Generative Modeling: VAEs

Vikash Gilja

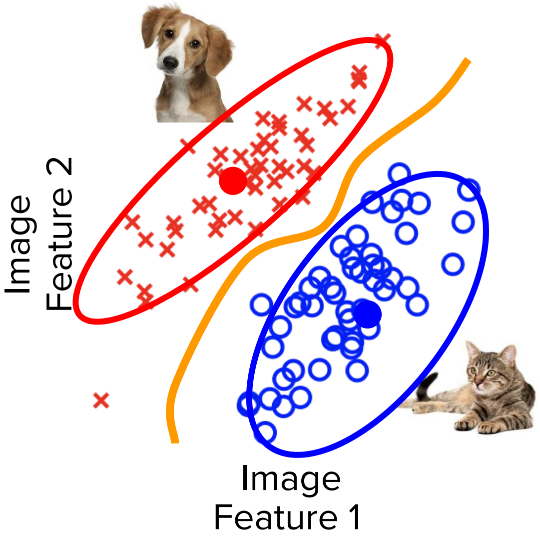
neuromatch academy

Generative vs. Discriminative Models

Lecture 1

neuromatch academy

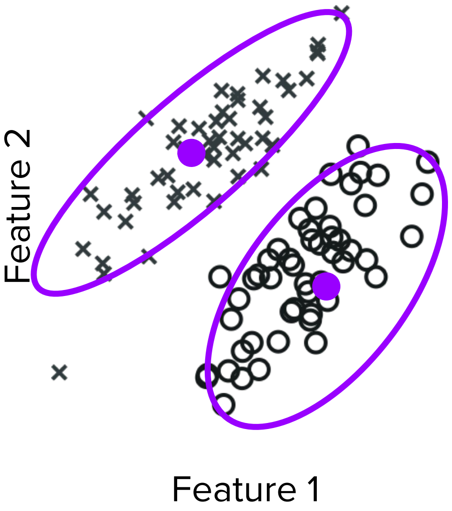
Generative vs. Discriminative Modeling



Discriminative Modeling: Is it a cat or a dog?

Generative Modeling: What do cats and dogs "look" like?
(supervised)

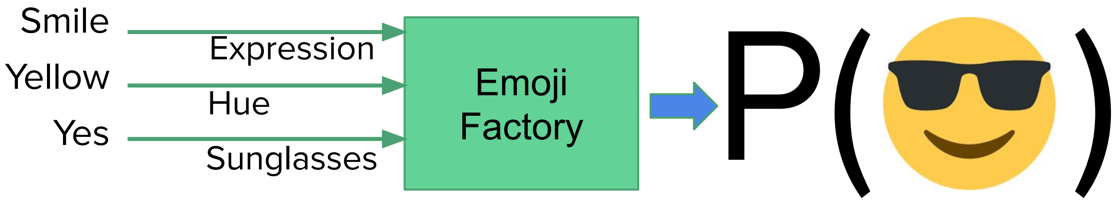
What do the data "look" like?
(unsupervised)



Why Create a Generative Model?

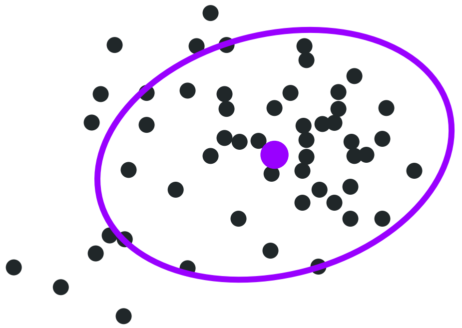
Modeling “how” the data are structured enables this model to be used in many ways:

- Generate new examples / interpolation
- Anomaly detection: Was it likely that the current example was generated?
- Structured modifications of the model
- Potential for generalization
- …



Generative Modeling & Probability Theory

Training Data: $\mathbf{X} = \{ x_1, x_2, \ldots, x_n \}$



What's the probability of x?

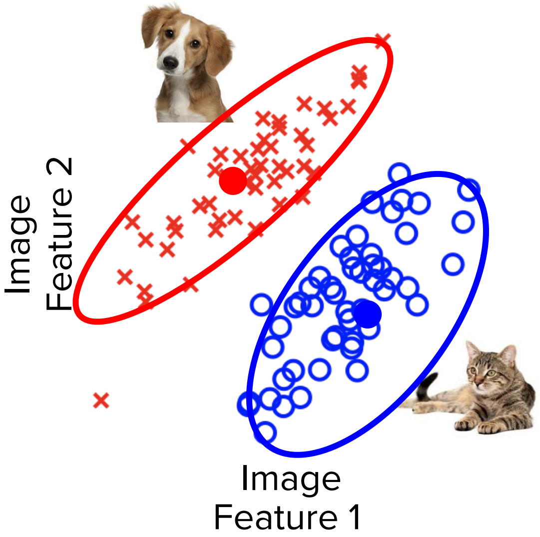
$P(x) = ?$

One simple choice:

$P(x) = \mathcal{N}(\mu, \boldsymbol{\Sigma})$ = Gaussian(mean, covariance)

We'll explore models that are much more "expressive"

Let's condition on another variable...



