TTIC 31230, Fundamentals of Deep Learning

David McAllester, Winter 2020

Rate-Distortion Autoencoders (RDAs)

Rate-Distortion Autoencoders (Image Compression)

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.

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

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

Common Distortion Functions

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

It is common to take

$$\mathrm{Dist}(y, \hat{y}) = ||y - \hat{y}||^2 \qquad (L_2)$$

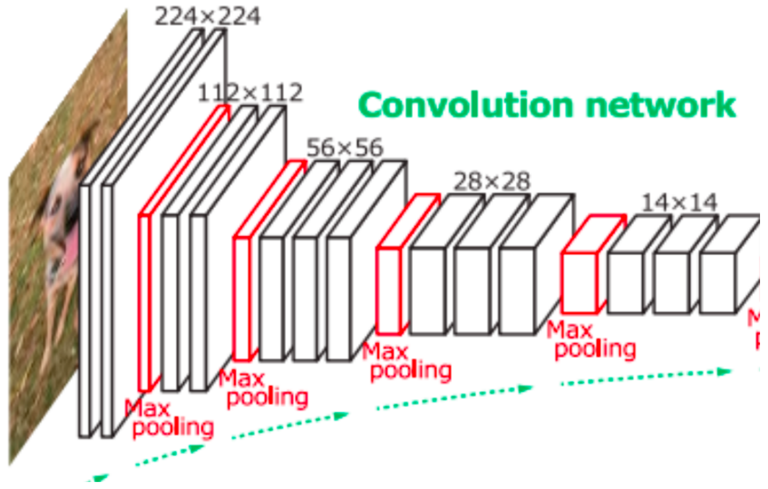
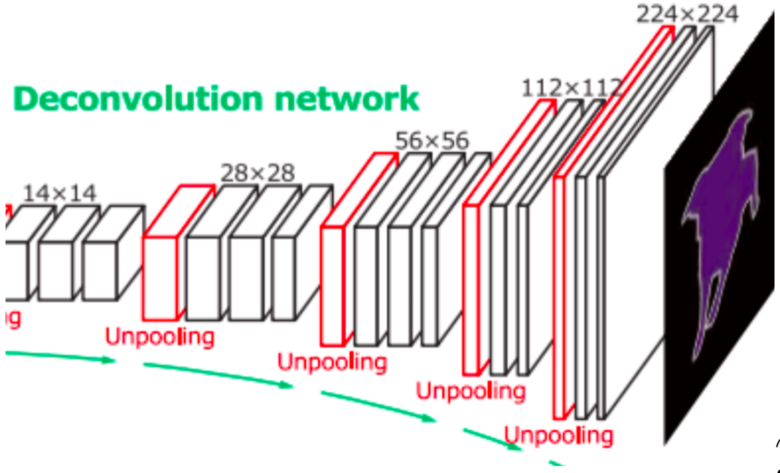
or

$$\mathrm{Dist}(y, \hat{y}) = ||y - \hat{y}||_1 \qquad (L_1)$$

CNN-based Image Compression

These slides are loosely based on

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

y  $\tilde{z}$  $\hat{y}$

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

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

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_Φ on the reals. For example we can take p_Φ to be continuous and piecewise linear so that the rate becomes differentiable.

$$|\tilde{z}_\Phi(y)| \approx \sum_{x,y,i} -\ln p_\Phi(z_\Phi(y)[x,y,i])$$

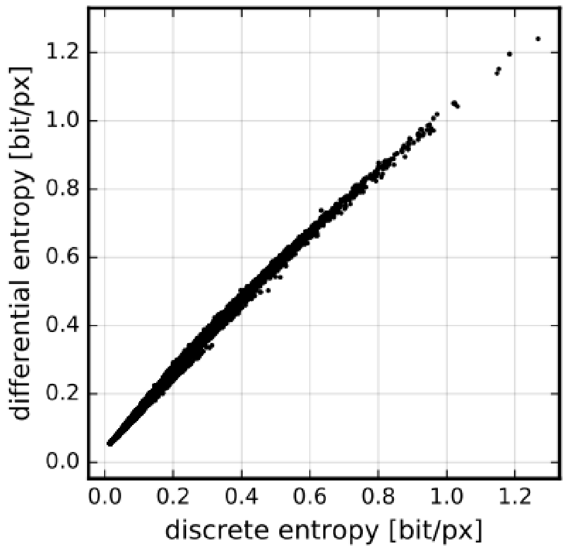
Distortion: Replacing Rounding with Noise

