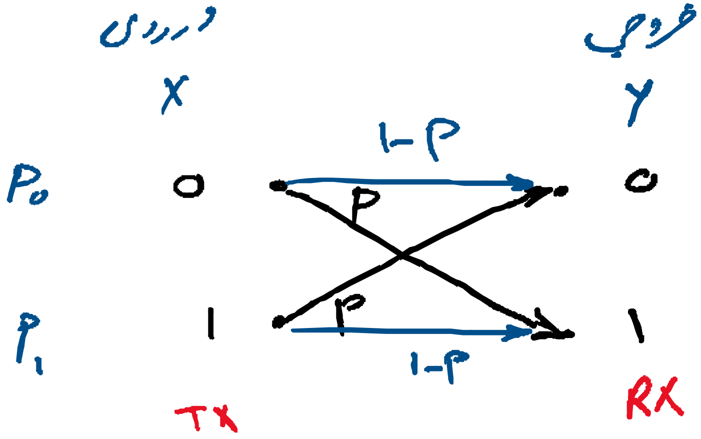
به نام زندگی

کدهای کانال Channel Codes

در این بخش می‌خواهیم با کمک یک مثال ساده نحوه‌ی کدینگ و تا حد زیادی تشخیص و تصحیح خطا را در کدهای کانال بررسی کنیم.

مثال: یک منبع اطلاعات داریم که پیام‌های «بله» و «خیر» را تولید می‌کند. می‌خواهیم این پیام‌ها را روی یک کانال BSC با احتمال گذار $P < \frac{1}{2}$ ارسال کنیم. راه‌حل‌های ممکن را هم بررسی می‌کنیم.



راه حل اول: پیام "بله" را با "۱" و پیام "خیر" را با "۰" نمایش می‌دهیم، با یک بار استفاده از کانال به سمت گیرنده می‌فرستیم.

- پیچیدگی سیستم کم است (یک بار استفاده از کانال)
- نرخ کد، بیشترین مقدار ممکن است

$$R \text{ نرخ کد } = \frac{\text{تعداد بیت‌های اطلاعات}}{\text{کل تعداد بیت‌های ارسالی}} = \frac{1}{1} = 1$$

$$R \leq 1$$

$$P_r(R_0|T_0) = 1-P \qquad P_r(R_1|T_1) = 1-P$$

$$P_r(R_1|T_0) = P \qquad P_r(R_0|T_1) = P$$

دریافت ۱ — ارسال ۰

$$P_e = P_1 \underbrace{P_r(R_0|T_1)}_{P} + P_0 \underbrace{P_r(R_1|T_0)}_{P} = P(P_0+P_1) = P$$

