UNIVERSITY OF NOTRE DAME

Lesson 8: Variational Autoencoders

ML Summer 2021 Reading Group.

General remarks.

- Based on a Generative Process to model probability distributions, defined over datapoints X in a high-dimensional space $\mathbb{X}$, where the encoding process introduce a latent variables z.

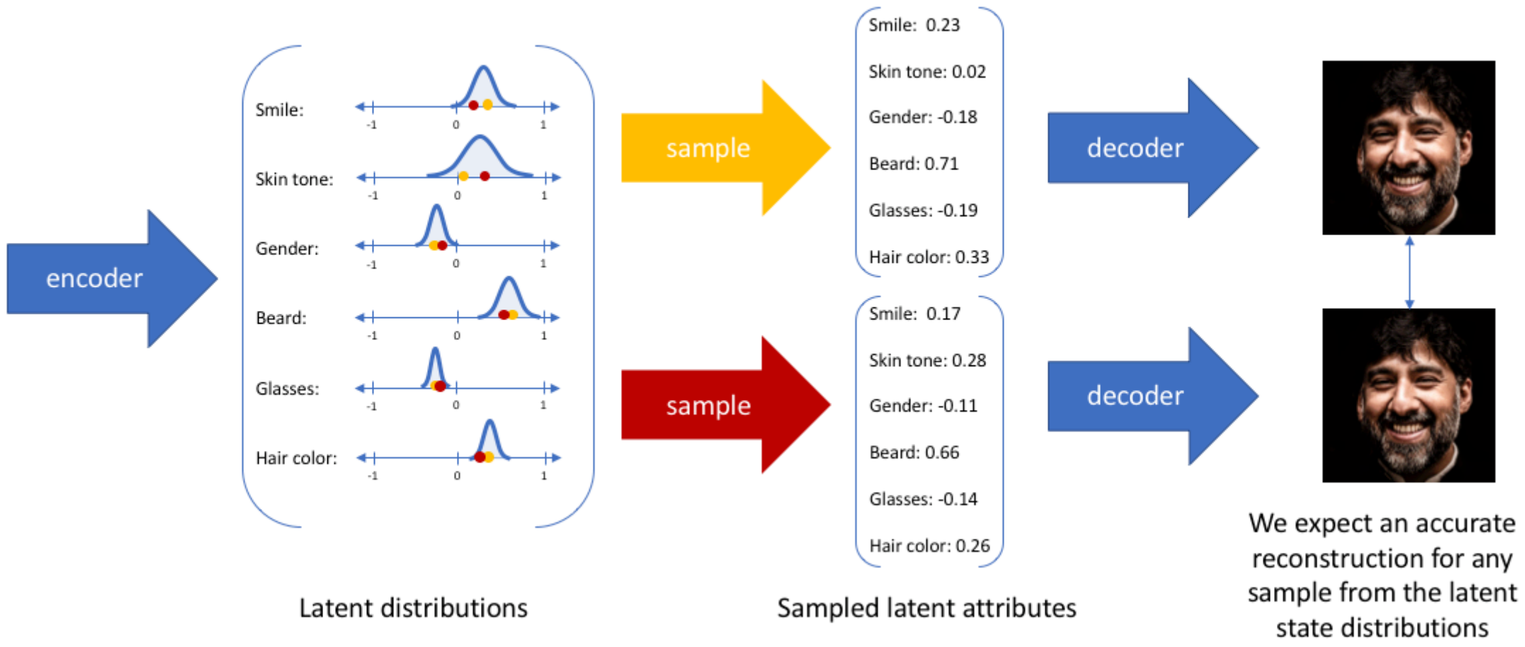
- Maximizing the probability of each X in training set under a generative process, according to: $P(X) = \int P(X|z;\theta)P(z)dz$. (Maximum likelihood condition)

- For Variational Autoencoders (VAE), the choice of the output distribution is a multivariable Gaussian, $P(X|z;\theta) = N(X|f(x;\theta), \sigma^2 * I)$. Multi-layer neural network represented by $f(x;\theta)$.

