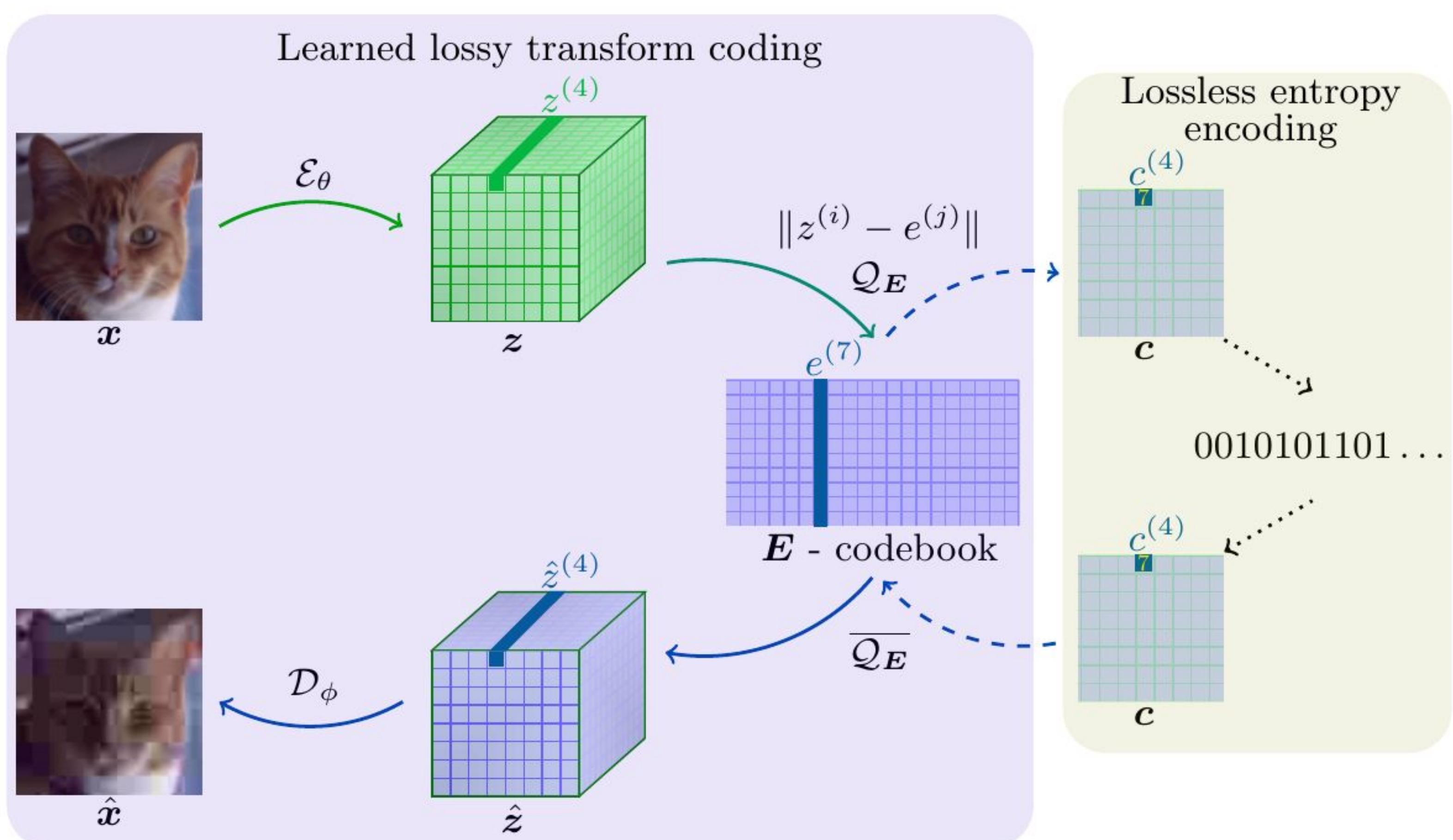
Learned transform compression with optimized entropy encoding

Magda Gregorová, Marc Desaules Alexandros Kalousis
[name.surname]@hesge.ch
Geneva School of Business Administration,
HES-SO University of Applied Sciences of Western Switzerland

Learned transform coding with vector quantization and lossless entropy encoding



Rate-distortion trade off

Compression loss: learn encoder, decoder and quantization codebook

$$\mathcal{E}_\theta, \mathcal{D}_\phi, \mathcal{Q}_{\boldsymbol{E}} \qquad \mathcal{L} := \underbrace{\mathbb{E}_{\mu_x} d(\mathsf{x}, \hat{\mathsf{x}})}_{distortion} + \lambda \underbrace{\mathbb{E}_{\mu_c} l(\mathsf{c})}_{rate}$$

