**VECTOR QUANTIZATION (VQ)**

**Introduction**

Consider an arbitrary fixed-rate lossy source code that operates independently on blocks (vectors) of $k$ real-valued samples: $k$ samples into encoder, $L$ bits out

- samples $X_1...X_k$
- bits $Z_1...Z_L$
- reproductions $Y_1...Y_k$

- source coder = encoder + decoder
- encoder and decoder are described by functions called the encoding rule and decoding rule, respectively
- the sets of all possible encoder outputs and all possible decoder outputs play important roles
- the partition induced on space $k$-dimensional input vectors plays an important role
- a lossy source code that operates independently on fixed-length blocks, producing fixed-length blocks of bits is called a

  *fixed-rate or fixed-length (memoryless) vector quantizer* (VQ)

- fixed-rate VQ is a very general paradigm that includes many lossy source codes as special cases, e.g. fixed-rate transform coding. Since it is quite generable and also *analyzable*, it provides an excellent framework for studying lossy source codes.
- JPEG has variable, not fixed, rate. Except for the encoding of dc coefficients, it operates independently on blocks of 64 pixels. (It's a variable-rate VQ with memory.)
**Key Characteristics** (high-level, input-output)

**Dimension:** \( k \)

**Encoding rule:** \( \alpha: \mathbb{R}^k \rightarrow \{0,1\}^L \)
- \( Z_1 \ldots Z_L = \alpha(X_1 \ldots X_k) \),
- \( Z_{L+1} \ldots Z_{2L} = \alpha(X_{k+1} \ldots X_{2k}) \),
- \( Z_{2L+1} \ldots Z_{3L} = \alpha(X_{2k+1} \ldots X_{3k}) \), ...

**Binary codebook:**
- \( C_b = \{e(x): x \in \mathbb{R}^k\} = \{c_1, c_2, \ldots, c_M\} \)
- where \( c_i = (c_{i1}, c_{i2}, \ldots, c_{iL}) \)
  - = \( i \)th binary codeword

**Size of code:** \( M \)

**Decoding rule** \( \beta: C_b \rightarrow \mathbb{R}^k \)
- \( Y_1 \ldots Y_k = d(Z_1 \ldots Z_L) \),
- \( Y_{k+1} \ldots Y_{2k} = d(Z_{L+1} \ldots Z_{2L}) \),
- \( Y_{2k+1} \ldots Y_{3k} = d(Z_{2L+1} \ldots Z_{3L}) \), ...

**(Quantization) Codebook:**
- \( C = \{\beta(c_1), \beta(c_2), \ldots, \beta(c_M)\} = \{w_1, w_2, \ldots, w_M\} \)
- where \( w_i = (w_{i1}, w_{i2}, \ldots, w_{ik}) = \) ith codevector
  (code/reproduction, vector/point)

**Quantization rule:** \( Q: \mathbb{R}^k \rightarrow \mathbb{R}^k \)
- \( Q(x) = \beta(\alpha(x)) = \) reproduction produced by decoder for \( x \)

**(Quantizing) Partition:** \( S = \{S_1, S_2, \ldots, S_M\} \)
- where \( S_i = \{x \in \mathbb{R}^k: \alpha(x) = c_i\} = \{x \in \mathbb{R}^k: Q(x) = w_i\} \)
  - = \( i \)th (quantizing) cell

(A **partition** of \( \mathbb{R}^k \) is a collection of disjoint subsets of \( \mathbb{R}^k \) whose union is \( \mathbb{R}^k \). The elements of the collection are called **cells**.)
SUMMARY:  A VQ IS CHARACTERIZED BY

Dimension:  k
Size:  M

Encoding rule:  \( \alpha \)
characterized by
  Partition:  \( S = \{S_1, S_2, \ldots, S_M\} \) and
  Binary codebook:  \( C_b = \{c_1, c_2, \ldots, c_M\} \)

Decoding rule:  \( \beta \)
characterized by
  Codebook  \( C = \{w_1, w_2, \ldots, w_M\} \)

Quantization rule:  \( Q \)
characterized by  \( S \) and  \( C \)
  \( Q(x) = \beta(\alpha(x)) = w_i \) when  \( x \in S_i \)

Notes
- When thinking about VQ, many people think first about the codebook  \( C \).
- It's probably better to think first about the partition  \( S \).
- You must learn to use this terminology and notation.
Examples (in k=2 dimensions)
Scalar Quantizer (k=1)
PERFORMANCE

RATE

\[ R = \frac{L}{k} \]

\[
= \begin{cases} 
\frac{1}{k} \log_2 M, & \text{if } M=\text{power of 2} \\
\frac{1}{k} \lceil \log_2 M \rceil, & \text{if } M \neq \text{power of 2} & \text{& one index coded at a time} \\
\frac{1}{mk} \lceil \log_2 M^m \rceil \approx \frac{1}{k} \log_2 M, & \text{if } M \neq \text{power of 2} & \text{& m indices coded at a time}
\end{cases}
\]

Accordingly, we assume

\[ R = \frac{1}{k} \log_2 M \quad \text{unless there is need to be picky.} \]

Units: bits/sample

Note: Rate is determined by the encoder, not the decoder.
DISTORTION

\[ D = \text{MSE} = \text{mean squared error (normalized by dimension)} \]

\[ = \frac{1}{k} \sum_{i=1}^{k} E(X_i-Y_i)^2 = \frac{1}{k} E \sum_{i=1}^{k} (X_i-Y_i)^2 \]

\[ = \frac{1}{k} E \|X-Y\|^2 = \frac{1}{k} E \|X-Q(X)\|^2 = \frac{1}{k} \int \|x-Q(x)\|^2 f_X(x) \, dx \]

\[ = \frac{1}{k} \sum_{i=1}^{M} \int \|x-w_i\|^2 f_X(x) \, dx \]

where

\[ \|x-y\| = \text{Euclidean distance} = \sqrt{\sum_{i=1}^{k} (x_i-y_i)^2} \]

expected value is with respect to probability distribution on \( X = (X_1, \ldots, X_k) \)

Source vector:
We assume \( X \) is modeled as a vector of jointly continuous random variables \( X = (X_1, \ldots, X_k) \), whose probability distribution is characterized by a joint density denoted \( f_X(x) \) (or just \( f(x) \)).

When a VQ operates on sequence of vectors, \( X_1, X_2, \ldots \), we usually assume these vectors come from a stationary random process.

Note: Distortion is determined by \( Q \) (or partition and codebook). The specific encoding rule, decoding rule and binary codebook do not matter, except as they determine \( Q \).