TTIC 31230, Fundamentals of Deep Learning

David McAllester, Autumn 2020

Rate-Distortion Autoencoders (RDAs)

Cross Entropy for Continuous Structured y

Cross entropy is a challenging objective for continuous structured values y such as images and sounds.

$$\Phi^* = \operatorname*{argmin}_{\Phi} \; E_{y \sim \mathrm{pop}} \; - \ln p_\Phi(y)$$

$$\Phi^* = \operatorname*{argmin}_{\Phi} \; E_{\langle x, y \rangle \sim \mathrm{pop}} \; - \ln p_\Phi(y|x)$$

GANs replace the differential cross-entropy loss with an adversarial discrimination loss.

Rate-Distortion Auto-Encoders (RDAs) and Variational Auto-Encoders (VAEs) use the differential cross-entropy loss more directly.

Rate-Distortion Autoencoders (RDAs)

A rate-distortion autoencoder (RDA) replaces differential cross-entropy loss with a compression rate and a reconstruction loss (distortion).

The primary example is lossy compression of images and audio.

We first consider the case where the compressed object is discrete — a file with a well defined length in bits.

Rate-Distortion Autoencoders (RDAs)

We compress a continuous signal y to a bit string (or other discrete object) $\tilde{z}_\Phi(y)$.

We decompress $\tilde{z}_\Phi(y)$ to $y_\Phi(\tilde{z}_\Phi(y))$.

We can then define a rate-distortion loss.

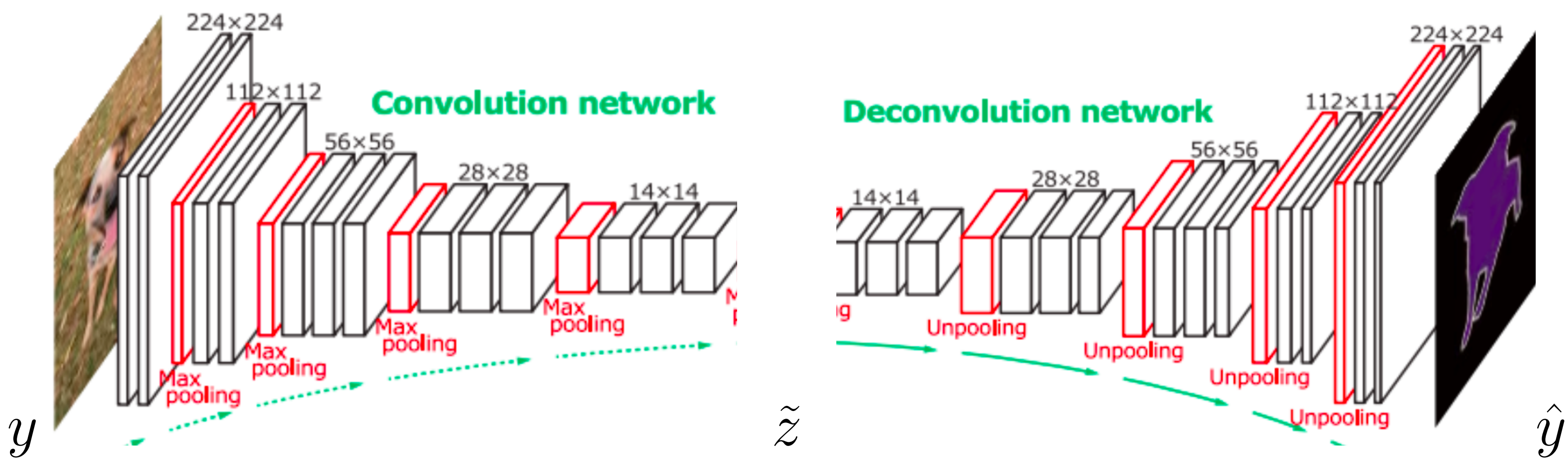
$$\mathcal{L}(\Phi) = E_{y\sim\mathrm{Pop}}\ -\ln P_\Phi(\tilde{z}_\Phi(y)) + \lambda \mathrm{Dist}(y, y_\Phi(\tilde{z}_\Phi(y)))$$

Here the rate is defined as a discrete cross-entropy.

CNN-based Image Compression

These slides are loosely based on

End-to-End Optimized Image Compression, Balle, Laparra, Simoncelli, ICLR 2017.



