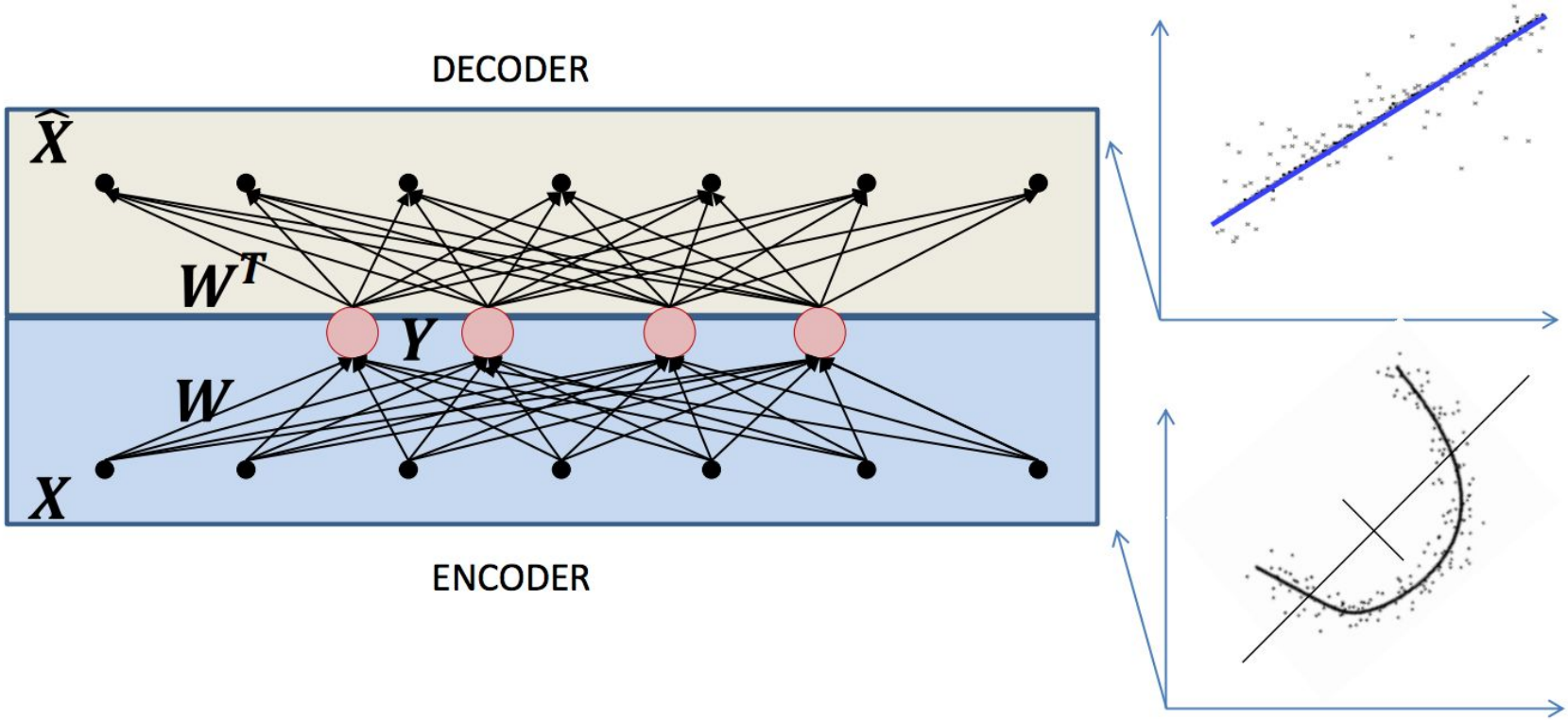
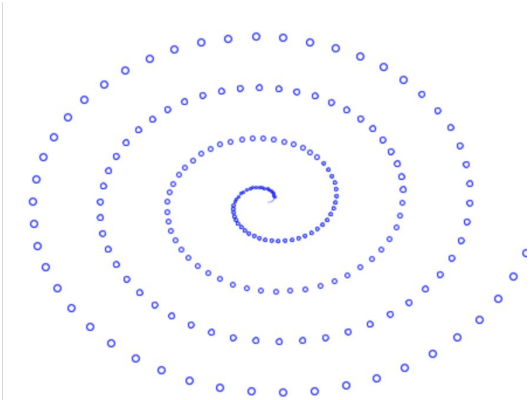
Variational Autoencoders

