Mathematical Institute

Autoencoders: learning lower dimensional structure

Theories of Deep Learning: C6.5,
Lecture / Video 13

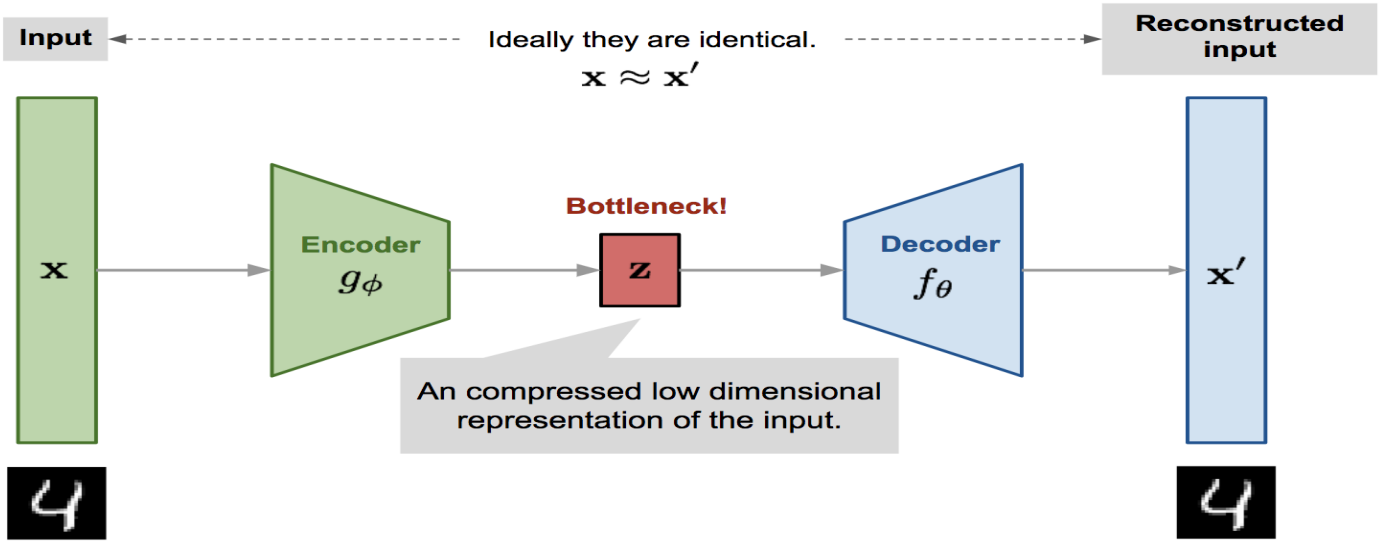
Prof. Jared Tanner
Mathematical Institute
University of Oxford

Oxford Mathematics

Autoencoder (AE) Illustration

Restricting the number of data parameters

Mathematical Institute



The parameters, (θ, ϕ), of the autoencoder are then learned:

$$\mathcal{L}(\theta, \phi) = m^{-1} \sum_{\mu=1}^{m} l(x_\mu, f_\theta(g_\phi(x_\mu)))$$

https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html

The parameters, (θ, ϕ), of the autoencoder are then learned:

$$\mathcal{L}(\theta, \phi) = n^{-1} \sum_{\mu=1}^{n} l(x_\mu, f_\theta(g_\phi(x_\mu)))$$

Consider a simple model where the encoder and decoder are linear, that is $g_\phi(x) = \Phi x$ where $\Phi \in \mathbb{R}^{r \times p}$ with $r < p$, and the linear decoder $f_\theta(z) = \Theta z$ with $\Theta \in \mathbb{R}^{p \times r}$.
Moreover, consider an entrywise ℓ_2^2 error for $l(x_\mu, f_\theta(g_\phi(x_\mu)))$, then

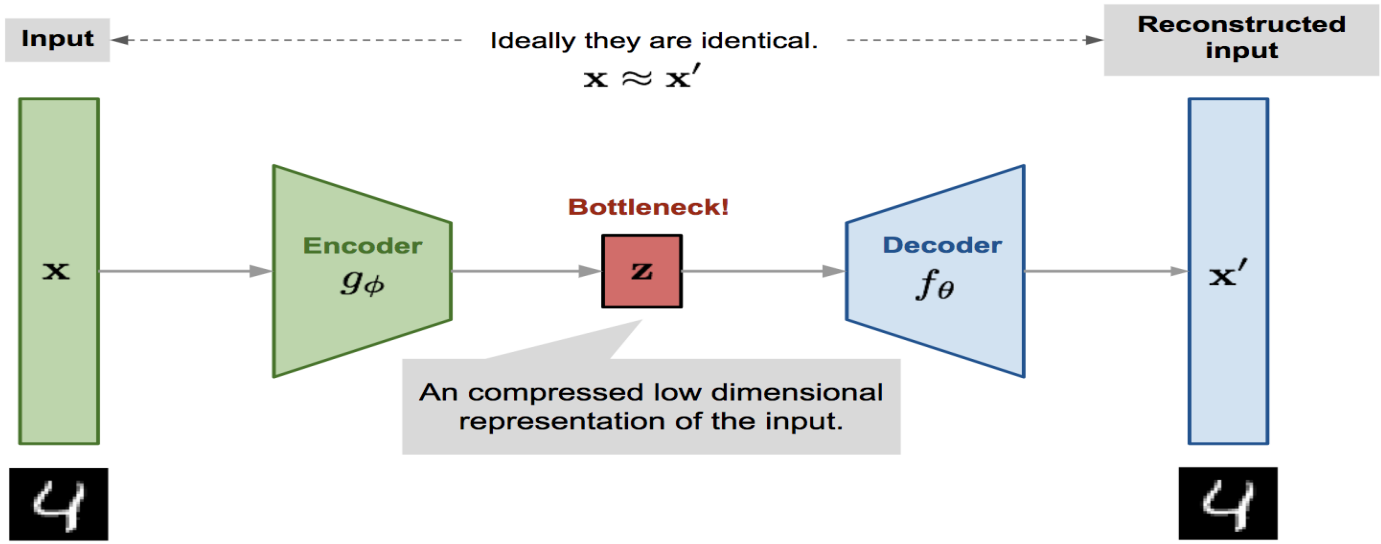
$$\mathcal{L}(\theta, \phi) = n^{-1} \|X - \Theta\Phi X\|_F^2$$

