UNIVERSITY OF TORONTO
SCHOOL OF CONTINUING STUDIES

3546 – Deep Learning

Module 8: Representational Learning & Variational Methods

Course Syllabus

Module	Topic	Deliverables
1	Course Intro + Review	Term Project Released
2	Model Tuning	Assignment 1 Released
3	Convolutional Networks	
4	Deep Computer Vision	Assignment 1 Due, A2 Released
5	Recurrent Neural Networks	
6	Natural Language Processing	
7	Deep Models for Text	Assignment 2 Due, A3 Released
8	Representational Learning & Variational Methods	Submit Final Project Topic Approvals**
9	Deep Generative Models	Assignment 3 Due, A4 Released
10	Speech and Music Recognition & Synthesis	
11	Term Project Presentations A	Term Project Due
12	Term Project Presentations B	Assignment 4 Due

Pick Project Topic →

Learning Outcomes for this Module

By the end of this module:

- Learners will have a comprehensive understanding of representational learning, generative models, and their applications.
- Learners will know when and how to train and use (Variational) Autoencoders and,
  - are poised to follow and comprehend new advances in these domains;
  - have the foundational knowledge necessary to understand Fully Visible Belief Nets (FVBN) and Generative Adversrial Networks (GAN) that are introduced in the following module.

Module 8 - Section 1

Representational Learning

Aside: Chess Grandmasters

A chess Grandmaster can memorize the positions of all pieces on a chess board in under five seconds. But only if those pieces are placed in realistic positions from actual games, not if placed randomly.

Chess experts don’t have a better memory. Rather, they see chess patterns more easily, thanks to their experience with the game. This in turn helps them play better chess!

Efficient Representations of Data

Which of the following number sequences do you find easiest to memorize?

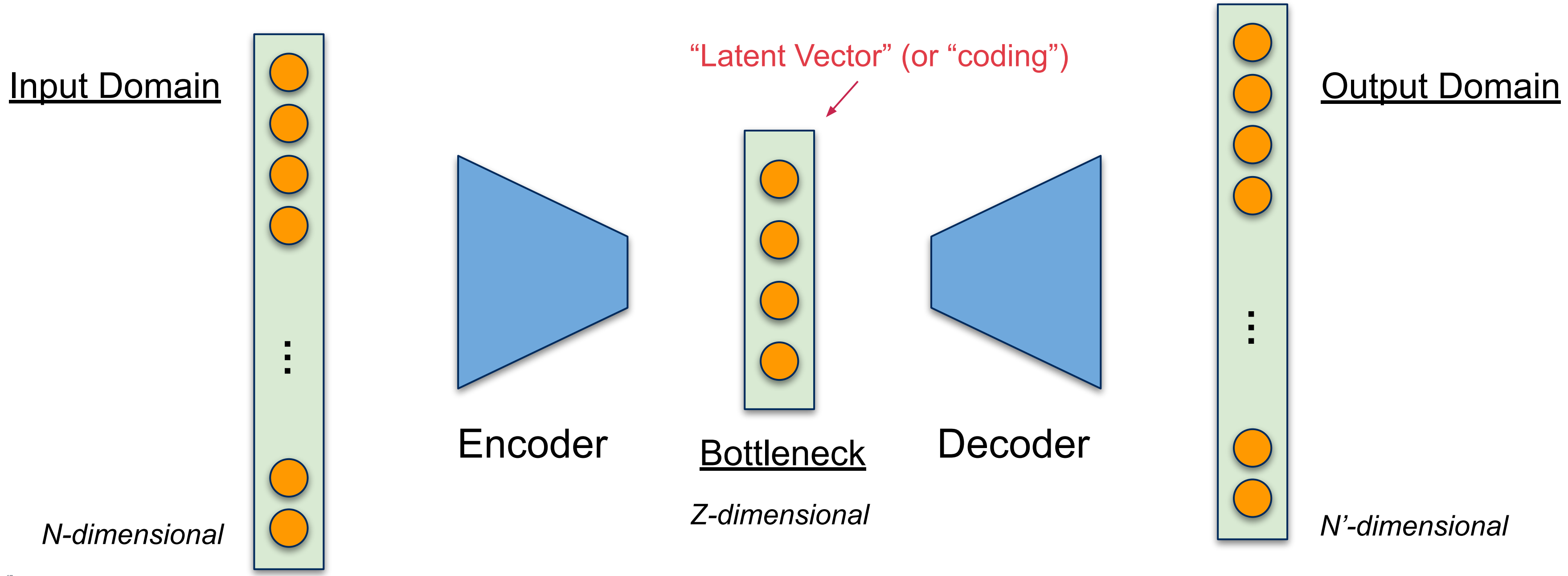
- 40, 27, 25, 36, 81, 57, 10, 73, 19, 68
- 50, 48, 46, 44, 42, 40, 38, 36, 34, 32, 30, 28, 26, 24, 22, 20, 18, 16, 14

Takeaways:

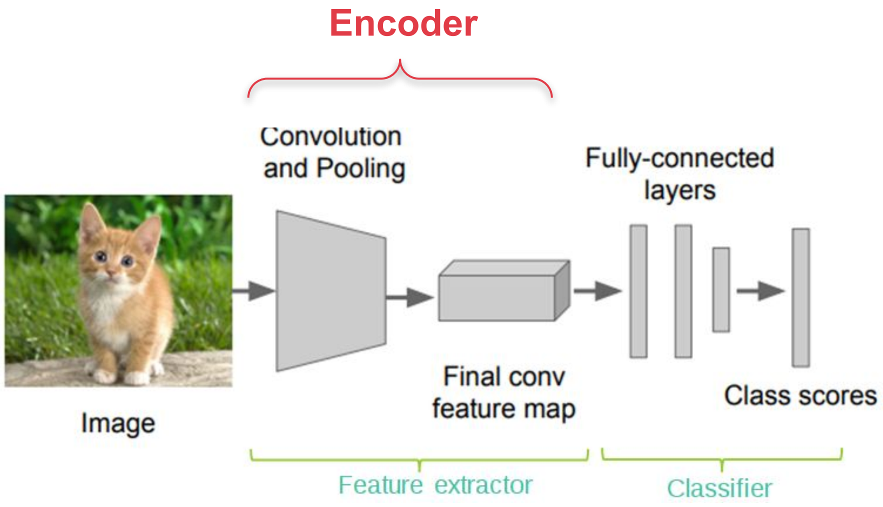
- Noticing patterns helps you store information more efficiently.
- Conversely, forcing yourself to store information efficiently can help you discover useful patterns.

