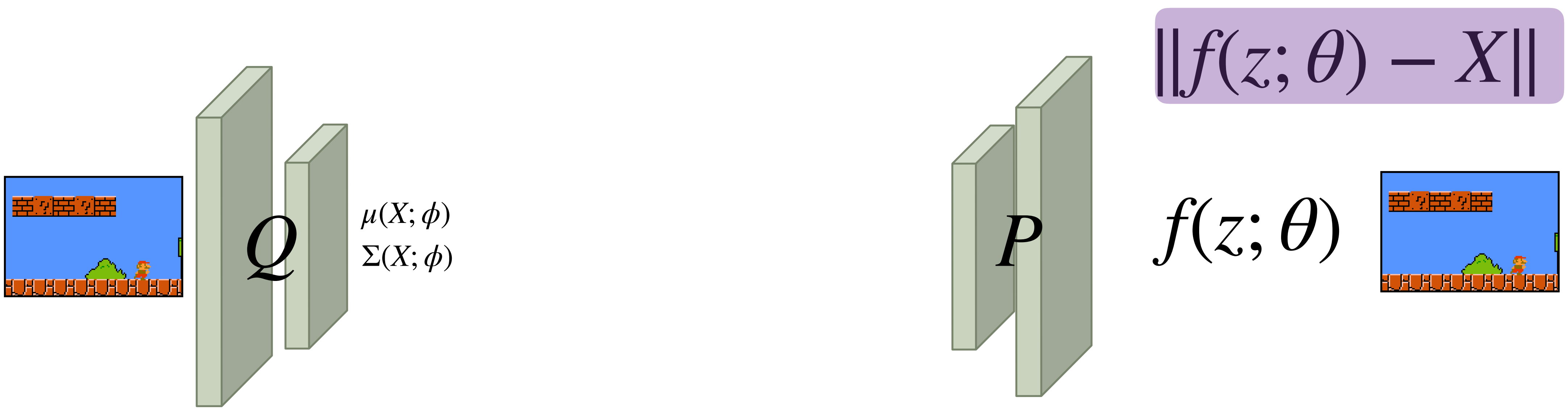
$$D_{KL}\left(\mathcal{N}\left(\mu(X;\phi), \Sigma(X;\phi)\right) \,||\, \mathcal{N}(0,I)\right) = \tfrac{1}{2}\left(\mathrm{tr}\left(\Sigma(X)\right) + (\mu(X))^{\top}(\mu(X)) - k - \log\det\left(\Sigma(X)\right)\right)$$

https://en.wikipedia.org/wiki/Variational_Bayesian_methods

Deep Variational Inference

$$\min_{\phi,\theta} . \quad D_{KL}(Q(z\,|\,X;\phi)\,||\,P(z)) - \mathbb{E}_Q \log P(X\,|\,z;\theta)$$

Adding the losses!



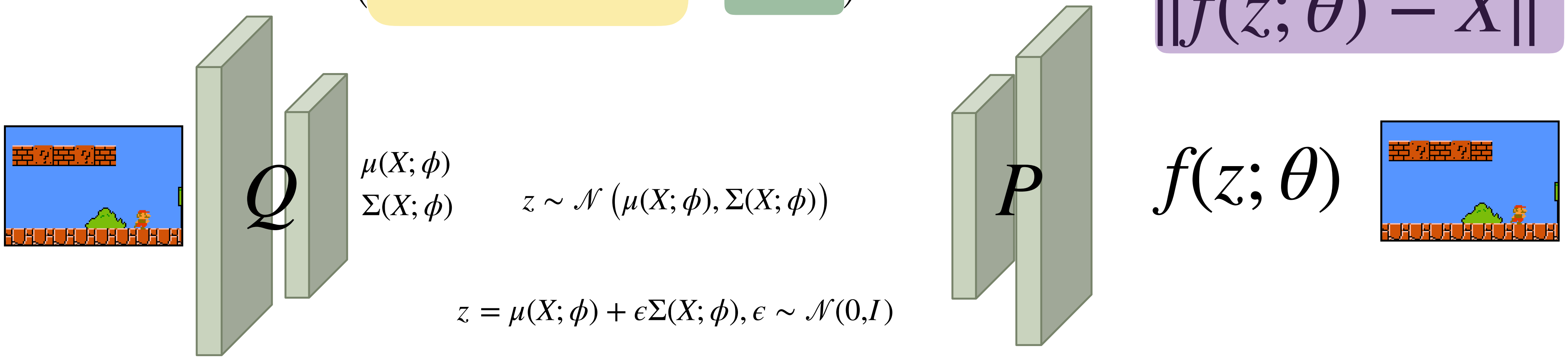
https://en.wikipedia.org/wiki/Variational_Bayesian_methods

