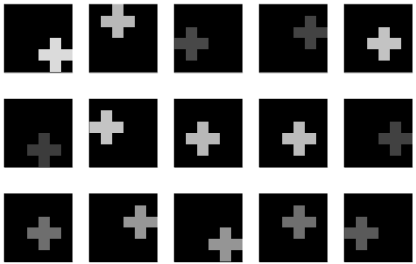
Autoencoders and Generative Adversarial Networks

Johanni Brea

Introduction to Machine Learning

Low-Dimensional Generators



Often we can describe a data generator as generating a low-dimensional random event Z with probability $P(Z)$ (e.g. choose position and gray level of the cross) followed by a complicated transformation f to produce the observation $X = f(Z)$ (e.g. the image with the cross) with a complicated distribution $P(X)$ over observations.

Goal 1 (**compression**): find a low-dimensional representation $\hat{Z}$ of X.

Goal 2 (**generation**): find $\hat{f}$ such that we can generate artificial examples, by sampling $\hat{Z}$ and producing an artificial observation $\hat{X} = \hat{f}(\hat{Z})$.

Table of Contents

Autoencoders

Variational Autoencoders (VAE)

Table of Contents

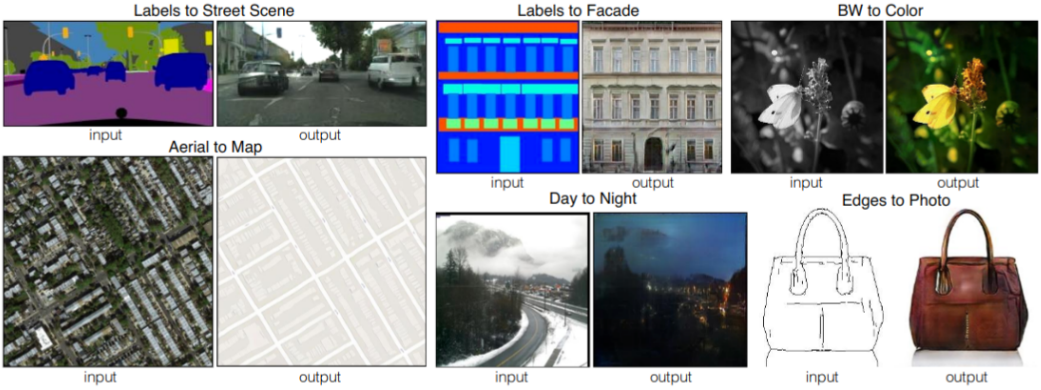
Recurrent Generators

$$P(X_1, X_2, \ldots, X_T) = P(X_1)P(X_2|X_1)P(X_3|X_1, X_2) \cdots P(X_T|X_1, \ldots, X_{T-1})$$

Table of Contents

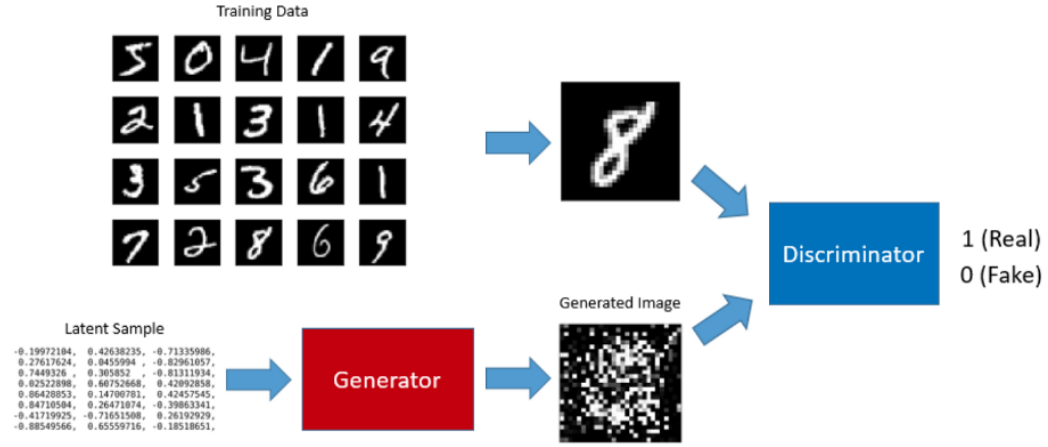
GAN Examples

GAN Examples



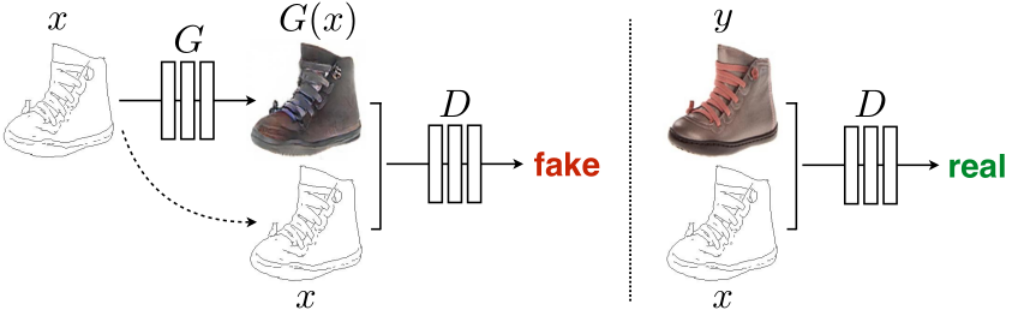
How GANs Work

How GANs Work



https://naokishibuya.medium.com/understanding-generative-adversarial-networks-4dafc963f2ef

How Conditional GANs Work



https://arxiv.org/abs/1611.07004

Example Code

GAN on MNIST

https://github.com/FluxML/model-zoo/blob/master/vision/dcgan_mnist

Conditional GAN on MNIST

https://github.com/FluxML/model-zoo/blob/master/vision/cdcgan_mnist