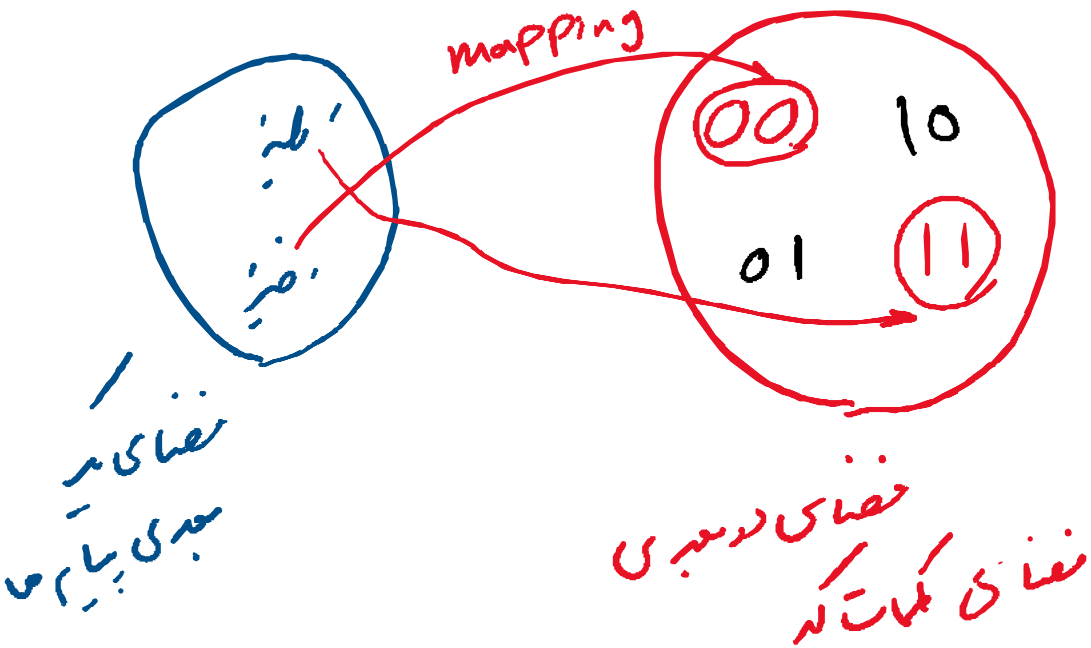
احتمال گذر کانال

احتمال خطا

احتمال اینکه ۱ بفرستیم و ۰ دریافت کنیم یا ۰ بفرستیم و ۱ دریافت کنیم

با توجه به اینکه احتمال خطا با احتمال گذر کانال یکسان است، هیچ دستاوردی در مورد تشخیص یا تصحیح خطا نداریم.

راه حل دوم : پیام 'بله' را با '11' و پیام 'خیر' را با '00' نمایش می دهیم و با دو بار استفاده از کانال به سمت گیرنده می فرستیم



* پیچیدگی نسبت به حالت قبل بیشتر شده است.

* نرخ کد نسبت به حالت اول کاهش پیدا کرده است.

$$R = \frac{1}{2} < 1$$

با این راه حل، قابلیت تشخیص یک بیت خطا را خواهیم داشت زیرا اگر '01' یا '10'
دریافت کنیم متوجه می شویم که خطایی در کانال رخ داده است.

سؤالی که مطرح می شود، این است که آیا این کد قابلیت تصحیح خطا را نیز دارد یا خیر.
فرض کنیم که در گیرنده '01' دریافت کرده ایم، آیا می توانیم حدس بزنیم که در فرستنده
'00' ارسال شده یا '11'؟ برای حدس زدن بر اساس احتمال کار می کنیم. یعنی احتمال های
زیر که احتمالات پسین یا پس از مشاهده هستند.

$$P_r(T_{00} | R_{01})$$

$$P_r(T_{11} | R_{01})$$

در روشهای بازیابی اطلاعات، بر اساس بیشترین احتمال
پسین یا بیشترین شباهت یا کمترین احتمال خطا،

بازیابی اطلاعات را انجام می‌دهیم

$$P_r(T_{00} \mid R_{01}) \sim (1-P)P$$

$$P_r(T_{11} \mid R_{01}) \sim P(1-P)$$

با توجه به اینکه احتمال‌های پیش از شده با هم یکسان هستند، نمی‌توانیم حدس بزنیم که '00' ارسال شده یا '11'