where $\Theta\Phi$ is a learned rank r matrix, whose optimal solution is the projector of X to its leading r singular space.

Autoencoder extend PCA

More complex maps to low parameter space

Mathematical Institute



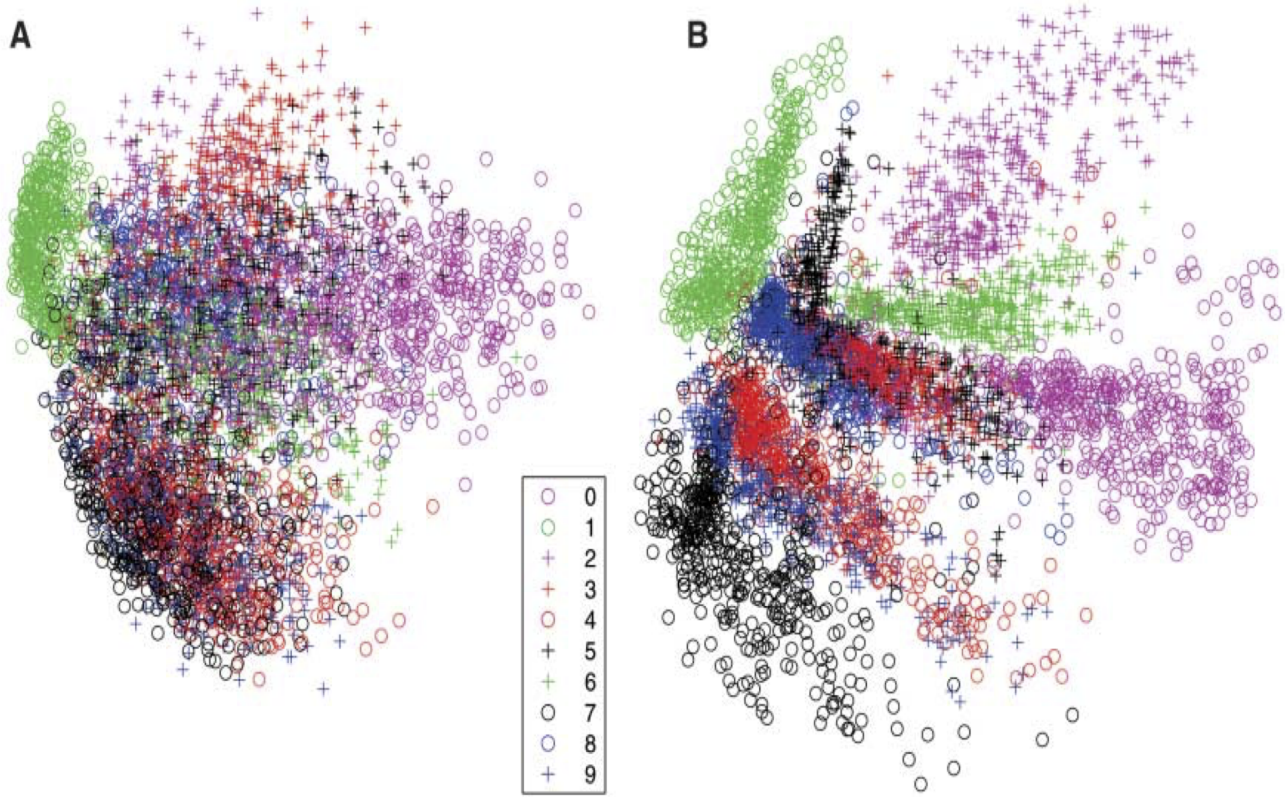
The autoencoder framework allows $g_\phi(\cdot)$ and $f_\theta(\cdot)$ to be more general than linear, and in particular to benefit from the expressively of depth and introduce variation.
`https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html`

PCA vs 3 layer Autoencoder: MNIST (Hinton et al. 06')

Improved separation of data classes

Mathematical Institute

Fig. 3. (**A**) The two-dimensional codes for 500 digits of each class produced by taking the first two principal components of all 60,000 training images. (**B**) The two-dimensional codes found by a 784-1000-500-250-2 autoencoder. For an alternative visualization, see (*8*).



http://science.sciencemag.org/content/313/5786/504

k-sparse autoencoders (Makhzani et al. 13')

Low dimensionality through sparsity

UNIVERSITY OF OXFORD
Mathematical Institute

k-Sparse Autoencoders:

Training:

1) Perform the feedforward phase and compute

$$\boldsymbol{z} = W^\top \boldsymbol{x} + \boldsymbol{b}$$

2) Find the k largest activations of $\boldsymbol{z}$ and set the rest to zero.

$$\boldsymbol{z}_{(\Gamma)^c} = 0 \quad \text{where} \quad \Gamma = \text{supp}_k(\boldsymbol{z})$$

3) Compute the output and the error using the sparsified $\boldsymbol{z}$.

$$\hat{\boldsymbol{x}} = W\boldsymbol{z} + \boldsymbol{b}'$$
$$E = \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2^2$$

3) Backpropagate the error through the k largest activations defined by Γ and iterate.

