# Un- and Self-Supervised Learning

## Lecture 13

Automatic Image Analysis

July 8, 2021

Technische
Universität
Berlin

- 1,281,167 training images
- 1000 object classes

- How much are 1000 concepts compared to all the concepts humans use?
- Imagine we would need to label 1000 images per concept.
- New concepts are created and change all the time.

- Can we learn without a supervision signal in form of labels?
- In an un- or rather self-supervised manner?

- Similar to a human child in the first few month after birth.
- Purely by observing the world.
- It's hard to define what truly unsupervised learning could be. Therefore the term self-supervised learning is a better fit.

- Pretext tasks
- Energy based methods
- Generative learning

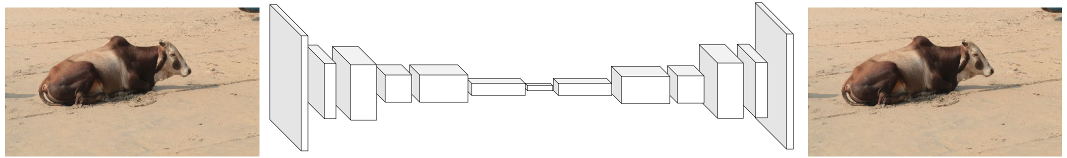
- We will look at three big topics today.
- At least the second and third topic could not only fill a lecture but a full course on their own.
- E.g. CS 236: Deep Generative Models (Stanford) or CS 294-158 Deep Unsupervised Learning (Berkeley)

- In generative learning often, people often just want to generate visual content though.

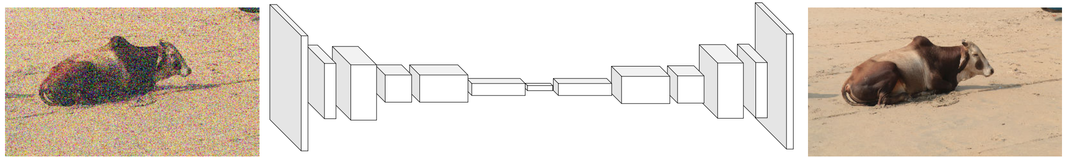
Idea:

- Train a neural network with an objective that doesn't need labels.
- Evaluate representation on a downstream task. E.g. performance on ImageNet with or without finetuning.

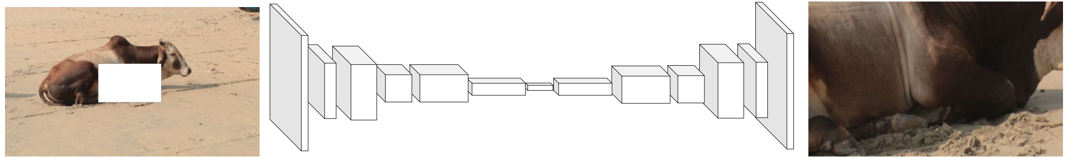
What objective could that be?



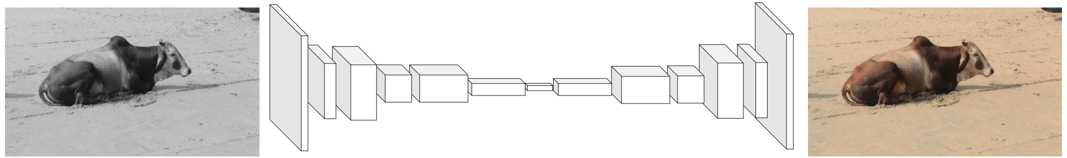
- Train an autoencoding network reconstruct an image after coarse feature layer.
- Use encoding network for downstream task.



- Same as before but apply distortion function $d(I)$ before feeding the image into the network.
- Use encoding network for downstream task.

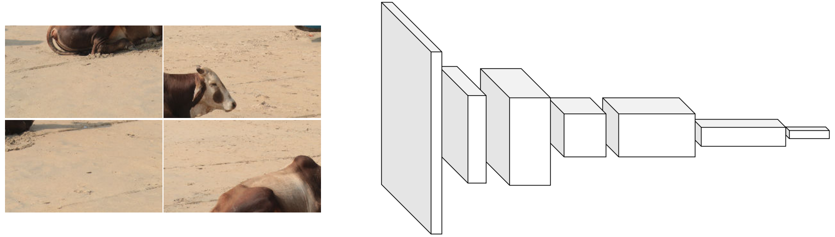


- Predict one part of the data from another.
- Can also be a random part of the image or e.g. the bottom half or frames of a video sequence.
- Context Encoders: Feature Learning by Inpainting, Pathak et al., CVPR 2016



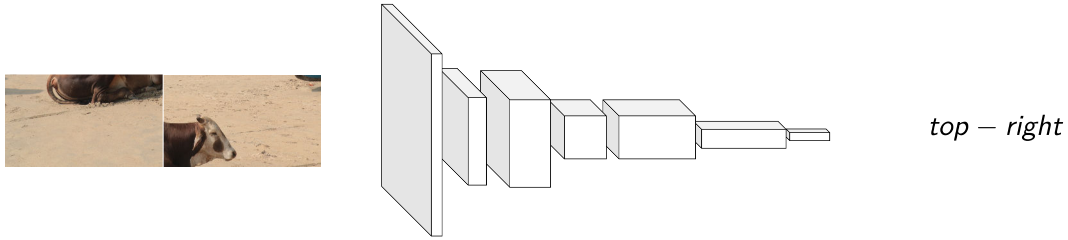
- Similar to inpainting we predict a left-out property the data.
- Colorful Image Colorization, Zhang et al., ECCV 2016
- Tracking Emerges by Colorizing Videos, Vondrick et al., ECCV 2018