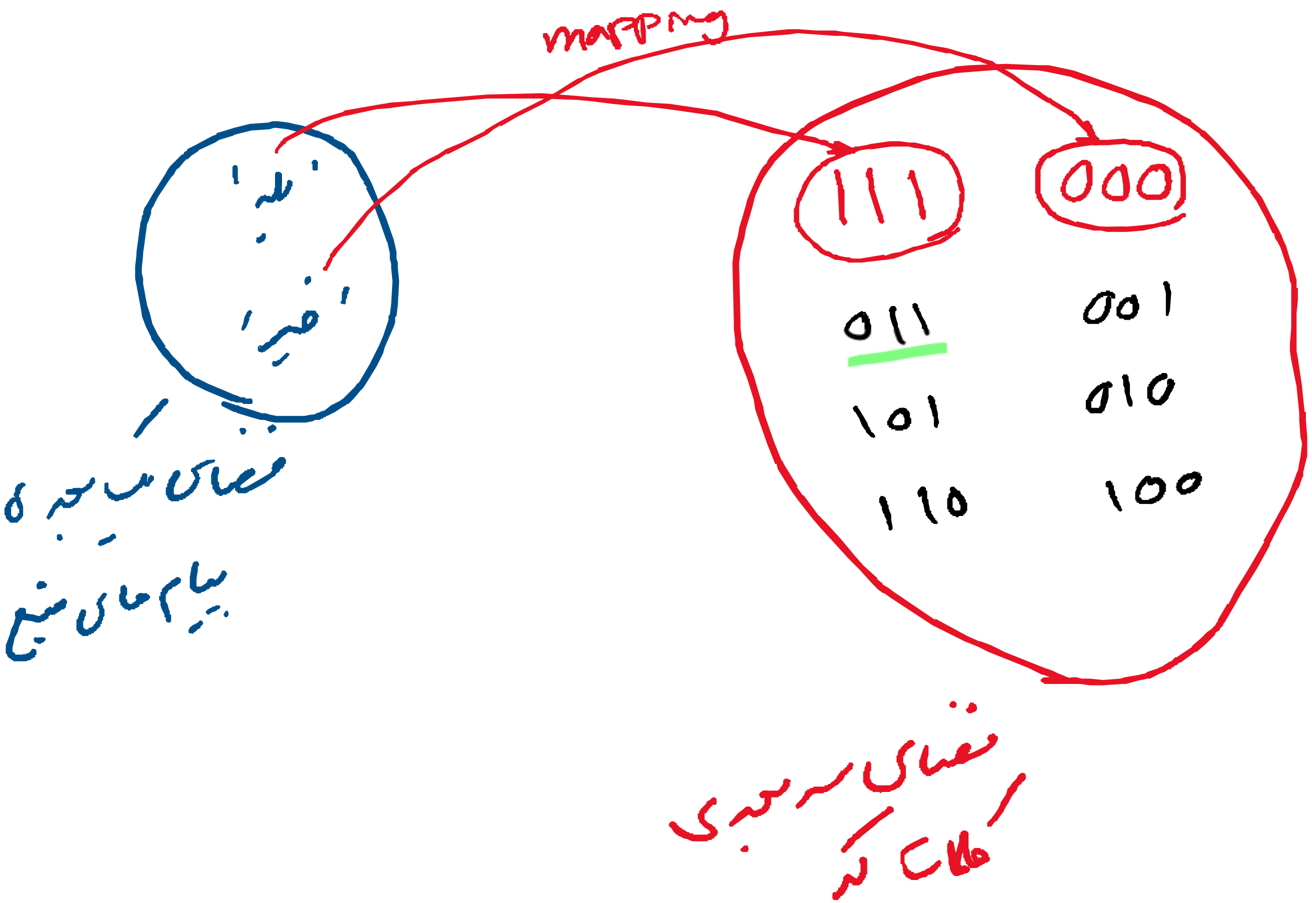
$\Longleftarrow$ قابلیت تصحیح خطا نداریم.

راه‌حل سوم: پیام «بله» را با '111' و پیام «خیر» را با '000' نمایش می‌دهیم و با سه بار استفاده از کانال به سمت گیرنده می‌فرستیم.

* پیچیدگی نسبت به حالت‌های قبلی افزایش پیدا کرده است.

* نرخ کد نسبت به حالت‌های قبلی کاهش یافته است.

$$R = \frac{1}{3} < \frac{1}{2} < 1$$



mapping
بله
خیر
فضای مجموعه پیام‌های منبع
111
000
011
001
101
010
110
100
فضای مجموعه کلمات کد

* با توجه به بیت تکرار استفاده شده برای ارسال اطلاعات، در این روش قابلیت تشخیص دو بیت خطا را داریم. زیرا هر برداری غیر از '000' و '111' دریافت کنیم، متوجه می‌شویم که خطایی در کانال رخ داده است.

