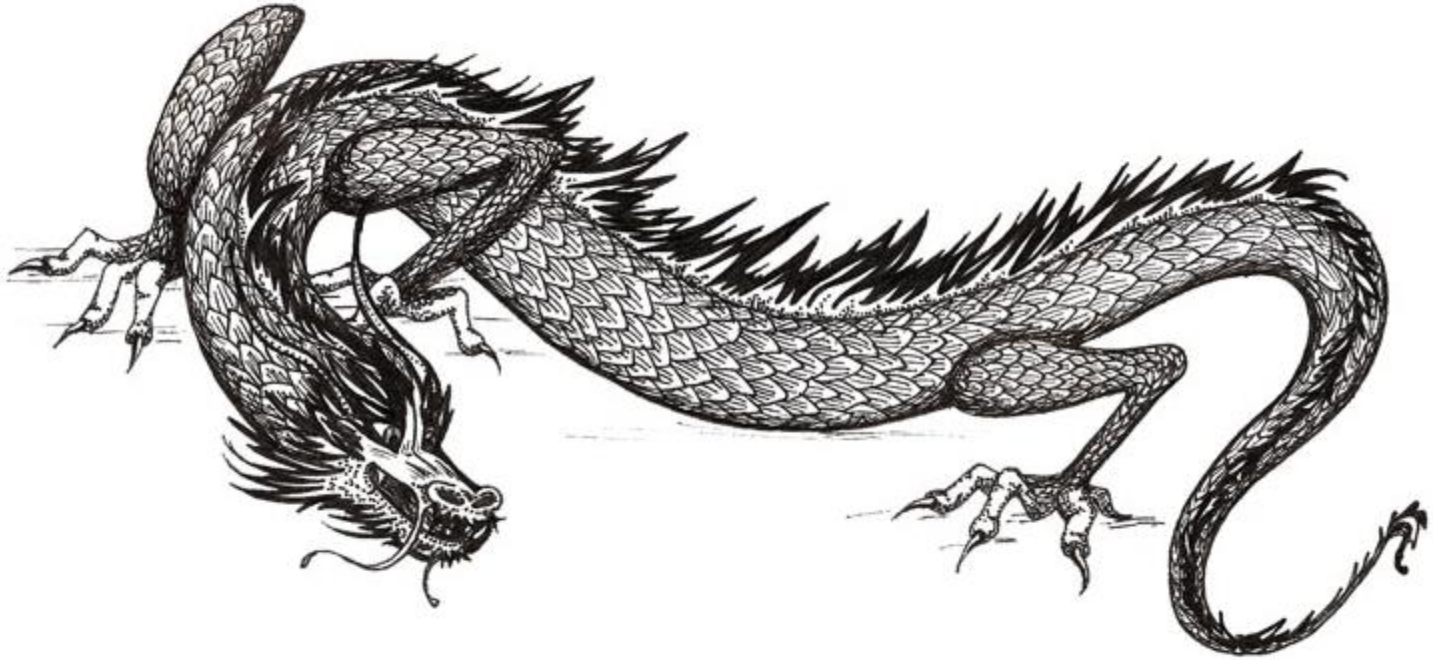
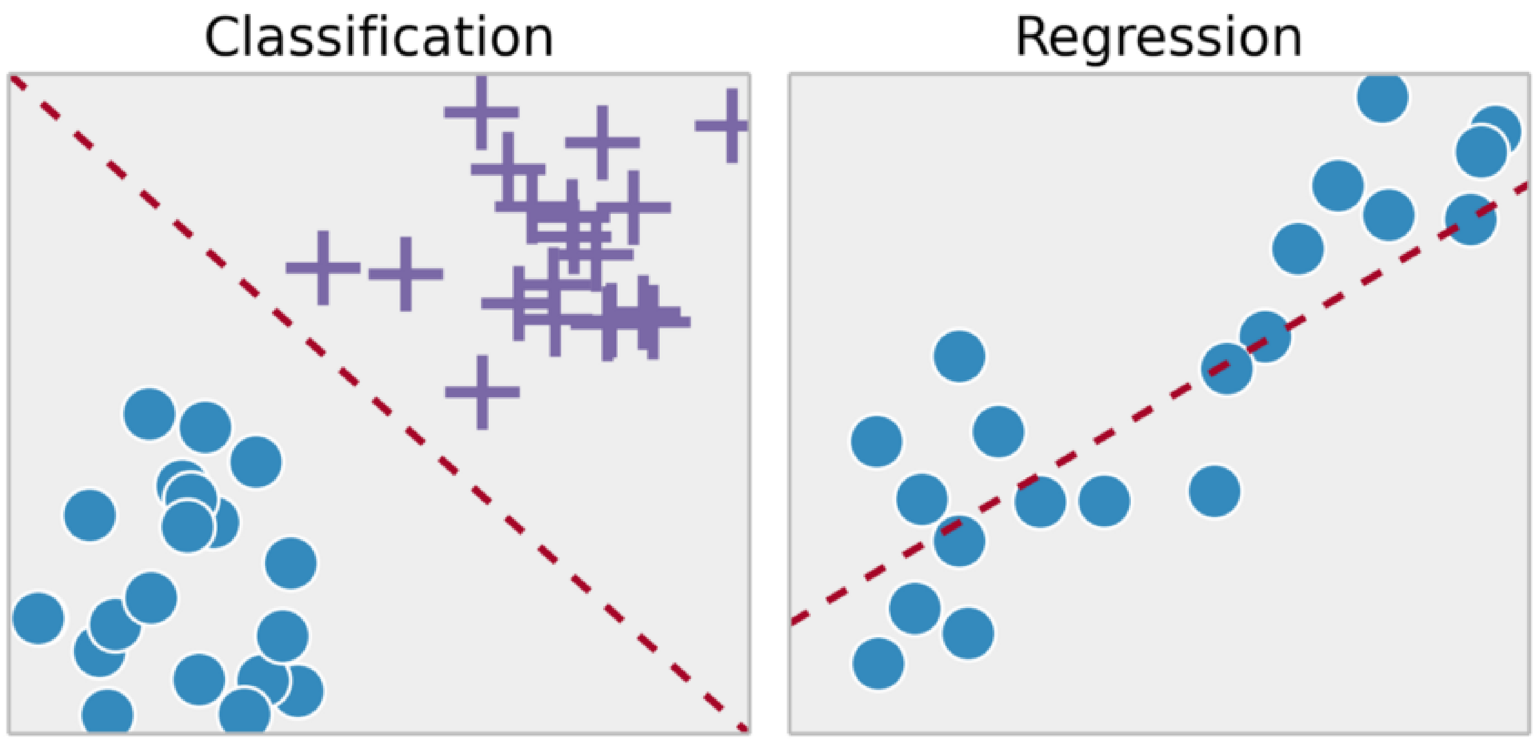