Deep Variational Inference

$$\min_{\phi,\theta} . \quad D_{KL}(Q(z\,|\,X;\phi)\,||\,P(z)) - \mathbb{E}_Q \log P(X\,|\,z;\theta)$$

Adding the losses!

$$D_{KL}\left(\mathcal{N}\left(\mu(X;\phi), \Sigma(X;\phi)\right)\,||\,\mathcal{N}(0,I)\right) \qquad \|f(z;\theta) - X\|$$



How do we sample z so that we can take gradients through the sampling?

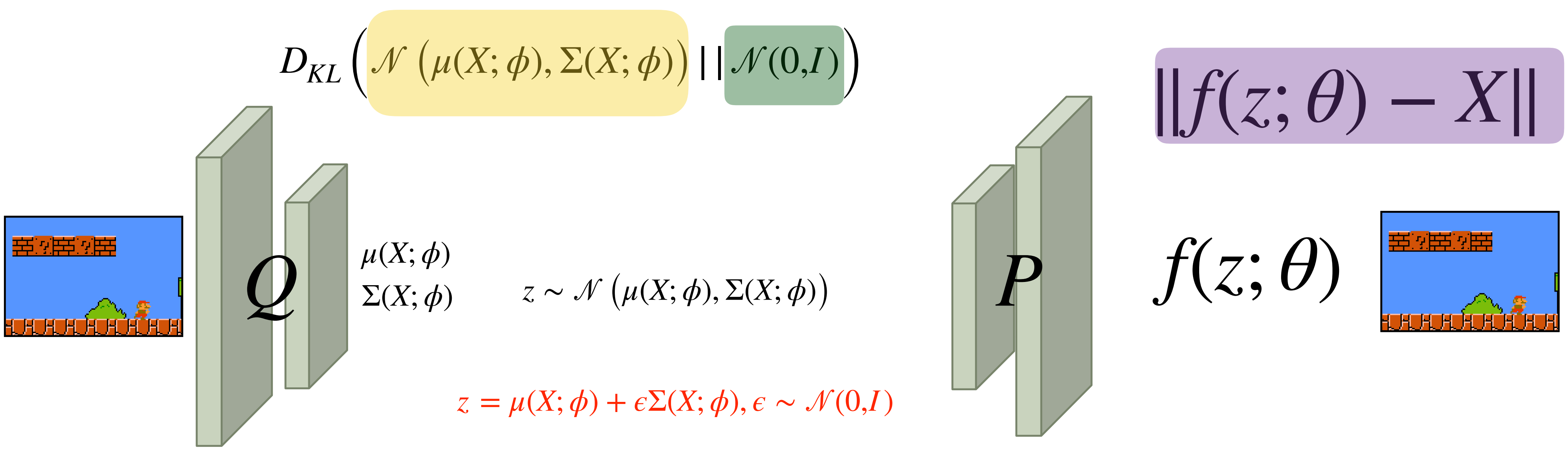
$$\min_{\phi,\theta} . -\mathbb{E}_{z\sim Q(X,\phi)} \log P(X\,|\,z;\theta)$$

https://en.wikipedia.org/wiki/Variational_Bayesian_methods

Deep Variational Inference

$$\min_{\phi,\theta} . \quad D_{KL}(Q(z \mid X; \phi) \,||\, P(z)) - \mathbb{E}_Q \log P(X \mid z; \theta)$$

Adding the losses!



