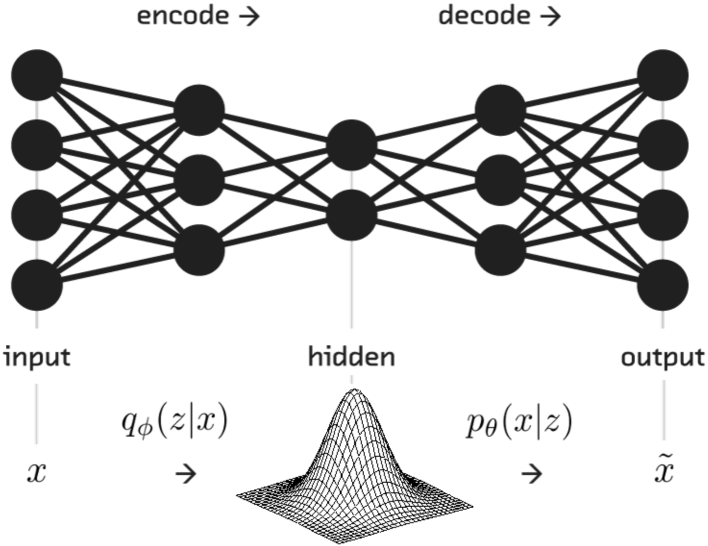
Interpolating and Extrapolating in Latent Space

Ryan Pyle, ACMS

April 23, 2019

Latent Space



Latent Space

A latent space should be

- Simple - lower dimensional
- Complete - contain sufficient information
- Intuitive - promote understanding

To what extent does the VAE latent space satisfy these goals?

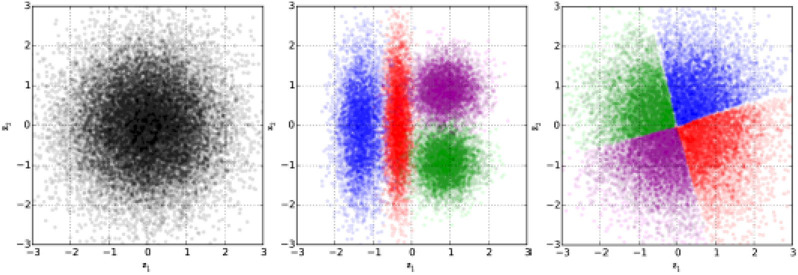
VAE Latent Space Issues

VAE assumes that each latent dimension is independently Gaussian distributed.

It is possible to fit to other distributions, but the theory may not fit as well, and this requires us to know a 'good' target distribution.

Instead, use auto-regressive flow and treat final probability density as a learned, parameterized part of our model.

VAE Latent Space



(a) Prior distribution (b) Posteriors in standard VAE (c) Posteriors in VAE with IAF

Auto-Regressive Flow

The idea has been developed throughout several papers. One of the earliest is "NICE: Non-Linear Independent Components Estimation" by Dinh, Krueger, and Bengio.

Consider a change of variables $h = f(x)$, which assumes that f is invertible and the dimension of h is the same as the dimension of x, in order to fit a distribution p_H. This gives us

$$P_X(x) = P_H(f(x))|det\frac{df(x)}{dx}|$$

Training is promoted by having the jacobian $det\frac{df(x)}{dx}$ easily obtained, and sampling is promoted by having f^{-1} easily available e.g. draw h from p_H, $x = f^{-1}(h)$

Auto-Regressive Flow

The main idea is to split x into (x_1, x_2), and then use

$$y_1 = x_1$$

$$y_2 = x_2 + m(x_1)$$

where m is an arbitrary function e.g. a neural network layer. Note that this transformation has unit determinant, and is easily invertible. Multiple transformations of this type can be stacked together.

These transformation expand the latent space around regions of high density e.g. data points, allowing for a more refined representation.