* این کد قابلیت تصحیح یک بیت خطا را دارد. به طور مثال اگر فرض کنیم دیکدر ما '011' را دریافت کرده‌ایم با مقایسه‌ی احتمال‌های پسین به صورت زیر می‌توانیم حدس بزنیم که با احتمال بیشتری '111' فرستاده شده است.

$$P_r(T_{000} \mid R_{011}) \sim (1-P)PP = (1-P)P^2$$

$$P_r(T_{111} \mid R_{011}) \sim P(1-P)(1-P) = (1-P)^2 P$$

$$\xrightarrow{P<1/2} P_r(T_{111} \mid R_{011}) > P_r(T_{000} \mid R_{011})$$

در این مثال، هرچه طول کد را زیادتر کنیم (مثلاً برای 'بله' $\underbrace{11\ldots1}_{n\text{ بار}}$ و برای 'خیر' $\underbrace{00\ldots0}_{n\text{ بار}}$ را ارسال کنیم) قابلیت تشخیص و تصحیح خطا بالاتر می‌رود ولی در عوض پیچیدگی افزایش می‌یابد و نرخ کد نیز کاهش پیدا می‌کند

$$R=\frac{1}{n}$$

بنابراین هدف در کدینگ کانال، این است که بتوانیم در بالاترین نرخ ممکن، کدی طراحی کنیم که کمترین احتمال خطا را در بازیابی داشته باشد و دارای پیچیدگی قابل قبول برای پیاده‌سازی باشد.

* توجه به این نکته لازم است که در بازیابی وجود دارد اما چون

براساس بیشترین احتمال پسین یا بیشترین شباهت بازیابی را انجام می دهیم، احتمال خطای بازیابی

اطلاعات کمترین مقدار ممکن است در اساس پارامترهای کدینگ قابل کنترل است به

طوری که می توان خطا را به میزان دلخواه کم کرد. خطای بازیابی اطلاعات به صورت

تئوری قابل محاسبه و بررسی است.

با این مقدمه می خواهیم به بررسی کدهای کانال بپردازیم. کدهای کانال را به دو دسته کلی

کدهای تشخیص خطا و کدهای تصحیح خطا، دسته بندی می کنند.

- کدهای تشخیص خطا

در این دسته از کدها، دیکدر سعی می‌کند که رخداد خطا را تشخیص بدهد، و عملیاتی برای تصحیح خطا انجام نمی‌شود. در این گونه کدها معمولاً با استفاده از یک کانال فیدبک، در صورت تشخیص خطا، درخواست ارسال دوباره‌ی اطلاعات می‌شود. یکی از مثال‌های متداول از این گونه کدهای ARQ هستند (Automatic Request to send)

این کدها معمولاً در لایه‌ی فیزیکی استفاده نمی‌شوند.

* کدهای تصحیح خطا

Forward Error Correcting Codes (FEC)

Error Correcting Code (ECC)

در کدهای تصحیح خطا، پس از اینکه خطای رخ داده در حال تشخیص داده می‌شود، وارد مرحله‌ی تصحیح خطا می‌شوند و با کمک الگوریتم‌های مناسب دکدینگ به یافتن بیشترین شباهت یا بیشترین احتمال پسین (پس از مشاهده) یا ... تلاش

Maximum Likelihood (ML)

Maximum Aposteriori Prob. (MAP)

می‌کنند که خطای رخ داده را تا حد ممکن تصحیح کنند و بازیابی اطلاعات را انجام دهند.

این عملیات همواره با خطا همراه است که به آن خطای دیکدینگ می‌گوییم. این خطای دیکدینگ قابل کاهش است و الزاماً می‌توانند به میزان دلخواه کم باشد.