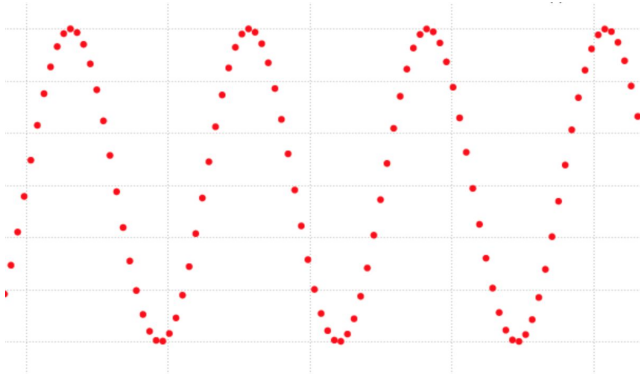
Recitation 9

Problem Setup



Problem Setup

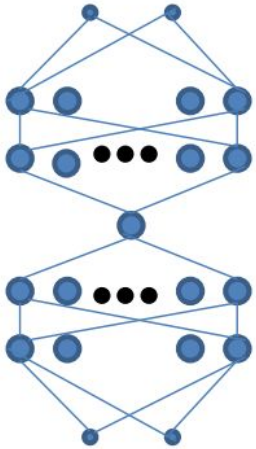


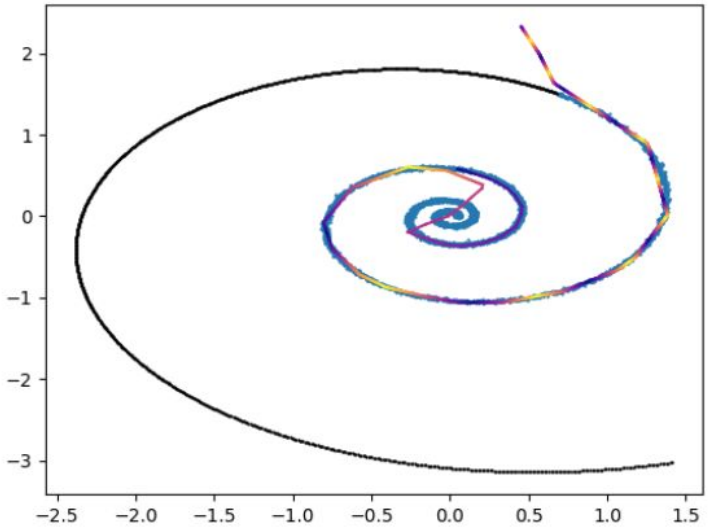


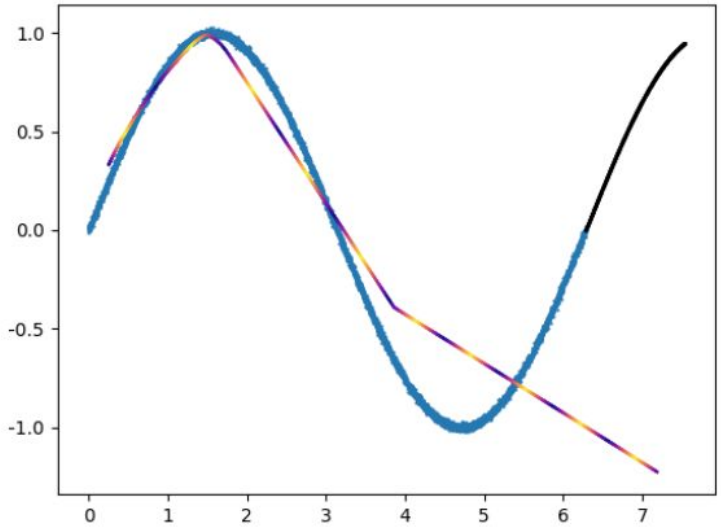
Autoencoders are able to find lower dimensional representations of data