Shannon: optimal code length ≈ entropy

$$\mathbb{E}_{\mu_c} l^*(c) = -\mathbb{E}_{\mu_c} \log p_c(\mathsf{c}) = \mathbb{H}_{\mu_c}(\mathsf{c})$$

Code distribution: approximate by learned distribution

$$q_c \approx p_c \qquad -\mathbb{E}_{\mu_c} \log q_c(\mathsf{c}) \approx -\mathbb{E}_{\mu_c} \log q_c(\mathsf{c}) \qquad \mathbb{H}_{\mu_c|q_c}(\mathsf{c}) \approx \mathbb{H}_{\mu_c}(\mathsf{c})$$

Cross-entropy loss: rate as cross-entropy instead of entropy

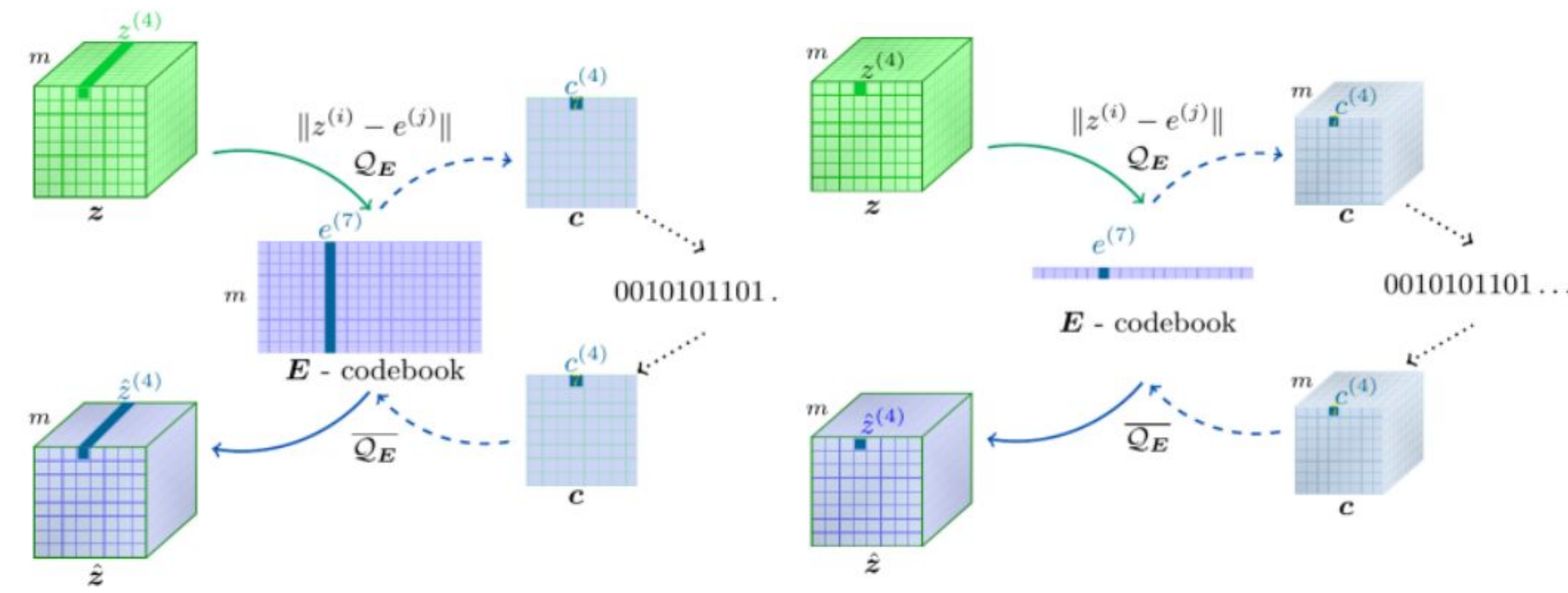
$$\mathcal{E}_\theta, \mathcal{D}_\phi, \mathcal{Q}_{\boldsymbol{E}}, \mathcal{P}_\psi \qquad \mathcal{L} := \underbrace{\mathbb{E}_{\mu_x} d(\mathsf{x}, \hat{\mathsf{x}})}_{distortion} + \lambda \underbrace{\mathbb{H}_{\mu_c|q_c}(\mathsf{c})}_{rate}$$

Quantization

Nearest neighbour search

$$\mathcal{Q}_{\boldsymbol{E}}: \quad \hat{\boldsymbol{z}}^{(i)} = \arg\min_{\boldsymbol{e}^{(j)}} \|\boldsymbol{z}^{(i)} - \boldsymbol{e}^{(j)}\| \qquad c^{(i)} = \{j : \hat{\boldsymbol{z}}^{(i)} = \boldsymbol{e}^{(j)}\}$$

