

# **TTIC 31230, Fundamentals of Deep Learning**

David McAllester, Winter 2019

Rate-Distortion Autoencoders (RDAs)

Noisy Channel RDAs

Gaussian Variational Autoencoders (Gaussian VAEs)

## Rate-Distortion Autoencoders

Given a continuous signal  $y$  we can compress it into a (discrete) bit string  $\tilde{z}_\Phi(y)$ .

We let  $y_\Phi(\tilde{z}_\Phi(y))$  be the decompression of  $\tilde{z}_\Phi(y)$ .

We can then define a rate-distortion loss.

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

## Common Distortion Functions

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

It is common to take

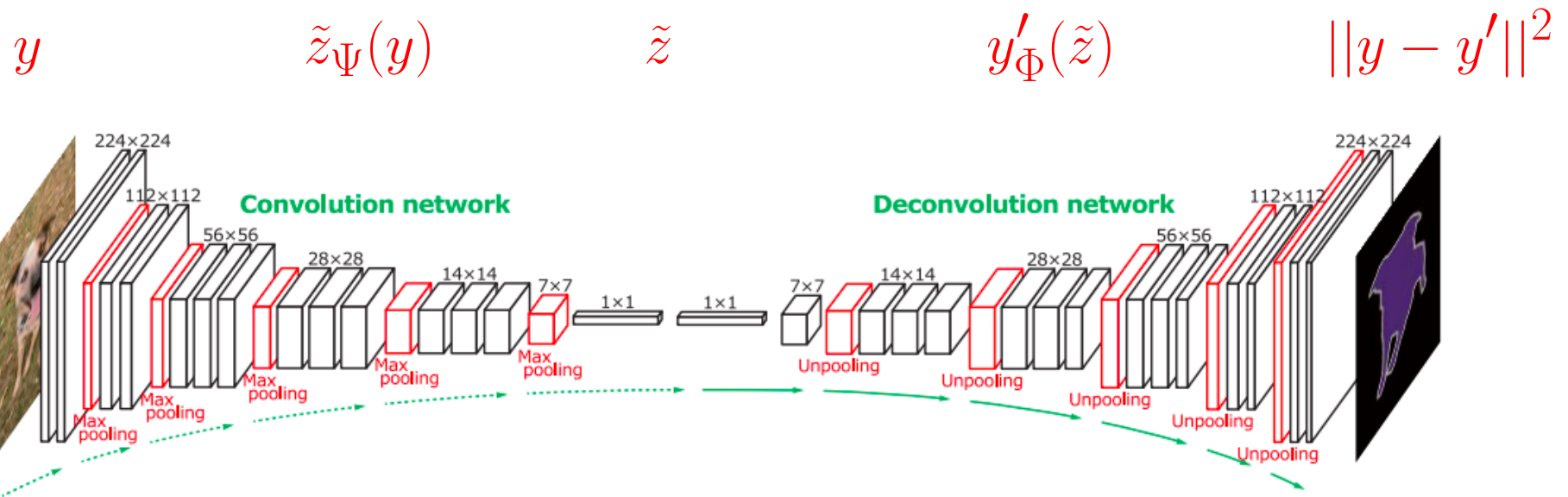
$$\text{Dist}(y, y') = \|y - y'\|^2 \quad (L_2)$$

or

$$\text{Dist}(y, y') = \|y - y'\|_1 \quad (L_1)$$

# A Case Study in Image Compression

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



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**

## A CNN Encoder

A three layer CNN is used as an encoder.

We let  $z_{\Phi}(y)$  be the final layer of this CNN.

Each continuous value in the final layer  $z_{\Phi}(y)$  is then rounded to a (small) integer giving a discrete encoding  $\tilde{z}(y)$ .



## Rate-Distortion Autoencoders

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

Oops: Because of rounding,  $\tilde{z}_{\Phi}(y)$  is discrete and the gradients are zero.

We will train using a differentiable approximation.

