$P(y|x) = \frac{P(x|y)P(y)}{P(x)}$ | $P(x|y) = \frac{P(y|x)P(x)}{P(y)}$ $P(y|\underline{x})$

Generative Models

(Recap of Autoencoder, Variational Autoencoder)

$P(y|x) \propto P(x|y)P(y)$

$P(x|y) \propto \frac{P(y|x)P(x)}{P(x,y)}$

Tirtharaj Dash

Dept. of CS & IS and APPCAIR
BITS Pilani, Goa Campus

December 02, 2021

Discriminative Model

$P(y|\underline{x})$ ←

Generative Model ?

$P(\underline{x}, y)$

$P(\underline{x})$ $P(x_1, x_2, x_3, \ldots, x_d)$

different kinds of $\underline{x}$

① $\underline{x}$: Tabular structure — MLP — every row in the table map. an instance

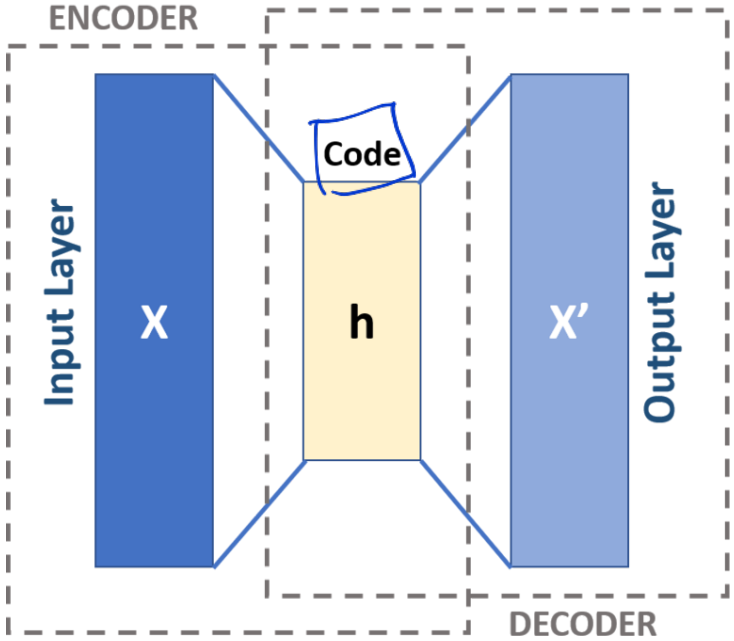
② $\underline{x}$: visual data (spatial) — CNN

③ $\underline{x}$: Temporal data (sequences) — RNN, Trans.

④ $\underline{x}$: Graph-structured (relational) data — GNN

Autoencoder – Discriminative Model

- AE is a neural network that learns to copy its input to its output.
- It has an internal (hidden) layer that describes a **code** used to represent the input.



Autoencoder – Discriminative Model

- It is constituted by two main parts: an encoder that maps the input into the code, and a decoder that maps the code to a reconstruction of the input.
- However, simply copying input to output would just *duplicate* the singal (rather than generalising). Why?
- Instead, AE reconstructs the input approximately, preserving the most relevant aspects of the data (we can call this: some important latent aspects).

Autoencoder – Discriminative Model

- Let **x** be an input example. The encoder and decoder do the following:

$$Enc : \mathcal{X} \mapsto \mathcal{H}$$

$$Dec : \mathcal{H} \mapsto \mathcal{X}$$

- Where, *Enc* and *Dec* are the functions obtained by minimising a recostruction loss.

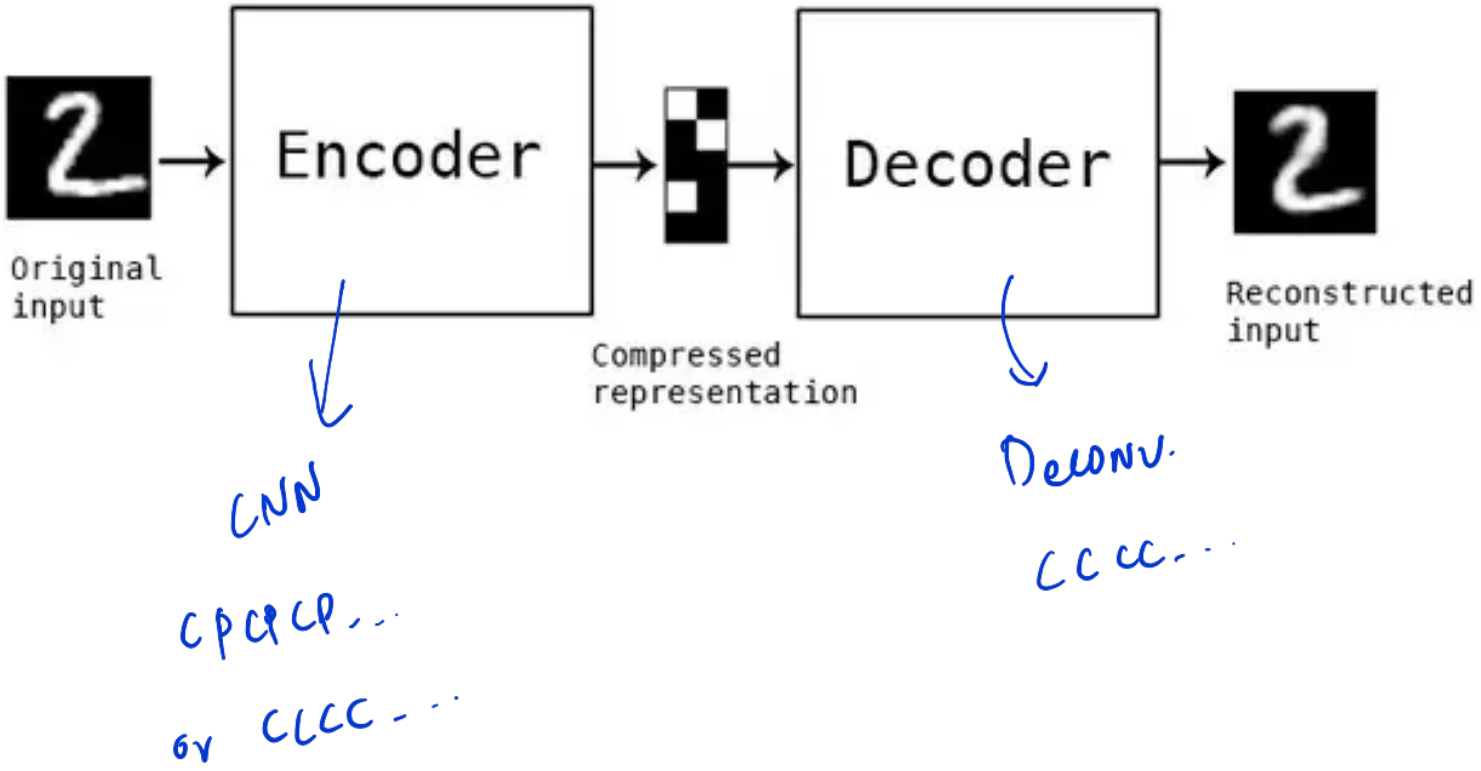
$$Enc, Dec = \arg \min_{Enc, Dec} ||\mathbf{x} - (Dec \circ Enc)(\mathbf{x})||^2$$

