Lecture 14

Practical Transform Coding; Audio Compression

Announcements

*) Bonus points for HW2 feedback
[check gradescope; due by Wed]

*) HW3 due this Wednesday

*) IT forum: Shirin Bidokhti
Learning-based data compressn
Fri, 2pm, Packard 202

Recap

> **Thumb-rule for lossy compression**: For a given distortion measure, allocate more bits to the components with higher variance.

1. Learnt about Water-Filling Intuition for Gaussian Sources

   Recall the problem of compressing two independent Gaussian sources X_1, X_2 with means 0 and variances σ_1^2 and σ_2^2. For the squared error distortion we saw in class, the rate distortion function is given by

$$R_G\left(\begin{bmatrix}\sigma_1^2\\ \sigma_2^2\end{bmatrix}, D\right) = min_{\frac{1}{2}(D_1+D_2)\le D}\ \frac{1}{2}\left[\left(\frac{1}{2}\log\frac{\sigma_1^2}{D_1}\right)_+ + \left(\frac{1}{2}\log\frac{\sigma_2^2}{D_2}\right)_+\right]$$

Quiz Q1

Now consider a setting with $\sigma_1^2 = 1$ and $\sigma_2^2 = 3$.

At $D = 1.5$, what are the optimal values of D_1 and D_2:

- ( ) $D_1 = D_2 = 1$
- ( ) $D_1 = D_2 = 0.5$
- ( ) $D_1 = 1, D_2 = 3$
- ( ) $D_1 = 1, D_2 = 2$ ✓

$D > \sigma_1^2 \Rightarrow D_1 = \sigma_1^2$

$D_2 = D - D_1$

Quiz Q2

At $D = 2$, what is the optimal rate

( ) 0 bits/source component
( ) 1 bits/source component
( ) 2 bits/source component
( ) 3 bits/source component

$D = 2 = \frac{\sigma_1^2 + \sigma_2^2}{2}$

$D_1 = \sigma_1^2 \Rightarrow R_1 = 0$

$D_2 = \sigma_2^2 \Rightarrow R_2 = 0$

$\frac{1}{2}\log(\sigma^2/D)$

Quiz Q3

Which of the following is correct?

$D_1 = D_2 \Rightarrow R_1 < R_2 \; (\sigma_1 < \sigma_2)$

[ ] For D below the two variances, we divide the distortions equally among the two components.

[ ] For D below the two variances, we use a higher bitrate for the component with higher variance.

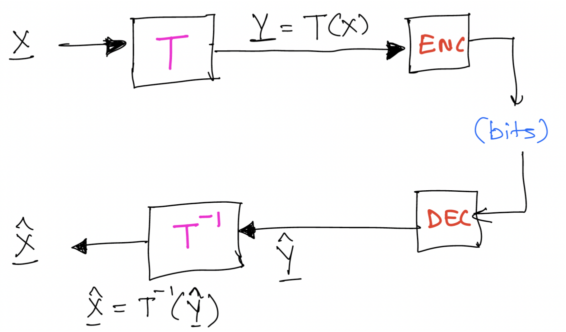
[ ] For D between the two variances, we use zero bitrate for one of the component.

[ ] For D between the two variances, we use zero bitrate for both of the components.

Recap

2. Learnt about Transform Coding setup

- Benefits:
  - **Decorrelation**: X can be correlated, aim to de-correlate it
    - Allows to use simpler quantization schemes
  - **Energy compaction**: more *energy* in first few components of $\underline{Y}$ than in the last few
    - Allows to allocate more bits to the components with higher energy



Recap

3. Learnt about Karhunen-Loeve Transform (KLT)

- The KLT is the eigenvalue-based linear transform.
- We can use this to get de-correlated components of X by using $Y = U^T X$, i.e. $T = U^T$, where U was the matrix of eigenvectors of the *covariance matrix* of X.