Sparse Encoding:

Compute the features $\boldsymbol{h} = W^\top \boldsymbol{x} + \boldsymbol{b}$. Find its αk largest activations and set the rest to zero.

$$\boldsymbol{h}_{(\Gamma)^c} = 0 \quad \text{where} \quad \Gamma = \text{supp}_{\alpha k}(\boldsymbol{h})$$

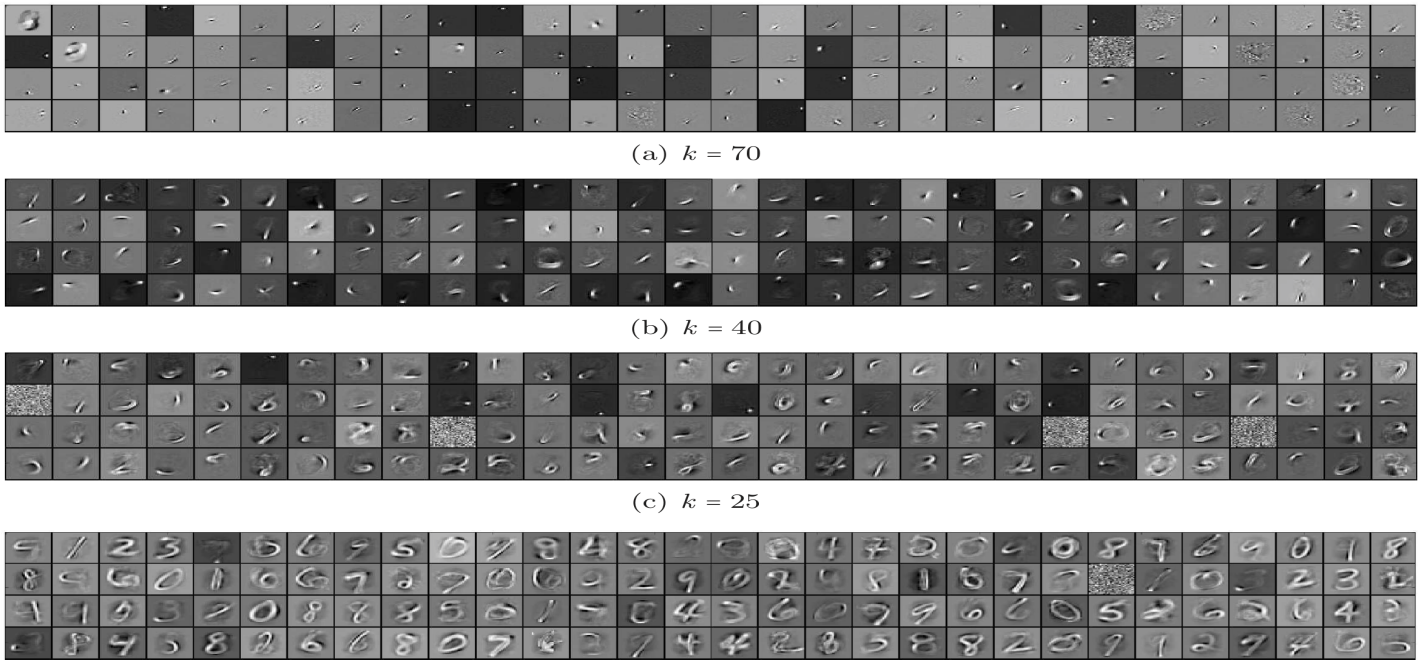
This framework includes nonlinearity and can be rigorously analysed using techniques from sparse approximation, but it lacks depth.

`https://arxiv.org/pdf/1312.5663.pdf`

k-sparse autoencoders (Makhzani et al. 13')

Learned elements: MNIST

Mathematical Institute



(a) $k = 70$

(b) $k = 40$

(c) $k = 25$

(d) $k = 10$

Figure 1. Filters of the k-sparse autoencoder for different sparsity levels k, learnt from MNIST with 1000 hidden units.

Elements learned depend on number of components, sparsity, allowed; k small are class elements, k large are basis elements.

`https://arxiv.org/pdf/1312.5663.pdf`

k-sparse autoencoders (Makhzani et al. 13')

Performance vs other autoencoders

UNIVERSITY OF OXFORD
Mathematical Institute

	Error Rate
Raw Pixels	7.20%
RBM	1.81%
Dropout Autoencoder (50% hidden)	1.80%
Denoising Autoencoder (20% input dropout)	1.95%
Dropout + Denoising Autoencoder (20% input and 50% hidden)	1.60%
k-Sparse Autoencoder, $k = 40$	1.54%
k-Sparse Autoencoder, $k = 25$	1.35%
k-Sparse Autoencoder, $k = 10$	2.10%

Table 1. Performance of unsupervised learning methods (without fine-tuning) with 1000 hidden units on **MNIST**.