Auto-Encoders

主讲：龙良曲

Outline

Supervised Learning

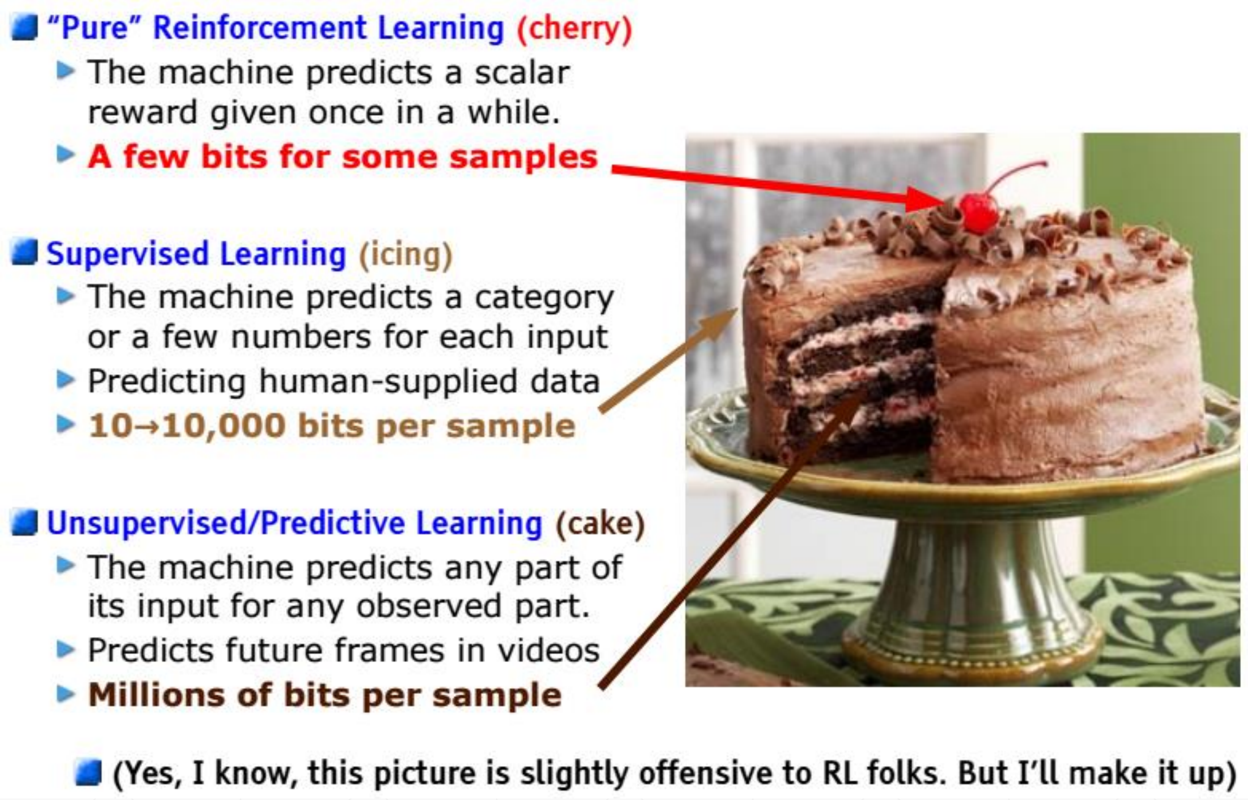


https://towardsdatascience.com/supervised-vs-unsupervised-learning-14f68e32ea8d

Massive Unlabeled data

Facebook: Users Upload 300 Million Images A Day (So Please Let Us Buy Instagram)

Unsupervised Learning

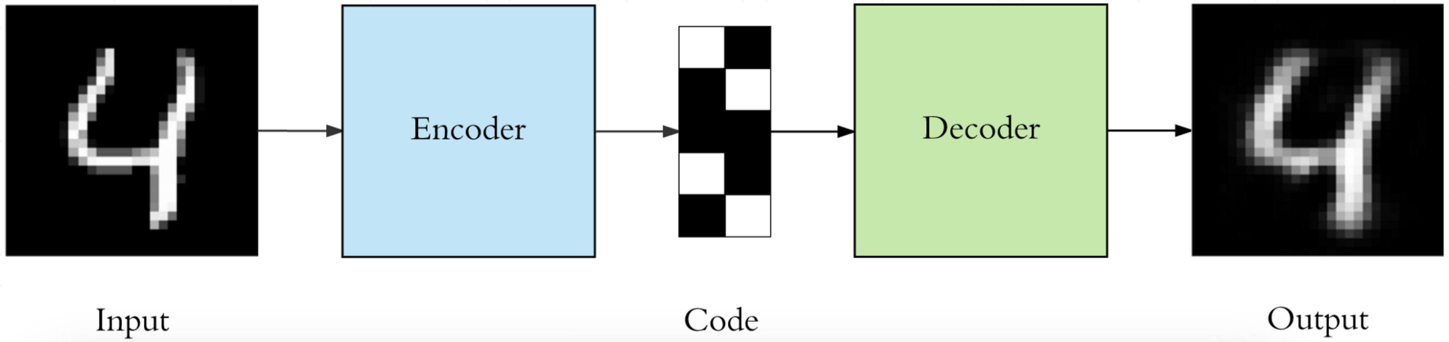


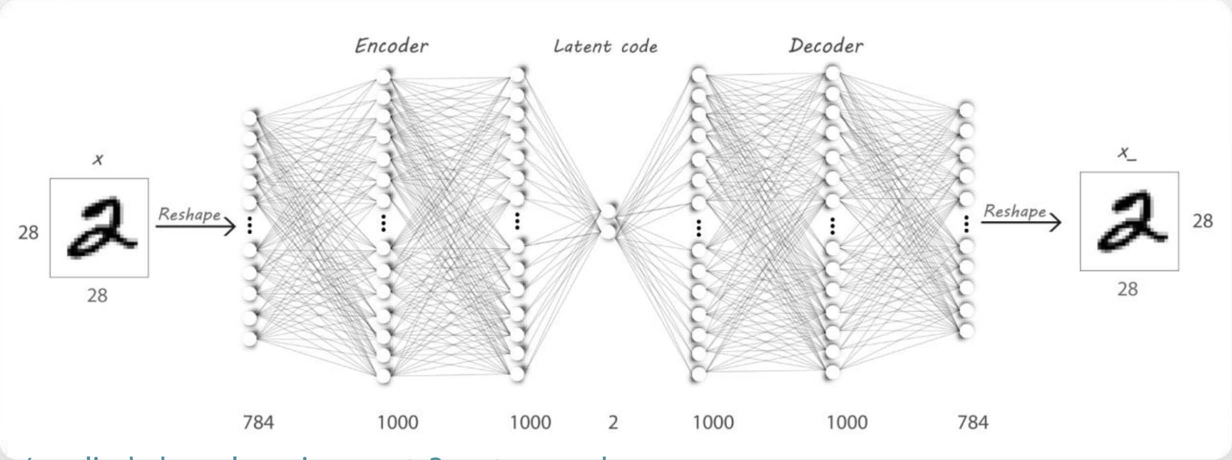
https://medium.com/intuitionmachine/predictive-learning-is-the-key-to-deep-learning-acceleration-93e063195fd0

Why needed

- Dimension reduction
- Preprocessing: Huge dimension, say 224x224, is hard to process
- Visualization: https://projector.tensorflow.org/
- Taking advantages of unsupervised data
- Compression, denoising, super-resolution …

Auto-Encoders





https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798
https://towardsdatascience.com/a-wizards-guide-to-adversarial-autoencoders-part-1-autoencoder-d9a5f8795af4

How to Train?

