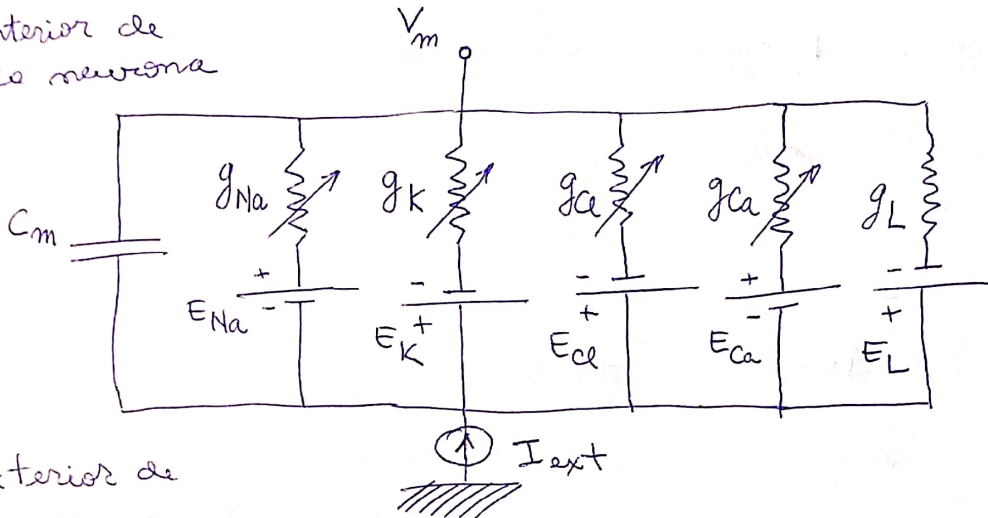


# 1. Dinámica Neuronal.

Una porción de membrana tiene canales de  $\left. \begin{array}{l} \text{Na} \\ \text{K} \\ \text{Cl} \\ \text{Ca} \end{array} \right\}$

- Grafique el circuito equivalente:

Interior de la neurona



Exterior de la neurona.

Siendo  $V_m$  el potencial de membrana de la neurona,

$C_m$  su capacitancia y  $I_{ext}$  la corriente externa aplicada,

$g_i$  la conductancia (que puede ser variable) del ion  $i \in \{Na, K, Cl, Ca\}$

$g_L$  la conductancia de pérdida o de "Leak", y

$E_i$  el potencial de equilibrio del ion  $i$  o de "Leak".

Siendo  $E_{Na} \approx 50 \text{ mV}$

$E_K \approx -80 \text{ mV}$

$E_{Cl} \approx -60 \text{ mV}$

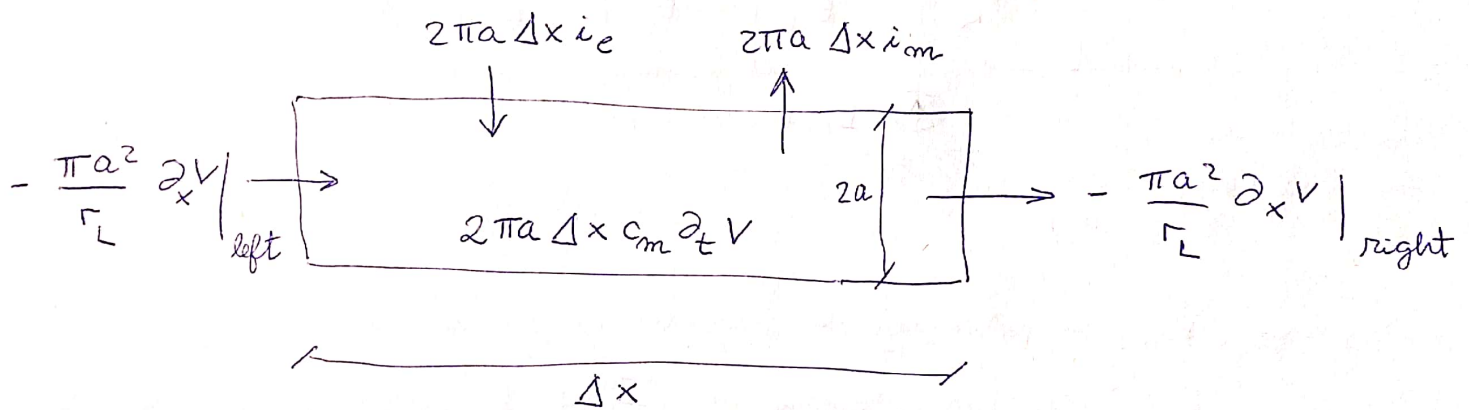
$E_{Ca} \approx 150 \text{ mV}$

y suponiendo que  $E_L$  es negativo, (por ejemplo  $E_L \approx -50 \text{ mV}$ ).

