Coming back to generating a signal, instead of training an autoencoder and modeling the distribution of $Z$, we can try an alternative approach:

**Impose a distribution for $Z$** and then train a decoder $g$ so that $g(Z)$ matches the training data.
We consider the following two distributions:

- \( p \) is the distribution on \( \mathcal{X} \times \mathbb{R}^d \) of a pair \((X, Z)\) composed of an encoding state \( Z \sim \mathcal{N}(0, I) \) and the output of the decoder \( g \) on it.

- \( q \) is the distribution on \( \mathcal{X} \times \mathbb{R}^d \) of a pair \((X, Z)\) composed of a sample \( X \) taken from the data distribution and the output of the encoder on it.

Our goal is that \( p(X) \) mimics the data-distribution \( q(X) \), that is to find \( g \) that maximizes the log-likelihood

\[
\frac{1}{N} \sum_n \log p(x_n) = \hat{E}_{q(X)} \left[ \log p(X) \right].
\]

However, with a complicated \( g \), we can sample \( z \) and compute \( g(z) \), but cannot compute \( p(x) \) for a given \( x \), and even less compute its derivatives.

The Variational Autoencoder proposed by Kingma and Welling (2013) relies on a tractable approximation of this log-likelihood.

Note that their framework involves stochastic encoder \( f \), and decoder \( g \), whose outputs depend on both their inputs and additional randomness.
Remember that $q(X)$ is the data distribution, and $q(Z \mid X = x)$ is the distribution of the latent encoding $f(x)$. We want to maximize

$$
E_{q(X)} \left[ \log p(X) \right],
$$

and we can show that

$$
- E_{q(X)} \left[ \log p(X) \right] \leq E_{q(X)} \left[ D_{KL}(q(Z \mid X) \| p(Z)) \right] - E_{q(X,Z)} \left[ \log p(X \mid Z) \right].
$$

So it makes sense to minimize this latter quantity.

So the final loss is

$$
\mathcal{L} = E_{q(X)} \left[ D_{KL}(q(Z \mid X) \| p(Z)) \right] - E_{q(X,Z)} \left[ \log p(X \mid Z) \right].
$$

with

- $q(X)$ is the data distribution
- $p(Z) = \mathcal{N}(0, I)$.

Kingma and Welling propose that both the encoder $f$ and decoder $g$ map to a Gaussian with diagonal covariance. Hence they map to twice the dimension (e.g. $f(x) = (\mu^f(x), \sigma^f(x)))$ and

- $q(Z \mid X = x) \sim \mathcal{N}(\mu^f(x), \text{diag}(\sigma^f(x)))$
- $p(X \mid Z = z) \sim \mathcal{N}(\mu^g(z), \text{diag}(\sigma^g(z)))$. 
The first term of $\mathcal{L}$ is the average of

$$
\mathbb{D}_{KL}(q(Z | X = x) \| p(Z)) = -\frac{1}{2} \sum_d \left( 1 + 2 \log \sigma_d^f(x) - \left( \mu_d^f(x) \right)^2 - \left( \sigma_d^f(x) \right)^2 \right)
$$

over the $x_n$s.

This can be implemented as

```python
param_f = model.encode(input)
mu_f, logvar_f = param_f.split(param_f.size(1)//2, 1)
kl = -0.5 * (1 + logvar_f - mu_f.pow(2) - logvar_f.exp())
kl_loss = kl.sum() / input.size(0)
```
As Kingma and Welling (2013), we use a constant variance of 1 for the decoder, so the second term of $\mathcal{L}$ becomes the average of

$$-\log p(X = x \mid Z = z) = \frac{1}{2} \sum_d (x_d - \mu^g_d(z))^2 + \text{cst}$$

over the $x_n$, with one $z_n$ sampled for each, i.e.

$$z_n \sim \mathcal{N} \left( \mu^f(x_n), \sigma^f(x_n) \right), \quad n = 1, \ldots, N.$$

This can be implemented as

```python
std_f = torch.exp(0.5 * logvar_f)
z = torch.empty_like(mu_f).normal_() * std_f + mu_f
output = model.decode(z)
fit = 0.5 * (output - input).pow(2)
fit_loss = fit.sum() / input.size(0)
```
We had for the standard autoencoder

\[
z = \text{model.encode}(\text{input}) \\
\text{output} = \text{model.decode}(z) \\
\text{loss} = 0.5 \times (\text{output} - \text{input})^2 \sum / \text{input.size(0)}
\]