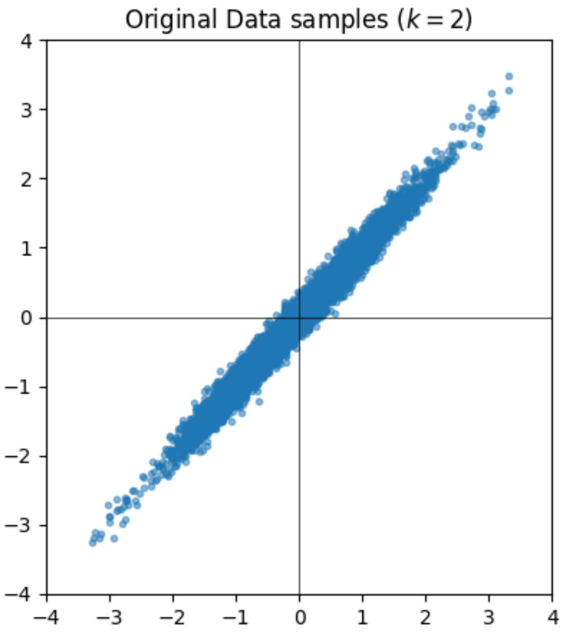
> Main idea: transform the data to a new basis where the components are uncorrelated and have different variance.

Today

1. More examples
2. Practical transforms: DCT, ...
3. Application: Audio Compression

Decorrelation Example

Example: consider a source $X_n = \rho X_{n-1} + \sqrt{1-\rho^2}\mathcal{N}(0,\sigma^2)$, $X_0 \sim \mathcal{N}(0,\sigma^2)$.
We will work with blocks of 2, i.e. $k = 2$.



Gauss-Markov

$$\boxed{X_0 \sim \mathcal{N}(0, \sigma^2)}$$

$$X_n = \rho X_{n-1} + \sqrt{1-\rho^2}\,\underbrace{\mathcal{N}(0,\sigma^2)}_{\text{iid}}$$

$$\Downarrow$$

$$X_1 = \overset{?}{\rho} X_0 + \sqrt{1-\rho^2}\,\mathcal{N}(0,\sigma^2)$$

$$\hookrightarrow X_1 \sim \mathcal{N}\left[0+0,\ (\rho^2 + (1-\rho^2))\sigma^2\right]$$

$$\sim \mathcal{N}(0, \sigma^2)$$

$$X_1 \sim \mathcal{N}(0, \sigma^2)$$

$$\vdots$$

$$X_n \sim \mathcal{N}(0, \sigma^2)$$

KLT Example

Example: consider a source $X_n = \rho X_{n-1} + \sqrt{1-\rho^2}\mathcal{N}(0,\sigma^2)$, $X_0 \sim \mathcal{N}(0,\sigma^2)$. We will work with blocks of 2, i.e. $k = 2$.

Quiz-4: What is the 2×2 covariance matrix Σ of X?

HINT: your sequence is stationary!

$$\Sigma = \mathbb{E}\left[\begin{bmatrix} X_i - \mathbb{E}X_i \\ X_{i+1} - \mathbb{E}X_{i+1} \end{bmatrix}\begin{bmatrix} X_i - \mathbb{E}X_i & X_{i+1} - \mathbb{E}X_{i+1} \end{bmatrix}\right]$$

Process is stationary!

$$\Sigma = \begin{bmatrix} \mathbb{E}(x_0 x_0^T) = \sigma^2 & \mathbb{E}(x_0 x_1^T) = \rho\sigma^2 \\ \mathbb{E}(x_1 x_0^T) = \rho\sigma^2 & \mathbb{E}(x_1 x_1^T) = \sigma^2 \end{bmatrix} \quad (\because \text{Means are } 0)$$