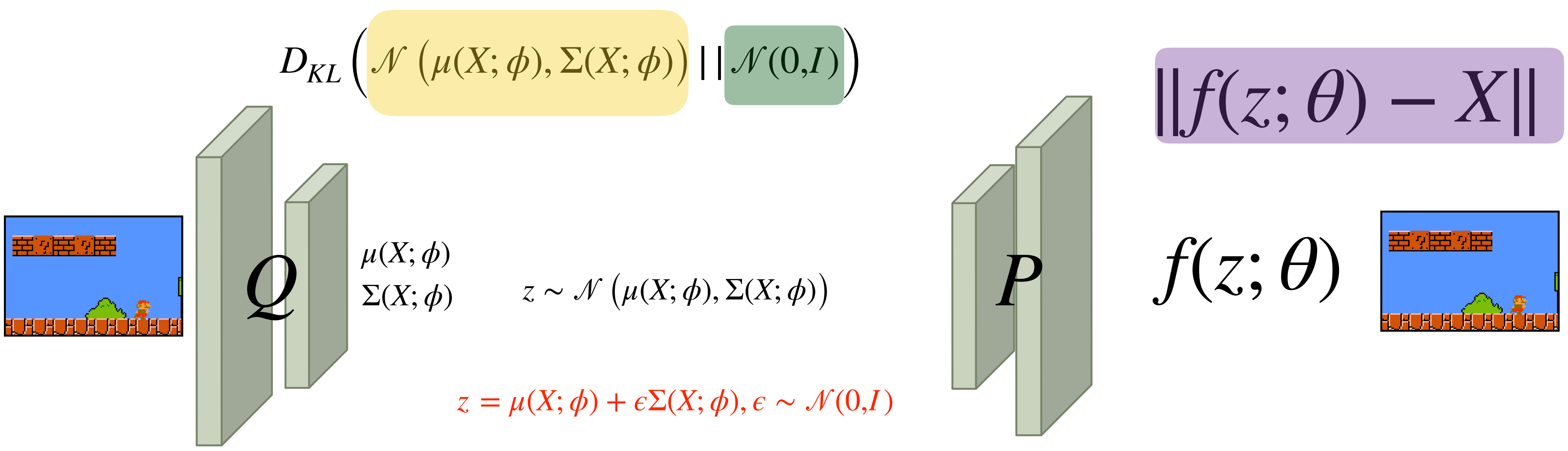
How do we sample z so that we can take gradients through the sampling? Reparametrization trick!

$$\min_{\phi,\theta} . - \mathbb{E}_{\epsilon \sim \mathcal{N}(0,I)} \log P(X \mid \mu(X;\phi) + \epsilon\Sigma(X;\phi); \theta)$$

https://en.wikipedia.org/wiki/Variational_Bayesian_methods

Deep Variational Inference-Training time

$$\min_{\phi,\theta} . \quad D_{KL}(Q(z \,|\, X; \phi) \,||\, P(z)) - \mathbb{E}_Q \log P(X \,|\, z; \theta)$$



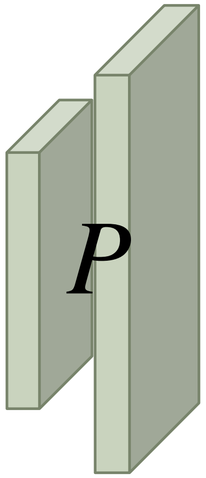
It looks like an autoencoder! That is why it is called variational autoencoder.

https://en.wikipedia.org/wiki/Variational_Bayesian_methods

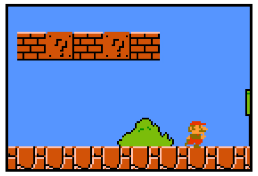
Deep Variational Inference-Test time

We only use the decoder!

