Autoencoders

Forest Agostinelli
University of South Carolina

Outline

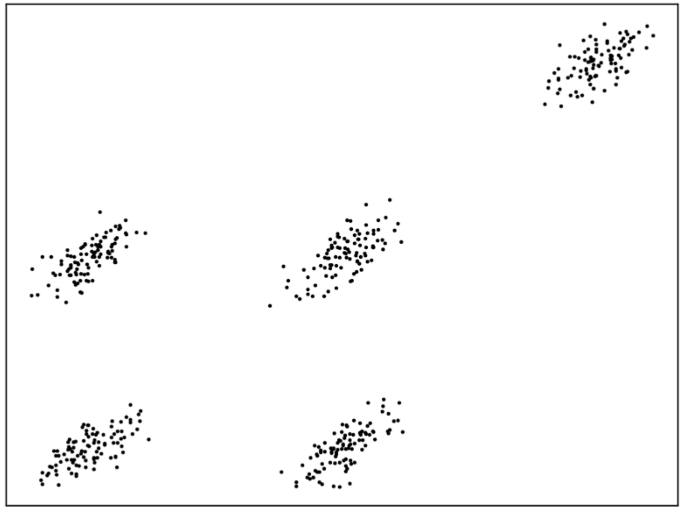
- Clustering
- Autoencoders
- Variational autoencoders
- Conditional variational autoencoders

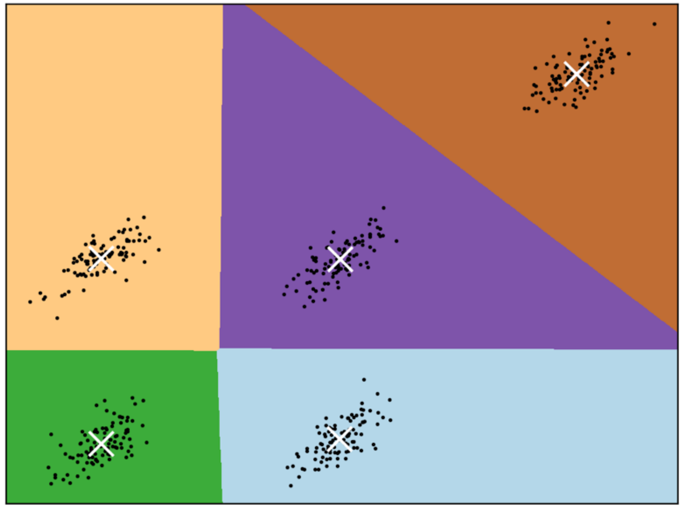
Motivating Example

- Imagine you are an alien species that comes across the MNIST dataset that only has images, but no corresponding labels
- You suspect that there is some structure to this data, and you hope to discover it
- Furthermore, you hope to generate data similar to it to imitate humans

Clustering

- Grouping objects into clusters where objects in a cluster are more similar than compared to those in other clusters
- Natural sciences
  - High-energy physics
  - Biology
  - Chemistry
- Medicine
  - Patients
  - Diseases
- Reinforcement learning
  - Cluster similar states for hierarchy
  - Cluster similar actions to create meta-actions



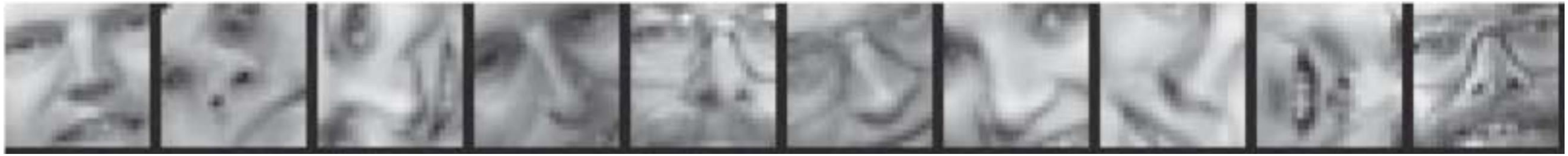


Curse of Dimensionality

- The phrase was coined by Richard Bellman in reference to solving problems with dynamic programming
- However, this is relevant to many other cases
- In high-dimensions, data has many possibly surprising properties
- In particular, data points tend to be sparse when the dimensionality is increased
  - Euclidean distance becomes less meaningful
  - This makes partitioning data into meaningful clusters difficult or impossible

Curse of Dimensionality: Examples

- Suppose we have a, relatively small, 28 x 28 images
  - There are $28 \times 28 = 784$ –dimensional data points
  - Running K-means on this data will most likely result in meaningless clusters

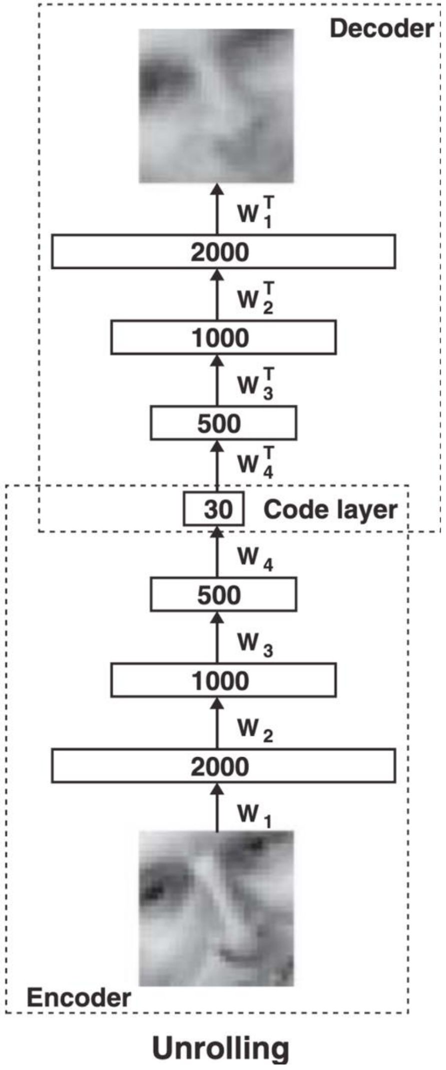


Outline

- Clustering
- Autoencoders
- Variational autoencoders
- Conditional variational autoencoders

Autoencoders

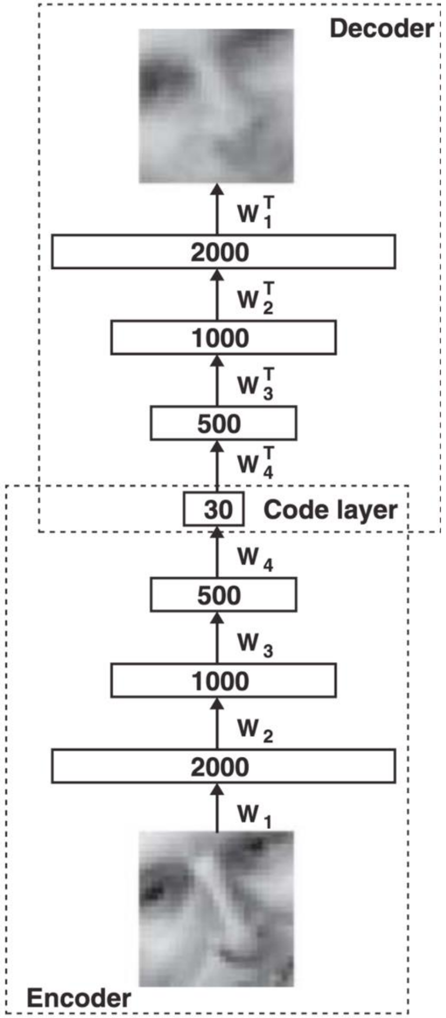
- Neural networks that are trained without labels
- The input is passed through an encoder
  - The dimensionality of the output of the encoder is usually much less than the dimensionality of the input
  - Called code layer or bottleneck layer
- The output of the encoder is then passed to the decoder which is trained to mimic the input
- This is known as minimizing the reconstruction error



Hinton, Geoffrey E., and Ruslan R. Salakhutdinov. "Reducing the dimensionality of data with neural networks." science 313.5786 (2006): 504-507.

