Machine Learning Methods for Neural Data Analysis

Variational Autoencoders (VAEs)

Scott Linderman

STATS 220/320 (NBIO220, CS339N)

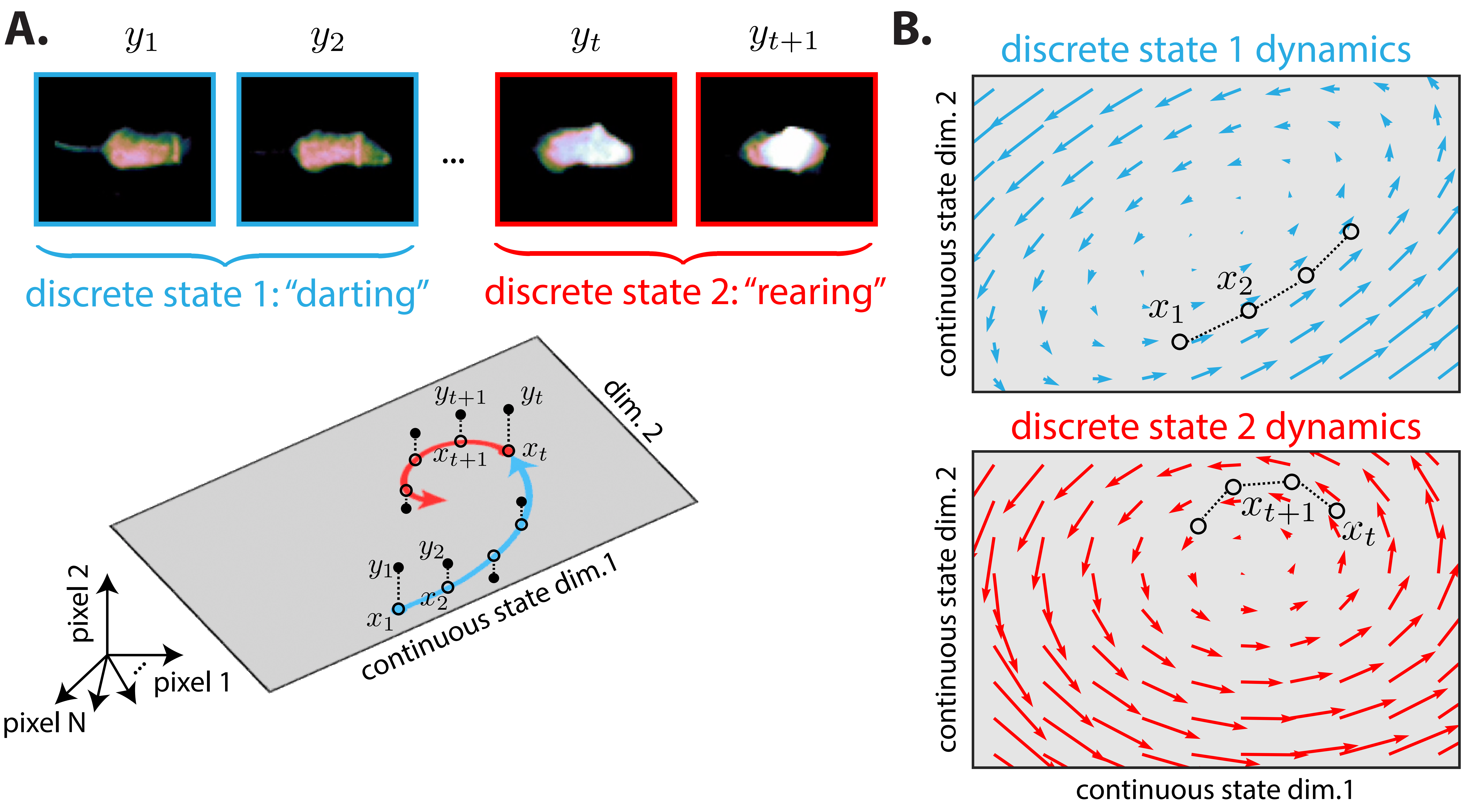
Outline

- Revisiting Factor Analysis
- SGD on the ELBO
- Generalizing to nonlinear factor models

Factor Analysis

A component of SLDS

- Recall Lab 8: Mixture of Factor Analyzers and SLDS.
- The model assumed data live near a low dimensional manifold (a plane).



