Generative Modeling and Variational Auto-Encoders (VAEs)

Unsupervised Learning:
From observations (x) to latent variables (z), without labeled data

$$x \in \mathcal{X}$$

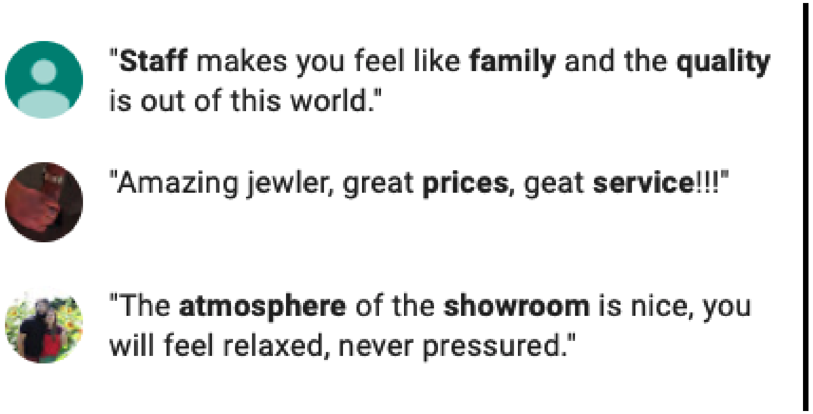
Observed variable: Because we see this data, both at training and at testing. This is what we sample. E.g. an image, or the vector representing a data point.

$$z \in \mathcal{Z}$$

Latent variable: We do not know this information.
Given x, we have to infer it.

z: label of digit, thickness, slope …
x: image of digit

z: content, lighting angle, zoom …
x: the photo

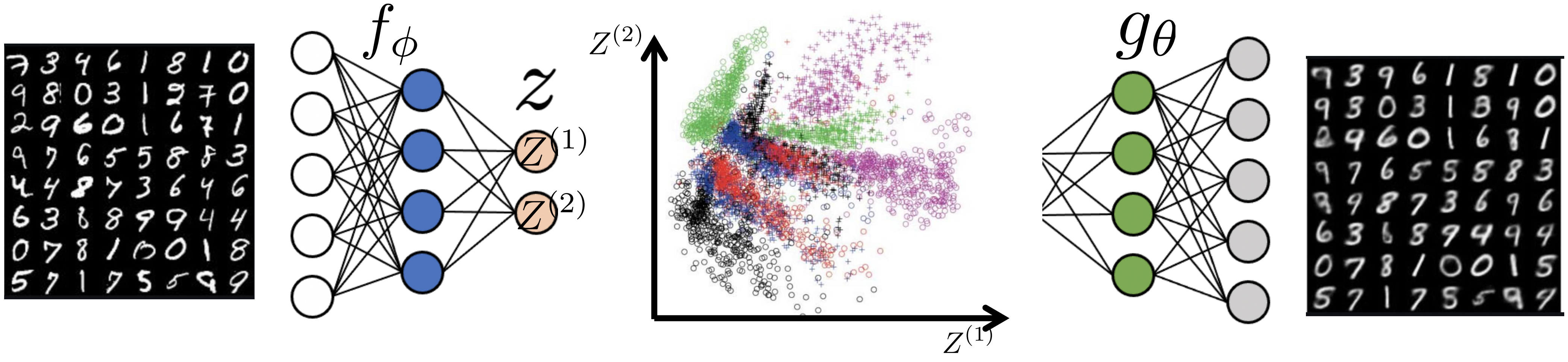


z: sentiment, vocabulary, grammar skills …
x: written review

Standard (basic) Auto-Encoder

Motivation:
Learn function $f : \mathcal{X} \rightarrow \mathcal{Z}$ (**Encoder**)
where representation z is hopefully useful.

Introduced $g : \mathcal{Z} \rightarrow X$ (**Decoder**)
to enable training with reconstruction loss.