Autoencoders

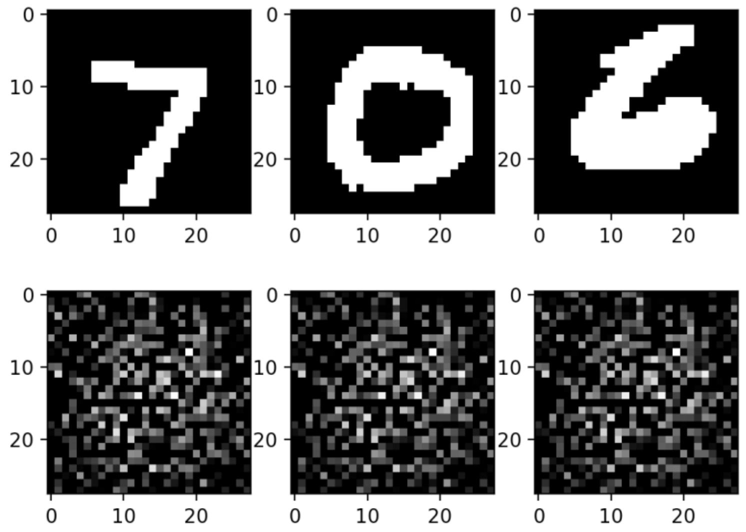
- The larger the code layer, the better the reconstruction error tends to be
  - However, the autoencoder may then start capturing features that are irrelevant to the relevant structure of the data, such as background data
- We can use this autoencoder to dimensionally reduce the MNIST dataset and then do clustering on it



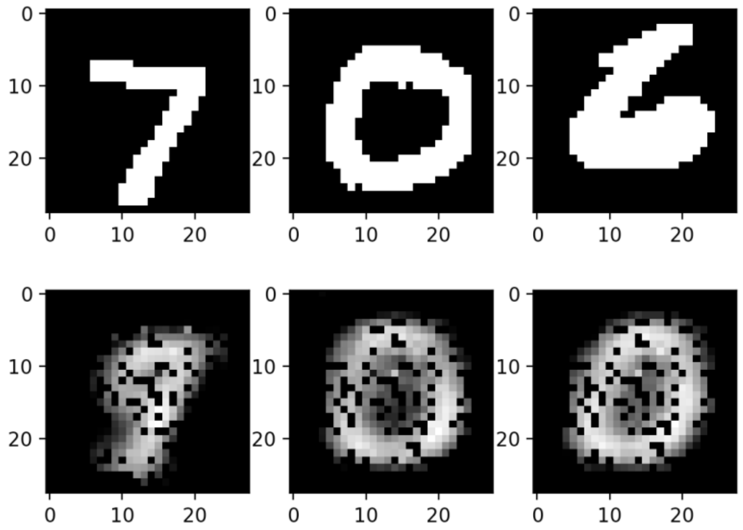
Unrolling

Autoencoders

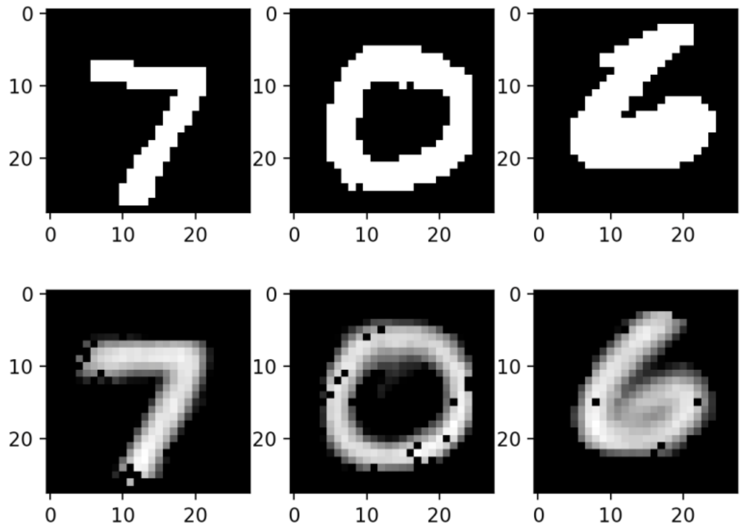
- The reconstructed input initially looks nothing like the input image
- However, after training, the output starts to match the input



Initialization



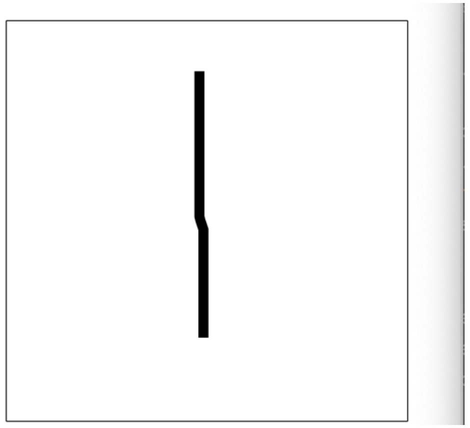
100 iterations

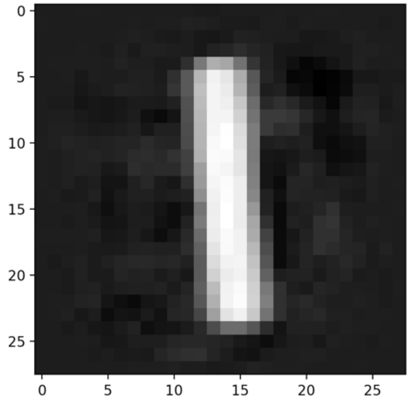


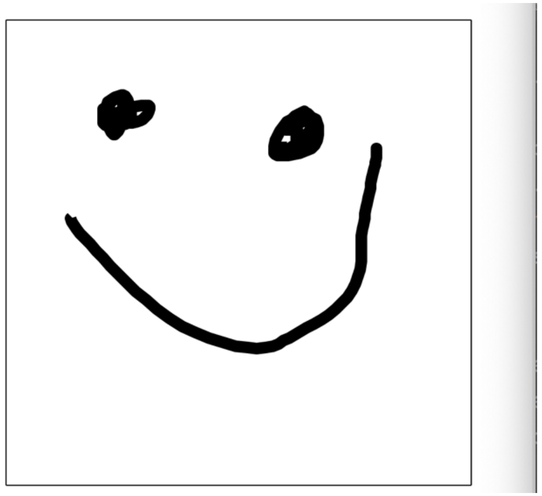
10,000 iterations

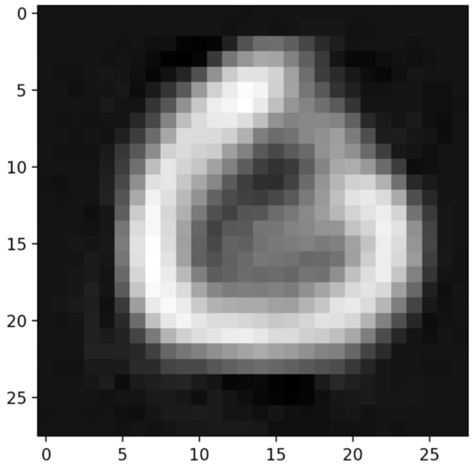
Out of Distribution Data

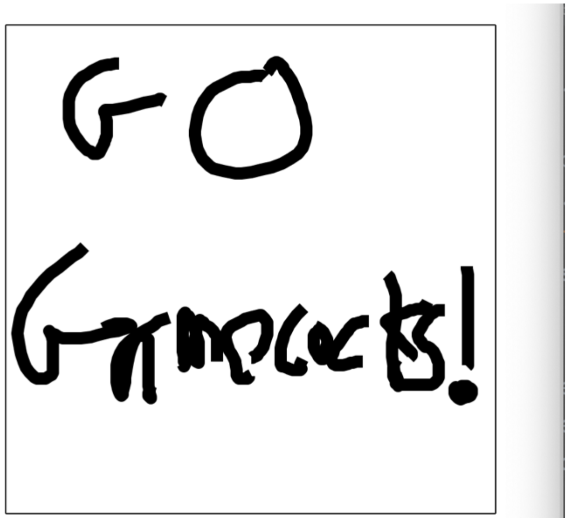
- As we saw in supervised learning, when testing on data that is significantly different than the training data, the performance of the neural network may be much worse