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

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

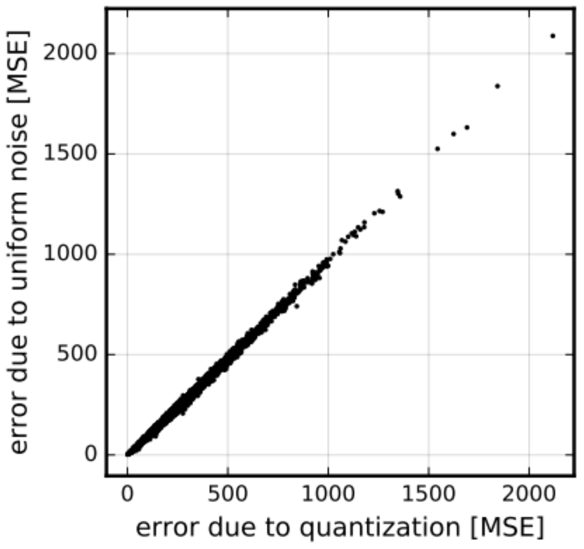
Here ϵ is a noise vector each component of which is drawn uniformly from $(-1/2, 1/2)$.

Rate: Differential Entropy vs. Discrete Entropy



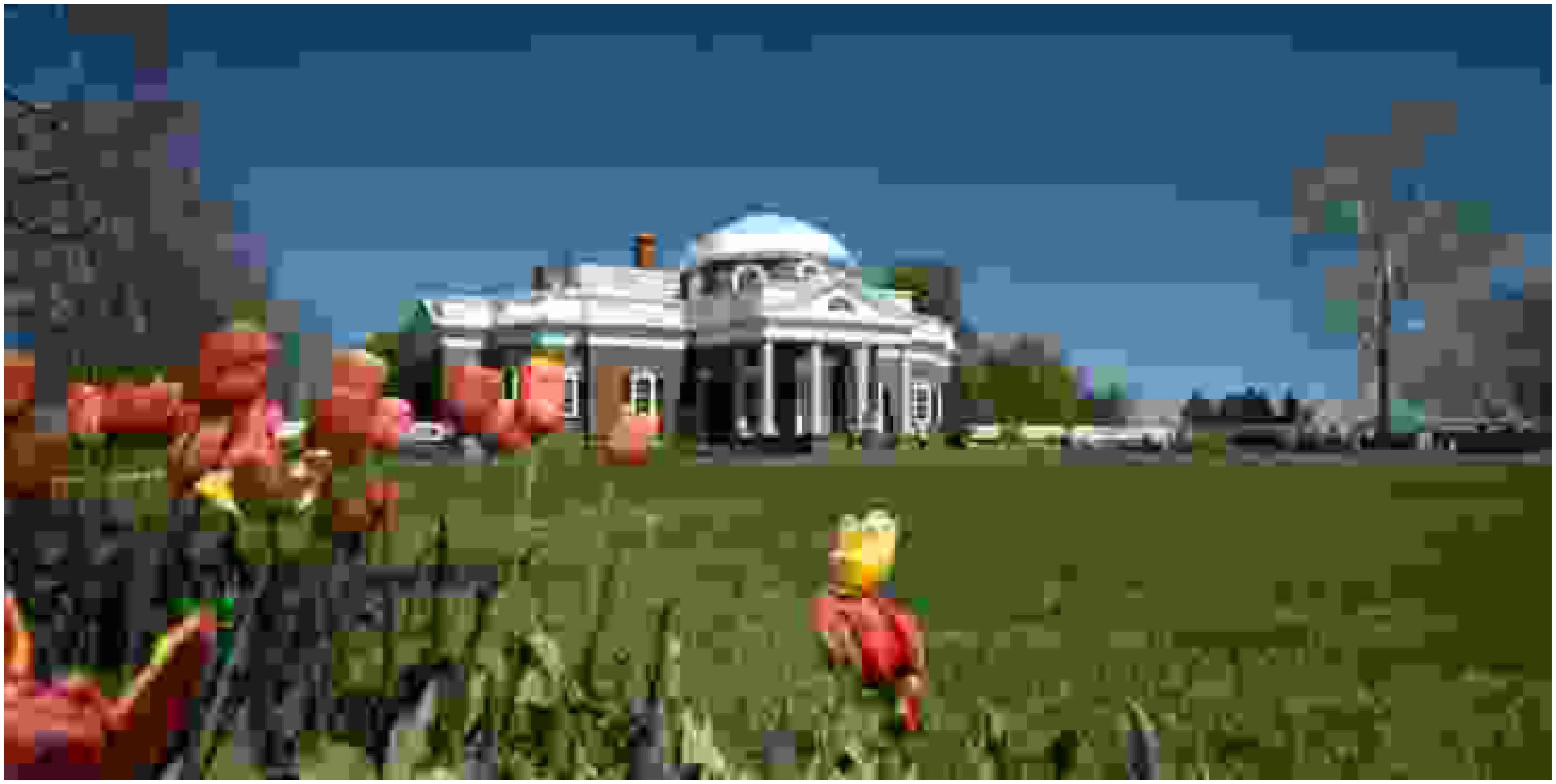
Each point is a rate for an image measured in both differential entropy and discrete entropy. The size of the rate changes as we change the weight λ.