- In the simplest case, both encoder and decoder are single layered. That is:

$$\mathbf{h} = \sigma(\mathbf{w}\mathbf{x} + b)$$

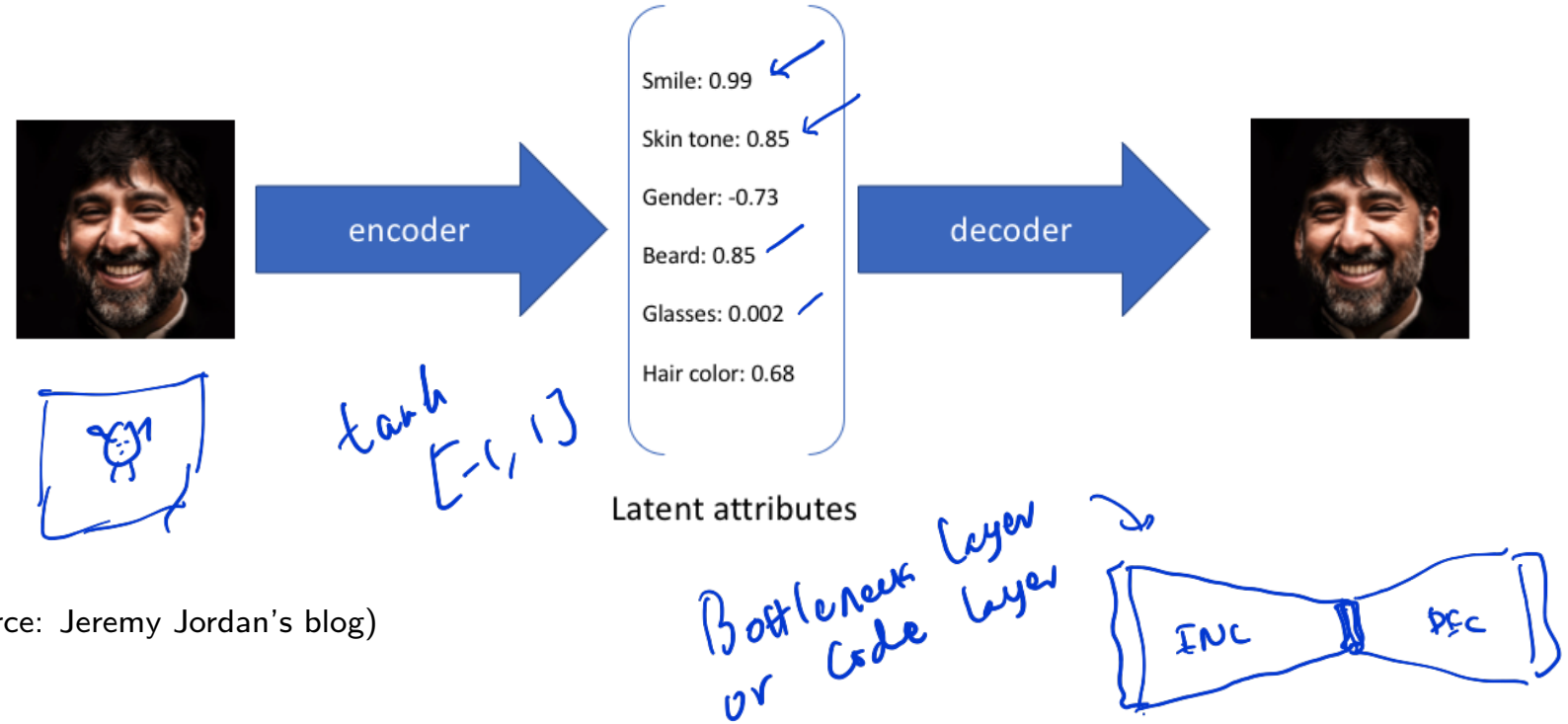
Autoencoder – Discriminative Model

- **h** is referred to as code or latent variable or latent representation.



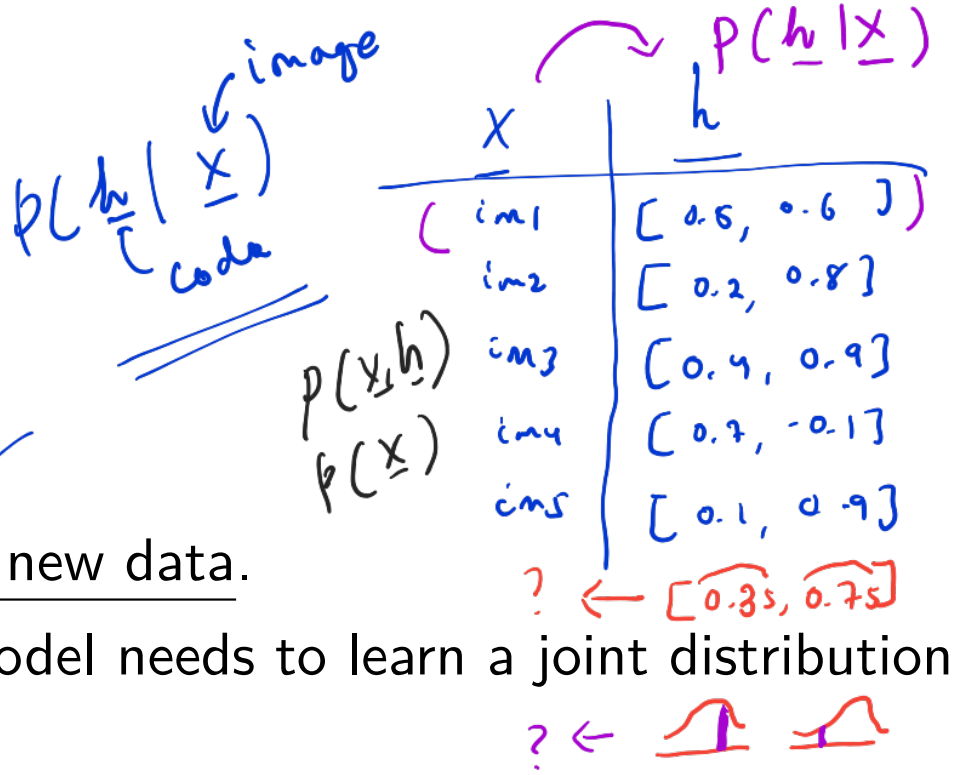
Autoencoder – Discriminative Model

- Each hidden dimension represents some latent feature learned about the input.
- For example (the features mentioned are hypothetical for demonstration purpose only):