Representational Learning

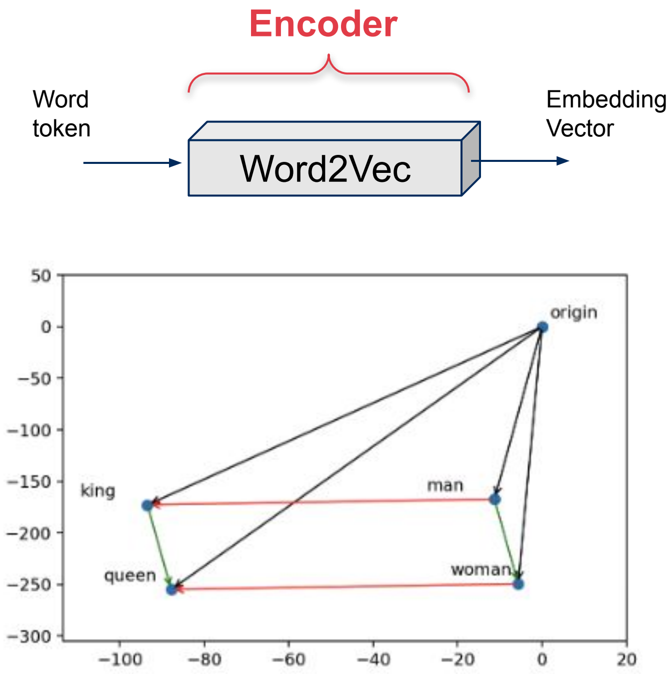
Techniques that automatically learn **efficient** and **robust** representations of data (i.e. good features). Often use the concepts of **encoder** and **decoder**.



Examples of Encoders

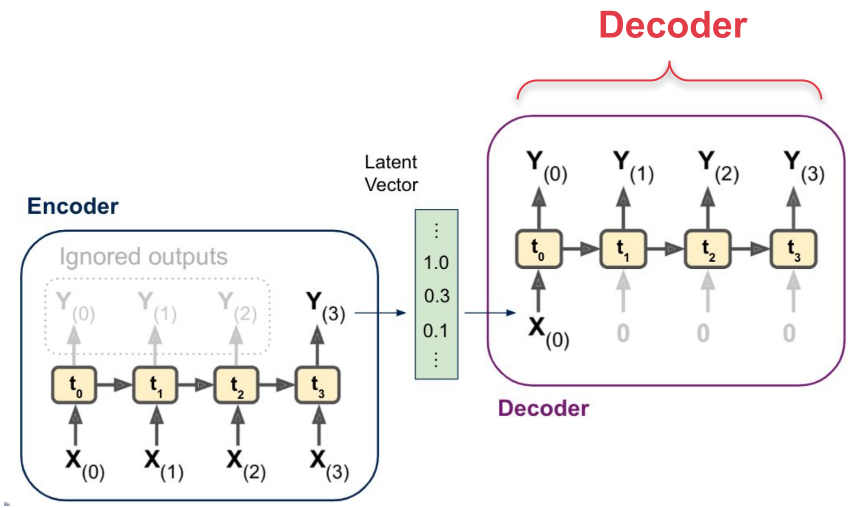


Transfer Learning

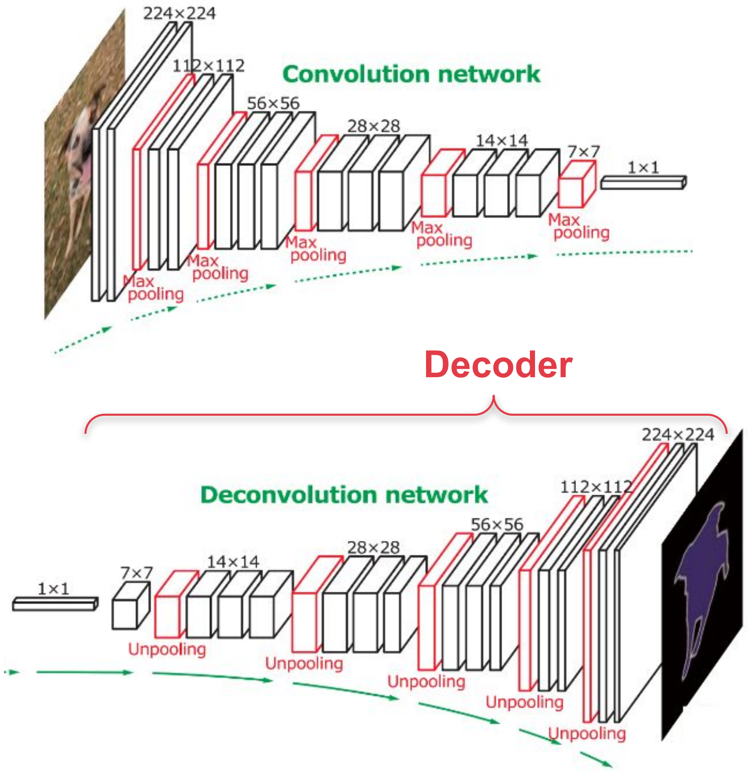


Word Embeddings

Examples of Decoders



Encoder-Decoder Language Translation



Convolution-Deconvolution networks for Semantic Segmentation

Challenges for Feature Learning

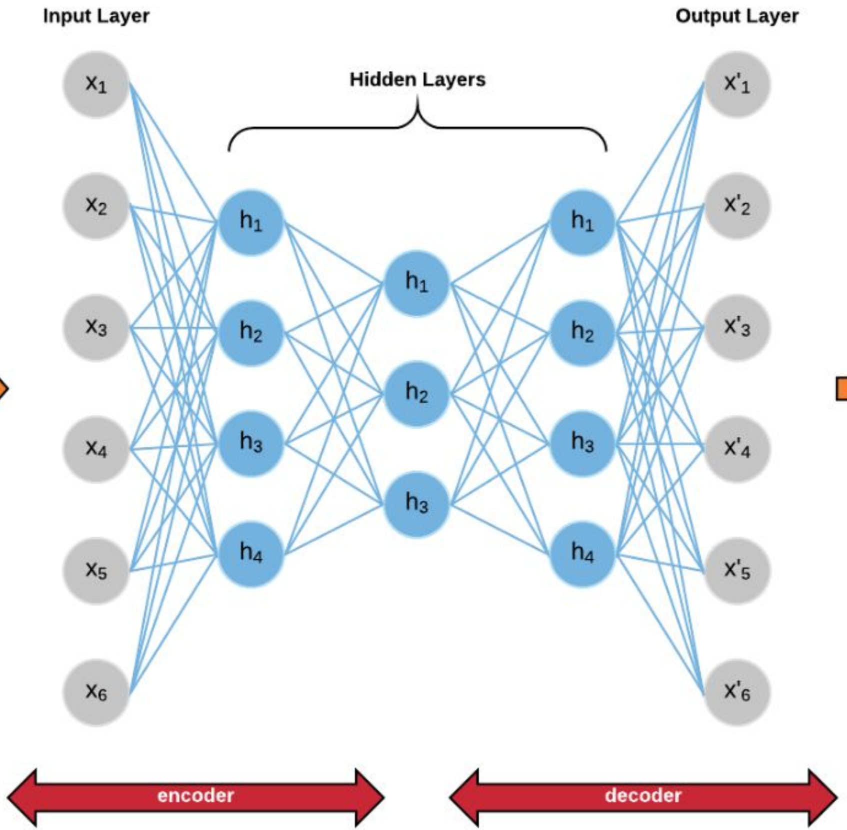
- Most data is **unlabelled**.
  - Can we still learn useful features?
- Data can be **noisy**.
  - How can we make our features robust against noise?

