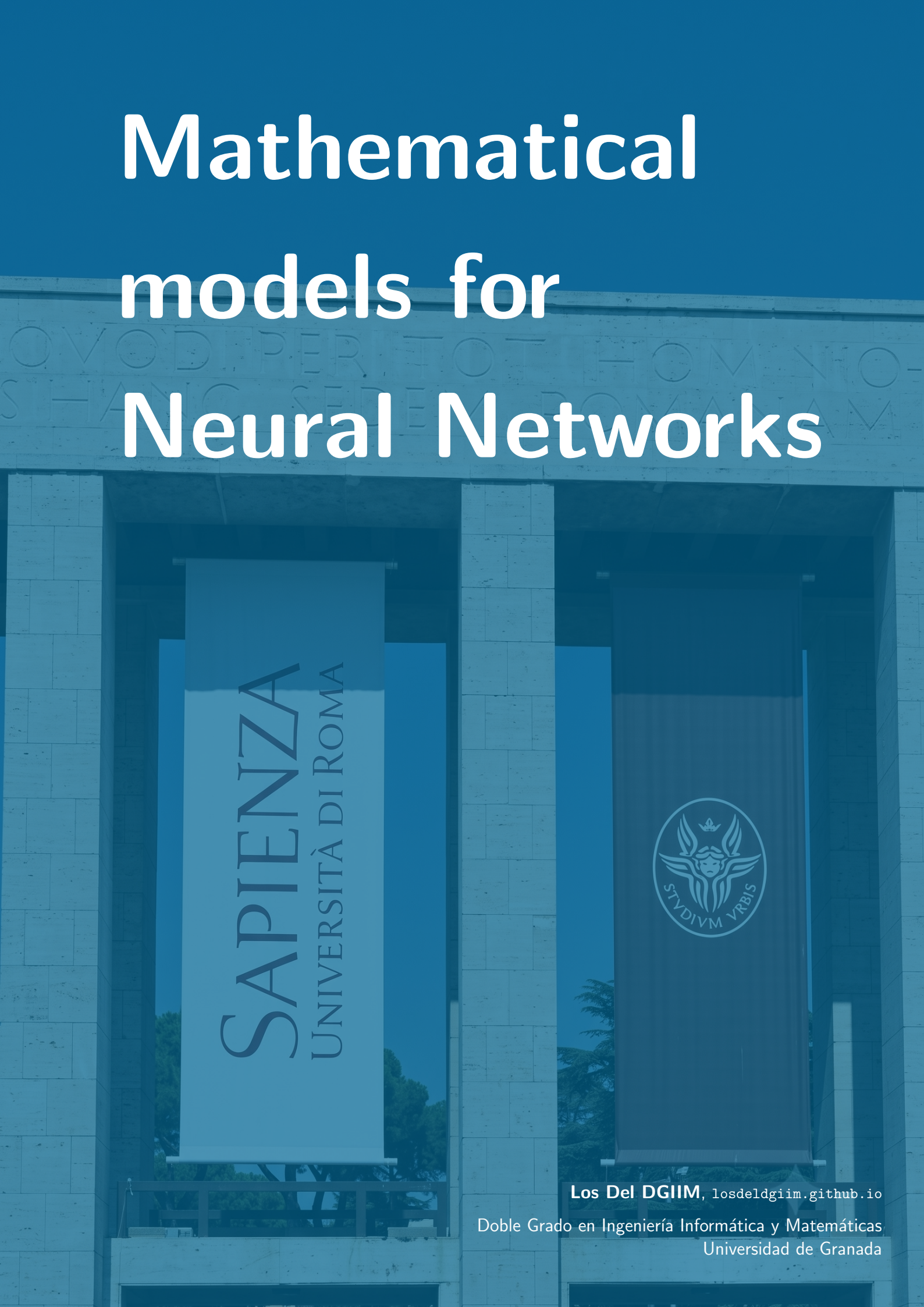
Mathematical models for Neural Networks



Los Del DGIIM, losdeldgiim.github.io

Doble Grado en Ingeniería Informática y Matemáticas
Universidad de Granada

Mathematical models for Neural Networks

Los Del DGIIM, losdeldgiim.github.io

Joaquín Avilés de la Fuente

Granada, 2025-2026

Índice general

1. Introduction to Stochastic and Mechanical Statistics

En esta sección se introducen los conceptos básicos de la estadística estocástica y mecánica, necesarios para el estudio de los modelos matemáticos en redes neuronales.

1.1. Stochastic Statistics

La estadística estocástica se ocupa del análisis de sistemas que involucran incertidumbre y aleatoriedad.

Definición 1.1.1. (Sistema estocástico) Un sistema determinista es un sistema predecible, donde sabiendo las reglas y el estado inicial puedes predecir el estado futuro, mientras un **sistema estocástico** es un sistema que incorpora elementos de aleatoriedad o incertidumbre, lo que significa que su comportamiento futuro no puede predecirse con certeza absoluta, incluso si se conoce el estado inicial y las reglas que rigen el sistema.

1.1.1. Cadena de Markov

La cadena de Markov es un modelo matemático que usaremos en las redes neuronales, donde destacaremos la **Propiedad de Markov**.

Teorema 1.1.1 (Propiedad de Markov)**.** *El futuro del sistema no depende del pasado, solo del presente, es decir, del estado en el que se encuentra el sistema en un momento dado, y no de como llegó a él.*

Observación. En la apliación a las redes neuronales, el estado de la red en el momento $t+1$ depende únicamente del estado en el momento t, y no de los estados anteriores.

1.2. El Modelo de Ising y su Interpretación Neuronal

El modelo de Ising proporciona el marco de la mecánica estadística para estudiar el comportamiento colectivo de las redes neuronales recurrentes. Al mapear neuronas a espines, transformamos el problema de "procesamiento de información.[en] un problema de relajación de energía", es decir, el modelo busca minimizar la energía (Hamiltoniano) del sistema.

1.2.1. Componentes del Modelo

- **Variables de estado:** $\sigma_i \in \{-1, +1\}$. La simetría alrededor de cero facilita el cálculo de correlaciones $\langle \sigma_i \sigma_j \rangle$.
- **Energía (Hamiltoniano):**

$$E(\sigma) = -\frac{1}{2} \sum_{i \neq j} J_{ij} \sigma_i \sigma_j - \sum_i h_i \sigma_i \tag{1.1}$$

En física, los sistemas siempre buscan el estado de "mínima energía"(como una pelota que siempre rueda hacia el fondo del valle). En una red neuronal, esa .[en]ergía.º Hamiltoniano es una medida de la frustración del sistema.¿Cómo se lee la fórmula?

$$E(\sigma) = \underbrace{-\frac{1}{2} \sum_{i \neq j} J_{ij} \sigma_i \sigma_j}_{\text{Interacciones}} \quad \underbrace{- \sum_i h_i \sigma_i}_{\text{Presión externa}}$$

1. El primer término (Interacciones): mide cuánto çaso"se hacen las neuronas entre sí.
   - Si J_{ij} es positivo (amigos), la energía baja si ambas neuronas están en el mismo estado (ambas $+1$ o ambas -1).
   - Si están en desacuerdo, la energía sube.
   - La red está "feliz" (baja energía) cuando las neuronas que deben estar juntas, lo están.
2. El segundo término (Campo externo): es como un viento o una fuerza que empuja a una neurona hacia un lado específico, sin importar lo que digan las demás.

Conclusión: el Hamiltoniano es una función que nos dice qué tan estable es una configuración. Si la energía es baja, el sistema ha encontrado un recuerdo.º un estado estable. Si es alta, el sistema está en un estado de confusión o transición

- **Matriz de interacción:** J_{ij}. En modelos de memoria (Hopfield), se construye mediante la regla de Hebb: $J_{ij} = \frac{1}{N} \sum_\mu \xi_i^\mu \xi_j^\mu$.
  Si el Hamiltoniano nos dice dónde están los valles, la **Regla de Hebb** es el proceso mediante el cual çavamos.[es]os valles en el paisaje para guardar información.

  Se basa en la premisa biológica: *"Neuronas que se disparan juntas, se conectan juntas"*. Matemáticamente, si queremos que la red memorice P patrones distintos (llamados ξ^μ), definimos los pesos J_{ij} como:

$$J_{ij} = \frac{1}{N} \sum_{\mu=1}^{P} \xi_i^\mu \xi_j^\mu \tag{1.2}$$

¿Cómo funciona la lógica de esta suma?

1. Tomamos un patrón que queremos recordar (por ejemplo, una imagen de una letra .ª").
2. Miramos dos neuronas cualesquiera, i y j.
3. Si en el patrón ambas están encendidas $(+1 \cdot +1)$ o ambas apagadas $(-1 \cdot -1)$, su producto es $+1$. Esto hace que el peso J_{ij} aumente: la red aprende que estas dos neuronas suelen ir de la mano.
4. Si una está encendida y la otra apagada, el producto es -1. El peso disminuye: la red aprende que cuando una se activa, la otra debe callarse.