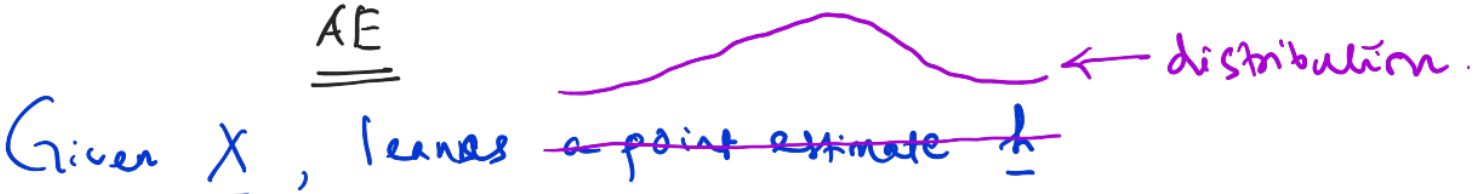
(Fig source: Jeremy Jordan's blog)

Autoencoder – Discriminative Model



- This is a discriminative model.
- AE can be used for:
  - compressing data
  - greedy layerwise pre-training
- AE cannot used for: generating new data.
- For generating new data, the model needs to learn a joint distribution of some kind $p(\mathbf{x})$ or $p(\mathbf{x}, \mathbf{h})$.

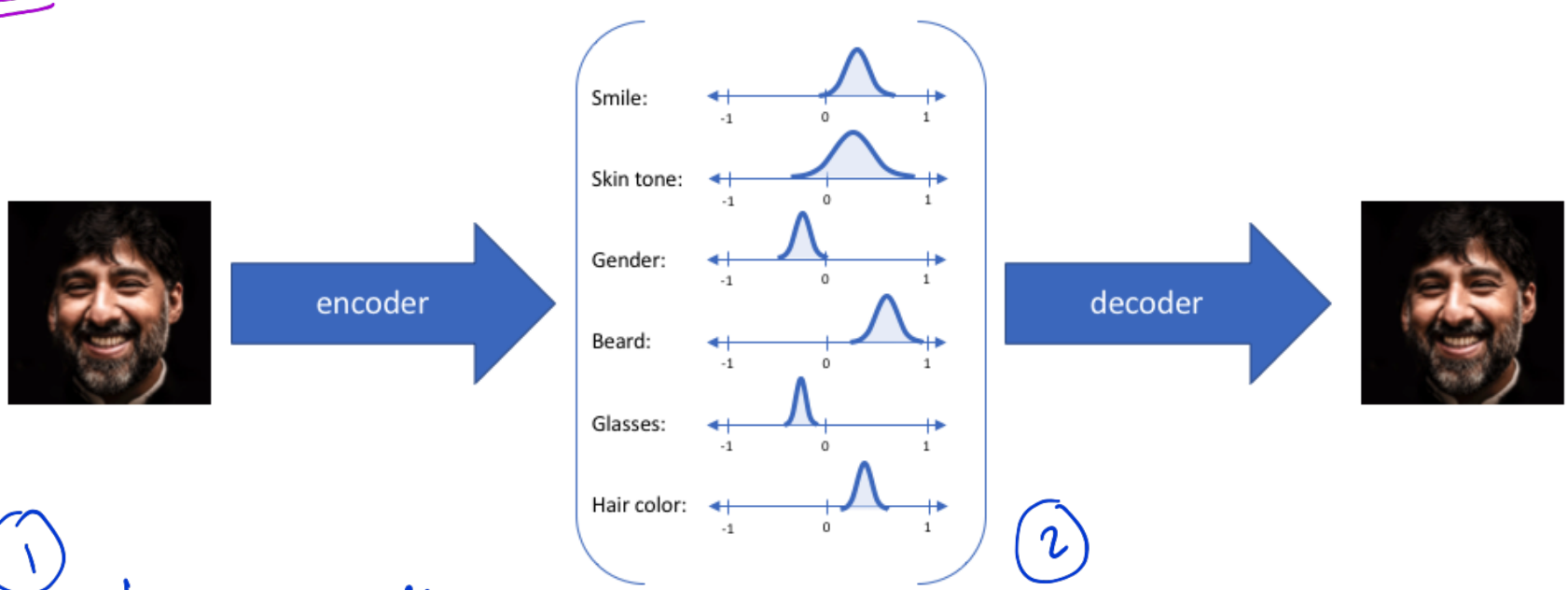
- Or, a model can be considered as "generative" when the input latent variable has probability distribution associated with it – a kind of autoencoder that does this is 'Variational AE'.



VAE

"Variational" AE : Generative AE

- VAE imposes a specific probabilistic structure on the hidden units.