## Pretext tasks: Frame permutation

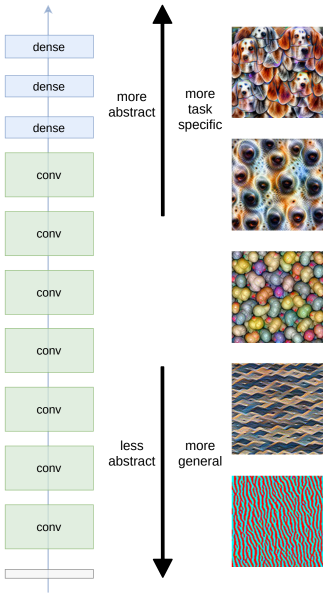


$4, 2, 1, 3$

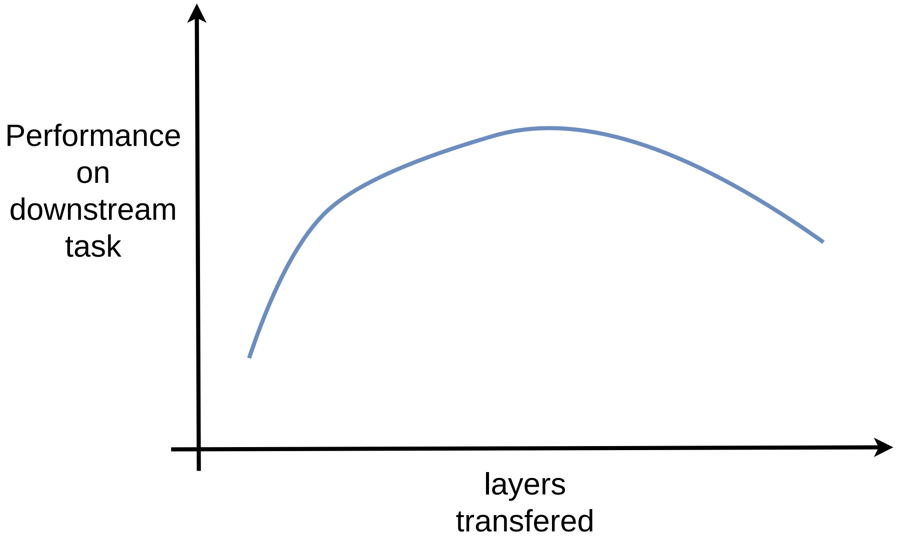
- We can also formulate the pretext task as classification problem. Here one of $n!$ possible permutations.
- Can also be done with video frames.
- Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles, Noroozi & Favaro, ECCV 2016




- Or as a discrete spatial relation
- Unsupervised Visual Representation Learning by Context Prediction, Doersch et al., ICCV 2015

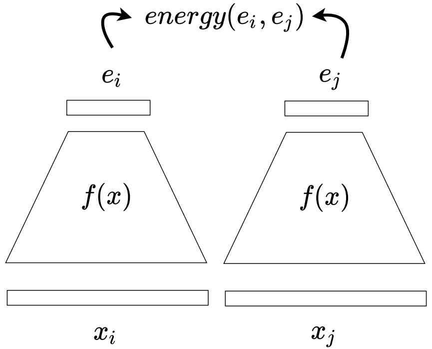



- Same as for transfer learning with supervised pretraining.
- Replace some layers, fine tune some layers.