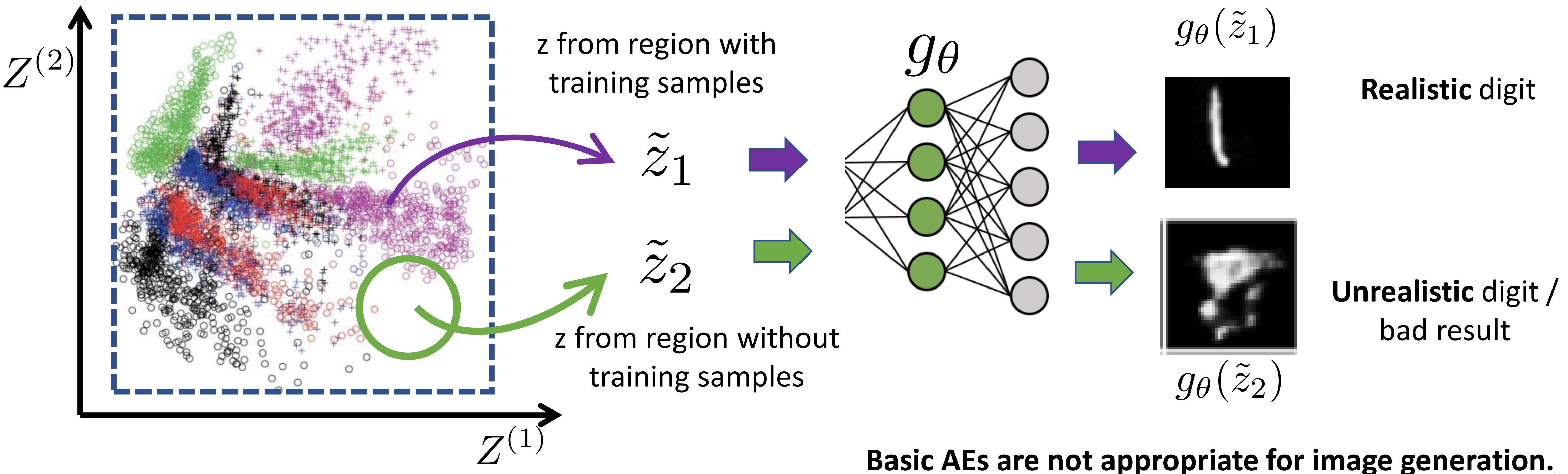
Ideal for encoding x to lower dimension (compression) and then decode.

Problems generating new data with basic AE

Step 1: Sample random z

E.g. With uniform probability between [min, max] values z seen during training

Step 2: Decode



Problems:

a) No “**real”** digits were encoded in that area during training. Hence these z values do not encode “realistic” digits.

b) Decoder has not learned to decode such z values

Basic AEs are not appropriate for image generation. Reconstruction loss does not train AE for generation.

We will see how ***Generative Models (VAEs and GANs)*** are trained appropriately for generation.

Generative Models

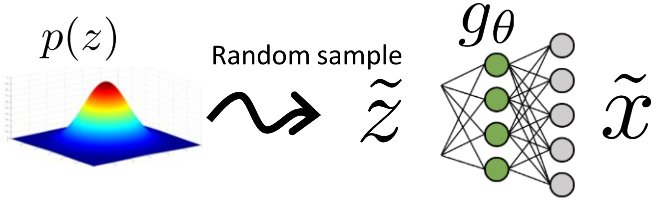
Nice intro to Generative Modelling by David Foster:
https://www.oreilly.com/library/view/generative-deep-learning/9781492041931/ch01.html

What is a Generative Model?

"A generative model describes **how a dataset is generated**, in terms of a **probabilistic model**. By **sampling** from this model, we are **able to generate new data**."

Generative Deep Learning, by David Foster

Prior distribution of z

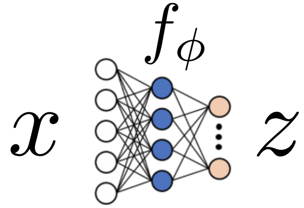


Generated new data point

Generative models: $g : \mathcal{Z} \rightarrow X$

We assume: z "causes" x

In contrast to:
Recognition (discriminative) models: $f : \mathcal{X} \rightarrow \mathcal{Z} \;\; (\text{or } \mathcal{Y})$



Basic AEs: Learn to infer meaningful representation z of data

Given only input samples x,
how to learn a meaningful encoding

$$f_\theta : \mathcal{X} \rightarrow \mathcal{Z}$$

This was the original motivation for the creation of Auto-Encoders

They are not trained / designed for generation (sampling) of new data points.

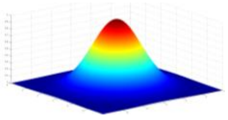
Generative Models:
How to learn model of process that generates realistic samples?

(in unsupervised manner)

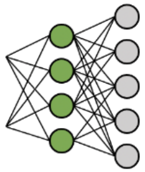
Generative models: $g : \mathcal{Z} \rightarrow \mathcal{X}$

Prior distribution of z

$p(z)$



sample $\rightsquigarrow$ $\tilde{z}$



$\tilde{x}$

Output must be realistic

What is a training objective for learning such a model?

