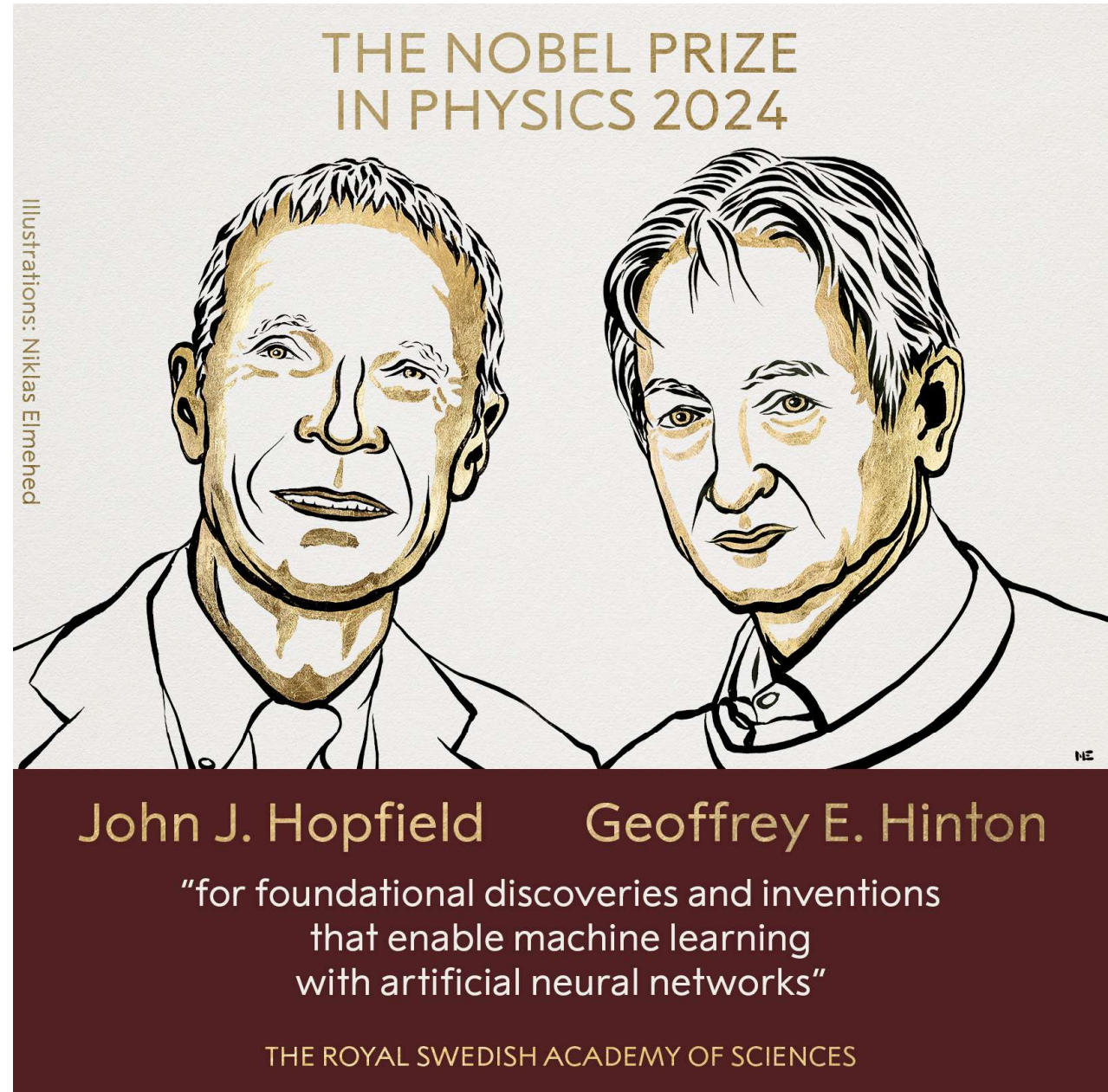


Física Computacional – Ejercicio 2:

Modelo de Hopfield de red neuronal.