We want the representations to not only reconstruct the inputs, but also be able to interpolate and/or extend the data

Problem Setup

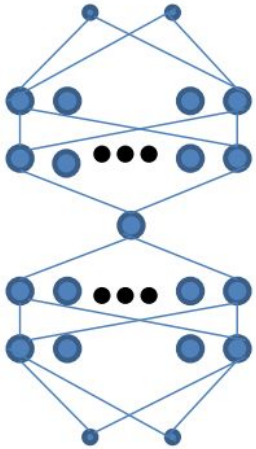


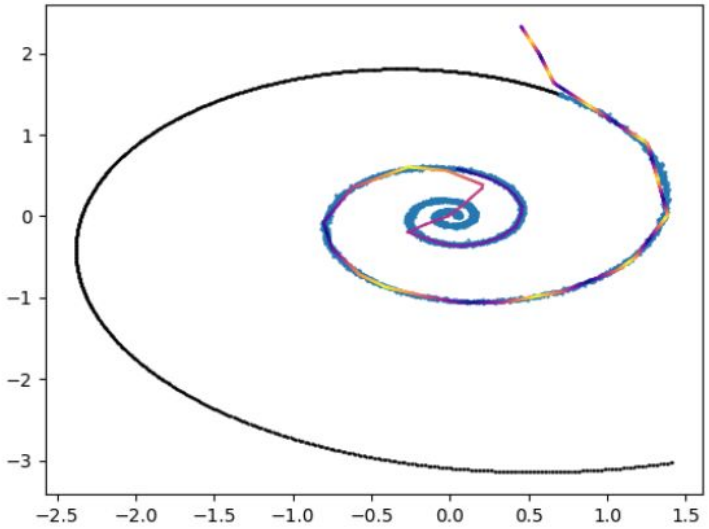




Data that are evidently generated from a single parameter cannot be ‘learnt’ by a vanilla AE

Problem Setup

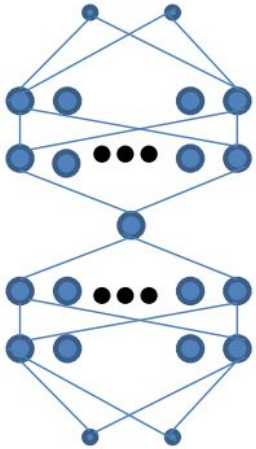


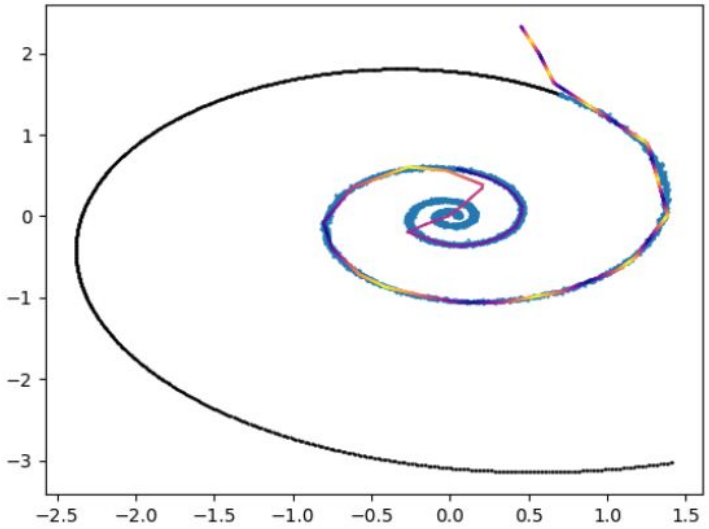


Spiral fails to smoothly interpolate between training data; unable to follow the curve beyond the last training example.