P(x) = P(x | dog) P(dog) + P(x | cat) P(cat)

How likely is a dog? (*prior*)

It's a dog, what's the probability it's x?

P(dog | x) = What's the probability that x is a dog?

P(cat | x) = What's the probability that x is a cat?

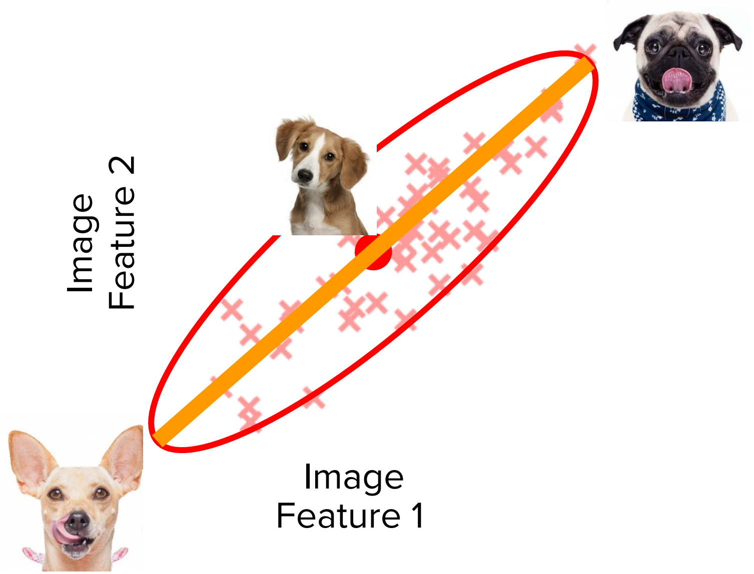
cat or dog? Apply Bayes rule to classify!

Let's add another variable...

P(x | dog, z)

What "kind" of dog are you?

Pick a value for z and generate a dog!



Latent Variable Models

Lecture 2

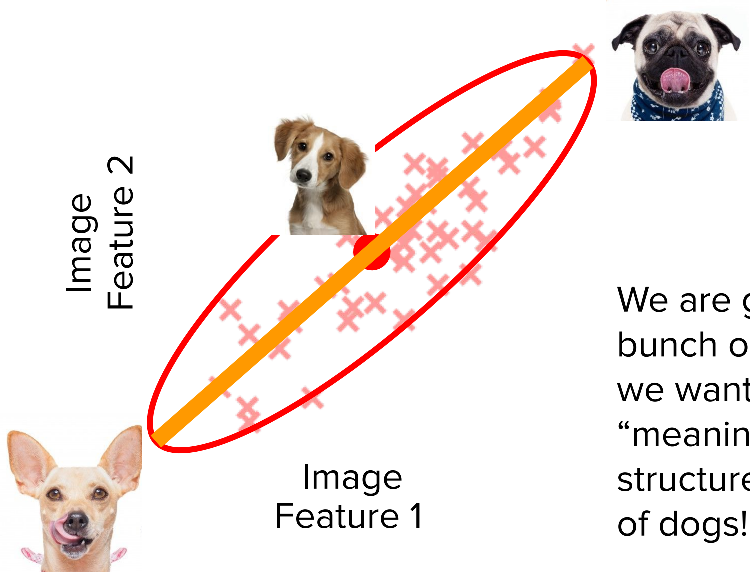
neuromatch academy

Remember z? It's latent!

$P(x \mid \text{dog}, z)$

What "kind" of dog are you?

Pick a value for **z** and generate a dog!



We are given a bunch of dogs and we want to learn "meaningful" structure in the space of dogs!

- Training Data: $\mathbf{X} = \{ x_1, x_2, \ldots, x_n \}$
- We are told that these are all dogs
- If we aren't given **z**, then **z** is a *hidden* or *latent* variable.

Latent Variable Models

A model with *hidden* or *latent* variables (variables that are not part of the training set) is a ***Latent Variable Model***

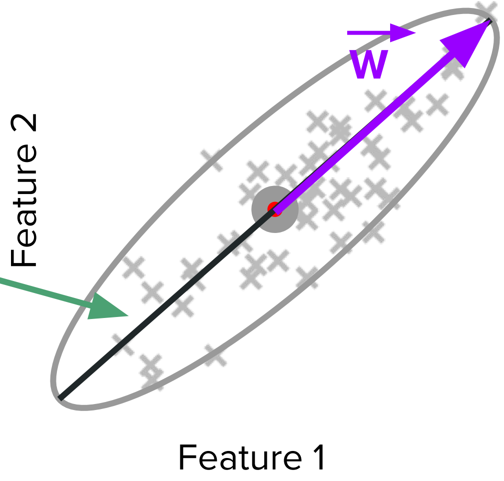
Latent variable modeling techniques can produce powerful generative models

Before we look at Deep Learning based *Latent Variable Models*, let's consider a model you likely know well

Principal Component Analysis (PCA)

Is PCA a latent variable model?

I'm the first principal component axis,
I'm the linear projection that describes the greatest variability in the data. Project onto me and I'll give you a latent variable!