Introducción a Métodos Monte Carlo.

- Una **red de Hopfield** es **red neuronal artificial recurrente** con la **propiedad de memoria asociativa**, compuesta por **neuronas binarias**.
- El modelo de Hopfield fue propuesto por John Hopfield **inspirándose directamente en los modelos de espines** de la **física estadística**, como el **modelo de Ising**.
- Hopfield formalizó una función de **energía** análoga a la energía de sistemas magnéticos, permitiendo **describir la dinámica de la red como un proceso de minimización energética**. Este paralelismo permitió aplicar herramientas de la mecánica estadística, como la teoría del equilibrio térmico y las transiciones de fase, al análisis de redes neuronales.
- Investigadores como Amit y Sompolinsky usaron métodos de física estadística para caracterizar el comportamiento colectivo de estas redes, conectando la neurociencia computacional con la teoría de sistemas desordenados, como los **vidrios de espines**.
- **La red de Hopfield es un precursor de las redes neuronales artificiales** y el aprendizaje automático al introducir la idea de una memoria distribuida y dinámica basada en minimización de la energía. Su arquitectura influyó en modelos posteriores como las Boltzmann Machines y las redes neuronales profundas.



Física Estadística y Métodos Monte Carlo

- En el contexto de la **Mecánica Estadística de equilibrio** se deriva que, en un sistema en equilibrio termodinámico con una temperatura T , número de partículas N y volumen V , la probabilidad de encontrar a las N partículas en un estado $X = [x[0], x[1], \dots, x[N-1]]$ está dada por

$$P_N = Z^{-1} e^{-\beta H_N(X)}, \quad Z = \int dX e^{-\beta H_N(X)}$$

donde $\beta = 1/(k_B T)$ y k_B es la constante de Boltzmann. Z es la **función de partición** y $H_N(X)$ es el hamiltoniano que define la dinámica microscópica correspondiente a las N partículas.

- Conocida $H_N(X)$, podemos calcular el **valor esperado** de cualquier variable (que coincidirá con el **valor observado** en un experimento):

$$\langle A \rangle = Z^{-1} \int dX e^{-\beta H_N(X)} A_N(X)$$

La Mecánica Estadística conecta las ecuaciones del movimiento microscópicas de un sistema de partículas con sus propiedades termodinámicas macroscópicas. Sin embargo, en la práctica, sólo podemos realizar de forma analítica alguna de esas integrales N dimensionales en unos pocos casos de interés. De esta forma debemos acudir a **métodos numéricos** para conocer el comportamiento del sistema.

- La estrategia más natural que se aplica es usar la ley de los grandes números. Ésta nos dice que, **independientemente** de la distribución de probabilidad $P_N(X)$, **el valor esperado de cualquier observable puede ser hallado mediante sumas parciales de M realizaciones concretas de los mismos**, esto es

$$\langle A \rangle = M^{-1} \sum_{i=1}^M A(X_i) + O(M^{-\frac{1}{2}})$$

donde X_i deben seguir la distribución $P_N(X)$ y han de ser **estadísticamente independientes**.