Conclusión: la matriz J_{ij} no es más que un "promedio"de todas las correlaciones de los patrones que le hemos enseñado a la red.

- El **campo local** (ϕ_i): es la suma $\sum J_{ik}\sigma_k + h_i$. Representa la "fuerza total"que siente la neurona i en un momento dado.

1.2.2. ¿Qué es exactamente ξ_i^μ?

Para entender la matriz de pesos J_{ij}, debemos entender los "ladrillosçon los que se construye: los vectores ξ.
Definición: ξ^μ representa un **patrón de memoria** específico (una imagen, una palabra, un estado). Si tenemos una red de N neuronas y queremos que memorice P cosas distintas, tendremos P vectores diferentes.

Desglose de la notación: En la expresión ξ_i^μ:

- **El índice superior μ (mu):** Indica *cuál* recuerdo estamos mirando. Si $\mu = 1$ es la cara de tu abuela, si $\mu = 2$ es el número de tu teléfono. Va desde 1 hasta P.
- **El índice inferior i:** Indica *qué neurona* del recuerdo estamos mirando. Va desde 1 hasta N.
- **El valor $\xi_i^\mu \in \{-1, +1\}$:** Es el estado que *debería* tener la neurona i para representar correctamente el recuerdo μ.

Ejemplo Visual: Imagina una cuadrícula de 3×3 neuronas $(N = 9)$ para representar letras. Si queremos guardar la letra "X"(nuestro patrón $\mu = 1$):

- $\xi_1^1 = +1, \xi_3^1 = +1$ (las esquinas superiores se encienden).
- $\xi_5^1 = +1$ (el centro se enciende).
- $\xi_2^1 = -1$ (la neurona de la parte superior central se queda apagada).

1.2.3. El papel de ξ en la Regla de Hebb

Recordando la fórmula:

$$J_{ij} = \frac{1}{N} \sum_{\mu=1}^{P} \xi_i^\mu \xi_j^\mu \tag{1.3}$$

Aquí, el producto $\xi_i^\mu \xi_j^\mu$ actúa como un **comparador**:

- Si en el recuerdo μ, las neuronas i y j deben estar igual (ambas $+1$ o ambas -1), el producto es $+1$. La conexión J_{ij} se fortalece.
- Si deben estar opuestas, el producto es -1. La conexión se debilita.

Conclusión: Los ξ_i^μ son las ïnstrucciones de montaje"para los valles del Hamiltoniano. Al sumar todos los productos $\xi_i\xi_j$ de todos los patrones, el peso J_{ij} acaba siendo un compromiso (un promedio) de cómo deben relacionarse esas dos neuronas para satisfacer todos los recuerdos a la vez.

1.2.4. El efecto de la Temperatura

La temperatura T en este modelo no es calor físico, sino una medida del *ruido sináptico*.

- Para $T < T_c$ (Temperatura crítica): El sistema presenta **ruptura de ergodicidad**. La red puede quedar .ªtrapada.en un valle de energía, lo que equivale a recordar un patrón de forma estable.
- Para $T > T_c$: El sistema se vuelve desordenado. El ruido impide que las neuronas colaboren para mantener el patrón.

1.2.5. Mapeo de Sistemas

La relación entre el umbral de una neurona MP (U_i^*) y el campo externo de Ising (h_i) es:

$$h_i = \sum_{j=1}^{N} J_{ij} - 2U_i^* \tag{1.4}$$

Este mapeo demuestra que cualquier red de neuronas binarias puede ser analizada como un sistema físico de espines interactuantes bajo un campo magnético.

2. Biological inspiration

2.1. McCulloch-Pitss neuron

The model aims to represent a basic neuron, that receives a signal from the N neighbouring neurons (as we can see in Figure 2.1), whose state is represented by the vector $x = (x_1, x_2, \ldots, x_N)$ and which are associated to a synaptic weight vector $J = (J_1, J_2, \ldots, J_N)$; the neuron elaborates this information an returns as output the variable y

$$y = \theta\left(\sum_{j=1}^{N} J_j x_j - U^*\right) \tag{2.1}$$

where θ is the Heavyside function and U^* is the **activation threshold of the neuron**.

Definición 2.1.1. (Heavyside function) The main purpose of the **Heavyside function** is to switch something on or off depending on whether the input is positive or not. Formally, it is defined as

$$\theta(x) = \begin{cases} 1 & \text{if } x \geqslant 0 \\ 0 & \text{if } x < 0 \end{cases}$$

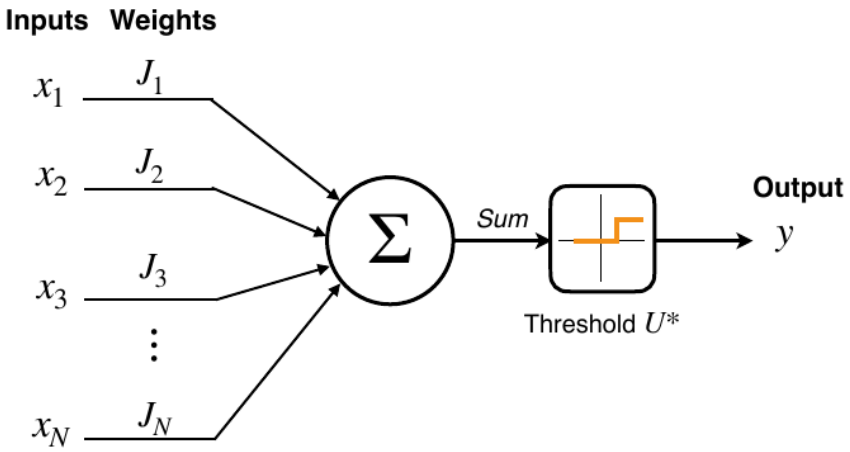


Figura 2.1: Schematic representation of a MP neuron.

Before proceeding with the construction of networks of MP neurons it is worth noticing that there are two main classes of networks: **feed-forward networks** and

recurrent networks.

In the former, one distinguishes different layers of neurons, say layer 1, 2, ..., L and connections among them are directed in such a way that a neuron in layer n receives input from one (or more) neurons in layer n - 1 and contributes to the input to one (or more) neuron in layer n + 1. Thus, in this neural network, information only moves in one direction, forward, with respect to entry nodes, through nodes belonging to intermediate (if any) layers - these are also called hidden nodes - to exit nodes. Otherwise stated, in this case, $J_{ij} \neq 0$ and $J_{ji} = 0$ if neurons j and i belong, respectively, to layers $n-1$ and n, but $J_{ki} \neq 0$ if neuron k belongs to layer $n+1$.

Conversely, in recurrent networks, connections are not directed and if a neuron x_i is connected to a neuron x_j then x_i affects the state of x_j which, in turn, affects the state of x_i ; clearly, in this case, we have the existence of cycles. In this case it is convenient to label the MP neuron as well, namely to recast y as x_{N+1}; accordingly, the vector J is reshaped as a matrix $J = \{J_{ij}\}_{i,j=1,\ldots,N+1}$, in such a way that eq. (2.1) turns into

$$x_i = \theta(\sum_{j=1}^{N+1} J_{ij} x_j - U_i^*) \quad i = 1, \ldots, N+1 \tag{2.2}$$

where J_{ij} therefore represents the amplification factor of the signal due to the j-th neuron on the i-th neuron and, in general, the effect can be asymmetric, namely $J_{ij} \neq J_{ji}$, and the sum possibly includes self-interactions, that is, J_{ii} can be non zero. In the following we will only consider symmetric recurrent networks, where the connection matrix is symmetric, that is, $J_{ij} = J_{ji}$ for any i, j.

2.1.1. Universality of McCulloch-Pitts neurons

Here we will show that networks of MP neurons have universal computational capabilities in the sense that the operation of any finite, deterministic, digital-information processing machine can be emulated by a properly constructed network of MP neurons. Since the basic logical units of digital machines can be constructed with MP neurons, all the theorems of computer science (on computability, Turing machines, etc.) that apply to such digital machines will also apply to MP neural networks.

We are going to demonstrate how any such machine can be reduced to a collection of simpler single-output binary machines:

- Every finite digital machine can be regarded as a device that maps finite-dimensional binary input vectors $x \in \Omega \subseteq \{0,1\}^N$ (representing characters, numbers, keys typed on a keyboard, camera images, etc.) onto finite dimensional binary output vectors $y \in \{0,1\}^K$. Therefore every finite deterministic digital machine can be specified in full by specifying the output vector $y = S(x)$ for every possible input vector $x \in \Omega$, that is, by specifying the mapping M :

$$M : \Omega \to \{0,1\}^K \quad Mx = S(x) \quad \forall x \in \Omega$$

- Finally, we can always construct K independent sub-machines M_l ($l = 1, \ldots, K$), each of which takes care of one of the K output bits of the full mapping M :

$$M_l : \Omega \to \{0,1\} \quad M_l x = S_l(x) \quad \forall x \in \Omega.$$

We conclude that we are allowed to concentrate only on the question of whether it is possible to perform any single-output mapping $M : \Omega \subseteq \{0,1\}^N \to \{0,1\}$ with networks of MP neurons. This question will be answered in the affirmative in two steps: (i) we first show that any such single-output Boolean function of N variables can be expressed in terms of combinations of three logical operations performed on the inputs (AND, OR, NOT), (ii) the three elementary logical operations can be performed by MP neurons.
For further details on these demonstrations, please refer to page 24 of the book.