در این درس تمرکز ما بر روی کدهای FEC است و می‌خواهیم اصول کلی در مورد کدینگ کانال را بررسی کنیم و یک کد ساده‌ی کانال را معرفی کنیم.

کدهای FEC را نیز می‌توانیم به دو دسته‌ی کلی کدهای بلوکی و کدهای کانولوشنال (دارای حافظه) دسته‌بندی کنیم.

● کدهای بلوکی Block Codes

در کدهای بلوکی، اطلاعات ارسالی را به صورت بلوک هایی از پیامهای ورودی در می آورند (کلمه پیام) و به هر بلوک پیام، یک کلمه کد (بلوک کد) براساس الگوریتم کدینگ تخصیص می دهند. اگر طول بلوک پیام را برابر k در نظر بگیریم، در عملیات کدینگ به تعداد $n-k$ بیت یا سمبل redundant به بلوک پیام اضافه می شود و یک بلوک کد به طول n ساخته می شود.

$n > k$

کلمات پیام

بلوک پیام

redund. تعداد = $(n-k)$

کلمات کد

بلوک کد

$\underline{u} = (u_0, u_1, \ldots, u_{k-1})$ → **block Coding** → $\underline{v} = (v_0, v_1, \ldots, v_{n-1})$

$k =$ طول بلوک پیام

Message word

$n =$ طول بلوک کد

Code word

نرخ کد $R = \dfrac{k}{n} \leq 1$

با توجه به اینکه طول کلمات پیام برابر k در نظر گرفته شده است، در حالت باینری 2^k حالت ممکن برای کلمات پیام وجود دارد (در حالت کلی M-ary تعداد M^k حالت مختلف برای کلمات پیام وجود دارد)

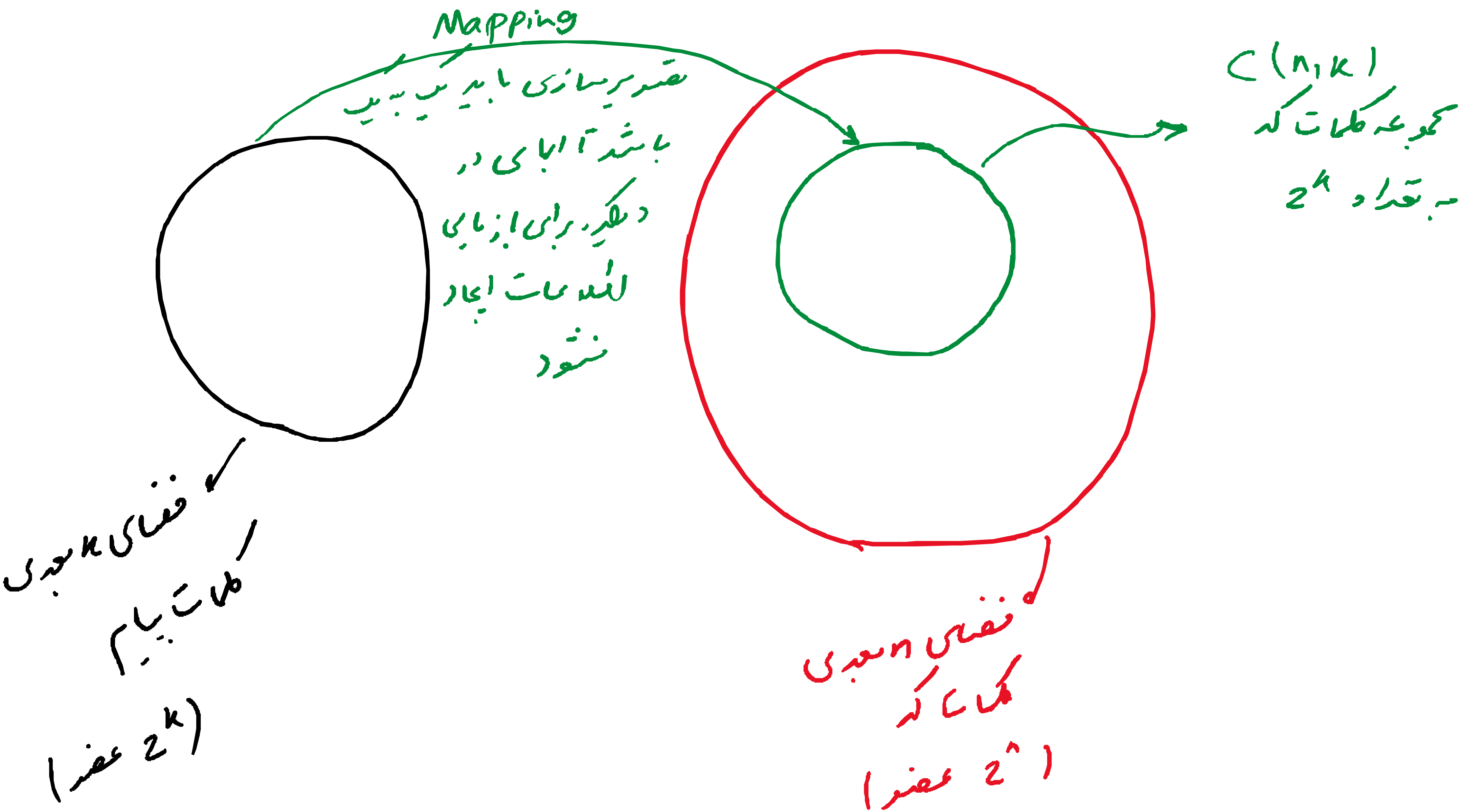
$$\underline{u} = (u_0, u_1, \dots, u_{k-1})$$