Física Estadística y Métodos Monte Carlo

- Para generar esas configuraciones podemos utilizar la **técnica del rechazo**. Ésta, se basa en el hecho obvio de que si $f(x)$ es una cierta densidad de probabilidad, entonces si $f(y) = 2f(x)$, entonces y es dos veces más probable que x . Por tanto, para reconstruir $f(x)$ basta con **encontrar la frecuencia relativa** entre eventos. Uno podría por tanto plantear el siguiente algoritmo:
 1. Generar un valor x aleatoriamente y uniformemente distribuido en su dominio de definición.
 2. Generar un número aleatorio uniforme $u \in [0,1]$.
 3. Si $u < \frac{f(x)}{\max_y f(y)}$, entonces x se acepta como representante de la distribución $f(x)$.
 4. Ir a (1).
- Si intentamos utilizar este sencillo algoritmo para generar configuraciones que sigan la distribución $P_N(X)$, nos pasaríamos **todo el tiempo rechazando configuraciones**. Esto es debido a que cuando N es grande (como es el caso en los sistemas que se quieren simular), el hamiltoniano es una variable extensiva en N y la distribución de probabilidad es
$$P_N \approx e^{-N(e(X) - \langle e \rangle)},$$
lo que la hace **extremadamente picuda alrededor de sus valores de equilibrio termodinámico**.
- De esta forma si aplicamos el **algoritmo de rechazo** tendríamos dos efectos adversos: **(1)** las configuraciones aleatorias son, normalmente, mucho más numerosas que las demás y por lo tanto, la probabilidad de escoger al azar una configuración aleatoria es prácticamente uno y **(2)** las configuraciones aleatorias son las que tienen más energía (las correspondientes en un gas a temperatura infinita) por lo que $P_N(X) \approx 0$ y **rechazaríamos la inmensa mayoría de las configuraciones generadas**.

Física Estadística y Métodos Monte Carlo

- Para solucionar este problema, [Metropolis, Rosenbluth, Rosenbluth, Teller y Teller](#) desarrollaron en 1953 un algoritmo que abandona la idea original de generar configuraciones estadísticamente independientes. En este caso, las configuraciones son construidas a través de **cadenas de Markov**. Una cadena de Markov de eventos sucesivos X_0, X_1, \dots, X_{N-1} tiene como probabilidad:

$$P_N(X_0, X_1, \dots, X_{N-1}) = P_0(X_0)T(X_0 \rightarrow X_1)T(X_1 \rightarrow X_2) \dots T(X_{N-2} \rightarrow X_{N-1})$$

donde $T(X \rightarrow Y)$ es la probabilidad de transición del estado X al estado Y (obviamente $\sum_Y T(X \rightarrow Y) = 1$). En el contexto del modelo de Hopfield, $T(X \rightarrow Y)$ se conoce también como **función de transferencia**.

- Si $g(X, t)$ es la probabilidad de que en el paso t de la cadena de Markov nos encontremos en el estado X , entonces podemos derivar la **ecuación maestra**:

$$\begin{aligned} g(X, t+1) &= \sum_{X'} g(X', t)T(X' \rightarrow X) \\ &= \sum_{X' \neq X} g(X', t)T(X' \rightarrow X) + g(X, t)T(X \rightarrow X) \\ &= g(X, t) + \sum_{X' \neq X} [g(X', t)T(X' \rightarrow X) - g(X, t)T(X \rightarrow X')] \end{aligned}$$

- Encontramos una solución estacionaria al proceso de Markov imponiendo

$$g(X')T(X' \rightarrow X) = g(X)T(X \rightarrow X')$$

donde $g(X)$ es la probabilidad estacionaria de que el sistema se encuentre en el estado X . Esta condición suficiente se denomina **condición de balance detallado**.

Física Estadística y Métodos Monte Carlo

- En este contexto, la estrategia a seguir para generar estados X cuya distribución estacionaria sea $P_N(X)$ supuestamente conocida es la siguiente:

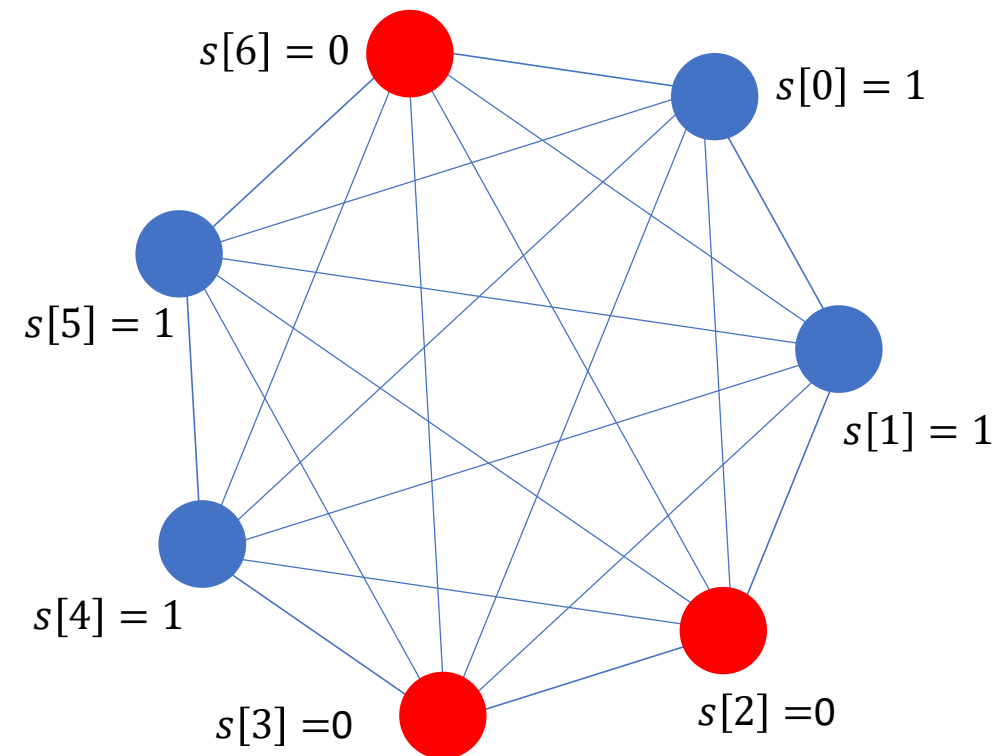
1. Construimos una $T(X' \rightarrow X)$ a partir de la condición de balance detallado. Dada $P_N(X) \approx \exp[-\beta H_N(X)]$, la condición de balance detallado es

$$T(X' \rightarrow X) = e^{-\beta[H_N(X) - H_N(X')]}$$

Hay infinitas funciones $T(X' \rightarrow X)$ que cumplen esta ecuación. Un ejemplo de particular relevancia es la utilizada por Metropolis et al.: $T(X' \rightarrow X) = \min(1, e^{-\beta[E(X') - E(X)]})$ para simular el modelo de espines de Ising. En nuestro caso, consideraremos la función de transferencia $T(X' \rightarrow X) = \frac{1}{2}(1 + \tanh(-\beta(E(X') - E(X))))$.

2. Dar un estado inicial $X = X_0$.
 3. Elegir un estado X' y calcular $T(X' \rightarrow X)$. Generar un número aleatorio uniforme $u \in [0,1]$. Si $u < T(X' \rightarrow X)$, entonces mover el sistema al estado X' .
 4. Ir a (2).
- Esta dinámica de estados que nos hemos inventado **recorre todos los estados accesibles por el sistema** y, para tiempos largos, **tiende a muestrear los estados con probabilidad $P_N(X)$** . Obviamente, si elegimos al azar X' tendríamos el mismo problema que nos encontrábamos con el algoritmo de rechazo. El truco aquí es que **podemos elegir un X' que sea que una pequeña modificación de X** . De esta forma si en algún momento de la cadena de Markov las X son las más probables según $P_N(X)$, los siguientes estados generados seguirán estando concentrados alrededor de ese máximo de probabilidad.

Modelo de Hopfield de red neuronal



- Cada elemento del sistema representa una **neurona binaria** cuyo estado viene dado por $s[i] = 0,1$, donde 1 indica que la neurona está disparando, y 0 que no.
- La red de Hopfield asume interacciones o **pesos sinápticos** w_{ij} de largo alcance, y en particular una **red totalmente conectada**, excluyendo las autoconexiones: $w_{ii} = 0$.
- Los pesos pueden ser tanto positivos como negativos. La matriz de pesos es simétrica: $w_{ij} = w_{ji}$.

- El **Hamiltoniano del sistema** es

$$H(s) = -\frac{1}{2} \sum_{i,j} w_{ij} s_i s_j + \sum_i \theta_i s_i$$

donde los umbrales θ_i vienen dados por $\theta_i = \frac{1}{2} \sum_j w_{ij}$.