- Loss function for binary inputs

$$l(f(\mathbf{x})) = -\sum_k \left(x_k \log(\widehat{x}_k) + (1 - x_k)\log(1 - \widehat{x}_k)\right)$$

  - Cross-entropy error function (reconstruction loss) $f(\mathbf{x}) \equiv \widehat{\mathbf{x}}$

- Loss function for real-valued inputs

$$l(f(\mathbf{x})) = \frac{1}{2}\sum_k (\widehat{x}_k - x_k)^2$$

  - sum of squared differences (reconstruction loss)
  - we use a linear activation function at the output

PCA V.S. Auto-Encoders

- PCA, which finds the directions of maximal variance in high-dimensional data, select only those axes that have the largest variance.

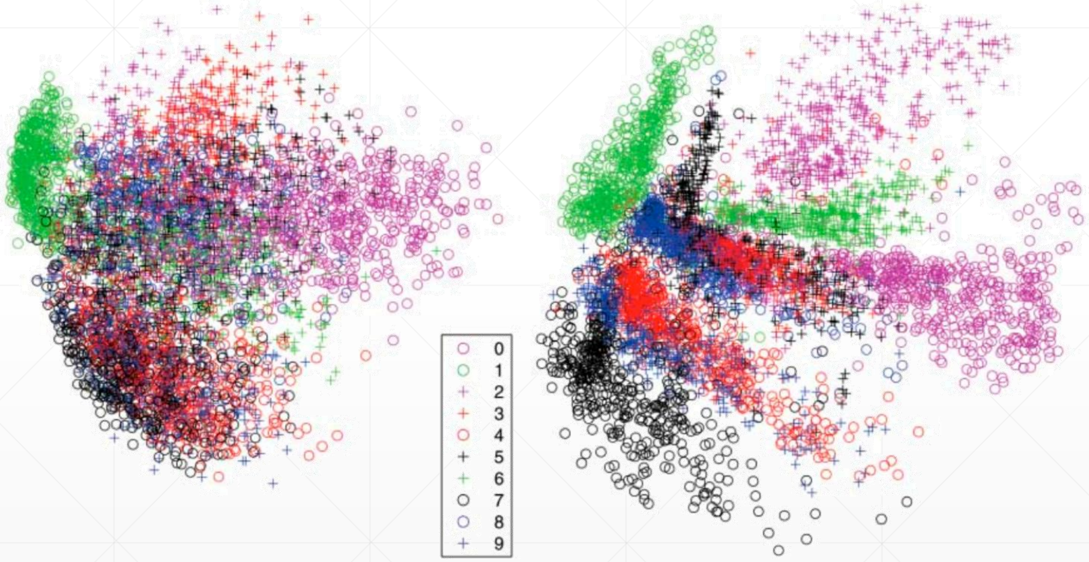
- The linearity of PCA, however, places significant limitations on the kinds of feature dimensions that can be extracted.

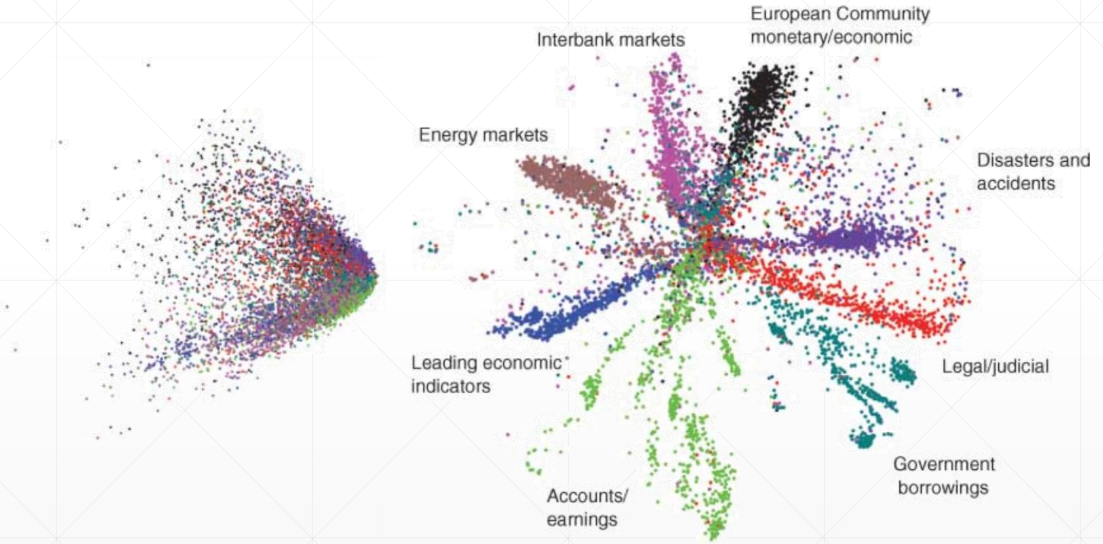
PCA V.S. Auto-Encoders

A comparison of reconstruction by an autoencoder (middle) and PCA (bottom) to original image inputs (top)

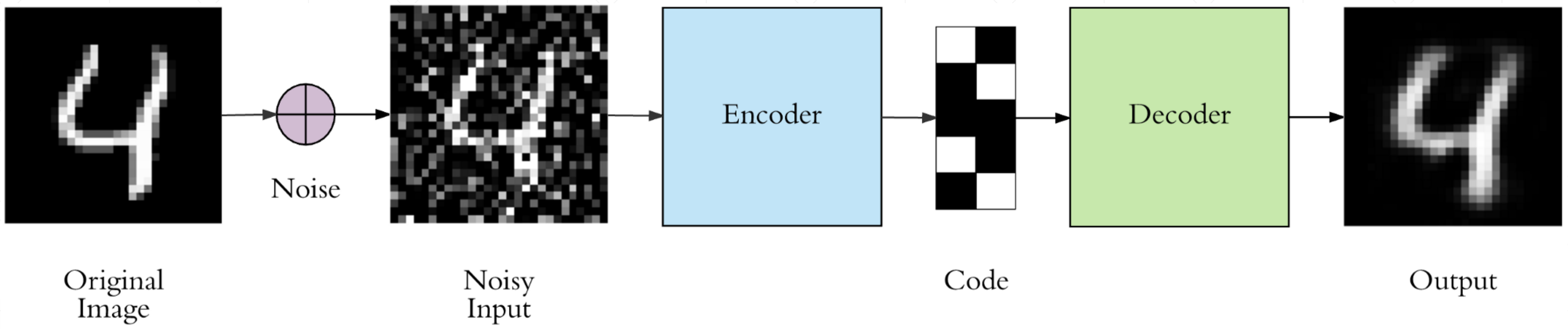
http://nikhilbuduma.com/2015/03/10/the-curse-of-dimensionality/
https://www.cs.toronto.edu/~urtasun/courses/CSC411/14_pca.pdf