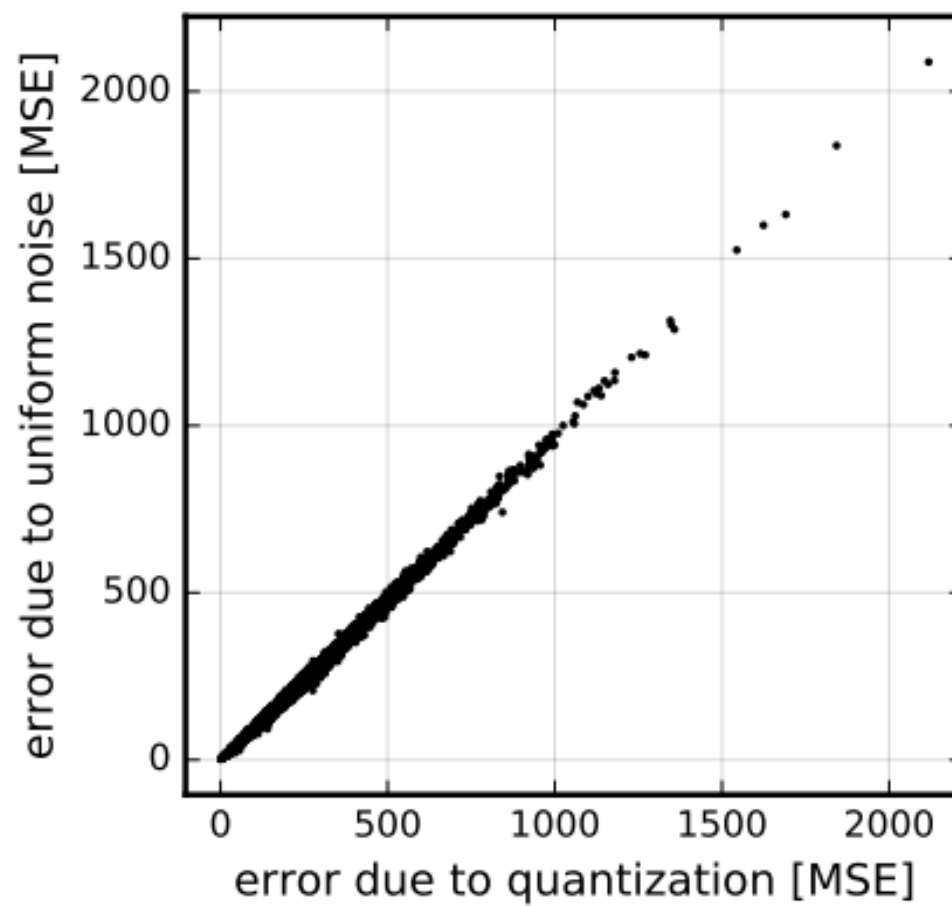
## A Noise Model of Rounding

We can make distortion differentiable by modeling rounding as the addition of noise.

$$\begin{aligned}\mathcal{L}_{\text{dist}}(\Phi) &= E_{y \sim \text{Pop}} \text{Dist}(y, y_{\Phi}(\tilde{z}_{\Phi}(y))) \\ &\approx E_{y, \epsilon} \text{Dist}(y, y_{\Phi}(z_{\Phi}(y) + \epsilon))\end{aligned}$$

Here  $\epsilon$  is a noise vector each component of which is drawn uniformly from  $(-1/2, 1/2)$ .

## Noise vs. Rounding



## A Differentiable Approximation of Code Length

$$\mathcal{L}_{\text{rate}}(\Phi) = E_{y \sim P_{\text{op}}} |\tilde{z}_{\Phi}(y)|$$

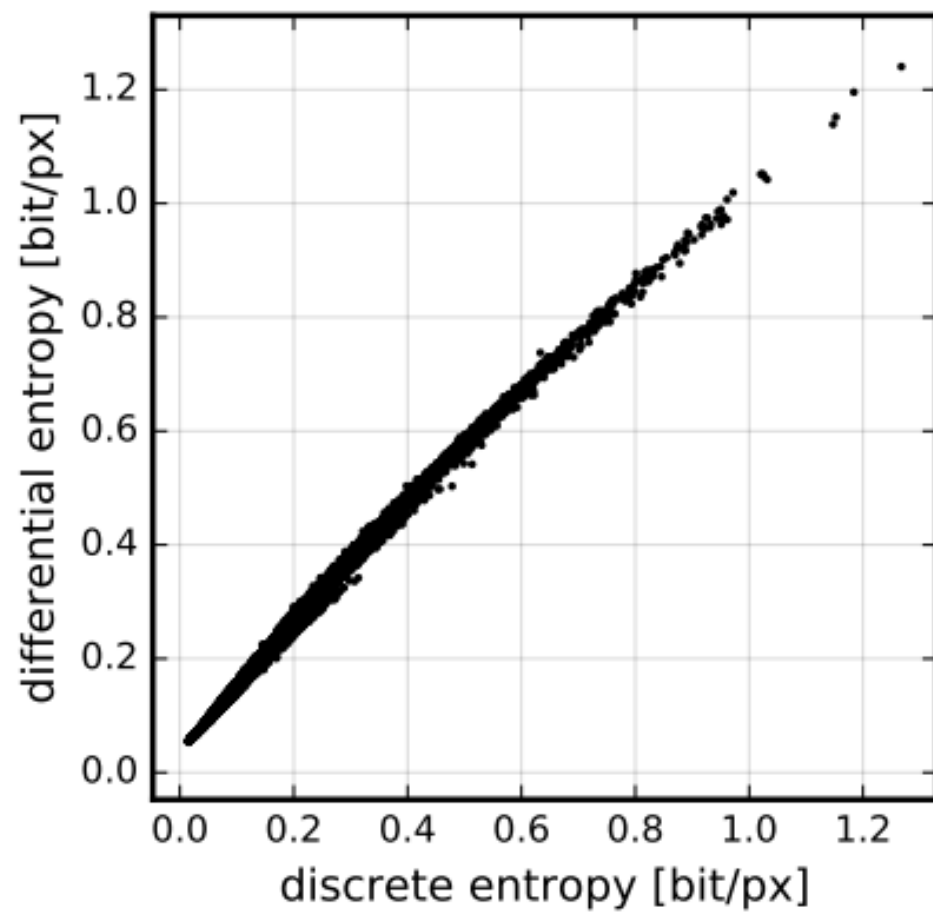
Recall that  $\tilde{z}_{\Phi}(y)$  is a rounding of a continuous encoding  $z_{\Phi}(y)$ .

