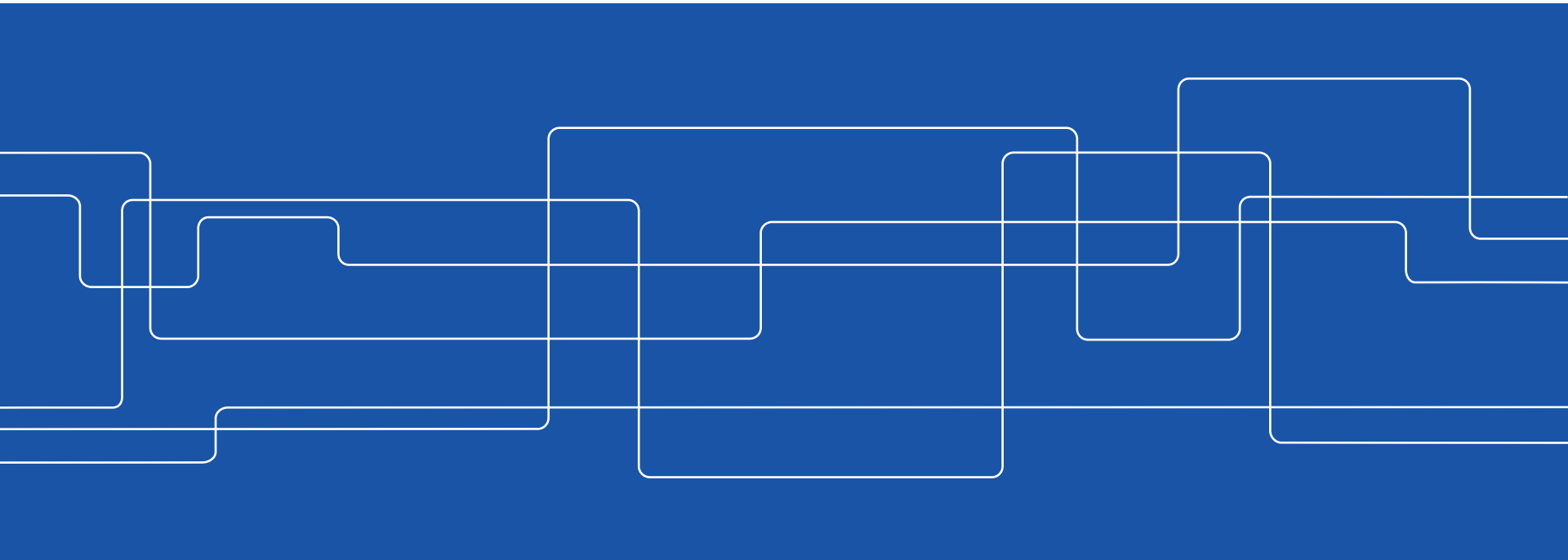
KTH ROYAL INSTITUTE
OF TECHNOLOGY

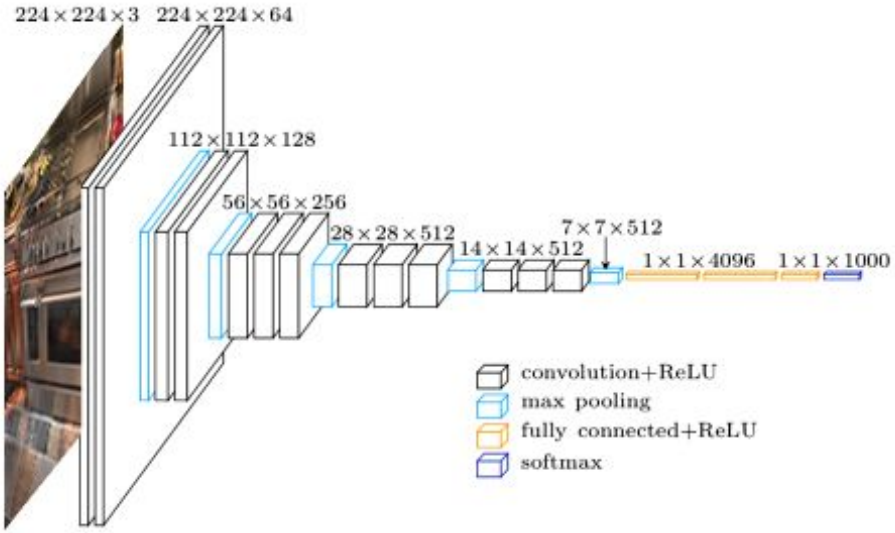
From Autoencoders to β-VAE

Emir Konuk

2018.10.09



Starting point

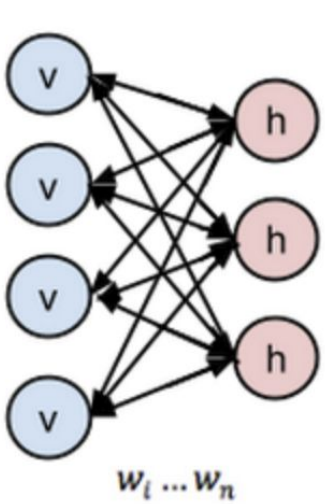


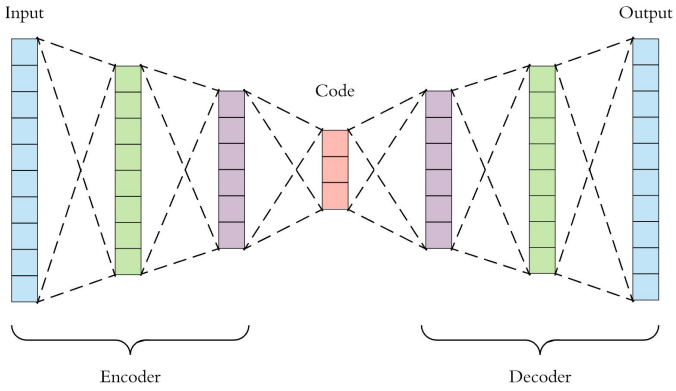
An image classifier

* http://agnesmustar.com/2017/04/19/build-vgg16-scratch-keras-part/

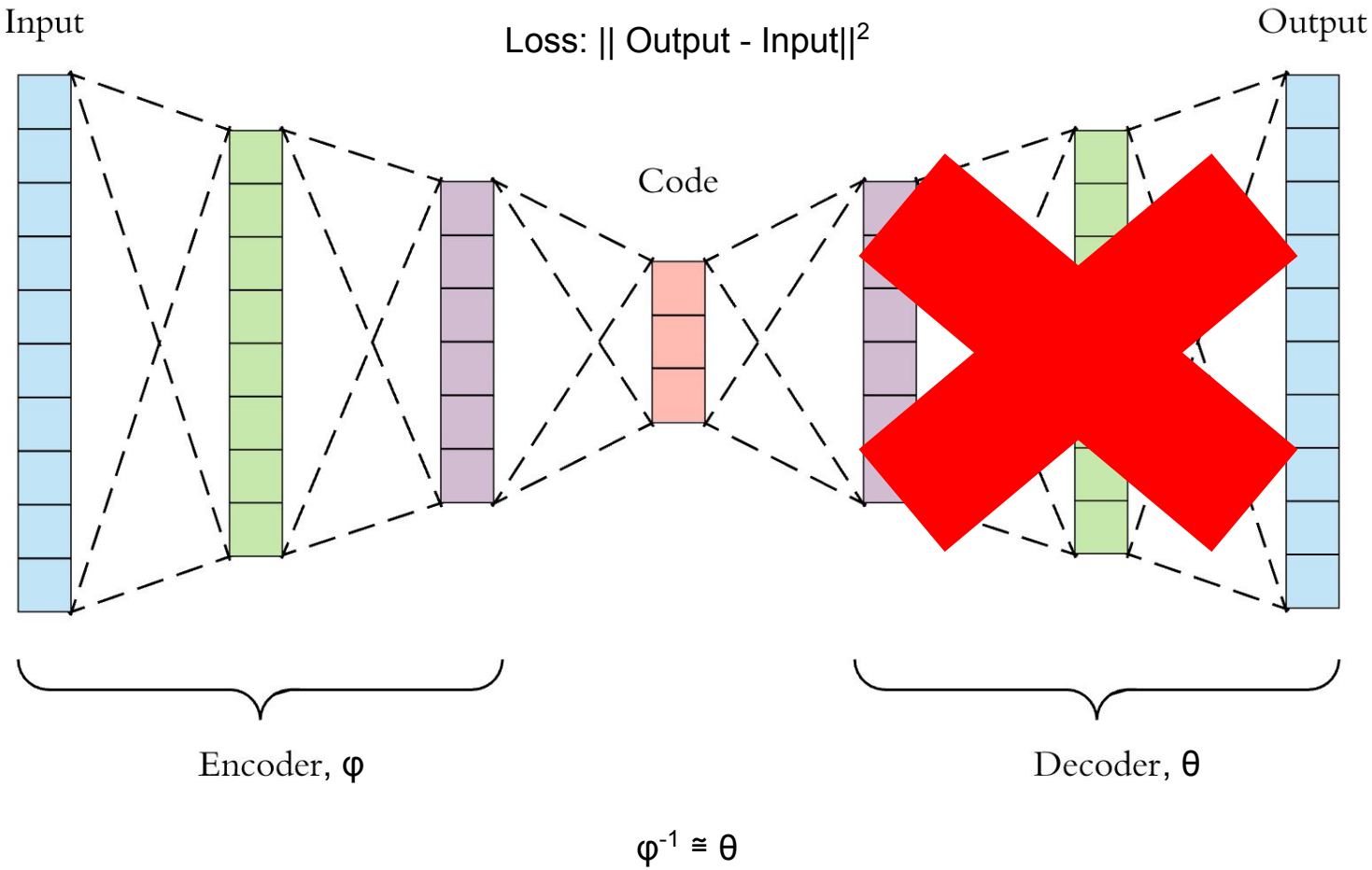
