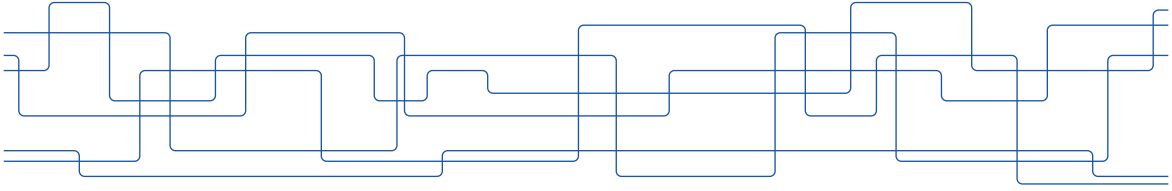
Learning Product Codebooks using Vector-Quantized Autoencoders for Image Retrieval

Hanwei Wu and Markus Flierl
KTH Royal Institute of Technology, Stockholm, Sweden



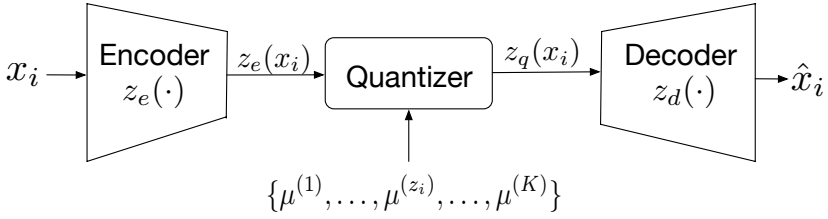
Motivation

- Variational Autoencoders (VAE) [1] and its variations have emerged as a popular unsupervised learning method for representation learning.
- Large-scale image retrieval requires structured latent representations, i.e. learning discrete representations.
- Design a product quantized autoencoder and leverage the bottleneck quantizer to enforce a similarity-preserved representation mapping at the encoder.

[1] D. Kingma, M. Welling, "Auto-Encoding Variational Bayes", in ICLR, 2014

Background: Vector-quantized Autoencoders[2]

- Encoder-decoder structure



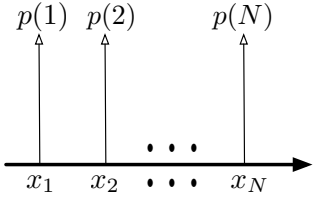
- Notation
  - i: Datapoint index i
  - x: Datapoint
  - $z_e(x_i)$: Output of the encoder
  - $\mu^{(z_i)}$: Latent codeword for input x_i
  - K: Size of the codebook
  - z: Index of the latent codeword
  - $z_q(x_i)$: Input of the decoder $z_d(\cdot)$
  - $\hat{x}$: input reconstruction

[2] A. Oord, O. Vinyals and K. Kavukcuoglu, "Neural Discrete Representation Learning", in NIPS, 2017

Background: Vector-quantized Autoencoders

- Model for the training data[3]
  - Index-datapoint pairs

$$p(x) = \sum_{i}^{N} p(x|i)p(i) = \sum_{i}^{N} \delta(x - x_i)p(i).$$



  - Assume $p(i) = \frac{1}{N}$ for N training samples.
- Bottleneck Quantizer
  - Nearest neighbor index assignment

$$z_i = \underset{j \in [1, \cdots, K]}{\operatorname{argmin}} \|z_e(x_i) - \mu^{(j)}\|_2^2.$$

  - Input for the decoder $z_q(x_i) = \mu^{(z_i)}$.

[3] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep variational information bottleneck", in ICLR, 2017

Entropy-distortion Formulation for Vector-quantized Autoencoders

- Define the information bottleneck distortion measure $d_{\mathrm{IB}}(I, Z)$ as the Kullback–Leibler (KL) divergence between $p(x|i)$ and $p(x|z)$[4].
- Define the representational cost as the entropy of the latent index $H(Z)$.
- Use Lagrangian formulation to exploit the entropy/distortion trade-off

$$L_{\mathrm{IB}} = d_{\mathrm{IB}}(I, Z) + \epsilon H(Z),$$

where $\epsilon > 0$ is the Lagrangian parameter.

[4] A. Gilad-Bachrach, A. Navot, and N. Tishby, "An information theoretic tradeoff between complexity and accuracy", in COLT, 2003

Information Bottleneck Distortion Measure

- $d_{\mathrm{IB}}(I, Z)$ can be decomposed into two terms, where the second term is determined solely by the given data distribution $p(i, x)$

$$d_{\mathrm{IB}}(I, Z) = \mathrm{KL}(p(x|i) \| p(x|z)) = \int \sum_z p(x, z) \log \frac{p(z)}{p(x, z)} dx - \int \sum_i p(i, x) \log \frac{p(i)}{p(i, x)} dx.$$

- A tractable upper bound of d_{IB} can be obtained by replacing the intractable $p(x|z)$ with a variational approximation $q(x|z)$

$$\int \sum_z p(x, z) \log \frac{p(z)}{p(x, z)} dx \leq - \sum_i p(i) \int p(x|i) \sum_z p(z|i) \log q(x|z) dx.$$

- With the model $q(x|z) = \mathcal{N}(x|\hat{x}, 1)$, the log-likelihood of $q(x|z)$ is proportional to the squared difference between the input x and the output $\hat{x}$.