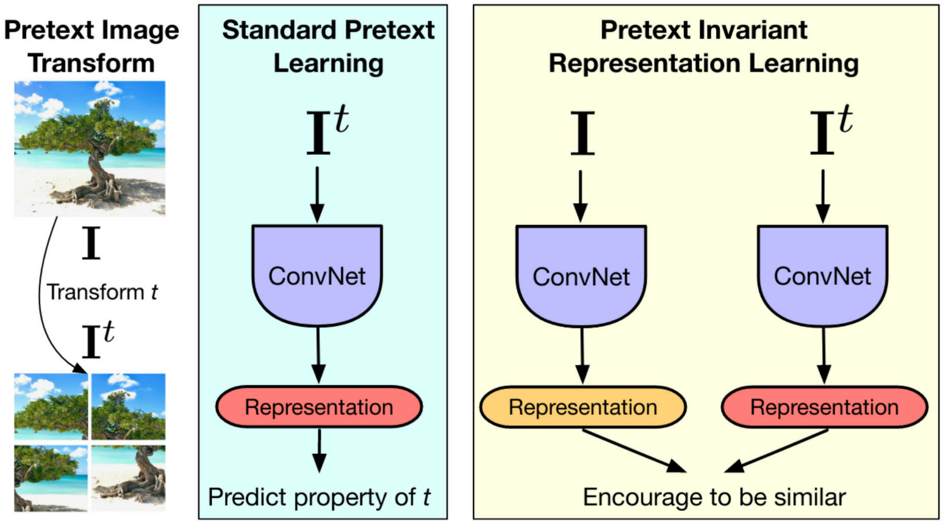



- Problem: learned representations are very task specific

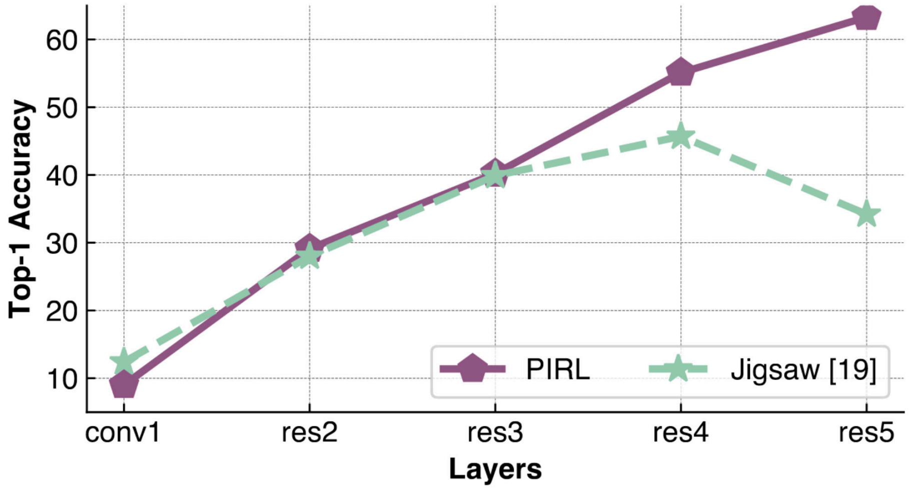
$$\mathit{similarity}(x_i, x_j) > \mathit{similarity}(x_i, x_k) \Rightarrow \mathit{energy}(e_i, e_j) < \mathit{energy}(e_i, e_k)$$



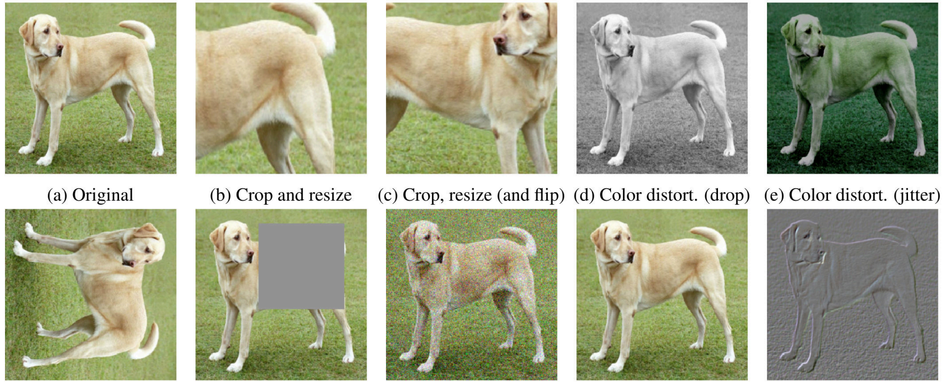

- For energy-based learning we often use what is called Siamese networks.
- Two (almost) identical networks, that share weights.
- We could summarize the methods in this chapter also as Siamese Representation Learning.
- If the inputs to the two networks are compatible in some way, the energy should be low, otherwise high.
- Similarity does not mean similar appearance in pixel space.




- Image from Self-Supervised Learning of Pretext-Invariant Representations, Misra & Maaten, CVPR 2020




- Image from Self-Supervised Learning of Pretext-Invariant Representations, Misra & Maaten, CVPR 2020



(a) Original (b) Crop and resize (c) Crop, resize (and flip) (d) Color distort. (drop) (e) Color distort. (jitter)

(f) Rotate $\{90°, 180°, 270°\}$ (g) Cutout (h) Gaussian noise (i) Gaussian blur (j) Sobel filtering

- Image from A Simple Framework for Contrastive Learning of Visual Representation, Chen et al., ICML 2020

Good negative samples are very important

- Have huge batch sizes
- Use memory banks (momentum of activations)
- Momentum on the weights of the siamese twin

- Huge batch sizes are easy to implement but have heavy compute demands
  A Simple Framework for Contrastive Learning of Visual Representations, Chen et al., ICML 2020
- Compute efficient but memory bank needs a lot of RAM
  Unsupervised Feature Learning via Non-Parametric Instance-level Discrimination, Wu et al., CVPR 2018
- Saves memory but needs extra forward pass
  Momentum Contrast for Unsupervised Visual Representation Learning, He et al., CVPR 2020