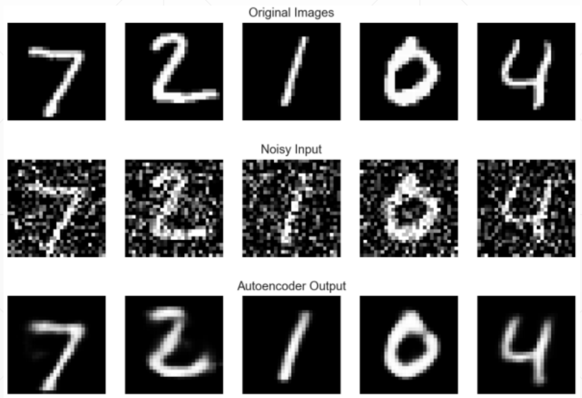
PCA V.S. Auto-Encoders



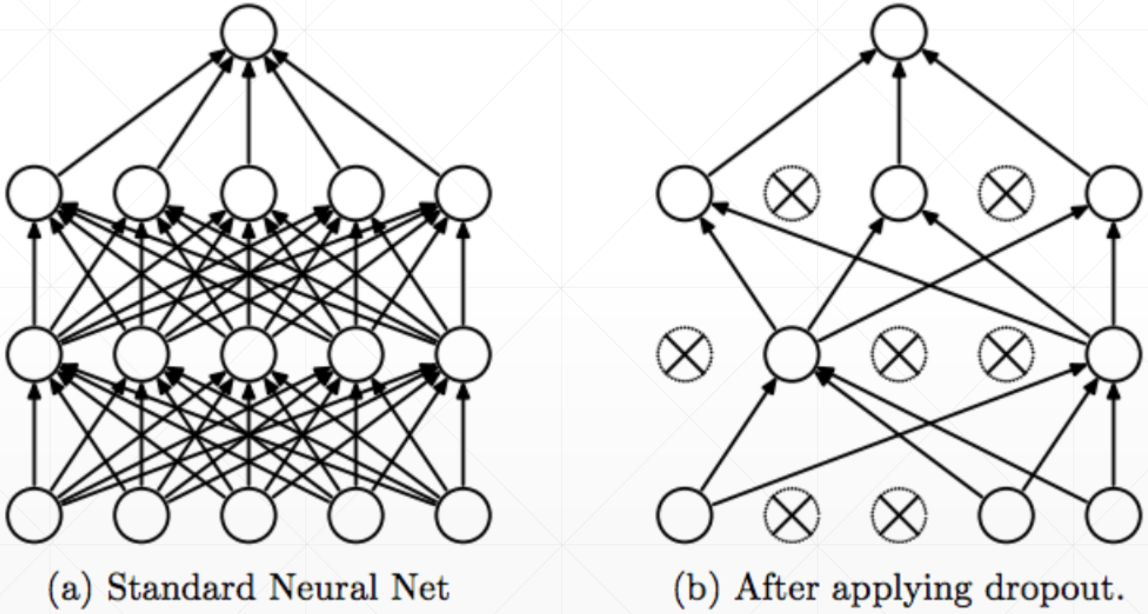


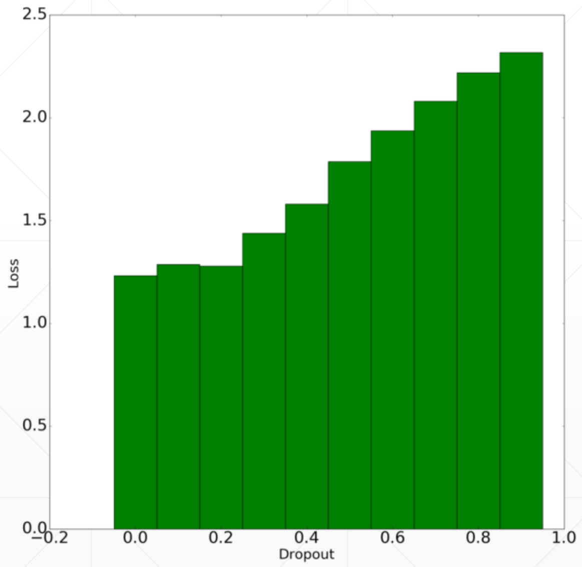
Denoising AutoEncoders

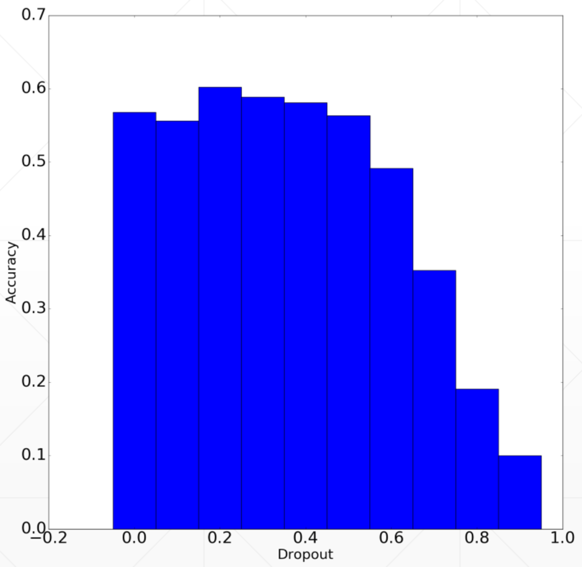




Dropout AutoEncoders



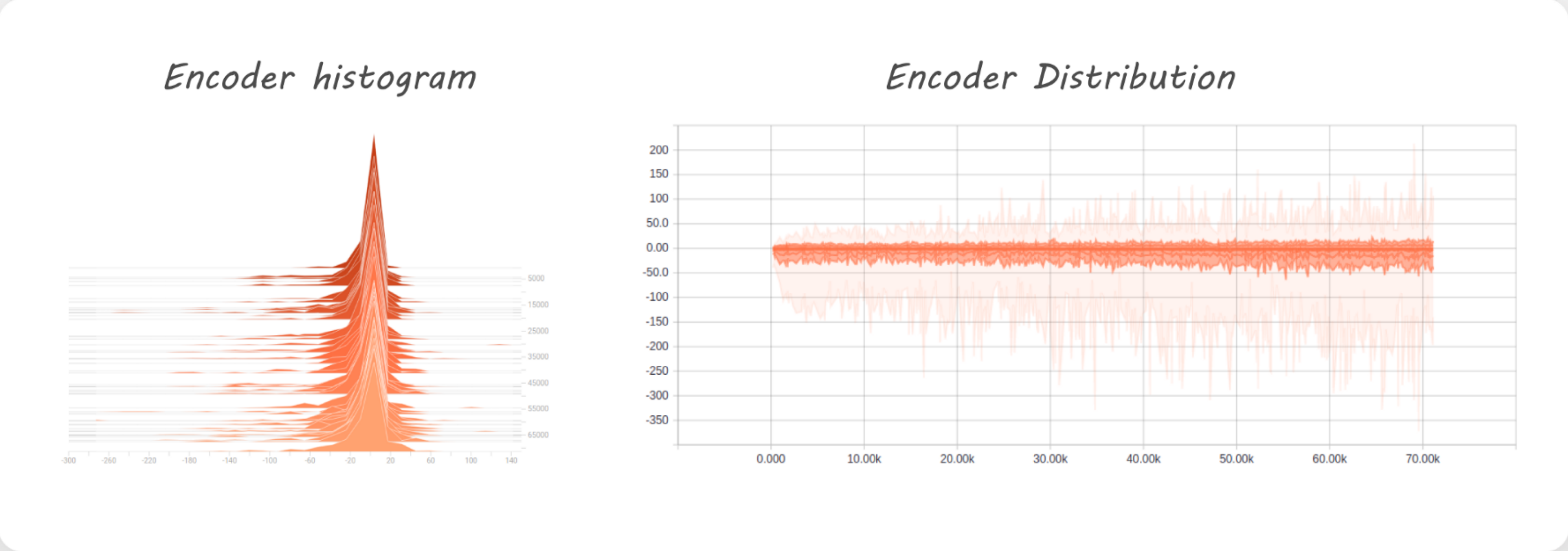




https://medium.com/@amarbudhiraja/https-medium-com-amarbudhiraja-learning-less-to-learn-better-dropout-in-deep-machine-learning-74334da4bfc5