Training woes

- Data hungry models
- Cheap unlabeled data
- Pretraining on unlabeled data
- Restricted Boltzmann Machines and Autoencoders





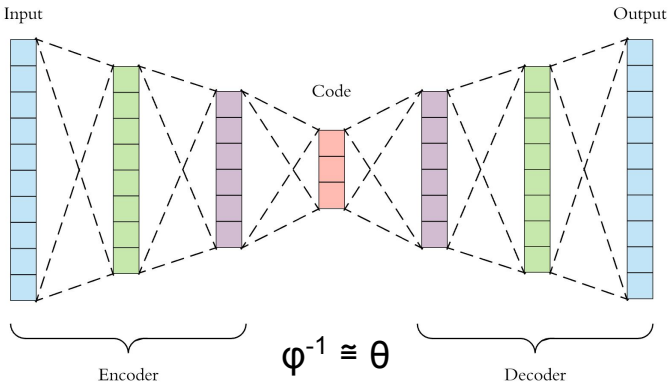
What is an autoencoder?



$\varphi^{-1} \cong \theta$

* https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798

Aside: Decoder == Least Squares ?



- Why not find the inverse directly?
  - elu, Leaky Relu, sigmoid, etc. all invertible
  - What we have is consecutive matrix multiplications => consecutive (pseudo) inverses
    - At least one underdetermined system => *minimum norm* least squares solution

$$||Ax - b||_2^2 + ||x||_2^2$$

Aside: Minimum Norm Least Squares

- Is this point/solution (i.e., image) the one our “trained” decoder creates?
- Should it be?