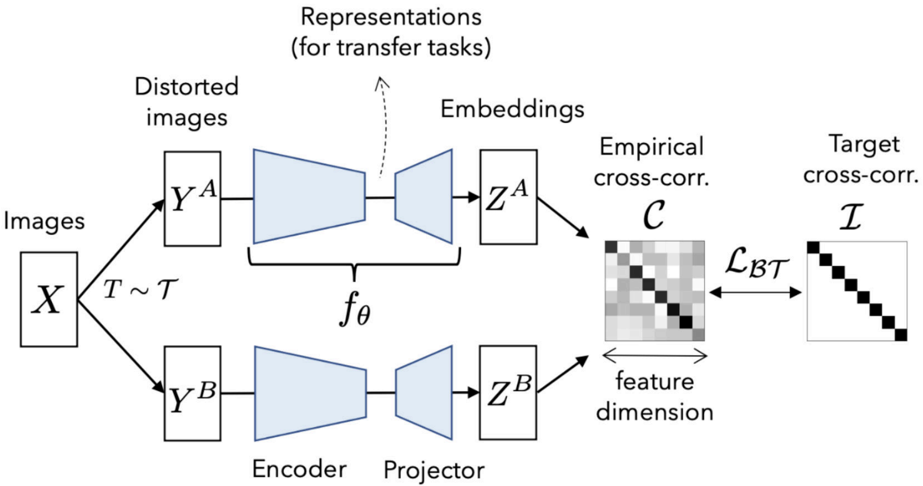
There are other ways to approach this (clustering, distillation)

- DeepCluster, Sela, SwAV
- BYOL, SimSiam

- We are gonna skip those for today. Unfortunately we can't talk about everything :(
- There is a very nice lecture by Ishan Misra though, if you want to learn more: https://www.youtube.com/watch?v=8L10w1KoOU8

$$f_i(I) = f_i(d(I))$$
$$f_i(I) \neq f_j(d(I))$$

- This equations are a dramatically oversimplified sketch of the idea.
- While neurons should respond the same to an image and its distorted version, they should all respond differently.
- We don't have spare neurons, so we don't want redundancy in their activations.
- Possible principles underlying the transformation of sensory messages, Horace Barlow, 1961




- Our objective is to make the correlation matrix a diagonal matrix.
- To prevent constant but decorrelated output, $Z_a$ and $Z_b$ are standardized before the correlation matrix is computed.
- Image from Barlow Twins: Self-Supervised Learning via Redundancy Reduction, Zbontar et al., ICML 2021

**Algorithm 1** PyTorch-style pseudocode for Barlow Twins.

```
# f: encoder network
# lambda: weight on the off-diagonal terms
# N: batch size
# D: dimensionality of the embeddings
#
# mm: matrix-matrix multiplication
# off_diagonal: off-diagonal elements of a matrix
# eye: identity matrix

for x in loader: # load a batch with N samples
    # two randomly augmented versions of x
    y_a, y_b = augment(x)

    # compute embeddings
    z_a = f(y_a) # NxD
    z_b = f(y_b) # NxD

    # normalize repr. along the batch dimension
    z_a_norm = (z_a - z_a.mean(0)) / z_a.std(0) # NxD
    z_b_norm = (z_b - z_b.mean(0)) / z_b.std(0) # NxD

    # cross-correlation matrix
    c = mm(z_a_norm.T, z_b_norm) / N # DxD

    # loss
    c_diff = (c - eye(D)).pow(2) # DxD
    # multiply off-diagonal elems of c_diff by lambda
    off_diagonal(c_diff).mul_(lambda)
    loss = c_diff.sum()

    # optimization step
    loss.backward()
    optimizer.step()
```

- Image from Barlow Twins: Self-Supervised Learning via Redundancy Reduction, Zbontar et al., ICML 2021

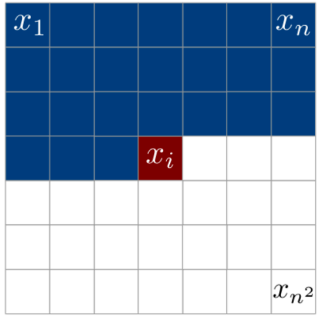
What's generative learning?

We want to model the data distribution $p(x)$ directly.

- Where $x$ is a sample image.
- How is this even possible? Let's see ...

## PixelRNN/PixelCNN

$$p(x) = p(x_1, ..., x_n) = \prod_{i}^{n^2} p(x_i) p(x_i | x_1, ..., x_{i-1})$$




- Fully Visible Belief Network
- Product of distributions using chain rule (decompose likelihood of an image into pixel probabilities).
- Train RNN to classify pixels (e.g. 1 out of 255).
- Also possible to formulate as CNN, but still one forward pass per pixel necessary at test time.
- Image from Pixel Recurrent Neural Networks, van den Oord, ICML 2016

$$z = \begin{pmatrix} haircolor \\ skintone \\ beard \\ gender \\ classes \\ expression \end{pmatrix} \Rightarrow$$

- Image from Auto-Encoding Variational Bayes, Kingma & Welling, ICLR 2014
- 

$$p_\theta(x) = \int p_\theta(x|z)p_\theta(z)dz$$

$$z = \begin{pmatrix} haircolor \\ skintone \\ beard \\ gender \\ classes \\ expression \end{pmatrix} \Rightarrow$$

- $p_\theta$ is the data likelihood we want to maximize.
- We can approximate $p(z)$ e.g. as Gaussian.
- We can learn $p(x|z)$ e.g. with a generator network.
- However, the integral over $z$ is intractable.