Adversarial AutoEncoders

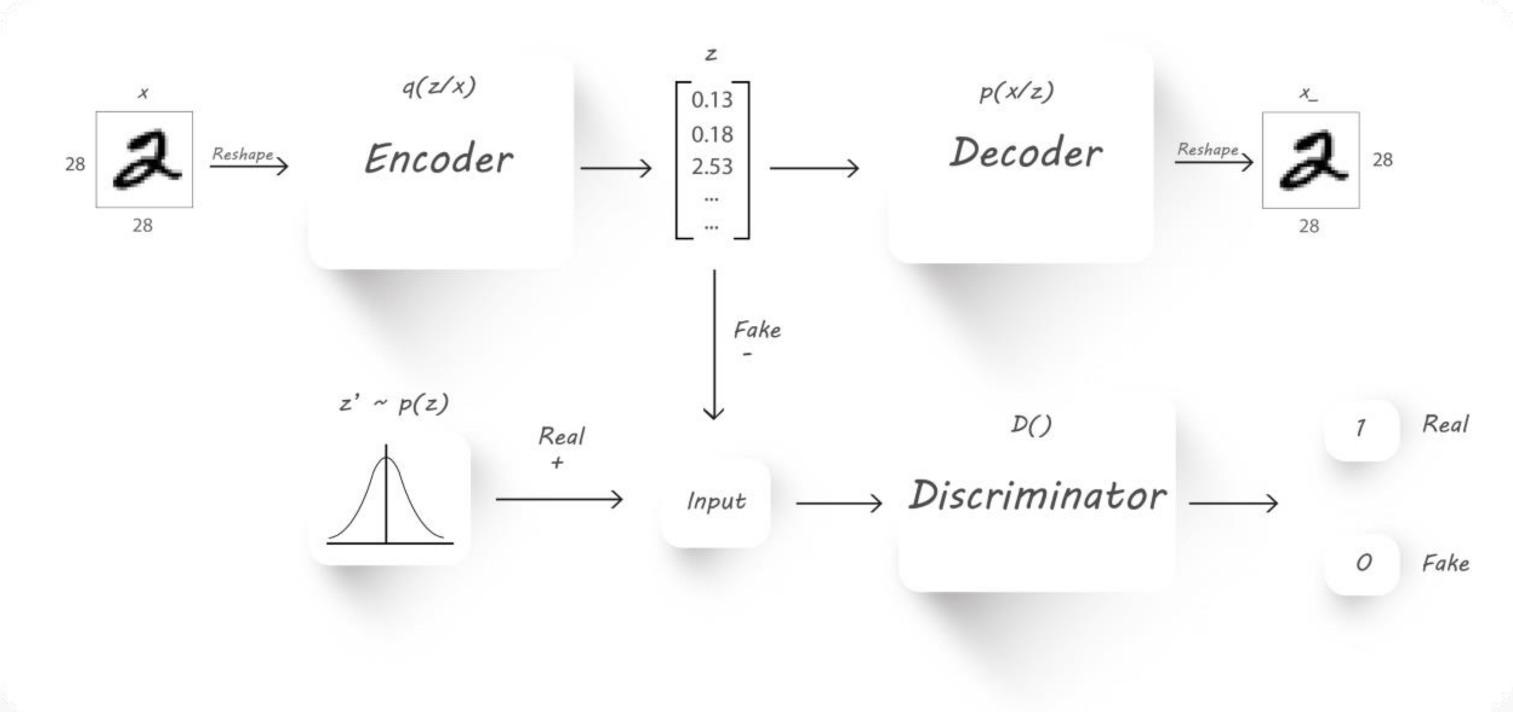
- Distribution of hidden code



https://towardsdatascience.com/a-wizards-guide-to-adversarial-autoencoders-part-2-exploring-latent-space-with-adversarial-2d53a6f8a4f9

Adversarial AutoEncoders

- Give more details after GAN



https://towardsdatascience.com/a-wizards-guide-to-adversarial-autoencoders-part-2-exploring-latent-space-with-adversarial-2d53a6f8a4f9