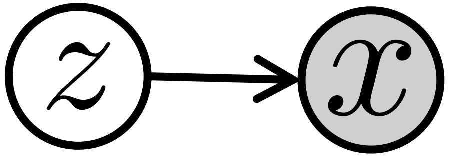
z: content, lighting angle, zoom …
x: the photo

In the process, we will also "have" to learn a representation Z that encodes meaningful info about the data.

"What I cannot create, I do not understand."

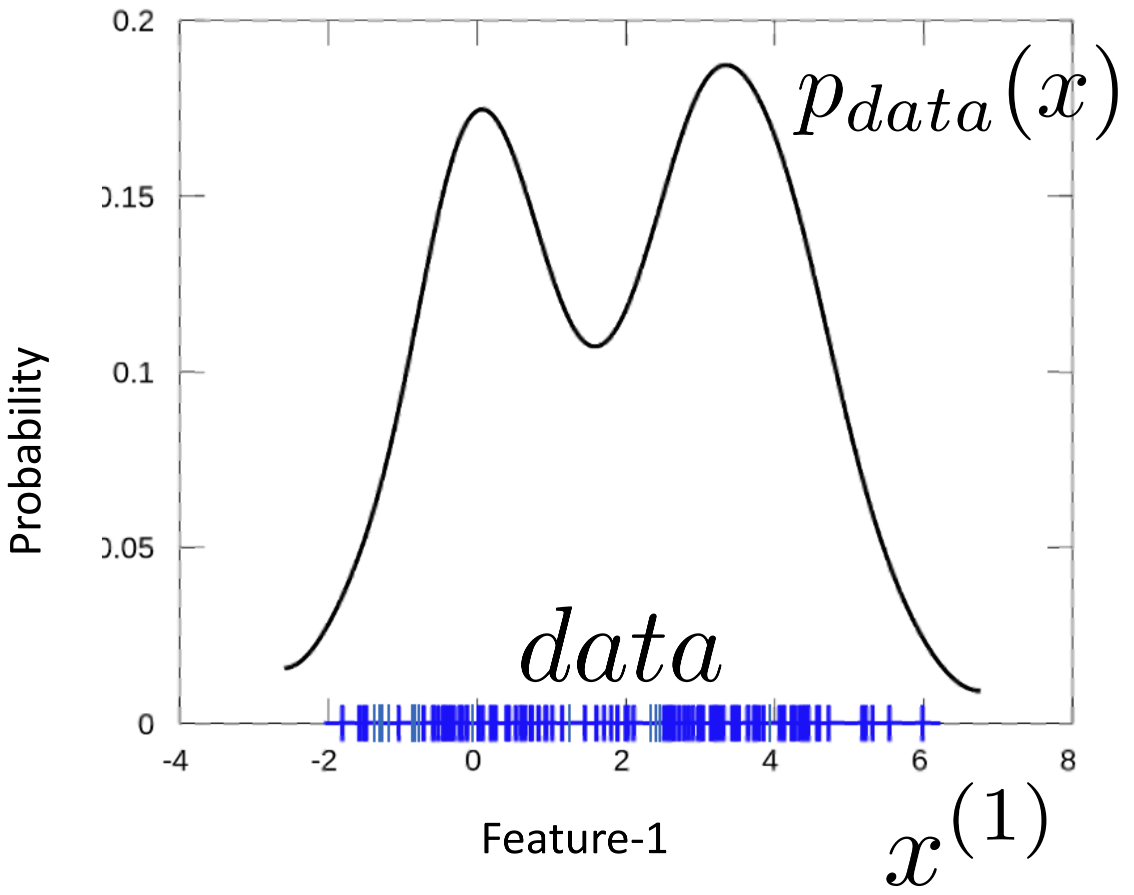
Richard Feynman

z "causes" x



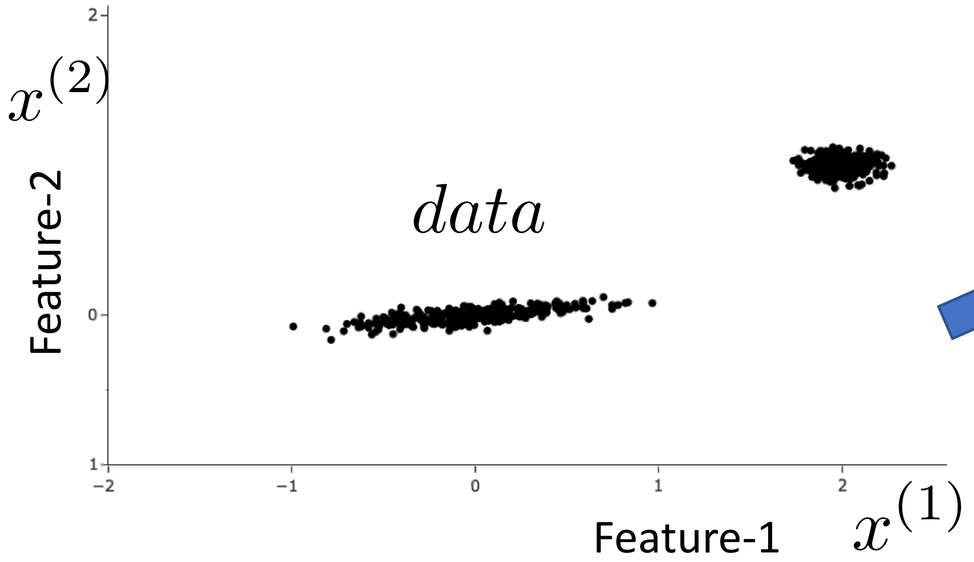