• Describa cuantitativamente el balance de corrientes en un cilindro de membrana de longitud  $\Delta x$  y radio  $a$ , incluyendo tanto las corrientes a través de la membrana como las axiales.

Estructura espacial de una neurona y

la Ecuación del Cable:



Siendo  $r_L$  la resistividad intracelular,

resultando en una resistencia:  $R_L = r_L \frac{\Delta x}{\pi a^2}$ ,

con  $\pi a^2$  la sección transversal del cilindro.

Siendo  $i_m$  la contribución de corrientes iónicas y

sinápticas, por unidad de área del cilindro, y

$i_e$  la corriente externa aplicada por unidad de área.

El área del cilindro es  $2\pi a \Delta x$ .

Por último, el balance de corrientes longitudinales, por izquierda y por derecha, puede expresarse como una derivada:

$$\frac{1}{\Delta x} \left[ \left( \frac{\pi a^2}{r_L} \partial_x V \right) \Big|_{\text{right}} - \left( \frac{\pi a^2}{r_L} \partial_x V \right) \Big|_{\text{left}} \right] \underset{\uparrow}{\approx} \partial_x \left( \frac{\pi a^2}{r_L} \partial_x V \right).$$

Si  $\Delta x \rightarrow 0$ .

Por lo tanto, se obtiene la ecuación del cable:

$$c_m \partial_t V = \frac{1}{2a r_L} \partial_x \left( a^2 \partial_x V \right) - i_m + i_e$$

Donde  $c_m$  es la capacitancia por unidad de área del cilindro, por eso,  $2\pi a \Delta x c_m \partial_t V$  era la tasa con la que se acumula carga en esta sección de membrana.

## 2. Modelos de Memoria

• ¿Bajo qué condiciones está garantizado que la dinámica de un modelo neuronal converja a un punto fijo?

### Modelo de Hopfield Determinista:

Para que un modelo de Hopfield converja a un punto fijo, debemos tener:

Patrones de entrenamiento aleatorios  $\xi_i^\mu$  ( $i = 1, \dots, N$ ,  $\mu = 1, \dots, p$ )

con  $N$  número de dimensiones de los patrones y

$p$  cantidad de patrones.

Estos patrones adoptan valores  $\pm 1$  con igual probabilidad, resultando el promedio  $\langle \xi_i^\mu \rangle = 0$ .

Los patrones deben ser independientes entre sí, es decir,

$\xi_i^\mu$  y  $\xi_{i'}^{\mu'}$  son independientes, con  $i \neq i'$ ,  $\mu \neq \mu'$ ,

si pensamos a  $\xi_i^\mu$  y  $\xi_{i'}^{\mu'}$  como variables aleatorias.

Por último, la regla de actualización del modelo de Hopfield

$$S_i = \text{sgn} \left( \sum_{j=1}^N J_{ij} S_j \right), \text{ siendo } \text{sgn}(x) = \begin{cases} 1, & x \geq 0 \\ -1, & x < 0 \end{cases}$$

debe ser "asincrónica" (también llamada "secuencial"), con

$$S_i \text{ la salida de la red y } J_{ij} = \frac{1}{N} \sum_{\mu=1}^p \xi_i^\mu \xi_j^\mu,$$



fijando además los términos de autoacoplamiento

$$J_{ii} = 0 \quad \text{con } i = 1, \dots, N$$

Resultando así una matriz de conexiones  $J$  simétrica.

- ¿Cómo se define una dinámica estocástica?
- ¿Qué parámetro controla el grado de estocasticidad?

### Modelo de Hopfield Estocástico:

Este modelo está inspirado en el modelo de Ising de mecánica estadística de espines interactuantes.