ELBO (evidence lower bound)

$$E_{q(z|x)} \log p(x|z) - KL(q(z|x)||p(z))$$

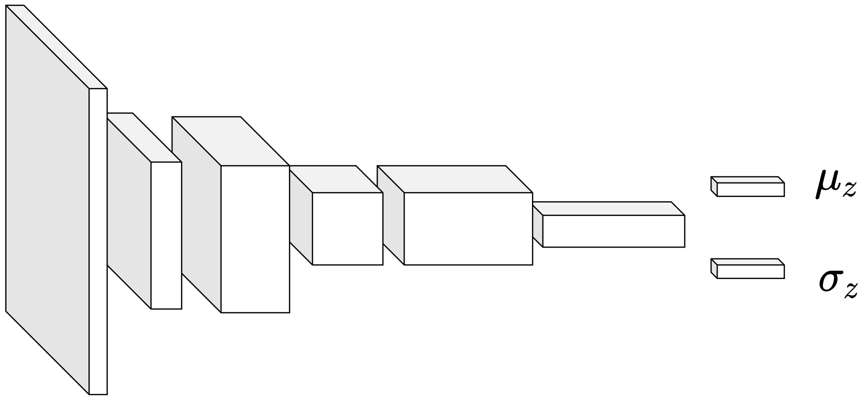
- Luckily it turns out that this term is a lower bound on our intractable data likelihood.
- $KL$ is the Kullback-Leibler divergence, a similarity measurement for probability distributions.
- $q(z|x)$ is a tractable approximation of the intractable $p(z|x)$.
- Yes, there is a lot of math we just skipped. You can find a full derivation here: https://www.youtube.com/watch?v=uaaqyVS9-rM&t=1182s

$$E_{q(z|x)} \log p(x|z) - KL(q(z|x)||p(z))$$

Let's maximize it!

- Luckily it turns out that this term is a lower bound on our intractable data likelihood.
- $KL$ is the Kullback-Leibler divergence, a similarity measurement for probability distributions.
- $q(z|x)$ is a tractable approximation of the intractable $p(z|x)$.

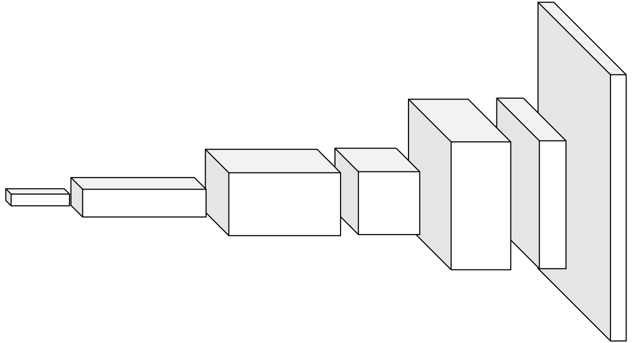
$$q(z|x) = \mathcal{N}(\mu_z, \sigma_z)$$



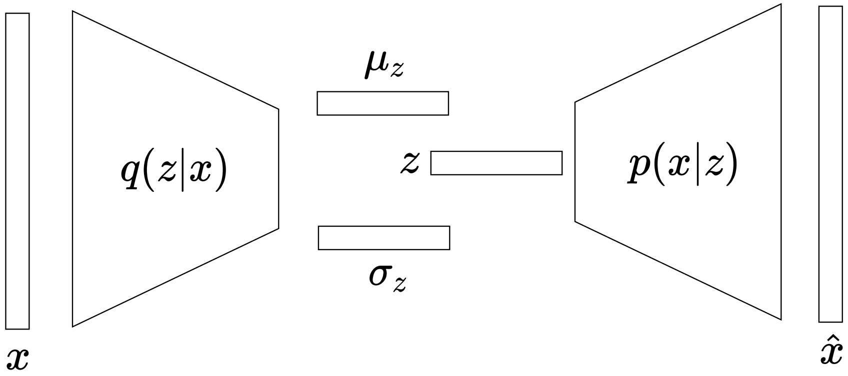

- Let's just assume the $p(z)$ is gaussian distributed.
- And let's additionally assume all elements of $z$ are independent.
- To approximate $q(z|x)$ we learn a mapping with a neural net.
- This is the encoder part of the variational autoencoder (sometimes called recognition model).

$MSE(x, \hat{x})$

$z \sim \mathcal{N}(\mu_z, \sigma_z)$



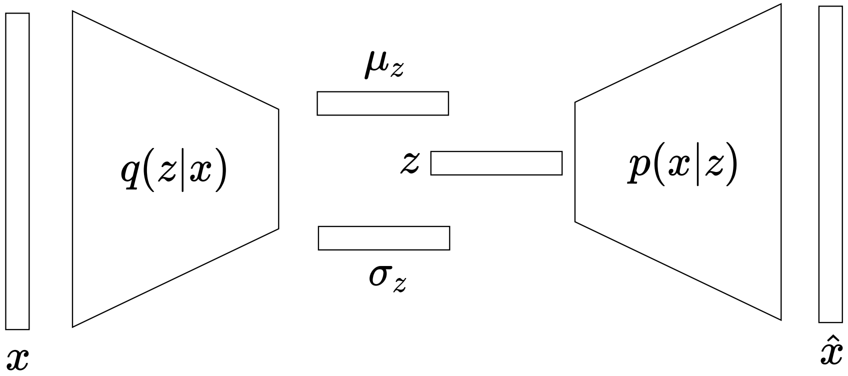
- minimize $MSE(x, \hat{x})$ to maximize $E_{q(z|x)} \log p(x|z)$
- This is the decoder part of the variational autoencoder (sometimes called generator model).