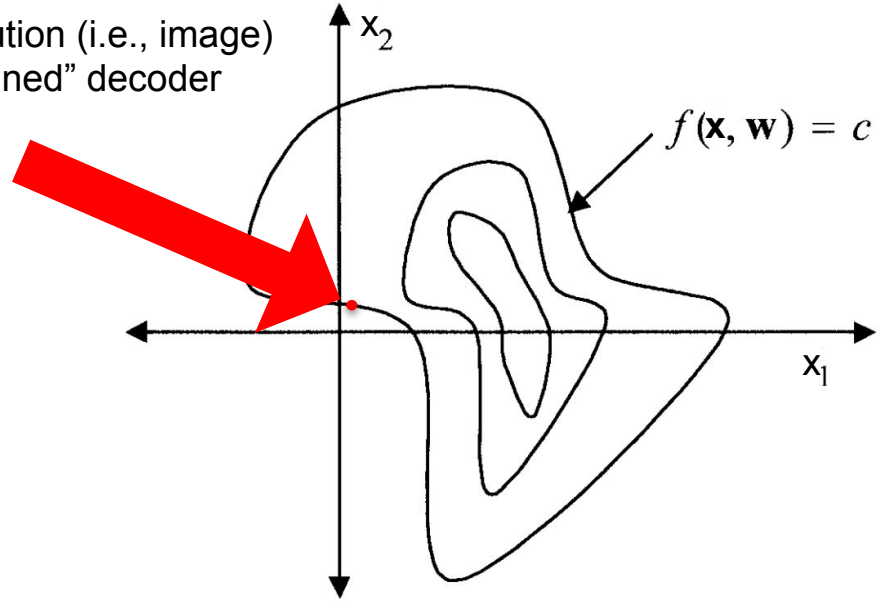


Fig. 1. The inversion of a neural network typically has numerous solutions. Each of the input ordered pairs lying on a given contour generate the same neural network output. The contours can be disjoint.

* Jensen, Craig A., et al. "Inversion of feedforward neural networks: Algorithms and applications." Proceedings of the IEEE 87.9 (1999): 1536-1549.

The real question:
Can you generate an image of a doggy?

Aim: Generate a sample as good as $x \sim p_r$

How to use an AE for this purpose?

- Sample $x^* \sim \theta(p_{latent})$
- $\theta(.)$ is matmul, but what is $p_{latent} \triangleq \varphi(p_r)$?
- p_{latent} is,
  - Weird
  - Sparse *but **everything** is sparse

* Kaggle Cats&Dogs

What about a hendog?

"A disentangled representation can be defined as one where single latent units are sensitive to changes in single generative factors, while being relatively invariant to changes in other factors." [1]

Claim: If $z \sim p_{latent} \in \mathcal{R}^N$ has N semantically meaningful features, some **weird** stuff would be possible.

A practical observation: Steve Carell becomes Maggie Gyllenhaal on an "entangled" direction

"I care more about clustering (and convexity) in the latent space." [me]

* Large Scale GAN Training for High Fidelity Natural Image Synthesis
* Sainburg, Tim, et al. "Generative adversarial interpolative autoencoding: adversarial training on latent space interpolations encourage convex latent distributions." arXiv preprint arXiv:1807.06650 (2018)

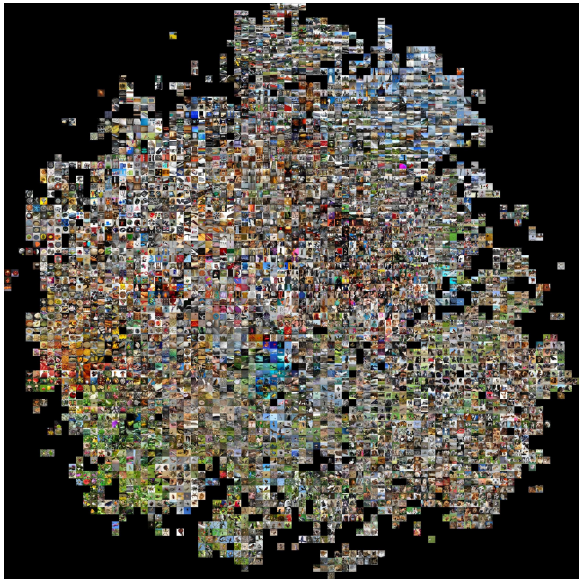
Again with p_{latent}

1) Somehow be able to generate samples from the weird p_{latent}

* Not necessarily the exact p_{latent}

2) Force the φ to make a “nice” p_{latent}, maybe $N(z \mid \mu = 0, \sigma = I)$

* Not necessarily an exact Normal distribution



* https://lvdmaaten.github.io/tsne/

Aside: Sample from the non-regularized p_{latent}

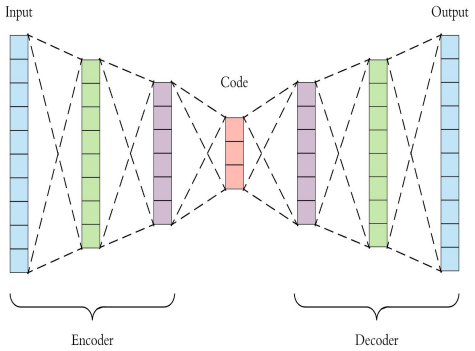
This route takes us to adversarial training/GAN. How?