Se reemplaza la regla de actualización determinista por una probabilística; los componentes  $s_i(t+1)$  de la salida de la red, en la iteración  $t+1$ , adoptan valores

$$s_i(t+1) = \begin{cases} 1 & \text{con probabilidad } f_{\beta}[h_i(t)] \\ -1 & \text{con probabilidad } 1 - f_{\beta}[h_i(t)] \end{cases},$$

el factor de

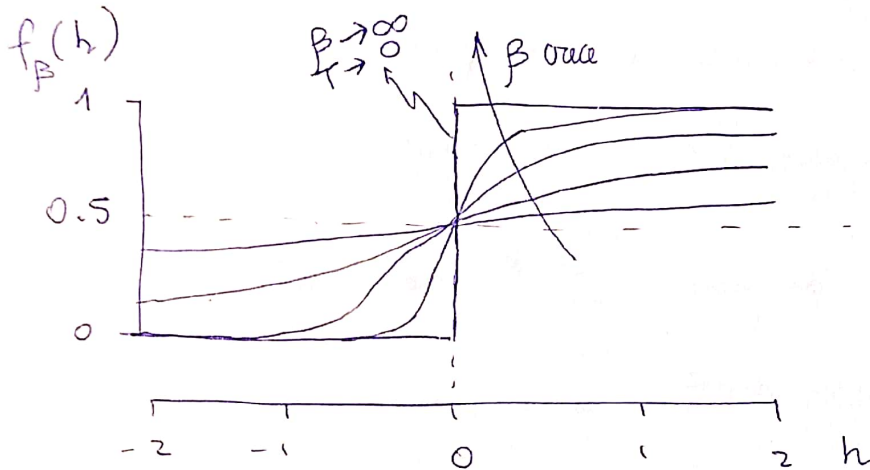
donde  $f_{\beta}[h_i(t)]$  es ~~la~~ probabilidad de Boltzmann:

$$f_{\beta}[h_i(t)] = \frac{1}{1 + \exp[-2\beta h_i(t)]},$$

siendo  $h_i(t) = \sum_{j=1}^N J_{ij} s_j(t)$  y definiendo el parámetro  $\beta = \frac{1}{T}$ , donde  $T$  es el análogo a la

temperatura termodinámica en el modelo de Ising, con constante de Boltzmann  $k_B = 1$ .

Denomamos a  $T$  "pseudo-temperatura" y es el parámetro que controla el grado de estocasticidad de este modelo.

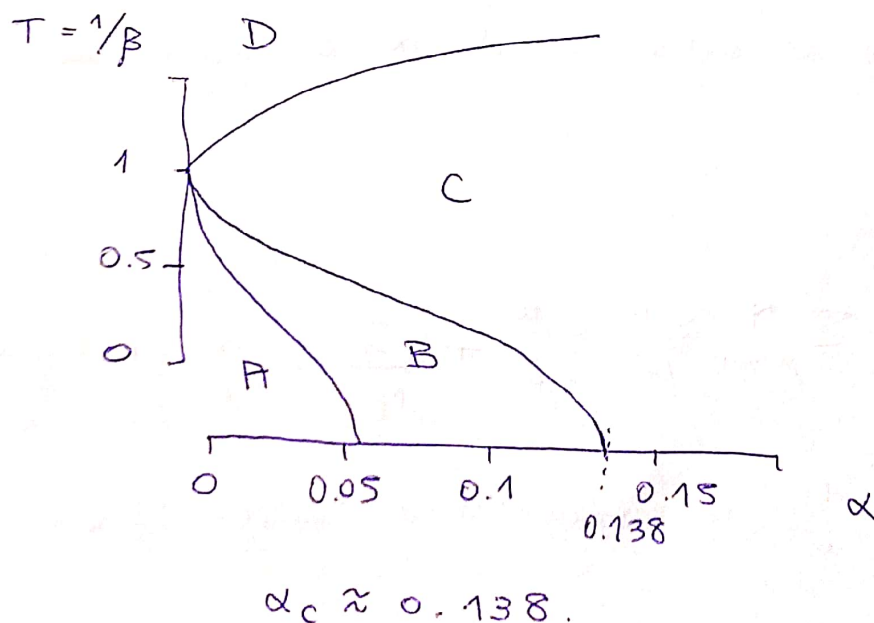


Si  $\beta \rightarrow \infty$  ( $T \rightarrow 0$ ) recuperamos la dinámica determinista

