Machine Learning 2020

Volker Roth

Department of Mathematics & Computer Science
University of Basel

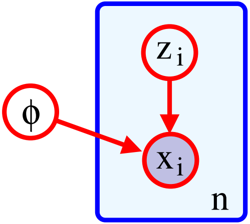
25th May 2020

Section 11

Non-linear latent variable models

Non-linear latent variable models

Latent variable $\boldsymbol{z} \rightsquigarrow$ Gaussian likelihood with **nonlinearly transformed mean** $\boldsymbol{\mu} = f(\boldsymbol{z}, \phi)$.

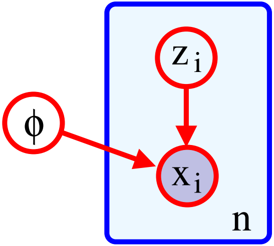


Prior and likelihood:

$$p(\boldsymbol{z}) = N(\boldsymbol{0}, I)$$
$$p(\boldsymbol{x}|\boldsymbol{z}, \phi) = N(\boldsymbol{f}(\boldsymbol{z}, \phi), \sigma^2 I).$$

- Given observed $\boldsymbol{x}$, we want to understand what possible values of the hidden variable $\boldsymbol{z}$ were responsible for it:
$$p(\boldsymbol{z}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})}{p(\boldsymbol{x})}.$$
- No closed form expression available. Cannot evaluate denominator $p(\boldsymbol{x})$ and so we can't even compute the numerical value of the posterior for a given pair $\boldsymbol{z}$ and $\boldsymbol{x}$.

Sampling



- ...but it is easy to generate a new sample $\boldsymbol{x}^*$ using sampling:
  - Draw $\boldsymbol{z}^*$ from the prior $p(\boldsymbol{z})$, pass this through $f(\boldsymbol{z}^*, \phi)$ $\rightsquigarrow$ mean of likelihood $p(\boldsymbol{x}^*|\boldsymbol{z}^*)$,
  - then draw $\boldsymbol{x}^*$ from this distribution.
- Prior and likelihood are normal distributions $\rightsquigarrow$ sampling is easy.

Evaluating marginal likelihood (evidence)

$$
\begin{aligned}
p(\boldsymbol{x}|\phi) &= \int p(\boldsymbol{x}, \boldsymbol{z}|\phi) d\boldsymbol{z} \\
&= \int p(\boldsymbol{x}|\boldsymbol{z}, \phi) p(\boldsymbol{z}) d\boldsymbol{z} \\
&= \int N(\boldsymbol{f}[\boldsymbol{z}, \phi], \sigma^2 I) \cdot N(\boldsymbol{0}, I)\, d\boldsymbol{z}.
\end{aligned}
$$

No closed form for this integral $\rightsquigarrow$ lower bound (Jensen's inequality):

$$
\begin{aligned}
\log[p(\boldsymbol{x}|\phi)] &= \log\left[\int p(\boldsymbol{x}, \boldsymbol{z}|\phi) d\boldsymbol{z}\right] \\
&= \log\left[\int q(\boldsymbol{z}) \frac{p(\boldsymbol{x}, \boldsymbol{z}|\phi)}{q(\boldsymbol{z})} d\boldsymbol{z}\right] \\
&\geq \int q(\boldsymbol{z}) \log\left[\frac{p(\boldsymbol{x}, \boldsymbol{z}|\phi)}{q(\boldsymbol{z})}\right] d\boldsymbol{z},
\end{aligned}
$$

Known as the evidence lower bound ELBO, because $p(\boldsymbol{x}|\phi)$ is the evidence (= marginal likelihood) in the context of Bayes' rule.