- Podemos reescribir el Hamiltoniano como

$$H(s) = -\frac{1}{2} \sum_i (h_i - 2\theta_i) s_i, \quad h_i = \sum_j w_{ij} s_j$$

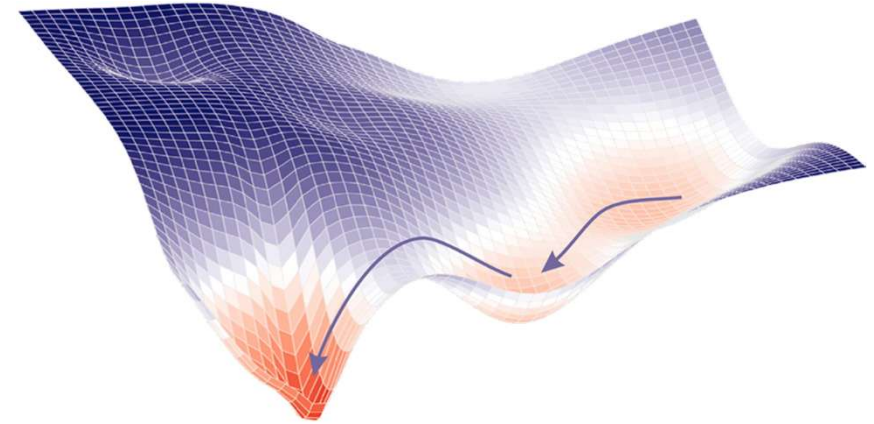
donde h_i es el campo local que siente la neurona i -ésima debido a las interacciones con sus vecinos.

Modelo de Hopfield de red neuronal

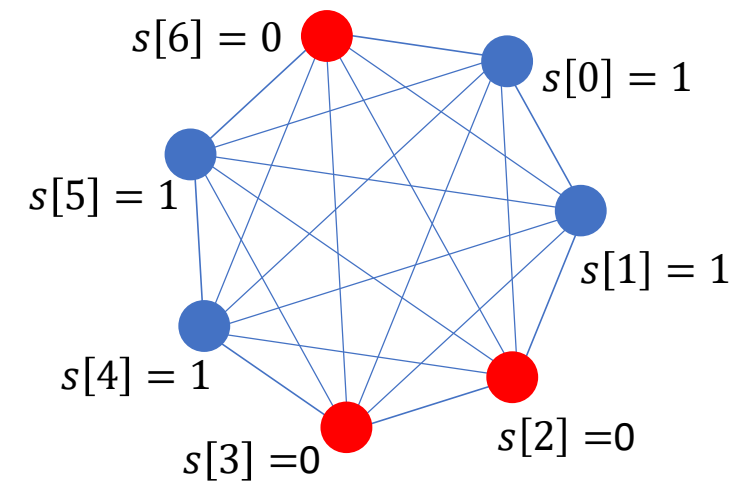
- Elegimos los pesos de modo que un conjunto de P configuraciones particulares de la red $\xi^\mu = \{\xi_i^\mu = 1, 0 \ i = 0, \dots, N\}$, llamados **patrones o memorias**, se conviertan en **atractores de la dinámica del sistema** (es decir, en mínimos del Hamiltoniano). Esto lo conseguimos la **regla de aprendizaje Hebbiano**:

$$w_{ij} = \frac{1}{a(1-a)N} \sum_{\mu=1}^P (\xi_i^\mu - a)(\xi_j^\mu - a)$$

con $a = \langle \xi_i^\mu \rangle$.



Ejemplo de memoria o patrón:



Modelo de Hopfield de red neuronal: Regla de Hebb

- La **teoría Hebbiana**, introducida por Donald Hebb en 1949, describe un **mecanismo básico de plasticidad sináptica** en el que el valor de una conexión sináptica se incrementa si las neuronas de ambos lados de dicha sinapsis se activan repetidas veces de forma simultánea:

Supongamos que la persistencia de una actividad repetitiva (o "señal") tiende a inducir cambios celulares duraderos que promueven su estabilidad. ... Cuando el axón de una célula A está lo suficientemente cerca como para excitar a una célula B y repetidamente toma parte en la activación, ocurren procesos de crecimiento o cambios metabólicos en una o ambas células de manera que tanto la eficiencia de la célula A, como la capacidad de excitación de la célula B son aumentadas.