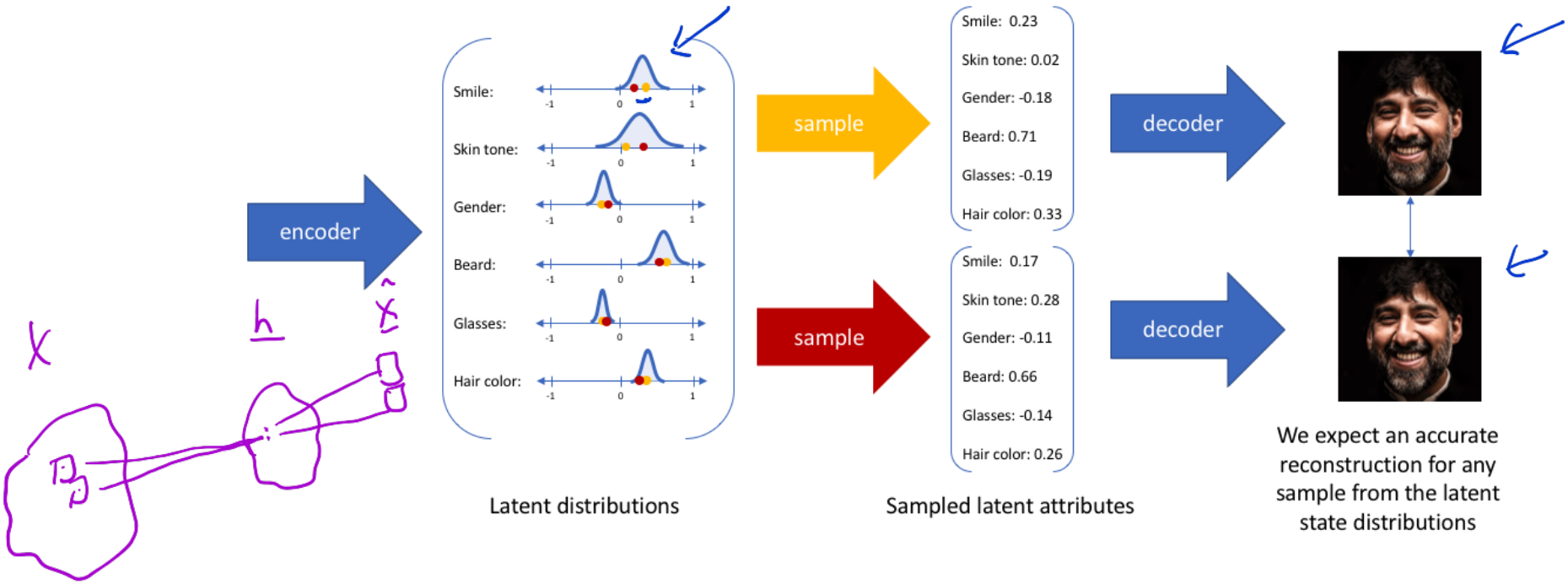


(Fig source: Jeremy Jordan's blog)

- The AE network is sometimes called 'recognition model' whereas the decoder network is sometimes referred to as the 'generative model'.

VAE

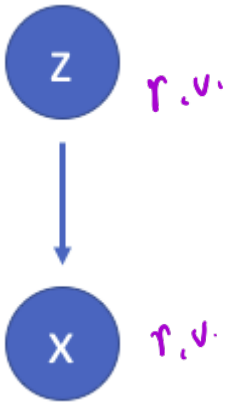
- The encoder model outputs a range of possible values (a statistical distro) from which, we can randomly sample and input to the decoder to re-construct the input. This enforces a continuous and smooth latent space representation.
- It is expected that values that are close enough in the latent space would result in similar reconstructions.



VAE

(Fig source: Jeremy Jordan's blog)

- Suppose there exists some hidden variable **z** which generates an observation **x**. As a Bayesian network, it looks like:



- The difficulty is that we know nothing of **z**, we can only see **x**. But, we can infer some characteristics of **z**:

$$p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{p(\mathbf{x})}$$

likelihood : $P(\underline{x}, \underline{z})$

prior

marginal.

$p(\underline{z}|\underline{x}) \propto p(\underline{x}|\underline{z})\,p(\underline{z})$

$\int p(x|z)p(z)\,dz$

unknown.

VAE

Exact estimate for $p(z|x)$ is not possible.

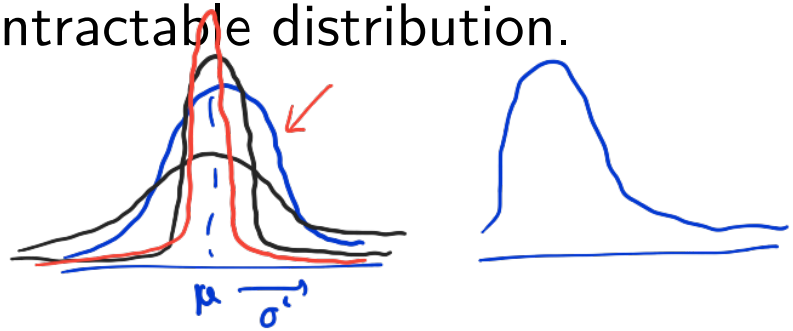
oblem Computing the denominator $p(\mathbf{x})$ is hard:

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$$

$x \in \mathbb{R}^d$

olution Variational inference → approximate if

- Let's approximate $p(\mathbf{z}|\mathbf{x})$ by another distribution $q(\mathbf{z}|\mathbf{x})$ such that it has a tractable distribution.
- If we can define the parameters of $q(\mathbf{z}|\mathbf{x})$ such that it is very similar to $p(\mathbf{z}|\mathbf{x})$, we can use it to perform approximate inference of the intractable distribution.



$\phi \xrightarrow{①} q$ ↓ ② params ?

VAE

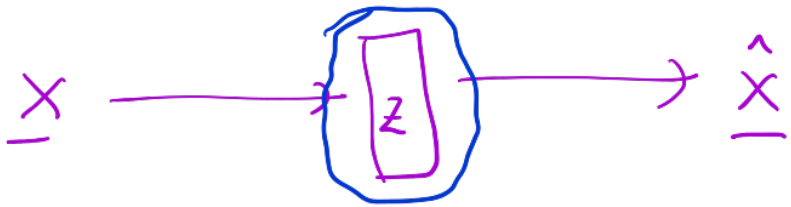
- If we minimise KL divergence between $q(\mathbf{z}|\mathbf{x})$ and $p(\mathbf{z}|\mathbf{x})$, then these two distros will become similar to each other.

$$\min KL(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z}|\mathbf{x}))$$

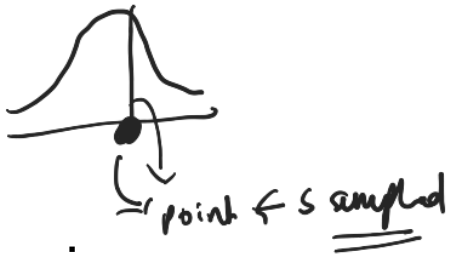
Which in turn can be solved by maximising the following (not showing the proof here):

$$E_{q(\mathbf{z}|\mathbf{x})} \log p(\mathbf{x}|\mathbf{z}) - KL(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))$$

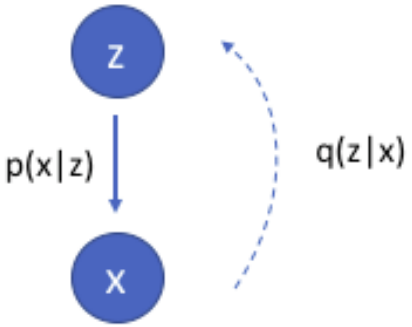
The first term represents reconstruction likelihood. The second term enforces that q distro is similar to the true distro $p(\mathbf{z})$.