- From the encoding process, resulting in a probability distribution function $Q(z)$, it used the Kullback-Leibler divergence to measure the distance to the reconstructed $P(X)$ (decoding): $\log P(X) - D_{KL}[Q(z|X)||P(z|X)] = E[\log P(X|z)] - D_{KL}[Q(z|X)||P(z)]$

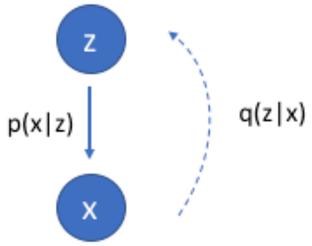
VAE: Example

For any sampling of the latent distributions, we're expecting the decoder model to be able to accurately reconstruct the input. Thus, values which are nearby to one another in latent space should correspond with very similar reconstructions.

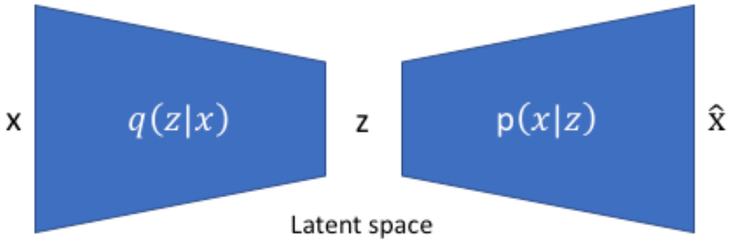


Jeremy Jordan, **VAE**. https://www.jeremyjordan.me/variational-autoencoders/.

VAE for NN



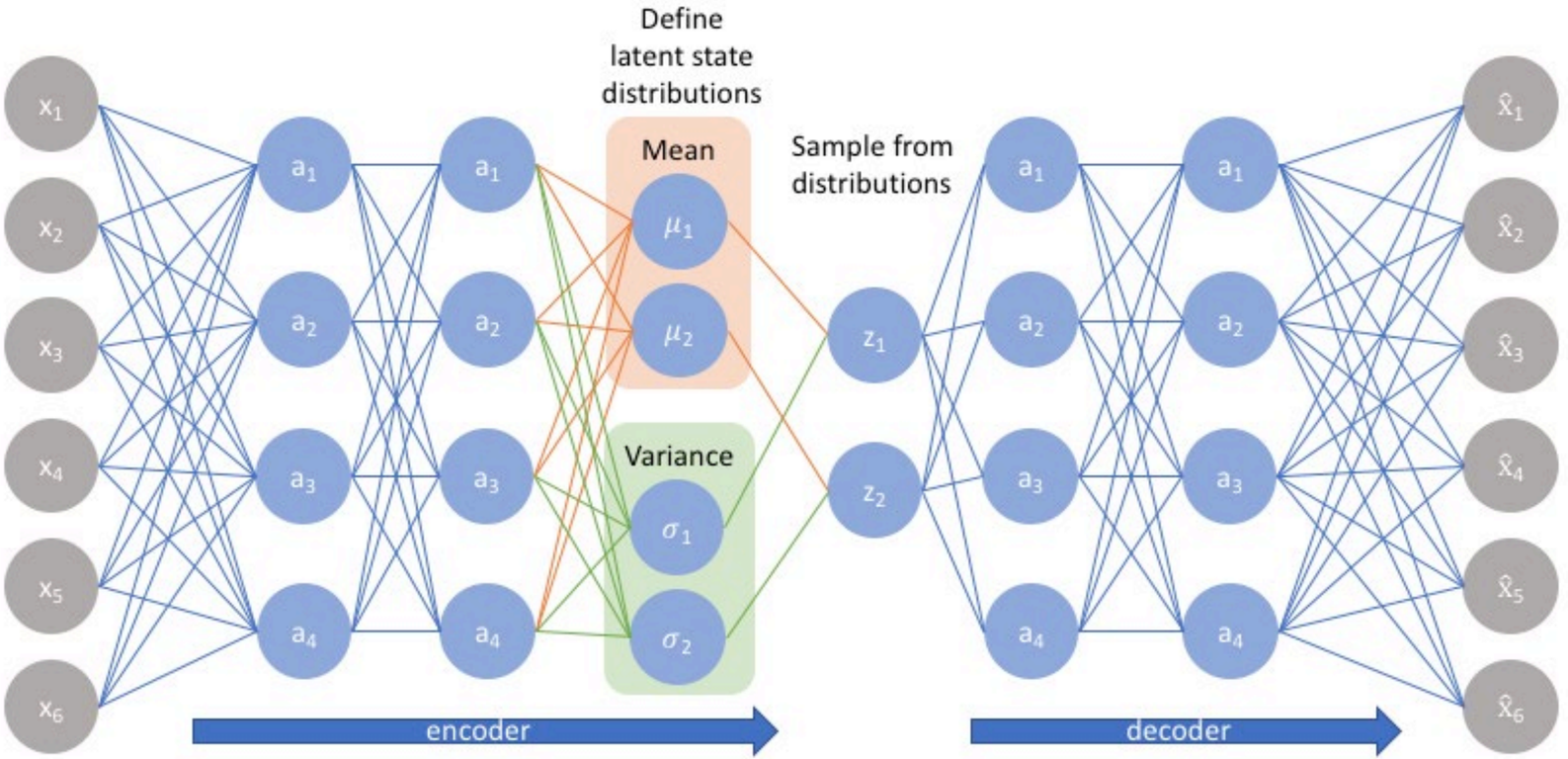
We'd like to use our observations to understand the hidden variable.