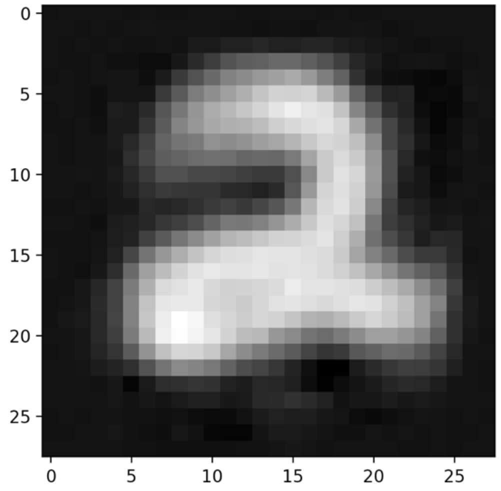






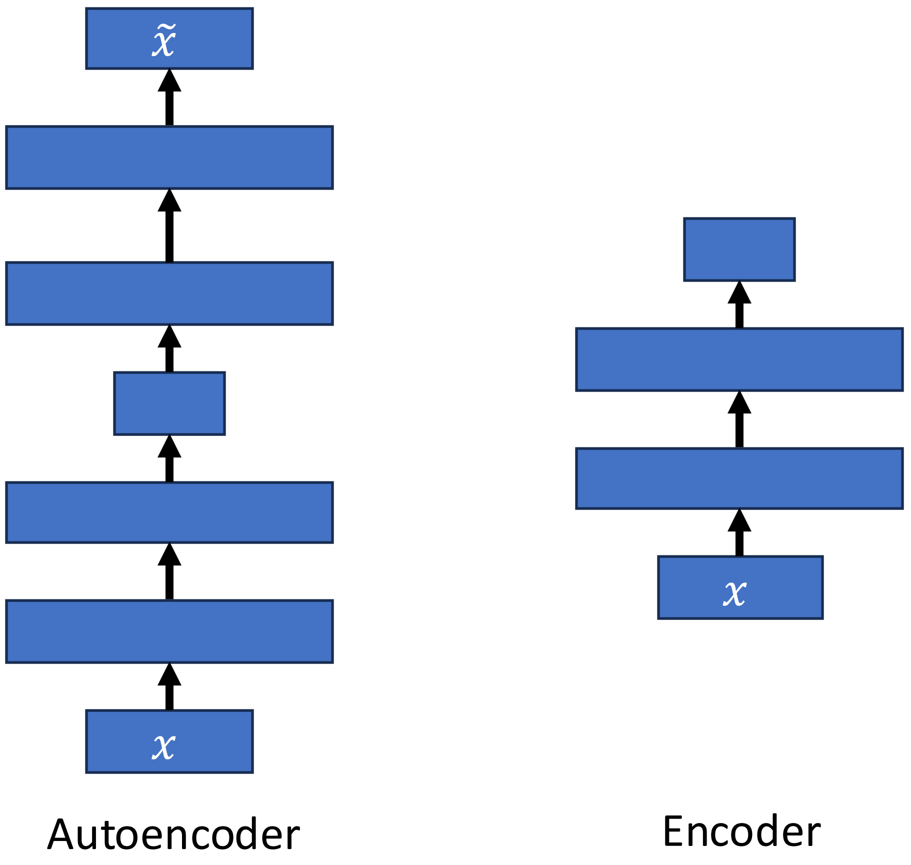






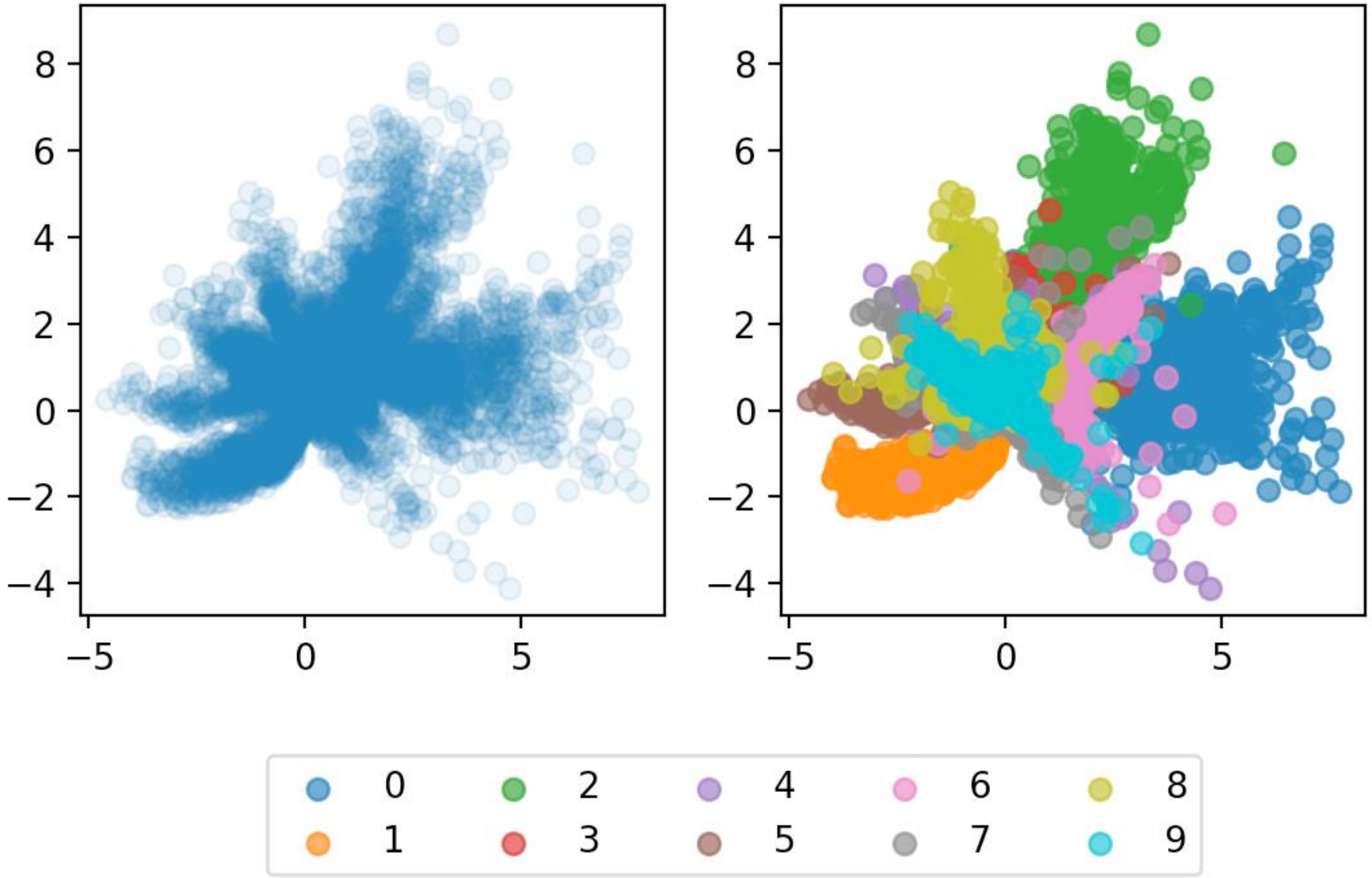
Dimensionality Reduction

- An autoencoder is composed of an encoder and decoder
- We can encoder inputs into a space of a smaller dimension by just putting data through the encoder and obtaining its output



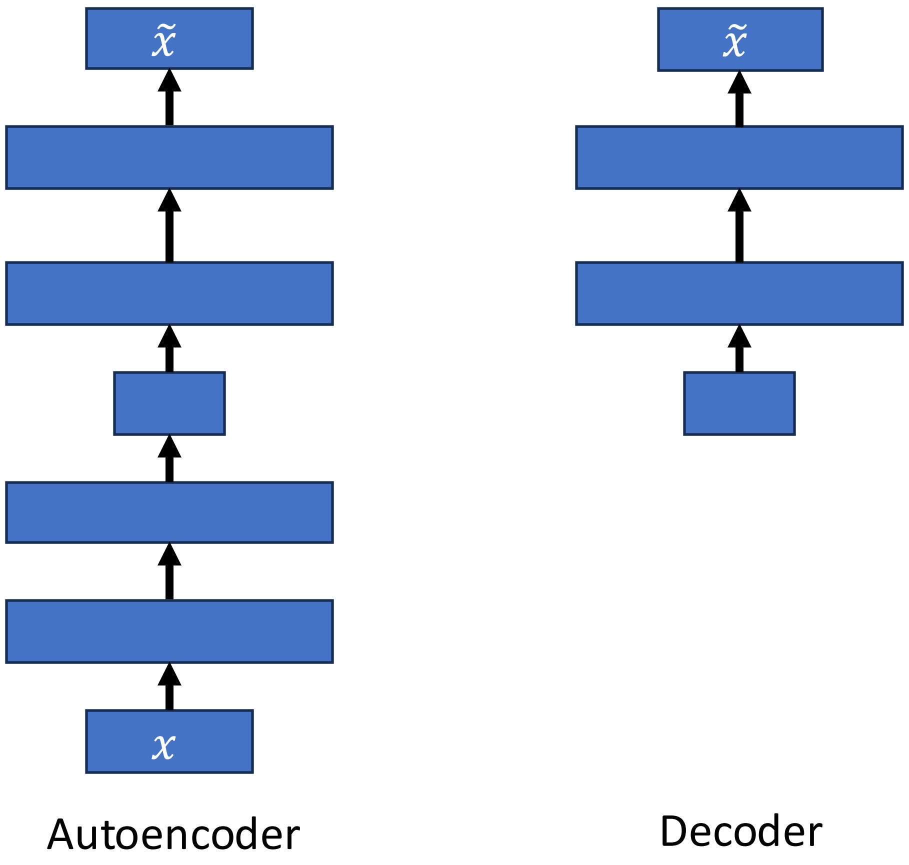
Dimensionality Reduction Results

- When visualizing our trained autoencoder with a bottleneck size of 2, we can see that there appear to be clusters
- When adding in label information, we see that similar digits do, indeed, cluster together
- How can we generate new data with an autoencoder?



