

TTIC 31230, Fundamentals of Deep Learning

David McAllester, Winter 2020

Assumptions on Latent Variables

Latent Variable Models

$$P_{\Phi}(y) = \sum_z P_{\Phi}(z)P_{\Phi}(y|z) = E_{z \sim P_{\Phi}(z)} P_{\Phi}(y|z)$$

Or

$$P_{\Phi}(y|x) = \sum_z P_{\Phi}(z|x)P_{\Phi}(y|z, x) = E_{z \sim P_{\Phi}(z|x)} P_{\Phi}(y|z, x)$$

Here z is a latent variable.

Latent Variable Models

$$P_{\Phi}(y) = \sum_z P_{\Phi}(z)P_{\Phi}(y|z) = E_{z \sim P_{\Phi}(z)} P_{\Phi}(y|z)$$

Here we often think of z as the causal source of y .

For example z might be a physical scene causing image y .

Or z might be the intended utterance causing speech signal y .

In these situations a latent variable model should more accurately represents the distribution on y .

Latent Variable Models

$$P_{\Phi}(y) = \sum_z P_{\Phi}(z)P_{\Phi}(y|z) = E_{z \sim P_{\Phi}(z)} P_{\Phi}(y|z)$$

$P_{\Phi}(z)$ is called the prior.

Given an observation of y (the evidence) $P_{\Phi}(z|y)$ is called the posterior.

Variational Bayesian inference involves approximating the posterior.

Colorization with Latent Segmentation



Input

Our Method

Ground-truth

x

\hat{y}

y

Larsson et al., 2016

Colorization is a natural self-supervised learning problem — we delete the color and then try to recover it from the grey-level image.

Can colorization be used to learn segmentation?

Segmentation is latent — not determined by the color label.

Colorization with Latent Segmentation



Input

Our Method

Ground-truth

x

\hat{y}

y

Larsson et al., 2016

x is a grey level image.

y is a color image drawn from $\text{Pop}(y|x)$.

\hat{y} is an arbitrary color image.

$P_{\Phi}(\hat{y}|x)$ is the probability that model Φ assigns to the color image \hat{y} given grey level image x .

Colorization with Latent Segmentation



Input

Our Method

Ground-truth

x

\hat{y}

y

$$P_{\Phi}(\hat{y}|x) = \sum_z P_{\Phi}(z|x) P_{\Phi}(\hat{y}|z, x).$$

input x

$P_{\Phi}(z|x) = \dots$ semantic segmentation

$P_{\Phi}(\hat{y}|z, x) = \dots$ segment colorization

Assumptions

We assume models $P_{\Phi}(z)$ and $P_{\Phi}(y|z)$ are both samplable and computable.

In other words, we can sample from these distributions and for any given z and y we can compute $P_{\Phi}(z)$ and $P_{\Phi}(y|z)$.

These are nontrivial assumptions.

A loopy graphical model is neither (efficiently) samplable nor computable.

Cases Where the Assumptions Hold

In CTC we have that z is the sequence with blanks and y is the result of removing the blanks from z .

In a hidden markov model z is the sequence of hidden states and y is the sequence of emissions.

An autoregressive model, such as an autoregressive language model, is both samplable and computable.

Image Generators

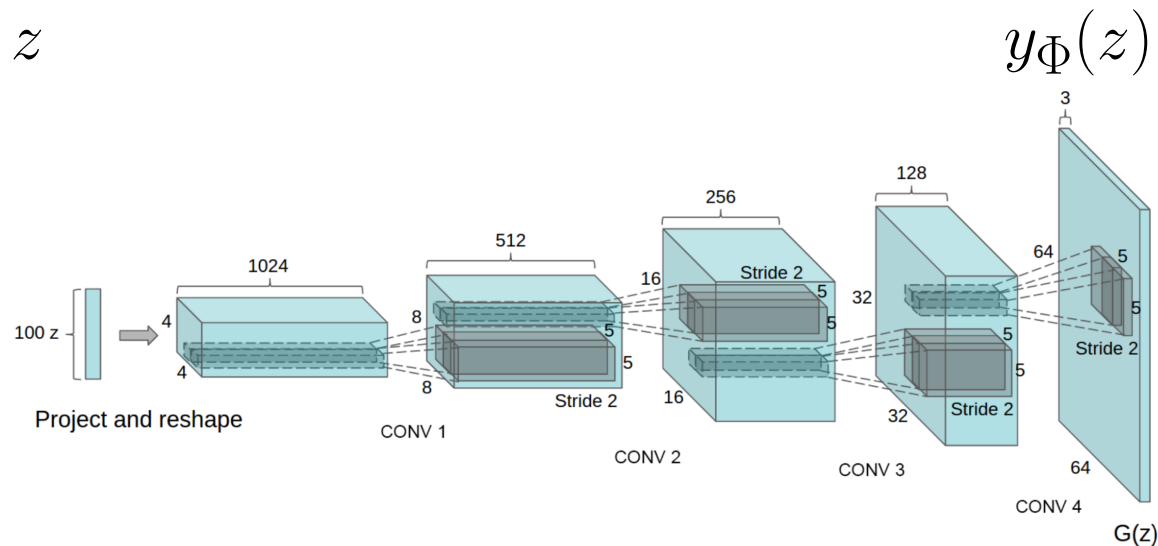


Figure 1: DCGAN generator used for LSUN scene modeling. A 100 dimensional uniform distribution Z is projected to a small spatial extent convolutional representation with many feature maps.

We can generate an image y from noise z where $p_{\Phi}(z)$ and $p_{\Phi}(y|z)$ are both samplable and computable.

Typically $p_{\Phi}(z)$ is $\mathcal{N}(0, I)$ reshaped as $z[X, Y, J]$

Assumptions

Even when $P_{\Phi}(z)$ and $P_{\Phi}(y|z)$ are samplable and computable we cannot typically compute $P_{\Phi}(y)$.

Specifically, for $P_{\Phi}(y)$ defined by a GAN generator we cannot compute $P_{\Phi}(y)$ for a test image y .

Hence it is not obvious how to optimize the fundamental equation.

$$\Phi^* = \operatorname{argmin}_{\Phi} E_{y \sim P_{\text{op}}} - \ln P_{\Phi}(y)$$

END