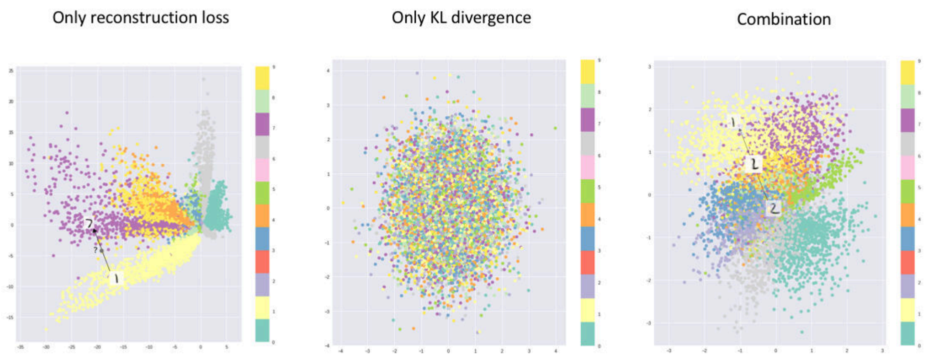

- Full VAE architecture for training.

## VAE: Reparameterization Trick

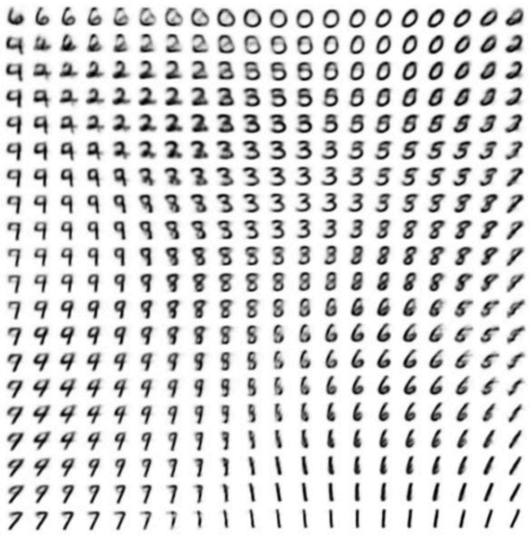
$$z = \mu_z + \epsilon\sigma_z \text{ with } \epsilon\ \mathcal{N}(0,1)$$



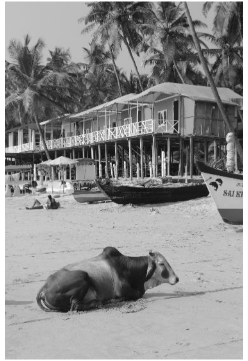

- We cannot backpropagate through $z \sim \mathcal{N}(\mu_z, \sigma_z)$
- Therefor we set to $z = \mu_z + \epsilon\sigma_z$ with $\epsilon\ \mathcal{N}(0,1)$
- This is called the reparameterization trick.

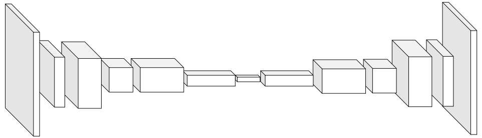



- At test time we draw $z$ from $p(z) = \mathcal{N}(0, 1)$.
- Enforcing $KL(q(z|x)||p(z))$ leads to a smooth latent state.
- Image from https://towardsdatascience.com/intuitively-understanding-variational-autoencoders-1bfe67eb5daf

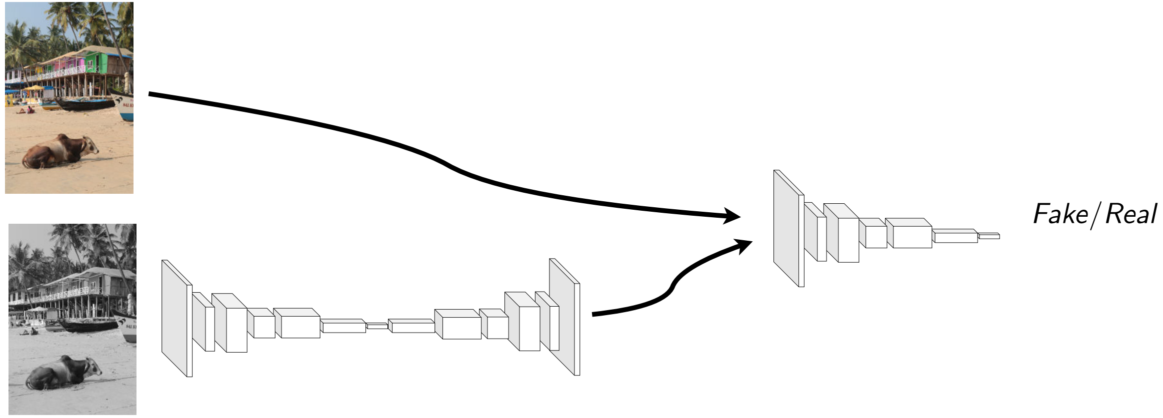


- A VAE trained to generate MNIST digits.
- A grid in the latent space leads to consistent generations in pixelspace.
- Image from Auto-Encoding Variational Bayes, Kingma & Welling, ICLR 2014