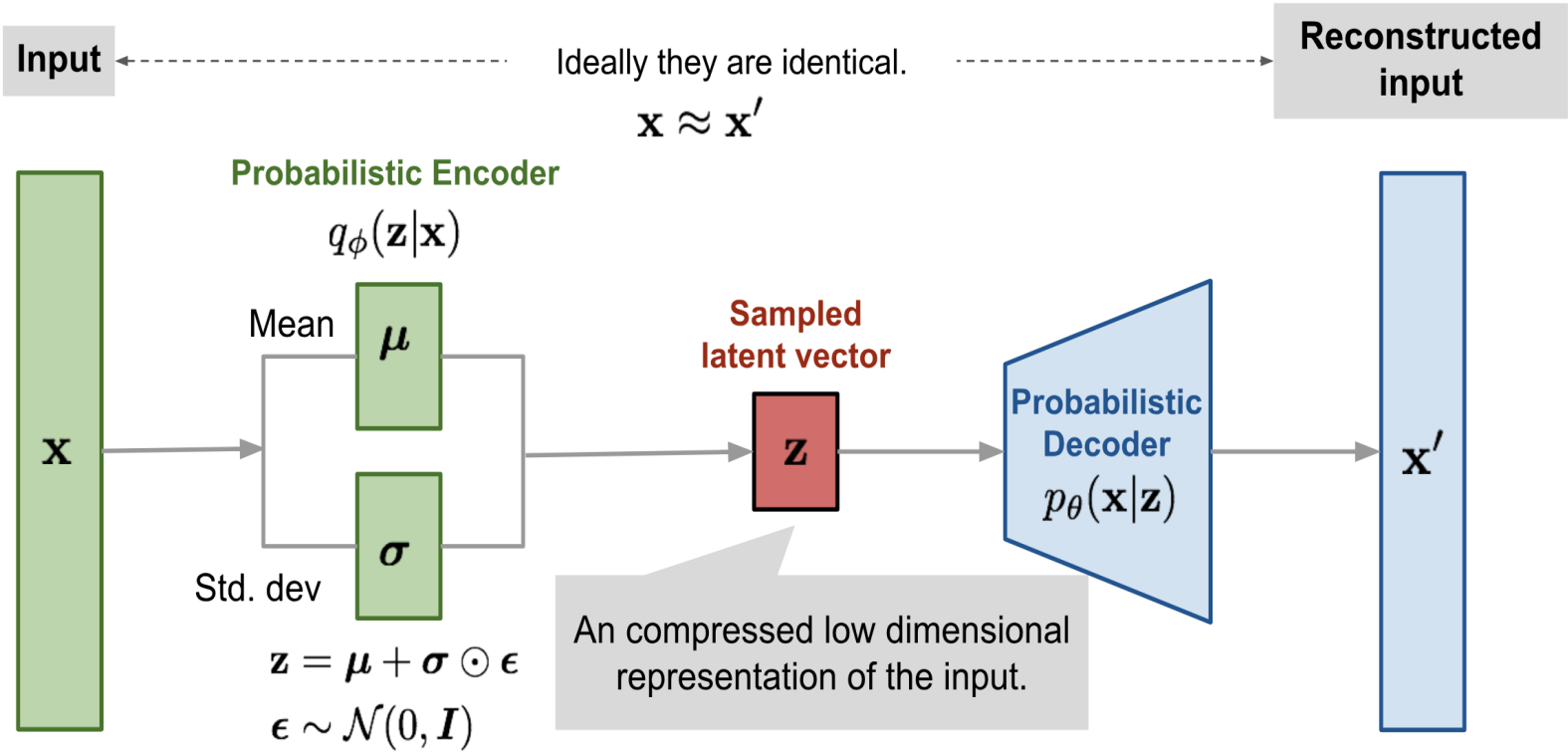
	Error
Without Pre-Training	1.60%
RBM + F.T.	1.24%
Shallow Dropout AE + F.T. (%50 hidden)	1.05%
Denoising AE + F.T. (%20 input dropout)	1.20%
Deep Dropout AE + F.T. (Layer-wise pre-training, %50 hidden)	0.85%
k-Sparse AE + F.T. (k=25)	1.08%
Deep k-Sparse AE + F.T. (Layer-wise pre-training)	0.97%

Table 3. Performance of supervised learning methods on **MNIST**. Pre-training was performed using the corresponding unsupervised learning algorithm with 1000 hidden units, and then the model was fine-tuned.

