System Model:

$$\underline{Y_i} = \underline{S_i}^T \underline{X_i} + \underline{N_i}$$

where $\underline{X_i} \in R^{m \times q}$ is the channel input vector taking real value with dimension q at time slot i.

For example, $\underline{X_1} = \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 3 & 4 \\ 4 & 5 \\ 1 & 6 \\ 2 & 7 \end{bmatrix}$ here $M = 6$.
$q = 2$.

$\underline{S_i}$ is an one-hot vector with dimension M.

For example, $\underline{S_1} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$ here $M = 6$.

$\underline{N_i} \in R^{1 \times q}$ is the Gaussian distributed channel noise. with mean zero and variance one

$$\underline{N_i} \sim \mathcal{N}(0, 1).$$

$\underline{Y_i} \in R^{1 \times q}$ is the channel output.

*. $\underline{S_i}$ remains the same for Q time slots.

* The average power of the channel input $\underline{X_i}$ for Q time slots should not be greater than B. $\Rightarrow E[\|\underline{X_i}\|^2] \le B$.

* An one bit feedback z_i is sent back to the transmitter from the receiver at the end of each time slot,

where $z_i = 1$ if $\|Y_i\|^2 \geq \alpha$.

$z_i = 0$ if $\|Y_i\|^2 < \alpha$

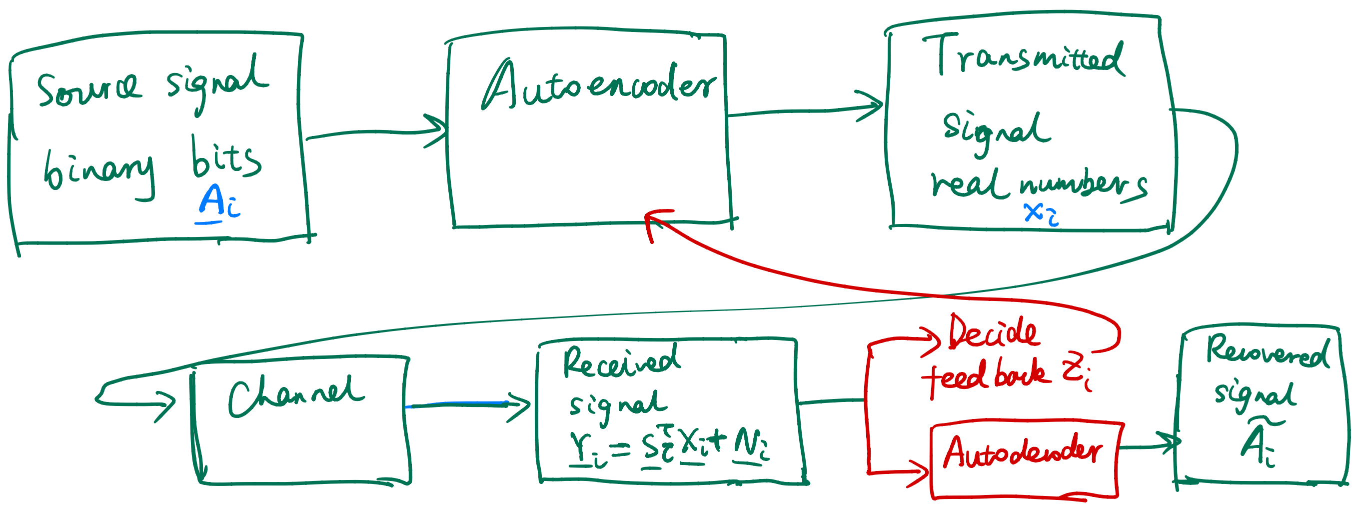
Design an autoencoder & autodecoder for this channel model to minimize the bit error rate (BER)

$$BER = \frac{\#\text{ of bit errors}}{\text{Total }\#\text{ of transmitted bits}}$$

for a given rate R.

For example, given $R = \frac{1}{3}$, $\Rightarrow$ it takes 3 time slots to transmit 1 bit.

Objective:



Design autoencoder & autodecoder using python programming language.

Evaluate the performance of your design based on different sets of q, Q, B and plot the BER.