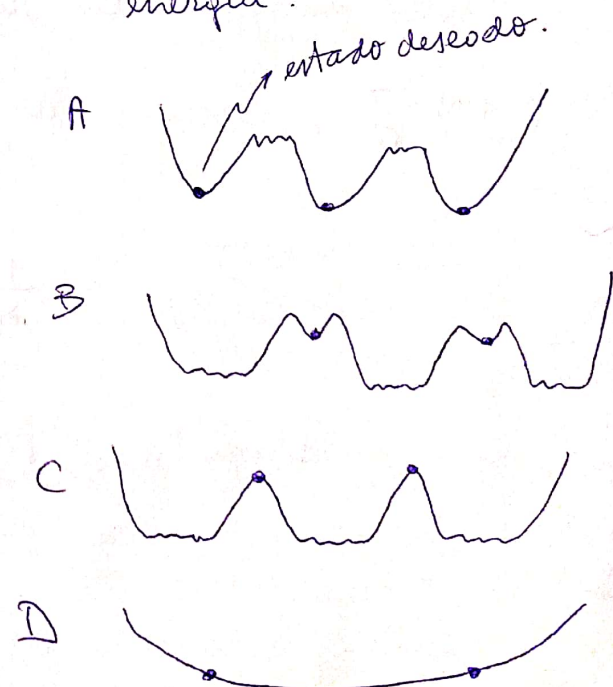
y, si  $\beta \rightarrow 0$  ( $T \rightarrow \infty$ ) tenemos actualización completamente aleatoria

con  $s_i(t+1) = \pm 1$  con igual probabilidad.

• Describe el diagrama de fases



Esquema de perfiles de energía:



En región C tenemos estados estables no nulos, pero no correlacionados con los estados deseados  $\xi_i^\mu$ .

Si se aumenta  $T$ , se alcanza la región D, donde se degeneran los estados estables y se hacen de media nula  $\langle s_i \rangle = 0$ .

En regiones A y B recuperamos los patrones de entrenamiento  $\xi_i^\mu$ , pero también tenemos estados espurios.

En A, los estados deseados  $\xi_i^\mu$  son mínimos globales de energía, mientras que en B los espurios son de menor energía.

• ¿Cómo se debe modificar la regla de aprendizaje para que el sistema pueda almacenar secuencias de patrones en vez de puntos fijos?

Se modifica la regla de entrenamiento de la matriz de pesos  $J$ , resultando

$$J_{ij} = \frac{1}{N} \sum_{\mu=1}^N \xi_i^\mu \xi_j^\mu + \frac{\lambda}{N} \sum_{\mu=1}^N \xi_i^{\mu+1} \xi_j^\mu,$$

con condición inicial  $\xi_i^1$  y parámetro de "delay"  $\lambda > 1$ .

3. Teoría de Aprendizaje:

- ¿Qué se minimiza en el algoritmo de backpropagation?

Se minimiza una función de costo  ~~$E(\vec{w}) = \frac{1}{2} \sum_{\mu, i} (y_i^{\mu} - o_i^{\mu})^2$~~   $E[\vec{w}]$ ,

que depende paramétricamente de los parámetros de la red  $\{\vec{w}\}$

y es función de los salidos deseados o "verdaderas" o "ground-truth"  $\sum_i^{\mu} y_i^{\mu}$  y de los entrees de la red  $\sum_k^{\mu} x_k^{\mu}$ .

En particular, en clase usamos el error cuadrático medio:

$$E[\vec{w}] = \frac{1}{2} \sum_{\mu, i} \left[ \sum_j^{\mu} x_j^{\mu} - o_i^{\mu} \right]^2,$$

donde  $o_i^{\mu}$  son las salidas obtenidas o "outputs" de la red, asociados a los "inputs"  $\sum_k^{\mu} x_k^{\mu}$ .



• Describir la diferencia entre el error de entrenamiento y error de generalización. ¿Qué factores los controlan?

Es deseable que entrenemos un modelo para que sea capaz de devolver outputs apropiados frente a entradas nunca antes vistas.

Por este motivo, nos interesa que el modelo que utilizemos sea capaz de "generalizar" bien ante inputs novedosos.