$$\mathbb{E}(x_0 x_0^T) = \mathbb{E}(x_1 x_1^T) = \sigma^2$$

$$\because (x_0 = x_1 \sim \mathcal{N}(0, \sigma^2))$$

$$\mathbb{E}(x_0 x_1^T) = \mathbb{E}(x_1 x_0^T) = \mathbb{E}\left[\left(\rho x_0 + \sqrt{1-\rho^2}\, \mathcal{N}(0,\sigma^2)\right) x_0\right]$$

$$= \underbrace{\mathbb{E}(\rho x_0^2)}_{\rho\, \mathbb{E}(x_0^2) = \rho\sigma^2} + \underbrace{\mathbb{E}\left(\sqrt{1-\rho^2}\, \mathcal{N}(0,\sigma^2) \cdot x_0\right)}_{= \mathbb{E}(\sqrt{1-\rho^2}\, \mathcal{N}(0,\sigma^2)) \cdot \mathbb{E}(x_0) = 0}$$

KLT Example

Example: consider a source $X_n = \rho X_{n-1} + \sqrt{1-\rho^2}\mathcal{N}(0,\sigma^2)$, $X_0 \sim \mathcal{N}(0,\sigma^2)$. We will work with blocks of 2, i.e. $k=2$.

Quiz-4: What is the 2×2 covariance matrix Σ of X?

$$\Sigma = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} \sigma^2$$

$$\Sigma = U^T \Lambda U$$

U : eigenvector matrix

Λ : diagonal matrix of eigenvals

$U = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$

KLT Example

$T = U^T$

Example: consider a source $X_n = \rho X_{n-1} + \sqrt{1-\rho^2}\mathcal{N}(0,\sigma^2)$, $X_0 \sim \mathcal{N}(0,\sigma^2)$. We will work with blocks of 2, i.e. $k = 2$.

Can show that the eigenvalues of Σ are
- $\lambda_1 = (1+\rho)\sigma^2$ and $\lambda_2 = (1-\rho)\sigma^2$
- corresponding eigenvectors are $u_1 = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}$ and $u_2 = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\-1\end{bmatrix}$.

Quiz-5: What is the eigenvalue-based transform at block-size $k = 2$ and transformed components Y?

$= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$

KLT Example

Example: consider a source $X_n = \rho X_{n-1} + \sqrt{1-\rho^2}\mathcal{N}(0,\sigma^2)$, $X_0 \sim \mathcal{N}(0,\sigma^2)$. We will work with blocks of 2, i.e. $k = 2$.

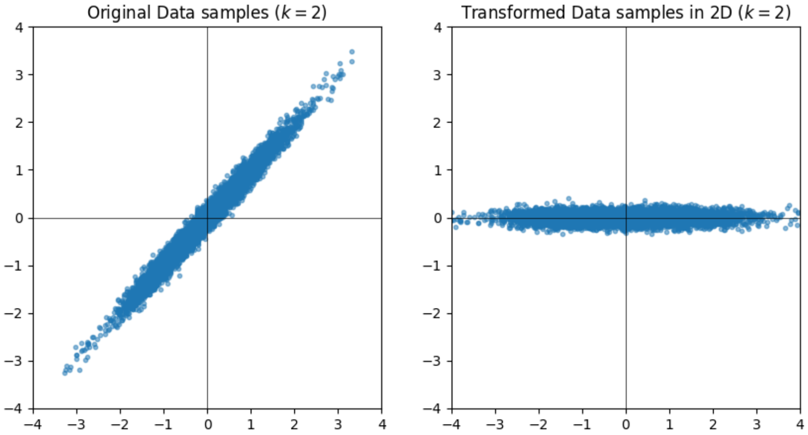
Quiz-5: What is the eigenvalue-based transform at block-size $k = 2$, transformed components Y?

$T = U^T = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1 \\ 1 & -1\end{bmatrix}$ and therefore $Y = TX = \frac{1}{\sqrt{2}}\begin{bmatrix}X_i + X_{i+1} \\ X_i - X_{i+1}\end{bmatrix}$

KLT Example

Example: consider a source $X_n = \rho X_{n-1} + \sqrt{1-\rho^2}\mathcal{N}(0, \sigma^2)$, $X_0 \sim \mathcal{N}(0, \sigma^2)$. We will work with blocks of 2, i.e. $k = 2$.