- La teoría se resume a menudo como: **"las neuronas que se disparan juntas permanecerán conectadas"** (*neurons that fire together wire together*), aunque esto es una simplificación del sistema nervioso no debe tomarse literalmente, así como no representa con exactitud la declaración original de Hebb sobre la causalidad necesaria en los disparos.
- La teoría es comúnmente evocada para explicar algunos tipos de **aprendizajes asociativos** en los que la activación simultánea de las células conduce a un pronunciado aumento de la fuerza sináptica.

Modelo de Hopfield de red neuronal

- Elegimos los pesos de modo que un conjunto de P configuraciones particulares de la red $\xi^\mu = \{\xi_i^\mu = 1, 0 \ i = 0, \dots, N\}$, llamados **patrones o memorias**, se conviertan en **atractores de la dinámica del sistema** (es decir, en mínimos del Hamiltoniano). Esto lo conseguimos la **regla de aprendizaje Hebbiano**:

$$w_{ij} = \frac{1}{a(1-a)N} \sum_{\mu=1}^P (\xi_i^\mu - a)(\xi_j^\mu - a)$$

con $a = \langle \xi_i^\mu \rangle$.

- La probabilidad de que una neurona dispare viene dada por la función de transferencia en función de su campo local y umbral:

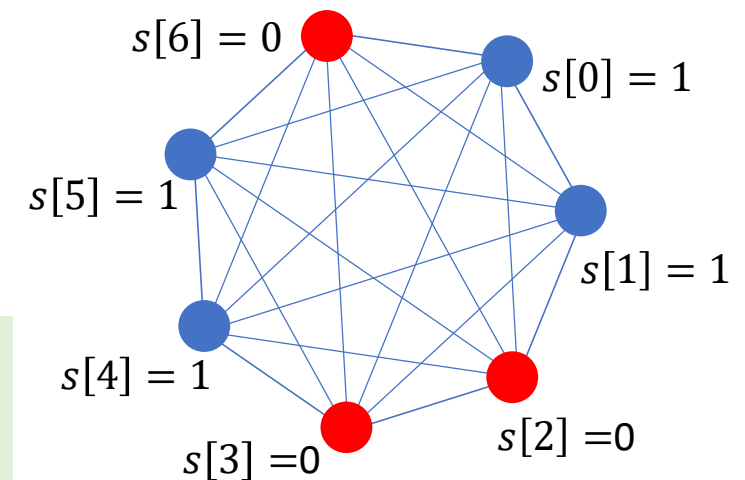
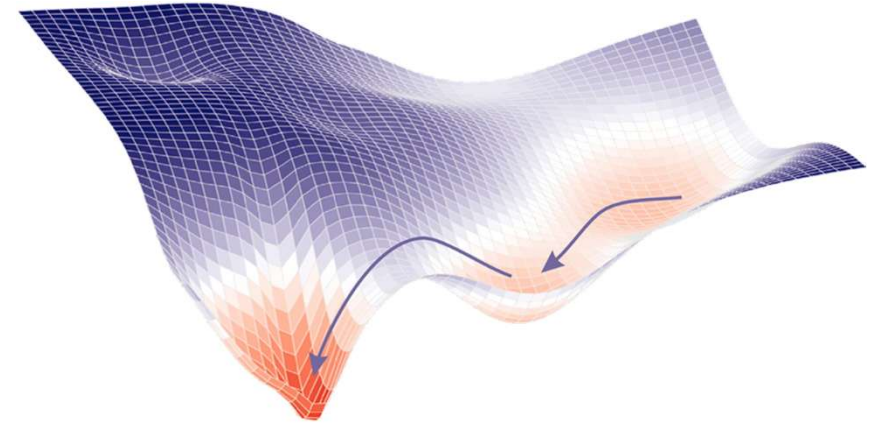
$$P(s[i](t+1) = 1) = \frac{1}{2} (1 + \tanh(\beta(h[i](t) - \theta[i])))$$

donde $\beta = 1/T$ es un parámetro que controla la estocasticidad (análogo al inverso de la temperatura en física estadística).