Por eso, se suele dividir el conjunto de datos disponibles durante el desarrollo de un modelo en datos de entrenamiento y datos de generalización

$$\{\text{datos disponibles}\} = \{\text{entrenamiento}\} \cup \{\text{generalización}\}$$

$$\text{con } \{\text{entrenamiento}\} \cap \{\text{generalización}\} = \emptyset$$

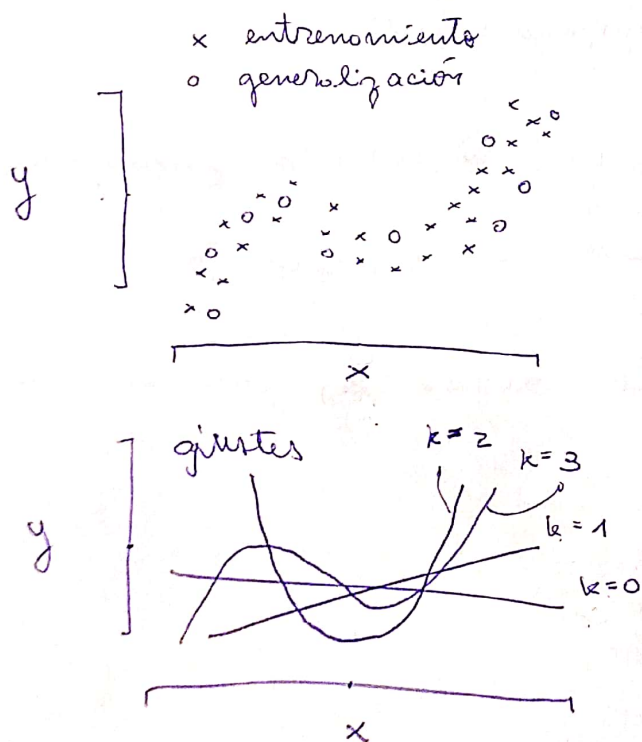
Por lo tanto, se entrena un modelo con los datos de entrenamiento y se estima su performance frente a nuevos datos con el conjunto de generalización.

El error de entrenamiento y el error de generalización se calculan utilizando los respectivos conjuntos de datos.

Cuanto mayor sea la cantidad de datos de entrenamiento, menor serán los errores de entrenamiento y de generalización.

También, los errores dependen de la "complejidad" del modelo utilizado. Utilizar modelos muy complejos o con muchos parámetros de ajuste puede llevarnos a un "overfitting" si no tenemos datos de entrenamiento suficientes. Al contrario, si usamos modelos muy simples, podemos también tener problemas por la generalización del modelo.

• ¿Qué acciones se pueden tomar para optimizar el error de generalización?



Supongamos que queremos hacer un ajuste polinomial de la forma

$$y = a_k x^k + a_{k-1} x^{k-1} + \dots + a_0$$

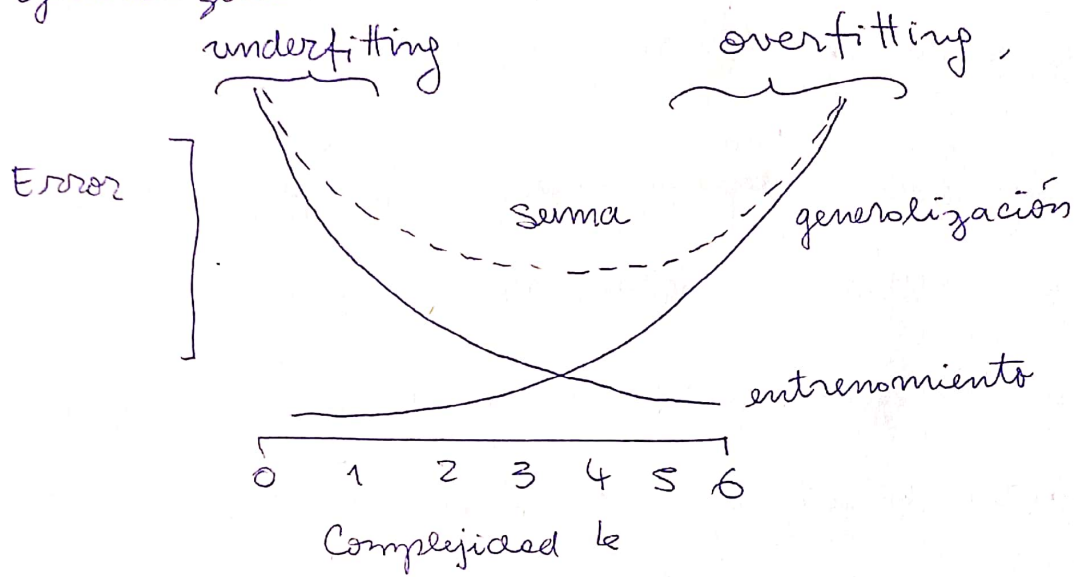
con los datos que tenemos.