Vector vs scalar quantization: message length vs flexibility



vector quantization
message: **short**
flexibility: low

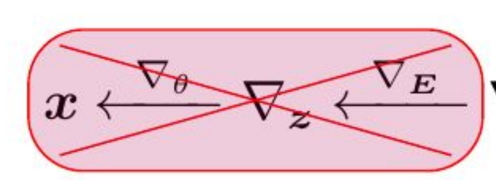
scalar quantization
message: long
flexibility: **high**

Problems

Quantization gradients: codebook and encoder not updated

$$\text{Forward:} \quad \boldsymbol{x} \xrightarrow{\mathcal{E}_\theta} \boldsymbol{z} \xrightarrow{\mathcal{Q}_{\boldsymbol{E}}} \hat{\boldsymbol{z}} \xrightarrow{\mathcal{D}_\phi} \hat{\boldsymbol{x}} \longrightarrow d(\boldsymbol{x}, \hat{\boldsymbol{x}})$$

$$\text{Backward:} \quad \cancel{\boldsymbol{x} \xleftarrow{\nabla_\theta} \nabla_{\boldsymbol{z}} \xleftarrow{\nabla_{\boldsymbol{E}}}} \nabla_{\hat{\boldsymbol{z}}} \xleftarrow{\nabla_\phi} \nabla_{\hat{\boldsymbol{x}}} \longleftarrow d(\boldsymbol{x}, \hat{\boldsymbol{x}})$$



Entropy minimization: cross-entropy objective not learning code transform

$$\mathbb{H}_{\mu_c|q_c}(\mathsf{c}) = \overbrace{D_{\mathrm{KL}}(p_c \| q_c)}^{\geq 0} + \mathbb{H}_{\mu_c}(\mathsf{c})$$

$$\mathcal{P}_\psi: \quad \min_{q_c} \ \mathbb{H}_{\mu_c|q_c}(\mathsf{c}) \approx -\frac{1}{n} \sum_i^n \log q_c(c_i), \quad c_i \sim \mu_c$$

$$\Leftrightarrow \ \min_{q_c} \ D_{\mathrm{KL}}(p_c \| q_c) + \cancel{\mathbb{H}_{\mu_c}(\mathsf{c})}$$

$$q_c \to p_c \qquad p_c \ \text{fixed}$$

Minimization of cross-entropy only with respect to the learned distribution $\mathbf{q_c}$ approximating the unknown code distribution $\mathbf{p_c}$ is suboptimal. It brings $\mathbf{q_c}$ close to $\mathbf{p_c}$ and hence reduces the number of extra bits needed due to using $\mathbf{q_c}$ instead of the true $\mathbf{p_c}$ but it does not encourage low entropy of the true code distribution.

Solutions

Push-forward mesure & soft quantization: *hard and soft cross-entropy*

1) $$\mu_c[\mathsf{c} \in \boldsymbol{A}] = \mu_c[\mathcal{T}_{\boldsymbol{E},\theta}(\mathsf{x}) \in \boldsymbol{A}] = \mu_x[\mathsf{x} \in \mathcal{T}^{-1}_{\boldsymbol{E},\theta}(\boldsymbol{A})] \qquad \mathcal{T}_{\boldsymbol{E},\theta} = \mathcal{Q}_{\boldsymbol{E}} \circ \mathcal{E}_\theta$$

$$\mathcal{Q}_{\boldsymbol{E}}, \mathcal{E}_\theta \to \mu_c \to \mathbb{H}_{\mu_c}$$

2) $$p_c(c = j) = \begin{cases} 1 & \text{if } \hat{\boldsymbol{z}} = \boldsymbol{e}^{(j)} \\ 0 & \text{otherwise} \end{cases} \qquad h_{ce} = -\frac{1}{n} \sum_i^n \log q_c(c_i), \quad c_i \sim \mu_c$$