We approximate the code length after rounding using a differentiable function of the value before rounding.

$$|\tilde{z}_{\Phi}(y)| \approx \sum_i (\log_2 z_{\Phi}(y)[i])^+$$

This continuous value can be interpreted as a “differential entropy”.

## Differential Entropy vs. Discrete Entropy



## Details

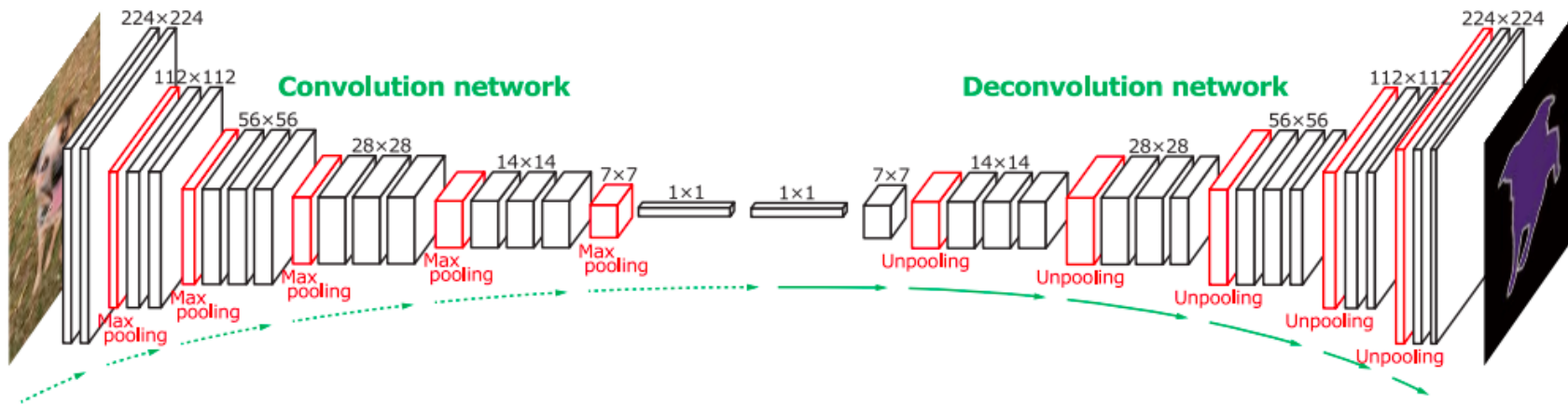
The first layer is computed stride 4.

The next two layers are computed stride 2.

Final image dimension is reduced by a factor of 16 with 192 channels per pixel (192 channels is for color images).

$$192 < 16 \times 16 \times 3 = 768$$

# Increasing Spatial Dimension in Decoding



[Hyeonwoo Noh et al.]

In the ICLR 17 paper the deconvolution network has the shape as the input CNN but with independent parameters.