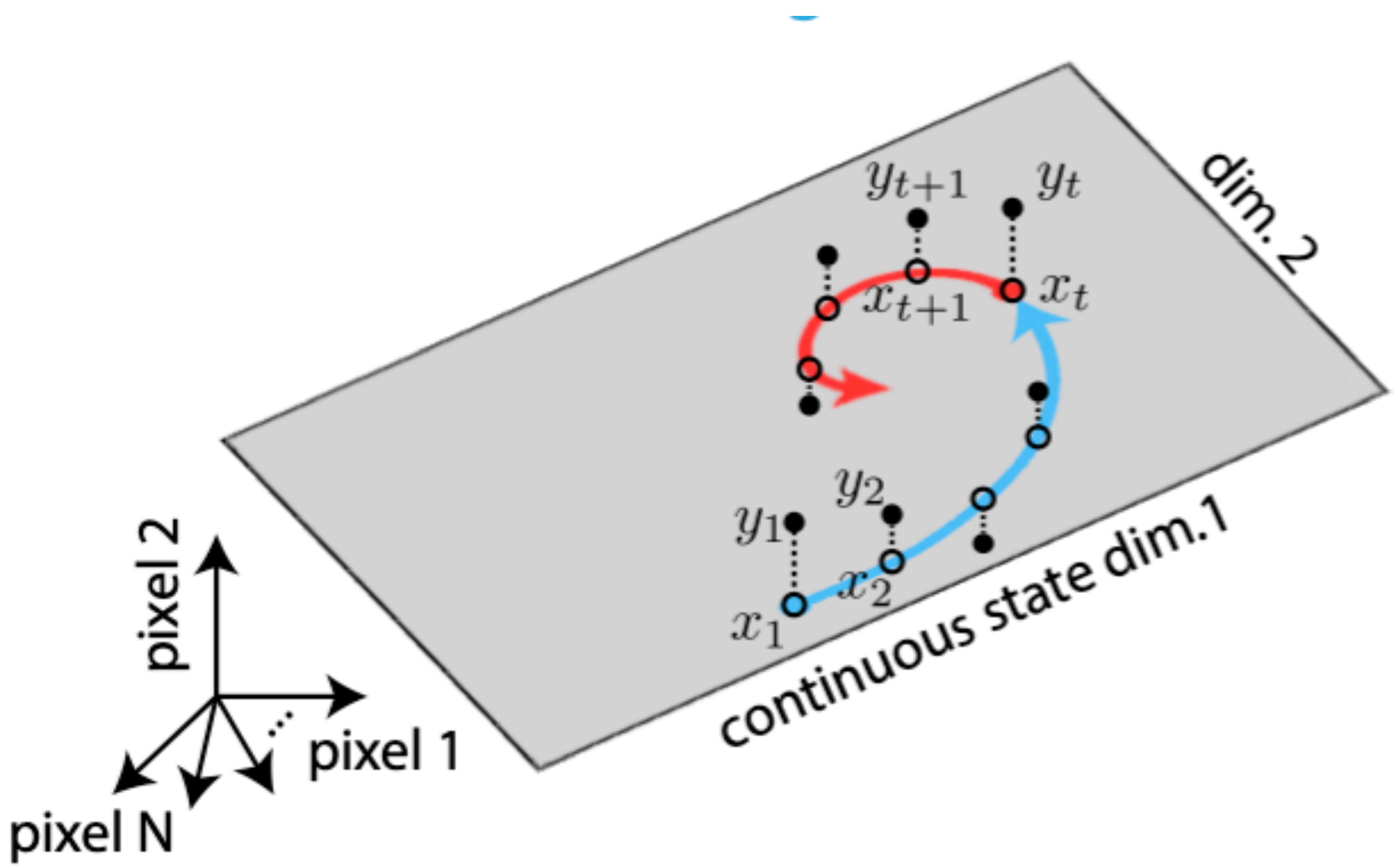
Factor Analysis

Generative Model

- The generative model for factor analysis is

$$x_t \sim \mathcal{N}(0, I)$$
$$y_t \sim \mathcal{N}(Cx_t + d, R)$$

where $x_t \in \mathbb{R}^D$ are the **continuous latent states** and $y_t \in \mathbb{R}^N$ are the **observations.**



- Contrast this with discrete mixture models.

Factor Analysis

EM Algorithm

E step: Solve for the posterior, $q(x_t) = p(x_t \mid y_t; \theta)$

M step: $\theta^\star = \arg\max \mathbb{E}_q[\log p(x, y; \theta)]$ has a closed form solution too.

Factor Analysis

Stochastic M-step

We can approximate the ELBO with Monte Carlo,

$$\begin{aligned}\mathscr{L}(q, \theta) &= \mathbb{E}_{q(x_t)}[\log p(x_t, y_t; \theta) - \log q(x_t)] \\ &\approx \frac{1}{M} \sum_{m=1}^{M} [\log p(x_t^{(m)}, y_t; \theta) - \log q(x_t^{(m)})] \qquad x_t^{(m)} \overset{\text{iid}}{\sim} q(x_t)\end{aligned}$$

Factor Analysis

Stochastic M-step

We can also approximate the **gradient of the ELBO** with Monte Carlo,

$$\nabla_\theta \mathscr{L}(q, \theta) = \nabla_\theta \mathbb{E}_{q(x_t)}[\log p(x_t, y_t; \theta) - \log q(x_t)]$$

$$\approx \frac{1}{M} \sum_{m=1}^{M} [\nabla_\theta \log p(x_t^{(m)}, y_t; \theta)] \qquad x_t^{(m)} \overset{\text{iid}}{\sim} q(x_t)$$

Often, we just take one sample! I.e., set $M = 1$.

Factor Analysis

Revisiting the E-step

- The posterior mean in $q(x_t) = \mathcal{N}(x_t; \mu_t, \Sigma_t)$ is a **linear function** of y_t.