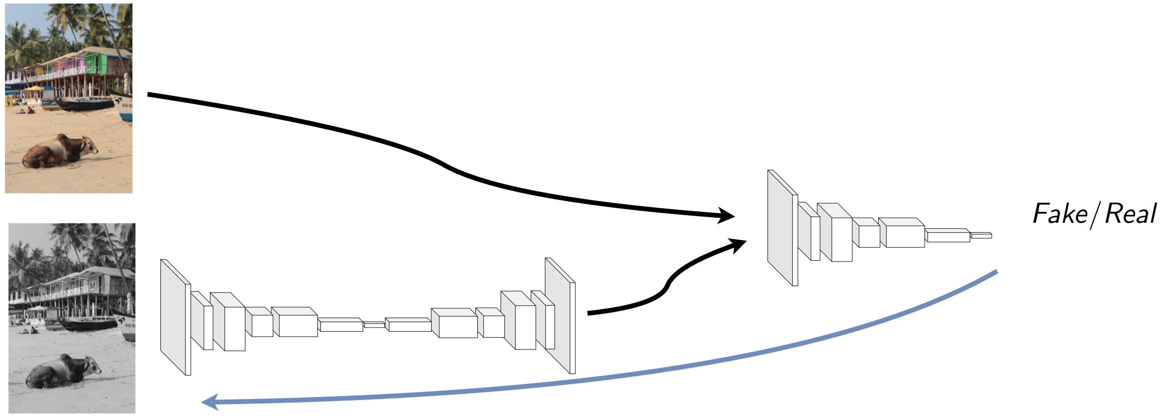




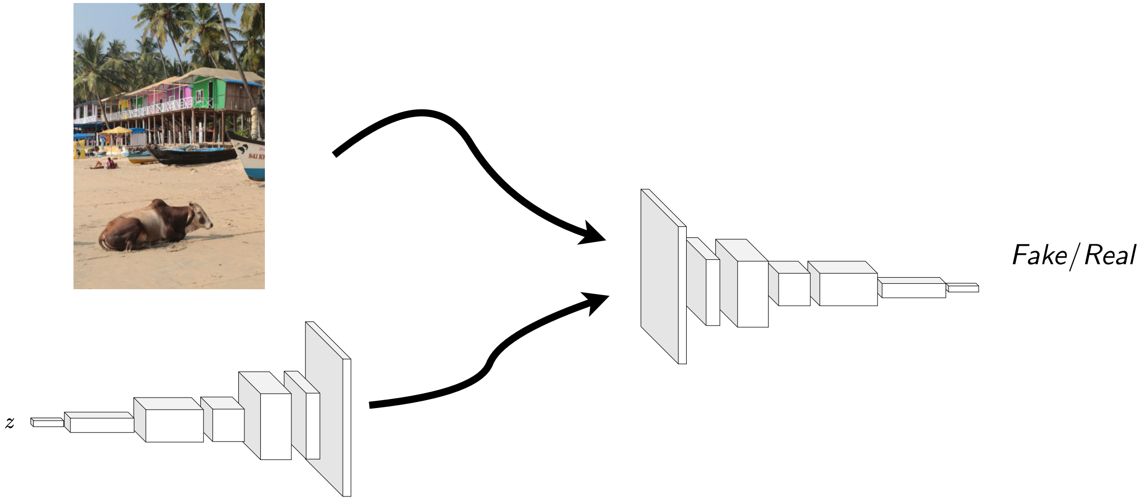
- If we have a generated image (e.g. from the VAE or from colorizing a grey scale image), we do not actually care if the image is exactly the same as the input image.
- We just want it to be realistic. But the MSE forces the output to be the same as the reference.




- Instead of formulating a good error measurement ourselves, we can train a classifier to distinguish between a real image and a generated (fake) image.
- This way we do not measure if the image looks similar to the original but only if the image looks realistic.



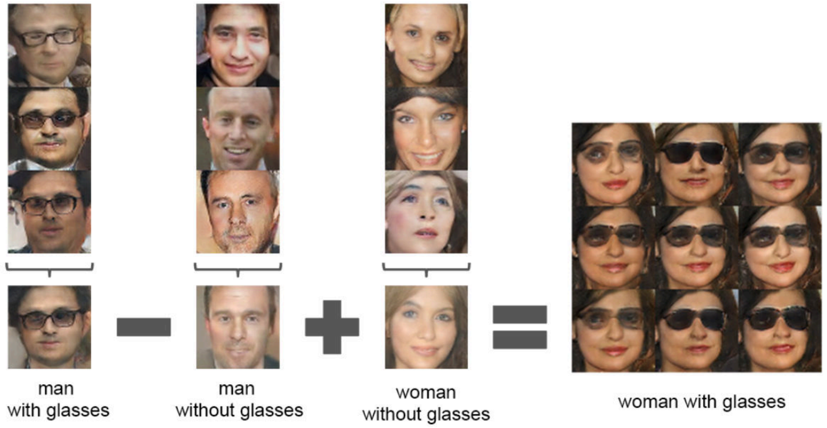

- After training the classifier (discriminator), we can backpropagate the negative gradient of the discriminator into the generator network.
- This way we train the generator to become a better forger. We can train both networks alternatingly, leading to ever better generator and discriminator.