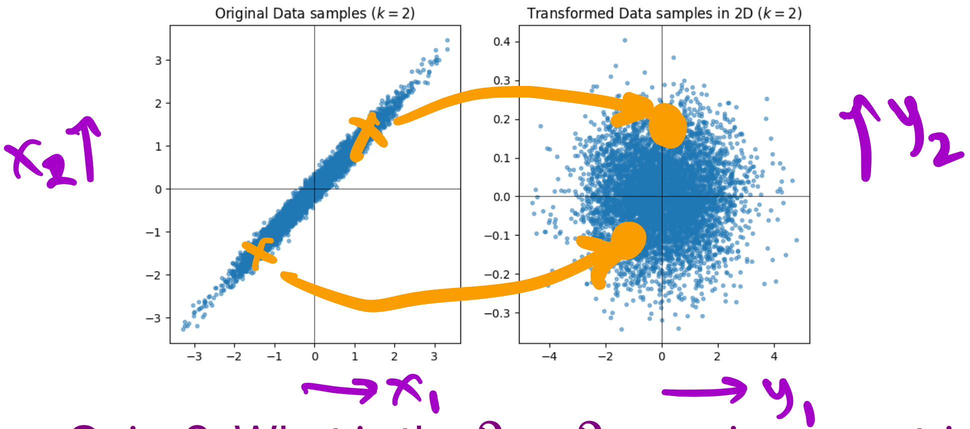
$$Y = TX = \frac{1}{\sqrt{2}} \begin{bmatrix} X_i + X_{i+1} \\ X_i - X_{i+1} \end{bmatrix}$$



Quiz-6: What is the 2×2 covariance matrix Σ of Y?

KLT Example

Example: consider a source $X_n = \rho X_{n-1} + \sqrt{1-\rho^2}\mathcal{N}(0,\sigma^2)$, $X_0 \sim \mathcal{N}(0,\sigma^2)$.



$\text{var}(y_1) = (1+\rho)\sigma^2$

$\text{var}(y_2) = (1-\rho)\sigma^2$

Quiz-6: What is the 2×2 covariance matrix Σ_Y of Y?

$\Sigma_Y = \begin{bmatrix} (1+\rho) & 0 \\ 0 & (1-\rho) \end{bmatrix} \sigma^2$, i.e. Y_1 and Y_2 are uncorrelated!

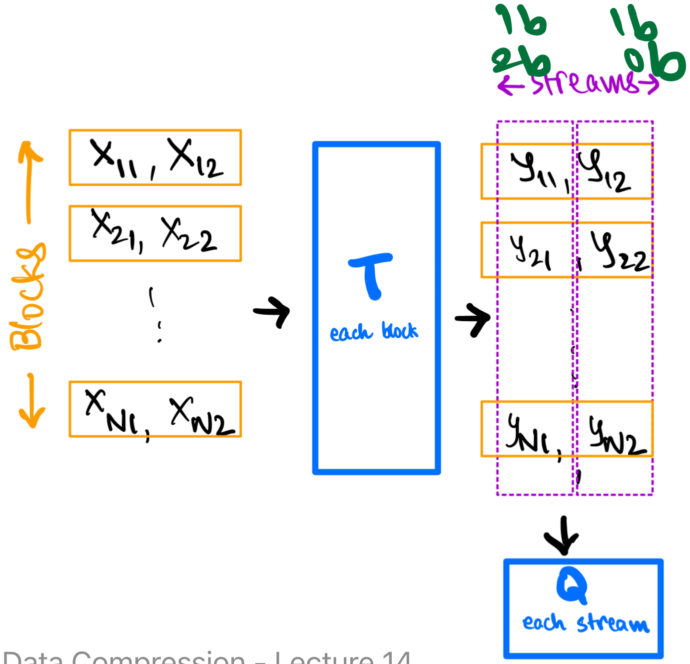
Moreover, the variances of Y_1 and Y_2 are such that Y_1 has higher variance than Y_2. This is the energy compaction property of the transform. (recall: water-filling!)