Latent space representation.

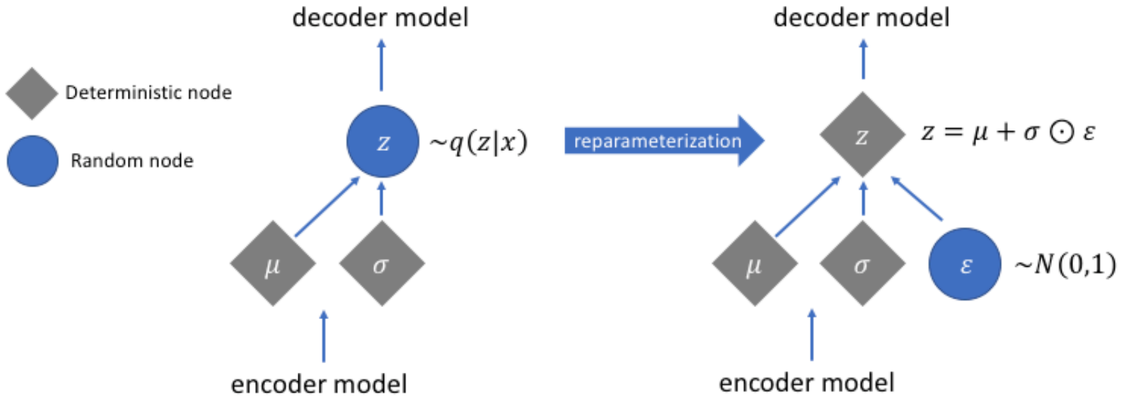
Neural network mapping x to z.

Neural network mapping z to x.



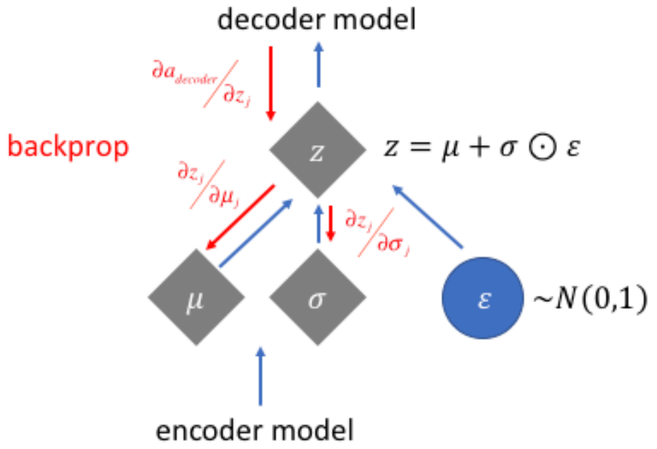
Jeremy Jordan, **VAE**. https://www.jeremyjordan.me/variational-autoencoders/.