~~Supongamos que los datos tienen media  $x=0$  y  $y=0$  para más simplicidad.~~

Siendo  $k$ , el grado del polinomio, la complejidad del modelo.  
 $k \in \mathbb{N}_0$

Entonces, tendremos errores de entrenamiento y de

generalización de la forma:



Este fenómeno se conoce como dilema sesgo - varianza (Bias - variance trade off) y se combate buscando una complejidad de modelo  $k$  que minimice la suma del error de generalización y de entrenamiento.

Si solamente queremos reducir el error de generalización, podríamos reducir la complejidad del modelo o aumentar la cantidad de datos disponibles para entrenar.

#### 4. Código Neuronal:

Símbolos  $\{a, b, c, d\}$   
Conjunto  $X$  de estímulos

Probabilidad  $p$ :  $\frac{0.5}{a}, \frac{0.25}{b}, \frac{0.125}{c \text{ o } d}$

- Entropía de salida con máxima eficiencia:

$$H = - \sum_{x \in X} p(x) \ln(x)$$

$$= 1.213 //$$

- Un código ineficiente tendrá menor entropía.

- Código neuronal  $\left\{ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} \right\}$  disparos  $\Rightarrow$  conjunto de respuestas  
 $R = \{1, 2, 3, 4\}$

Costo energético  $E(r) = \alpha r$  donde  $r \in R$  es el número de disparos y  $\alpha$  es una constante.

Quiero minimizar el costo energético y codificar  $X$ .

$\Rightarrow$  Asocio

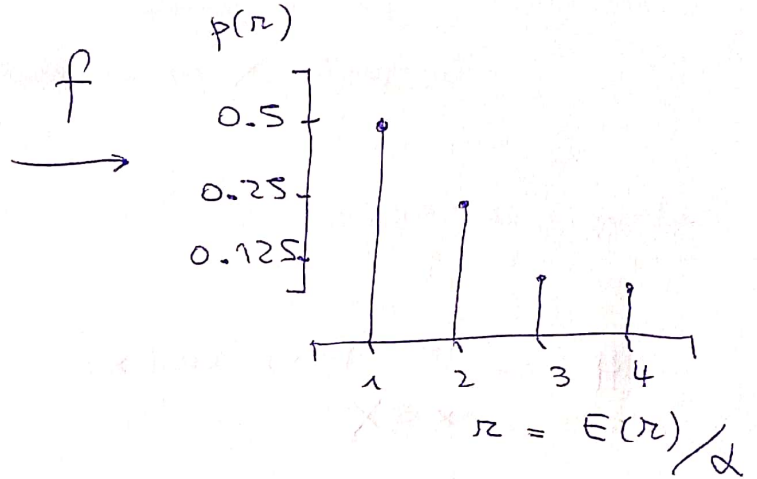
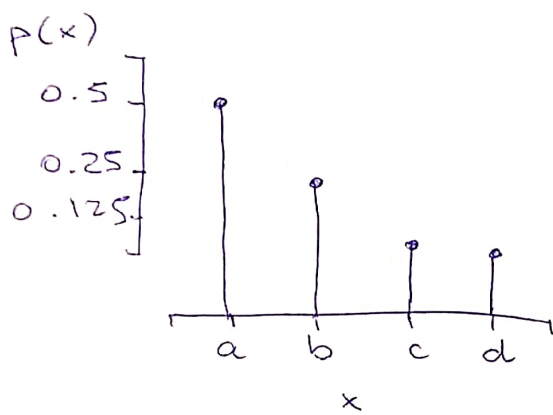
$X$	$R$
a	1
b	2
c	3
d	4

y esta es la relación entre estímulo  $e \in X$  y respuesta  $r \in R$  que minimizo el costo energético.

$$R = f(X)$$



• Analice el costo por mensaje y la probabilidad de ocurrencia y discuta en qué sentido es más uniforme la representación de salida.



Este código no modifica la distribución de los respuestas.

Tenemos un costo energético medio

$$\begin{aligned} \langle E(r) \rangle &= \sum_{r \in R} p(r) \cdot E(r) \\ &= 1.875 \cdot \alpha \end{aligned}$$

Si en lugar de optimizar por costo energético, quisiéramos optimizar la uniformidad de la distribución de respuestas, tendríamos que usar un código  $g$  tal que

$$p(x) \xrightarrow{g} p(r) = k \text{ constante, con } k = \frac{1}{4}.$$

$$\text{con } r = g(x) = \frac{1}{k} \sum_{x'=a}^x p(x').$$

así, el mapeo  $g$  sigue a la distribución acumulada de  $X$ .