https://arxiv.org/pdf/2208.11970.pdf

Generative Models

- the goal of a generatice model is, given samples x from a distribution of interesting p(x), to learn to model the true data distribution p(x)
- once learned, we can generate new samples from this approximate model at will
- and under some formulations, we can use the learned model to evaluate the likelihood of observed or sample data as well i.e. we can use g(x) ~ p(x) to estimate p(x|z)

There are several well known research directions

- GANs model the sampling procedure of a complex distribution, which is learned in an adversarial manner
- Likelihood based generative models seek to learn a model that assigns a high likelihood to the observed data samples
  - autoregressive models
  - normalizing flows
  - VAEs
- energy based models learn a distribution as an arbitrarily flexible energy function that is then normalized
  - Q: idk what this means
- score based generative models: instead of learning to model the energy function itself, ttry and learn the score of the energy-based model as a neural network
- diffusion models
  - combine both likelihood based and score based interpretations

Variational Autoencoders

In the classical variational autoencoder model, we directly maximize the ELBO.

$$ELBO(q(z))$$
$$= E_{q(z)}[log(p(z,\ x))] - E_{q(z)}[log(q(z))]$$
- $$= E_{q(z)}[log(p(x|z)p(z))] - E_{q(z)}[log(q(z))]$$
$$= E_{q(z)}[log(p(x|z))] + E_{q(z)}[log(p(z))] - E_{q(z)}[log(q(z))]$$
$$= E_{q(z)}[log(p(x|z))] - KL(q(z)||p(z))\ (3)$$
- In other words, ELBO = expected log likelihood of the data observed = reconstruction likelihood of the decoder - KL divergence between q(z) and the prior p(z) = prior matching term

In this case, we learn an intermediate blocking distribution $q_\phi(\mathbf{z}|\mathbf{x})$ that can be treated as an encoder: it transforms inputs into a distribution over possible latents.
Similarly, we learn a deterministic function $p_\theta(\mathbf{x}|\mathbf{z})$ to convert a given latent vector **z** into an

observation **x**, which can be interpreted as a decoder.

The reconstruction error term measures the reconstruction likelihood of the decoder from the variational distribution: this ensures that the learned distribution is modelling effective latents that the original data can be regenerated from

The second term mesaures how similar the learned variational distribution isa to a prior belief held over latent variables

- minimizing this term encourages the ecnoder to learn a distribution rather than collapse into a Dirac delta function

A defining feature of the VAE is how the ELBO is optimized jointly over both the parameters ϕ that feeds q_ϕ and θ that feeds p_θ
The encoder of the VAE is commonly chosen to model a multivariate Guassian with diagonal covariance, and the prior is often selected to be a standard multivariate Gaussian
$q_\phi(\mathbf{z}|\mathbf{x}) = N(\mathbf{z};\ \boldsymbol{\mu}_\phi(\mathbf{x}),\ \sigma^2_\phi(\mathbf{x})\mathbf{I}$
$p(\mathbf{z}) = N(\mathbf{z};\ \mathbf{0},\ \mathbf{I})$

Then the KL divergence term of the ELBO can be computed analytically, and the reconstruction term can be approximated using a Monte Carlo estimate

$$\arg\max_{\boldsymbol{\phi},\boldsymbol{\theta}} \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\right] - D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \parallel p(\boldsymbol{z})) \approx \arg\max_{\boldsymbol{\phi},\boldsymbol{\theta}} \sum_{l=1}^{L} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}^{(l)}) - D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \parallel p(\boldsymbol{z})) \quad (2$$

Question: how does this framing of the reconstruction term lead to the sum of squared error implementation used in the VAE implementation we seein mober?

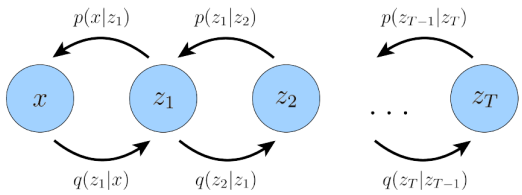
The reparameterization trick rewrites a random variable as a deterministic function of a noise variable -- this allows for the optimization of the non-stochastic terms fhrough gradient descent
- we can sample from an arbitrary Gaussian by sampling from a standard Gaussian and scaling it

Hierarchical Variational Autoencoders

This is a generalization of a VAE that extends to multiple hierarchies over latent variables. Under this formulation, latent variables themselves are interpreted as generatived from other higher-level, more abstract latents.

In the general HVAE with T hierarchical levels, each latent is allowed to condition on all previos latest, in this case we focus on a special case -- called a Markovian HVAE.
In a MHVAE, the generative process is a Markov chain -- that is, each transition down the hierarchy is Markovian, where decoding each latent z_t only conditions on previous latent z_{t+1}.

Intuitively, and visually, this can eb seen as simply stakcing VAEs on top of each other -- so another appropriate term to describe this setup may be a Recursive VAE.