$z \sim \mathcal{N}(0, I)$



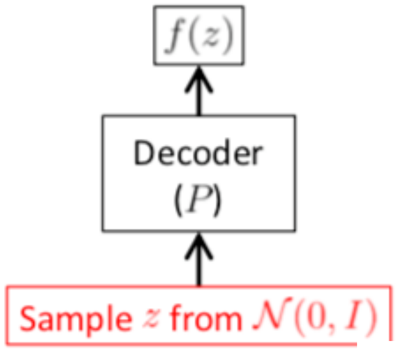
$f(z; \theta)$



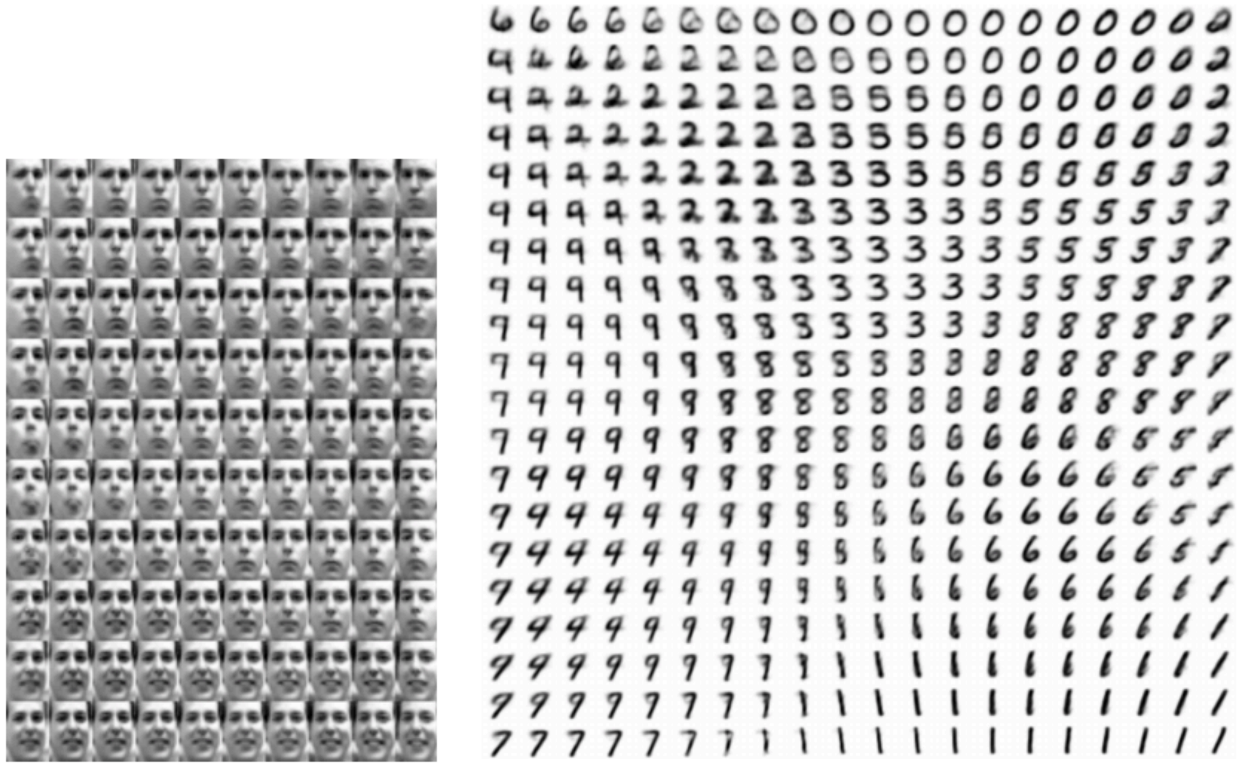
https://en.wikipedia.org/wiki/Variational_Bayesian_methods

Variational Autoencoder

At test time



Let's add back conditioning of this sampling!