For statement (ii), the answer is positive only for $N = 1$ that is, only for the trivial case of a single input variable $x \in \{0,1\}$, we can check all possible operations $M : \{0,1\} \to \{0,1\}$ (of which there are four, to be denoted by M_a, M_b, M_c, and M_d), and verify that one can always construct an equivalent MP neuron $y = S(x) = \theta(Jx - U^*)$, as outlined by Table 2.1.

x	$M_a(x)$	$\theta(-1)$	$M_b(x)$	$\theta(-x+1/2)$	$M_c(x)$	$\theta(x-1/2)$	$M_d(x)$	$\theta(1)$
0	0	0	1	1	0	0	1	1
1	0	0	0	0	1	1	1	1

Tabla 2.1: The table shows the possible 4 mappings conceivable when $N = 1$.

For $N > 1$, however, the answer is no. There are several ways for showing this. The simplest is to give first a counterexample for $N = 2$, which can subsequently be used to generate counterexamples for any $N \geqslant 2$, the so-called XOR operation (exclusive OR).

Proposición 2.1.1. *No set of real numbers* $\{J_1, J_2, U^*\}$ *exists such that*

$$\theta(J_1 x_1 + J_2 x_2 - U^*) = XOR(x_1, x_2) \quad \forall (x_1, x_2) \in \{0,1\}^2$$

Geometric representation

The set of all possible operations $M : \{0,1\}^N \to \{0,1\}$ consists of all possible ways of assigning the output values 0 or 1 to the set of values $\{0,1\}^N$; since there are 2^N possible values in the domain of the function and each one can be assigned either 0 or 1, the total number of different operations is 2^{2^N}.

A MP neuron, performing the operation

$$S : \{0,1\}^N \to \{0,1\}, \quad S(x) = \theta\left(\sum_{k=1}^{N} J_k x_k - U^*\right) \tag{3}$$

can be seen as dividing the space $\mathbb{R}^N$ into two subsets which are separated by the hyperplane $\sum_{k=1}^{N} J_k x_k = U^*$. The information conveyed by the outcome of the

operation of a MP neuron, given an input $x \in \{0,1\}^N$, is simply in which of the two subsets the corner x of the hypercube is located:

$$S(x) = \begin{cases} 1 & \text{if } x \text{ is in the subset } \sum_{k=1}^{N} J_k x_k > U^* \\ 0 & \text{if } x \text{ is in the subset } \sum_{k=1}^{N} J_k x_k < U^* \end{cases}$$

For the moment, let us not worry about the pathological case where x is on the separating plane. For $N = 2$, the hypercube is a square,. There are $2^2 = 4$ corners and $2^{2^2} = 16$ operations, $M : \{0,1\}^2 \to \{0,1\}$, that is, sixteen ways of colouring the four corners as we can see in Figure 2.2.

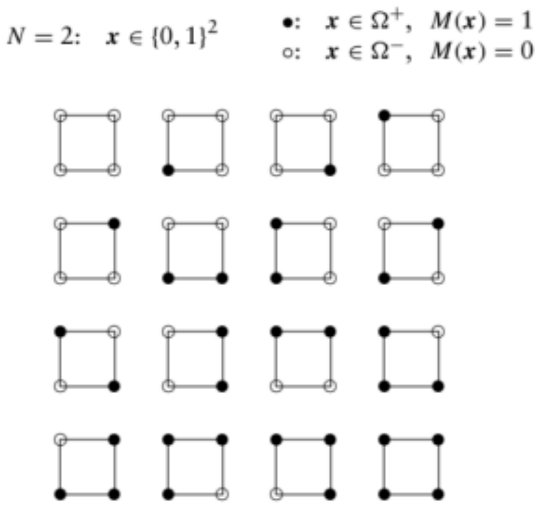


Figura 2.2: All the possible operations for $N = 2$.

Next, by defining linear separability, we can see in Figure 2.3 which operations are linearly separable.

Definición 2.1.2. Let X_1 and X_2 be two sets of points in an n-dimensional Euclidean space. Then X_1 and X_2 are **linearly separable** if there exist $n + 1$ real numbers $w_1, w_2, \ldots, w_n, k$, such that every point $x \in X_1$ satisfies $\sum_{i=1}^{n} w_i x_i > k$ and every point $x \in X_2$ satisfies $\sum_{i=1}^{n} w_i x_i < k$. The expression $x \cdot w = k$ identifies a hyperplane that is referred to as the **separator hyperplane**. Setting $x_0 = 1$ and $w_0 = -k$, the separator hyperplane can be recast as $x \cdot w = 0$.

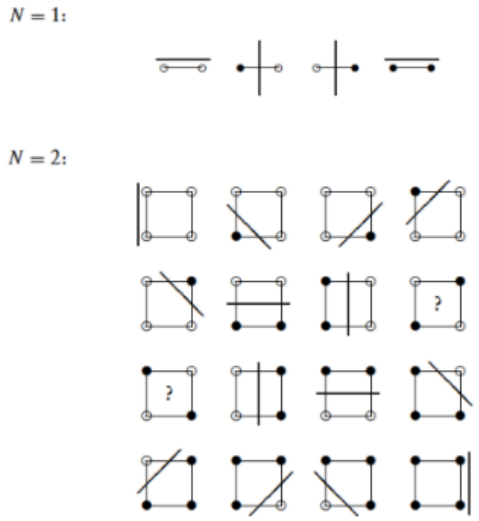


Figura 2.3: Non linearly separable sets of points in a 2D space.

2.2. Multilayer networks of MP neurons

In accordance with what was previously agreed, a single MP neuron cannot realize every mapping $M : \{0,1\}^N \longrightarrow \{0,1\}$, however, we also noticed that combinations of simple operations (each achievable by a single MP neuron) can emulate any operation M, so in this section we are going to work with the "grandmother-cell" (or "Jennifer Aniston neuron") concept. The main idea of this concept is that a single neuron, called the grandmother cell, is a hypothecal neuron that represents a complex but specific concept or object. It activates when a person "sees, hears, or otherwise sensibly discriminates.[a] specific entity.

Notación. The configuration of the middle (or 'hidden') layer will be denoted as $y^{(1)} = (y_1^{(1)}, \ldots, y_L^{(1)})$; this notation allows a straightforward generalization in the case of multiple hidden layers, that is, $y^{(i)} = (y_1^{(i)}, \ldots, y_j^{(i)}, \ldots, y_n^{(i)})$ with $i \in \mathbb{N}$, whose sizes can, in principle, be arbitrary.

Se busca una estructura de una MP neuron, es decir, se buscan weights W_{ij} and firing thresholds V_i for hidden neurons that satisfying

$$y_i^{(1)}(x) = \theta\left(\sum_{j=1}^{N} W_{ij}x_j - V_i\right) = \delta_{x,x^i} \quad \text{with } x^i \in \Omega^+ \text{ and } x \in \{0,1\}^N$$

For the case $N = 2$, let us see how to construct a function to differentiate the vectors that return $+1$ from those that return -1, so given $\Omega^+ = \{x^1, x^2\}$ where $x^1 = (1,0), x^2 = (0,1)$ we can define

$$\sum_{j=1}^{2}(2x_j^i - 1)(2x_j - 1) = \sum_{j=1}^{2} 2(2x_j^i - 1)x_j - \sum_{j=1}^{2}(2x_j^i - 1) = \begin{cases} 2 & \text{if } x = x^i \\ \leqslant 0 & \text{otherwise} \end{cases}$$

Here, the purpose is exploiting the knowledge of Ω^+ and design the structure in such a way that, for each vector $x^i \in \Omega^+$, there is precisely one neuron i in the middle (or 'hidden') layer which produces an output +1, thereby signaling that the state of the output neuron is to be +1 when x^i occurs as input, that is, $y_i^{(1)} = 1$ if $x = x^i$ where $x^i \in \Omega^+$ and $y_i^{(1)} = 0$ otherwise.

For the resolution of this case step by step, please refer to page 31 of the book, but we can see the main idea of the final sketch in Figure 2.4.

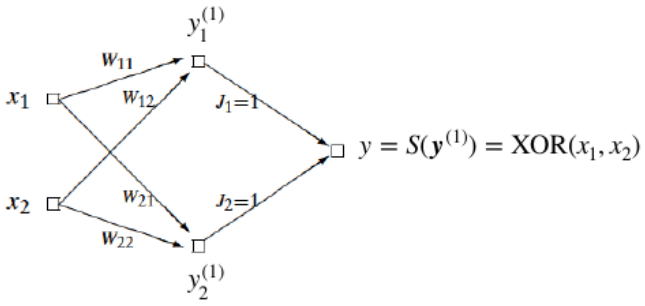


Figura 2.4: Schematic representation of a multilayer network of MP neurons.