Module 8 - Section 2

Autoencoders

Autoencoders

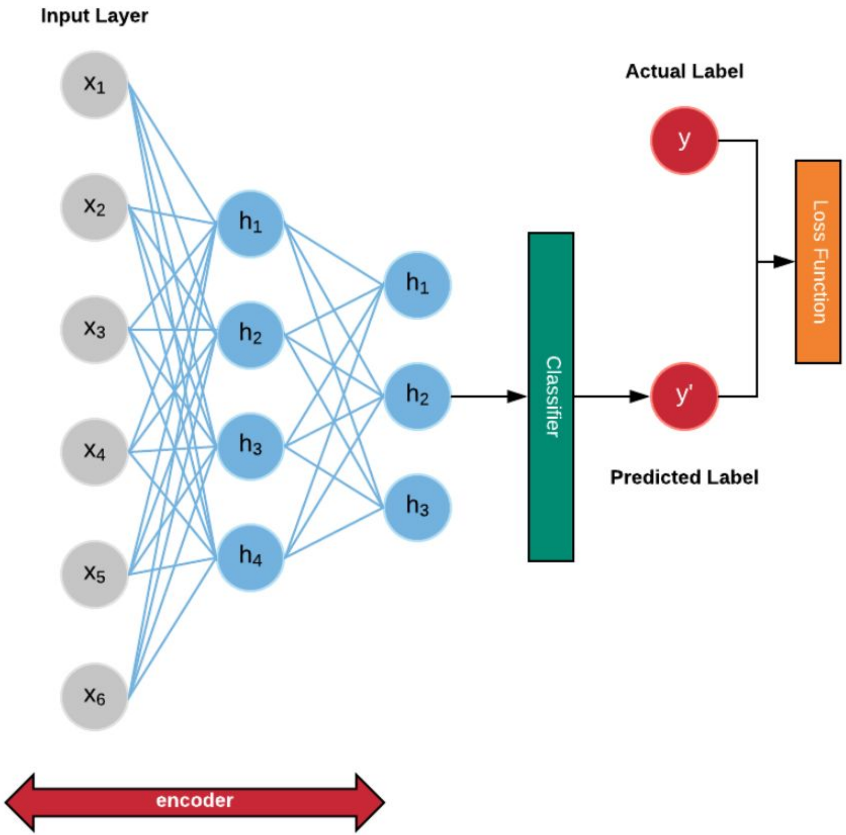
A form of **self-supervised** learning



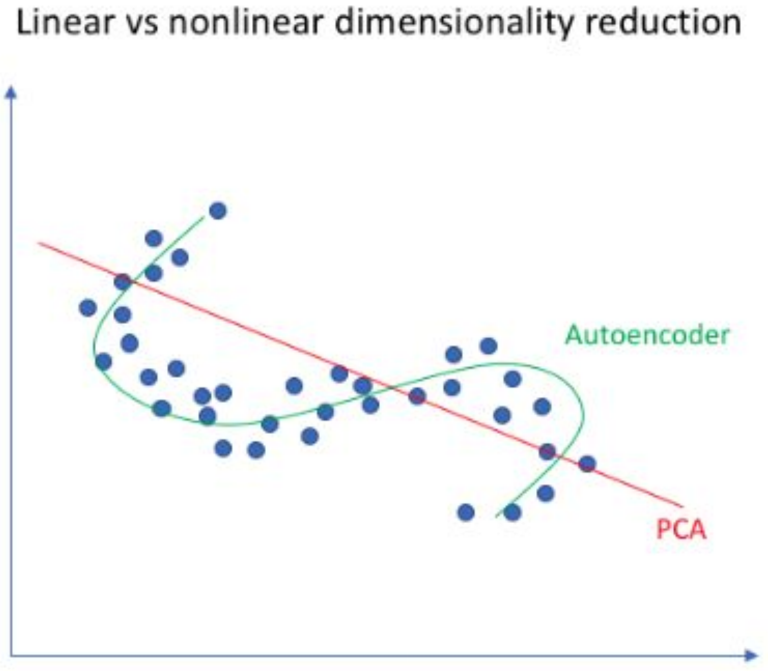
Reconstruction Error:

$$||x - x'||^2$$

Self-Supervised Pre-Training



Nonlinear Dimensionality Reduction



Jordan, J. (2018, March). Introduction to Autoencoders.
Retrieved from https://www.jeremyjordan.me/autoencoders/

Review: The Curse of Dimensionality

High dimensional spaces are **sparse**. We tend to have a low sampling density. The risk of overfitting is high. To illustrate this point:

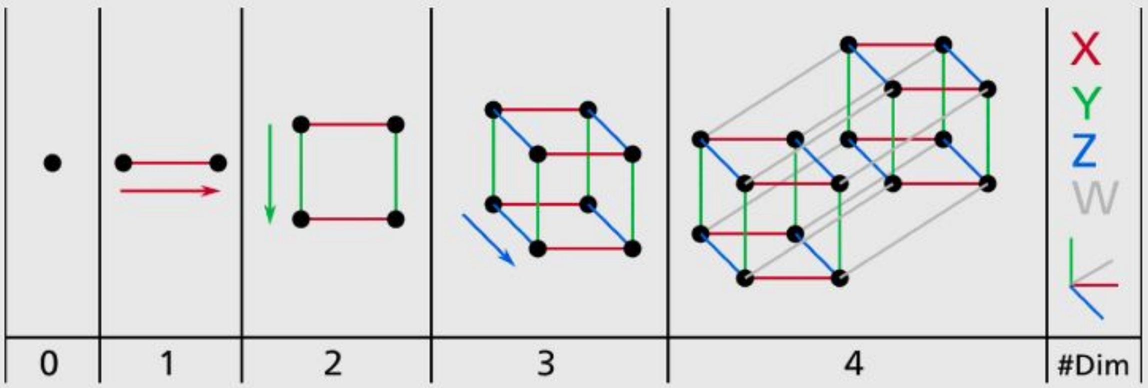


Figure 8-1. Point, segment, square, cube, and tesseract (0D to 4D hypercubes)[2]

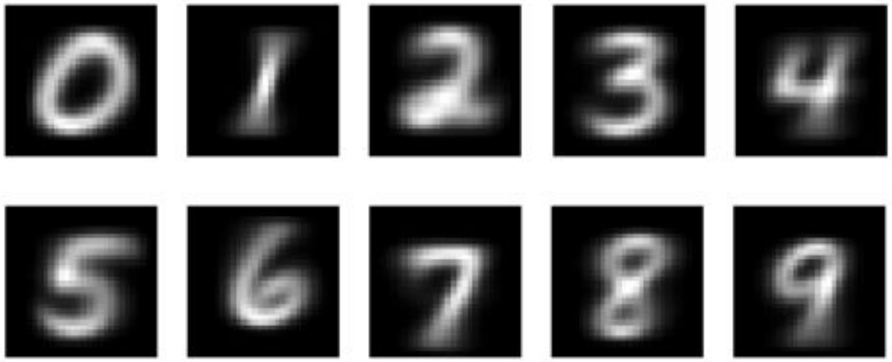
- If you pick two points randomly in a unit square, the distance between these two points will be, on average, roughly 0.52.
- If you pick two random points in a unit 3D cube, the average distance will be roughly 0.66.
- In a 1,000,000-dimensional hypercube? The average distance, believe it or not, will be about 408.25 (roughly $\sqrt{1000000/6}$)

SCHOOL OF CONTINUING STUDIES

The Manifold Hypothesis

In most real-world problems, **training instances are not spread out uniformly** across all dimensions. Rather, (i) **many features are almost constant**; and (ii) **others are highly correlated**.