Data that are evidently generated from a single parameter cannot be learnt by a vanilla AE

Problem Setup



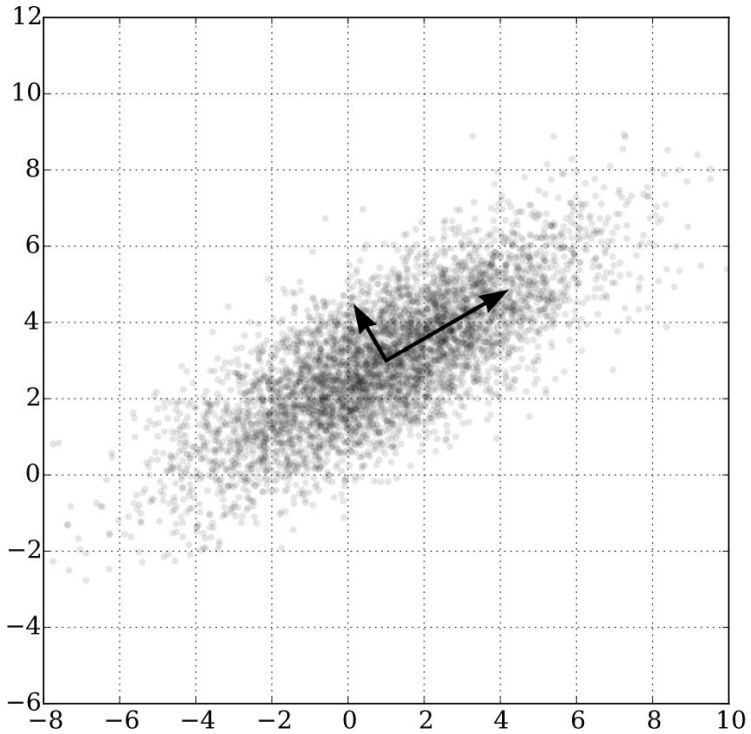


Spiral fails to smoothly interpolate between training data; unable to follow the curve beyond the last training example.

How do you find an appropriate representation?

Data that are evidently generated from a single parameter cannot be learnt by a vanilla AE