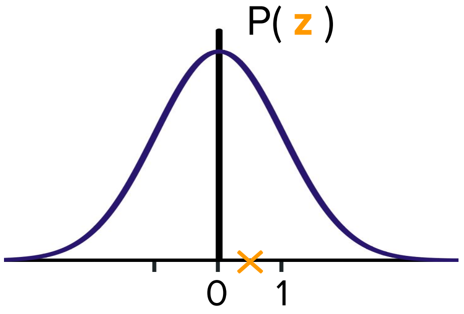
Is PCA a generative model?

Nope. The principal components are just vectors, the don't define P(x)

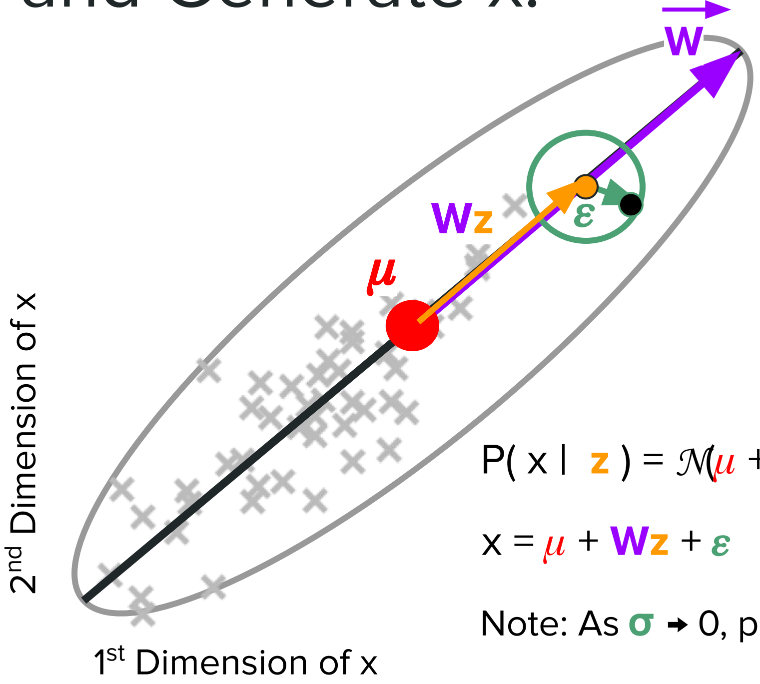
Let’s “Upgrade” PCA and Generate x!

Probabilistic PCA (pPCA) extends PCA and is a generative model

P(**z**) = $\mathcal{N}(0, \mathrm{I})$ = “the unit Gaussian”



Note: **z** can be multivariate!!
Our illustration shows a 1D **z**



P(x | **z**) = $\mathcal{N}(\mu + \mathbf{Wz}, \sigma^2\mathrm{I})$

x = $\mu + \mathbf{Wz} + \varepsilon$ P(ε) = $\mathcal{N}(0, \sigma^2\mathrm{I})$

Note: As $\sigma \to 0$, pPCA reduces to PCA

Autoencoders

Lecture 3

neuromatch academy

PCA: Powerful, but limited

- PCA is a commonly used tool for dimensionality reduction
- PCA only models linear relationships

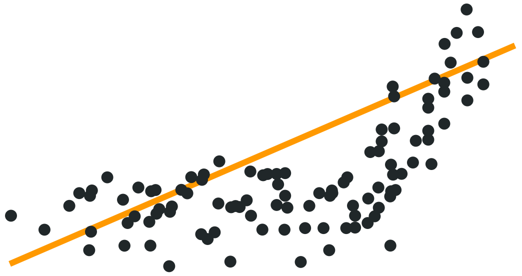
$$\mathbf{x} \approx \mathbf{F}(\mathbf{z}) = \mathbf{W}\mathbf{z} + \mu$$

- W, μ minimizes the squared error between **x** and $\mathbf{W}\mathbf{z} + \mu$

$$\min_{\mathbf{W}} \; \| \mathbf{x} - \mathbf{W}\mathbf{z} - \mu \|^2$$

- Non-linear relationships?
- Modify our definition of error?
- Structure? (e.g. convolutional and recurrent structure)

Goal: Describe **x** with lower dimensional **z**



Autoencoders to the rescue!

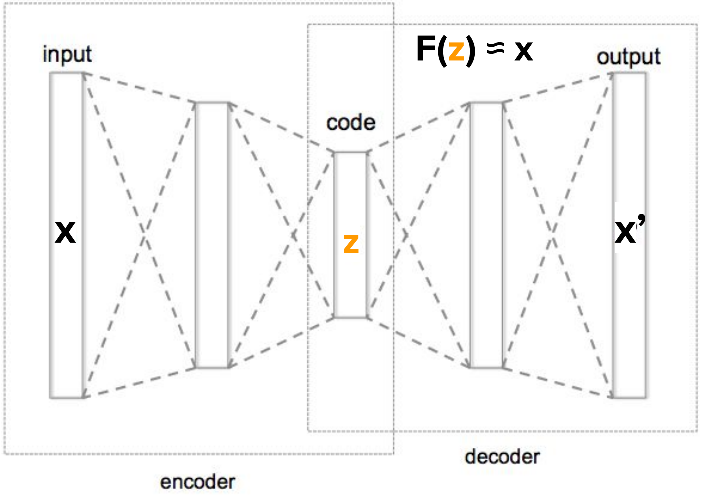


Image Credit: Chervinskii, CC BY-SA 4.0, via Wikimedia Commons

One or more *encoder* layers to *encode* **x** to lower dimensional **z**

Decoder layers *decode* **z** to estimate **x** as **x'**

For training, the objective (error assessment) is designable!
We aren't limited to minimizing squared error (although this is the typical choice).