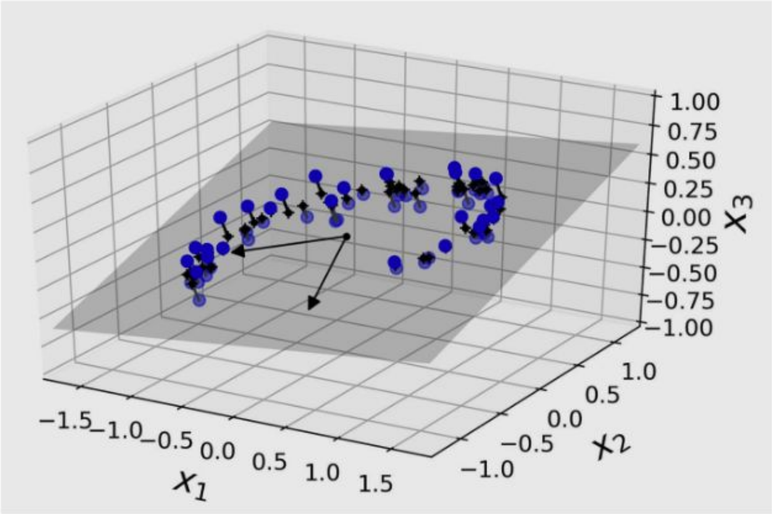
Example: mean pixel values of MNIST classes. Pixels around image borders are usually zero. If the class is 1, pixel values tend to be high near the x=0 axis, and low away from it.



The Manifold Hypothesis

… As a result, the **training instances tend to lie within (or close to) a much lower-dimensional subspace** of the original high dimensional space.

Example: 3D feature space



2D projection that preserves most of the information

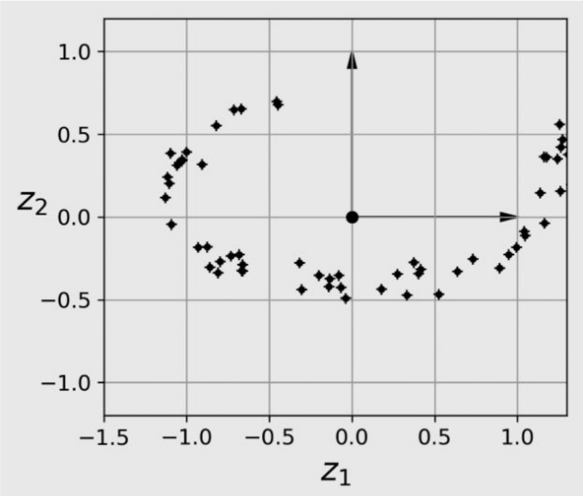


Figure 8-3. The new 2D dataset after projection

Goal: learn the subspace that best encodes our data!

UNIVERSITY OF TORONTO
SCHOOL OF CONTINUING STUDIES

Demo Time!

Module 8 - Section 3

Generative Models: Naive Bayes

→ ***See Jupyter Notebook***

Module 8 - Section 4

Variational Autoencoders

Motivation

- Naive-Bayes makes the assumption that the features are independent, which is clearly not true.
- The values of the latent vector elements that comprise a **plausible output** are not independent. They co-vary. Therefore, we need to learn a joint probability distribution over the coding, for each class, to ensure we are choosing vectors that make sense.
- But what are these distributions in latent space, and what region of feature space should we sample over?
- <u>Idea</u>: What if we could **force** these distributions to be Gaussian distributions? Not only for each class, but also for the entire latent space? If we knew that the latent space was a Unit Gaussian, then we'd already know how to sample it in a way that produces plausible outputs.