Upper Bound for the Representational Cost

- The representational cost can be upper bounded by the cross entropy between $p(z)$ and the variational $r(z)$

$$H(Z) \leq -\sum_{z} p(z) \log r(z) = -\sum_{i} \sum_{z} p(i)p(z|i) \log r(z) = -H(p, r)$$

- If $r(z)$ is assumed to be uniform, the cross entropy upper bound is a constant that equals to $\log K$, where K is the number of codewords.

- As a result, the constraint on the learned representations is predetermined by the size of the embedded codebook.

Training of the Vector-quantized Autoencoders

- Objective function

$$L_{\text{VQ-VAE}} = \frac{1}{N}\sum_{i=1}^{N}\left[\log q(x|z_q(x_i)) + \|\text{sg}(z_e(x_i)) - z_q(x_i)\|_2^2 + \beta\|z_e(x_i) - \text{sg}(z_q(x_i))\|_2^2\right],$$

  where the third term is the commitment loss which forces the encoder output to commit to a codeword.

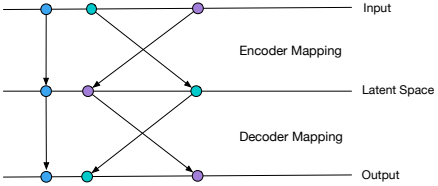
- The stop gradient operator $\text{sg}(\cdot)$ is used to separate the gradient update of the encoder-decoder pair and the codebook

$$sg(x) = \begin{cases} x & \text{forward pass} \\ 0 & \text{backward pass} \end{cases}$$

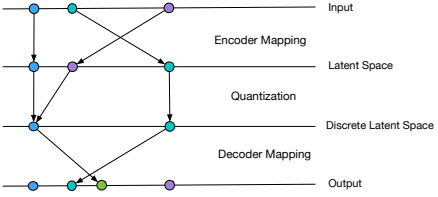
  - $\log q(x|z_q(x_i)))$: The Gradients of this term only update encoder and decoder.
  - $\|\text{sg}\left(z_e(x_i)\right) - z_q(x_i)\|_2^2$: The gradients of this term only update the codebook.
  - $\|z_e(x_i) - \text{sg}\left(z_q(x_i)\right)\|_2^2$: The gradients of this term only update the encoder.

Regularization Effects of the Bottleneck VQ

- Vanilla autoencoder model [5]



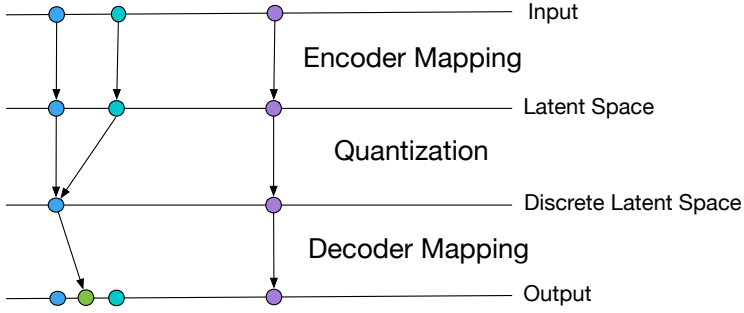
- Autoencoders with embedded quantizer



[5] D. E. Rumelhart, G. E. Hinton and R. J. Williams, "Learning representations by back-propagating errors", in book: PDP, 1986

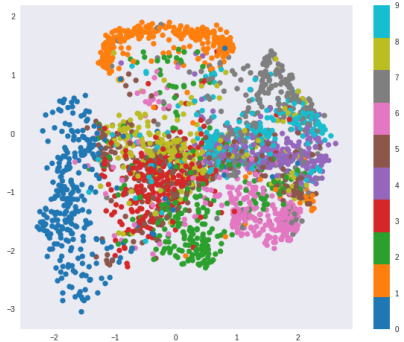
Regularization Effects of the Bottleneck VQ

- VQ enforces similarity-preserved mapping

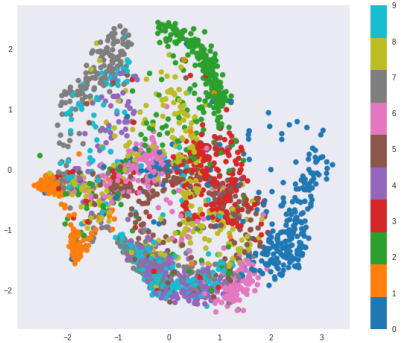


Impact of the Codebook Size

- Smaller codebooks provide low discriminability of the input data. In order to have a small reconstruction loss, the encoder is forced to ensure that neighboring data points are also represented closely together in the latent space.
- Larger codebooks provide high discriminability of the input data. To achieve a small reconstruction loss, the output of the encoder is more likely to be quantized into codewords that are far away from each other.