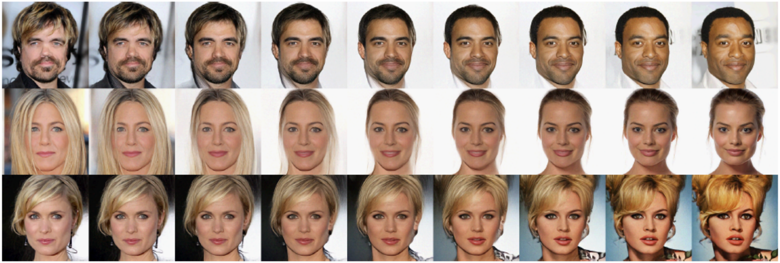
Auto-Regressive Flow

This idea is further expanded in "Improved Variational Inference with Inverse Autoregressive Flow" and "Glow: Generative Flow with Invertible 1·1 Convolutions", both by Kingma et. al.

These use the same basic idea of stacking multiple transformations, where each one is computationally simple. The most powerful, GLOW uses a set of 3 transformations: Normalization, 1x1 Convolution (Permutation), and the step fom NICE.

GLOW results

GLOW results



GLOW results

(a) Smiling

(b) Pale Skin

(c) Blond Hair

(d) Narrow Eyes

(e) Young

(f) Male

Informative Latent Variables

How do we encourage informative latent variables?

No one best solution, but a frequent approach is to encourage disentangled representations.

Two methods that utilize this idea successfully are INFO-GAN and β-VAE. I will focus on β-VAE

β-VAE

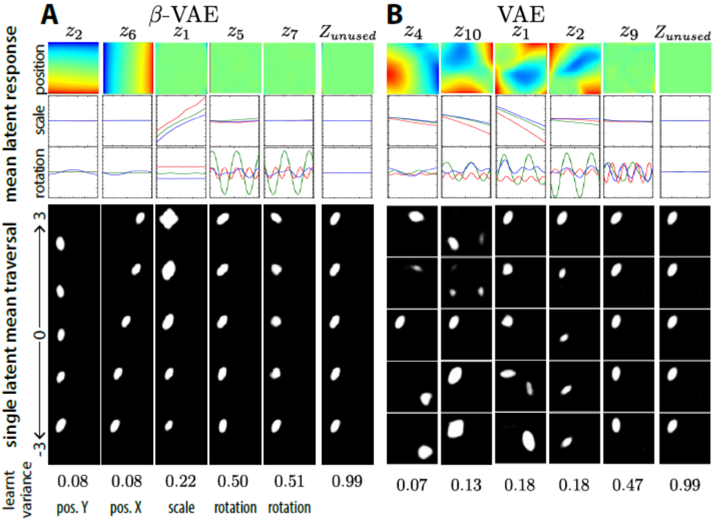
The idea behind β-VAE is simple: having a disentangled representation is equivalent to having latent variables respond to single generating factors. VAE tries to minimize the KL divergence between latent distribution q and target distribution p, where p is usually independent Gaussian.

$$E_{q_\phi(z|x)}[log p_\theta(x|z)] - D_{KL}(q_\phi(z|x)||p(z))$$

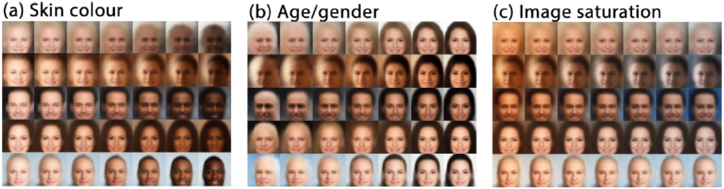
the second term is a constraint on the latent variable information, and emphasis latent independence. Having independent latents should correlate with disentanglement, so what happens if we encourage that more? β-VAE is then