and putting everything together we get for the VAE

\[
\text{param}_f = \text{model.encode}(\text{input}) \\
\mu_f, \text{logvar}_f = \text{param}_f.\text{split}() \text{param}_f.\text{size(1)}//2, 1) \\
\text{kl} = -0.5 \times (1 + \text{logvar}_f - \mu_f^2 - \text{logvar}_f.\exp()) \\
\text{kl}\_\text{loss} = \text{kl}\_\text{sum}() / \text{input.size(0)} \\
\text{std}_f = \text{torch.exp}(0.5 * \text{logvar}_f) \\
\text{z} = \text{torch.empty_like}(\mu_f).\text{normal_()} \times \text{std}_f + \mu_f \\
\text{output} = \text{model.decode}(\text{z}) \\
\text{fit} = 0.5 \times (\text{output} - \text{input})^2 \sum \\
\text{fit}\_\text{loss} = \text{fit}\_\text{sum}() / \text{input.size(0)} \\
\text{loss} = \text{kl}\_\text{loss} + \text{fit}\_\text{loss}
\]

During inference we do not sample, and instead use $\mu_f$ and $\mu_g$ as prediction.

Note in particular the re-parameterization trick:

\[
z = \text{torch.empty_like}(\mu_f).\text{normal_()} \times \text{std}_f + \mu_f \\
\text{output} = \text{model.decode}(\text{z})
\]

Implementing the sampling of $z$ that way allows to compute the gradient w.r.t $f$’s parameters without any particular property of \text{normal_()}.
We can look at two latent features to check that they are Normal for the VAE.
Autoencoder sampling ($d = 32$)

Variational Autoencoder sampling ($d = 32$)

Making the embedding $\sim \mathcal{N}(0, 1)$, often results in “disentangled” representations.

This effect can be reinforced with a greater weight of the KL term

$$\mathcal{L} = \beta \mathbb{E}_{q(X)}\left[\mathbb{D}_{\text{KL}}(q(Z \mid X) \parallel p(Z))\right] - \mathbb{E}_{q(X, Z)}\left[\log p(X \mid Z)\right],$$

resulting in the $\beta$-VAE proposed by Higgins et al. (2017).
We propose augmenting the original VAE framework with a single hyperparameter $\beta$ that corresponds to the original VAE framework (Kingma & Welling, 2014; Rezende et al., 2014), which brings scalability and training stability. While the original VAE work has been shown to achieve limited disentangling performance at the cost of training instability and reduced sample diversity. Furthermore, InfoGAN requires some a priori knowledge of the data, since its performance is sensitive to the choice of the prior distribution and the number of the regularised noise variables. InfoGAN also lacks a principled protocol to quantitatively compare the degree of disentanglement learnt by different models; 3) we demonstrate both qualitatively and quantitatively that our framework to quantitatively compare the degree of disentanglement learnt by different models.

Our main contributions are the following: 1) we propose using unsupervised learning for developing more human-like learning and reasoning in algorithms as right direction, we believe that further improvements are necessary to achieve a principled way of transfer learning or zero-shot inference scenarios. Hence, while InfoGAN is an important step in the outperforms all our baselines on this measure (ICA, PCA, VAE Kingma & Ba (2014), DC-IGN Kulkarni et al., 2015), prompting the development of more elaborate semi-supervised VAE-based approaches for disentangled factor learning on a number of benchmark datasets, such as CelebA (Liu et al., 2015), chairs (Aubry et al., 2014; Paysan et al., 2009; Liu et al., 2015), and faces (Paysan et al., 2009) using qualitative evaluation. Finally, to help quantify the differences, we develop a new measure of disentanglement and show that InfoGAN also lacks a principled way of developing more human-like learning and reasoning in algorithms as right direction, we believe that further improvements are necessary to achieve a principled way of transfer learning or zero-shot inference scenarios.

Recently a scalable unsupervised approach for disentangled factor learning has been developed, that are independent. We show that this simple modification allows $\beta$-VAE to significantly improve the degree of disentanglement achieved by different models or when optimising the hyperparameters of a single model. $\beta$-VAE achieves state of the art disentangling performance against both the best unsupervised (InfoGAN: Chen et al., 2016) and semi-supervised (DC-IGN: Kulkarni et al., 2015) approaches for disentangled factor learning on a number of benchmark datasets, such as CelebA (Liu et al., 2015), chairs (Aubry et al., 2014) and faces (Paysan et al., 2009) using qualitative evaluation. Finally, to help quantify the differences, we develop a new measure of disentanglement and show that $\beta$-VAE significantly outperforms all our baselines on this measure (ICA, PCA, VAE Kingma & Welling, 2014, DC-IGN: Kulkarni et al., 2015, and faces (Paysan et al., 2009) using qualitative evaluation. Finally, to help quantify the differences, we develop a new measure of disentanglement and show that $\beta$-VAE significantly outperforms all our baselines on this measure (ICA, PCA, VAE Kingma & Welling, 2014, DC-IGN: Kulkarni et al., 2015, and InfoGAN).
References