Predicting a Latent Vector

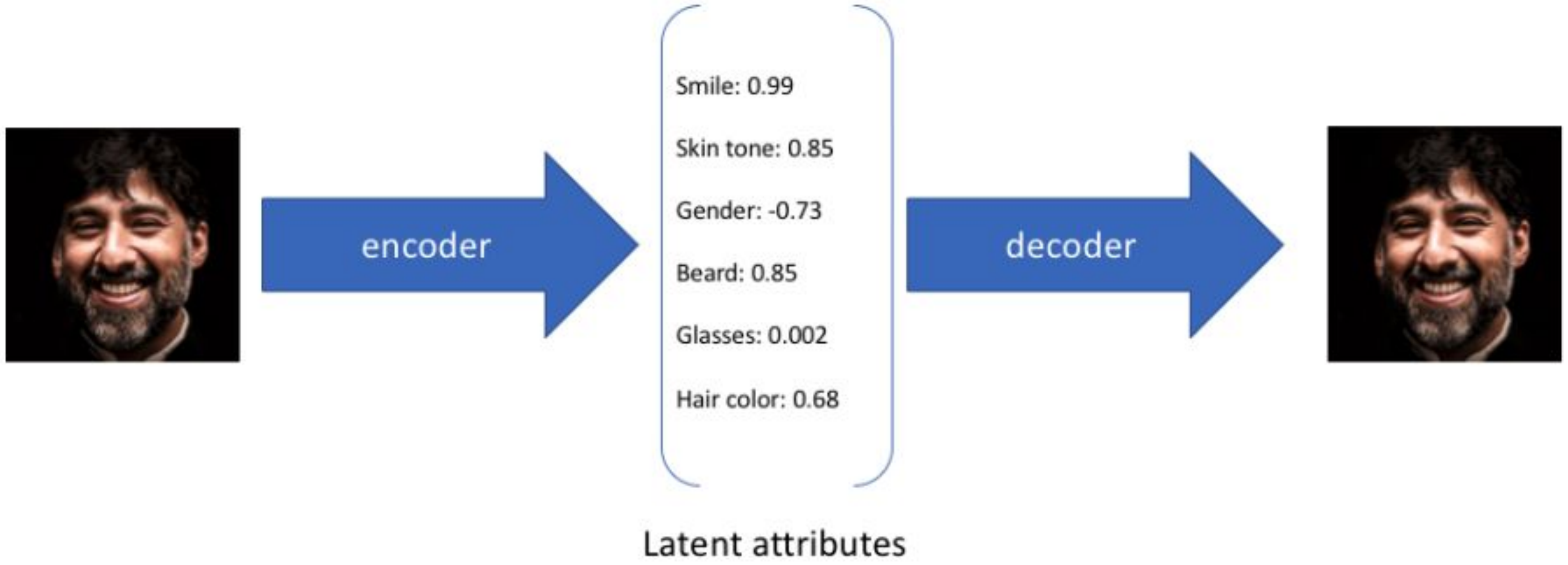


Image Source: https://www.jeremyjordan.me/variational-autoencoders/

Predicting a Distribution

μ = mean
σ = variance

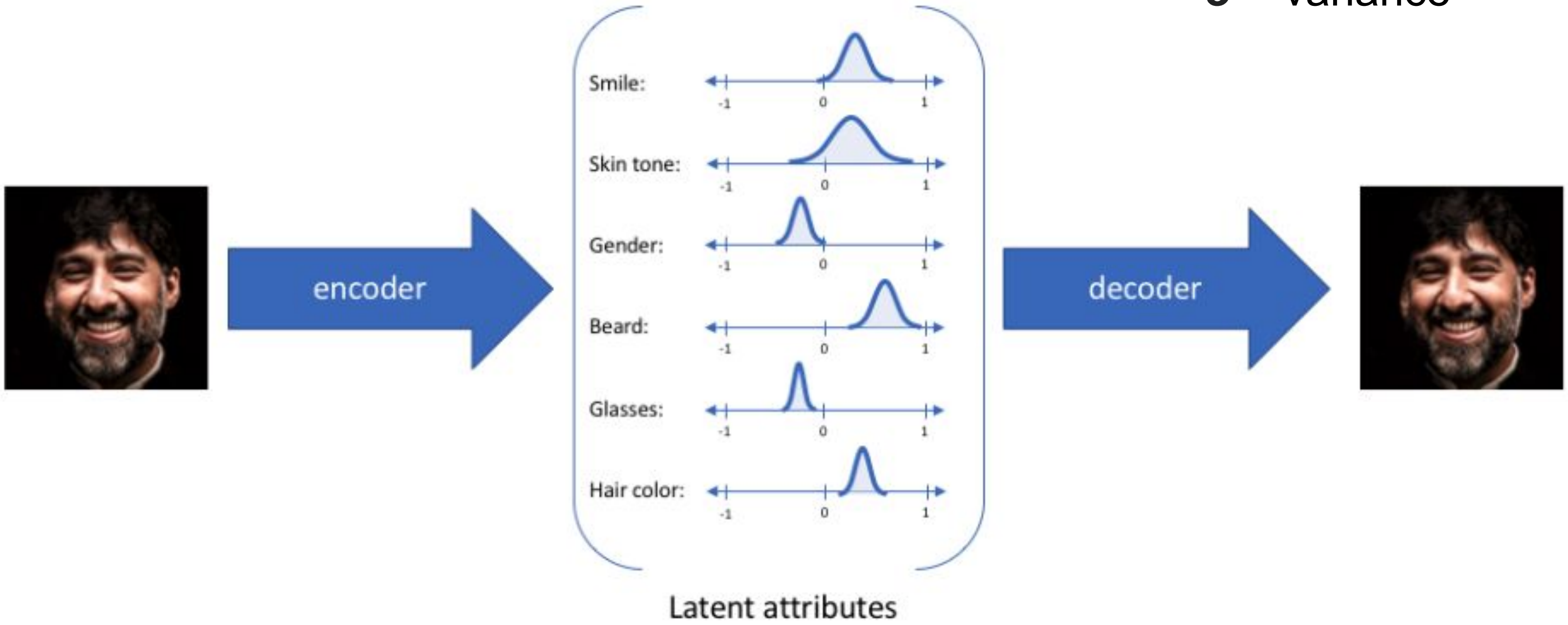


Image Source: https://www.jeremyjordan.me/variational-autoencoders/

Sampling from a Distribution

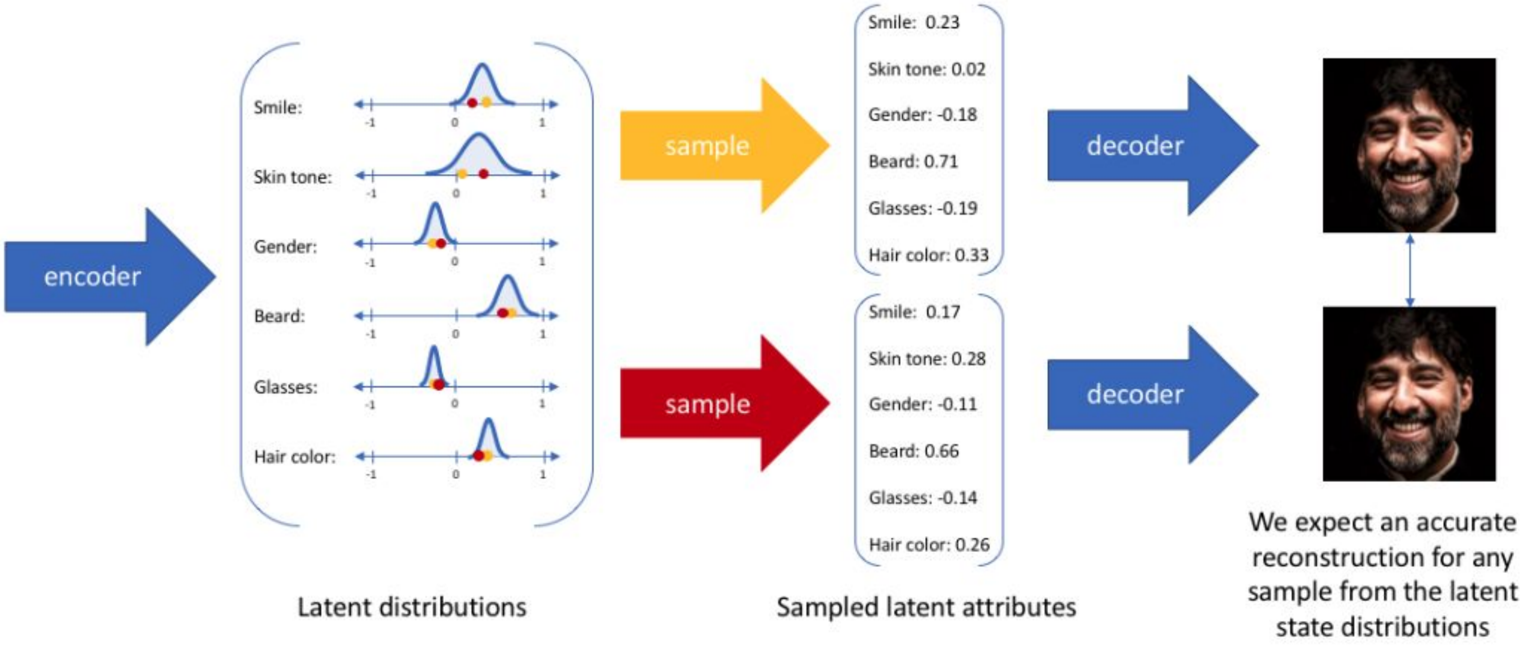
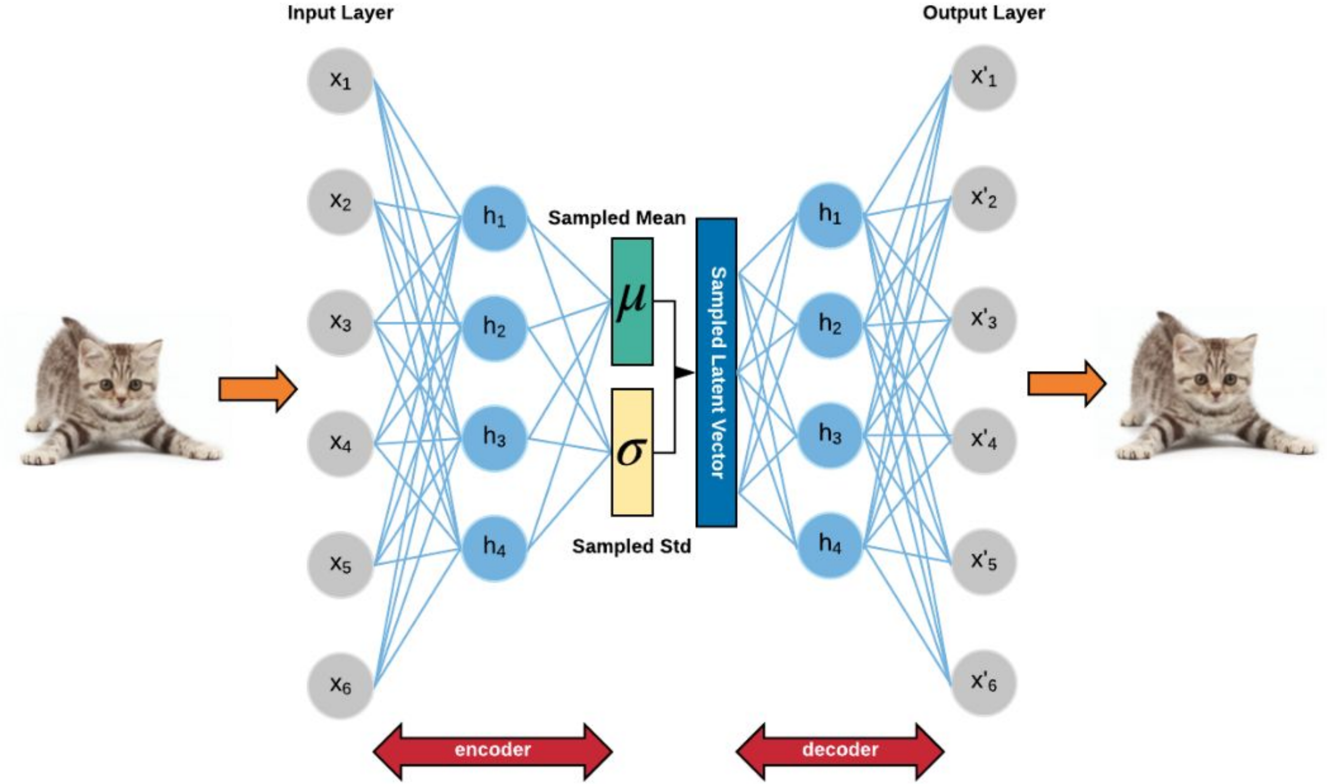