- Para $T = 0$, el proceso es determinista:

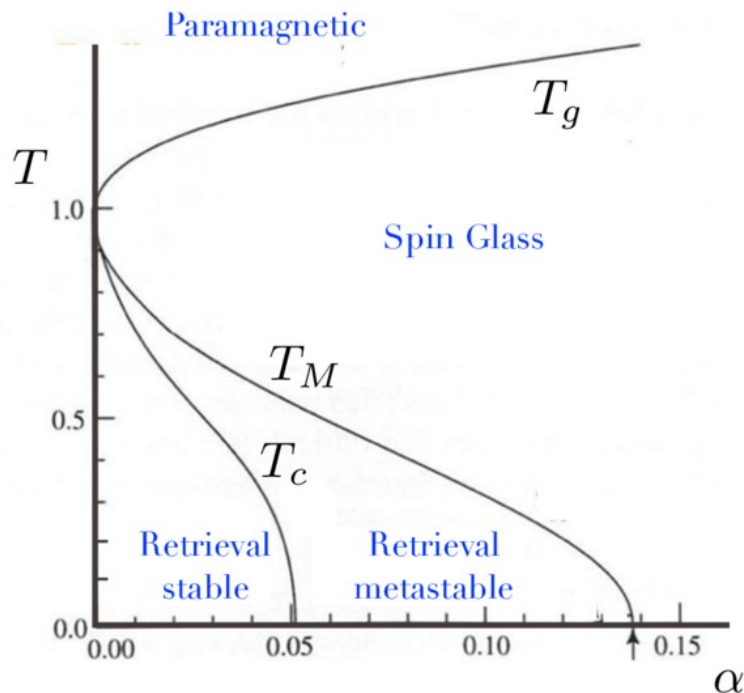
$$P(s[i](t+1) = 1) = 1 \text{ si } h[i](t) > \theta[i]$$

- Por el contrario, para $T = \infty$, $P(s) = cte$ y todas las configuraciones son igualmente probables.



Ejemplo de memoria o patrón:

Modelo de Hopfield de red neuronal: Diagrama de fases



- **Parámetros de control:** temperatura T , carga de memoria $\alpha = P/N$.
- **Parámetros de orden:** Solapamiento, que mide el grado de recuperación de las memorias:

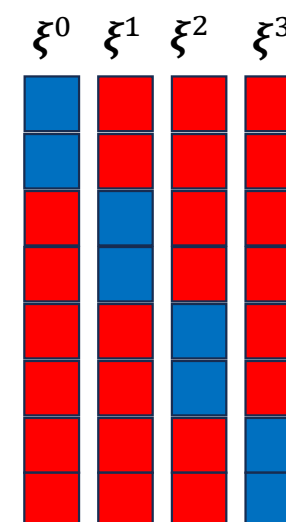
$$m^\mu(s) = \frac{1}{a(1-a)N} \sum_i (\xi_i^\mu - a)(s_i - a)$$

- Distinguimos tres fases:
 1. **Fase de memoria (retrival).** La red es capaz de recuperar correctamente los patrones almacenados. Se da a **baja temperatura y baja carga**. Las memorias son atractores estables: $m^i \rightarrow 1$ para un cierto patrón i .
 2. **Fase de vidrio de espines (spin glass).** Existe un gran número de estos metaestables (mínimos locales de la energía. Aparece a **temperaturas intermedias y alta carga**. El solapamiento con los patrones es cercano a cero, pero el sistema no es completamente desordenado.
 3. **Fase paramagnética (desordenada).** El estado de la red es completamente aleatorio. No hay memoria, el sistema es incapaz de recuperar patrones. Se da a **temperaturas altas**. El solapamiento con cualquier patrón es $m^i \rightarrow 0 \forall i$.