RD knobs for Transform Coding

In our example, we have two knobs to control the rate-distortion performance of transform coding:

1 bit /sample

1. Per-channel (tranformed component) bitrate split
2. Quantization scheme for each channel



$x_1, x_2, \dots, x_N$

x_{11}, x_{12}

x_{21}, x_{22}

$\vdots$

$\mathrm{var}(\text{stream } 1\ y) = (1+\rho)\sigma^2$

$\mathrm{var}(\text{stream } 2\ y) = (1-\rho)\sigma^2$

1 bit/symbol

Step I :

how many bits to allocate

bit-stream

$\boxed{(1, 1)}$; (2, 0)

Step II

Choose quantizatn for each stream independently

Component I : Scalar Quantizer

$|\text{Codebook}_{\text{I}}| = 2$

Component II : Vector Quantizer

$K = 2$

$|\text{Codebook}_{\text{II}}| = 4$

Transform Coding Notebook

https://colab.research.google.com/drive/1ZcnjIco0HEbiTQWvcpiPYA9HbtfB829x?usp=sharing

Transform Coding Performance on our Example

Example: consider a source $X_n = \rho X_{n-1} + \sqrt{1-\rho^2}\mathcal{N}(0, \sigma^2)$

```
==================================================
Processing rho: 0.9
==================================================
Vector Quantization Experiment
==================================================
[VQ][Bit per symbol: 1][Block Size: 2]Rate: 1.0, Distortion: 0.163
[VQ][Bit per symbol: 1][Block Size: 4]Rate: 1.0, Distortion: 0.095
==================================================
TC Vector Quantization Experiment
==================================================
[TC_VQ][Bit per symbol: 1][Block Size: 2][Bitrate Split: [1, 1]]Rate: 1.0, Distortion: 0.276
[TC_VQ][Bit per symbol: 1][Block Size: 2][Bitrate Split: [0, 2]]Rate: 1.0, Distortion: 0.970
[TC_VQ][Bit per symbol: 1][Block Size: 2][Bitrate Split: [2, 0]]Rate: 1.0, Distortion: 0.122
================================================
```

$(1+\rho)\sigma^2$
$(1-\rho)\sigma^2$

Transform Coding Performance on our Example

$\rho\uparrow$; Gap$\uparrow$

Example: consider a source $X_n = \rho X_{n-1} + \sqrt{1-\rho^2}\mathcal{N}(0,\sigma^2)$

```
==================================================
Processing rho: 0.99
==================================================
Vector Quantization Experiment
==================================================
[VQ][Bit per symbol: 1][Block Size: 2]Rate: 1.0, Distortion: 0.107
[VQ][Bit per symbol: 1][Block Size: 4]Rate: 1.0, Distortion: 0.020
==================================================
TC Vector Quantization Experiment
==================================================
[TC_VQ][Bit per symbol: 1][Block Size: 2][Bitrate Split: [1, 1]]Rate: 1.0, Distortion: 0.204
[TC_VQ][Bit per symbol: 1][Block Size: 2][Bitrate Split: [0, 2]]Rate: 1.0, Distortion: 0.890
[TC_VQ][Bit per symbol: 1][Block Size: 2][Bitrate Split: [2, 0]]Rate: 1.0, Distortion: 0.030
================================================
```