ELBO

- In practice, the distribution $q(\boldsymbol{z})$ will have some parameters $\boldsymbol{\theta}$:
$$\text{ELBO}[\boldsymbol{\theta}, \phi] = \int q(\boldsymbol{z}|\boldsymbol{\theta}) \log \left[\frac{p(\boldsymbol{x}, \boldsymbol{z}|\phi)}{q(\boldsymbol{z}|\boldsymbol{\theta})} \right] d\boldsymbol{z}.$$
- To learn the non-linear latent variable model, we'll maximize this quantity as a function of both ϕ and $\boldsymbol{\theta}$.
- We will see: the maximum is obtained (theoretically) if the variational distribution is the true posterior, $q(\boldsymbol{z}|\boldsymbol{\theta}) = p(\boldsymbol{z}|\boldsymbol{x}, \phi)$.
- In practice, we maximize ELBO over some tractable family of distributions $q(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta})$ to obtain an approximation of the intractable posterior.
- The neural architecture that computes this is the **variational autoencoder.**

Tightness of ELBO

$$
\begin{aligned}
\mathrm{ELBO}[\boldsymbol{\theta}, \phi] &= \int q(\boldsymbol{z}|\boldsymbol{\theta}) \log \left[\frac{p(\boldsymbol{x}, \boldsymbol{z}|\phi)}{q(\boldsymbol{z}|\boldsymbol{\theta})} \right] d\boldsymbol{z} \\
&= \int q(\boldsymbol{z}|\boldsymbol{\theta}) \log \left[\frac{p(\boldsymbol{z}|\boldsymbol{x}, \phi) p(\boldsymbol{x}|\phi)}{q(\boldsymbol{z}|\boldsymbol{\theta})} \right] d\boldsymbol{z} \\
&= \int q(\boldsymbol{z}|\boldsymbol{\theta}) \log [p(\boldsymbol{x}|\phi)] \, d\boldsymbol{z} + \int q(\boldsymbol{z}|\boldsymbol{\theta}) \log \left[\frac{p(\boldsymbol{z}|\boldsymbol{x}, \phi)}{q(\boldsymbol{z}|\boldsymbol{\theta})} \right] d\boldsymbol{z} \\
&= \log[p(\boldsymbol{x}|\phi)] + \int q(\boldsymbol{z}|\boldsymbol{\theta}) \log \left[\frac{p(\boldsymbol{z}|\boldsymbol{x}, \phi)}{q(\boldsymbol{z}|\boldsymbol{\theta})} \right] d\boldsymbol{z} \\
&= \log[p(\boldsymbol{x}|\phi)] - \mathrm{D}_{KL} \left[q(\boldsymbol{z}|\boldsymbol{\theta}) \| p(\boldsymbol{z}|\boldsymbol{x}, \phi) \right].
\end{aligned}
$$

ELBO is the log marginal likelihood minus $\mathrm{D}_{KL}[q(\boldsymbol{z}|\boldsymbol{\theta}) \| p(\boldsymbol{z}|\boldsymbol{x}, \phi)]$.
D_{KL} zero when $q(\boldsymbol{z}|\boldsymbol{\theta}) = p(\boldsymbol{z}|\boldsymbol{x}, \phi) \rightsquigarrow \mathrm{ELBO} = \log[p(\boldsymbol{x}|\phi)]$.

ELBO as reconstruction loss minus KL to prior

$$
\begin{aligned}
\mathrm{ELBO}[\boldsymbol{\theta}, \phi] &= \int q(\boldsymbol{z}|\boldsymbol{\theta}) \log \left[\frac{p(\boldsymbol{x}, \boldsymbol{z}|\phi)}{q(\boldsymbol{z}|\boldsymbol{\theta})} \right] d\boldsymbol{z} \\
&= \int q(\boldsymbol{z}|\boldsymbol{\theta}) \log \left[\frac{p(\boldsymbol{x}|\boldsymbol{z}, \phi) p(\boldsymbol{z})}{q(\boldsymbol{z}|\boldsymbol{\theta})} \right] d\boldsymbol{z} \\
&= \int q(\boldsymbol{z}|\boldsymbol{\theta}) \log \left[p(\boldsymbol{x}|\boldsymbol{z}, \phi) \right] d\boldsymbol{z} + \int q(\boldsymbol{z}|\boldsymbol{\theta}) \log \left[\frac{p(\boldsymbol{z})}{q(\boldsymbol{z}|\boldsymbol{\theta})} \right] d\boldsymbol{z} \\
&= \int q(\boldsymbol{z}|\boldsymbol{\theta}) \log \left[p(\boldsymbol{x}|\boldsymbol{z}, \phi) \right] d\boldsymbol{z} - \mathrm{D}_{KL}[q(\boldsymbol{z}|\boldsymbol{\theta}), p(\boldsymbol{z})]
\end{aligned}
$$

- First term measures the average agreement $p(\boldsymbol{x}|\boldsymbol{z}, \phi)$ of the hidden variable and the data (reconstruction loss)
- Second one measures the degree to which the auxiliary distribution $q(\boldsymbol{z}, \boldsymbol{\theta})$ matches the prior.