Distortion: Noise vs. Rounding

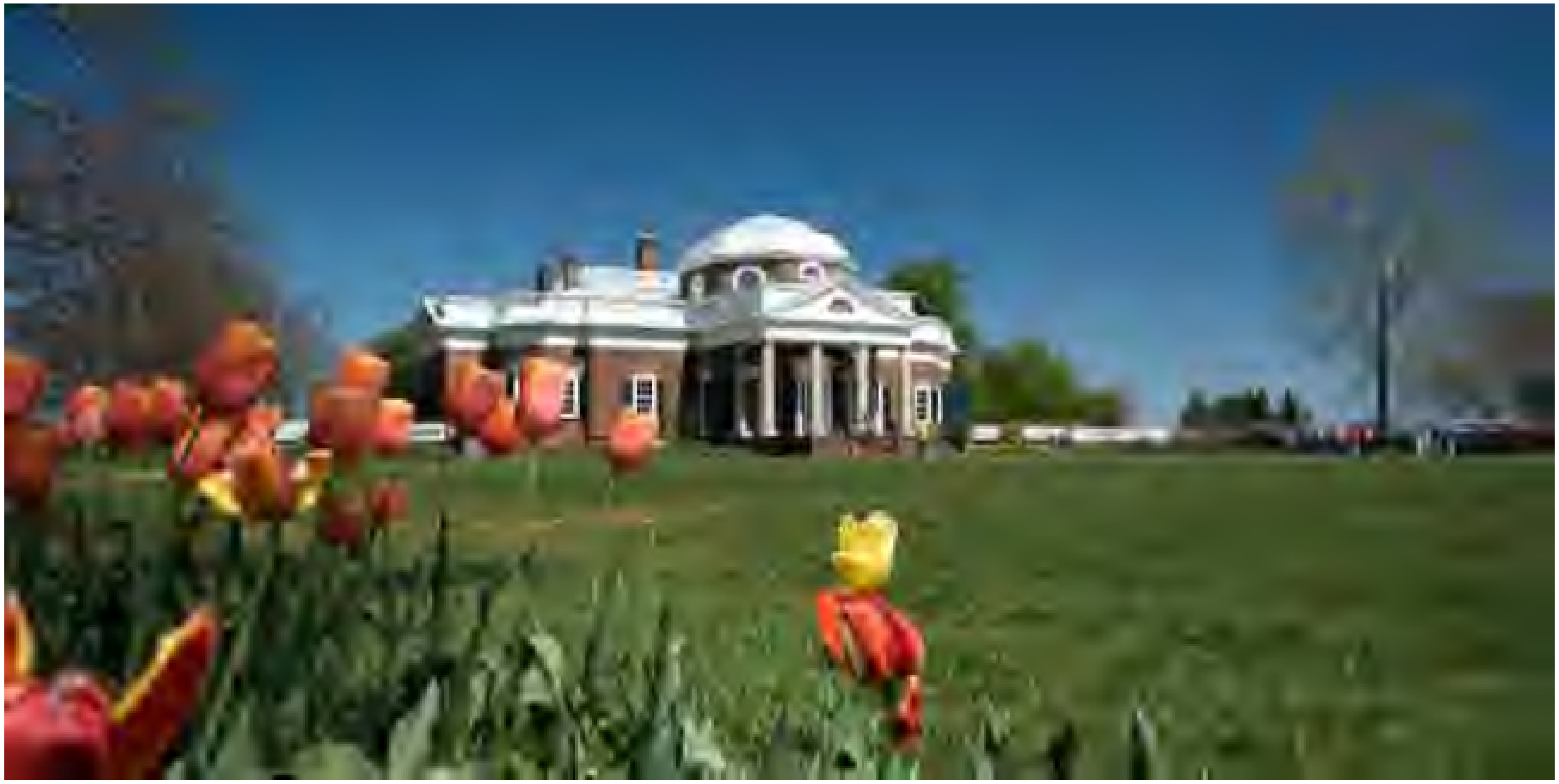


Each point is a distortion for an image measured in both a rounding model and a noise model. The size of the distortion changes as we change the weight λ.