`https://arxiv.org/pdf/1312.5663.pdf`

Variational Autoencoders (VAE) (Kingma et al. 13')

Introduction of noise as a generative model

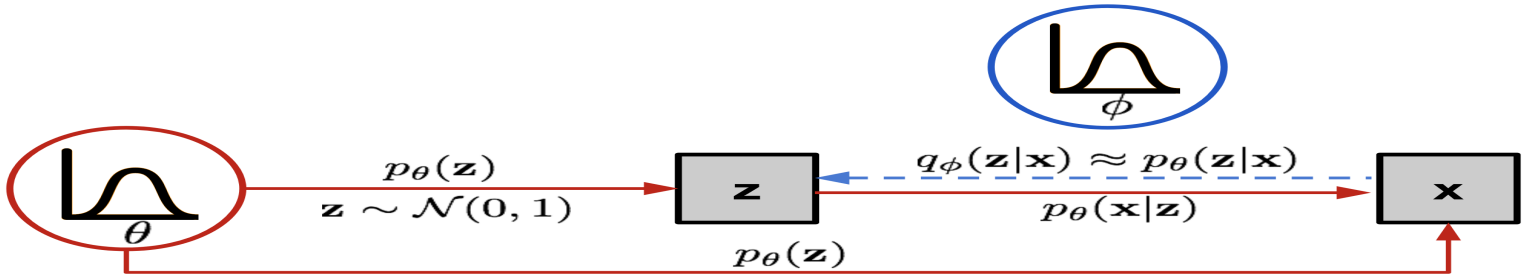


https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html

https://arxiv.org/pdf/1312.6114.pdf

Variational Autoencoders (VAE) (Kingma et al. 13')

Distribution for generative model



$p_\theta(x|z)$ acts as the generators, analogous to the decoder $f_\theta(x|z)$, and is called a probabilistic decoder

$q_\phi(z|x)$ acts as the encoder, analogous to $g_\phi(z|x)$, and is used to approximate $p_\theta(z|x)$.

The parameters ϕ, θ for a model are then learned so minimize a distance, or divergence, between $q_\phi(z|x)$ and $p_\theta(z|x)$; Kingma proposed minimising the Kullback-Leibler divergence.

`https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html`

University of Oxford
Mathematical Institute

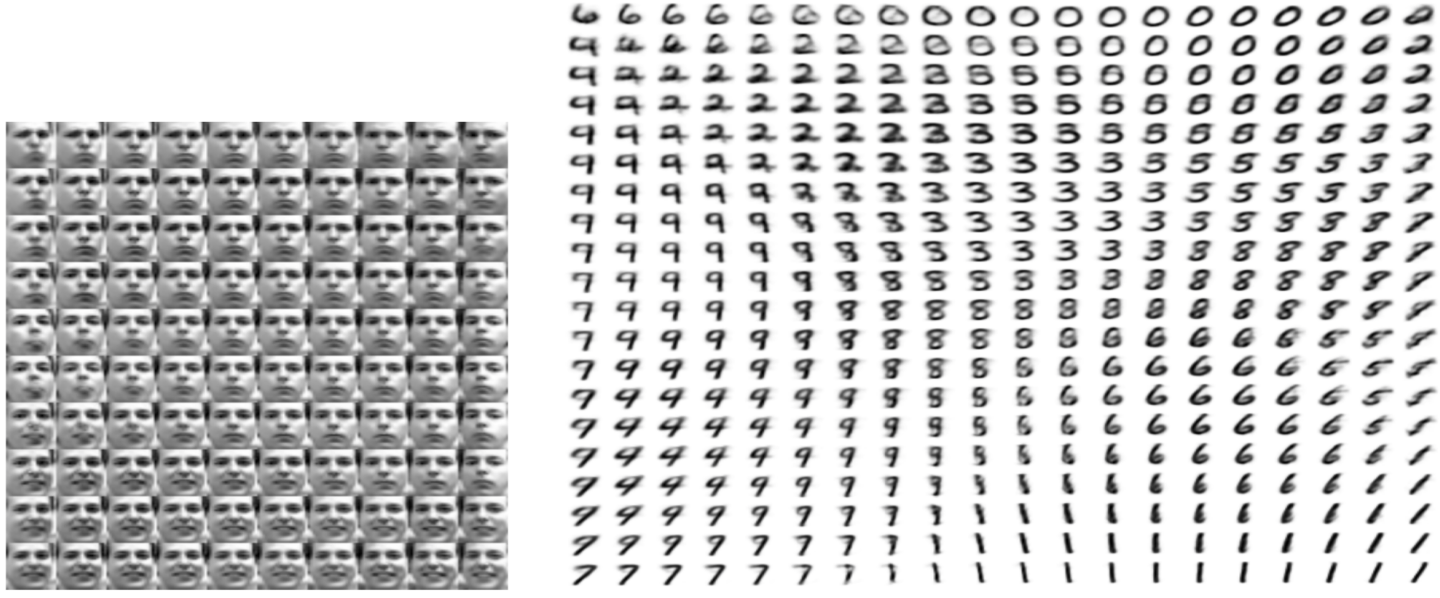
The formulae for $\mathcal{L}_{ELBO}(\phi, \theta; X)$, the evidence lower bound (ELBO), follows from minimising a lower bound of $\sum_{\mu=1}^{n} \log p_\theta(x_\mu)$:

$$\begin{aligned}
\log p_\theta(x) &= \log\left(\int p_\theta(x|z)p_\theta(z)dz\right) \\
&= \log\left(\int p_\theta(x|z)\frac{p_\theta(z)}{q_\phi(z)}q_\phi(z)dz\right) \\
&\geq \int \log\left(p_\theta(x|z)\frac{p_\theta(z)}{q_\phi(z)}\right) q_\phi(z)dz \\
&= \mathbb{E}_{q_\phi(z|x)} \log(p_\theta(x|z)) - D_{KL}(q_\phi(z|x)|p_\theta(z|x)) \\
&=: \mathcal{L}_{ELBO}(\phi, \theta; x)
\end{aligned}$$

https://arxiv.org/pdf/1312.6114.pdf

Variational Autoencoder: manifold (Kingma et al. 13')