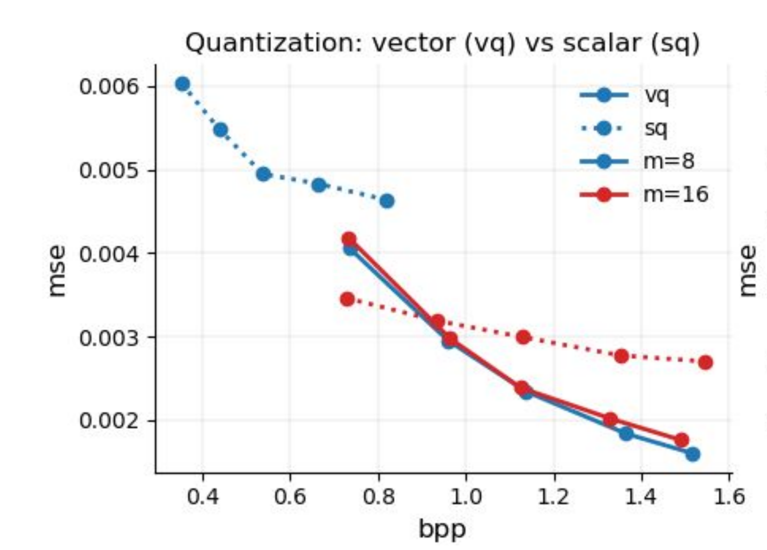
$$\hat{p}_c(c = j) = \frac{\exp(-\sigma \|\boldsymbol{z} - \boldsymbol{e}^{(j)}\|)}{\sum_j^k \exp(-\sigma \|\boldsymbol{z} - \boldsymbol{e}^{(j)}\|)} \qquad s_{ce} = -\frac{1}{n} \sum_{i,j}^{n,k} \hat{p}_c(c_i = j) \log \mathrm{sg}[q_c(j)]$$

3) $$\tilde{\boldsymbol{z}} = \sum_j^k \hat{p}_c(c = j) \, \boldsymbol{e}^{(j)} \qquad \hat{d}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = d(\boldsymbol{x}, \mathcal{D}_\phi[\mathrm{sg}(\hat{\boldsymbol{z}} - \tilde{\boldsymbol{z}}) + \tilde{\boldsymbol{z}}])$$

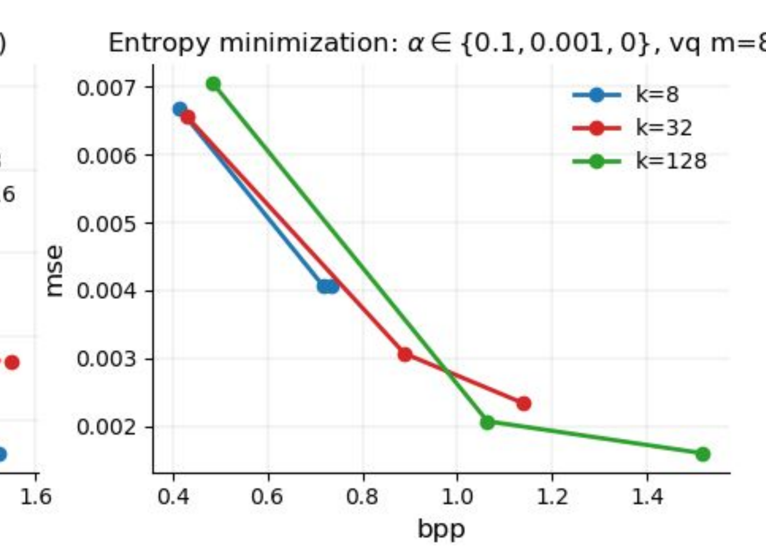
$$\arg\min_{\mathcal{E}_\theta, \mathcal{Q}_{\boldsymbol{E}}, \mathcal{D}_\phi, \mathcal{P}_\psi} \ \frac{1}{n} \sum_i^n \hat{d}(\boldsymbol{x}_i, \hat{\boldsymbol{x}}_i) + \alpha \, s_{ce}(\boldsymbol{c}_i) + \beta \, h_{ce}(\boldsymbol{c}_i), \quad \boldsymbol{x}_i \sim \mu_x$$

Proof of concept experiments

CIFAR: vector vs scalar quantization, effect of soft cross-entropy term



solid / dashed lines - vector / scalar quantization; blue / red - m = 8 / 16 channels in latent representation; points from left to right - increasing number of codebook words k ={8, 16, 32, 64, 128}; α=0, β=1, σ=1



all lines vector quantization with m=8 channels in latent representation; blue / red / green - k = 8 / 32 / 128 codebook words; points from right to left increasing strength of soft cross-entropy α = {0, 0.001, 0.1}, β=1, σ=1

References

Agustsson, E., Mentzer, F., Tschannen, M., Cavigelli, L., Timofte, R., Benini, L., & Van Gool, L. (2017). "Soft-to-Hard Vector Quantization for End-to-End Learning Compressible Representations." arXiv:1704.00648.

Balle, J., Laparra, V. & Simoncelli, E. P. (2017). "End-to-end Optimized Image Compression." ICLR.

Mentzer, F., Agustsson, E., Tschannen, M., Timofte, R., Van Gool, L. (2018). "Conditional Probability Models for Deep Image Compression." CVPR.

Theis, L., Shi, W., Cunningham, A. & Huszar, F. (2017). "Lossy Image Compression with Compressive Autoencoders." ICLR.

van den Oord, A., Vinyals, O. & Kavukcuoglu, K. (2017). "Neural Discrete Representation Learning." NeurIPS.