- Start with, $N(z \mid 0, I)$
- Apply some bijections[*actually not], g_β, to $z \sim N(z \mid 0, I)$
- Learn the β for which the $g_\beta(z) \cong p_{latent}$
- Arbitrarily denote the distribution "difference" as $W(p_G, p_{latent})$
- Remember that $p_{latent} = \varphi(p_r)$
  - We could as well minimize $W(p_G, p_r)$
  - That is, find the θ where $g_\theta(z) \cong p_r$
  - Discarding the φ would always be good, right?
- Minimizing the (earthmovers) distance between the distributions corresponds to minimizing this loss (WGAN) w.r.t. the generator:

$$\min_G \max_{D \in \mathcal{D}} \mathbb{E}_{\boldsymbol{x} \sim \mathbb{P}_r} \left[D(\boldsymbol{x}) \right] - \mathbb{E}_{\tilde{\boldsymbol{x}} \sim \mathbb{P}_g} \left[D(\tilde{\boldsymbol{x}}) \right]$$

What is an autoencoder, again?

Small Print Disclaimer: I'm playing loose with the notation to tell a story



Loss: || Output - Input||^2

Q: What do we want the decoder, θ, to do?

A: Make the output *x* as close as possible to the input *x*

- Decoder has to act on the latent code *z*
- Loosely, find the θ which maximizes the likelihood

$$\mathcal{L}(\theta) = \log(p_\theta(x \mid z))$$

- In words: Given a latent code, the decoder should output the input that generated that code.

Where is the encoder, φ?

$$\mathcal{L}(\theta, \varphi) = \log(p_\theta(x \mid \varphi(x)))$$

Gaussian => $L(\theta, \varphi) = || x - \theta(\varphi(x))||^2$

What is an autoencoder, again?

$$\mathcal{L}(\theta,\varphi) = \mathbb{E}_{q(z \mid x;\, \varphi)} \log(\, p_\theta(x \mid z))$$

What happens when this is a Dirac Delta, i.e., standard AE?

$$\mathcal{L}(\theta,\varphi) = \mathbb{E}_{\delta(z - \varphi(x))} \log(\, p_\theta(x \mid z))$$

$$\mathcal{L}(\theta, \varphi) = \log(\, p_\theta(x \mid \varphi(x)))$$

…and we end up with the standard AE loss

What is a variational autoencoder?

$$\mathcal{L}(\theta,\varphi) = \mathbb{E}_{q(z \mid x;\varphi)} \log(p_\theta(x \mid z))$$

When this is not a
Delta but a Gaussian

Q: What does this mean?

A: Not much, except now we need the “reparameterization trick”

Also, the latent vectors are still “weird”

Latent, be nice!

$$\mathcal{L}(\theta, \phi; \mathbf{x}, \mathbf{z}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{KL}\left(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z})\right)$$

- Force the distribution from which we sample the latent vectors to be "nice"
- "Nice" is an acronym for "closed form equations are easy with Gaussians"
- Also, we can sample from a Gaussian, feed it to the decoder and generate reasonable images
- Note that we minimize the KL divergence for **each** sample. Not cool - in fact, KL is not cool at all.
  - This line of thought ends up in adversarial autoencoders. I would think the first thing to try would be to approximate Wasserstein distance between $q_\varphi(z|x)$ and $p(z)$ using another network. ~~I haven't seen this yet, might not be possible.~~ ICLR 2018

What is β-VAE?

~~A disappointment.~~ Adding a scale factor in front of the regularization term:

$$\mathcal{L}(\theta, \phi; \mathbf{x}, \mathbf{z}, \beta) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \beta\, D_{KL}\big(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z})\big)$$

- It is enforcing a stronger force on the latent distributions to be smoother
- The real paper should have been “Understanding disentangling in β-VAE.” (same authors) [1]
- So, why is this better?