$M-1$ تا 0

(M تایی)

با توجه به اینکه به ازای هر کلمه ی پیام، لازم است یک کلمه کد متناظر داشته باشیم تعداد کلمات کد برابر تعداد کلمات پیام و برابر 2^k (در حالت باینری) یا M^k (در حالت M) است.

کد کامل : $C(n,k) = \{\underline{c}_0, \underline{c}_1, \dots, \underline{c}_{2^k-1}\}$


Mapping
تصویرسازی باید یک به یک باشد تا ابهامی در دیکد، برای بازیابی کلمات ایجاد نشود
C(n,k)
مجموعه کلمات کد به تعداد 2^k
فضای k بعدی کلمات پیام (2^k عضو)
فضای n بعدی کلمات کد (2^n عضو)

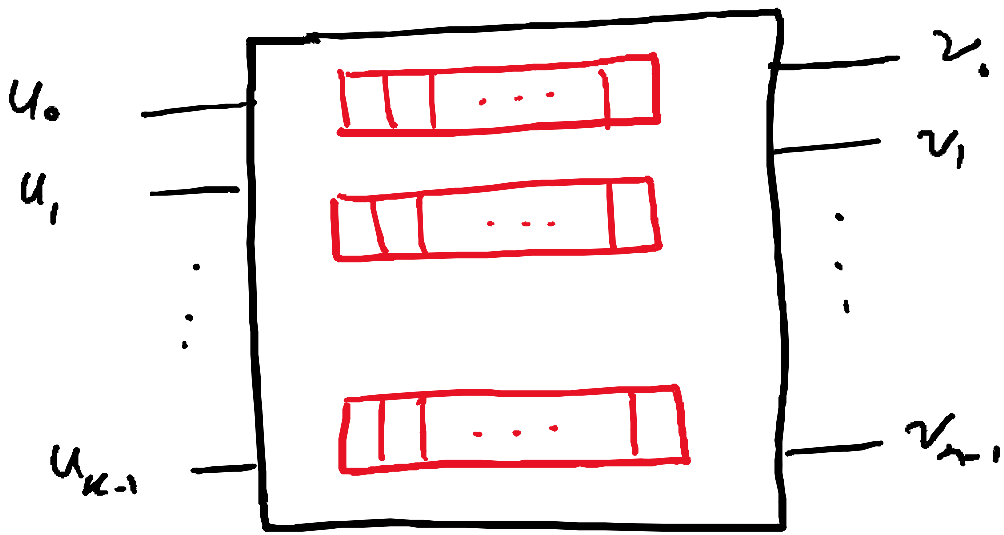
بنابراین آنکدینگ کد تنها معادل انتخاب 2^k کلمه کد مناسب از بین 2^n حالت ممکن در فضای n بعدی است.