$$E_{q_\phi(z|x)}[log p_\theta(x|z)] - \beta D_{KL}(q_\phi(z|x)||p(z))$$

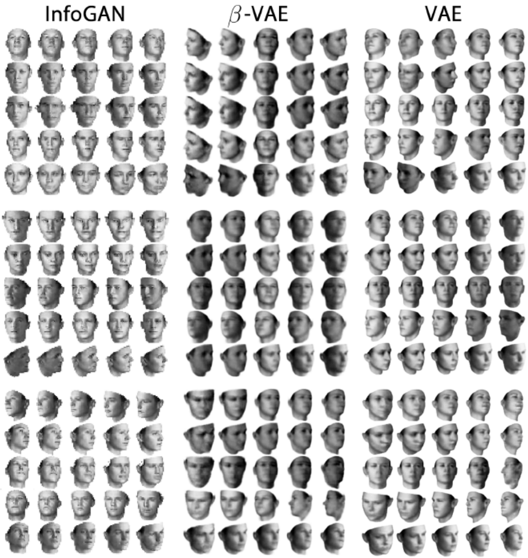
for $\beta > 0$. $\beta = 1$ corresponds to VAE, while larger values promote disentangled representations


A
β-VAE
z2 z6 z1 z5 z7 Zunused
B
VAE
z4 z10 z1 z2 z9 Zunused
mean latent response
position
scale
rotation
single latent mean traversal
3
0
-3
learnt variance
0.08 0.08 0.22 0.50 0.51 0.99
pos. Y pos. X scale rotation rotation
0.07 0.13 0.18 0.18 0.47 0.99

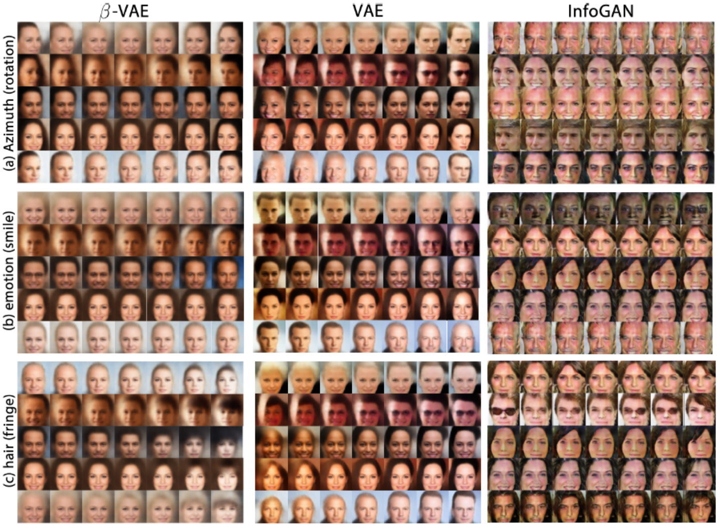
β-VAE



β-VAE



β-VAE


β-VAE
VAE
InfoGAN
(a) Azimuth (rotation)
(b) emotion (smile)
(c) hair (fringe)

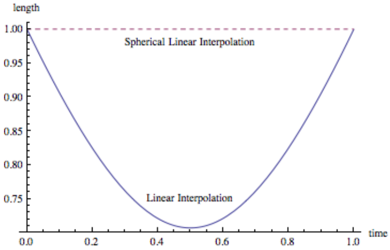
Using Latent Variables

What can we do with an informative and flexible latent space?

Generation, interpolation, and extrapolation.

High Dimensional Extrapolation

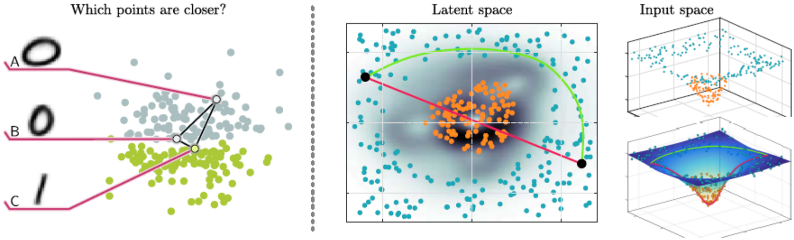
This may be more involved than just drawing a line in latent space



Also need to take into account the underlying geometry

Latent Space Geometry

"Latent Space Oddity: On the Curvature of Deep Generative Models" by Arvanitidis et. al. explores this concept.