## Increasing Spatial Dimension in Decoding

Consider a stride 2 convolution

$$L_{\ell+1}[x, y, j] = \sigma \left( \sum_{\Delta x, \Delta y, i} W[\Delta x, \Delta y, i, j] L_{\ell}[2x + \Delta x, 2y + \Delta y, i] \right)$$

For deconvolution we use stride 1 with 4 times the features.

$$L'_{\ell}[x, y, i] = \sigma \left( \sum_{\Delta x, \Delta y, j} W[\Delta x, \Delta y, j, i] L'_{\ell+1}[x + \Delta x, y + \Delta y, j] \right)$$

The channels at each  $L'_{\ell}[x, y]$  are divided among four higher resolution pixels.

This is done by a simple reshaping of  $L'_{\ell}[x, y, i]$ .



## Noisy-Channel RDAs (TZ)

Consider the differentiable loss used in training.

$$\Phi^* = \operatorname{argmin}_{\Phi} E_{y \sim P_{\text{op}}} -\ln p(z_{\Phi}(y)) + \lambda E_{\epsilon} \operatorname{Dist}(y, y_{\Phi}(z_{\Phi}(y) + \epsilon))$$

Intuitively,  $-\ln p(z_{\Phi}(y))$  is a proxy for the number of bits used in the (intuitively rounded) encoding  $z_{\Phi}(y) + \epsilon$ .

By the channel capacity theorem the number of bits that  $z_{\Phi}(y) + \epsilon$  can carry about  $y$  is the mutual information between  $y$  and  $z_{\Phi}(y) + \epsilon$ .

$$I(y, z_{\Phi}(y) + \epsilon)$$

## Noisy-Channel RDAs

We now consider  $p_{\Phi}(z|y)$  as a generalization of  $z_{\Phi}(y) + \epsilon$ .

The channel capacity theorem motivates

$$p_{\Phi}(z) = E_{y \sim P_{\text{Op}}} p_{\Phi}(z|y)$$
$$I(y, z) = H(z) - H(z|y) = E_{y \sim P_{\text{Op}}} KL(p_{\Phi}(z|y), p_{\Phi}(z))$$

$$\begin{aligned} \Phi^* &= \underset{\Phi}{\operatorname{argmin}} \left( I(y, z) + \lambda E_{y \sim P_{\text{Op}}, z \sim p_{\Phi}(z|y)} \operatorname{Dist}(y, y_{\Phi}(z)) \right) \\ &= \underset{\Phi}{\operatorname{argmin}} E_{y \sim P_{\text{Op}}} \left( \begin{aligned} &KL(p_{\Phi}(z|y), p_{\Phi}(z)) \\ &+ \lambda E_{z \sim p_{\Phi}(z|y)} \operatorname{Dist}(y, y_{\Phi}(z)) \end{aligned} \right) \end{aligned}$$

## A Variational Upper Bound

Unfortunately we cannot compute  $p_{\Phi}(z) = E_{y \sim P_{\text{op}}} p_{\Phi}(z|y)$ .

We now replace  $p_{\Phi}(z)$  by a friendly (variational) model  $p_{\Psi}(z)$ .

$$\begin{aligned} I_{\Phi}(y, z) &= E_{y \sim P_{\text{op}}} KL(p_{\Phi}(z|y), p_{\Phi}(z)) \\ &= E_{y, z \sim P_{\Phi}(z|y)} \left( \ln \frac{p_{\Phi}(z|y)}{p_{\Psi}(z)} + \ln \frac{p_{\Psi}(z)}{p_{\Phi}(z)} \right) \\ &= E_{y \sim P_{\text{op}}} KL(p_{\Phi}(z|y), p_{\Psi}(z)) + \left( E_{y \sim P_{\text{op}}, z \sim p_{\Phi}(z|y)} \ln \frac{p_{\Psi}(z)}{p_{\Phi}(z)} \right) \end{aligned}$$

## A Variational Upper Bound

$$I(y, z)$$

$$= E_{y \sim \text{Pop}} KL(p_{\Phi}(z|y), p_{\Psi}(z)) + \left( E_{y \sim \text{Pop}, z \sim p_{\Phi}(z|y)} \ln \frac{p_{\Psi}(z)}{p_{\Phi}(z)} \right)$$

$$= E_y KL(p_{\Phi}(z|y), p_{\Psi}(z)) + E_{z \sim p_{\Phi}(z)} \ln \frac{p_{\Psi}(z)}{p_{\Phi}(z)}$$

$$= E_y KL(p_{\Phi}(z|y), p_{\Psi}(z)) - KL(p_{\Phi}(z), p_{\Psi}(z))$$

$$\leq E_{y \sim \text{Pop}} KL(p_{\Phi}(z|y), p_{\Psi}(z))$$

## The Noisy-Channel RDA

$$\Phi^*, \Psi^* = \operatorname{argmin}_{\Phi, \Psi} E_{y \sim \text{Pop}} \left( \begin{array}{l} KL(p_{\Phi}(z|y), p_{\Psi}(z)) \\ + \lambda E_{z \sim p_{\Phi}(z|y)} \text{Dist}(y, y_{\Phi}(z)) \end{array} \right)$$