Rounding a Tensor

Take $z_\Phi(y)$ can be a layer in a CNN applied to image y. $z_\Phi(y)$ can have with both spatial and feature dimensions.

Take $\tilde{z}_\Phi(y)$ to be the result of rounding each component of the continuous tensor $z_\Phi(y)$ to the nearest integer.

$$\tilde{z}_\Phi(y)[x, y, i] = \lfloor z_\Phi(y)[x, y, i] + 1/2 \rfloor$$

Rate-Distortion Autoencoders (RDAs)

Since rounding is not differentiable, at training time we replace rounding by additive noise.

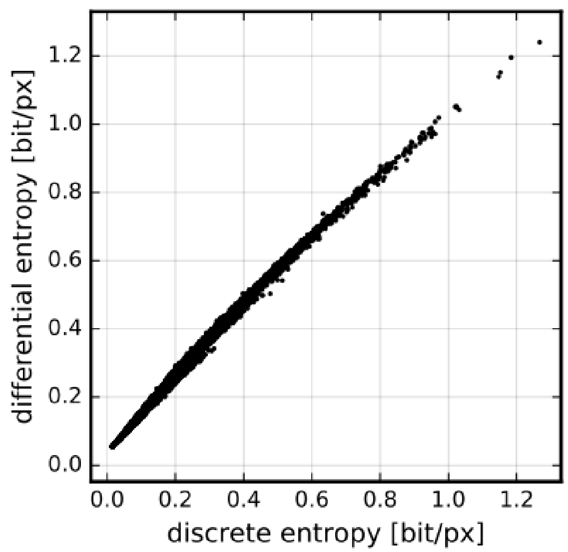
$$\Phi^* = \underset{\Phi}{\operatorname{argmin}}\; E_{y\sim\mathrm{Train}}\; E_{\epsilon\sim[-1/2,1/2]^d} \left\{ \begin{array}{l} \quad -\ln p_\Phi(z_\Phi(y)+\epsilon) \\ \\ + \;\; \lambda \mathrm{Dist}(y, y_\Phi(z_\Phi(y)+\epsilon)) \end{array} \right.$$

The continuous density $p_\Phi(z)$ is parameterized in a way that guarantees

$$p_\Phi(z) \approx P_\Phi(\tilde{z})$$

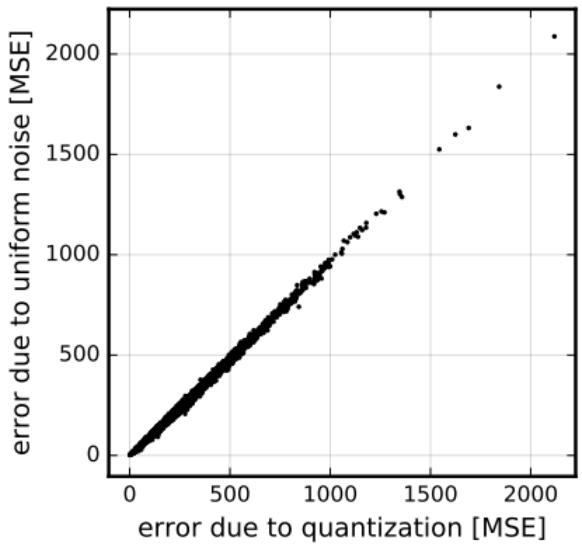
At test time we use rounding.

Rate: Differential Entropy vs. Discrete Entropy



Each point is a rate for an image measured in both differential entropy and discrete entropy. The size of the rate changes as we change the weight λ.

Distortion: Noise vs. Rounding



Each point is a distortion for an image measured in both a rounding model and a noise model. The size of the distortion changes as we change the weight λ.

JPEG at 4283 bytes or .121 bits per pixel

JPEG, 4283 bytes (0.121 bit/px), PSNR: 24.85 dB/29.23 dB, MS-SSIM: 0.8079

JPEG 2000 at 4004 bytes or .113 bits per pixel

JPEG 2000, 4004 bytes (0.113 bit/px), PSNR: 26.61 dB/33.88 dB, MS-SSIM: 0.8860

Deep Autoencoder at 3986 bytes or .113 bits per pixel

Proposed method, 3986 bytes (0.113 bit/px), PSNR: 27.01 dB/34.16 dB, MS-SSIM: 0.9039

END