Modelo de Hopfield de red neuronal: Descripción del algoritmo

1. Crear un **vector de estados s** de tamaño N (número de neuronas) con un estado inicial aleatorio: $s_i = 0,1$ con probabilidad $1/2$.
2. Definir un conjunto de **P memorias** $\{\xi^\mu\}_{\mu=0}^{P-1}$, $\xi^\mu = \{\xi_i^\mu = 0, 1\}_{i=0}^{N-1}$. Para empezar, podemos definir la memoria μ -ésima como $\xi_i^\mu = 1$ para $i \in [i \cdot \mu, (i + 1) \cdot \mu]$, y $\xi_i^\mu = 0$ para el resto de nodos (ver ejemplo). Calcular la **activación media de los patrones** $a = \frac{1}{NP} \sum_{\mu} \sum_i \xi_i^\mu$.
3. Calcular la **matriz de pesos sinápticos** de acuerdo a la regla de Hebb: $w_{ij} = \frac{1}{a(1-a)N} \sum_{\mu=1}^P (\xi_i^\mu - a)(\xi_j^\mu - a)$.
4. Calcular el **vector de umbrales** $\theta_i = \frac{1}{2} \sum_j w_{ij}$.
5. **BUCLE DE EVOLUCIÓN:**
 1. Elegir una neurona i al azar. Calcular el campo local $h[i]$.
 2. Calcular la probabilidad de que la neurona dispare: $P(s[i](t+1) = 1) = \frac{1}{2} (1 + \tanh(\beta(h[i](t) - \theta[i])))$.
 3. Sortear un número aleatorio u a partir de una distribución uniforme entre 0 y 1.
 4. Si $P(s[i](t+1) = 1) > u$, la neurona dispara: hacemos $s[i] = 1$. En caso contrario, la neurona no dispara: hacemos $s[i] = 0$.
 5. Ir a 5.1.

Ejemplo con
 $N = 8$ y $P = 4$:



Nota sobre la simulación: La unidad de tiempo básica es el Paso Monte Carlo (MCS). Cada MCS equivale a realizar N intentos de actualización del sistema. Así, en promedio todos se intenta actualizar cada espín una vez por cada MCS.

Modelo de Hopfield de red neuronal: Objetivos

1. Ver cómo la red es capaz de recordar los patrones almacenados en función de T , considerando un conjunto pequeño de patrones ilustrativos (ver siguiente diapositiva), partiendo **a)** de una configuración aleatoria y **b)** del mismo patrón con una pequeña deformación. **Mostrar la evolución del sistema mediante un gif.**
 1. Pista: para que el estado inicial sea un patrón ξ^μ ligeramente deformado, puedes hacer $s(t=0) = \xi^\mu$ y a continuación cambiar el estado de las neuronas con una probabilidad pequeña p , que indicará el porcentaje de deformación.
 2. ¿Cuántos pasos Monte Carlo tarda el sistema aproximadamente en alcanzar el equilibrio?
2. Analizar qué sucede cuando aumenta mucho el número de patrones (por simplicidad considerar $T = 0$).
3. Estudiar qué sucede cuando el estado inicial es inverso a uno de las memorias (anti-patrón).
4. Considerando una red de $N = 400$ neuronas, estudiar m^μ en los siguientes casos:
 1. Para $P = 1$, calcular $|m| = |m^{\mu=0}|$ en función de T . ¿Cómo se comporta la curva? ¿Cuál es el comportamiento esperado?
 2. Para $T = 0$, en función del número de memorias almacenadas P . Asumiendo que una memoria ξ^μ se “recuerda” si $|m^\mu| > 0.75$, calcular la fracción máxima $\alpha_c = P_c/N$ de patrones que la red podrá almacenar de forma que aún pueda recordar los patrones.

Modelo de Hopfield de red neuronal: Ejemplos de patrones