Autoencoders enable non-linear modeling

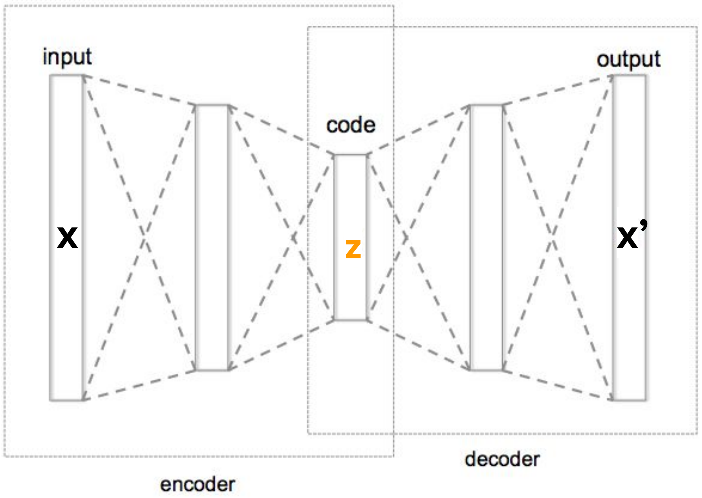


Image Credit: Chervinskii, CC BY-SA 4.0, via Wikimedia Commons

$z = \text{function}_e(W_e x + b_e)$ **Encoder**

Non-linear functions

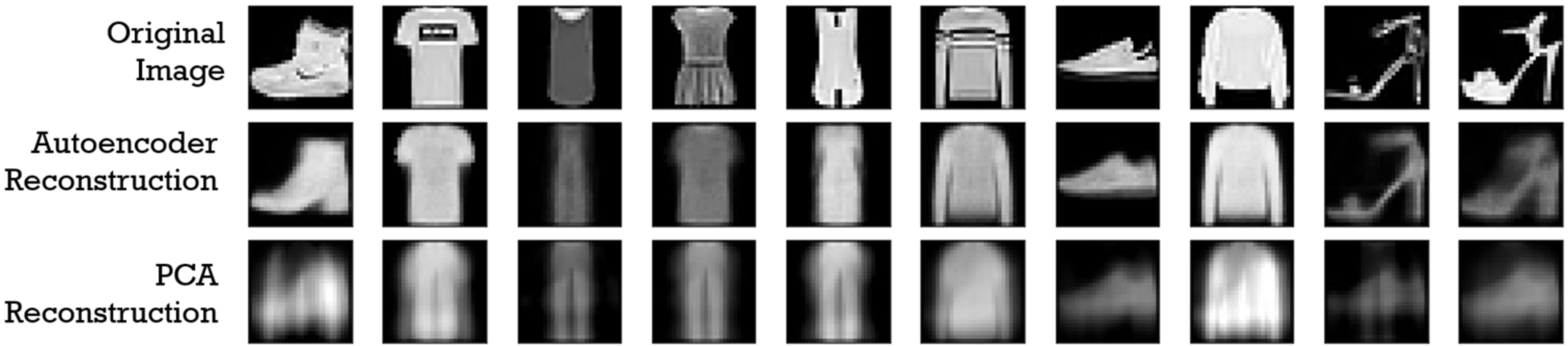
$x' = \text{function}_d(W_d z + b_d)$ **Decoder**

Multiple layers allow for increasingly complex mappings!

Autoencoders enable non-linear modeling

2 Dimensional Latent Spaces

Fashion MNIST: 10,000 images of wardrobe; 10 classes



PCA is (basically) a specific Autoencoder

IF the encoder & decoder are **linear**:

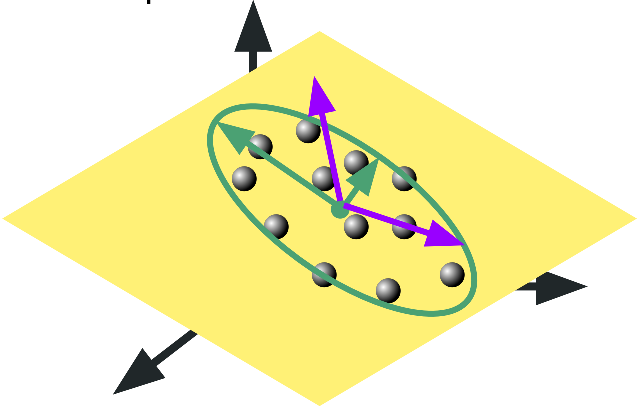
$$\mathbf{z} = \mathbf{W_e x} + \mathbf{b_e}$$

$$\mathbf{x'} = \mathbf{W_d z} + \mathbf{b_d}$$

AND the object is the (commonly used) minimization of squared error: **minimize || x' - x ||²**

THEN, the autoencoder will result in a **solution that is equivalent* to PCA!**

* The two solutions will span the same linear subspace.