The variational approximation

- ELBO is tight when we choose $q(\boldsymbol{z}|\boldsymbol{\theta}) = p(\boldsymbol{z}|\boldsymbol{x}, \phi)$.
- Intractable $\leadsto$ variational approximation: choose simple parametric form for $q(\boldsymbol{z}|\boldsymbol{\theta})$, use it as an approximation to the true posterior.
- Choose a normal distribution with parameters $\boldsymbol{\mu}$ and $\Sigma = \sigma^2 I$.
- Optimization $\leadsto$ find normal distribution closest to true posterior $p(\boldsymbol{z}|\boldsymbol{x})$. Corresponds to minimizing the KL divergence.
- True posterior $p(\boldsymbol{z}|\boldsymbol{x})$ depends on $\boldsymbol{x}$
  $\leadsto$ variational approximation should also depend on $\boldsymbol{x}$:

$$q(\boldsymbol{z}|\boldsymbol{\theta}, \boldsymbol{x}) = N(g_\mu[\boldsymbol{x}|\boldsymbol{\theta}], g_{\sigma^2}[\boldsymbol{x}|\boldsymbol{\theta}]),$$

where $g[\boldsymbol{x}, \boldsymbol{\theta}]$ is a neural network with parameters $\boldsymbol{\theta}$.

The variational autoencoder

Recall

$$\text{ELBO}[\boldsymbol{\theta}, \phi] = \int q(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta}) \log \left[p(\boldsymbol{x}|\boldsymbol{z}, \phi)\right] d\boldsymbol{z} - \mathrm{D}_{KL}[q(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta}), p(\boldsymbol{z})]$$

Involves an intractable integral, but it is an expectation
$\rightsquigarrow$ approximate with samples:

$$E_{q(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta})}[\log \left[p(\boldsymbol{x}|\boldsymbol{z}, \phi)\right]] \approx \frac{1}{N} \sum_{n=1}^{N} \log \left[p(\boldsymbol{x}|\boldsymbol{z}_n^*, \phi)\right]$$

where $\boldsymbol{z}_n^*$ is the n-th sample from $q(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta})$. Limit: use a single sample:

$$\text{ELBO}[\boldsymbol{\theta}, \phi] \approx \quad \log \left[p(\boldsymbol{x}|\boldsymbol{z}^*, \phi)\right] - \mathrm{D}_{KL}[q(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta}), p(\boldsymbol{z})]$$

The second term is just the KL divergence between two Gaussians and is available in closed form.

The reparameterization trick

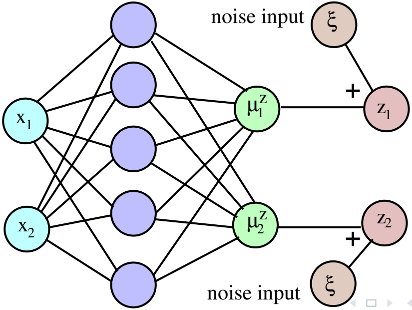
Recall: Want to sample from

$$q(\boldsymbol{z}|\boldsymbol{\theta}, \boldsymbol{x}) = N(g_{\boldsymbol{\mu}}[\boldsymbol{x}|\boldsymbol{\theta}], g_{\sigma^2}[\boldsymbol{x}|\boldsymbol{\theta}]),$$

To let PyTorch / Tensorflow perform automatic differentiation via backpropagation, we must avoid the sampling step.
Simple solution: draw a sample $\boldsymbol{\xi} \sim N(0, I)$ and use