Learned two dimensional space: faces and MNIST



(a) Learned Frey Face manifold

(b) Learned MNIST manifold

Figure 4: Visualisations of learned data manifold for generative models with two-dimensional latent space, learned with AEVB. Since the prior of the latent space is Gaussian, linearly spaced coordinates on the unit square were transformed through the inverse CDF of the Gaussian to produce values of the latent variables $\mathbf{z}$. For each of these values $\mathbf{z}$, we plotted the corresponding generative $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ with the learned parameters $\boldsymbol{\theta}$.

https://arxiv.org/pdf/1312.6114.pdf

Variational Autoencoder: manifold (Kingma et al. 13')

Dependence on manifold dimension: MNIST

(a) 2-D latent space (b) 5-D latent space (c) 10-D latent space (d) 20-D latent space

Figure 5: Random samples from learned generative models of MNIST for different dimensionalities of latent space.

`https://arxiv.org/pdf/1312.6114.pdf`

Inference Variational Autoencoders (Zhao et al. 17')

$p_\theta(x|z)$ acts as the generators, analogous to the decoder $f_\theta(x|z)$, and is called a probabilistic decoder; $q_\phi(z|x)$ acts as the encoder, analogous to $g_\phi(z|x)$, and is used to approximate $p_\theta(z|x)$.
The parameters ϕ, θ for a model are then learned to minimize a distance, or divergence, between $q_\phi(z|x)$ and $p_\theta(z|x)$; Kingma proposed minimising the Kullback-Leibler divergence, giving the evidence lower bound (ELBO)

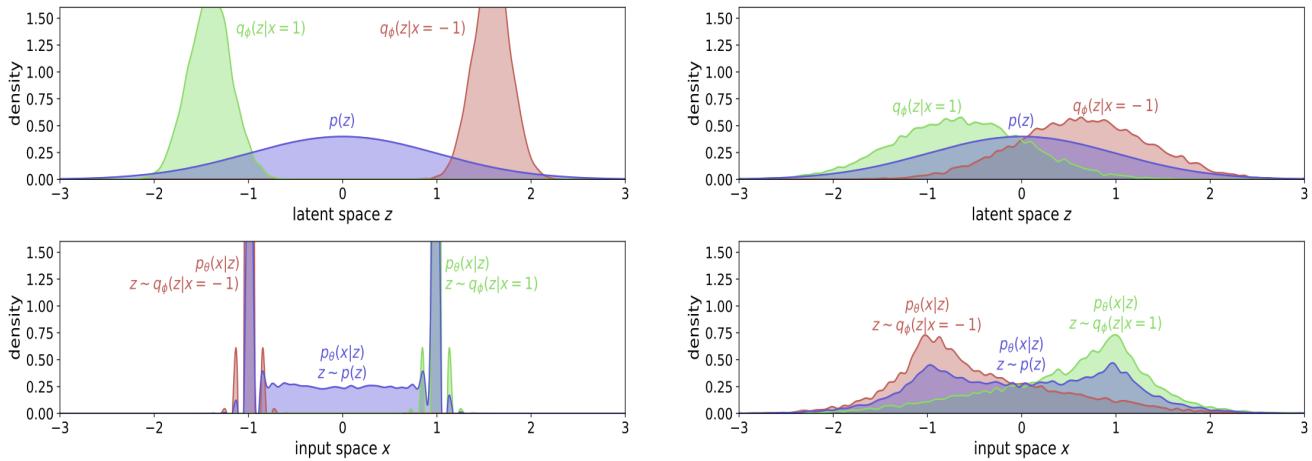
$$\mathcal{L}_{ELBO} := \mathbb{E}_{q_{\phi(z|x)}} \log(p_\theta(x|z)) - \beta D_{KL}(q_\phi(x,z)||p_\theta(x,z))$$

VAEs originally use $\beta = 1$, with larger $\beta > 1$ called β-VAEs.
Zhao et al. propose including a mutual information term to avoid mode separation and collapse.
`https://arxiv.org/pdf/1706.02262.pdf`

Inference Variational Autoencoders (Zhao et al. 17')

Impact of VAE objective

University of Oxford
Mathematical Institute



ELBO InfoVAE ($\lambda = 500$)

Figure 1: Verification of Proposition 1 where the dataset only contains two examples $\{-1, 1\}$. **Top:** density of the distributions $q_\phi(z|x)$ when $x = 1$ (red) and $x = -1$ (green) compared with the true prior $p(z)$ (purple). **Bottom:** The "reconstruction" $p_\theta(x|z)$ when z is sampled from $q_\phi(z|x = 1)$ (green) and $q_\phi(z|x = -1)$ (red). Also plotted is $p_\theta(x|z)$ when z is sampled from the true prior $p(z)$ (purple). When the dataset consists of only two data points, ELBO (**left**) will push the density in latent space Z away from 0, while InfoVAE (**right**) does not suffer from this problem.