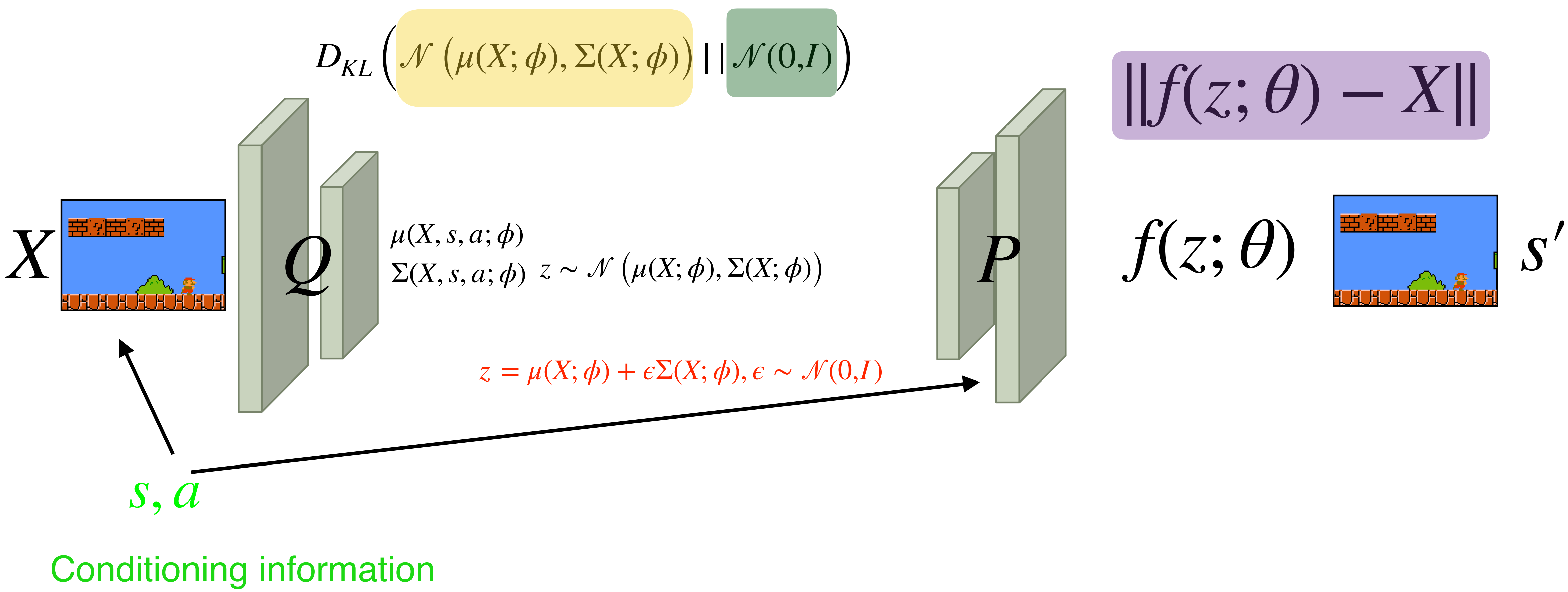
(a) Learned Frey Face manifold

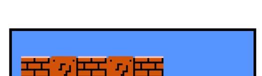
(b) Learned MNIST manifold

Auto-Encoding Variational Bayes, Kingma and Welling

Deep Variational Inference-Training time

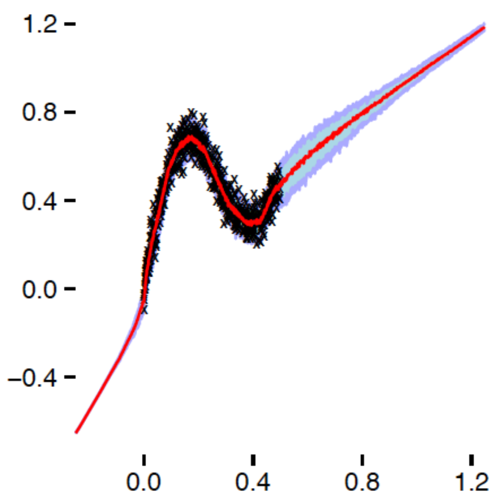
$$\min_{\phi,\theta} . \quad D_{KL}(Q(z|X;\phi)||P(z)) - \mathbb{E}_Q \log P(X|z;\theta)$$