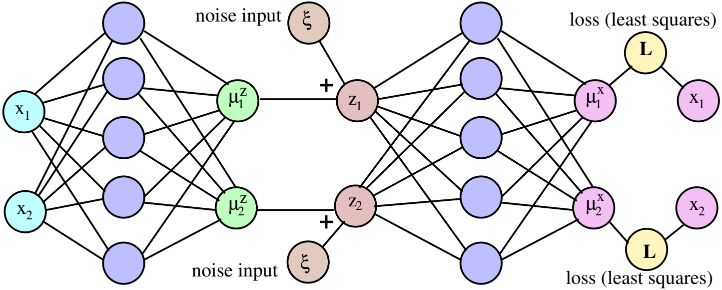
$$\boldsymbol{z}^* = g_{\boldsymbol{\mu}} + \sigma^{1/2}\boldsymbol{\xi}.$$

Now "the gradient can flow through the network". **Encoder network:**



VAE

- Finally, minimize negative expectation of ELBO over $p(\boldsymbol{x})$:
$$\min_{\phi,\boldsymbol{\theta}} -E_{p(\boldsymbol{x})}E_{q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\theta})}[\log[p(\boldsymbol{x}|\boldsymbol{z},\phi)]] + E_{p(\boldsymbol{x})}\mathrm{D}_{KL}[q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\theta}),p(\boldsymbol{z})]$$
- The first term is approximated as
$$E_{p(\boldsymbol{x})}E_{q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\theta})}[\log[p(\boldsymbol{x}|\boldsymbol{z},\phi)]] \approx \frac{1}{n}\sum_{i=1}^{n}\log[p(\boldsymbol{x}_i|\boldsymbol{z}_i^*,\phi)].$$
We assume $p(\boldsymbol{x}_i|\boldsymbol{z}_i^*,\phi) = \mathcal{N}(f_\phi(\boldsymbol{z}_i^*),\sigma^2)$,
where f is implemented via a neutral net: $\rightsquigarrow$ **Decoder network**

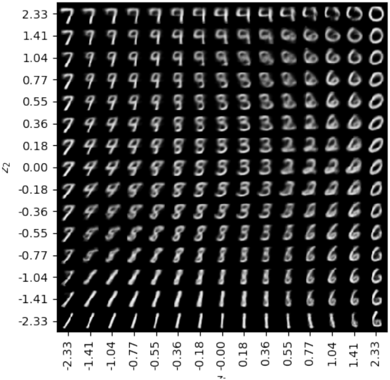


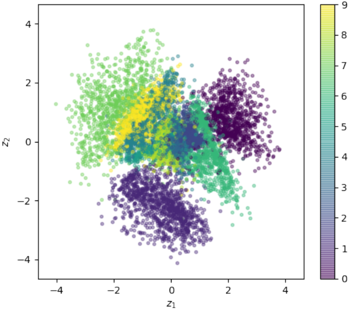
Further Variations

- For maximizing ELBO, we jointly optimize over the parameters of encoder and decoder network.
- When adjusting the decoder, we also change the "true" posterior that we are going to approximate!
- So approximation quality should not be our only goal...
  need "tuning knob" for steering the model into a desired direction.
- Solution: introduce parameter $\beta > 0$ that controls the relative importance of the two loss terms:

$$\min_{\boldsymbol{\theta},\phi} \frac{1}{n}\sum_{i=1}^{n} \mathrm{D}_{KL}[q(\boldsymbol{z}|\boldsymbol{x}_i,\boldsymbol{\theta}), p(\boldsymbol{z})] - \beta\frac{1}{n}\sum_{i=1}^{n} \log\left[p(\boldsymbol{x}_i|\boldsymbol{z}_i^*,\phi)\right]$$

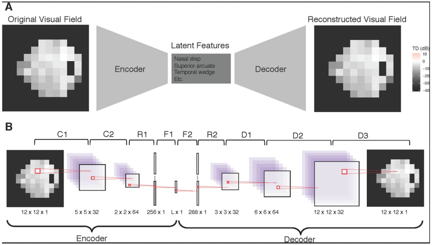
Applications: MNIST example

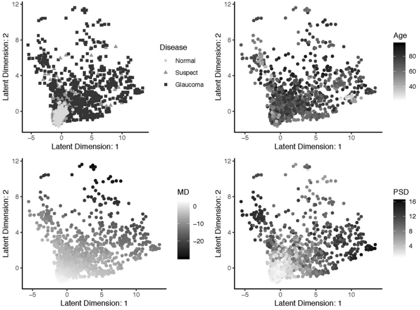




Taken from Louis Tiago: A Tutorial on Variational Autoencoders with a Concise Keras Implementation