- نشان داده می‌شود که در یک کد بلوکی $C(n,k)$ کلمات کد یک زیرفضای k بعدی روی فضای n بعدی تشکیل می‌دهند.

- کدهای کانولوشن

Convolutional Codes

در کدهای کانولوشنال برای تولید کلمات کد (بیت‌های redundant) از عناصر حافظه نیز استفاده می‌شود.



نحوه‌ی اتصالات عناصر حافظه (شیفت رجیسترها) براساس چند جمله‌ای خاص انجام می‌شود و بیت‌های خروجی براساس این اتصالات و وابسته به بیت‌های ورودی و بیت‌های قبلی (موجود در حافظه) تشکیل می‌شوند

$$R = \frac{k}{n}$$

m : طول بزرگترین عنصر حافظه (شیفت رجیستر)

در برخی متون دیگر (عمق کد) Code depth.

$$C(n, k, m)$$

عملیات دیکدینگ کدهای کانولوشنال تنها به دلیل وجود عناصر حافظه با کمک دیاگرام‌های
زمانی که به آنها دیاگرام‌های ترلیس گفته می‌شود، انجام می‌شوند. یکی از معروف‌ترین
الگوریتم‌های دیکدینگ کدهای کانولوشنال، الگوریتم ویتربی است.

Trellis Diagram / Viterbi Decoding

در این درس تمرکز ما بر روی کدهای بلوکی است که از نظر شهودی، درک بهتری از
کدینگ به دست می‌دهند و بدون نیاز به بیان مقدمات ریاضی، می‌توانیم عملکرد
آنها را بررسی کنیم.

برای شروع بحث، ابتدا مفاهیم و تعاریفی را مطرح می‌کنیم که در مورد تمامی کدهای کانال مورد استفاده هستند و به درک مفهوم کدینگ کمک می‌کنند.

* انواع خطاها در کانال

۱- خطاهای کاتوره‌ای

این خطا زمانی رخ می‌دهد که وضعیت کانال برای مدت طولانی، نامناسب باشد و اطلاعات ارسالی به صورت غیر قابل جبران دچار خرابی شوند. معمولاً کدهای کانال قابلیت تصحیح این نوع خطا را ندارند.

۳- خطاهای قرشای Burst

این خطاها زمانی رخ می‌دهد که وضعیت کانال برای مدت کوتاهی در حالت نامناسبی قرار بگیرد و باعث شود که بیت‌های اطلاعات در یک بازه‌ی کوتاه دچار خطا (پشت سرهم) شوند. الگوی این خطا را به صورت زیر نمایش می‌دهند.

الگوی خطای برست به طول l طول برست $= l$ (l عنصر از بردار $\underline{e}$)

$$\underline{e} = (0,0,\dots,0,\underbrace{1,x,x,\dots,x,1}_{l},0,0,\dots,0)$$

در حالت باینری $x \in (0,1)$, $P_r\{x=1\} >> P_r\{x=0\}$

در حالت کلی $x \in (0,1,..,M-1)$, $P_r\{x \neq 0\} >> P_r\{x=0\}$

همان طور که اشاره کردیم، در حالت کلی اگر $e_i \neq 0$ برقرار باشد به معنی این است که در مکان i ام خطایی رخ داده است که مقدار آن نیز می‌تواند مشخص شود.