Latent Space Geometry

Let z be a point in latent space, and δz_1 and δz_2 be infintesimals in latent space. Then

$$||f(z+\delta z_1) - f(z+\delta z_2)||^2 = (\delta z_1 - \delta z_2)^T (J_z^T J_z)(\delta z_1 - \delta z_2)$$

where J_z is the Jacobian $J_z = \frac{df}{dz}|_z$. Thus $\sqrt{det(J_z^T J_z)}$ can be seen as a measure of the local distortion or curvature in latent space, which modifies the distance of going through that point.

Similarly, we can measure distance as

$$Length[f(\gamma_t)] = \int_0^1 ||\dot{f}(\gamma_t)||dt = \int_0^1 ||J_{\gamma t}\dot{\gamma_t}||dt$$

across parameterized curve γ_t with $J_{\gamma t} = \frac{df}{dz}|_{z=\gamma t}$

Latent Space Geometry

Note that

$$||J_{\gamma t}\dot{\gamma}_t|| = \sqrt{(J_{\gamma t}\dot{\gamma}_t)^T(J_{\gamma t}\dot{\gamma}_t)} = \sqrt{\dot{\gamma}^T(J_\gamma^T J_\gamma)\dot{\gamma}} = \sqrt{\dot{\gamma}^T M_\gamma \dot{\gamma}}$$

for $M_\gamma = J_\gamma^T J_\gamma$, which is a symmetric positive definite matrix. This M_γ acts as a Riemann metric for our space at each point.

In a standard VAE, this can be simplified to

$$\bar{M}_z = E_p[M_z] = (J_z^\mu)^T(J_z^\mu) + (J_z^\sigma)^T(J_z^\sigma)$$

for J_z^μ and J_z^σ the Jacobians of the mean and variance functions. There is also a more expensive and complicated result in the paper that goes over how to solve for minimum distance (taking into account the Reimann metric) between any two points

Latent Space Geometry

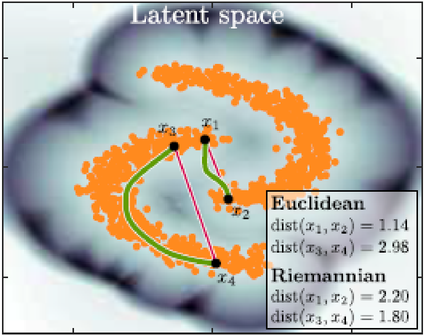


Figure 3: Example shortest paths and distances.

Latent Space Geometry

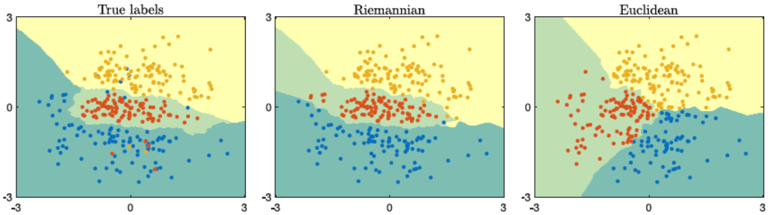


Figure 6: The result of k-means comparing the distance measures. For the decision boundaries we used 7-NN classification.

Latent Space Geometry

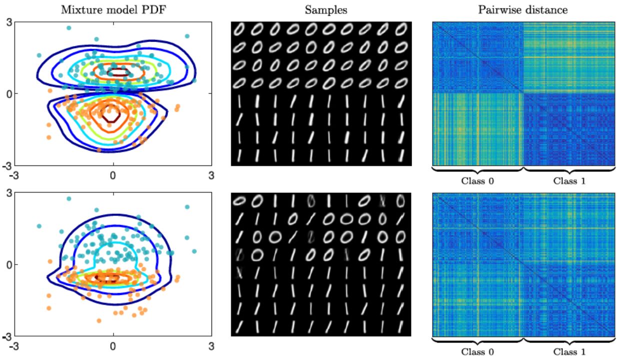


Figure 9: From *left* to *right*: the mixture models, generated samples, and pairwise distances. *Top* row corresponds to the Riemannian model and *bottom* row to the Euclidean model.