A long time ago in a semester far, far away

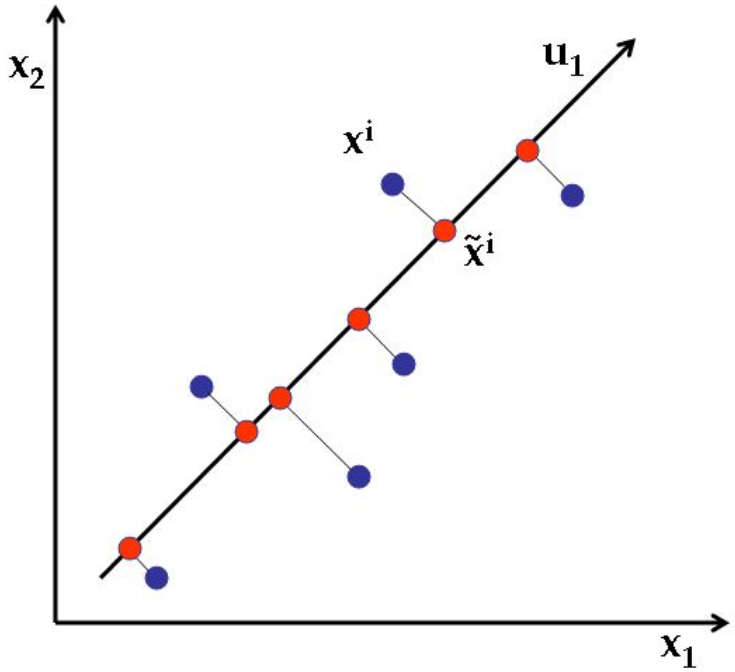


Principal Component Analysis

Fit an n-dimensional ellipsoid to the data

Minimize distortion

Principal component analysis

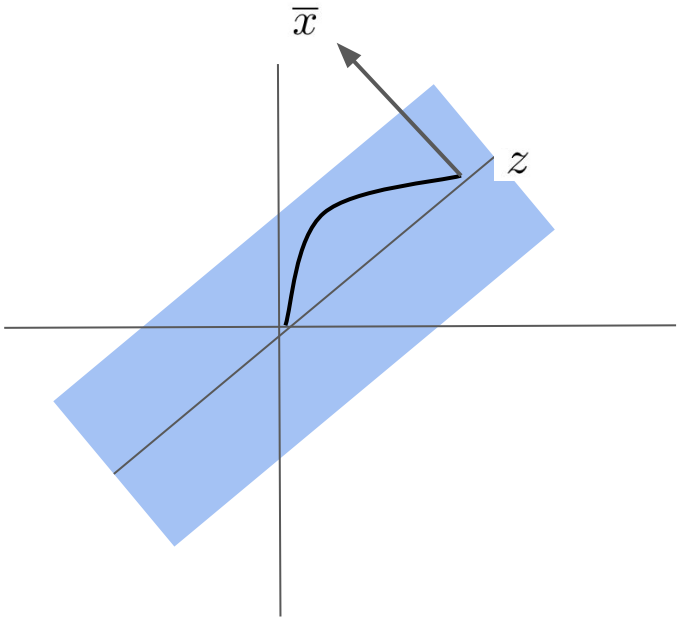


$$X = Vz + \epsilon$$

$$\underset{V,z}{\operatorname{argmin}} ||X - Vz||^2 + \Lambda(VV^T - \mathbb{I})$$

Fits training set by minimizing reconstruction error - no discussion of generality

Statistical interpretation of PCA



$$\overline{x} = Vz + \epsilon$$

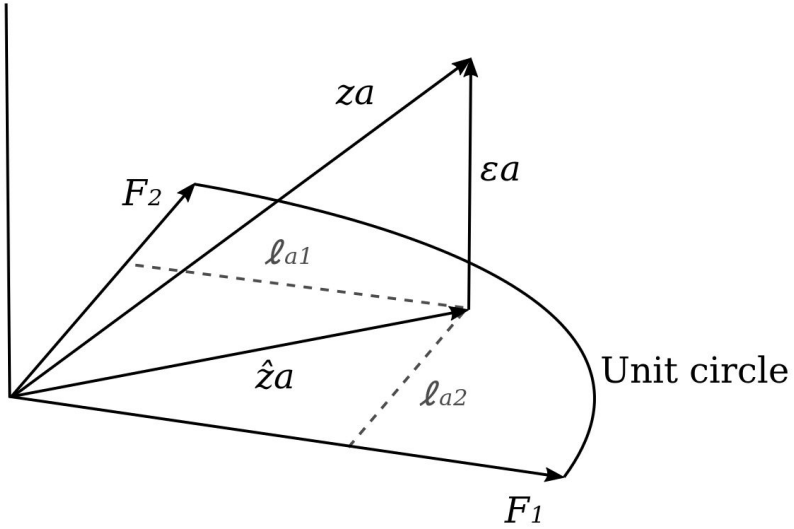
$$z \sim \mathcal{N}(0, B)$$

$$\epsilon \sim \mathcal{N}(0, E)$$

Assumption - error term is orthogonal to plane defined by V i.e. signal is purely along principal subspace, noise is purely orthogonal to principal subspace

Not true in general!