Transform Coding Performance on our Example

Example: consider a source $X_n = \rho X_{n-1} + \sqrt{1-\rho^2}\mathcal{N}(0, \sigma^2)$

```
==================================================
Processing rho: 0.5
==================================================
Vector Quantization Experiment
==================================================
[VQ][Bit per symbol: 1][Block Size: 2]Rate: 1.0, Distortion: 0.305
[VQ][Bit per symbol: 1][Block Size: 4]Rate: 1.0, Distortion: 0.271
==================================================
TC Vector Quantization Experiment
==================================================
[TC_VQ][Bit per symbol: 1][Block Size: 2][Bitrate Split: [1, 1]]Rate: 1.0, Distortion: 0.374
[TC_VQ][Bit per symbol: 1][Block Size: 2][Bitrate Split: [0, 2]]Rate: 1.0, Distortion: 0.786
[TC_VQ][Bit per symbol: 1][Block Size: 2][Bitrate Split: [2, 0]]Rate: 1.0, Distortion: 0.343
==================================================
```

Q. Why TC doesn't beat VQ for $\rho = 0.5$?!

Hint: $\{(1+\rho)\sigma^2, (1-\rho)\sigma^2\}$

Transform Coding + KLT: Issues

Quiz-1: Can you think of any issues with doing KLT in practice?

Ans:

- KLT is dependent on statistics of input data X!
  - KLT is optimal for a given covariance matrix Σ.
  - In practice, we do not know Σ and need to estimate it from data.
  - Moreover, data in real-life is not stationary, i.e., statistics change over time. Need to re-estimate Σ.
  - Therefore, in practice, KLT is computationally expensive!

Practical Transforms

Can we design a *structured* transform which is easy to compute and has good energy compaction properties?

How I Came Up with the Discrete Cosine Transform

Nasir Ahmed

Electrical and Computer Engineering Department, University of New Mexico, Albuquerque, New Mexico 87131

Source: [How I Came Up with the Discrete Cosine Transform]

Practical Transforms

What intrigued me was that the KLT was indeed the optimal transform on the basis of the mean-square-error criterion and the first-order Markov process model, and yet there was no efficient algorithm available to compute it. As such, the focus of my research was to determine whether it would be possible to come up with a good approximation to the KLT that could be computed efficiently. An approach that

Much to my disappointment, NSF did not fund the proposal; I recall one reviewer's comment to the effect that the whole idea seemed "too simple." Hence I de-

Source: How I Came Up with the Discrete Cosine Transform

Practical Transforms

Lots of options for practical transforms:

- DCT (Discrete Cosine Transform)
- DFT (Discrete Fourier Transform)
- Wavelets
- ...

Check out a nice list here.

Practical Transforms

- Most of these transforms are based on the idea of *orthogonal basis*.
- Many of them exploit the *sparsity* of the signal in some basis. E.g.:
  - DCT exploits the sparsity of the signal in the cosine (frequency) basis
- Leads to decorrelation and energy compaction properties because of the natural signal statistics. E.g.:
  - Natural audio and image signals are sparse in the frequency domain

Also, motivated by the fact that humans don't perceive high-frequency components as well as low-frequency components.

> Compress the high-frequency components more!