Benefits(?) of the β

1. Smoothness. Why? Less space, more smooth* but everything is still **really** sparse so...

2. Disentanglement. Why?

"Our key hypothesis is that β-VAE finds latent components which make different contributions to the log-likelihood term of the cost function. These latent components tend to correspond to features in the data that are intuitively qualitatively different, and therefore may align with the generative factors in the data." [1]

I don't get it.

Why so disentangled?

Another paper, “Isolating Sources of Disentanglement in Variational Autoencoders” [2], has a better explanation:

$$\mathbb{E}_{p(n)}\Big[\mathrm{D_{KL}}\big(q(z|n)||p(z)\big)\Big] = \mathbb{E}_{q(z,n)}\Big[\log q(z|n) + \log p(z) + \log q(z) - \log q(z) + \log \prod_j q(z_j) - \log \prod_j q(z_j)\Big]$$

$$= \underbrace{\mathrm{D_{KL}}\left(q(z,n)||q(z)p(n)\right)}_{\text{(i) Index-Code MI}} + \underbrace{\mathrm{D_{KL}}\big(q(z)||\prod_j q(z_j)\big)}_{\text{(ii) Total Correlation}} + \underbrace{\sum_j \mathrm{D_{KL}}\left(q(z_j)||p(z_j)\right)}_{\text{(iii) Dimension-wise KL}}$$

Thus, a higher β penalizes everything, including (*ii*), encouraging independence of features (somewhat related to disentanglement).

Penalizing mutual information (*i*), however, is **NOT** a good thing.

Metric and Results

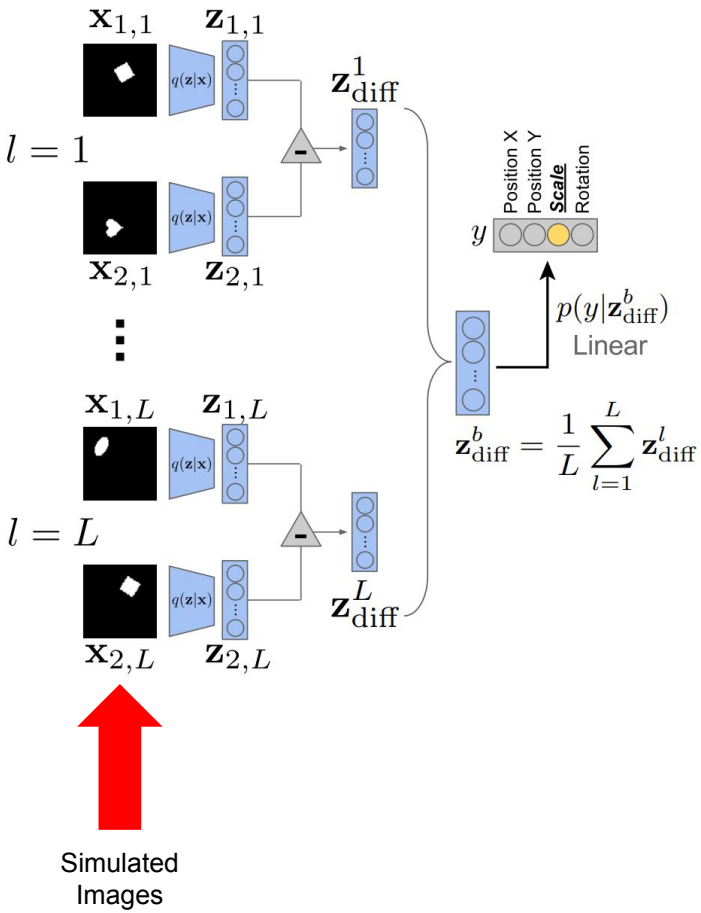


Figure : Schematic of the proposed disentanglement metric: over a batch of L samples, each pair of images has a fixed value for one target generative factor y (here $y = scale$) and differs on all others. A linear classifier is then trained to identify the target factor using the average pairwise difference $\mathbf{z}^b_{\text{diff}}$ in the latent space over L samples.