Factor Analysis

Amortized inference

Rather than solving for the posterior exactly for each data point, let's treat $\phi = (W, b, \Sigma)$ as shared **variational parameters** and learn them by stochastic gradient ascent.

$$q(x_t \mid y_t; \phi) = \mathcal{N}(x_t \mid Wy_t + b, \Sigma)$$

Then,

$$\mathcal{L}(\theta, \phi) = \mathbb{E}_{q(x_t \mid y_t; \phi)}\left[\log p(x_t, y_t; \theta) - \log q(x_t \mid y_t; \phi)\right]$$

Factor Analysis

Reparameterization trick

We can **reparameterize** x_t as a function of y_t, ϕ, and **independent noise**.

$$x_t \sim \mathcal{N}(Wy_t + b, \Sigma) \iff \begin{aligned} x_t &= Wy_t + b + \Sigma^{\frac{1}{2}}\epsilon_t; \quad \epsilon_t \sim \mathcal{N}(0, I) \\ &= x_t(y_t, \epsilon_t; \phi) \end{aligned}$$

Then,

$$\mathcal{L}(\theta, \phi) = \mathbb{E}_{\epsilon_t}\left[\log p(x_t(y_t, \epsilon_t; \phi), y_t; \theta) - \log q(x_t(y_t, \epsilon_t; \phi) \mid y_t; \phi)\right]$$

Factor Analysis

Reparameterization gradients

After reparameterizing, we can use Monte Carlo to approximate the ELBO and its gradient with respect to ϕ,

$$\begin{aligned}\mathscr{L}(\theta, \phi) &= \mathbb{E}_{\epsilon_t}\left[\log p(x_t(y_t, \epsilon_t; \phi), y_t; \theta) - \log q(x_t(y_t, \epsilon_t; \phi) \mid y_t; \phi)\right] \\ &\approx \log p(\hat{x}_t, y_t; \theta) - \log q(\hat{x}_t \mid y_t; \phi)\end{aligned}$$

where $\hat{x}_t = x_t(y_t, \epsilon_t; \phi)$ and $\epsilon_t \sim \mathcal{N}(0, I)$.