`https://arxiv.org/pdf/1706.02262.pdf`

Inference Variational Autoencoders (Zhao et al. 17')

Impact of VAE objective

UNIVERSITY OF OXFORD
Mathematical Institute

reconstruction generation

Figure 5: Samples generated by ELBO vs. MMD InfoVAE ($\lambda = 1000$) after training on 500 samples (plotting mean of $p_\theta(x|z)$). **Top:** Samples generated by ELBO. Even though ELBO generates very sharp reconstruction for samples on the training set, model samples $p(z)p_\theta(x|z)$ is very poor, and differ significantly from the reconstruction samples, indicating over-fitting, and mismatch between $q_\phi(z)$ and $p(z)$. **Bottom:** Samples generated by InfoVAE. The reconstructed samples and model samples look similar in quality and appearance, suggesting better generalization in the latent space.

https://arxiv.org/pdf/1706.02262.pdf

β-VAEs disentangling features pt. 1 (Higgins et al. 17')

Explainable latent space

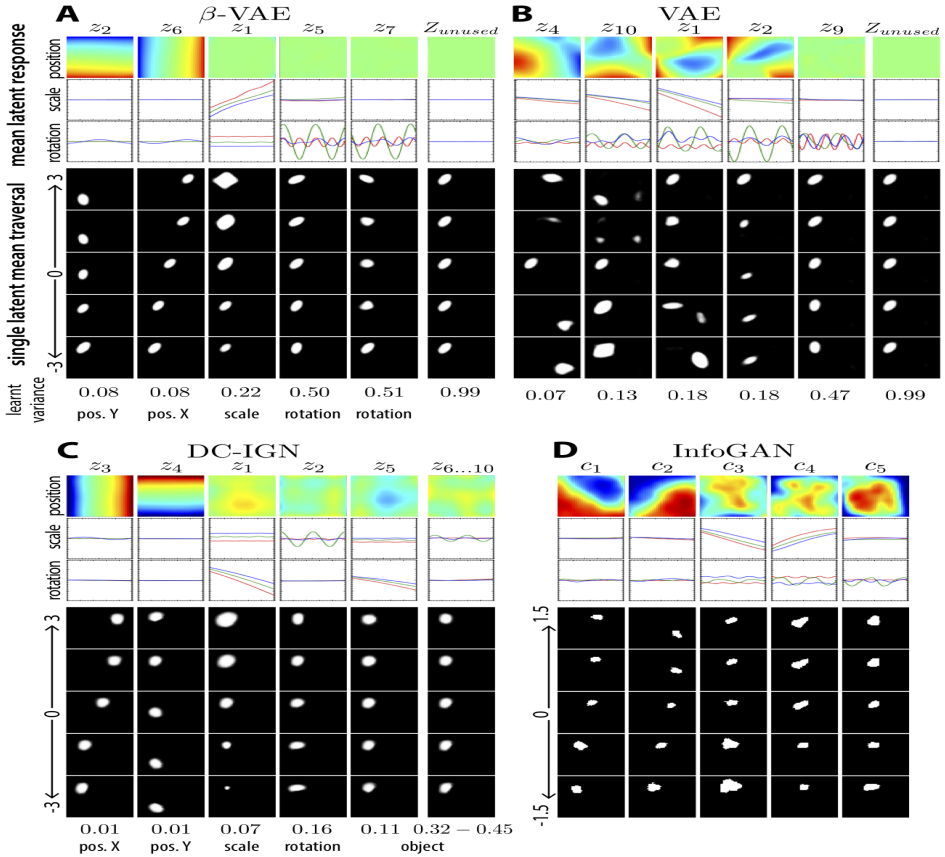


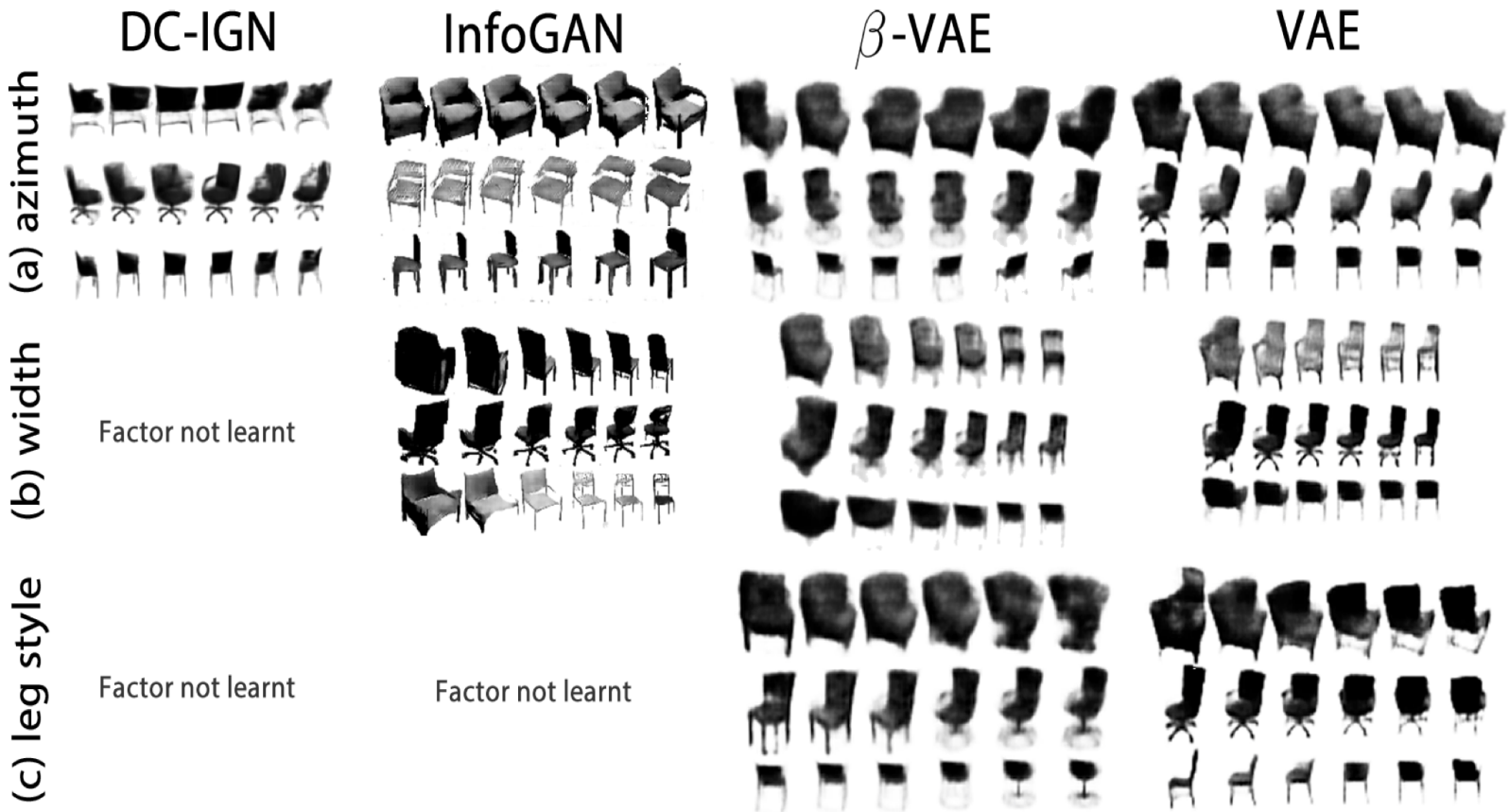
Figure 7: **A:** Representations learnt by a β-VAE ($\beta = 4$). Each column represents a latent z_i, ordered according to the learnt Gaussian variance (last row). Row 1 (position) shows the mean activation (red represents high values) of each latent z_i as a function of all 32x32 locations averaged across objects, rotations and scales. Row 2 and 3 show the mean activation of each unit z_i as a function of scale (respectively rotation), averaged across rotations and positions (respectively scales and positions). *Square* is red, *oval* is green and *heart* is blue. Rows 4-8 (second group) show reconstructions resulting from the traversal of each latent z_i over three standard deviations around the unit Gaussian prior mean while keeping the remaining 9/10 latent units fixed to the values obtained by running inference on an image from the dataset. **B:** Similar analysis for VAE ($\beta = 1$). **C:** Similar analysis for DC-IGN, clamping a single latent each for scale, positions, orientation and 5 for shape. **D:** Similar analysis for InfoGAN, using 5 continuous latents regularized using the mutual information cost, and 5 additional unconstrained noise latents (not shown).