2.2.1. General case

We now give the general construction, which works for arbitrary input dimensions N and for arbitrary operations $M : \{0,1\}^N \longrightarrow \{0,1\}$. Let $\Omega^+ = \{x \in \{0,1\}^N | M(x) = 1\}$ the set of all input vectors that are mapped to output value +1 by the operation M, and $L := |\Omega^+| \in \mathbb{N}$. Then, we construt the neurons y_i in the hidden layer such that the operation of each of them is defined as determining whether the input vector x equals a neuron-specific prototype vector from Ω^+, in other words, each $y_i^{(1)}$ is a **detector** for the prototype vector $x^i \in \Omega^+$, where its value would be 1 if $x = x^i$ and 0 otherwise.

Definición 2.2.3. For the construction of the associated grandmother neurons we need to define the **building block** G:

$$G(x, x^*) = \theta\left(\sum_{i=1}^{N}(2x_i - 1)(2x_i^* - 1) - N + 1\right), \quad x, x^* \in \{0,1\}^N$$

Let's look at the main property that will interest us of this function.

Proposición 2.2.2. *For the building block G defined above we have*

$$G(x, x^*) = 1 \iff x = x^*$$

Namely $G(x, x^*) = SAME(x, x^*)$.

Demostración. If $x = x^*$, then

$$G(x,x) = \theta\left(\sum_{i=1}^{N}(2x_i-1)^2 - N + 1\right) = \theta(1) = 1$$

Now we show that $G(x,x^*) = 0$ as soon as $x \neq x^*$:

$$\forall x_i, x_i^* \in \{0,1\}: \quad (2x_i-1)(2x_i^*-1) = \begin{cases} 1 & \text{if } x_i = x_i^* \\ -1 & \text{if } x_i \neq x_i^* \end{cases}$$

Consequently,

$$\sum_{i=1}^{N}(2x_i-1)(2x_i^*-1) = N - 2 \times (\text{number of indices } i \text{ with } x_i \neq x_i^*)$$

in such a way that

$$x \neq x^* \implies \sum_{i=1}^{N}(2x_i-1)(2x_i^*-1) \leqslant N-2 \implies G(x,x^*) = 0$$

which completes the proof. □

En este sentido, reestructurando el bloque G se obtiene que la configuración buscada es

$$G(x,x^*) = \theta\left(2\sum_{i=1}^{N}(2x_i^*-1)x_i - 2\sum_{i=1}^{N}x_i^* + 1\right)$$

thus, $G(x,x^*)$ is clearly of the MP form.
We can now construct our two-layer network from these building blocks, by assigning to each vector $x \in \Omega^+$ a hidden neuron of the form $G(x,x^*)$

$$\Omega^+ = \{x \in \{0,1\}^N | M(x) = 1\} = \{x^1, x^2, \ldots, x^{L-1}, x^L\}$$

$$\forall i \in \{1,\ldots,L\} \quad y_i^{(1)} : \{0,1\}^N \to \{0,1\} \quad y_i^{(1)} = G(x,x^i).$$

The required operation of the output neuron S, finally, is to simply detect whether or not any of the hidden layer neuron states $y_i^{(1)}$ equals one. Our final result is the following network

$$y_i^{(1)}(x) = \theta\left(\sum_{j=1}^{N} W_{ij}x_j - V_i\right), \quad y = S(y^{(1)}) = \theta\left(\sum_{i=1}^{L} y_i^{(1)} - \frac{1}{2}\right)$$

where $W_{ij} = 2(2x_j^i - 1)$ and $V_i = 2\sum_{j=1}^{N} x_j^i - 1$. Destacar que tenemos $U^* = \frac{1}{2}$ y $J_i = 1, \forall i = 1,\ldots,L$ en la neurona de salida.

2.3. The perceptron

For a MP neuron $S(x) = \theta(J \cdot x - U^*)$ learning means the adaptation of the set of connections $\{J_i\}_{i=1}^N$ and the threshold U^*, in order to improve the accuracy in the execution of a given task $M : \{0,1\}^N \to \{0,1\}$.

We don't know the task M explicitly, we only have some available examples provided by some "teacher" of "questions" (input vectors $x \in \{0,1\}^N$) with corresponding "answers" (the associated output values $M(x)$), and the main idea is that $S(x)$ classifies examples $x \in \{0,1\}^N$ "mimicking" $M(x)$. A so-called "online" training session consists of iterating the following procedure:

- STEP 1: Draw at random a question $x \in \Omega$
- STEP 2: Check whether teacher and student agree on the answer:
  - if $M(x) = S(x)$: do nothing, return to STEP 1.
  - if $M(x) \neq S(x)$: modify parameters

$$\begin{array}{rcl} J & \longmapsto & J + \Delta J(J, U^*; x, M(x)) \\ U & \longmapsto & U^* + \Delta U^*(J, U^*; x, M(x)) \end{array}$$

  then return to STEP 1.

Definición 2.3.4. (Experience) Llamaremos **experience** al conjunto de ejemplos $\{x^{(i)}, M(x^{(i)})\}_{i=1,\dots,P}$ que el perceptrón utiliza para aprender a aproximar la función objetivo $M : \{0,1\}^N \to \{0,1\}$.

Definición 2.3.5. (Performance) We define the **performance** of the perceptron on a given experience as la exactitud de la función S respecto a la función objetivo M en los P ejemplos que componen la experiencia:

$$dist(\{x^{(i)}, M(x^{(i)})\}_{i=1,\dots,P})$$

The problem one faces is how to choose an appropriate recipe $(\Delta J, \Delta U^*)$ for updating the neuron's parameters that will achieve this. A perceptron is defined as a MP neuron which learns according to the following rule (perceptron learning rules) for updating its parameters:

$$\begin{cases} S(x) = 1, M(x) = 0 \Rightarrow \Delta U^* = 1, \Delta J = -x \\ S(x) = 0, M(x) = 1 \Rightarrow \Delta U^* = -1, \Delta J = x \end{cases}$$

Teorema 2.3.3. *(Perceptron convergence theorem, Rosenblatt [1958]) If the task M is linearly separable, then the above procedure will converge in a finite number of modification steps to a stationary configuration, where $\forall x \in \Omega : S(x) = M(x)$.*

2.4. Recurrent networks of MP neurons

First, having in mind a network of MP neurons that mutually interact each other, we can look at (2.1) as an evolutionary rule which can be applied in turn to all the neurons making up the network and which can be possibly iterated. We therefore introduce a dependence on a time variable as follows

$$x_i(t + \Delta t) = \theta\left(\sum_{j=1}^{N} J_{ij}x_j(t) - U_i^*\right) \tag{2.3}$$

where Δt represents the refractory time.

Notación. From now on, we will use the following notation $U_i(t) = \sum_{j=1}^{N} J_{ij}x_j(t)$ to indicate the post-synaptic potential acting on neuron i at time t.

En el modelo anterior de McCulloch-Pitts (MP), la neurona es un robot predecible. Compara su entrada $U_i(t)$ con un umbral fijo U_i^*. Si $U_i(t) > U_i^*$, se activa; si no, no. El problema es que este sistema se queda atascado en mínimos locales (memorias espurias). La solución estocástica es introducir ruido"para permitir que la neurona "se equivoque.[a] propósito, dándole la capacidad de saltar fuera de esos valles de energía.

Aplicamos ahora la teoría de Procesos Estocásticos introduciendo $z_i(t)$ que es una variable aleatoria que actúa como fuente de "ruido térmico" que recibe la neurona i en el instante t.

2.4.1. Stochastic neurons

Definición 2.4.6. (Sthocastic neurons) Dada la definición de la neurona MP recurrente en (2.3), una **neurona estocástica** es aquella que tiene un umbral fluctuante dado de la siguiente manera

$$U_i^*(t) = U_i^* - \frac{1}{2}Tz_i(t)$$

donde $T \in \mathbb{R}^+$ es la **controlador de la magnitud del ruido** (o también llamado **temperatura**) y $z_i(t)$ es una **variable aleatoria** independiente e idénticamente distribuida (i.i.d.), es decir, $z_i(t) \sim_{idd} P$, $\forall i = 1, \ldots, N$. Tomaremos estas propiedades como supuestos básicos en el modelo:

- $\mathbb{E}[z_i(t)] = \overline{z_i(t)} = 0$
- $\mathbb{E}[z_i(t)^2] = \overline{z_i(t)^2} = 1$