Image Source: https://www.jeremyjordan.me/variational-autoencoders/

Variational Autoencoder (VAE)



UNIVERSITY OF T
SCHOOL OF CONTINUIN

Each Sample Maps to a Gaussian

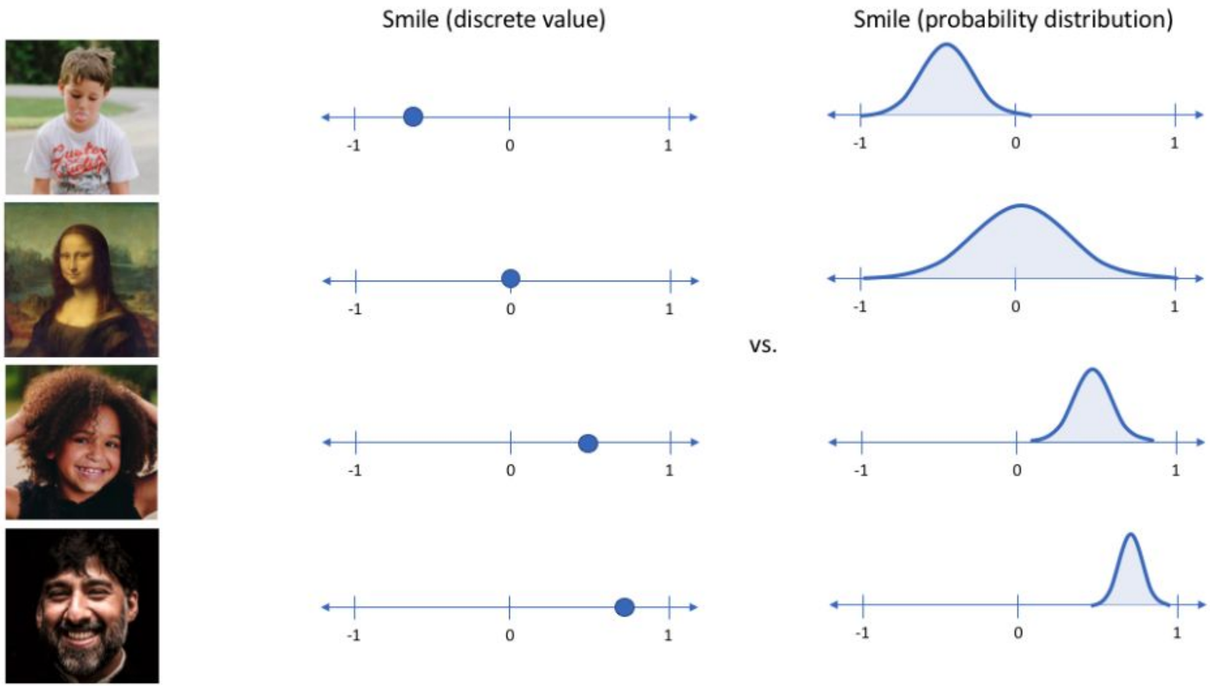
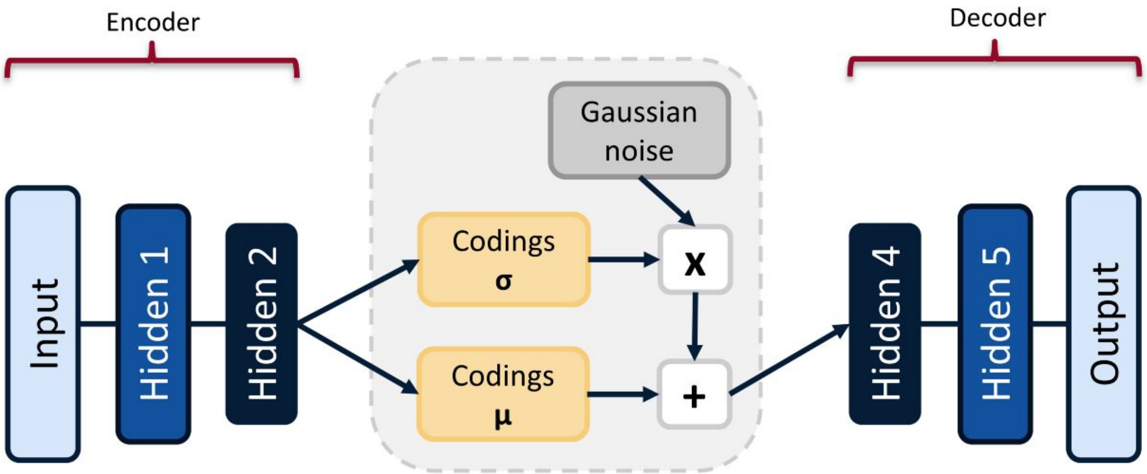
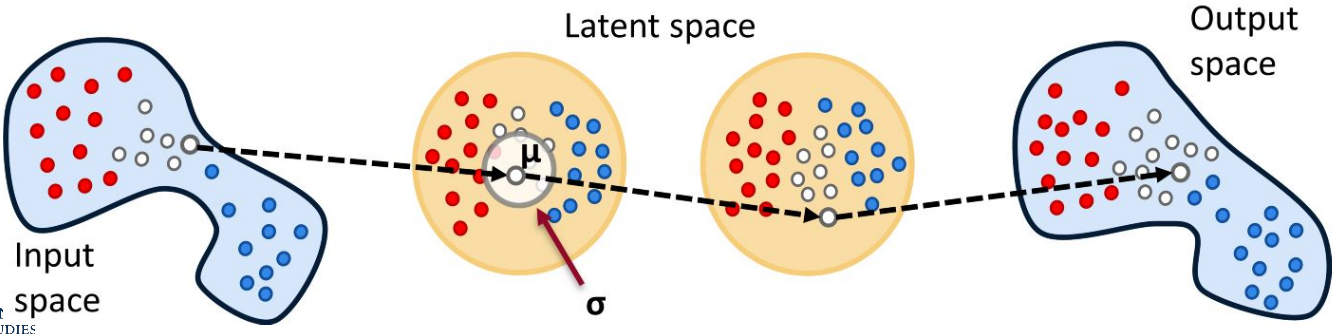