Visulization of $z_e(x_i)$ with $K = 20$



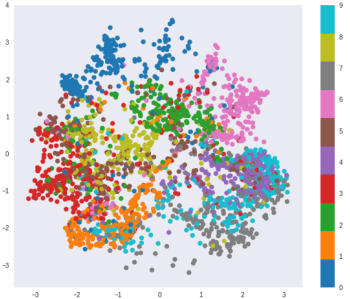
Visulization of $z_e(x_i)$ with $K = 25$

Control the Impact of the Bottleneck VQ

- We introduce a hyperparameter $\lambda > 0$ on the updating terms for quantizer and encoder to control the strength of the vector quantizer

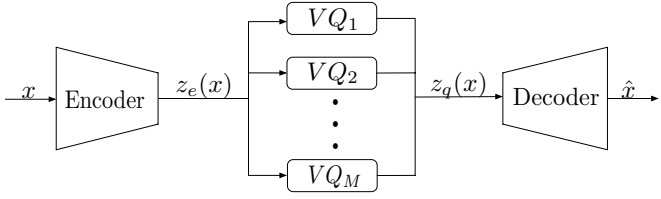
$$L_{\text{VQ-VAE}} = \frac{1}{N}\sum_{i=1}^{N} \log q(x|z) + \lambda \left(\|\text{sg}\left(z_e(x_i)\right) - z_q(x_i)\|_2^2 + \beta \|z_e(x_i) - \text{sg}\left(z_q(x_i)\right)\|_2^2 \right)$$

- $\lambda \uparrow \implies$ the discriminability of the latent code $\uparrow$.
- $\lambda \downarrow \implies$ the discriminability of the latent code $\downarrow$.



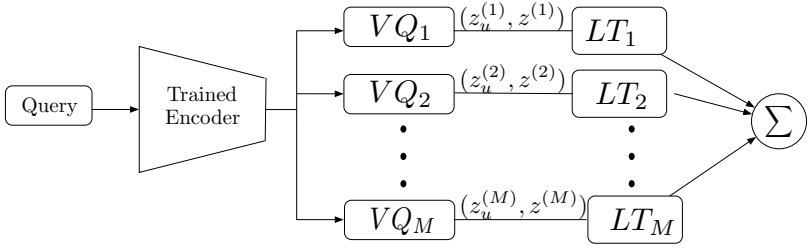
Visualization of $z_e(x_i)$ with $K = 25$ and $\lambda = 0.5$

Product-quantized Autoencoders

- The product quantizer has the advantage of generating large-size codebooks.
- Fast retrieval can be achieved by using lookup tables that store the distance between any pair of subcodewords.
- 

  - The output of the encoder is quantized by M independent sub-vector quantizers (SQ).
  - Product quantizer can generate K^M codewords.
  - After training, each sub-vector quantizer provides a lookup table (LT) with $K \times K$ entries.

Querying Process



- The querying is conducted in the quantized space.
- Distance between query q and data item x can be obtained by M lookup tables LT_m .

$$d(q, x) = LT_1\left(z_u^{(1)}, z^{(1)}\right) + \cdots + LT_M\left(z_u^{(M)}, z^{(M)}\right),$$

where z_u denotes the latent index of the query image, and z the latent index of the given database image x.

Retrieval Results

- CIFAR-10 dataset
- Mean average precision mAP-1000 metric
- Each x is represented by 32, 48 or 64 bits.

	32 bits	48 bits	64 bits
LSH (Datar et al., 2004)	12.00	12.00	15.07
Spectral Hashing (Weiss et al., 2008)	13.30	13.00	13.89
Spherical Hashing (Heo et al., 2015)	13.30	13.00	15.38
ITQ (Gong et al., 2013)	16.20	17.50	16.64
Deep Hashing (Liong et al., 2015)	16.62	16.80	16.69
PQ-VAE	**21.86**	**22.79**	**23.42**

Conclusions

- The vector-quantized autoencoder is studied with the information bottleneck framework
- The regularization term for the learned representation is determined by the size of the embedded codebook.
- We introduce a hyperparameter to control the impact of the vector quantizer such that we can further regularize the latent representation.
- We introduce the product quantizer into the bottleneck stage of the autoencoder to facilitate large-scale image retrieval.