Likewise,

$$\nabla_\phi \mathscr{L}(\theta, \phi) \approx \nabla_\phi \left(\log p(\hat{x}_t, y_t; \theta) - \log q(\hat{x}_t \mid y_t; \phi)\right)$$

(Don't forget that $\hat{x}_t$ is a function of ϕ!

Factor Analysis

Stochastic Gradient Ascent on the ELBO

Instead of coordinate ascent of the ELBO (CAVI), we can just do stochastic gradient as cent of the ELBO,

while not converged:

Sample index t uniformly at random

Sample $\epsilon_t \sim \mathcal{N}(0, I)$ and compute $\hat{x}_t = x_t(y_t, \epsilon_t, \phi)$.

Evaluate $\hat{\mathscr{L}}(\theta, \phi) = \log p(\hat{x}_t, y_t; \theta) - \log q(\hat{x}_t \mid y_t; \phi)$

Update parameters $\theta \leftarrow \theta + \alpha \nabla_\theta \hat{\mathscr{L}}(\theta, \phi)$, $\phi \leftarrow \phi + \alpha \nabla_\phi \hat{\mathscr{L}}(\theta, \phi)$ and decay step size.

Factor Analysis

ELBO Surgery

We can rearrange the ELBO in many ways,

$$\mathscr{L}(\theta, \phi) = \mathbb{E}_{q(x_t)}\left[\log p(x_t, y_t; \theta) - \log q(x_t)\right]$$
$$= \underbrace{\mathbb{E}_{q(x_t)}\left[\log p(y_t \mid x_t; \theta)\right]}_{\text{expected log likelihood}} - \underbrace{\mathrm{KL}\left(q(x_t) \,\|\, p(x_t; \theta)\right)}_{\text{KL to prior}}$$

Applying the reparameterization trick,

$$\mathscr{L}(\theta, \phi) \approx \mathbb{E}_{\epsilon_t}\left[\log p(y_t \mid \hat{x}_t; \theta)\right] - \mathrm{KL}\left(q(x_t \mid y_t; \phi) \,\|\, p(x_t; \theta)\right)$$

Factor analysis

As a linear autoencoder

Now let's substitute the factor analysis model. Assume $R = \sigma^2 I$ for simplicity.

Then the objective is,

$$\begin{aligned}
\mathscr{L}(\theta, \phi) &= \mathbb{E}_{\epsilon_t}\left[\log p(y_t \mid \hat{x}_t; \theta)\right] - \mathrm{KL}\left(q(x_t \mid y_t; \phi) \,\|\, p(x_t; \theta)\right) \\
&= \mathbb{E}_{\epsilon_t}\left[\log \mathcal{N}(y_t \mid C\hat{x}_t + d, \sigma^2 I)\right] - \mathrm{KL}\left(q(x_t \mid y_t; \phi) \,\|\, p(x_t; \theta)\right) \\
&= \underbrace{-\frac{1}{2\sigma^2}\|y_t - \hat{y}_t\|_2^2}_{\text{reconstruction loss}} - \mathrm{KL}\left(q(x_t \mid y_t; \phi) \,\|\, p(x_t; \theta)\right) + c
\end{aligned}$$

Factor analysis

As a linear autoencoder

Now let's substitute the factor analysis model. Assume $R = \sigma^2 I$ for simplicity.