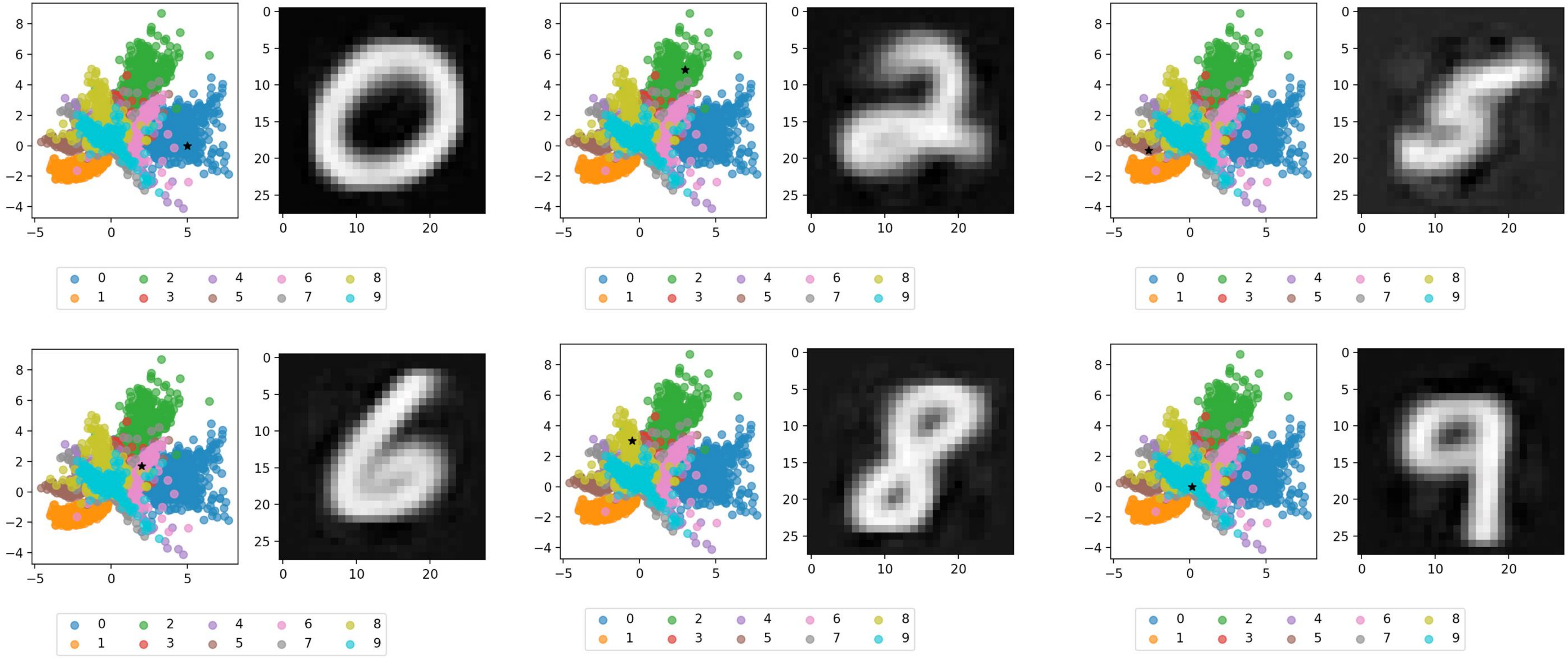
Generating New Data

- An autoencoder is composed of an encoder and decoder
- One can take data in the encoded space and put it though the decoder to generate new data