Probability Density Function (PDF) of data, $p_{data}(x)$

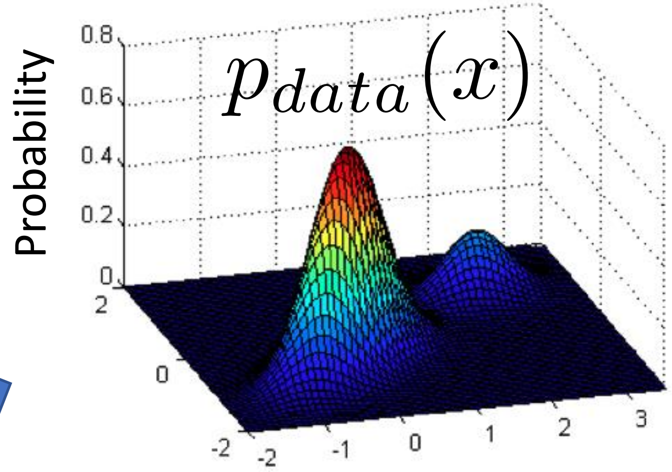
1-dimensional data:



Probability Density Function (PDF) of data, $p_{data}(x)$

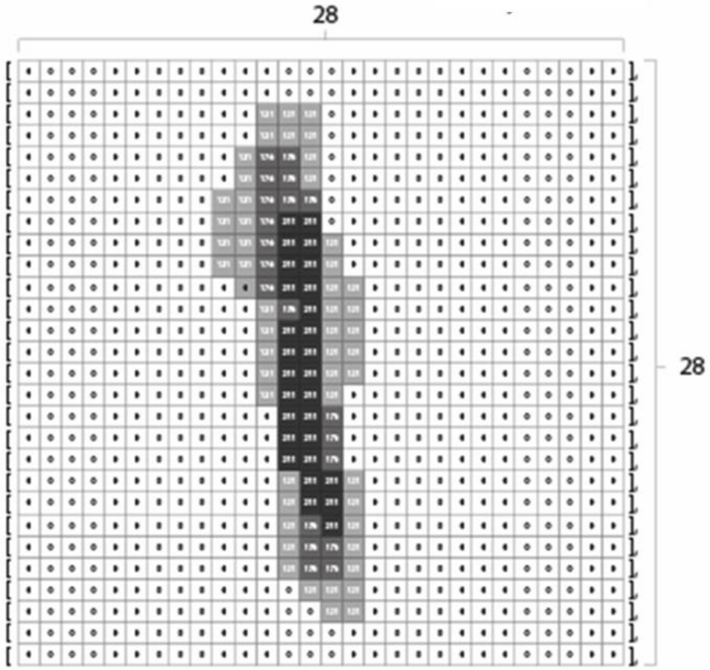
2-dimensional data:





Probability Density Function (PDF) of data, $p_{data}(x)$

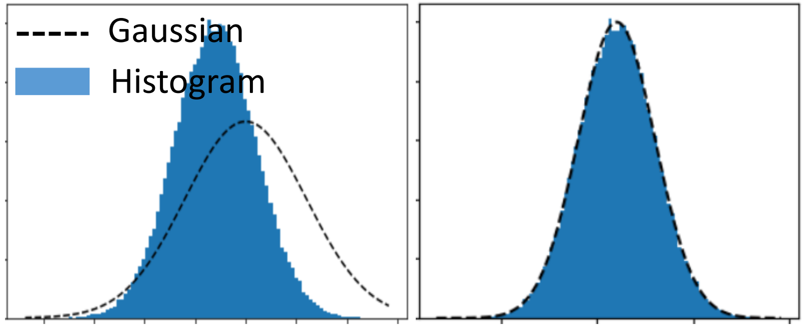
Multi-dimensional data:



MNIST: 28 x 28 = 784 features

- **Imagine a 784-dimensional space**. Each image of MNIST digit is 1 point in that space.
- Imagine all images of the MNIST database as points in that 784-d space.
- The probability of (MNIST) data to exist in the area of the space at point x is $p_{data}(x)$

Probability Density Estimation



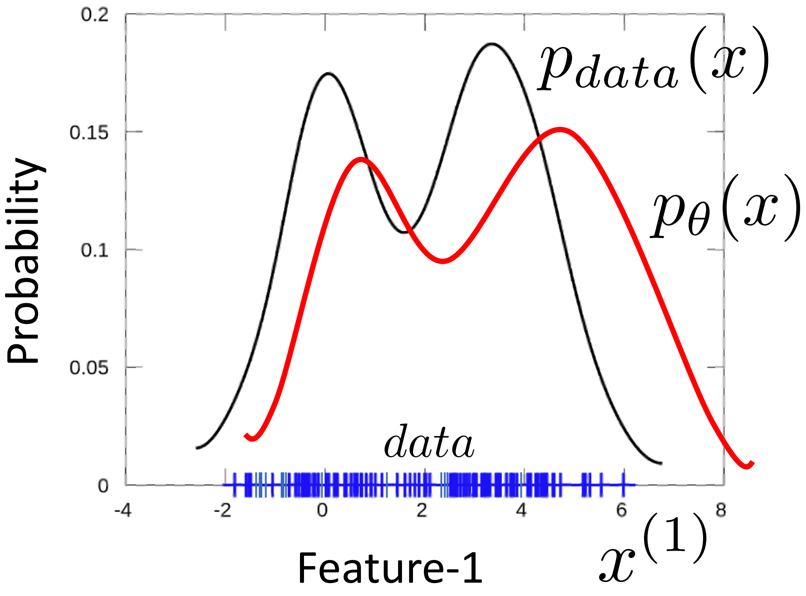
One of the main aims of unsupervised approaches and Generative Modelling.

Goal of Density Estimation:

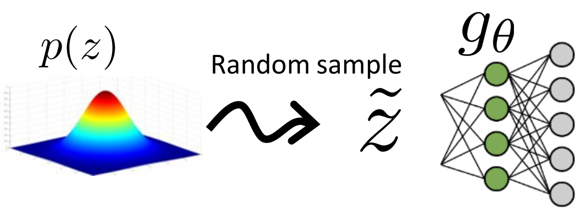
We could try to fit a probabilistic model $p_\theta(x)$ to the data,

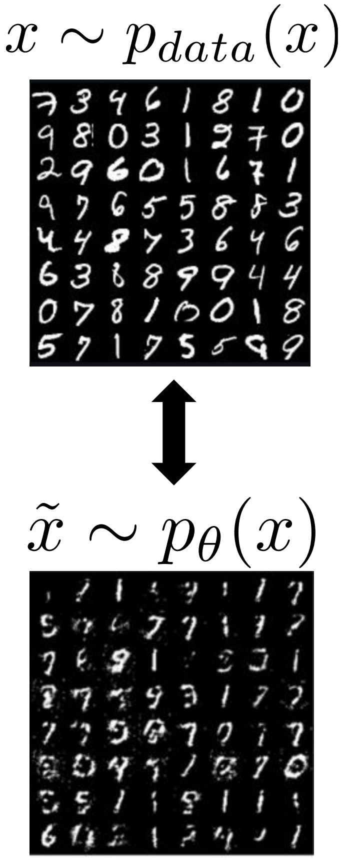
to learn their underlying distribution $p_{data}(x)$.

How? By learning its parameters θ so that: $p_\theta(x) \approx p_{data}(x)$



But we cannot always do that directly.
Perhaps we cannot compute $p_{data}(x)$ or $p_\theta(x)$.
Instead, we could do PDE *indirectly:*
Enforce samples from model to be similar to real data instead:





Both VAEs and GANs can be seen as following this approach.

Great progress thanks to VAEs and GANs

From: Deep Generative Modelling, David Foster

We know what we want to do now to learn a Generative model
(probability density estimation)

1. With what model?
2. How to train it?

Next video lecture:

Variational Auto-Encoders

Thank you very much