Mathematically, the joint distribution and posterior of a Markovian HVAE as

$$p(\mathbf{x},\ \mathbf{z}_{1:T}) \ = \ p(\mathbf{z}_T)p_\theta(\mathbf{x}|\mathbf{z}_1)\prod_{t=2}^{T} p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t)$$

$$q_\phi(\mathbf{z}_{1:T}|\mathbf{x}) \ = \ q_\phi(\mathbf{z}_1|\mathbf{x})\prod_{t=2}^{T} q_\phi(\mathbf{z}_t|\mathbf{z}_{t-1})$$

Then the ELBO can be easily extended to be

$$\log p(\mathbf{x}) \ = \ \log \int p(\mathbf{x},\ \mathbf{z}_{1:T})\, d\mathbf{z}_{1:T} \ \geqslant \ \mathrm{E}_{q_\phi(\mathbf{z}_{1:T}|\mathbf{x})}\left[\log \frac{p(\mathbf{x},\ \mathbf{z}_{1:T})}{q_\phi(\mathbf{z}_{1:T}|\mathbf{x})}\right]$$

$$= \quad \mathrm{E}_{q_\phi(\mathbf{z}_{1:T}|\mathbf{x})}\left[\log \frac{p(\mathbf{z}_T)p_\theta(\mathbf{x}|\mathbf{z}_1)\prod_{t=2}^{T} p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t)}{q_\phi(\mathbf{z}_1|\mathbf{x})\prod_{t=2}^{T} q_\phi(\mathbf{z}_t|\mathbf{z}_{t-1})}\right]$$

This objective can be further decomposed into interpretable components.

Variational Diffusion Models

A VDM is a MHVAE with three key restrictions.

1. The latent dimension is exactly equal to the data dimension
2. The structure of the latent encoder at each timestep is not learned. It is pre-defined as a linear Guassian model. In other words, it is a Gaussian distribution centred around the output

of the previous time step.
3. The Gaussian parameters of the latent encoders vary over time in such a way that the distribution of the latent at final timestep T is a standard Gaussian.

Further, we explicitly maintain the Markov property between hierarchical transitions from a standard MHVAE.

What do these assumptions mean in practice?

From the first assumption: with some abuse of notation, we can now represent both true data samples and latent variables as $\mathbf{x}_t$, where $t = 0$ represents true data samples and $t \, \epsilon \, [1, \, T]$ represents a corresponding latent with hierarchy indexed by t.

So the VDM posterior is the same as the MHVAE posterior, but can be rewritten as:

$$q_\phi(\mathbf{z}_{1:T}|\mathbf{x}) \; = \; q_\phi(\mathbf{z}_1|\mathbf{x}) \prod_{t=2}^{T} q_\phi(\mathbf{z}_t|\mathbf{z}_{t-1}) \implies q(\mathbf{x}_{1:T}|\mathbf{x}_0) \; = \; \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})$$

From the second assumption, we know that the distribution of each latent variable in the endoer is a Gaussian centred around its previous hierarchical latent. Unlike a MHVAE, the structure of the encoder at each timestep t is not learned -- it is fixed as a linear Gaussian model, where the mean and standard deviation can be set before hand as hyperparams, or learned as parameters.

We paramaterize the Gaussian encoder with mean $\boldsymbol{\mu}_t(\mathbf{x}_t) \; = \; \sqrt{\alpha_t}\mathbf{x}_{t-1}$ and variance $\boldsymbol{\Sigma}_t(\mathbf{x}_t) \; = \; (1 \, - \, \alpha_t)\mathbf{I}$, where the form of the coefficients are chosen so that the variance of the latent variables stay at a similar scale -- in other words, the encoding process is variance preserving.
So, the encoder transitions are denoted as

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) \; = \; N\Big(\mathbf{x}_t; \sqrt{\alpha_t}\mathbf{x}_{t-1}, \, (1 \, - \, \alpha_t)\mathbf{I}\Big)$$

From the third assumption, we know that α_t evolves over time according to a fixed or learnable schedule structured such that the distribution of the final latent $p(\mathbf{x}_T)$ is a standard Gaussian. That is,

$$p(\mathbf{x}_{0:T}) \; = \; p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) \text{ where } p(\mathbf{x}_T) \; = \; N(\mathbf{x}_T; \, \mathbf{0}, \, \mathbf{I})$$

This set of assumptions collectively describes a steady noisification of an image input over time -- we progressively corrupt an image by adding Gaussian noise until it becomes equivalent to pure Gaussian noise.
Note that the encoder distributions $q(\mathbf{x}_t|\mathbf{x}_{t-1})$ are no longer parameterized by a set of

parameters ϕ since they are completely specified as Gaussians with defined mean and variance at each timestep t.
Therefore, with a VDM, we are only interested in learning the conditionals $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ so that we can simulate new data. After the VDM has been optimized, the sampling procedure is:
1. sample Gaussian noise from $p(\mathbf{x}_T)$
2. iteratively run the denoising transitions $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ to generate a new $\mathbf{x}_0$

As with any other HVAE, the VDM can be optimized by maximizing the evidence $\log p(\mathbf{x})$ which in turn can be done by maximizing the ELBO.