Factor Analysis



$$\overline{x} = Vz + \epsilon$$

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

$$\epsilon \sim \mathcal{N}(0, E)$$

Here, E is full rank - no longer constrained to be perpendicular to the plane

FA assumes there are underlying latent factors that affect our observed data

Autoencoders: our set of assumptions

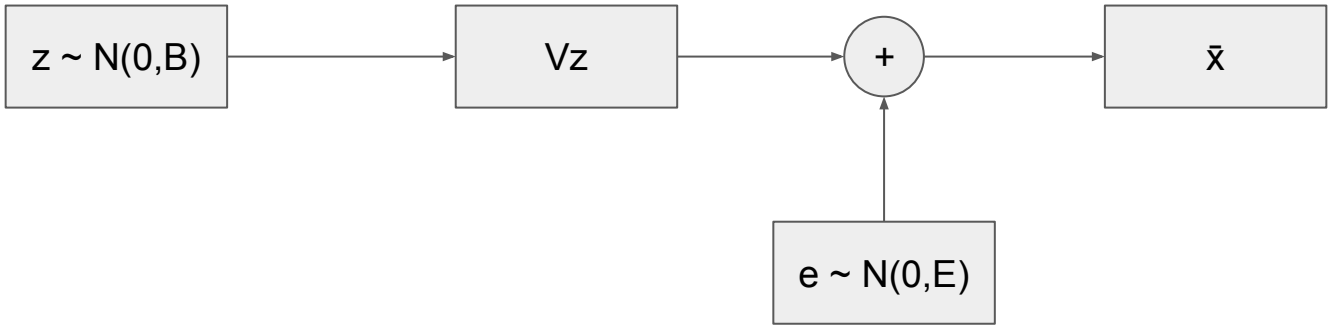
1. All data is generated as a continuous function of some latent variables
2. The exact choice off how latent variables map to data is up to us to decide
3. All our data is unlabeled, so nothing can be used as something related to the latent variables
4. (VAE's only) The function from latent variables to data is probabilistic in nature

What's the our reason behind holding each assumption?

Can you think of different problem setups where each assumption doesn't hold?

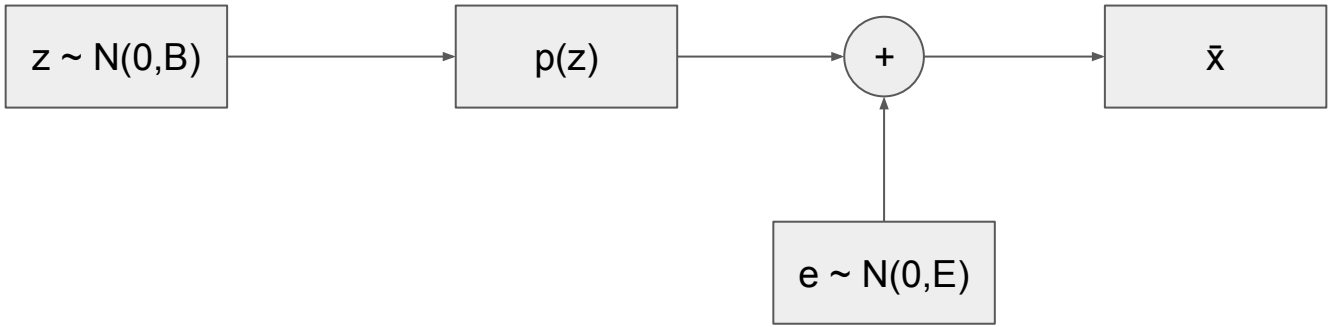
Why does having these assumptions force us into something similar to an encoder-decoder type of arrangement? How does EM address issues we face while holding these assumptions