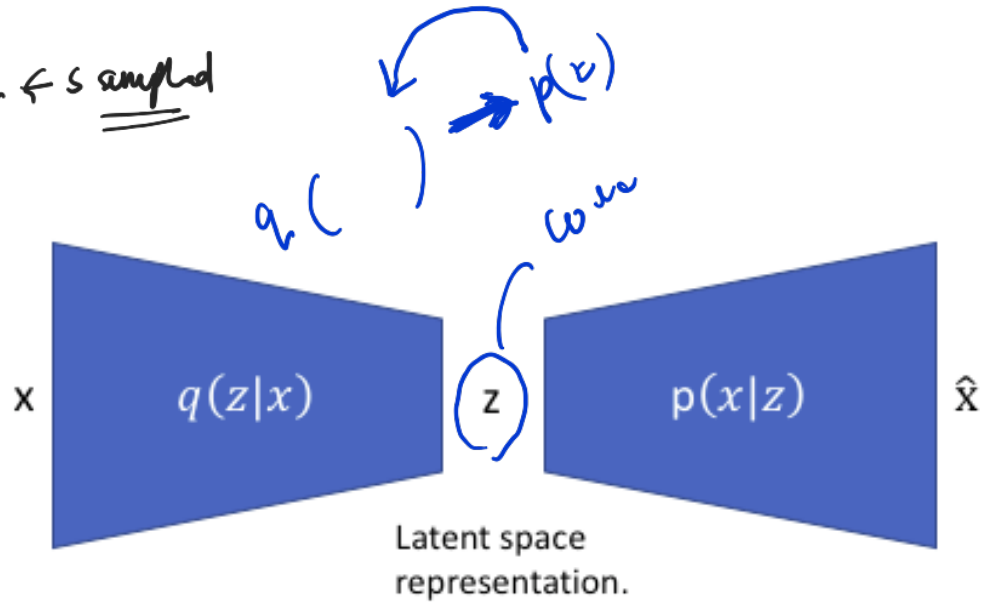


VAE

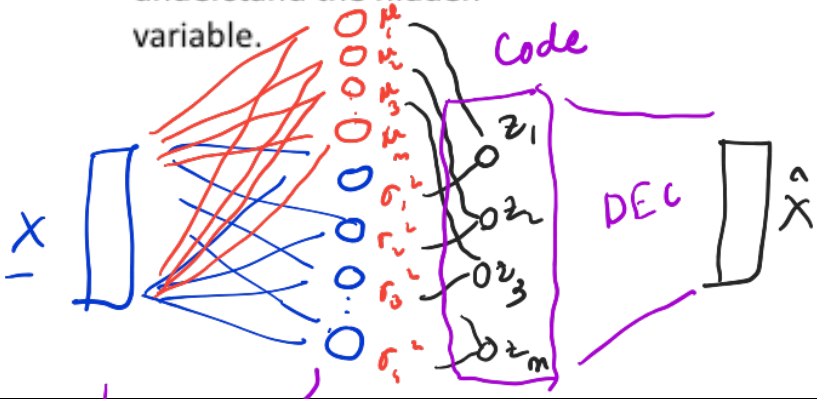


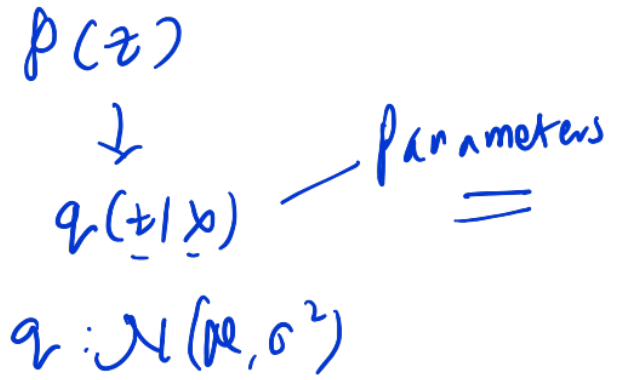
- This is what is happening:





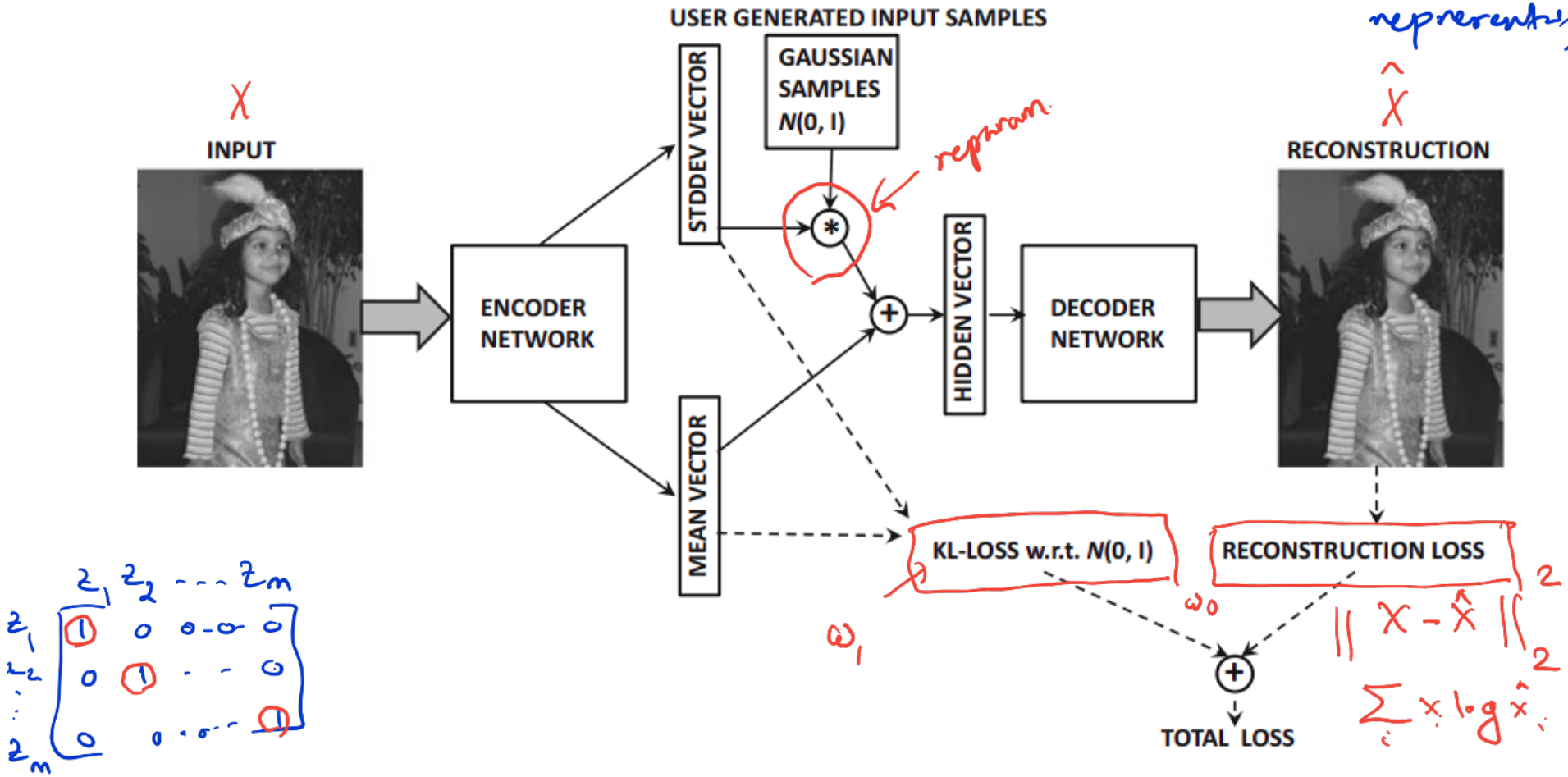
We'd like to use our observations to understand the hidden variable.