https://arxiv.org/pdf/1706.02262.pdf

β-VAEs disentangling features pt. 2 (Higgins et al. 17')

Explainable latent space: chairs

University of Oxford
Mathematical Institute



https://arxiv.org/pdf/1706.02262.pdf

β-VAEs disentangling features pt. 3 (Higgins et al. 17')

Architectures for encoder-decoders

Dataset	Optimiser		Architecture
2D shapes (VAE)	Adagrad 1e-2	Input	4096 (flattened 64x64x1).
		Encoder	FC 1200, 1200. ReLU activation.
		Latents	10
		Decoder	FC 1200, 1200, 1200, 4096. Tanh activation. Bernoulli.
2D shapes (DC-IGN)	rmsprop (as in Kulkarni et al., 2015)	Input	64x64x1.
		Encoder	Conv 96x3x3, 48x3x3, 48x3x3 (padding 1). ReLU activation and Max pooling 2x2.
		Latents	10
		Decoder	Unpooling, Conv 48x3x3, 96x3x3, 1x3x3. ReLU activation, Sigmoid.
2D shapes (InfoGAN)	Adam 1e-3 (gen) 2e-4 (dis)	Generator	FC 256, 256, Deconv 128x4x4, 64x4x4 (stride 2). Tanh.
		Discriminator	Conv and FC reverse of generator. Leaky ReLU activation. FC 1. Sigmoid activation.
		Recognition	Conv and FC shared with discriminator. FC 128, 5. Gaussian
		Latents	10: $z_{1\ldots5} \sim Unif(-1,1)$, $c_{1\ldots5} \sim Unif(-1,1)$
Chairs (VAE)	Adam 1e-4	Input	64x64x1.
		Encoder	Conv 32x4x4 (stride 2), 32x4x4 (stride 2), 64x4x4 (stride 2), 64x4x4 (stride 2), FC 256. ReLU activation.
		Latents	32
		Decoder	Deconv reverse of encoder. ReLU activation. Bernoulli.

https://arxiv.org/pdf/1706.02262.pdf

Autoencoders summary

NNs used as nonlinear maps to lower dimensional latent spaces

Mathematical Institute

- Principal component analysis (PCA) reveals the low dimensional latent space within a data matrix by projecting to the space of low-rank matrices.
- Autoencoders (AE) extend this notion allowing more general maps to and from a low dimensional parameter space.
- Variational AEs (VAEs) give a probabilistic notion that gives a natural generative model.
- Inference VAEs and $\beta-$VAEs are further extensions to improve VAEs and for interpretability respectively.
- (V)AEs are a flexible structure allowing general maps; an area open for great further analysis.