Another Approach: $q(z) \rightarrow p(z)$

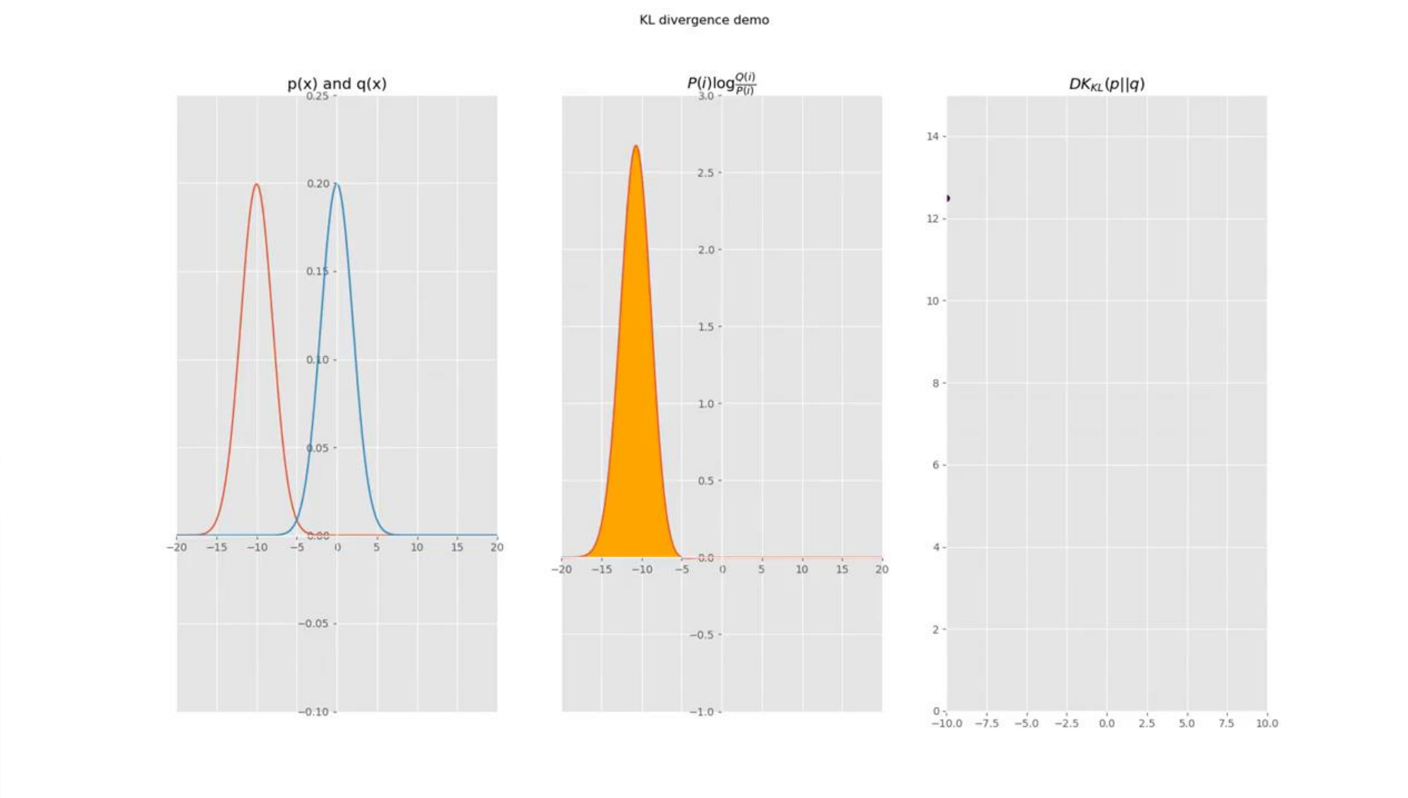
- Explicitly enforce

$$l_i(\theta, \phi) = -E_{z \sim q_\theta(z|x_i)}[\log p_\phi(x_i|z)] + KL(q_\theta(z|x_i)||p(z))$$

$$\mathrm{KL}(P||Q) = \int_{-\infty}^{\infty} p(x) \log \frac{p(x)}{q(x)}\, dx$$

https://stats.stackexchange.com/questions/188903/intuition-on-the-kullback-leibler-kl-divergence

Intuitively comprehend KL(p◎q)



Maximize Likelihood

$$E_{z \sim q_\theta(z|x_i)}[\log p_\phi(x_i|z)]$$

- Loss function for binary inputs

$$l(f(\mathbf{x})) = -\sum_k (x_k \log(\widehat{x}_k) + (1 - x_k)\log(1 - \widehat{x}_k))$$

  - Cross-entropy error function (reconstruction loss) $f(\mathbf{x}) \equiv \widehat{\mathbf{x}}$

- Loss function for real-valued inputs

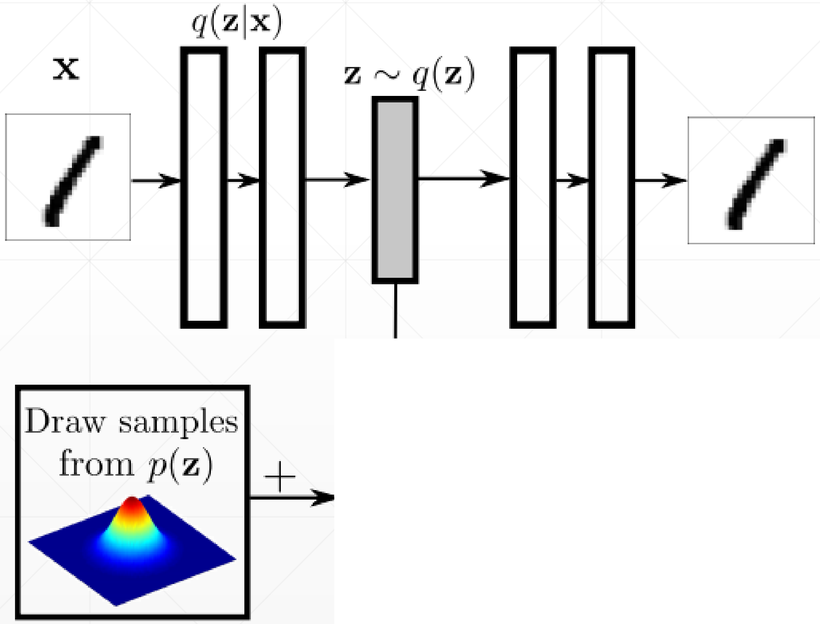
$$l(f(\mathbf{x})) = \frac{1}{2}\sum_k (\widehat{x}_k - x_k)^2$$

  - sum of squared differences (reconstruction loss)
  - we use a linear activation function at the output

Minimize KL Divergence

- Evidence Lower BOund

$$KL(q_\theta(z|x_i)||p(z))$$



How to compute KL between $q(z)$ and $p(z)$

$p(z_i) \sim N(\mu_1, \sigma_1^2)$

$q(z_i) \sim N(\mu_2,\ \sigma_2^2)$

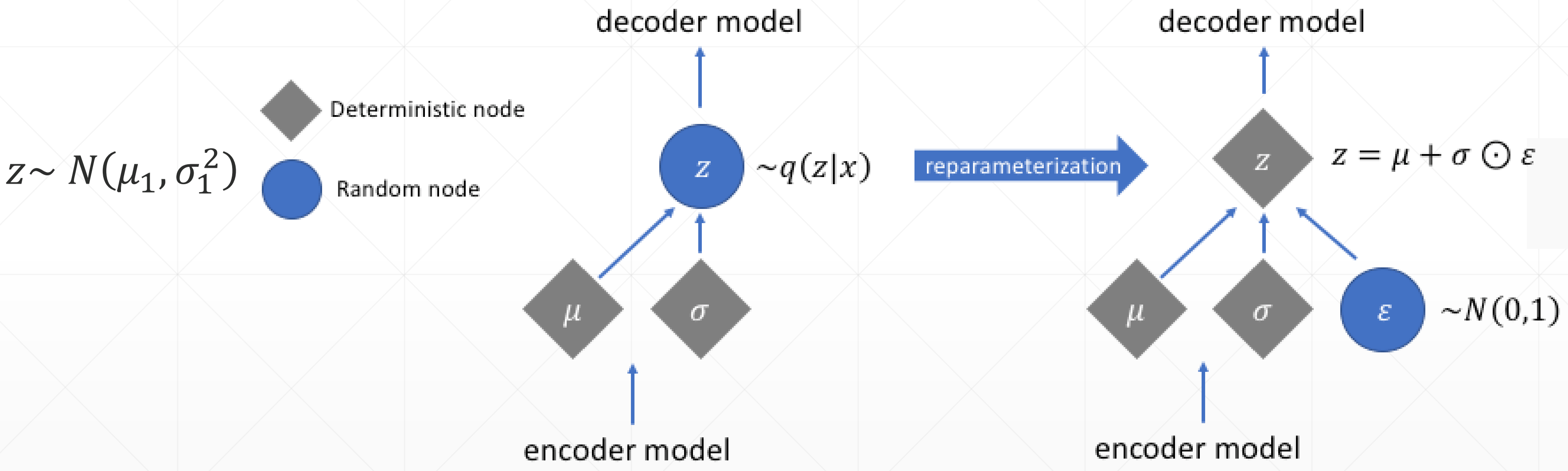
$$KL(p, q) = -\int p(x) \log q(x) dx + \int p(x) \log p(x) dx$$