VAE

- For VAE, we assume that the 'true' distribution $p(\mathbf{z})$ is Gaussian $\mathcal{N}(0, I)$ (I is the identity matrix).
- Therefore, VAE's loss function is this:



VAE


(Fig source: Charu Agarwal's NNDL book)

- The loss function can be written as the sum of the reconstruction loss, and sum of KL divergences over all dimensions of the latent space $\mathbf{z}$:

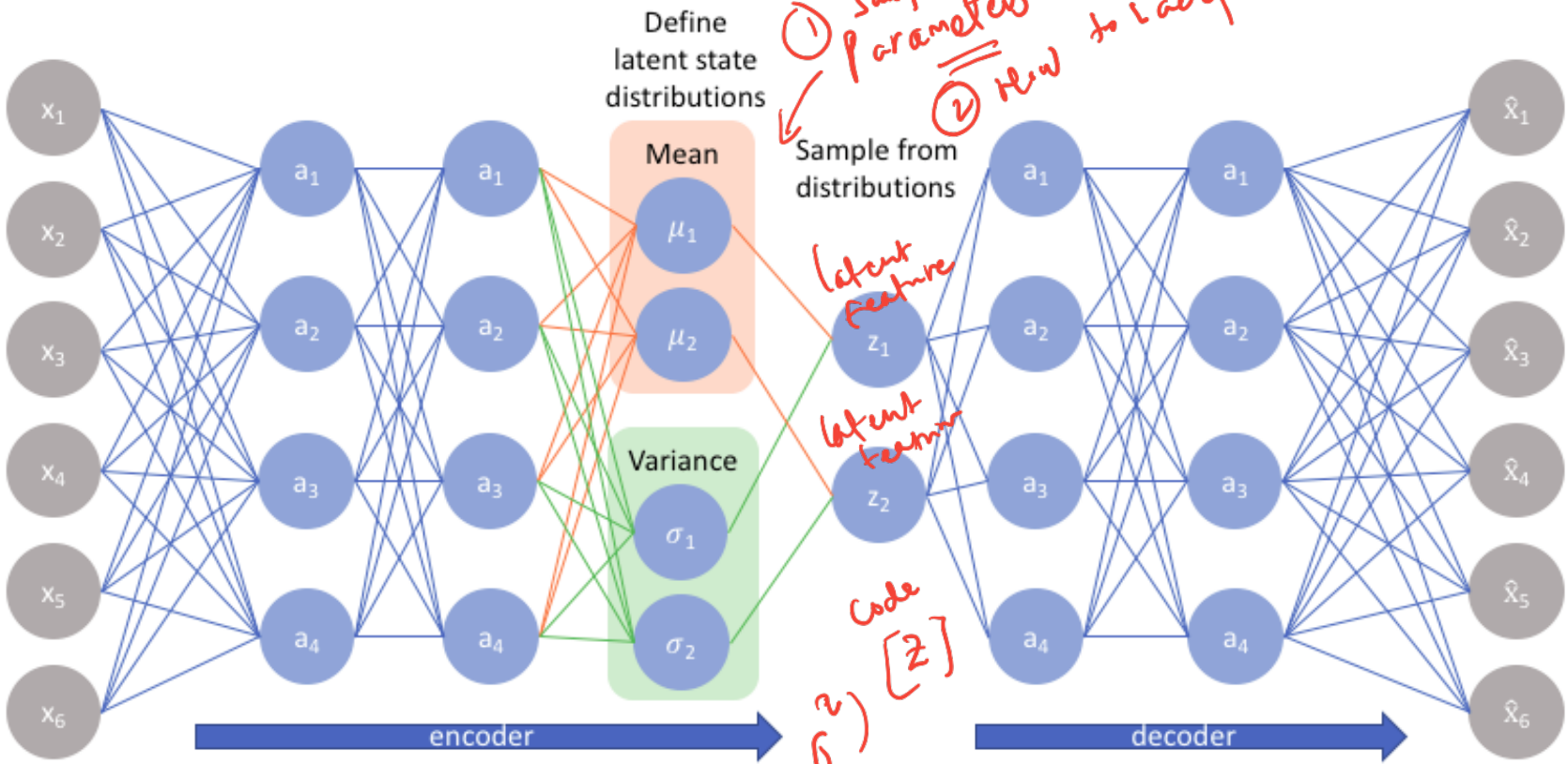
$$L(\mathbf{x}, \hat{\mathbf{x}}) + \sum_j KL(q_j(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))$$