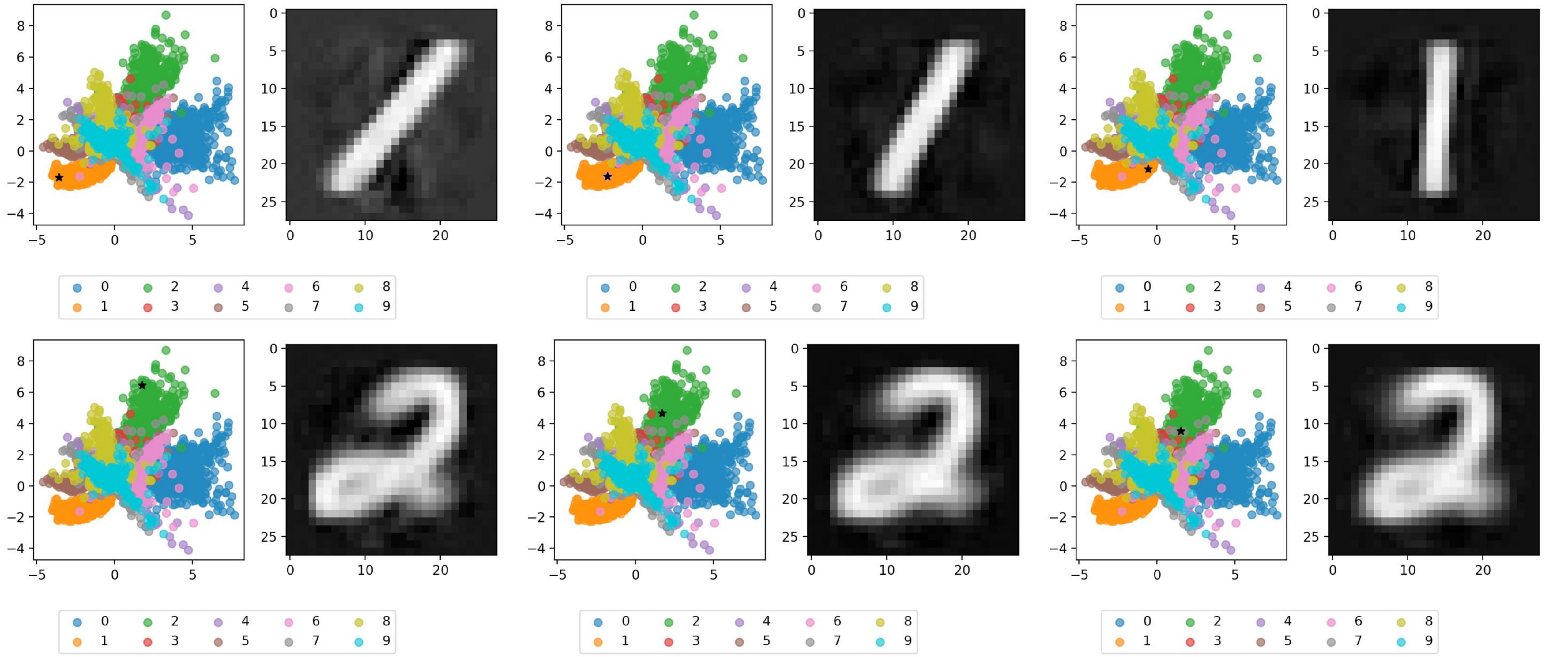
Generating New Data

- If we sample from data in distribution, we usually generate reasonable images



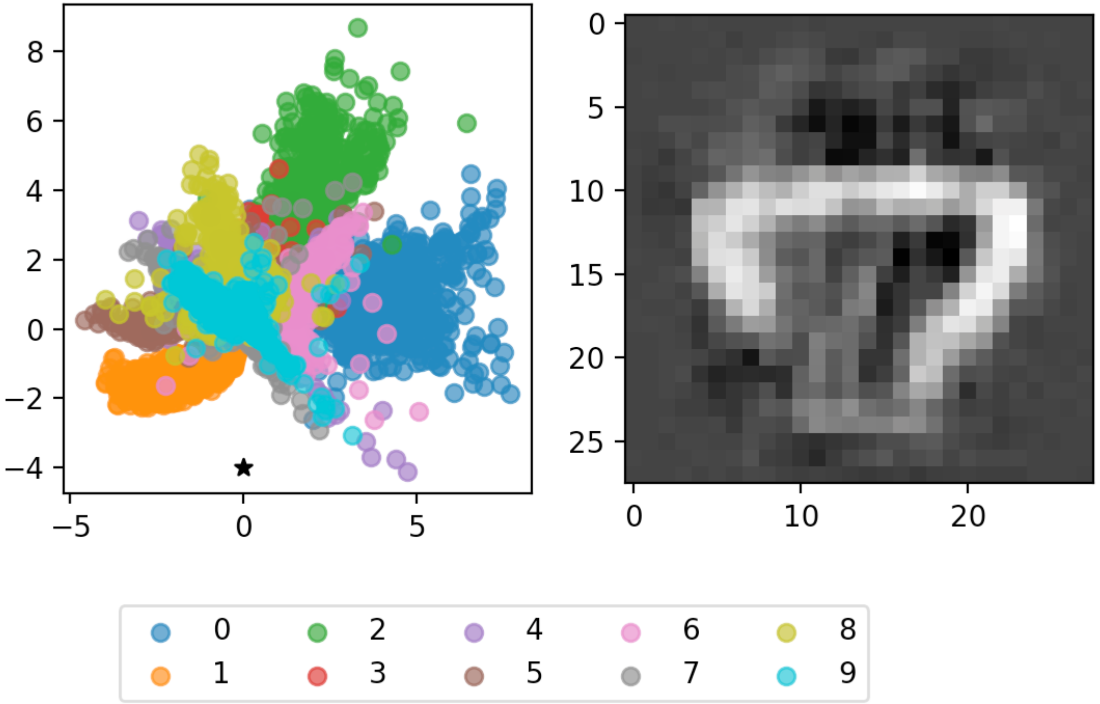
Traversing the Latent Manifold

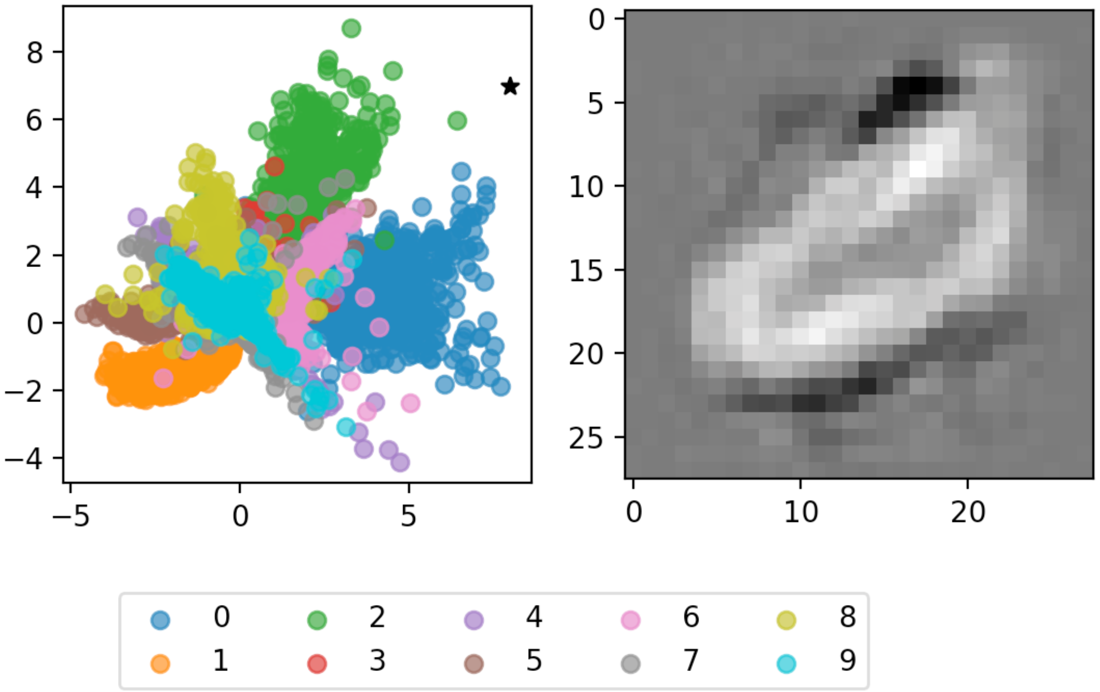
- We can see how digits change by sampling different points in the latent space that are related to that digit



Generating Data: Out of Distribution Data

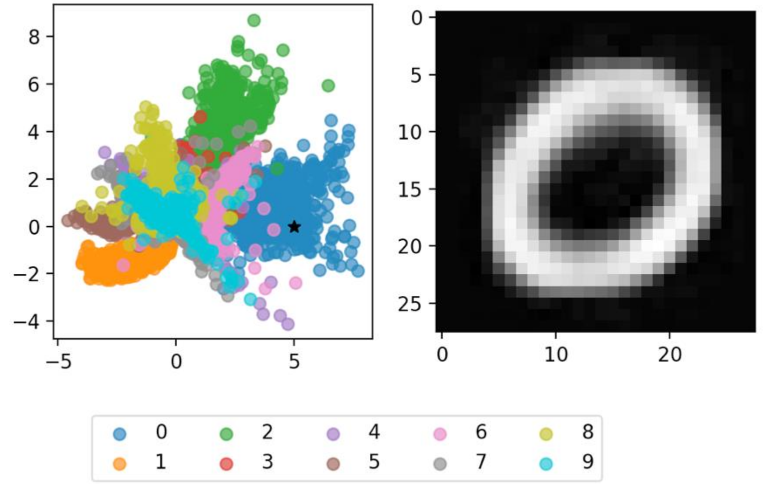
- Sampling from a point in the encoded space that the decoder has not observed during training can generate data that does not look like it comes from the original data distribution
  - Compare to silly ChatGPT outputs
- How can we ensure that we are only sampling data that is in distribution?



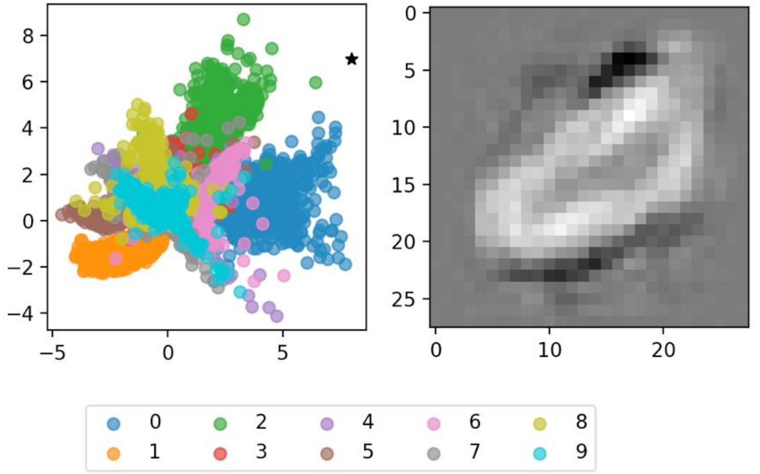


Distributions

- To generate data, we sample data from the latent space and put it through the decoder
- One of the difficulties of generating data is characterizing the distribution of the latent space
  - Gets much harder in higher dimensional spaces
- If we could accurately characterize the distribution, then we could more reliably sample realistic data



In distribution



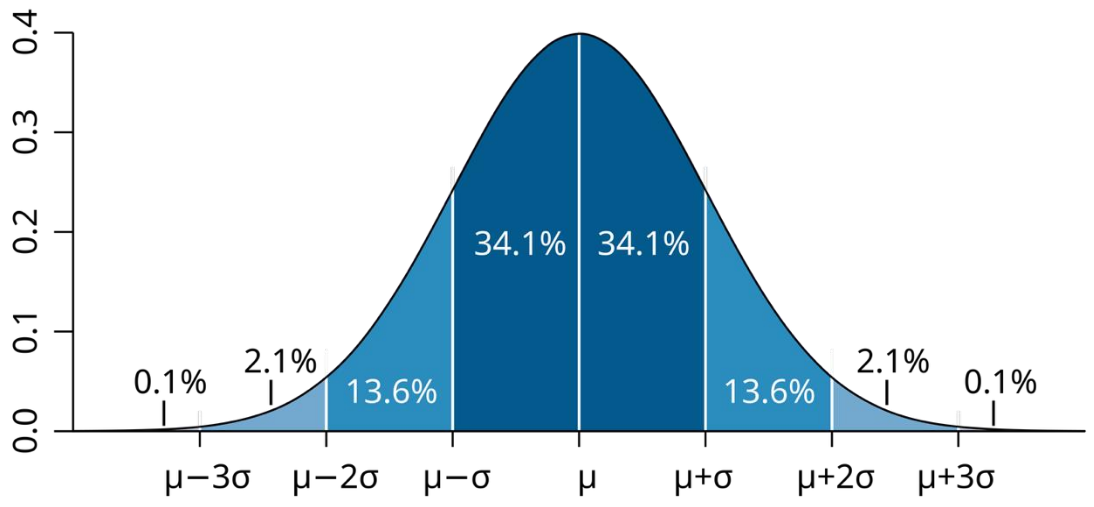
Out of distribution

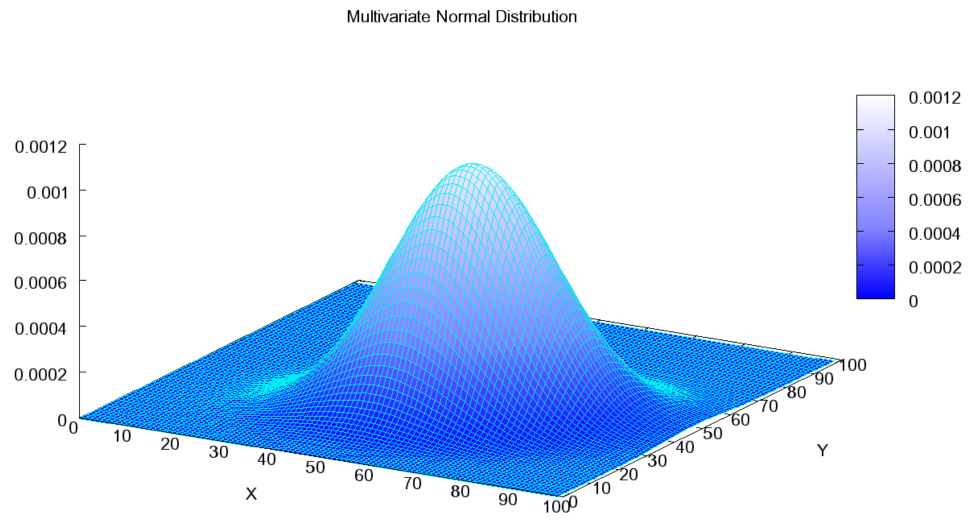
Outline

- Clustering
- Autoencoders
- Variational autoencoders
- Conditional variational autoencoders

Distribution

- A distribution (a probability distribution, in specific) is a function that maps outcomes to the probability that they occur
- One of the most common examples of this is the normal distribution (also known as the Gaussian distribution)