$$= \frac{1}{2}\log(2\pi\sigma_2^2) + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2}(1 + \log 2\pi\sigma_1^2)$$

$$= \log\frac{\sigma_2}{\sigma_1} + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2}$$

https://stats.stackexchange.com/questions/7440/kl-divergence-between-two-univariate-gaussians

Sample() is not differentiable

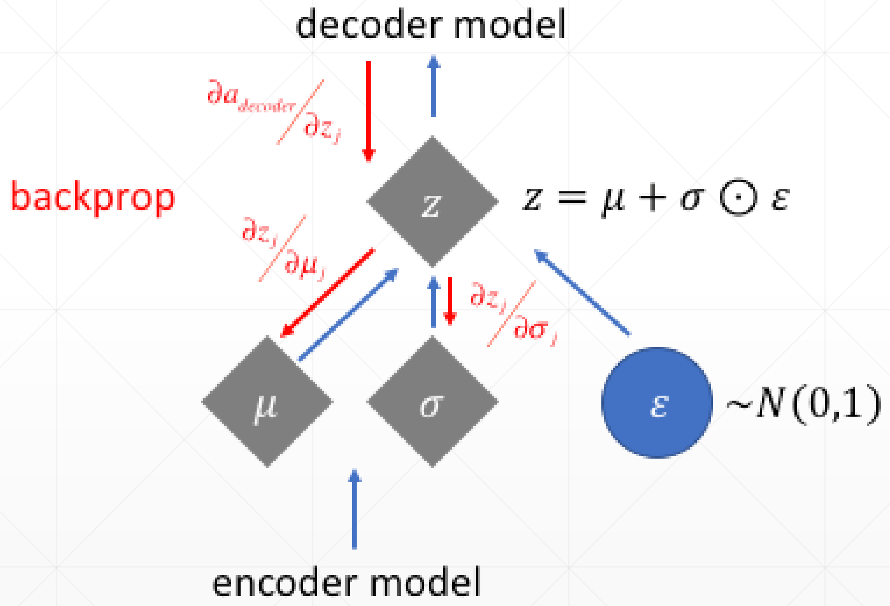


$$z = \mu + \sigma \odot \epsilon,$$

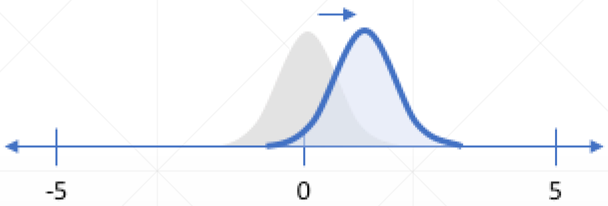
$$\epsilon \sim Normal(0, 1).$$

https://www.jeremyjordan.me/variational-autoencoders/

Reparameterization trick



Penalizing reconstruction loss encourages the distribution to describe the input



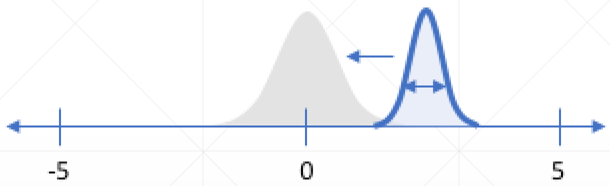
Our distribution deviates from the prior to describe some characteristic of the data

Without regularization, our network can “cheat” by learning narrow distributions

With a small enough variance, this distribution is effectively only representing a single value

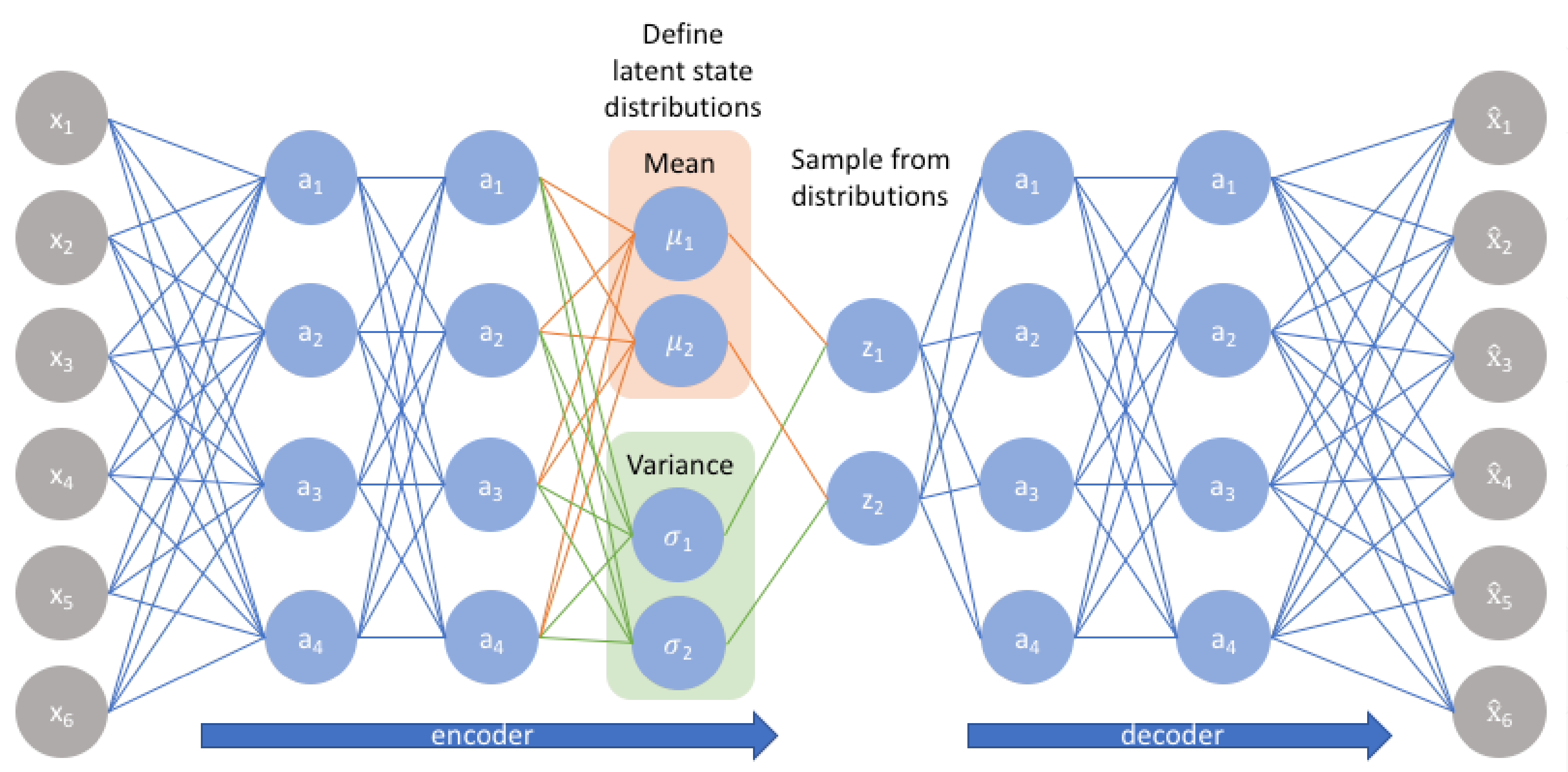
Penalizing KL divergence acts as a regularizing force