JPEG at 4283 bytes or .121 bits per pixel

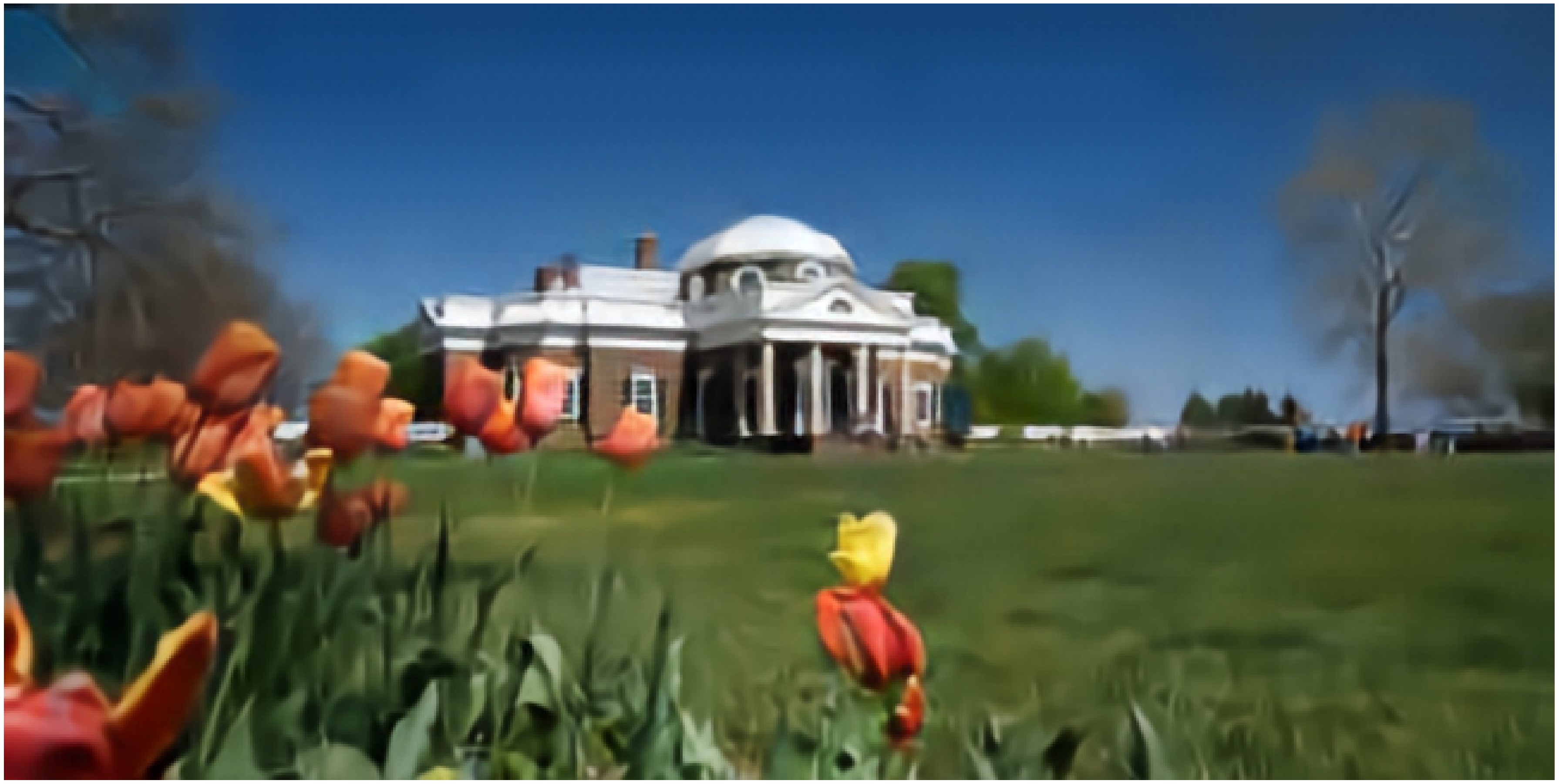

JPEG, 4283 bytes (0.121 bit/px), PSNR: 24.85 dB/29.23 dB, MS-SSIM: 0.8079

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