Then the objective is,

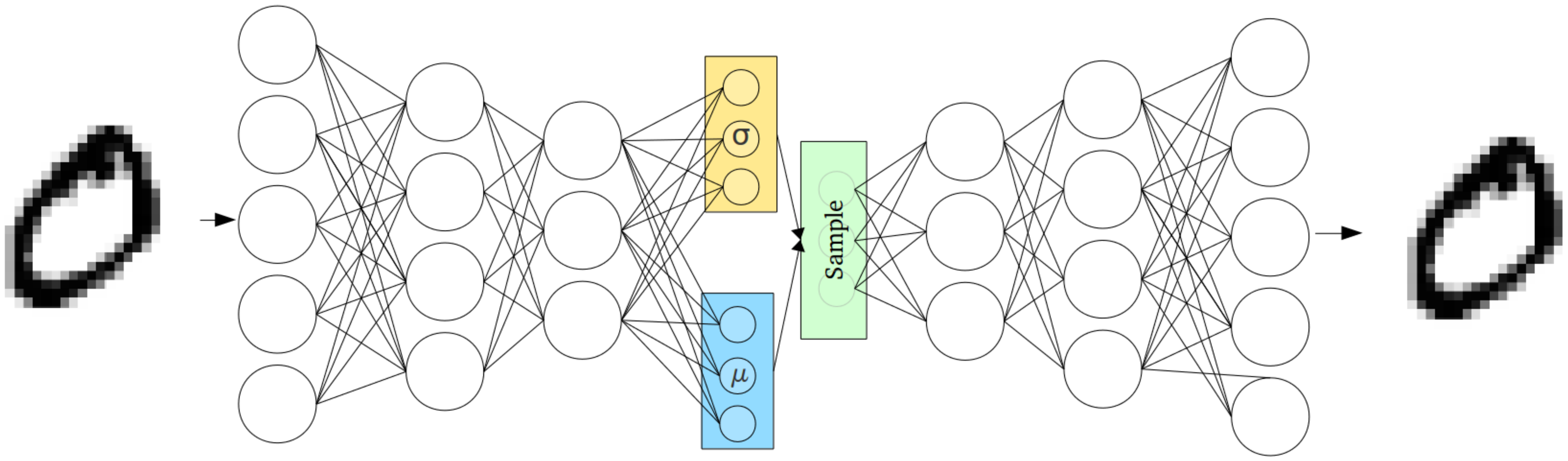
$$\begin{aligned}
\mathscr{L}(\theta, \phi) &= \mathbb{E}_{\epsilon_t}\left[\log p(y_t \mid \hat{x}_t; \theta)\right] - \mathrm{KL}\left(q(x_t \mid y_t; \phi) \,\|\, p(x_t; \theta)\right) \\
&= \mathbb{E}_{\epsilon_t}\left[\log \mathcal{N}(y_t \mid C\hat{x}_t + d, \sigma^2 I)\right] - \mathrm{KL}\left(q(x_t \mid y_t; \phi) \,\|\, p(x_t; \theta)\right) \\
&= \underbrace{-\frac{1}{2\sigma^2}\|y_t - \hat{y}_t\|_2^2}_{\text{reconstruction loss}} - \mathrm{KL}\left(q(x_t \mid y_t; \phi) \,\|\, p(x_t; \theta)\right) + c
\end{aligned}$$

Factor Analysis

In pictures

Variational Autoencoders (VAEs)

We can generalize this approach to **nonlinear factor analysis** using neural networks; a.k.a. **variational autoencoders (VAEs)**.



Variational Autoencoders

Amortization and Approximation gaps

- When we switch to nonlinear models, the posterior is no longer Gaussian ⇒ **approximation gap**
- Moreover, neural network encoder may not produce the best Gaussian approximation ⇒ **amortization gap**.
- Both lead to suboptimal inference and learning.