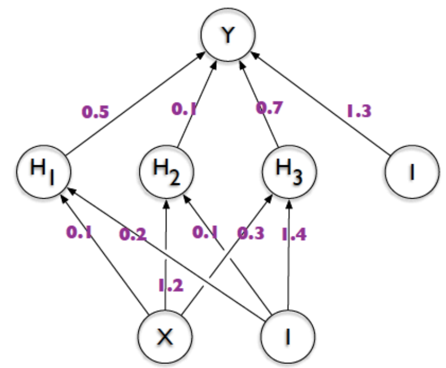


https://en.wikipedia.org/wiki/Variational_Bayesian_methods

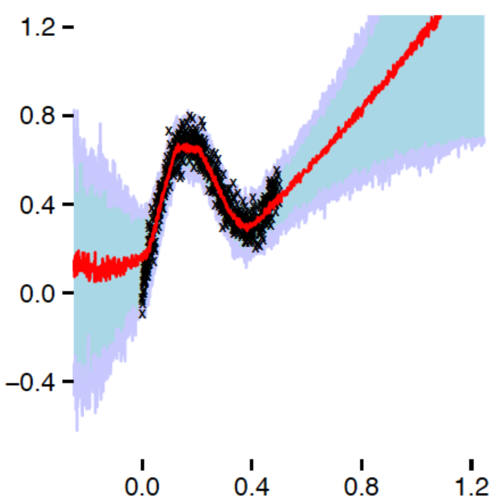
Bayesian Deep Networks



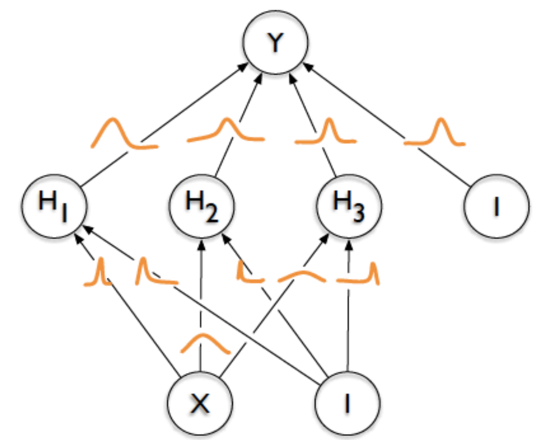
regression network



$$\mathbf{w}^{\mathrm{MAP}} = \arg\max_{\mathbf{w}} \log P(\mathbf{w}|\mathcal{D}) = \arg\max_{\mathbf{w}} \log P(\mathcal{D}|\mathbf{w}) + \log P(\mathbf{w}).$$



bayesian regression network



$P(\mathbf{w} \mid \mathscr{D})$

Bayesian Nets for representing epistemic uncertainty. Again an intractable integral:

$$P(X) = \int P(X \mid w; \theta) P(w) dw$$

Variational Inference for Bayesian Neural Networks

Variational approximation to the Bayesian posterior distribution of the weights.

We will consider Q to be a diagonal gaussian distribution: $\phi = (\mu, \sigma)$

We will consider prior P(\theta) to be a mixture of 0 mean gaussians:

$$P(\theta) = \prod_k \pi \mathcal{N}(\theta_k \mid 0, \sigma_1^2) + (1 - \pi)\mathcal{N}(\theta_k \mid 0, \sigma^2), \quad \pi, \sigma_1, \sigma_2 \text{ are chosen and fixed}$$

Weight Uncertainty in Neural Networks, Blundell et al.

Variational Inference for Bayesian Neural Networks

Variational approximation to the Bayesian posterior distribution of the weights;

$$D_{KL}(Q(\theta|\phi)||P(\theta|\mathscr{D})) = \int Q(\theta|\phi)\log\frac{Q(\theta|\phi)}{P(\theta|\mathscr{D})}d\theta$$

$$= \mathbb{E}_Q \log Q(\theta|\phi) - \mathbb{E}_Q \log P(\theta|\mathscr{D})$$

$$= \mathbb{E}_Q \log Q(\theta|\phi) - \mathbb{E}_Q \log \frac{P(\mathscr{D}|\theta)P(\theta)}{P(\mathscr{D})}$$