Applications: Medical example

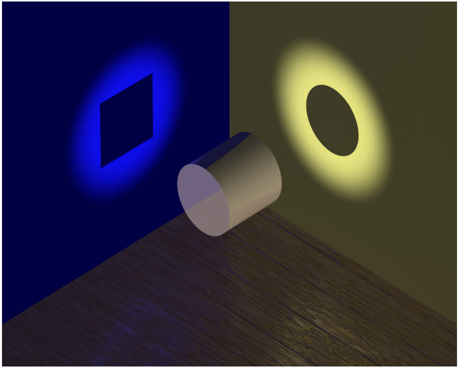




Berchuck, S.I., Mukherjee, S. & Medeiros, F.A. Estimating Rates of Progression and Predicting Future Visual Fields in Glaucoma Using a Deep Variational Autoencoder. Sci Rep 9, 18113 (2019). https://doi.org/10.1038/s41598-019-54653-6

Multiple Views: Deep Information Bottleneck

- Consider paired samples from different views.
- What is the dependency structure between the views ?
- Nonlinear model: dependency detected by deep IB.



Two-view version: The deep information bottleneck

- So far we argued that since the true posterior $p(\boldsymbol{z}|\boldsymbol{x})$ depends on $\boldsymbol{x}$, the variational approximation should also depend on $\boldsymbol{x}$.
- Restricted setting: explain posterior **only by external variable $\tilde{\boldsymbol{x}}$**:

$$q = q(\boldsymbol{z}|\boldsymbol{\theta}, \tilde{\mathbf{x}}).$$

$$\begin{aligned} \text{ELBO}[\boldsymbol{\theta}, \phi] &= \int q(\boldsymbol{z}|\tilde{\mathbf{x}}, \boldsymbol{\theta}) \log\left[p(\boldsymbol{x}|\boldsymbol{z}, \phi)\right] d\boldsymbol{z} - \mathrm{D}_{KL}[q(\boldsymbol{z}|\tilde{\mathbf{x}}, \boldsymbol{\theta}), p(\boldsymbol{z})] \\ &= E_{q(\boldsymbol{z}|\tilde{\mathbf{x}}, \boldsymbol{\theta})} \log\left[p(\boldsymbol{x}|\boldsymbol{z}, \phi)\right] - \mathrm{D}_{KL}[q(\boldsymbol{z}|\tilde{\mathbf{x}}, \boldsymbol{\theta}), p(\boldsymbol{z})] \end{aligned}$$

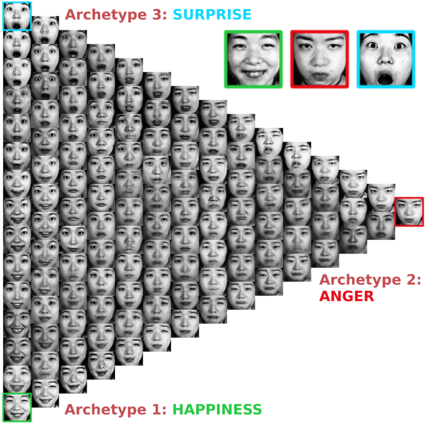
- Connection to IB:
  - Assume (or define) $q(\boldsymbol{z}|\tilde{\mathbf{x}}, \boldsymbol{\theta}) := p(\boldsymbol{z}|\tilde{\mathbf{x}}, \boldsymbol{\theta})$
  - Take expectation w.r.t. joint data distribution $p(\tilde{\mathbf{x}}, \boldsymbol{x})$:

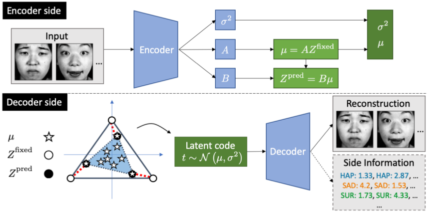
$$E_{p(\tilde{\mathbf{x}}, \boldsymbol{x})} E_{p(\boldsymbol{z}|\tilde{\mathbf{x}}, \boldsymbol{\theta})} \log\left[p(\boldsymbol{x}|\boldsymbol{z}, \phi)\right] - E_{p(\tilde{\mathbf{x}})} \mathrm{D}_{KL}[p(\boldsymbol{z}|\tilde{\mathbf{x}}, \boldsymbol{\theta}), p(\boldsymbol{z})]$$