VAE: Reparametrization for backpropagation



Treating the random step process for backpropagation. Introducing random noise at the latent step, which shift the randomly sampled $\mathcal{E}$ by the latent distribution's mean and scale it by the latent distribution's variance.

With this reparameterization, we can now optimize the *parameters* of the distribution while still maintaining the ability to randomly sample from that distribution.



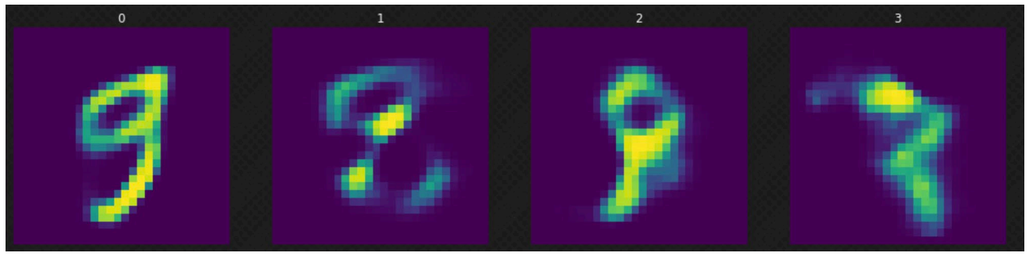
VAE: Using pytorch for MNIST dataset

- From lesson's notebook, the VAE model can be coded for different neural network architectures.

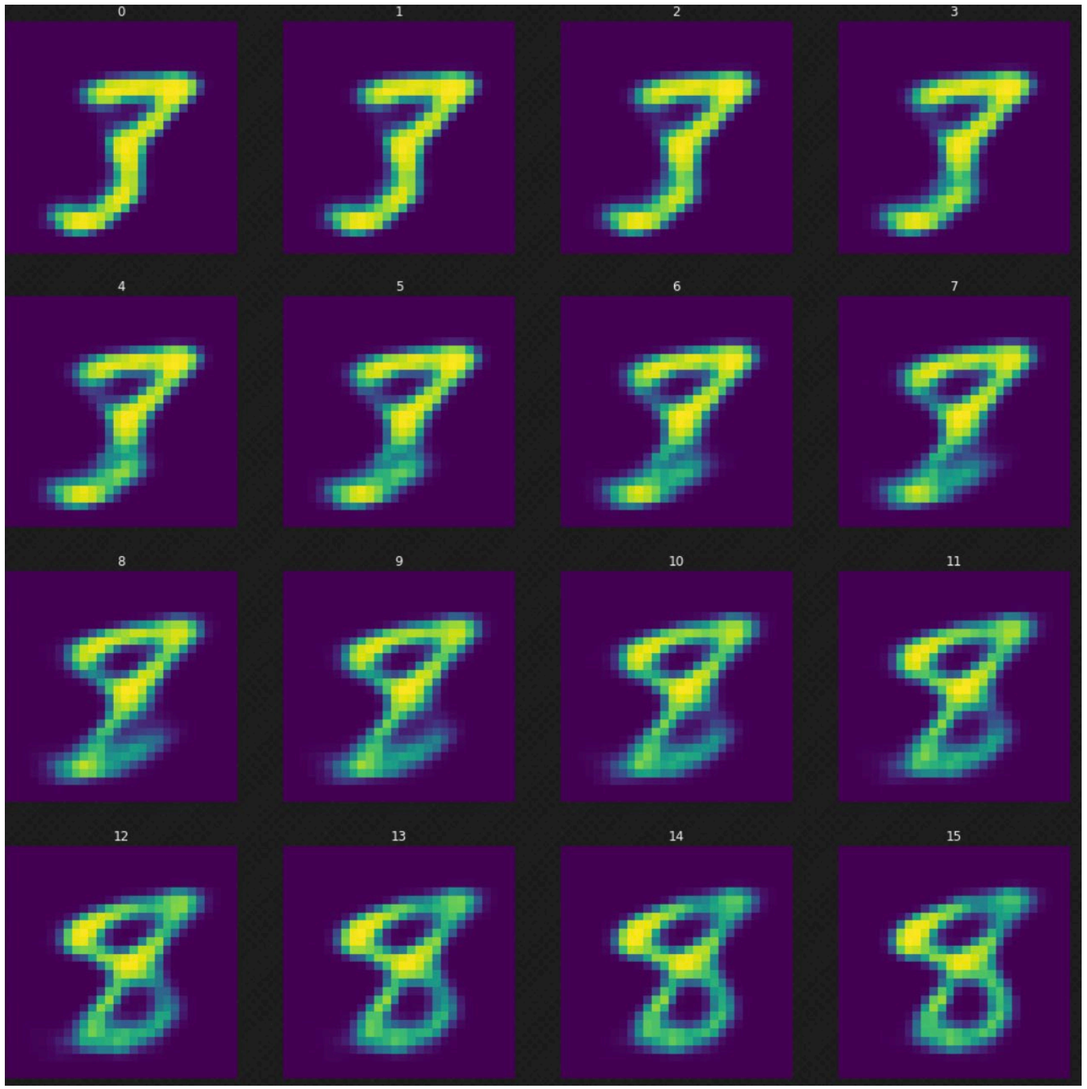
```
# Defining the model

d = 20

class VAE(nn.Module):
    def __init__(self):
        super().__init__()

        self.encoder = nn.Sequential(
            nn.Linear(784, d ** 2),
            nn.ReLU(),
            nn.Linear(d ** 2, d * 2)
        )

        self.decoder = nn.Sequential(
            nn.Linear(d, d ** 2),
            nn.ReLU(),
            nn.Linear(d ** 2, 784),
            nn.Sigmoid(),
        )

    def reparameterise(self, mu, logvar):
        if self.training:
            std = logvar.mul(0.5).exp_()
            eps = std.new_empty(std.size()).normal_()
            return eps.mul_(std).add_(mu)
        else:
            return mu

    def forward(self, x):
        mu_logvar = self.encoder(x.view(-1, 784)).view(-1, 2, d)
        mu = mu_logvar[:, 0, :]
        logvar = mu_logvar[:, 1, :]
        z = self.reparameterise(mu, logvar)
        return self.decoder(z), mu, logvar

model = VAE().to(device)
```