Cabe destacar que la esperanza de la diferencia del umbral fluctuante al cuadrado es

$$\mathbb{E}[(U_i^*(t) - U_i^*)^2] = \mathbb{E}\left[\left(-\frac{1}{2}Tz_i(t)\right)^2\right] = \frac{T^2}{4}\mathbb{E}[z_i(t)^2] = \frac{T^2}{4}$$

por lo que el parámetro T controla la magnitud de las fluctuaciones del umbral, es decir, la aleatoriedad de la neurona estocástica es controlada por T.
A partir de ahora el estado de la neurona estocástica i en el instante $t + \Delta t$ viene dado por

$$x_i(t + \Delta t) = \theta\left(U_i(t) - U_i^*(t)\right)$$

2.4.2. Stochastic symmetric neurons

El objetivo será mapear nuestro sistema de neuronas estocásticas con activación $\{0, 1\}$ a un sistema de neuronas simétricas con activación $\{-1, +1\}$, ya que este último es más sencillo de analizar. Para ello, teniendo en cuenta el **modelo de neuronas de Ising** sabemos que la neurona σ_i se activa (pasa a $+1$) si la entrada neta que recibe es positiva, es decir,

$$\sum_{k=1}^{N} J_{ik}\sigma_k + h_i > 0 \Longleftrightarrow \sum_{k=1}^{N} J_{ik}\sigma_k > -h_i$$

donde h_i es su preferencia personal o un estímulo que viene de fuera de la red. Por tanto, tomando la conversión de las neuronas $x_i(t) = \dfrac{1}{2}[\sigma_i(t) + 1]$ (para pasar de sistema $\{0, 1\}$ a $\{-1, +1\}$) tenemos que

$$\sum_{k=1}^{N} J_{ik}x_k > U_i^* \iff \sum_{k=1}^{N} J_{ik}\frac{1}{2}[\sigma_i(t) + 1] > U^* \Longleftrightarrow \sum_{k=1}^{N} J_{ik}\sigma_k > 2U_i^* - \sum_{k=1}^{N} J_{ik}$$

Igualando las dos expresiones obtenidas, tenemos que el mapeo al nuevo sistema simétrico es

$$\begin{cases} x_i(t) = \dfrac{1}{2}[\sigma_i(t) + 1] \\ U_i^* = \dfrac{1}{2}\left(\displaystyle\sum_{k=1}^{N} J_{ik} - h_i\right) \end{cases} \tag{2.4}$$

The stochastic version of the MP model is the following

$$x_i(t + \Delta t) = \theta\left(\sum_{j=1}^{N} J_{ij}x_j(t) - U_i^* + \frac{1}{2}Tz_i(t)\right)$$

thereby becomes

$$\frac{1}{2}[\sigma_i(t + \Delta t) + 1] = \theta\left(\sum_{k=1}^{N} J_{ik}\frac{1}{2}[\sigma_k(t) + 1] - \frac{1}{2}\left(\sum_{k=1}^{N} J_{ik} - h_i\right) + \frac{1}{2}Tz_i(t)\right) \Longrightarrow$$
$$\Longrightarrow \frac{1}{2}[\sigma_i(t + \Delta t) + 1] = \theta\left(\frac{1}{2}\sum_{k=1}^{N} J_{ik}\sigma_k(t) + \frac{1}{2}h_i + \frac{1}{2}Tz_i(t)\right)$$

then

$$\sigma_i(t + \Delta t) = sgn\left(\varphi_i(t) + Tz_i(t)\right) \quad \text{where} \quad \varphi_i(t) = \sum_{k=1}^{N} J_{ik}\sigma_k(t) + h_i$$

Definición 2.4.7. (Stochastic version of the MP model) The **stochastic version of the MP model** for a network of N symmetric neurons is defined by the following evolutionary rule:

$$\sigma_i(t+\Delta t) = sgn\left(\varphi_i(t) + Tz_i(t)\right) \quad \text{where} \quad \varphi_i(t) = \sum_{k=1}^{N} J_{ik}\sigma_k(t) + h_i$$

so $sgn(x>0) = +1$, $sgn(x<0) = -1$; $sgn(0)$ is drawn randomly from $\{-1,+1\}$. The quantity $\varphi_i(t)$ is called the **local field**. The first term of $\varphi_i(t)$ is called the **internal field** and is due to the first neighboring neurons, that is to other elements of the system; the second term is referred to as **external field** because it acts as a bias, independent of the neuronal configuration and favouring either one of the possible states.

Lo que hemos hecho aqui realmente es unir el concepto de MP neurona estocástica con el modelo de Ising, obteniendo así una neurona estocástica simétrica. A partir de aquí podemos analizar la probabilidad de que la neurona i tome el valor $\sigma_i(t+\Delta t) = \pm 1$.

La probabilidad de que la neurona i tome el valor $\sigma_i(t+\Delta t) = +1$ se puede expresar en términos de la distribución $\mathcal{P}(z)$ de la variable aleatoria $z_i(t)$. Para las distribuciones simétricas del ruido, es decir, $\mathcal{P}(z) = \mathcal{P}(-z)\ \forall z$ tenemos

$$P[\sigma_i(t+\Delta t) = 1] = P[\varphi_i(t+\Delta t) + Tz_i(t) > 0] = \int_{-\varphi_i(t)/T}^{\infty} \mathcal{P}(z)\, dz$$

$$P[\sigma_i(t+\Delta t) = -1] = \int_{-\infty}^{-\varphi_i(t)/T} \mathcal{P}(z)\, dz = \int_{\varphi_i(t)/T}^{\infty} \mathcal{P}(z)\, dz,$$

We can now combine the two probabilities into the single expression:

$$P[\sigma_i(t+\Delta t) = \pm 1] = g(\pm\varphi_i(t)/T)$$

with

$$g(x) = \int_{-\infty}^{x} \mathcal{P}(z)dz = \frac{1}{2} + \int_{0}^{x} \mathcal{P}(z)dz.$$

The function $g(x)$ is a cumulative distribution function and it has the following properties:

(i) $g(x) + g(-x) = 1$

(ii) $\lim_{x\to-\infty} g(x) = 0,\ g(0) = 1/2,\ \lim_{x\to\infty} g(x) = 1$

(iii) $g'(x) = P(x) :\ g'(x) \geqslant 0\ \forall x,\ g'(-x) = g'(x)\ \forall x,\ \lim_{x\to\pm\infty} g'(x) = 0.$

Una elección usual de la distribución del ruido es la distribución normal estándar, es decir, $z \sim \mathcal{N}(0,1)$ con función de densidad de probabilidad

$$\mathcal{P}(z) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(z-\mu)^2}{2\sigma^2}}$$

donde $\mu = 0$ y $\sigma = 1$, por supuesto al inicio de esta sección y por tanto tenemos

$$g(x) = \frac{1}{2} + \int_0^x \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz = \frac{1}{2}\left[1 + erf\left(\frac{x}{\sqrt{2}}\right)\right]$$

Definición 2.4.8. La función $erf(z)$ se define como el área bajo una curva Gaussiana específica (normalizada de una manera particular), desde 0 hasta un valor z:

$$erf(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt$$

Mide que tan probable es que una variable Gaussiana con media $\mu = 0$ y varianza $\sigma^2 = \frac{1}{2}$, caiga entre $-z$ y $+z$.

3. Neural networks for pattern retrieval

In a network of such units, updates can be carried on either synchronously (in parallel) or randomly asynchronously (one after the other). Veámos como calcular la probabilidad de que una red de este tipo recupere un patrón almacenado a partir de una versión ruidosa del mismo, es decir, la probabilidad de llegar a un estado concreto a partir de uno dado.

Tenemos a continuación los dos casos posibles, synchronously and asynchronously:

Synchronously La probabilidad combinada de encontrar un estado $\sigma(t+\Delta t) = \{\sigma_1, \sigma_2, \ldots, \sigma_N\} \in \{1,1\}^N$ en un sistema de N unidades estocásticas actualizadas en **paralelo** es el producto de las probabilidades individuales de cada unidad, es decir,

$$P(\sigma(t+\Delta t)) = \prod_{i=1}^{N} g(\sigma_i(t+\Delta t)\frac{\varphi_i}{T}).$$

Asynchronously We have to take into account that only one neuron changes its state at a time, so we can choose i randomly from $1, \ldots, n$ or following a certain sequence $[1:N]$, and

$$P[\sigma_i(t+\Delta t)] = g\left(\sigma_i(t+\Delta t)\frac{\varphi_i(t)}{T}\right).$$

Observación. We stress that the parameter T controls the impact of this noise, in fact, it adjusts the stochasticity level affecting the neuronal dynamics and, specifically,

$$\begin{cases} T = 0 \Longrightarrow \text{deterministic process such that } \sigma_i(t+\Delta t) = sgn[\varphi_i(\sigma(t))] \\ T \mapsto \infty \Longrightarrow \text{fully random process} \end{cases}$$

3.1. Noiseless Dynamics ($T = 0$)

In this subsection we focus on the noiseless case, whence the dynamics can take the following form (hereafter we shall set $\Delta t \equiv 1$ to lighten the notation)

Parallel. The state of neuron i at time $t+1$ is given by the sign function

$$\sigma_i(t+1) = sgn\left(\sum_j J_{ij}\sigma_j(t) + h_i\right) \quad \forall i$$

Sequential. A neuron i is extracted (either randomly of following a deterministic sequence) and updated according to the sign function

$$\text{choose} \begin{cases} i \text{ randomly from } \{1, \dots, N\} \\ \sigma_i(t+1) = sgn\left(\sum_j J_{ij}\sigma_j(t) + h_i\right) \end{cases}$$

Imagina que tu red tiene N neuronas. Como cada una puede estar en $+1$ o -1, el número total de configuraciones posibles es 2^N. Esto se llama el espacio de estados. Como el número de estados es finito, si el sistema es determinista, tarde o temprano tiene que volver a un estado en el que ya estuvo. En ese momento, se cierra un ciclo.