Variational Autoencoders

Lecture 4

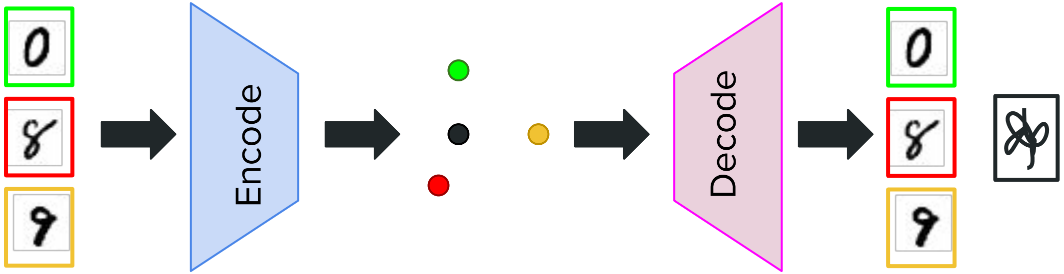
neuromatch academy

Are Autoencoders Generative Models?

No! Like PCA, autoencoders don't define a distribution over **x**

Why do we care?

This limits the interpretation of the latent space

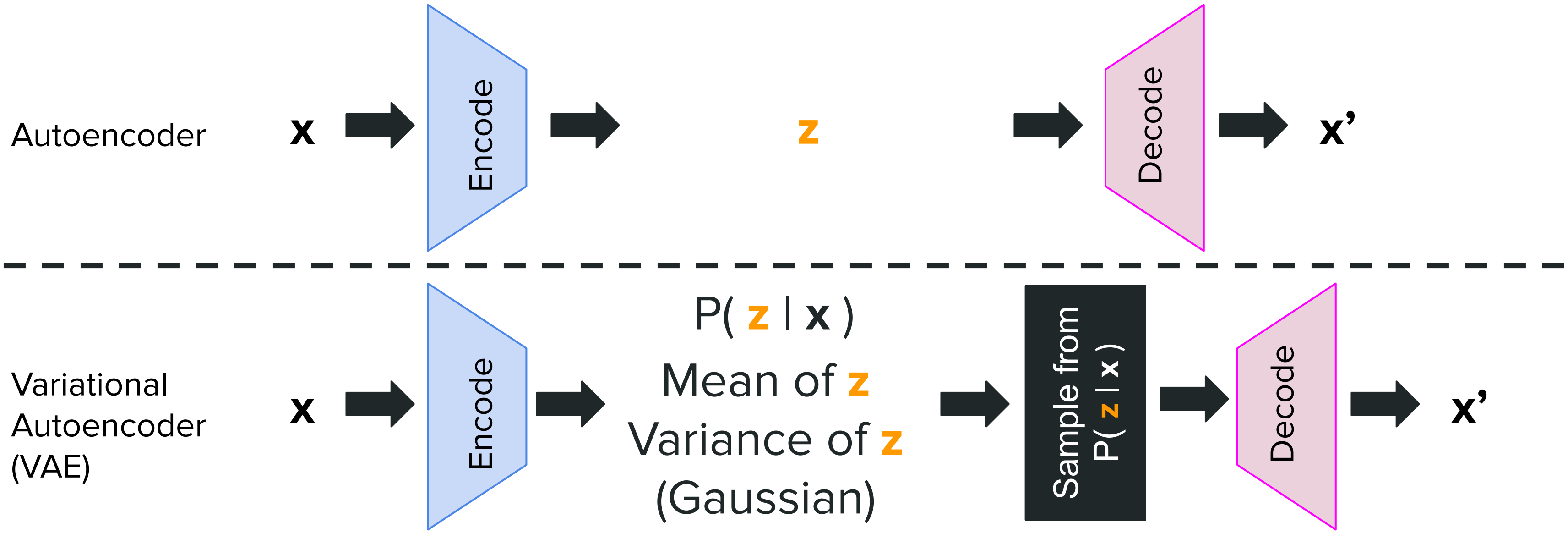


Latent space compresses input, but can have limited meaning

Example Latent Space

A Generative Autoencoder

Variational Autoencoders (VAEs) define a generative model



Variational Autoencoder (VAE)

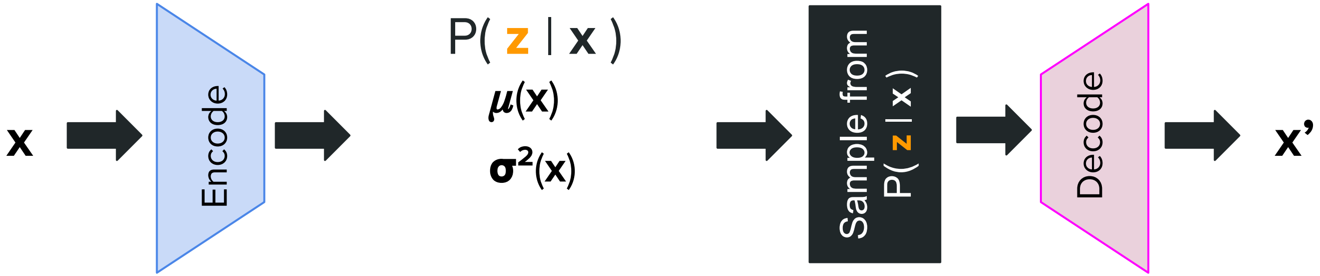
VAE training balances two objectives:

1) Encoder Objective: Estimate the *posterior* P(**z** | **x**) s.t. P(**z**) is a unit Gaussian: $\mathcal{N}(0, I)$
2) Decoder Objective: Estimate P(**x** | **z**) to reconstruct **x** with high probability

The encoder maps **x** to **vectors** representing the **mean** and **variance** of **z**:

P(**z** | **x**) = $\mathcal{N}(\boldsymbol{\mu}(\mathbf{x}), \mathrm{diag}(\boldsymbol{\sigma}^2(\mathbf{x})))$ -- **Note:** the dimensions of **z** are assumed to be uncorrelated

Variational Autoencoder (VAE)

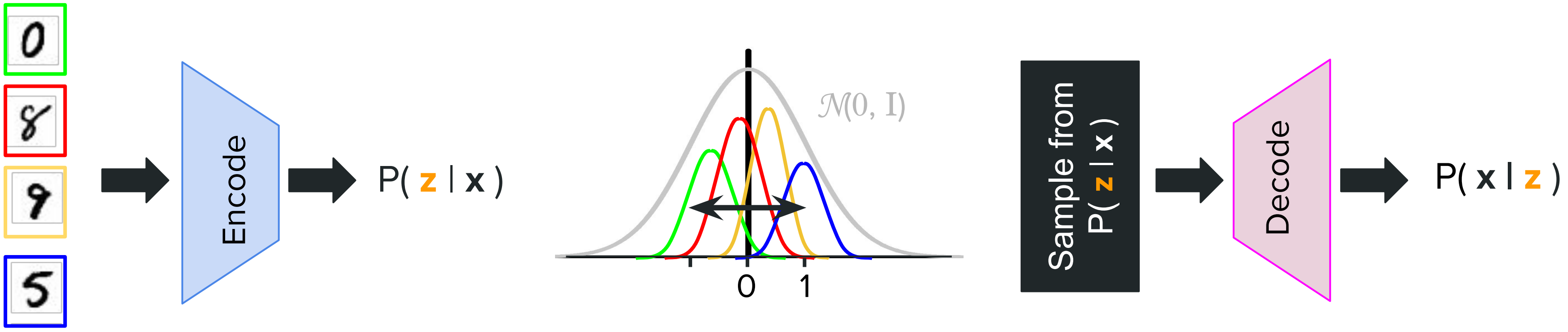


VAE Objective

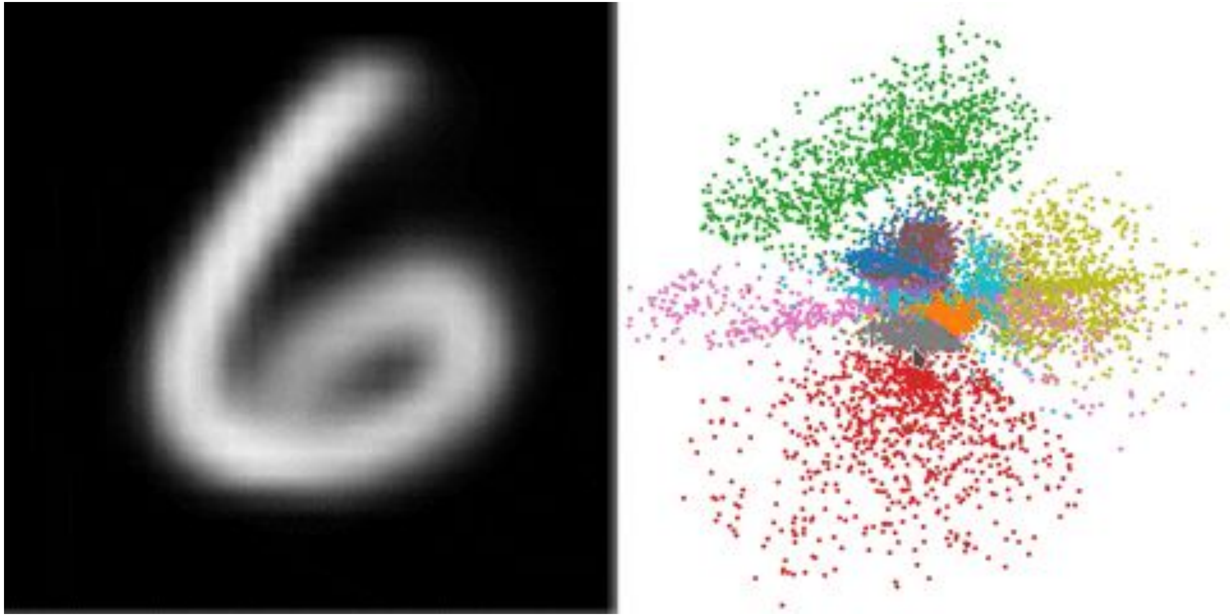
VAE training balances two objectives:

1) Encoder Objective: Estimate the *posterior* P(**z** | **x**) s.t. P(**z**) is a unit Gaussian: $\mathcal{N}(0, \mathrm{I})$
2) Decoder Objective: Estimate P(**x** | **z**) to reconstruct **x** with high probability

$$P(\mathbf{z}) = \int P(\mathbf{z} \mid \mathbf{x})\, P(\mathbf{x})\, d\mathbf{x}$$



A Generative Autoencoder



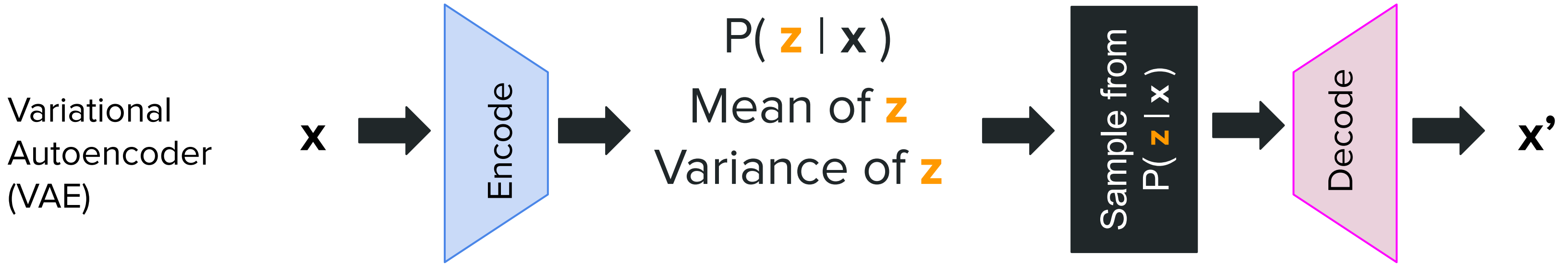
Taylor Denouden, VAE Latent Space Explorer, MIT License, GitHub