## Gaussian Noisy-Channel RDA

$$\Phi^*, \Psi^* = \operatorname{argmin}_{\Phi, \Psi} E_{y \sim \text{Pop}} \left( \begin{array}{c} KL(p_{\Phi}(z|y), p_{\Psi}(z)) \\ + \lambda E_{z \sim p_{\Phi}(z|y)} \text{Dist}(y, y_{\Phi}(z)) \end{array} \right)$$

$$p_{\Phi}(z[i] \mid y) = \mathcal{N}(z_{\Phi}(y)[i], \sigma_{\Phi}(y)[i])$$

$$p_{\Psi}(z[i]) = \mathcal{N}(\mu_{\Psi}[i], \sigma_{\Psi}[i])$$

$$\text{Dist}(y, y') = ||y - y'||^2$$

## Closed Form KL-Divergence

$$KL(p_{\Phi}(z|y), p_{\Psi}(z))$$
$$= \sum_i \frac{\sigma_{\Phi}(y)[i]^2 + (z_{\Phi}(y)[i] - \mu_{\Psi}[i])^2}{2\sigma_{\Psi}[i]^2} + \ln \frac{\sigma_{\Psi}[i]}{\sigma_{\Phi}(y)[i]} - \frac{1}{2}$$

## Standardizing $p_\Psi(z)$

The KL-divergence term is

$$\sum_i \frac{\sigma_\Phi(y)[i]^2 + (z_\Phi(y)[i] - \mu_\Psi[i])^2}{2\sigma_\Psi[i]^2} + \ln \frac{\sigma_\Psi[i]}{\sigma_\Phi(y)[i]} - \frac{1}{2}$$

We can adjust  $\Phi$  to  $\Phi'$  such that

$$\begin{aligned} z_{\Phi'}(y)[i] &= (z_\Phi(y)[i] - \mu_\Psi[i]) / \sigma_\Psi[i] \\ \sigma_{\Phi'}(y)[i] &= \sigma_\Phi(y)[i] / \sigma_\Psi[i] \end{aligned}$$

We then get  $KL(p_{\Phi'}(z|y), \mathcal{N}(0, I)) = KL(p_\Phi(z|y), p_\Psi(z))$ .



## Standardizing $p_{\Psi}(z)$

Without loss of generality the Gaussian noisy channel RDA becomes.

$$\Phi^* = \underset{\Phi}{\operatorname{argmin}} E_{y \sim \text{Pop}} \left( \begin{array}{l} KL(p_{\Phi}(z|y), \mathcal{N}(0, I)) \\ + \lambda E_{z \sim p_{\Phi}(z|y)} \text{Dist}(y, y_{\Phi}(z)) \end{array} \right)$$

## Reparameterization Trick for Optimizing Distortion

$$p_{\Phi}(z[i]|y) = \mathcal{N}(z_{\Phi}(y)[i], \sigma_{\Phi}[i])$$

$$E_{z \sim p_{\Phi}(z|y)} ||y - y_{\Phi}(z)||^2$$

$$= E_{\epsilon \sim \mathcal{N}(0, I)} z[i] = z_{\Phi}(y)[i] + \sigma_{\Phi}(y)[i]\epsilon[i]; \quad ||y - y_{\Phi}(z)||^2$$

## Sampling

Sample  $z \sim \mathcal{N}(0, I)$  and compute  $y_\Phi(z)$



[Alec Radford]

## Summary: Rate-Distortion

RDA:  $y$  continuous,  $\tilde{z}$  a bit string,

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

Gaussian RDA:  $z = z_{\Phi}(y) + \sigma_{\Phi}(y) \odot \epsilon$ ,  $\epsilon \sim \mathcal{N}(0, I)$

$$\Phi^* = \operatorname{argmin}_{\Phi} E_{y \sim \text{Pop}} \left( \begin{array}{l} KL(p_{\Phi}(z|y), \mathcal{N}(0, I)) \\ + \lambda E_{z \sim p_{\Phi}(z|y)} \text{Dist}(y, y_{\Phi}(z)) \end{array} \right)$$

Issue: Do we expect compression to yield useful features?

**END**