- Punto fijo (Periodo 1): es el escenario ideal. La red llega a una configuración (un recuerdo) y se queda ahí para siempre ($\sigma(t+1) = \sigma(t)$).
- Ciclo límite (Periodo > 1): la red no se queda quieta, sino que .oscila.entre varios estados en un bucle infinito (por ejemplo: A $\to$ B $\to$ C $\to$ A...).

Hay varios ejemplos de dinámica paralela en la página 44 del libro de referencia.

3.1.1. Sequential dynamics

En esta sección nos centraremos en la dinámica secuencial, donde estudiaremos la interacción de simetría y las Lyapunov functions.

Proposición 3.1.1. *Given a noiseless system ($T = 0$), if interactions are symmetric ($J_{ij} = J_{ji}\ \forall(i,j)$), self-interaction are non-negative ($J_{ii} \geqslant 0\ \forall i$), and the external field h is stationary , then*

$$L_{N,J,h}(\sigma) = -\frac{1}{2}\sum_i \sum_j J_{ij}\sigma_i\sigma_j - \sum_i h_i\sigma_i$$

is a bounded ***Lyapunov function*** *for the sequential dynamics*

$$\begin{cases} \sigma_i(t+1) = sgn\left(\sum_{k=1}^{N} J_{ik}\sigma_k(t) + h_i\right) \\ \sigma_k(t+1) = \sigma_k(t) \quad \forall k \neq i \end{cases}$$

with i drawn randomly and uniformly at each step.

Demostración. □

During the iteration of the map, $L_{N,J,h}(\sigma)$ decreases monotonically and, due to its boundedness, a stage is eventually reached where $\sigma_i(t+1) = \sigma_i(t), \forall i$.

In the deterministic limit ($T = 0$), the sequential (parallel) dynamics with symmetric interactions, non-negative self-interaction (not needed for parallel dynamics) and stationary external fields will always end up in a fixed point (limit cycle with period $\leqslant 2$). Es importante notar, que en el caso secuencial siempre se acabará en un punto fijo, mientras en el caso paralelo, puede ser un dos ciclo o un punto fijo.

Observación. La función de Lyapunov $L(\sigma)$ mide la .enërgía"del sistema en una configuración dada, es decir, actúa de forma equivalente a lo que llamamos Hamiltoniano en el modelo de Ising de espines.

Observación. Tenemos dos aspectos importantes a destacar:

- La simetría de las interacciones ($J_{ij} = J_{ji}$) es crucial para que no existan corrientes de probabilidad, es decir, evoluciona hacia un estado de equilibrio estático donde la probabilidad de pasar del estado A al B es igual a la de pasar de B al A.
- La condición $J_{ii} \geqslant 0$ es necesaria para garantizar que la función de Lyapunov disminuya en cada paso. Si $J_{ii} < 0$, podría darse el caso de que la actualización de una neurona aumente la función de Lyapunov, impidiendo la convergencia a un punto fijo. Veámos un caso simple:

$$\text{si } J_{ij} = J\delta_{ij}, \quad J < 0 \Longrightarrow \begin{cases} \sigma_i(t+1) = sgn(J)\sigma_i(t) = -\sigma_i(t) \\ \sigma_k(t+1) = \sigma_k(t) \end{cases}$$

  provocando que el sistema no converja a un punto fijo, sino que oscile indefinidamente entre dos estados.

3.2. Fundamentos de la Memoria Asociativa

Una red neuronal recurrente, como la **red de Hopfield**, puede ser utilizada como una **memoria asociativa** para almacenar patrones (palabras, imágenes, etc.) mediante la creación de atractores de punto fijo en su espacio de estados.

Definición 3.2.1. Un **atractor** representa una configuración estable en el "espacio de estados" (el conjunto de todas las combinaciones posibles de las neuronas) hacia la cual el sistema evoluciona con el tiempo. En las redes de memoria asociativa, los atractores son los recuerdos almacenados: cuando el sistema recibe una entrada ruidosa o incompleta, la dinámica de la red lo .empuja"hacia el atractor más cercano para recuperar la información original.

3.2.1. Configuración del Sistema

- **Arquitectura**: Red totalmente conectada donde cada nodo es una neurona de McCulloch-Pitts (MP).
- **Parámetros Congelados (*Quenched*)**: Los acoplamientos sinápticos J y los campos externos h se mantienen constantes durante la evolución de la red.
- **Dinámica sin Ruido ($T = 0$)**: La red evoluciona de forma determinista hacia un atractor determinado por los parámetros del sistema.

A modo de resumen, trabajaremos con un modelo de red neuronal recurrente sin ruido ($T = 0$), es decir, la dinámica es determinista (no depende de ninguna variable aleatoria), donde cada nodo es una MP neurona y las interacciones J y los campos externos h son "quenched"(fijos en el tiempo).

3.2.2. Representación de Patrones (ξ)

Los elementos que deseamos almacenar se representan como vectores de N bits en un espacio bipolar:

$$\xi^\mu = (\xi_1^\mu, \xi_2^\mu, \ldots, \xi_N^\mu) \in \{-1, +1\}^N, \quad \mu = 1, \ldots, P \tag{3.1}$$

Donde P es el número total de patrones almacenados. Cada componente ξ_i^μ indica el estado deseado de la neurona i para el patrón μ.

Definición 3.2.2. Se define el solapamiento o actividad promedio $m(t)$ respecto a un patrón como la semejanza entre el estado actual $\sigma(t)$ y el patrón ξ:

$$m(t) := \frac{1}{N} \sum_{i=1}^{N} \xi_i \sigma_i(t) \tag{3.2}$$

La recuperación exitosa ocurre cuando $m(0)$ es suficientemente grande, permitiendo que la red "caiga" en la cuenca de atracción del recuerdo.

3.3. La Regla de Hebb

El **problema inverso** consiste en calcular los pesos J_{ij} para que los vectores ξ^μ sean puntos fijos estables. La solución clásica es la **regla de Hebb**: *"Neurons that fire together wire together"*, que nos permitirá definir las conexiones sinápticas (J_{ij}) en función de los patrones a almacenar (ξ).

3.3.1. Almacenamiento de un Único Patrón ($P = 1$)

Para un solo patrón ξ, definimos las conexiones como:

$$J_{ij} = \frac{1}{N} \xi_i \xi_j \tag{3.3}$$

- Si ξ_i y ξ_j tienen el mismo signo, la conexión es excitatoria ($J_{ij} > 0$).
- Si tienen signos opuestos, la conexión es inhibitoria ($J_{ij} < 0$).
- **Simetría**: Debido a la naturaleza de la regla, tanto ξ como su inverso $-\xi$ se convierten en atractores.

Si en este caso definimos $h_i = h \ \forall i$, donde $|h| < 1$, tenemos tres posibles casos:

- **Recuperación Exitosa** ($\sigma = \xi$): Si el solapamiento inicial es mayor que la intensidad del campo ($m(0) > |h|$), la red converge al patrón deseado (τ^+, donde todos los $\tau_i = 1$, donde $\tau_i = \xi_i \sigma_i$).
- **Recuperación Inversa** ($\sigma = -\xi$): Si la red empieza muy "desalineada" ($m(0) < -|h|$), converge al patrón invertido debido a la simetría del modelo.
- **Estado Dominado por el Campo**: Si el solapamiento inicial es débil ($-|h| < m(0) < +|h|$), las neuronas simplemente se alinean con el campo externo h, resultando en un estado que no corresponde al patrón original.

3.3.2. Almacenamiento de Múltiples Patrones ($P > 1$)

Para almacenar P patrones, la regla de Hebb se generaliza sumando las contribuciones de cada uno:

$$J_{ij} = \frac{1}{N}(1 - \delta_{ij}) \sum_{\mu=1}^{P} \xi_i^\mu \xi_j^\mu \tag{3.4}$$

pero para que la red funcione correctamente, se establecen tres requisitos fundamentales:

- **Ortogonalidad** ($\xi^\mu \perp \xi^\nu$): los patrones deben ser linealmente independientes u ortogonales. Matemáticamente, su solapamiento debe ser nulo

  $$\frac{1}{N} \sum_{i=1}^{N} \xi_i^\mu \xi_i^\nu = \delta_{\mu\nu} \quad \forall \mu \neq \nu$$

  Esto limita el número de patrones a $P \leq N$.

- **Sin Auto-interacción** ($J_{ii} = 0$): las neuronas no se conectan consigo mismas para evitar inestabilidades.

- **Sin Campo Externo** ($h_i = 0$): se asume que no hay un sesgo externo que favorezca un estado sobre otro, permitiendo que la red se guíe solo por sus recuerdos.

Si analizamos la expresión de los J_{ij}, vemos que el término $(1 - \delta_{ij})$ es el que garantiza que $J_{ii} = 0$.