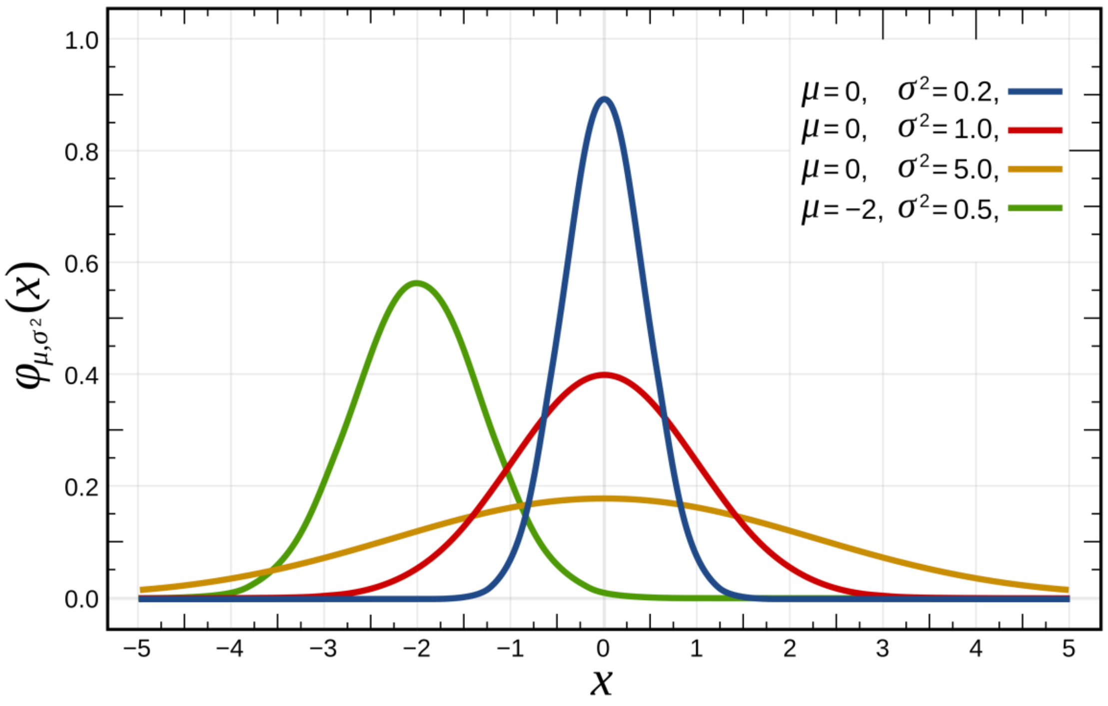




https://en.wikipedia.org/wiki/Probability_distribution
https://en.wikipedia.org/wiki/Normal_distribution
https://en.wikipedia.org/wiki/Multivariate_normal_distribution

Normal Distribution

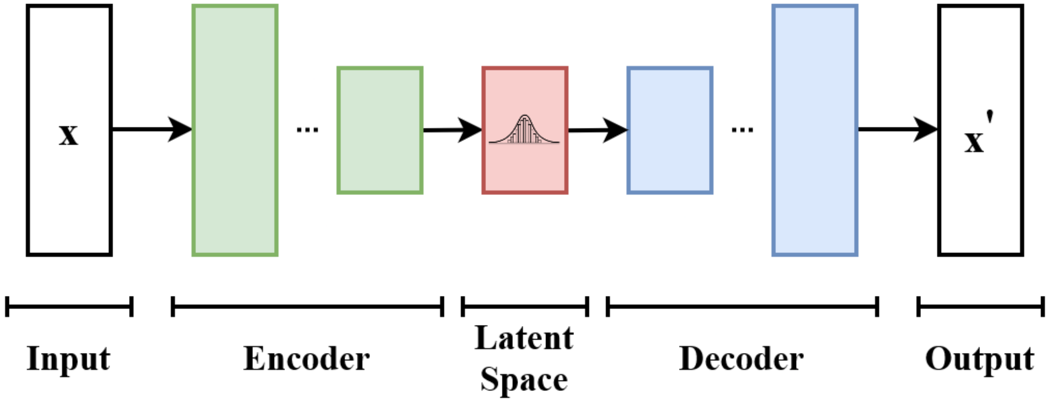
- The normal distribution is characterized by a mean (denoted μ) and standard deviation (denoted σ)
- A standard normal distribution is one with mean zero and standard deviation 1
- If we know the latent space follows a standard normal distribution, sampling from such a distribution becomes easy
- How can we ensure the latent space follows such a distribution?



By Inductiveload - self-made, Mathematica, Inkscape, Public Domain, https://commons.wikimedia.org/w/index.php?curid=3817954

Variational Autoencoders

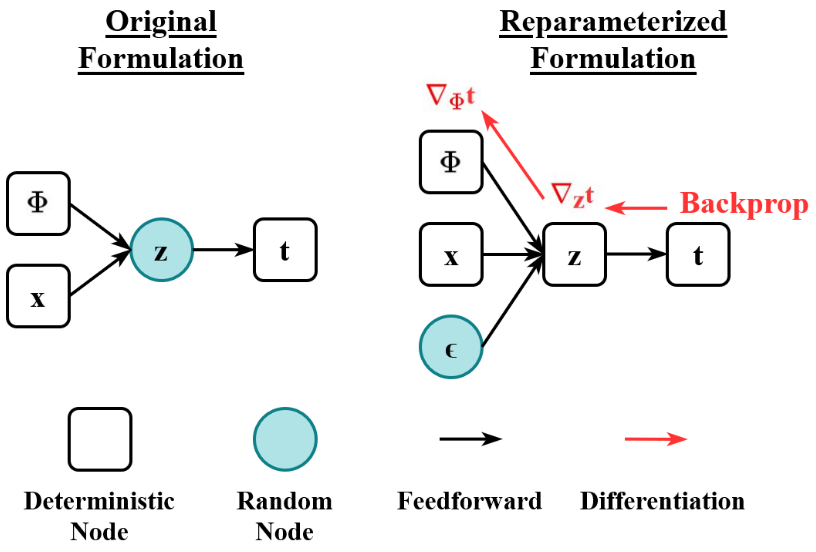
- Instead of mapping inputs to a point in a latent space, variational autoencoders map inputs to a distribution in the latent space
- They enforce a particular distribution on this latent space, which is usually a standard normal distribution

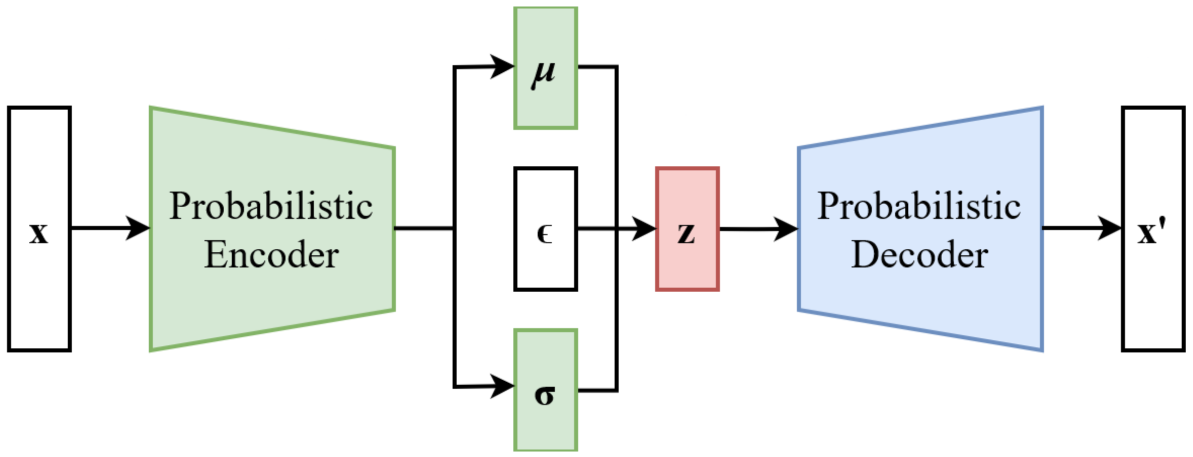


Kingma, Diederik P. "Auto-encoding variational bayes." arXiv preprint arXiv:1312.6114 (2013).

Reparameterization Trick

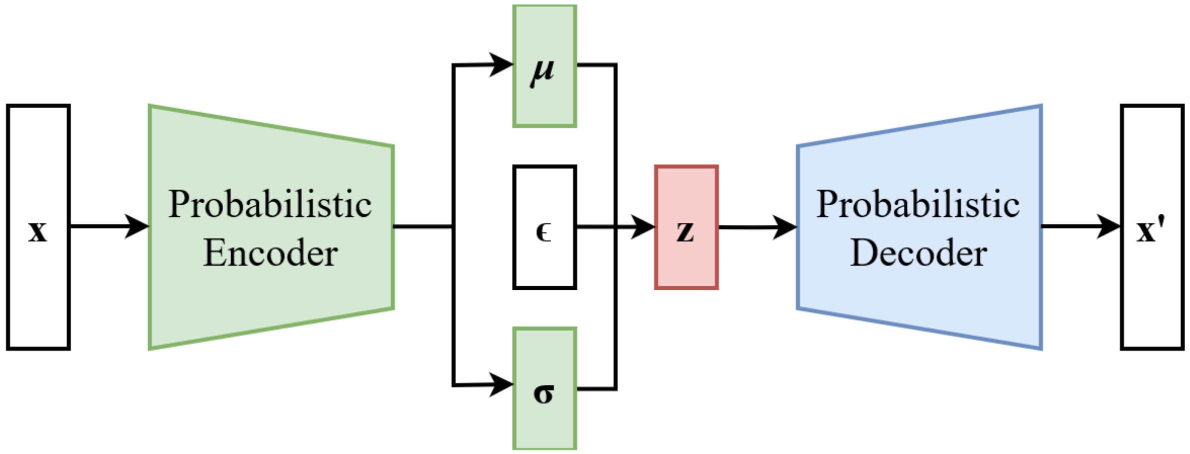
- We cannot directly map inputs to a distribution since we cannot backpropagate through randomness
- Therefore, we map inputs to the mean and variance, and combine these parameters with randomness to encode the input to a distribution