Recon loss

$KL(q \,||\, p) = -\sum q \log \frac{q}{p}$

VAE

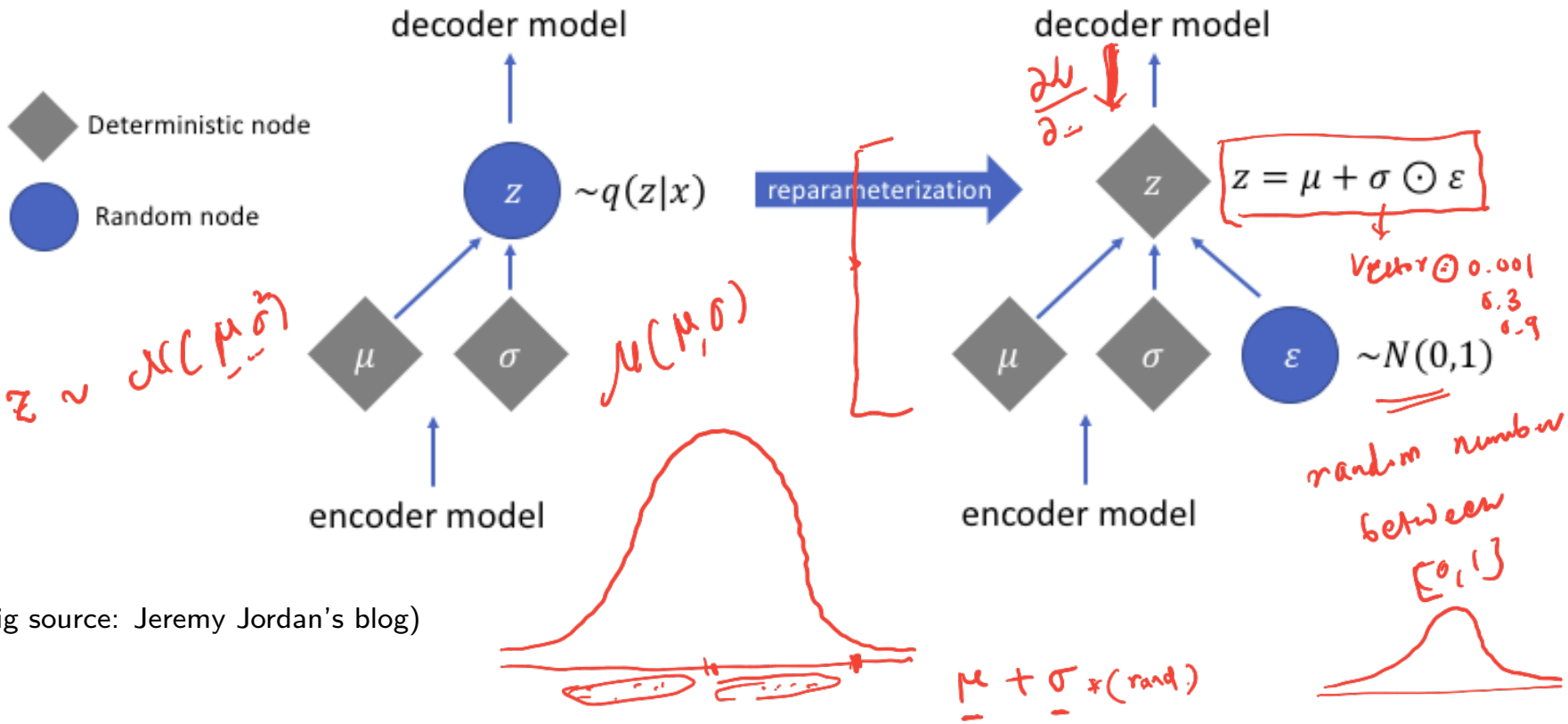
- So, if we see closely at these the NN structure is:



(Fig source: Jeremy Jordan's blog)

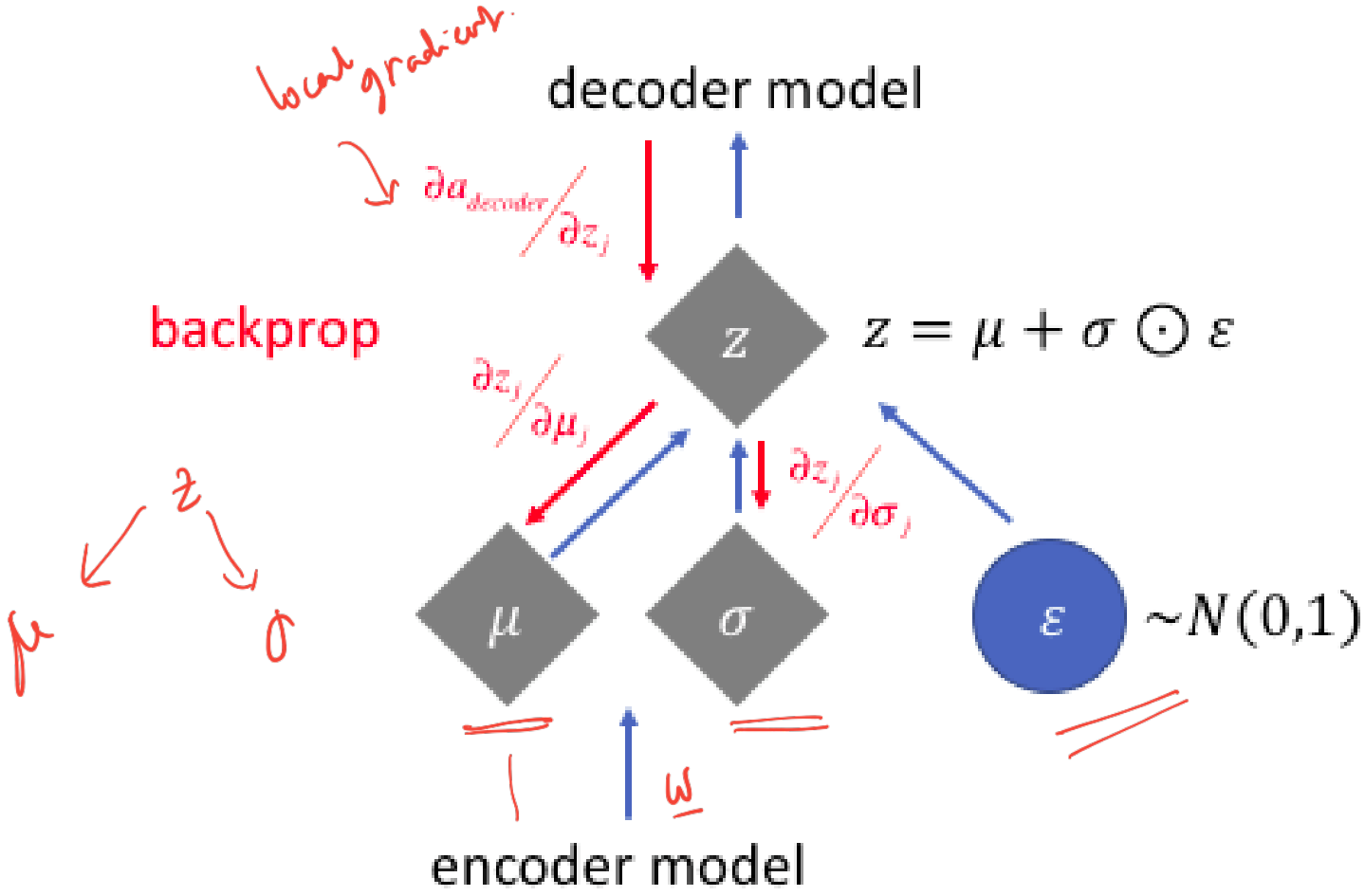
VAE

- Note z's are sampled from a normal distro. But, backprop cannot handle sampling directly. We use a trick called 're-parameterisation trick' to ensure that backprop can go inside sampling.
- This is:



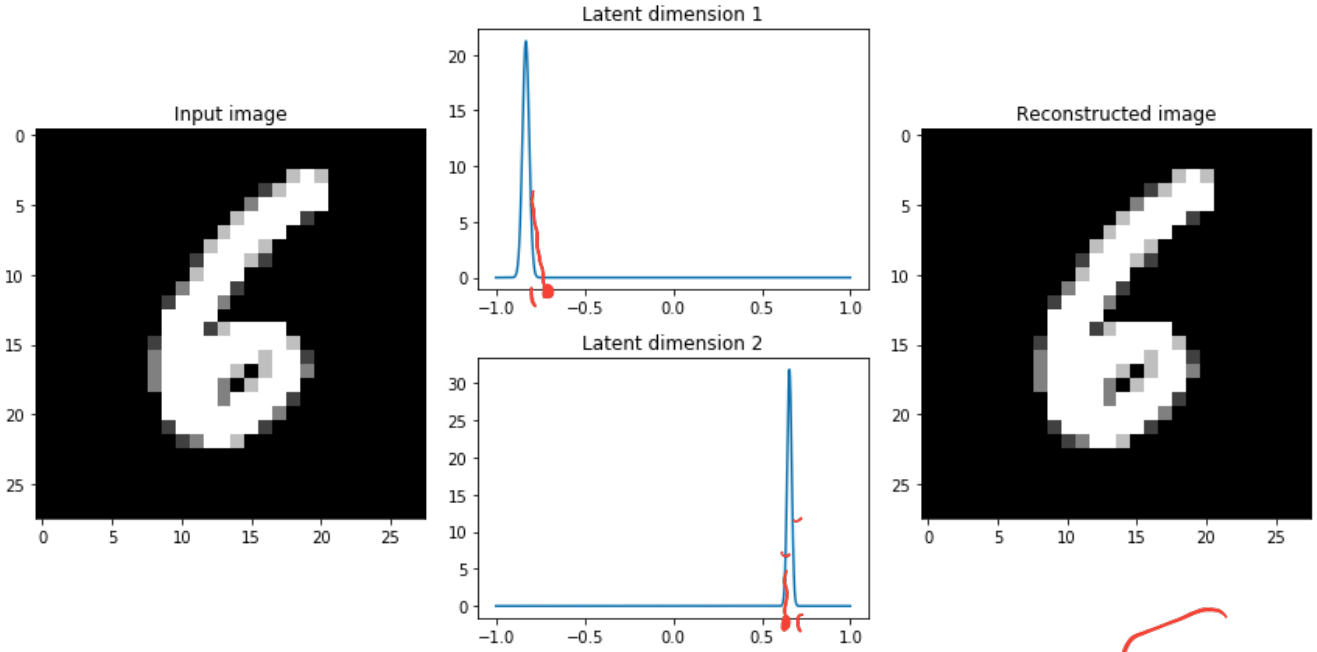
(Fig source: Jeremy Jordan's blog)

VAE



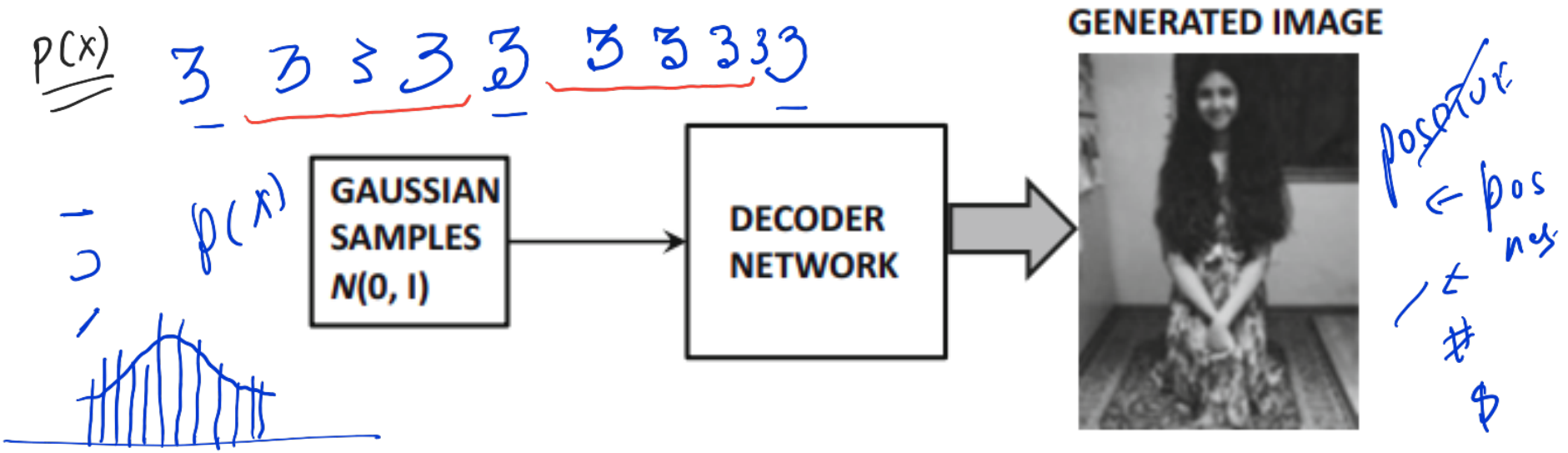
(Fig source: Jeremy Jordan's blog)

- A simple experimental visualisation with 2 latent feature dimensions:



VAE – as generative model

- The decoder network can be used as a generative model capable of generating new data similar to what was observed during training.
- Specifically, we will be sampling data from the (prior) distribution $p(\mathbf{z})$ that was constructed during training.



(Fig source: Charu Agarwal's NNDL book)

- In practice:
  - Requires large amount of data
  - Requires large amount of training iterations