Para el caso de múltiples patrones ortogonales, la función de energía (Lyapunov) se expresa como:

$$L(\sigma) = -\frac{1}{2} \sum_i \sum_j J_{ij} \sigma_i \sigma_j - \sum_i h_i \sigma_i = -\frac{1}{2} \sum_{i,j} \sigma_i \left[\frac{1}{N}(1 - \delta_{ij}) \sum_{\mu=1}^{P} \xi_i^\mu \xi_j^\mu \right] \sigma_j$$

donde tenemos los siguientes desarrollos

$$+ \frac{1}{2} \sum_i \sigma_i \left(\frac{1}{N} \sum_{\mu=1}^{P} \xi_i^\mu \xi_i^\mu \right) \sigma_i \overset{(*)}{=} \frac{1}{2N} \sum_i \sum_{\mu=1}^{P} 1 = \frac{1}{2N} NP = \frac{P}{2}$$

$$- \frac{1}{2} \sum_i \sum_j \sigma_i \left(\frac{1}{N} \sum_{\mu=1}^{P} \xi_i^\mu \xi_j^\mu \right) \sigma_j = -\frac{1}{2N} \sum_{\mu=1}^{P} \sum_{i,j} (\xi_i^\mu \sigma_i)(\xi_j^\mu \sigma_j) = -\frac{1}{2N} \sum_{\mu=1}^{P} \left(\sum_{i=1}^{N} \xi_i^\mu \sigma_i \right)^2$$

donde $(*)$ se cumple porque $\xi_i^\mu \in \{-1, +1\} \implies (\xi_i^\mu)^2 = 1$, y por tanto, la función de energía queda como

$$L(\sigma) = \frac{1}{2} P - \frac{1}{2} \sum_{\mu=1}^{P} \left(\frac{1}{\sqrt{N}} \sum_{i=1}^{N} \xi_i^\mu \sigma_i \right)^2$$

Ahora a partir de los patrones definimos una base de vectores normalizados

$$\hat{e}^\mu = \frac{1}{\sqrt{N}}\xi^\mu$$

y como los patrones son ortogonales ($\frac{1}{N}\sum \xi_i^\mu \xi_i^\nu = \delta_{\mu\nu}$), este conjunto $\{\hat{e}^\mu\}$ forma una base ortonormal en el espacio $\mathbb{R}^N$. Tenemos ahora la siguiente propiedad fundamental

$$x^2 \geqslant \sum_{\mu=1}^{P} (\hat{e}^\mu \cdot x)^2, \quad x \in \mathbb{R}^N$$

donde en nuestro caso el vector x es el estado actual de la red, σ. La desigualdad nos dice que la magnitud total de un vector (σ^2) siempre es mayor o igual a la suma de los cuadrados de sus proyecciones sobre un conjunto de ejes ortogonales (los patrones), de modo que podemos escribir

$$L(\sigma) = \frac{1}{2}P - \frac{1}{2}\sum_{\mu=1}^{P}(\hat{e}^\mu \cdot \sigma)^2 \geqslant \frac{1}{2}P - \frac{1}{2}\sigma^2 = \frac{1}{2}(P-N)$$

On the other hand, if we choose the state σ to be exactly one of the patterns, that is, $\sigma = \xi^\mu$ for some μ, we see that we satisfy exactly the lower bound

$$L(\pm\xi^\mu) = \frac{1}{2}(P-N)$$

Thus, all patterns and their inverses are stationary points for the dynamics because, by choosing the coupling between neurons according to Hebb's rule, the resulting Lyapunov function has configurations as minimum points $\sigma = \pm\xi^\mu \Rightarrow$ each $\pm\xi^\mu$ is a fixed point for this system.

Conclusión: esta desigualdad garantiza que ningún estado de la red puede tener menos energía que los patrones almacenados. Esto convierte a los recuerdos en "pozos"de energía donde la dinámica de la red tiende a caer y quedarse atrapada.

Observación. Es importante destacar que puede haber otros mínimos locales en la función de energía que no correspondan a los patrones almacenados. Estos mínimos adicionales pueden actuar como "falsos recuerdos"formados por mezcla de patrones, donde la red puede quedar atrapada, impidiendo la recuperación correcta de los patrones deseados.

3.4. Noisy dynamics ($T > 0$)

A diferencia de la dinámica sin ruido ($T = 0$), aquí el estado de la neurona $\sigma_i(t+1)$ no está determinado únicamente por el campo local, sino que se le suma un término de ruido aleatorio $Tz_i(t)$, es decir, tenemos

$$\sigma_i(t+1) = \text{sign}\left[\sum_{k=1}^{N} J_{ik}\sigma_k(t) + h_i + Tz_i(t)\right], \quad T > 0$$

donde T es la temperatura (representa el nivel de ruido en el sistema), y $z_i(t)$ son números aleatorios extraídos de una distribución de probabilidad $\mathcal{P}(z)$. En esta caso en particular, tomaremos $\mathcal{P}(z)$ como sigue

$$\mathcal{P}(z) = \frac{1}{2}[1 - \tanh(z)] \quad \Longrightarrow \quad g(x) = \frac{1}{2}[1 + \tanh(x)]$$

de manera que siguiendo el calculo de probabilidades que hicimos en el Tema 2, tenemos que

$$\text{parallel:} \quad P(\sigma(t+1)) = \prod_{i=1}^{N} g\left(\sigma_i(t+1)\frac{\varphi_i(t)}{T}\right) = \prod_{i=1}^{N} \frac{1}{2}\left[1 + \sigma_i(t+1)\tanh\left(\beta\varphi_i(\sigma(t))\right)\right]$$

$$\text{sequential:} \quad P[\sigma_i(t+1)] = g\left(\sigma_i(t+1)\frac{\varphi_i(t)}{T}\right) = \frac{1}{2}\left[1 + \sigma_i(t+1)\tanh\left(\beta\varphi_i(\sigma(t))\right)\right] \ \forall i$$

donde $\beta = T^{-1}$, $\varphi_i(\sigma) = \sum_{j=1}^{N} J_{ij}\sigma_j + h_i$ es el campo local en la neurona i, y se ha usado la propiedad de la función tangente hiperbólica $\tanh(ax) = a\tanh(x)$ para $a \in \mathbb{R}$.

3.4.1. Markov Chains (MCs)

MC is a model of the random evolution of a system in a discrete set $\mathcal{S}$ of possible states. There are two versions:

- **Discrete time**: the state of the MC is recorded every unit of time, that is, at times $0, 1, 2$, and so on.
- **Continuous time**: the state is observed at all times $t \geqslant 0$.

Definición 3.4.3 (Markov property). Sea un proceso estocástico $\{X_t, t = 0, 1, 2, ...\}$ que toma un número finito o numerable de valores posibles. Se dice que tiene la **propiedad de Markov** si estando en el estado $i \in \mathcal{S}$ en el tiempo t si $X_t = i$; se asume que existe una probabilidad fija W_{ij} de que si el proceso está en el estado i pase al estado j en el siguiente paso, lo que viene dado por

$$P(X_{t+1} = j | X_t = i, X_{t-1} = i_{t-1}, \ldots, X_0 = i_0) = P(X_{t+1} = j | X_t = i) = W_{ij}$$

Esto equivale a decir que la probabilidad de transición depende únicamente del estado actual y no de los estados anteriores, lo que caracteriza la **propiedad de Markov**.

Definición 3.4.4 (Cadena de markov). Una **cadena de Markov** es un proceso estocástico $\{X_t, t = 0, 1, 2, ...\}$ con la propiedad de Markov y un conjunto finito o numerable de estados posibles $\mathcal{S}$. La dinámica de la cadena está completamente determinada por la **matriz de transición** $W \in [0, 1]^{|\mathcal{S}|\times|\mathcal{S}|}$, donde W_{ij} representa la probabilidad de pasar del estado i al estado j en un solo paso de tiempo.

Como un proceso debe hacer una transición a algún estado, se cumple que

$$\sum_{j \in \mathcal{S}} W_{ij} = 1 \quad \forall i$$

No todas las cadenas se comportan igual. Para que una red neuronal recuerde"de forma estable bajo ruido, la cadena debe tener ciertas propiedades:

- **Homogeneous**: If the transition probabilities W do not change over time.
- **Irreducible**: If it is possible to reach any state from any other (the state graph is fully connected).
- **Ergodic**: This is the most important property. A chain is ergodic if it is irreducible, aperiodic (it has no fixed cycles) and positive recurrent (it returns to states in finite time).

Definición 3.4.5 (Homogeneity). A MC is referred to as **homogeneous** if W is independent of time, then:

$$Prob[X_{m+1} = j | X_m = i] = Prob[X_1 = j | X_0 = i] = W_{ij}$$

$$Prob[X_{m+n} = j | X_m = i] = W_{ij}^m$$

Definición 3.4.6 (Irreducibility). A MC is **irreducible** when all its states communicate. When representing the set of states by a graph, where two nodes are connected iff there exists a non-null probability to move from state to the other, this property equals to say that the graph is connected.