$$= \mathbb{E}_Q \log Q(\theta|\phi) - \mathbb{E}_Q \log \frac{P(\mathscr{D}|\theta)P(\theta)}{P(\mathscr{D})}$$

$$= \mathbb{E}_Q \log Q(\theta|\phi) - \mathbb{E}_Q \log P(\mathscr{D}|\theta) - \mathbb{E}_Q \log P(\theta) + \log P(\mathscr{D})$$

$$= D_{KL}(Q(\theta|\phi)|P(\theta)) - \mathbb{E}_Q \log P(\mathscr{D}|\theta) + \log P(\mathscr{D})$$

$$\min_{\phi}.\ D_{KL}(Q(\theta|\phi)||P(\theta)) - \mathbb{E}_Q \log P(\mathscr{D}|\theta)$$

weight complexity — data likelihood

Weight Uncertainty in Neural Networks, Blundell et al.

Variational Inference for Bayesian Neural Networks

Variational approximation to the Bayesian posterior distribution of the weights.

$$\min_{\phi}.\quad D_{KL}(Q(\theta;\phi)\,||\,P(\theta|\mathcal{D})) = \min_{\phi}.\quad D_{KL}(Q(\theta|\phi)\,||\,P(\theta)) - \mathbb{E}_Q \log P(\mathcal{D}|\theta)$$

weight complexity — data likelihood

We will consider Q to be a diagonal gaussian distribution: $\phi = (\mu, \sigma)$

We will consider prior P(\theta) to be a mixture of 0 mean gaussians:

$$P(\theta) = \prod_k \pi \mathcal{N}(\theta_k|0,\sigma_1^2) + (1-\pi)\mathcal{N}(\theta_k|0,\sigma^2),\quad \pi, \sigma_1, \sigma_2 \text{ are chosen and fixed}$$

Let's try to take gradients:

$$\nabla_\phi \left(D_{KL}(Q(\theta|\phi)\,||\,P(\theta)) - \mathbb{E}_Q \log P(\mathcal{D}|\theta) \right) = \nabla_\phi \left(\mathbb{E}_{Q(\theta|\phi)} \log \frac{Q(\theta|\phi)}{P(\theta)} - \mathbb{E}_{Q(\theta|\phi)} \log P(\mathcal{D}|\theta) \right)$$

$$= \nabla_\phi \mathbb{E}_{Q(\theta|\phi)} \left(\log \frac{Q(\theta|\phi)}{P(\theta)} - \log P(\mathcal{D}|\theta) \right)$$

The parameter is in the distribution! Reparametrization to the rescue:

$$\theta = t(\phi, \epsilon) = \mu\epsilon + \sigma,\quad \epsilon \sim \mathcal{N}(0, I)$$

Weight Uncertainty in Neural Networks, Blundell et al.

Variational Inference for Bayesian Neural Networks

We will consider Q to be a diagonal gaussian distribution: $\phi = (\mu, \sigma)$

We will consider prior P(\theta) to be a mixture of 0 mean gaussians:

$$P(\theta) = \prod_k \pi \mathcal{N}(\theta_k | 0, \sigma_1^2) + (1-\pi)\mathcal{N}(\theta_k | 0, \sigma^2), \quad \pi, \sigma_1, \sigma_2 \text{ are chosen and fixed}$$

Let's try to take gradients:

$$\nabla_\phi \Big(D_{KL}(Q(\theta|\phi) || P(\theta)) - \mathbb{E}_Q \log P(\mathcal{D}|\theta) \Big) = \nabla_\phi \left(\mathbb{E}_{Q(\theta|\phi)} \log \frac{Q(\theta|\phi)}{P(\theta)} - \mathbb{E}_{Q(\theta|\phi)} \log P(\mathcal{D}|\theta) \right)$$

$$= \nabla_\phi \mathbb{E}_{Q(\theta|\phi)} \left(\log \frac{Q(\theta|\phi)}{P(\theta)} - \log P(\mathcal{D}|\theta) \right)$$

The parameter is in the distribution! Reparametrization to the rescue:

$$\theta = t(\phi, \epsilon) = \mu\epsilon + \sigma, \quad \epsilon \sim \mathcal{N}(0, I)$$

1. Sample \epsilon
2. Form \theta
3. Take gradients w.r.t. \phi
4. Update \phi

Weight Uncertainty in Neural Networks, Blundell et al.