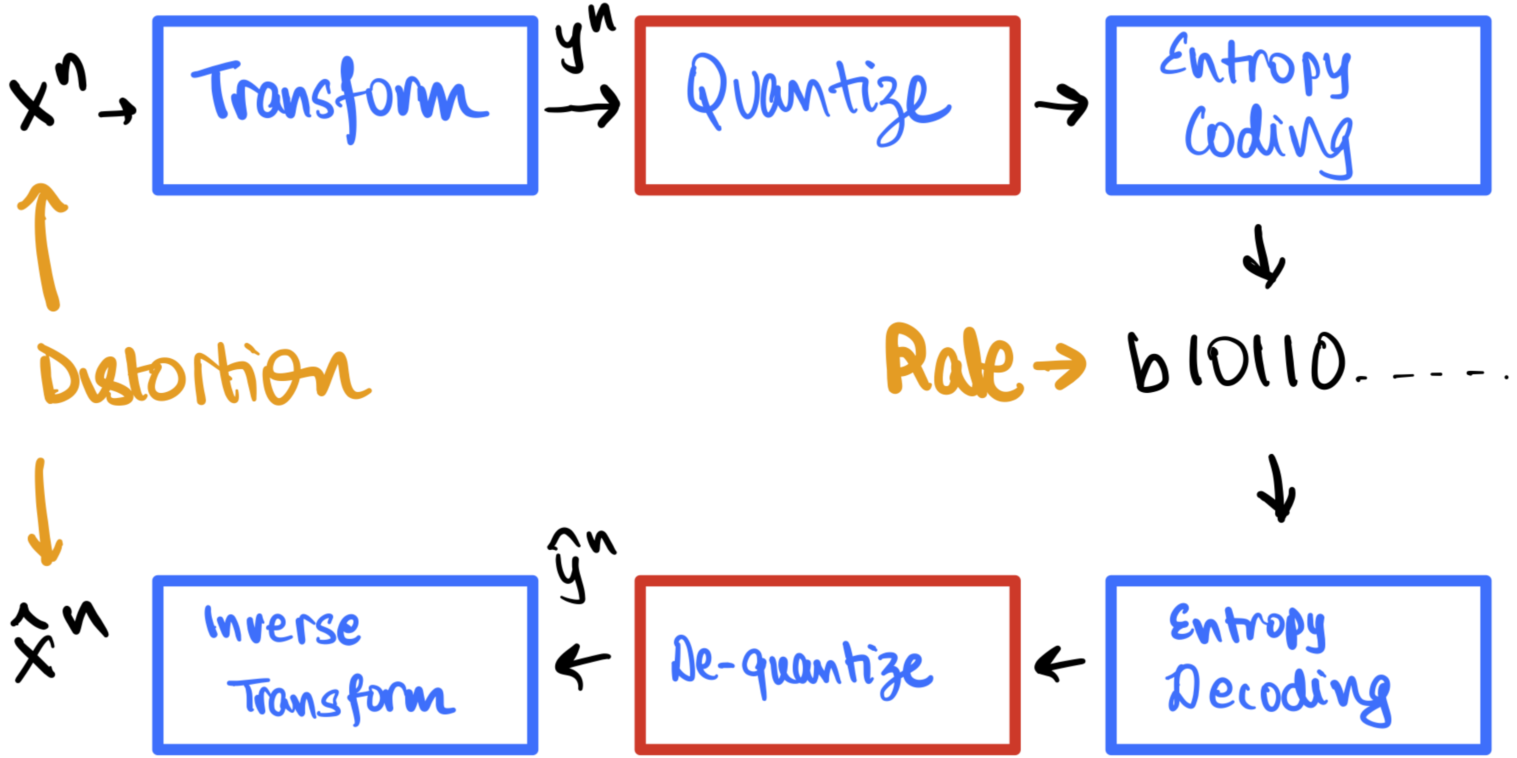
Practical Transforms: DCT

DCT is one of the most popular transforms used in practice for image and audio compression. DCT is

- values of cosine function at discrete points
- linear, in-fact, orthonormal transform which is a variant of the DFT (Discrete Fourier Transform)
- real-valued transform, i.e. the basis vectors are real-valued
- *lossless* transform

Various versions of DCT exist with different properties. We will focus on DCT-II which is the most popular version.
Let's build some intuition for DCT-II using Transform Coding Notebook.

Barebones Practical Lossy Compression



Lossy Compressor Design Decisions

- Choice of transform: DFT, DCT, …
- Choice of quantization
    - Only loss-step in the pipeline
    - Choice of quantization scheme: scalar, vector, …
    - Choice of quantization levels
    - Choice of distortion split between components
- Choice of entropy coding scheme: Huffman, ANS, …

Example: Audio Compression Notebook

https://colab.research.google.com/drive/13e81Rgv5KNbT1P_fcguPvIdtedogkEJZ?usp=sharing