Image Source: https://www.jeremyjordan.me/variational-autoencoders/

Latent Space is also Gaussian





Constraining the Distributions

For discrete probability distributions P and Q defined on the same probability space, the ***Kullback–Leibler divergence*** *from Q to P is defined to be:*

$$D_{\mathrm{KL}}(P\|Q) = -\sum_i P(i)\,\log\left(\frac{Q(i)}{P(i)}\right)$$

Total Loss:

$$L_{reconst} = \|x - x'\|^2$$
$$L_{latent} = D_{KL}(z\|\mathcal{N}(0,1))$$
$$Loss = L_{reconst} + L_{latent}$$

Image Source: https://www.jeremyjordan.me/variational-autoencoders/

Composite Loss Function

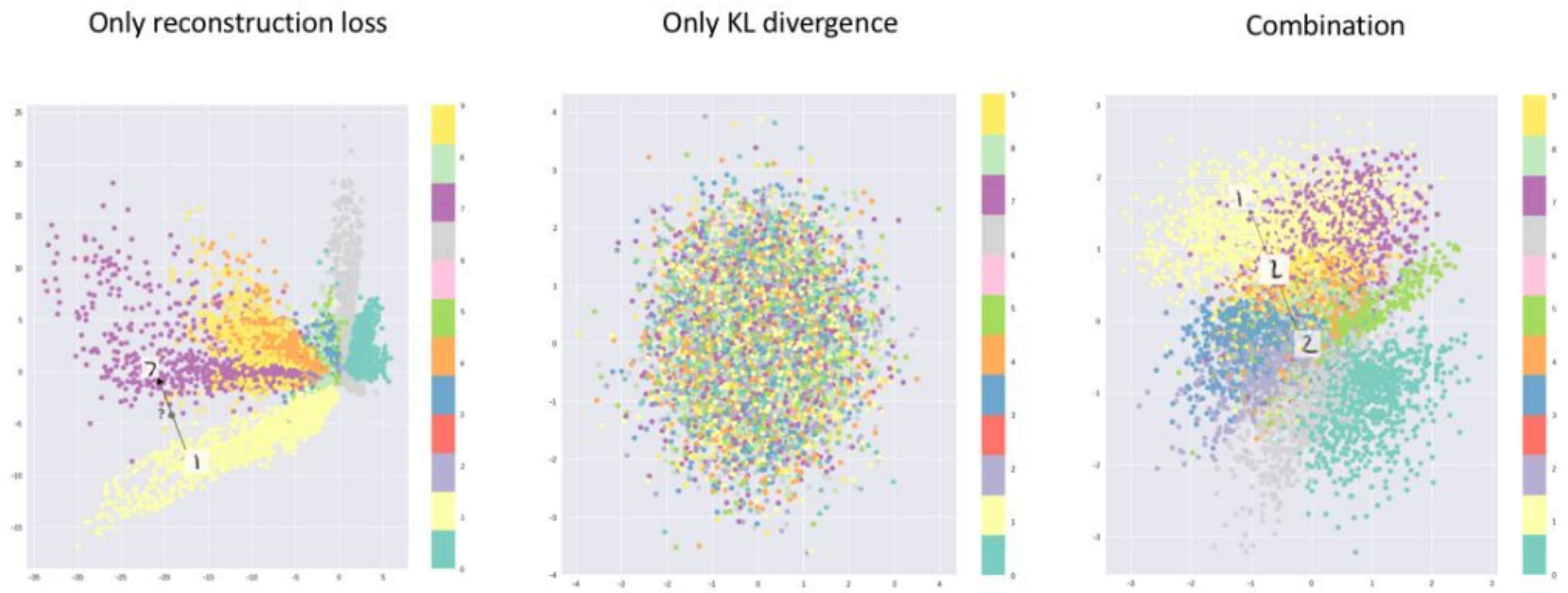


Image Source: https://www.jeremyjordan.me/variational-autoencoders/

Encodings without KL Penalty

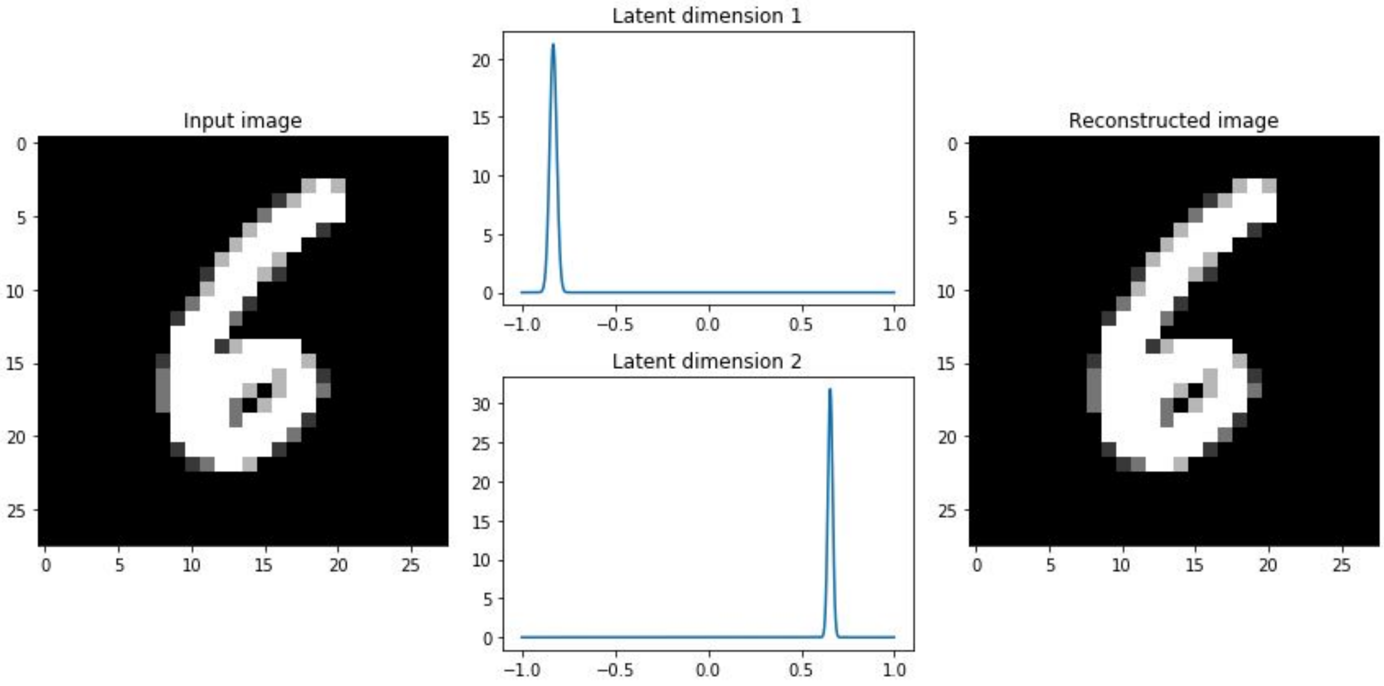


Image Source: https://www.jeremyjordan.me/variational-autoencoders/

Composite Loss Function

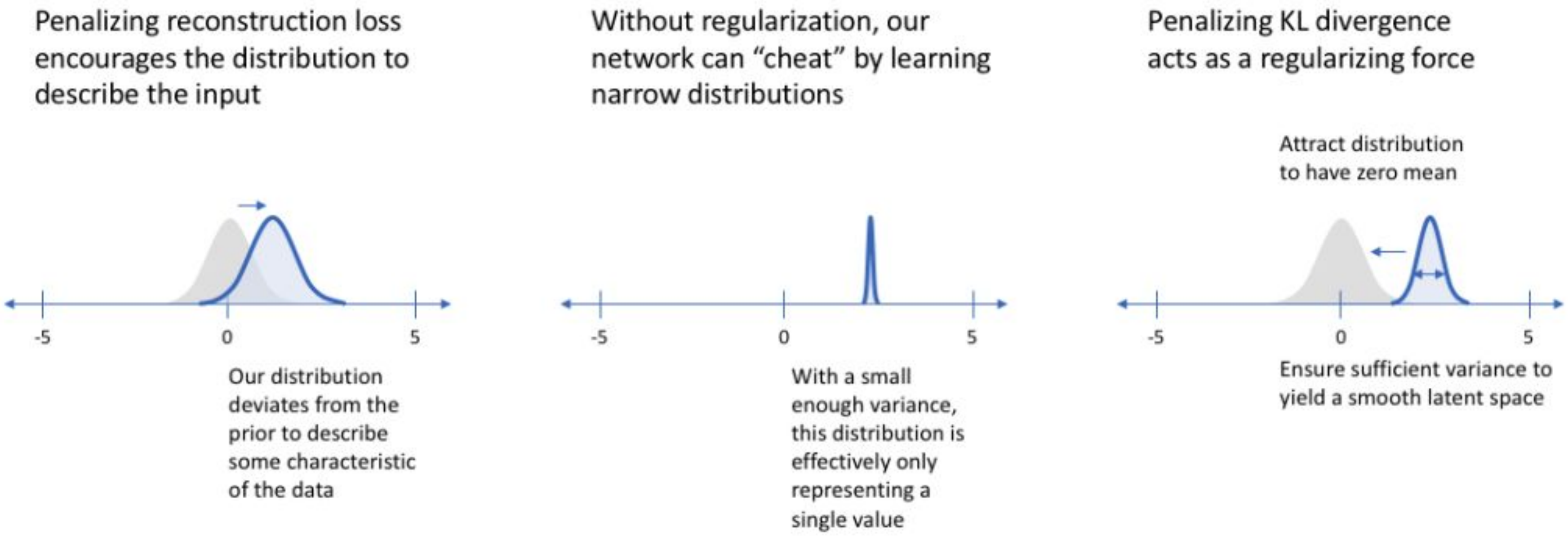