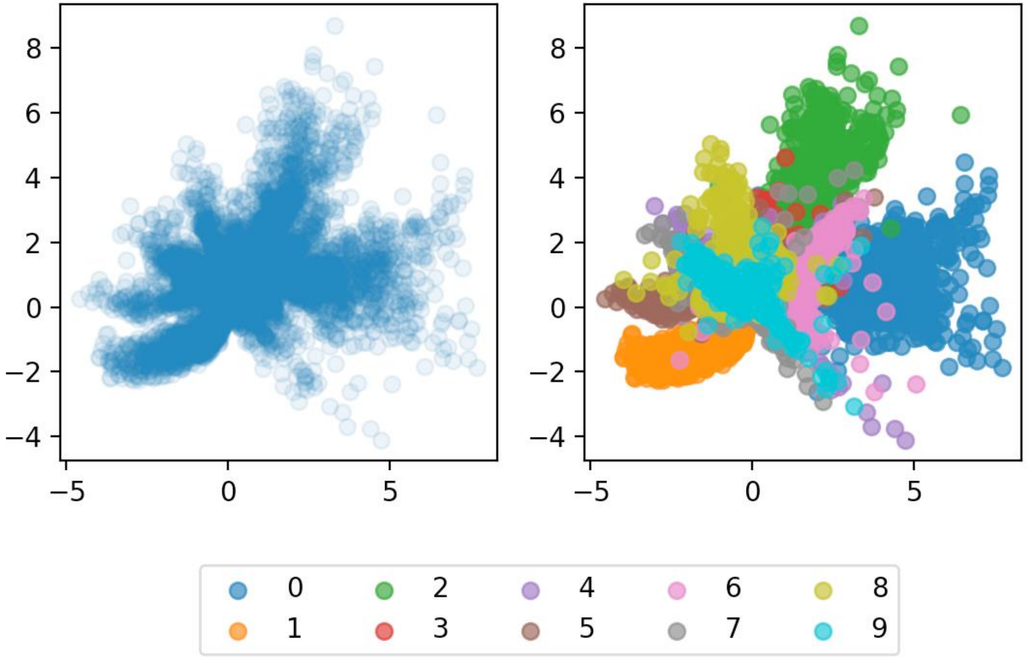
KL Divergence

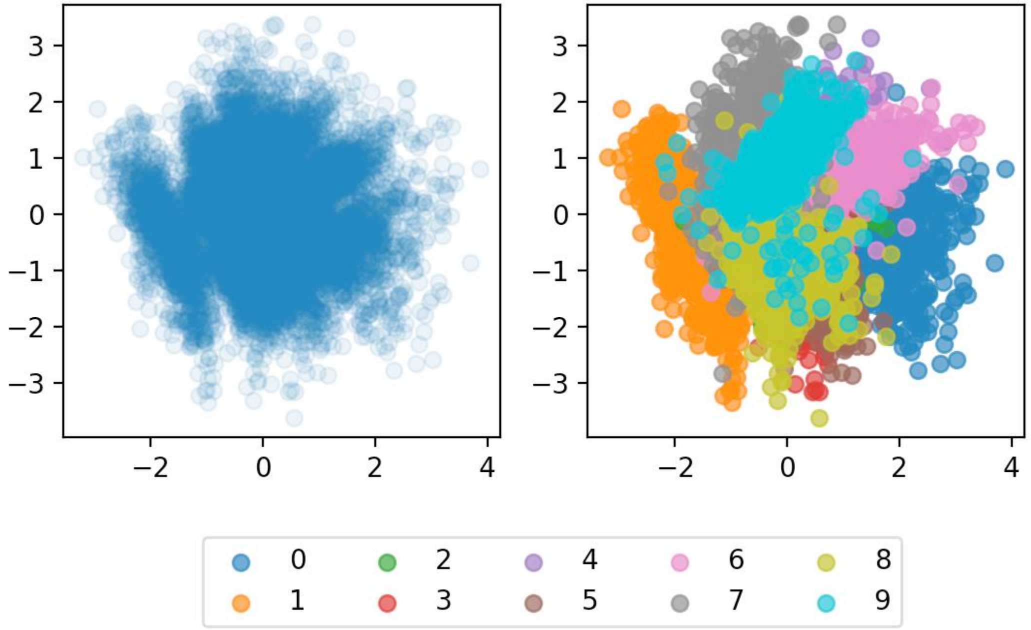
- The Kullback-Leibler (KL) divergence is a measure of similarity between two different distributions
- We can add the KL divergence to the loss function to encourage the latent space to follow a standard normal distribution



VAE Dimensionality Reduction Results

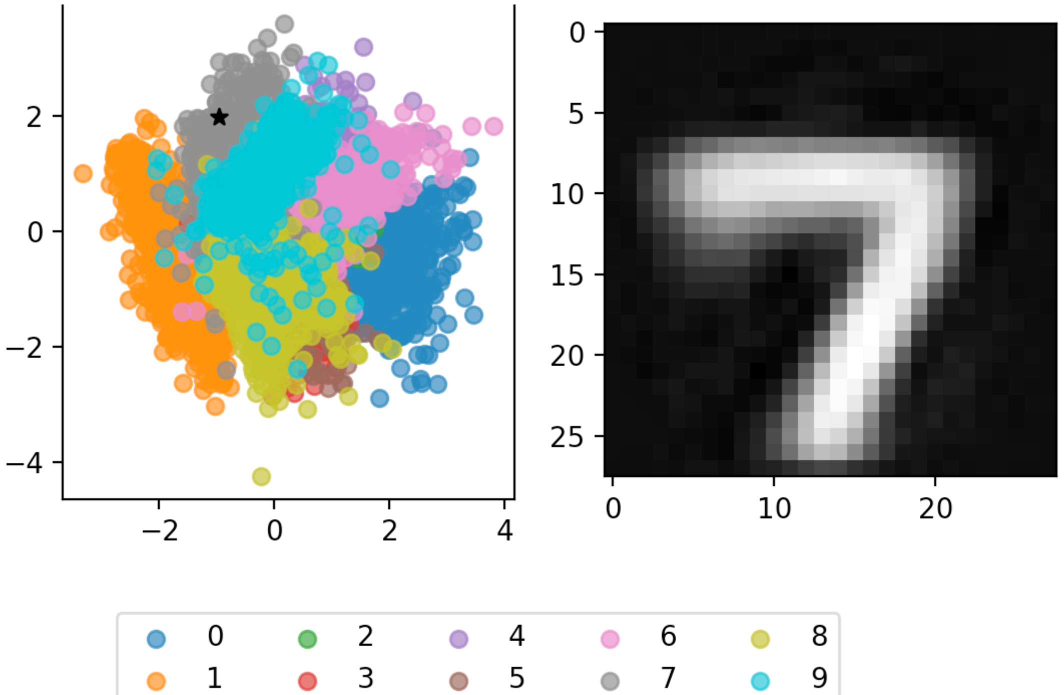
- The distribution of the encoder now approximately follows a normal distribution