FA as a generative model



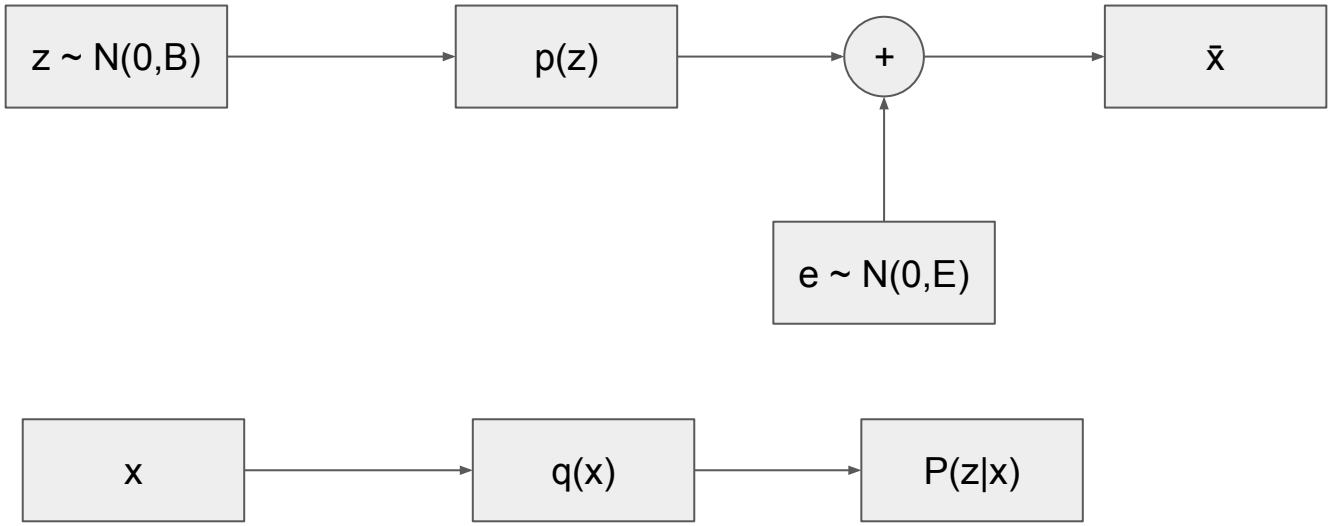
More powerful - error need not be orthogonal to V
Still linear in z!

Non-linear FA



Even more powerful - Gaussian step can now be along a curved surface!
This is the motivation behind VAE. Learn p(z) using nnet
p() can learn to transform random variable Z into a completely different distribution

VAE setup



Training objective for VAEs

We want to maximize log likelihood of data given model parameters.

$$\log P(\mathcal{X}) = \sum_{\boldsymbol{x} \in \mathcal{X}} \log P(\boldsymbol{x}; \theta)$$

Likelihood of each example is $P(\boldsymbol{x}; \theta) = \int P(\boldsymbol{x} \mid \boldsymbol{z}; \theta) P(\boldsymbol{z}) d\boldsymbol{z} \approx \sum_{\boldsymbol{z} \in S(\boldsymbol{x})} P(\boldsymbol{x} | \boldsymbol{z}; \theta)$

where $S(\boldsymbol{x}) := \{\boldsymbol{z}'; \boldsymbol{z}' \sim P(\boldsymbol{z})\}$

But sampling from high dimensional latent spaces is inefficient!

Sampling

$$S(\boldsymbol{x}) := \{\boldsymbol{z}'; \boldsymbol{z}' \sim P(\boldsymbol{z})\}$$

Introduce some Q(z|x) as distribution over z that likely produced this x to approximate the true P(z|x). From KL divergence we arrive at

$$\log P(X) - \mathcal{D}\left[Q(z|X) \| P(z|X)\right] = E_{z\sim Q}\left[\log P(X|z)\right] - \mathcal{D}\left[Q(z|X) \| P(z)\right]$$

We want to maximize LHS, and we are going to show how to optimize RHS.

Training

$$\log P(X) - \mathcal{D}\left[Q(z|X)\|P(z|X)\right] = E_{z\sim Q}\left[\log P(X|z)\right] - \mathcal{D}\left[Q(z|X)\|P(z)\right]$$

We want to optimize the above objective for X drawn from data.