Image Source: https://www.jeremyjordan.me/variational-autoencoders/

Implementation

(Shown for a 2-dimensional latent space)

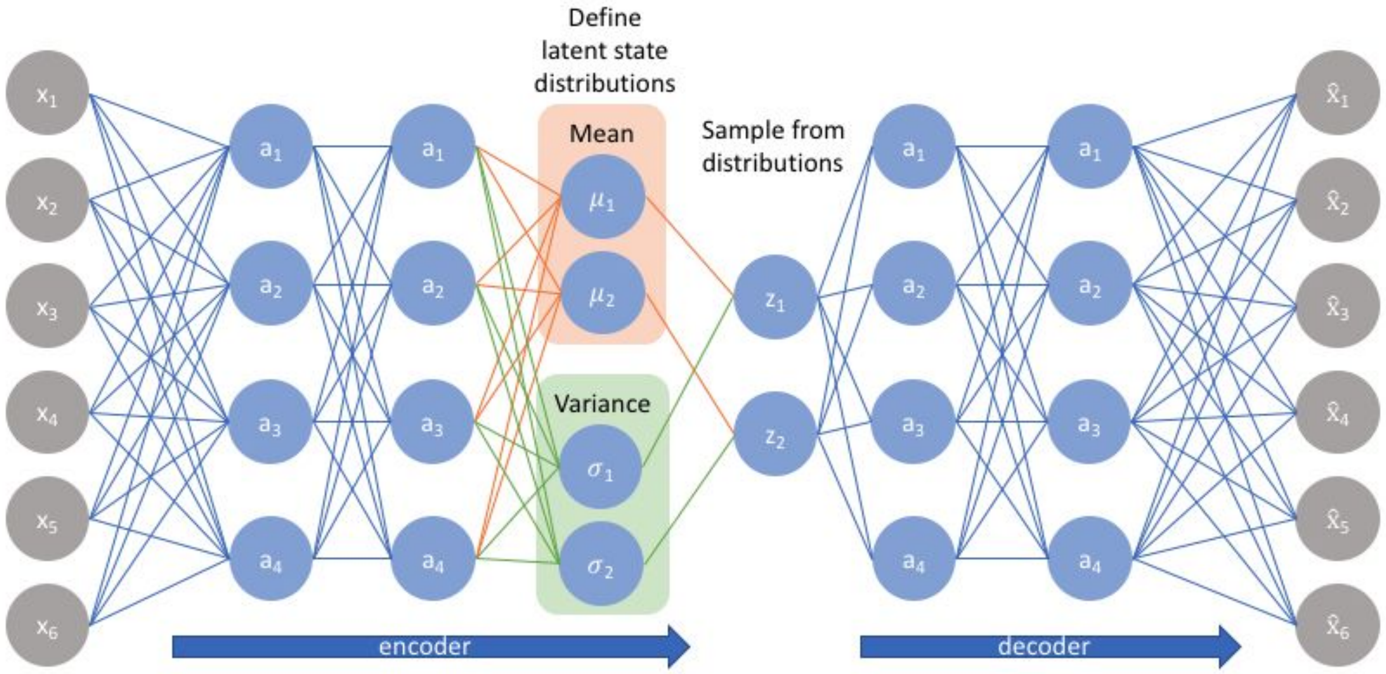


Image Source: https://www.jeremyjordan.me/variational-autoencoders/

UNIVERSITY OF TORONTO
SCHOOL OF CONTINUING STUDIES

Module 8

Resources and Wrap-up

Homework

- Review Module 8 notebook.
- Continue Assignment 3; due next week!
- Final Project topic should be chosen. Start working on it!

Next Class

- We’ll be covering Deep Generative Models, including Generative Adversarial Models (GANs), which are behind many state-of-the-art media synthesis approaches.
- Recommended: Read Géron Chapter 17, page 592 and onwards, to prep.

UNIVERSITY OF TORONTO
SCHOOL OF CONTINUING STUDIES

Any questions?

UNIVERSITY OF TORONTO
SCHOOL OF CONTINUING STUDIES

Thank You

Thank you for choosing the University of Toronto School of Continuing Studies