

# **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.

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

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

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

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

## Rate-Distortion Autoencoders (RDAs)

A rate-distortion autoencoder (RDA) replaces differential cross-entropy with a bi-objective — a compression rate and the reconstruction distortion.

The primary example is lossy compression of images and audio.

We take the compressed object to be discrete — a file with a well defined length in bits.

## Rate-Distortion Autoencoders (RDAs)

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

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

We can then define a rate-distortion loss.

$$\mathcal{L}(\Phi) = E_{y \sim P_{\text{op}}} |\tilde{z}_\Phi(y)| + \lambda \text{Dist}(y, y_\Phi(\tilde{z}_\Phi(y)))$$

where  $|\tilde{z}|$  is the number of bits in the bit string  $\tilde{z}$ .

## $L_2$ Distortion

$$\Phi^* = \operatorname{argmin}_{\Phi} E_{y \sim \text{Pop}} |\tilde{z}_{\Phi}(y)| + \lambda \text{Dist}(y, y_{\Phi}(\tilde{z}_{\Phi}(y)))$$

It is common to take

$$\begin{aligned} \text{Dist}(y, \hat{y}) &= ||y - \hat{y}||^2 \\ &= -\ln p(y|\hat{y}) + C \quad \text{for a Gaussian density} \end{aligned}$$

We will ignore the log probability interpretation and just call this distortion.

## $L_1$ Distortion

$$\Phi^* = \underset{\Phi}{\operatorname{argmin}} \ E_{y \sim \text{Pop}} |\tilde{z}_{\Phi}(y)| + \lambda \text{Dist}(y, y_{\Phi}(\tilde{z}_{\Phi}(y)))$$

Alternatively we have

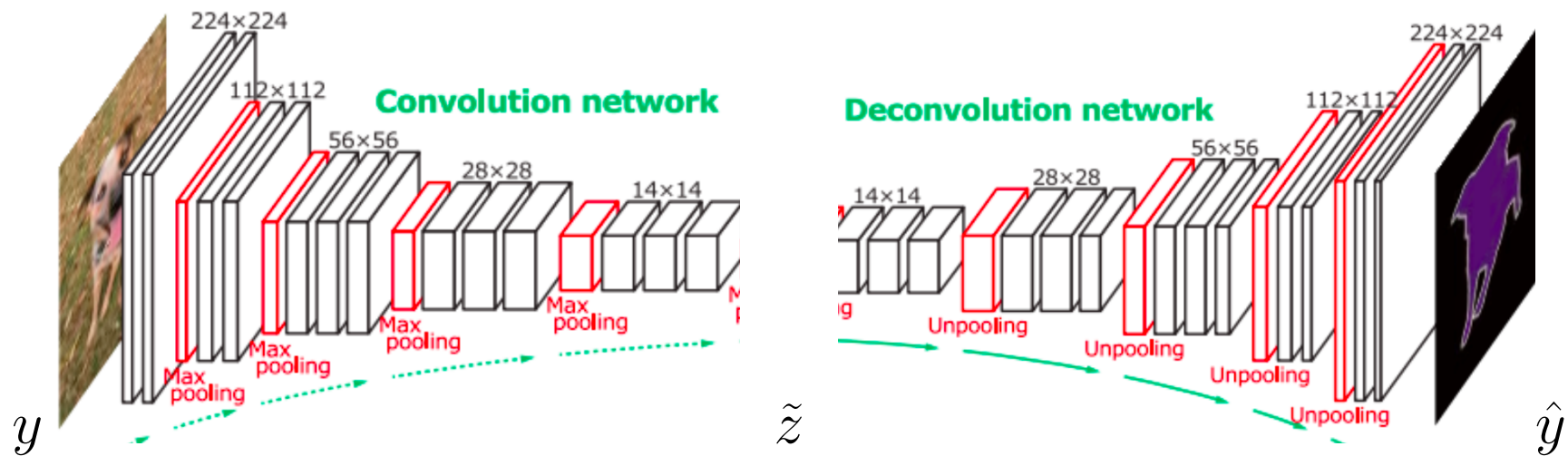
$$\begin{aligned} \text{Dist}(y, \hat{y}) &= ||y - \hat{y}||_1 && (L_2) \\ &= -\ln p(y|\hat{y}) + C \text{ for a Laplace density} \end{aligned}$$

Again, we will ignore the log probability interpretation and just call this distortion.

# 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 we train by replace rounding by additive noise.

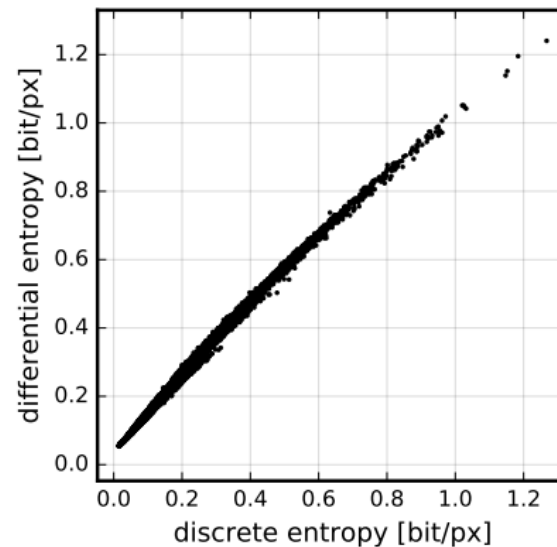
$$\mathcal{L}(\Phi) = E_{y \sim \text{Pop}} E_{\epsilon \sim [-1/2, 1/2]^d} \left\{ \begin{array}{l} -\ln p_{\Phi}(z_{\Phi}(y) + \epsilon) \\ + \lambda \text{Dist}(y, y_{\Phi}(z_{\Phi}(y) + \epsilon)) \end{array} \right.$$

The continuous density is constrained to be sufficiently smooth so that

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

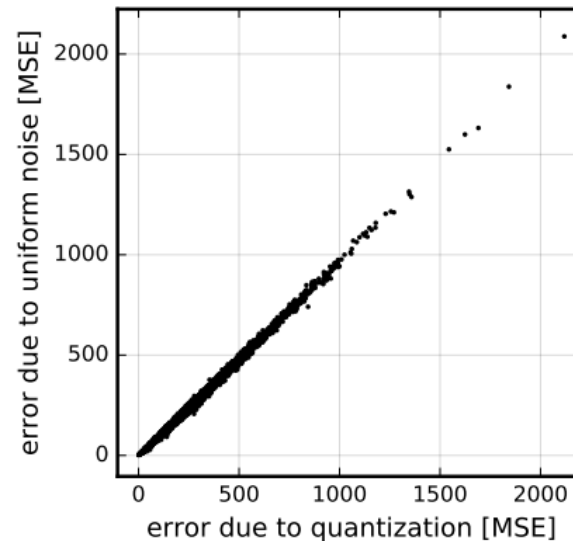
At test time (for actual compression) we use rounding.

## 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  $\lambda$ .

## 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  $\lambda$ .

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**