VAE: Using pytorch for MNIST dataset

- First sample with low number of epochs: The decoder is not able to cover the whole latent space.



- After improvements, the VAE methods helps us to morphs from one digit to another.



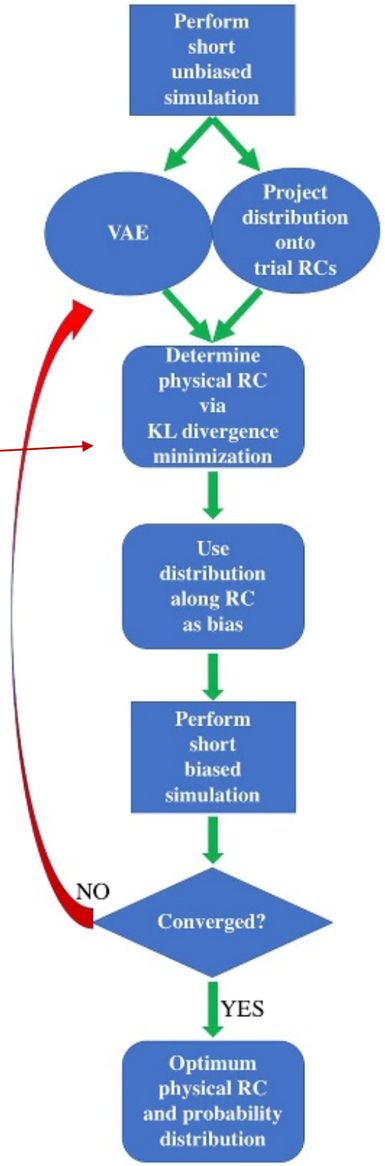
Reweighted autoencoded variational Bayes for enhanced sampling (RAVE)

