Rate-Distortion Autoencoders (RDAs)
The Fundamental Equation for Continuous $y$

If $y$ is continuous then the fundamental equation for estimating the distribution on $y$ (cross entropy) involves continuous probability densities.

$$\Phi^* = \arg\min_{\Phi} E_{y \sim \text{pop}} - \ln p_{\Phi}(y)$$

This occurs in unsupervised pretraining for sounds and images.

But differential entropy and differential cross-entropy are conceptually problematic.
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.

A compressed image does not have all the information of the original and the reconstructed image is a “distorted” version of the original.

The rate is given by the size of the compressed image (in bits or bytes).
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.

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

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

\[ \Phi^* = \arg\min_\Phi E_{y \sim \text{Pop}} |\tilde{z}_\Phi(y)| + \lambda \text{Dist}(y, y_\Phi(\tilde{z}_\Phi(y))) \]

It is common to take

\[ \text{Dist}(y, \hat{y}) = ||y - \hat{y}||^2 \quad (L_2) \]

or

\[ \text{Dist}(y, \hat{y}) = ||y - \hat{y}||_1 \quad (L_1) \]
CNN-based Image Compression

These slides are loosely based on

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.

$$\tilde{z}_\Phi(y)[x, y, i] = \lfloor z_\Phi(y)[x, y, i] + 1/2 \rfloor$$
Rounding is not Differentiable

\[ \Phi^* = \arg\min_{\Phi} E_{y \sim \text{Pop}} |\tilde{z}_\Phi(y)| + \lambda \text{Dist}(y, y_\Phi(\tilde{z}_\Phi(y))) \]

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

We will train using a differentiable approximation.
Rate: Replacing Code Length with Differential Entropy

\[ \mathcal{L}_{\text{rate}}(\Phi) = E_{y \sim \text{Pop}} |\tilde{z}_\Phi(y)| \]

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

Any probability distribution on integers can be approximated by a continuous density \( p_\Phi \) on the reals. For example we can take \( p_\Phi \) to be continuous and piecewise linear so that the rate becomes differentiable.

\[ |\tilde{z}_\Phi(y)| \approx \sum_{x,y,i} - \ln p_\Phi(z_\Phi(y)[x,y,i]) \]
Distortion: Replacing Rounding with Noise

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

\[ L_{\text{dist}}(\Phi) = E_{y \sim \text{Pop}} \text{Dist}(y, y_{\Phi}(\tilde{z}_{\Phi}(y))) \]

\[ \approx E_{y, \epsilon} \text{Dist}(y, y_{\Phi}(z_{\Phi}(y) + \epsilon)) \]

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

\[ \Phi^* = \arg\min_{\Phi} E_{y \sim \text{pop}} - \ln P_{\Phi}(z_{\Phi}(y)) + \lambda \text{Dist}(y, y_{\Phi}(z_{\Phi}(y))) \]

\( z_{\Phi}(y) \) discrete.
END