- Instead of generating an image from an input encoding, we can also just generate an image from a random vector.
- This way the generator learns to map the input distribution $p(z)$ to the data distribution $p(x)$.
- The discriminator learns to distinguish if an image $x$ is within $p(x)$ or out of distribution.

- Generative Adversarial Networks, Goodfellow et al., NeurIPS 2014
- Training of the pair of networks is a mini-max game.

$$\min_{\theta_g} \max_{\theta_d} [E_{x \sim p_{data}} \log D_\theta(x) + E_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z)))]$$




- Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, Radford et al., ICLR 2016
- Paper also uses discriminator features for image classification and lists design guidelines for ConvNet architectures for GANS.

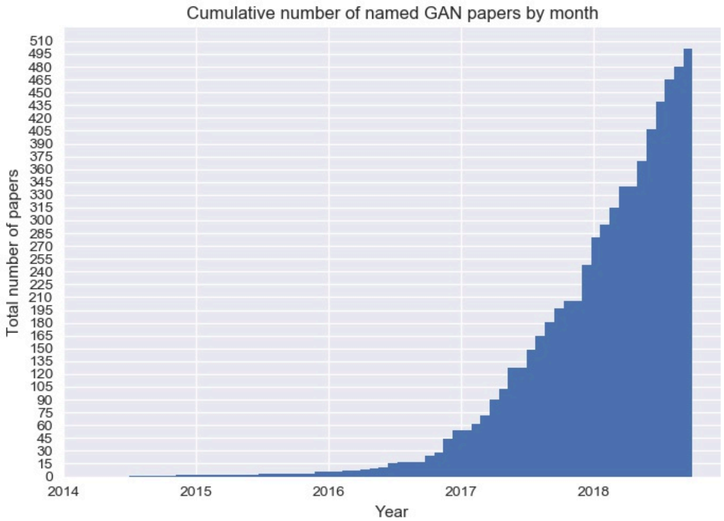
## Many improvements ...

- Wasserstein GAN, Arjovsky et al., 2017
- Improved Training of Wasserstein GANs, Gulrajani et al., 2017
- Progressive Growing of GANs for Improved Quality, Stability, and Variation, Karras et al., 2017

- GANs are hard to train and improvements to training stability were very important.

## GAN Zoo

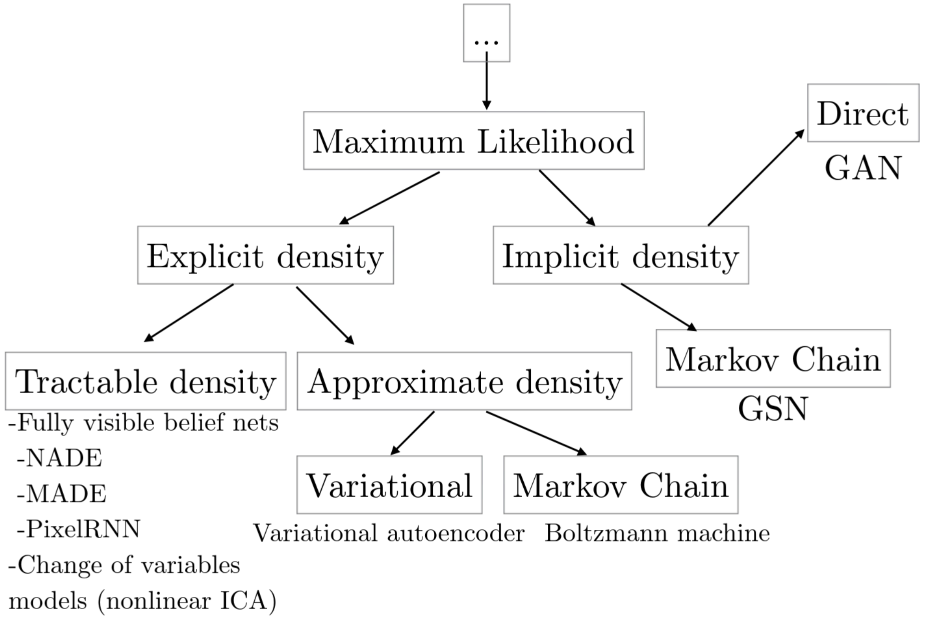
- https://github.com/hindupuravinash/the-gan-zoo



## BigGAN

- Large Scale GAN Training for High Fidelity Natural Image Synthesis, Brock et al., 2019
- Class conditional generation of images.

# Taxonomy of generative methods




- Image from Tutorial: Generative Adversarial Networks, Godfellow, NeurIPS 2016