Variational Inference for Bayesian Neural Networks

We will consider Q to be a diagonal gaussian distribution: $\phi = (\mu, \sigma)$

We will consider prior P(\theta) to be a mixture of 0 mean gaussians:

$$P(\theta) = \prod_k \pi \mathcal{N}(\theta_k | 0, \sigma_1^2) + (1 - \pi)\mathcal{N}(\theta_k | 0, \sigma^2), \quad \pi, \sigma_1, \sigma_2 \text{ are chosen and fixed}$$

Let's try to take gradients:

$$\nabla_\phi \Big(D_{KL}(Q(\theta | \phi) || P(\theta)) - \mathbb{E}_Q \log P(\mathcal{D} | \theta) \Big) = \nabla_\phi \left(\mathbb{E}_{Q(\theta|\phi)} \log \frac{Q(\theta | \phi)}{P(\theta)} - \mathbb{E}_{Q(\theta|\phi)} \log P(\mathcal{D} | \theta) \right)$$

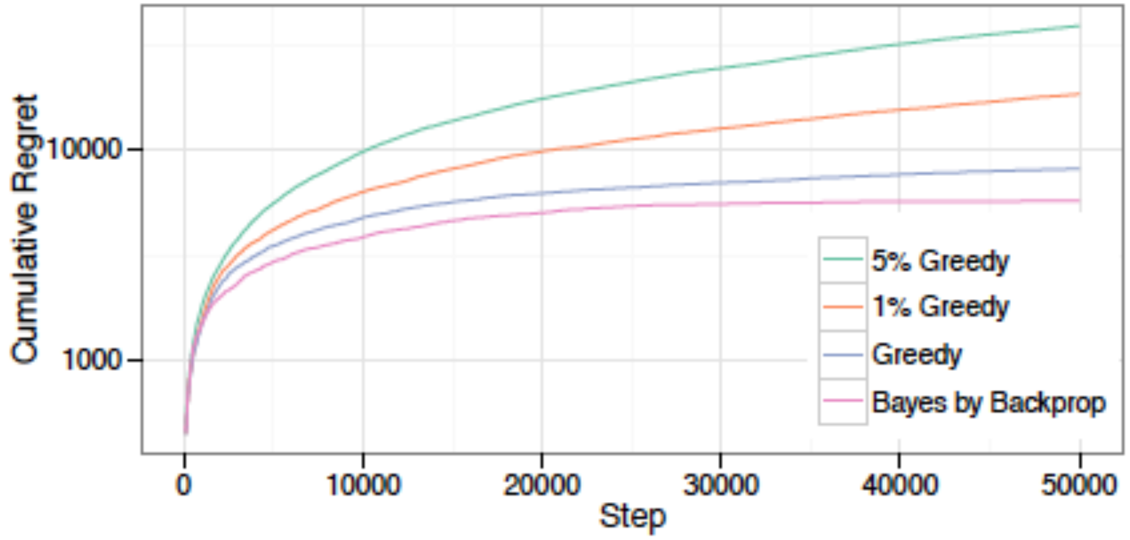
$$= \nabla_\phi \mathbb{E}_{Q(\theta|\phi)} \left(\log \frac{Q(\theta | \phi)}{P(\theta)} - \log P(\mathcal{D} | \theta) \right)$$

The parameter is in the distribution! Reparametrization to the rescue:

$$\theta = t(\phi, \epsilon) = \mu\epsilon + \sigma, \quad \epsilon \sim \mathcal{N}(0, I)$$

1. Sample \epsilon
2. Form \theta
3. Take gradients w.r.t. \phi
4. Update \phi

They used it for Thompson sampling!



Weight Uncertainty in Neural Networks, Blundell et al.