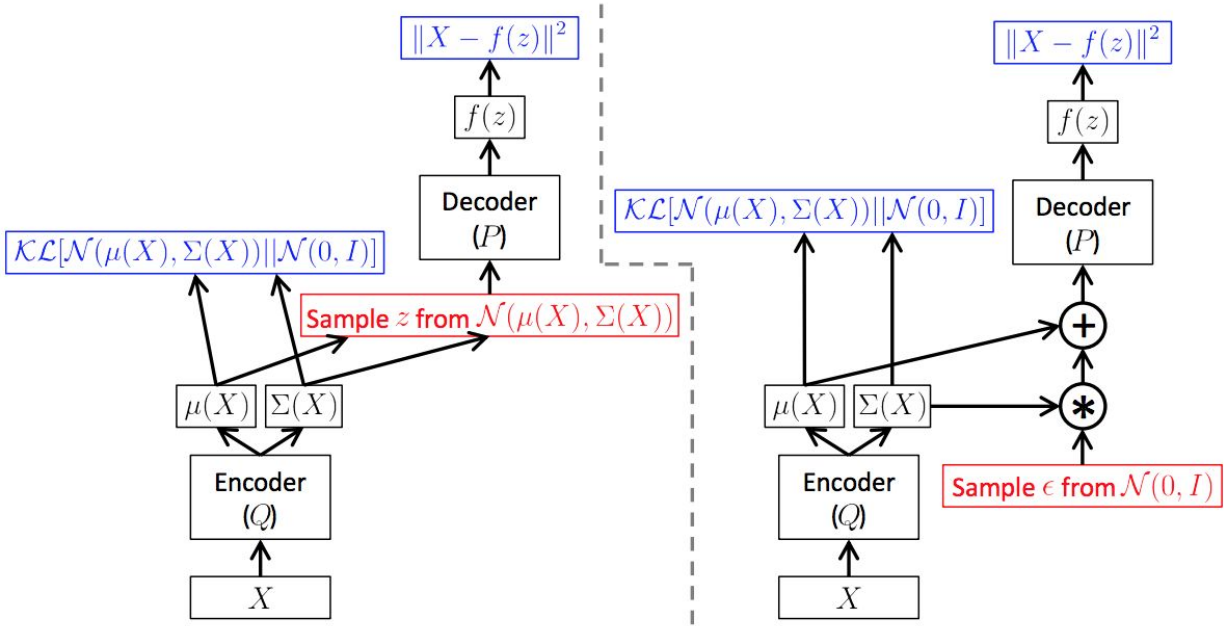
$Q(z|X) \sim \mathcal{N}(\mu, \Sigma)$

Sample one z, use it to approximate first term, then compute gradient of

$$\log P(X|z) - \mathcal{D}\left[Q(z|X)\|P(z)\right]$$

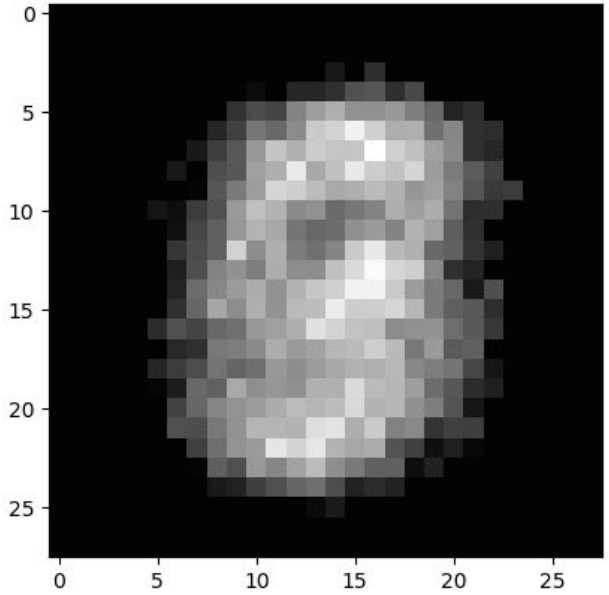
‘Reparametrize’ in order to backpropagate through the sampling term by treating $\mathcal{N}(0, \mathbb{I})$ as a component of the input layer

Backpropagation



Tutorial on Variational Autoencoders, Carl Doersch 2016

Notebook



MNIST example