  - First term $\leq \mathcal{I}_{\boldsymbol{\theta}, \phi}(\boldsymbol{z}; \boldsymbol{x}) + const$. Second term $= \mathcal{I}_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}; \boldsymbol{z})$,
- This defines the deep information bottleneck (with weight β)

$$\min_{\phi, \boldsymbol{\theta}} \mathcal{I}_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}; \boldsymbol{z}) - \beta \mathcal{I}^{\text{low}}_{\boldsymbol{\theta}, \phi}(\boldsymbol{z}; \boldsymbol{x}), \quad \text{where } \mathcal{I}^{\text{low}} \text{ is a lower bound of } \mathcal{I}.$$

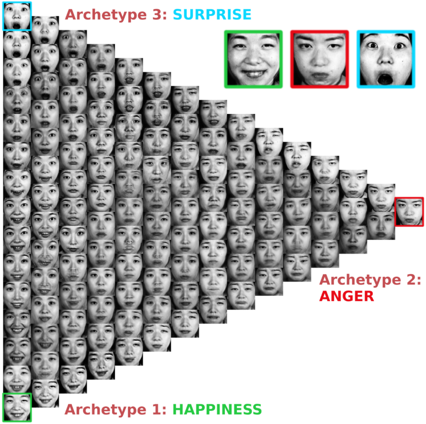
Applications: Face images

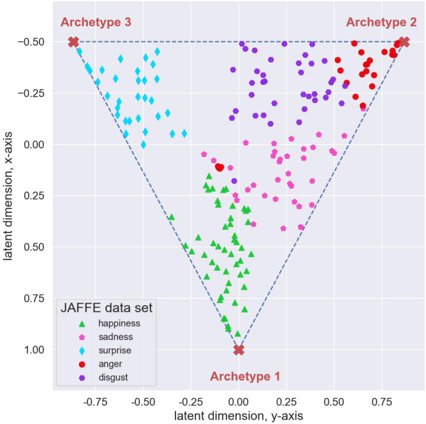




Keller et al. 2020: Learning Extremal Representations with Deep Archetypal Analysis

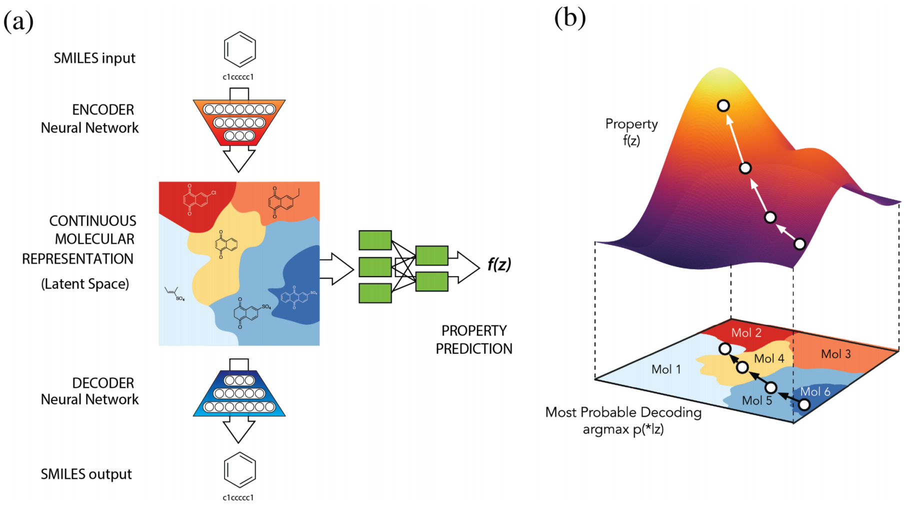
Applications: Face images





Keller et al. 2020: Learning Extremal Representations with Deep Archetypal Analysis

Applications: Deep Chemical Variational Autoencoders



(Gomez-Bombarelli et al., ACS Cent Sci, 2018)