Results are kind of misleading (as if we only aim for disentanglement)

Model	Disentanglement metric score
Ground truth	*100%*
Raw pixels	$45.75 \pm 0.8\%$
PCA	$84.9 \pm 0.4\%$
ICA	$42.03 \pm 10.6\%$
DC-IGN	$\mathbf{99.3 \pm 0.1\%}$
InfoGAN	$73.5 \pm 0.9\%$
VAE untrained	$44.14 \pm 2.5\%$
VAE	$61.58 \pm 0.5\%$
β-VAE	$\mathbf{99.23 \pm 0.1\%}$

Aside: Why so Gaussian?

- Yay: Closed form solutions!
- Nay: Isotropic Gaussians for images?

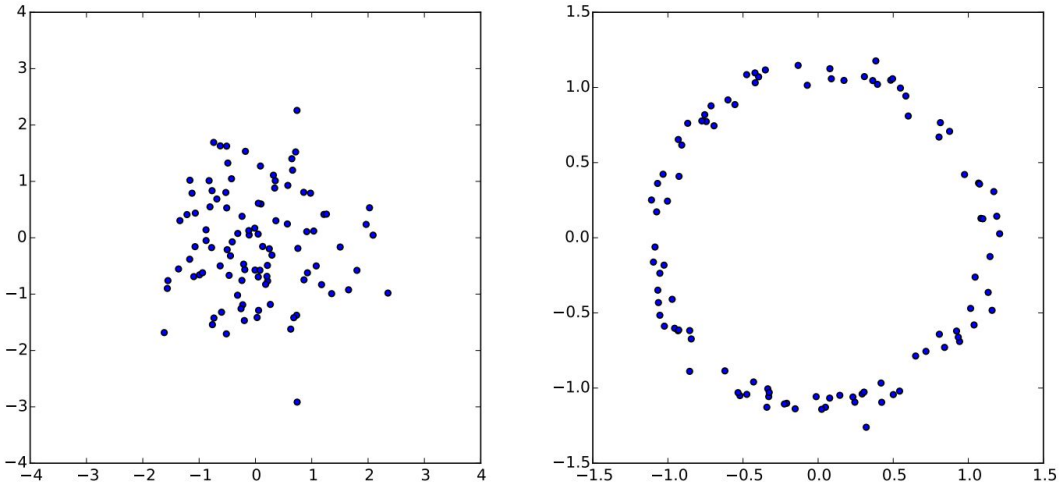


Figure : Given a random variable z with one distribution, we can create another random variable $X = g(z)$ with a completely different distribution. Left: samples from a gaussian distribution. Right: those same samples mapped through the function $g(z) = z/10 + z/||z||$ to form a ring. This is the strategy that VAEs use to create arbitrary distributions: the deterministic function g is learned from data.

- We hope so
- Explicitly encourage this?

* Doersch, Carl. "Tutorial on variational autoencoders." arXiv preprint arXiv:1606.05908 (2016).

Good to check

[1] Burgess, Christopher P., et al. "Understanding disentangling in β-VAE." arXiv preprint arXiv:1804.03599 (2018).

[2] Chen, Tian Qi, et al. "Isolating Sources of Disentanglement in Variational Autoencoders." arXiv preprint arXiv:1802.04942 (2018).

[3] Zhao, Shengjia, Jiaming Song, and Stefano Ermon. "Infovae: Information maximizing variational autoencoders." arXiv preprint arXiv:1706.02262 (2017).

[4] Zhao, Shengjia, Jiaming Song, and Stefano Ermon. "Towards deeper understanding of variational autoencoding models." arXiv preprint arXiv:1702.08658 (2017).

[5] Doersch, Carl. "Tutorial on variational autoencoders." arXiv preprint arXiv:1606.05908 (2016).

[6] Sønderby, Casper Kaae, et al. "Ladder variational autoencoders." Advances in neural information processing systems. 2016.

[7] Bengio, Y., A. Courville, and P. Vincent. "Representation learning: a review and new perspectives. arXiv. org." (2012).

[8] Tolstikhin, Ilya, et al. "Wasserstein auto-encoders." arXiv preprint arXiv:1711.01558 (2017).