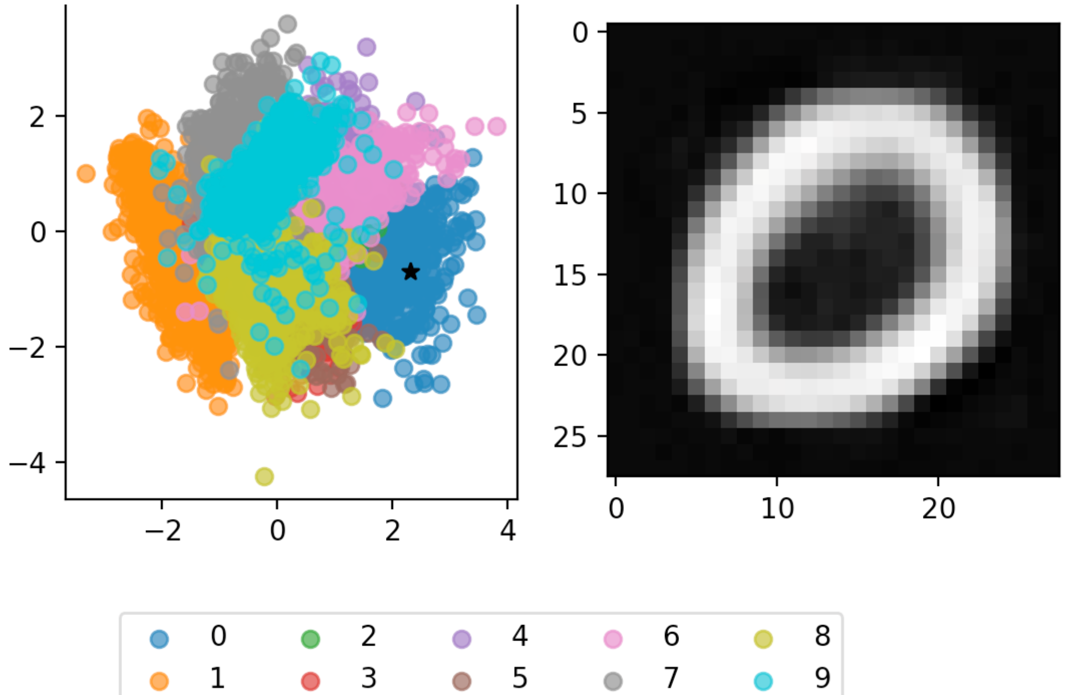




Generating Data with a VAE

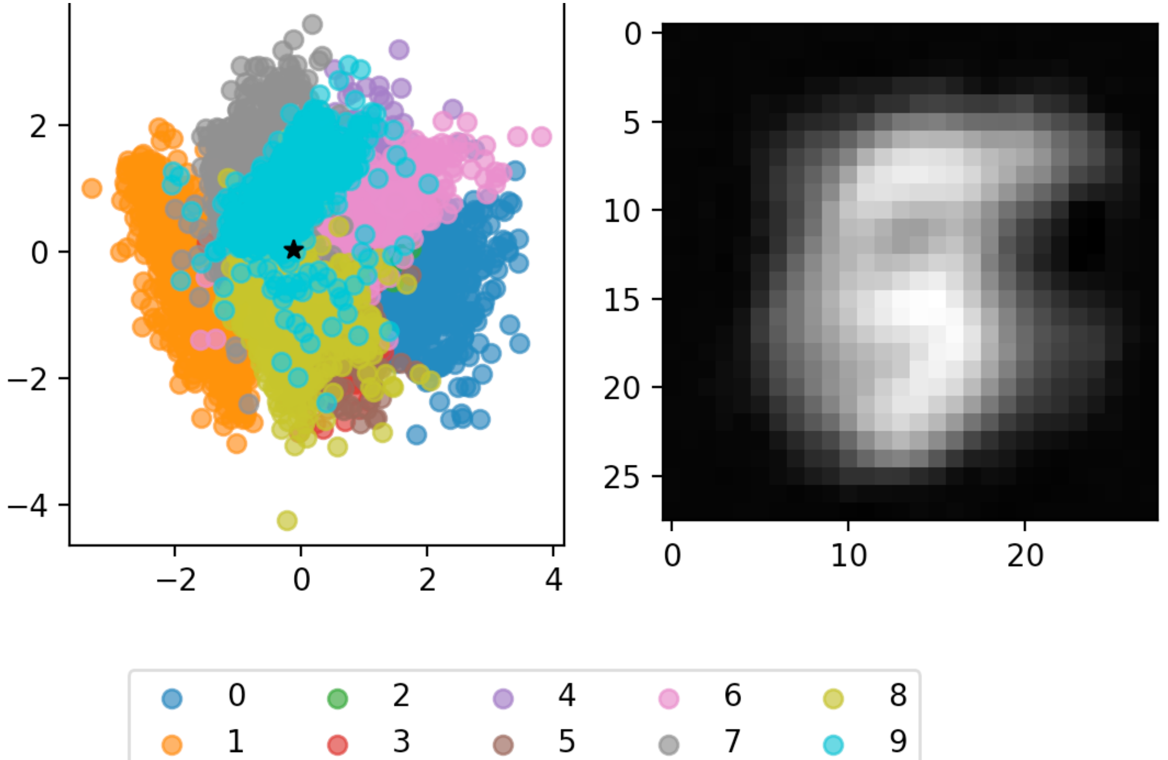
- Now, we can simply sample from a normal distribution with mean zero and standard deviation one
- We can also easily detect out of distribution data
- Does this mean we can always prevent generating undesirable data?

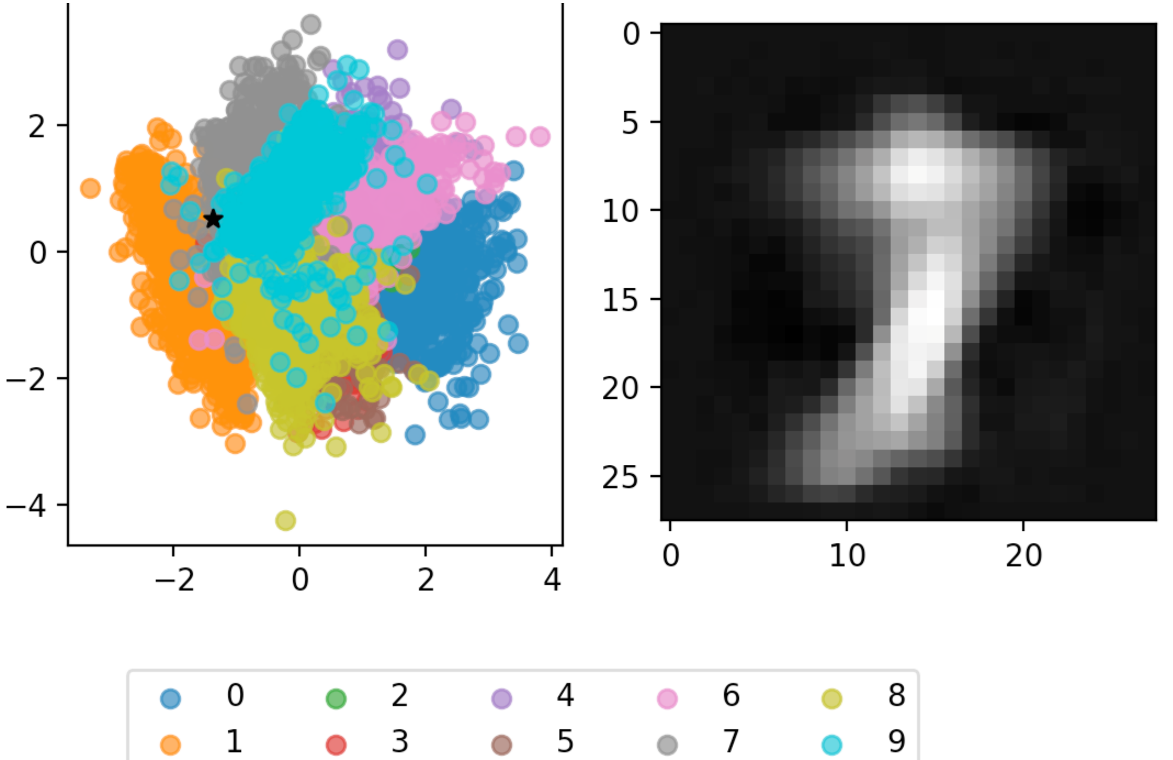




Generating Data with a VAE

- The same phenomena that caused adversarial examples to be such a problem is also present in VAEs
- Therefore, even if we stick to sampling samples that are highly likely for a normal distribution, we can still get undesirable outputs





Outline

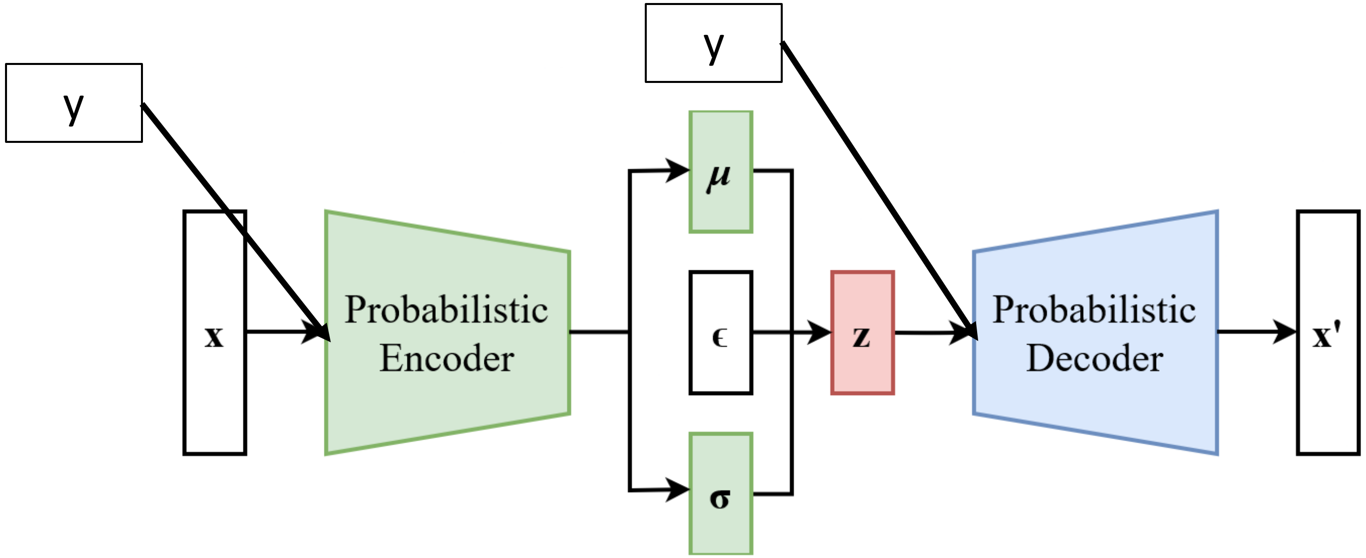
- Clustering
- Autoencoders
- Variational autoencoders
- Conditional variational autoencoders

Conditioning Data Generation on Properties

- Imagine we knew certain properties of the data, how can we train a generative model that can generate data with those specific properties?
- For example, digit class, size, angle, writing style, etc.

Conditional Variational Autoencoders

- We can add property information to the encoder and decoder
- This then encourages the inputs with each particular property to follow a normal distribution
- Then, to generate data, we sample from a normal distribution and give the desired properties to the decoder



Conditional Variational Autoencoders

- We can then explore the latent space for each property to better generate more fine-grained variations