در حالت باینری چون فقط دو حالت '0' و '1' را داریم، اگر $e_i = 1$ باشد به معنی این است که در مکان i ام خطایی رخ داده است.

(در این درس تمرکز روی حالت باینری است)

* خطای برست می‌تواند در ابتدا و انتهای الگوی خطا نیز گسترده باشد

$l_1 + l_2 = l$

خطای برست به طول l

(در ابتدا و انتهای الگوی خطا)

$$\underline{e} = (\underbrace{1, x, \dots, x, 1}_{l_2}, 0, 0, \dots, \underbrace{1, x, \dots, x, 1}_{l_1})$$

۳- خطای تصادفی Random

در این نوع خطا، بیت های خطا به صورت تصادفی در الگوی خطا پخش شده اند

$$\underline{e} = (0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, \ldots, 1)$$

در واقع توزیع خطا به صورت i.i.d است.

* کدهای کانال قابلیت تصحیح این نوع خطا را در محدوده ی عملکرد خود دارند.

* برخی کدهای کانال قابلیت تصحیح خطاهای خوشه ای را در محدوده عملکرد خود دارند.
به طور مثال کدهای RS (Reed - Solomon)

پارامتر دیگری که در مورد کدهای کانال قابل توجه است، کمترین فاصله‌ی همینگ بین کلمات کد است که آن را با d_{min} نمایش می‌دهیم. d_{min} معیاری از قابلیت تشخیص و تصحیح خطای کد را به دست می‌دهد. برای اینکه d_{min} را تعریف کنیم، لازم است و ابتدا مفهوم فاصله همینگ را معرفی کنیم.

* فاصله همینگ Hamming Distance

برای دو بردار هم طول x و y فاصله‌ی همینگ عبارت است از تعداد مکان‌های از این دو بردار که با هم متفاوت هستند.

اگر

$$\underline{x} = (x_0, x_1, \dots, x_{n-1})$$

$$\underline{y} = (y_0, y_1, \dots, y_{n-1})$$

با بیان ریاضی، فاصله همینگ به صورت زیر تعریف می‌شود

$$d_H(\underline{x}, \underline{y}) = \sum_{i=0}^{n-1} d(x_i, y_i)$$

که در آن

$$d(x_i, y_i) = \begin{cases} 1 & x_i \neq y_i \\ 0 & x_i = y_i \end{cases}$$

مثال: برای بردارهای زیر فاصله همینگ را بدست بیاورید:

$$\underline{x}_1 = (1, 0, 0, 1, 0, 1, 1)$$

$$\underline{y}_1 = (1, 1, 1, 1, 0, 0, 1)$$

$$\longrightarrow d_H(\underline{x}_1, \underline{y}_1) = 3$$

$$\underline{x}_2 = (1, 3, 0, 5, 2, 0, 0)$$

$$\underline{y}_2 = (0, 1, 0, 3, 2, 1, 3)$$

$$\longrightarrow d_H(\underline{x}_2, \underline{y}_2) = 5$$

قضیه: فاصله همینگ دو بردار x و y یک تابع متریک (Metric function) یا تابع فاصله (Distance function) است.

یادآوری: تابعی را متریک می‌گوییم که ویژگی های زیر را داشته باشد.

1) $d(x,y) \geq 0$ غیر منفی بودن

2) $d(x,y) = 0 \iff x = y$ انعکاسی بودن (reflective)

3) $\forall x,y;\quad d(x,y) = d(y,x)$ خاصیت تقارن (symmetry)

4) $\forall x,y,z;\quad d(x,y) + d(y,z) \geq d(x,z)$ نامساوی مثلثی (triangle inequality)

تساوی زمانی برقرار است که نقاط x، y و z در یک راستا باشند