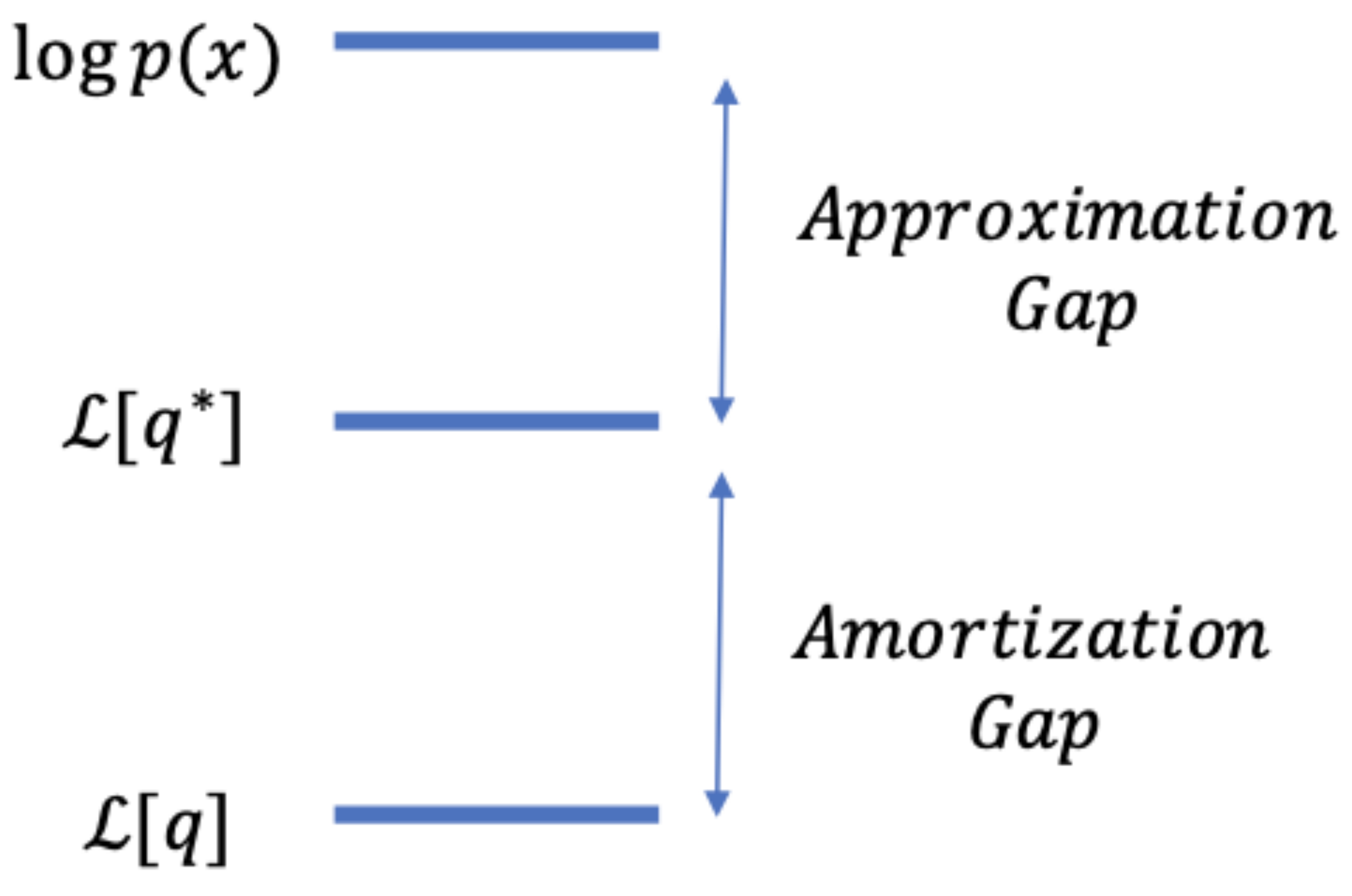


Figure 1. Gaps in Inference

Conclusion

- Instead of doing coordinate ascent on the ELBO, we can directly maximize it with SGD.
- To do so, we used an amortized variational posterior as a function of the data and variational parameters ϕ. Then we used the reparameterization trick to get Monte Carlo estimates of the ELBO and its gradients.
- This approach connects factor analysis to a linear variational autoencoder.
- The nice thing about this approach is that it generalizes to nonlinear factor models too!