State-of-the-Art VAEs

Lecture 5

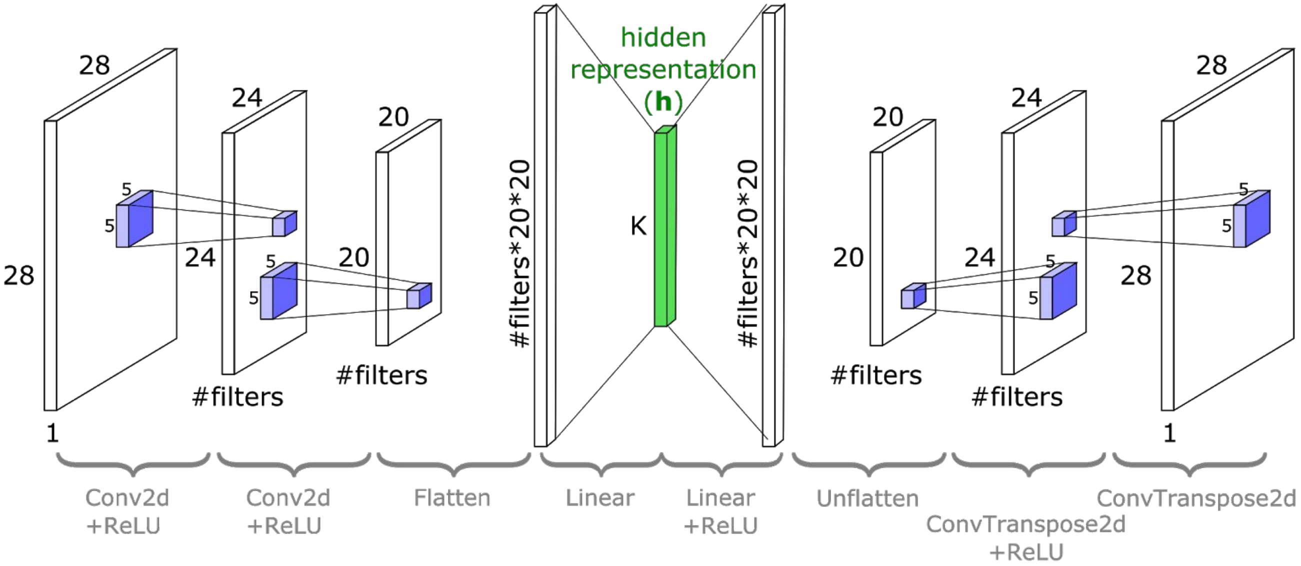
neuromatch academy

A Variety of VAEs

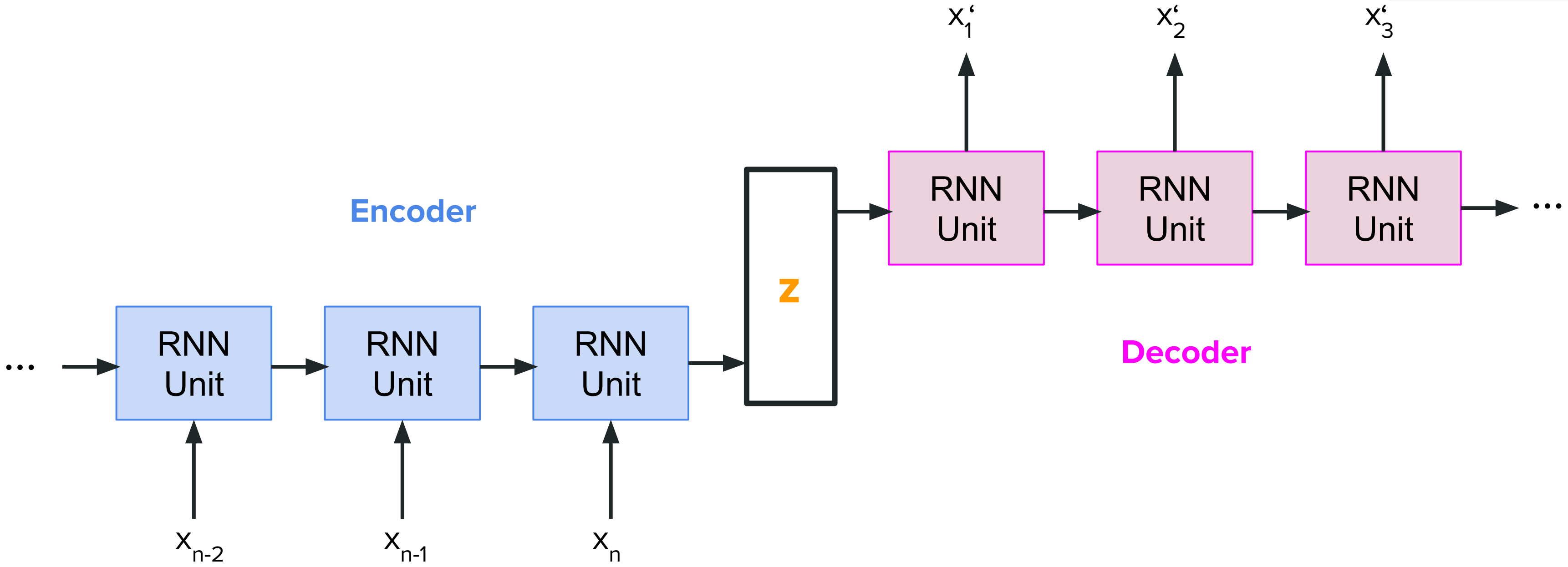


- VAEs / Autoencoders designed to better address specific problems with parameter sharing
  - **Image data** are often **encoded** with a **convolution neural network** and **decoded** with a **deconvolution neural network**
  - **Timeseries data, like text & audio,** are often **encoded/decoded** with **recurrent neural networks**
- VAEs variants with **modified objective functions** and **latent space definitions**

ConvAutoEncoder



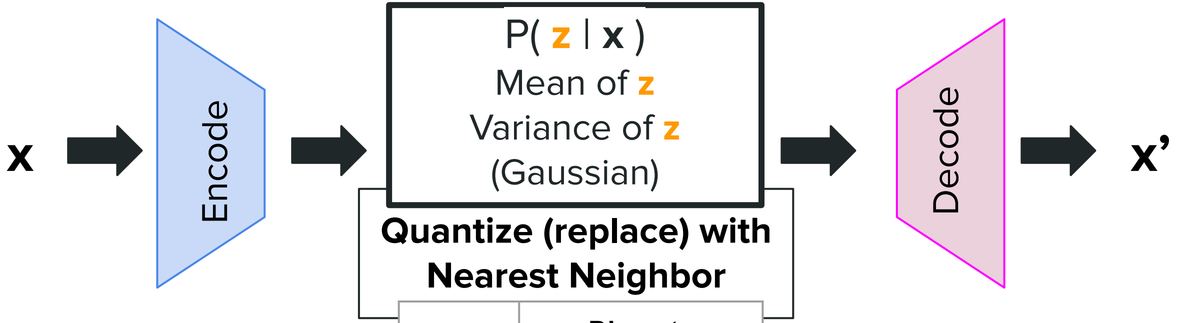
Seq2Seq Autoencoder



VQ-VAE

Vector Quantized Variational Autoencoder - Discretization of the Latent Space

Variational
Autoencoder
(VAE)



	Discrete Codebook
1	z^1
2	z^2
...	...
K	z^K

β-VAE (Disentangled VAE)

β-VAE modifies the objective, emphasizing the encoder objective, to "disentangle" the latent space.

VAE training balances two objectives:

1) Encoder Objective: Estimate the *posterior* P(**z** | **x**) s.t. P(**z**) is a unit Gaussian: $\mathcal{N}(0, I)$ **Weighted by β > 1**
2) Decoder Objective: Estimate P(**x** | **z**) to reconstruct **x** with high probability

Sampling latent direction best aligned to head angle

β-VAE

VAE