Attract distribution to have zero mean



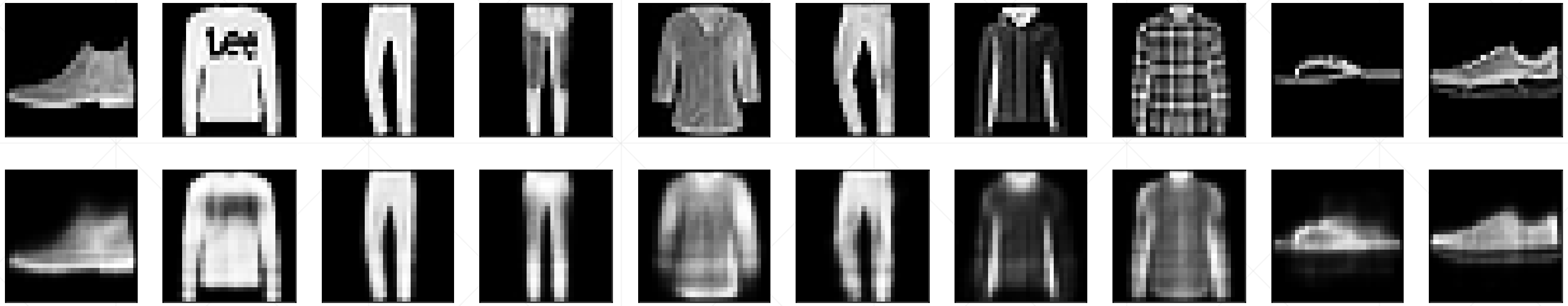
Ensure sufficient variance to yield a smooth latent space

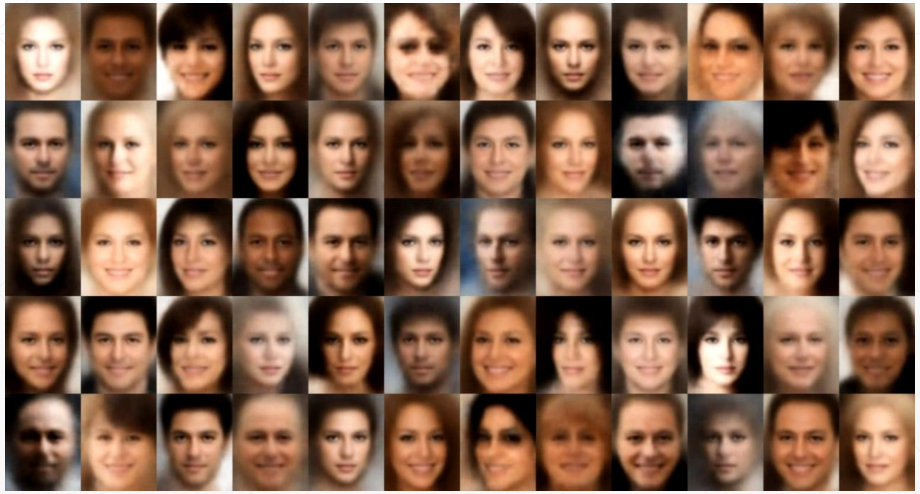
Too Complex!



https://www.jeremyjordan.me/variational-autoencoders/

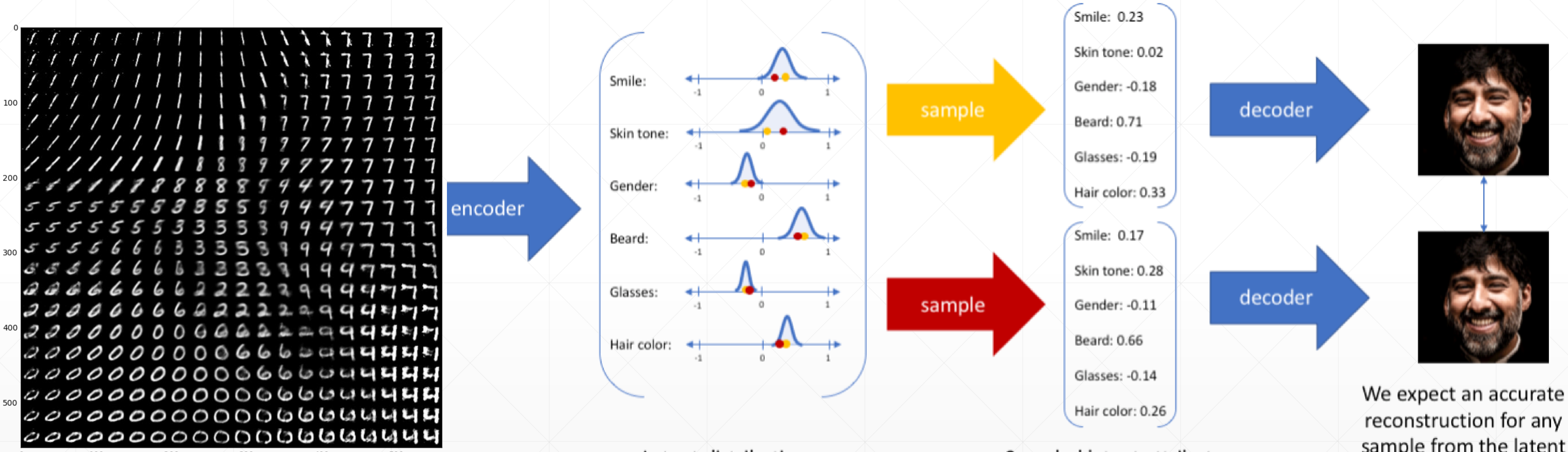
AE V.S. VAE





https://github.com/cryer/Variational_Auto-Encoder/blob/master/image/reconst_images_7.png

Generative model



Latent distributions

Sampled latent attributes

We expect an accurate reconstruction for any sample from the latent state distributions

https://jmetzen.github.io/2015-11-27/vae.html

VAE V.S. GAN

https://medium.com/@wuga/generate-anime-character-with-variational-auto-encoder-81e3134d1439

下一课时

Thank You.