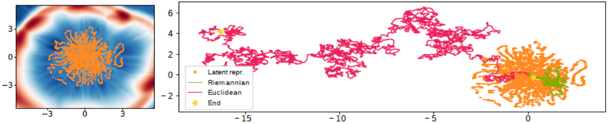
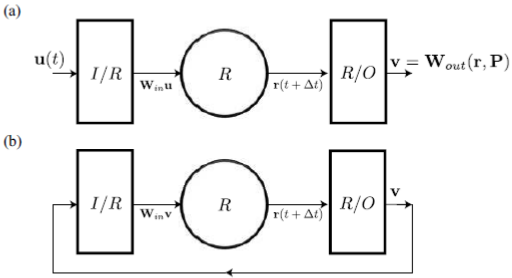
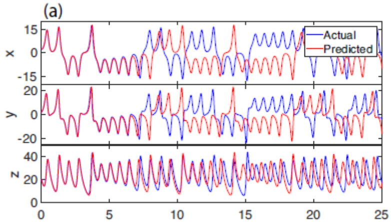


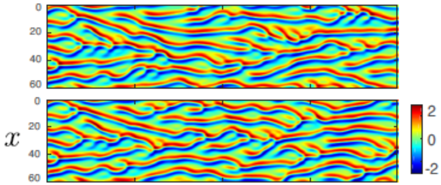
Figure 10: *Left:* the measure in the latent space. *Right:* the random walks.

RNNs are frequently used to generate or predict time series data.



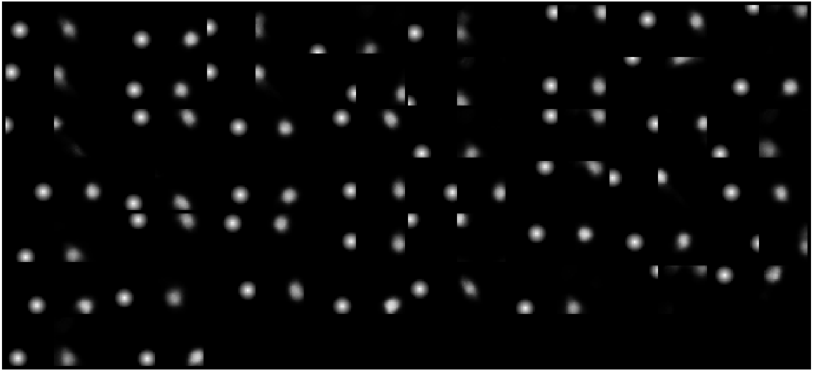
Recently, papers have confirmed that RNNs can also learn the 'climate' of a system, for example lyupanov exponents.





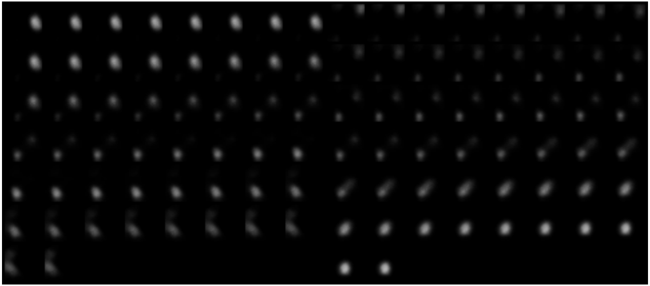
Predicting a Simple System via RNN

Sample inputs and reconstructions



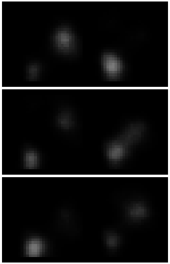
Predicting a Simple System via RNN

Varying latent dimensions



Predicting a Simple System via RNN

Predicting likely changes - varying 2nd latent least likely, 1 most



Predicting a Simple System via RNN

Predicting future movements