J. Chem. Phys. 149, 072301 (2018); https://doi.org/10.1063/1.5025487

João Marcelo Lamim Ribeiro[1], Pablo Bravo[2,3], Yihang Wang[4], *and* Pratyush Tiwary[1]

- RAVE proceeds to use the probability distribution to construct the bias, $V_{bias}(\chi)$, for posteriors MD simulations as.

$$V_{bias}(\chi) = k_B T \log P^u(\chi) = k_B T \log \langle \delta(\chi - \chi(t)) \rangle$$



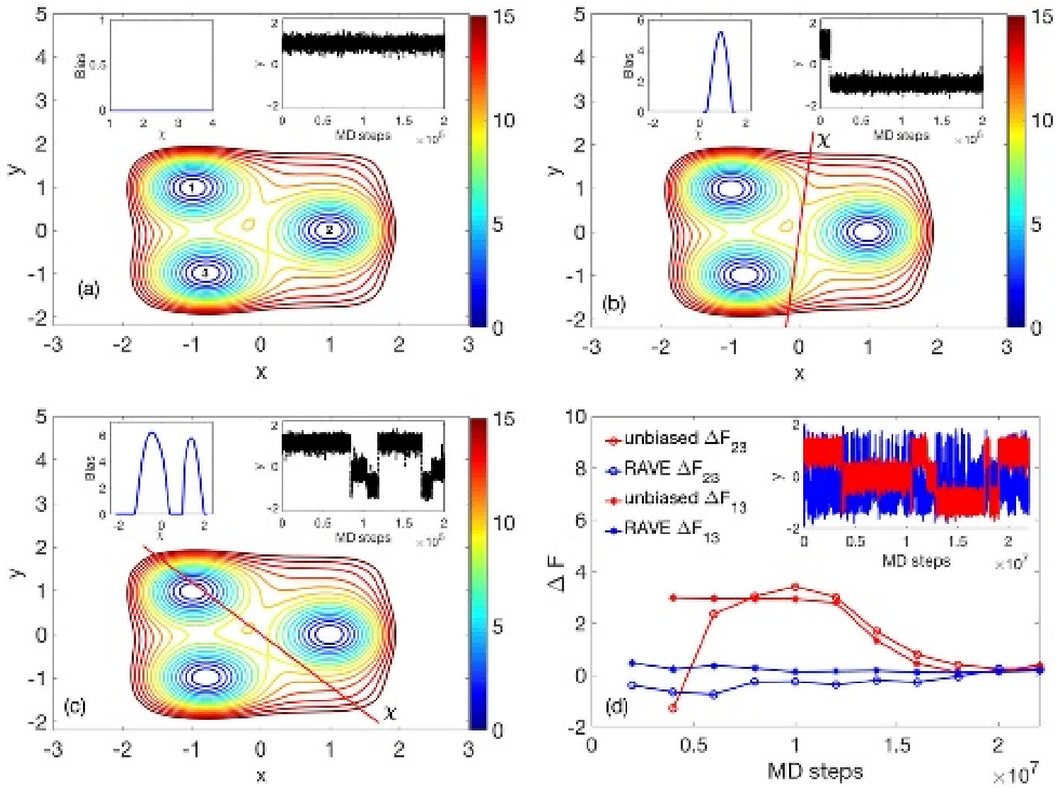
- Details of NN architecture: *Input layer*: The molecular dynamics (MD) trajectories, which for the two model potentials consists of 200 000 2-dimensional datapoints, while for the problem of fullerene unbinding consists of ~6000 3-dimensional datapoints.

- *Encoder hidden layers*: These first map each input MD datapoint into a sequence of three 512-dimensional vectors. Mean and variance as latent parameters.

- *Decoder hidden layers*: These first map a 1-dimensional latent variable, drawn from a Gaussian distribution using the parameters above, into a sequence of three 512-dimensional vectors.

RAVE: Applications

- Model three-state potential



- Hydrophilic ligand-cavity system in explicit water