Definición 3.4.7 (Absorbing). A MC is said **absorbing** if one of its states is absorbing (i.e., it is impossible to leave once it has been entered $P_{i \neq j} W_{ij} = 0$) and from a non-absorbing state we can reach an absorbing state.

Definición 3.4.8 (Periodicity). A state i is said to have a **period** k if

$$k = gcd\{t > 0 : Prob[X_t = i | X_0 = i] > 0\}$$

Further, A MC is **aperiodic** if every state is not periodic.

Definición 3.4.9 (Transience and recurrence). A state i is **transient** if, starting from i, there is a non-zero probability that the chain will never return to i. It is **recurrent** otherwise, and **positive recurrent** if the mean time to return in i is finite.

Definición 3.4.10 (Ergodicity). A MC is **ergodic** if there exists a positive integer τ such that for all pairs of states $(i, j) \in \mathcal{S} \times \mathcal{S}$, then

$$Prob[X_t = j | X_0 = i] > 0 \quad \forall t > \tau$$

Irreducibile MC whose states are aperiodic and positive recurrent is ergodic.

Definición 3.4.11 (Invariance). A distribution $p(X|X_0)$ over the possible states that can be taken by the MC is **invariant** if $p(X|X_0)W = p(X|X_0)$.

Teorema 3.4.2. *For any ergodic MC, there is an unique invariant distribution* $p_\infty(X)$ *that is the principal left eigenvector of* W*, such that if* $\nu(i,t)$ *is the number of visits to state* i *in* t *steps,*

$$\lim_{t\to\infty} \frac{\nu(i,t)}{t} = p_\infty$$

Lo que equivale a decir que, independientemente del estado inicial, la distribución de probabilidad sobre los estados converge a p_∞ *conforme* $t \to \infty$*.*

Definición 3.4.12 (Detailed balance). A stochastic matrix W and a measure p(X) are said to be **in detailed balance** if

$$p(X = i)W_{ij} = p(X = j)W_{ji} \quad \forall i, j$$

En dinámicas secuenciales, si la matriz de pesos es simétrica ($J_{ij} = J_{ji}$), el sistema cumple automáticamente con el balance detallado.

3.4.2. Distribución de Boltzmann

Este es el concepto más importante para entender por qué la red sigue funcionando con ruido:

- La probabilidad de encontrar la red en una configuración σ en el equilibrio es:

$$p_\infty(\sigma) \propto e^{-\mathcal{H}(\sigma)/T}$$

- ¿Qué significa esto? Que los estados con menor energía (donde $\mathcal{H}$ es mínimo) son los más probables. Como diseñamos la red (vía Hebb) para que los patrones guardados sean mínimos de energía, el ruido no impide que la red pase la mayor parte del tiempo en esos recuerdos.

¿Por qué no se pierde el recuerdo? Cuando la red es muy grande ($N \to \infty$), ocurre lo que se llama Ruptura de Ergodicidad o Confinamiento:

- Las "montañas"de energía entre un recuerdo y otro se vuelven tan altas que el ruido no es suficiente para que la red salte de un patrón a otro.
- El sistema se queda .ªtrapado.en el valle del patrón más cercano al input inicial, lo que permite una recuperación robusta.

4. Basics on Statistical Mechanics of simple systems

4.1. Curie-Weiss Model

El modelo de Curie-Weiss describe un sistema de N "spins"(que en nuestro ámbito representarán neuronas) que pueden estar en dos estados: $\sigma_i \in \{+1, -1\}$, equivalentes a una neurona activa o en reposo.

La característica clave del modelo de Curie-Weiss es que es un modelo de campo medio (Mean Field). Esto significa que cada neurona interactúa con todas las demás de la red con la misma intensidad, sin importar la distancia o la geometría. Es lo que llamamos un "grafo completo".

Definición 4.1.1 (Hamiltoniano del Modelo de Curie-Weiss). El **Hamiltoniano** (Función de Energía) define la energía total del sistema para una configuración dada de estados (σ). En el modelo CW, se define como:

$$\mathcal{H}_{N,J,h}(\sigma) = -\sum_{i,j} \sigma_i J_{ij} \sigma_j - \sum_{i=1}^{N} h_i \sigma_i$$

donde $J_{ij} = \dfrac{J}{N}$ para todo par (i, j), es decir, todas las neuronas están igualmente conectadas con un peso J/N, y $h_i = h \ \forall i$, por lo que otra definición equivalente es:

$$\mathcal{H}_{N,J,h}(\sigma) = -\frac{J}{N} \sum_{i>j} \sigma_i \sigma_j - h \sum_{i=1}^{N} \sigma_i$$

notar que el primer sumatorio debería ser el doble, pues solo cuenta cada par una vez y no las autointeracciones (de ahí la condición $i > j$), pero aún así es una definición válida. Las características son las siguientes:

- J: Es la constante de acoplamiento. Si $J > 0$, el sistema es ferromagnético: las neuronas prefieren estar alineadas (todas activas o todas inactivas) porque eso minimiza la energía.
- h: Representa un campo externo (un estímulo sensorial o sesgo) que empuja a las neuronas hacia un estado específico.

- $1/N$: Es un factor de escala necesario para que la energía sea .ªextensiva"(que crezca linealmente con el número de neuronas).

Para simplificar el análisis de N variables, introducimos una variable macroscópica llamada **magnetización** (m), que mide el promedio de actividad de la red

$$m_N(\sigma) = \frac{1}{N}\sum_{i=1}^{N}\sigma_i$$

que toma valores en $M_N = \{1 - 2i/N, i = 1, ..., N\}$, donde si $m_N(\sigma) = 1 - 2i/N$ entonces la configuración σ tiene i espines hacia abajo y $N - i$ hacia arriba. Ahora podemos reescribir el Hamiltoniano en términos de la magnetización

$$\mathcal{H}_{N,J,h}(\sigma) = -\frac{NJ}{2}[m_N(\sigma)]^2 + -hNm_N(\sigma) + \frac{J}{2} = N\phi_{J,h}(m_N(\sigma))$$

donde lo que hemos hecho es lo siguiente:

$$\left(\sum_{i=1}^{N}\sigma_i\right)^2 = \sum_{i=1}^{N}\sum_{j=1}^{N}\sigma_i\sigma_j \overset{(*)}{\implies} (Nm_N(\sigma))^2 = \sum_{i=1}^{N}\sigma_i^2 + 2\sum_{i>j}\sigma_i\sigma_j \implies$$
$$\implies \sum_{i>j}\sigma_i\sigma_j = \frac{(Nm_N(\sigma))^2 - N}{2} = \frac{N^2[m_N(\sigma)]^2 - N}{2}$$

donde en $(*)$ hemos usado que cuando $i = j$, $\sigma_i^2 = 1$, por lo que $\sum_{i=1}^{N}\sigma_i^2 = N$, y para pares $i \neq j$ contamos cada par dos veces por ser simétrica ($\sigma_i\sigma_j$ y $\sigma_j\sigma_i$), por lo que hay exactamente el doble de términos que en el sumatorio $i > j$, es decir $2\sum_{i>j}\sigma_i\sigma_j$.
Finalmente sustituimos en el Hamiltoniano el término $\sum_{i>j}\sigma_i\sigma_j$, de la siguiente forma:

$$\mathcal{H}_{N,J,h}(\sigma) = -\frac{J}{N}\sum_{i>j}\sigma_i\sigma_j - h\sum_{i=1}^{N}\sigma_i = -\frac{J}{N}\cdot\frac{N^2[m_N(\sigma)]^2 - N}{2} - hNm_N(\sigma) =$$
$$= -\frac{NJ}{2}[m_N(\sigma)]^2 - hNm_N(\sigma) + \frac{J}{2}$$

Como es lógico J puede ser tanto positiva como negativa, pero en el contexto de redes neuronales y sistemas biológicos, generalmente se asume que $J > 0$ (**ferromagneto**), ya que las neuronas tienden a activar a otras neuronas similares (excitación mutua) en lugar de inhibirse entre sí, es decir, si la neurona i está encendida (+1), empuja a su vecina j a encenderse también para que el producto $\sigma_i\sigma_j$ sea positivo. Esto minimiza el término $-J\sigma_i\sigma_j$ en el Hamiltoniano. En redes neuronales, esto es lo que permite crear atractores (recuerdos estables).

El caso $J < 0$ correspondería a un sistema antiferromagneto, donde las neuronas tienden a estar en estados opuestos (una activa y la otra inactiva). Este tipo de interacción es menos común en modelos de redes neuronales, pero podría representar situaciones donde la inhibición mutua es dominante.

El modelo de Curie-Weiss, usa como ya hemos comentado $J > 0$. Este modelo no nos indicará cómo evolucionan las neuronas en el tiempo, sino que nos dará información sobre las configuraciones de equilibrio del sistema, es decir, cuáles son las configuraciones más probables de encontrar en función de los parámetros J, h y la temperatura T, mediante la siguiente distribución de probabilidad

$$\rho_{N,J,h,\beta}(\sigma) = \frac{e^{-\beta \mathcal{H}_{N,J,h}(\sigma)}}{Z_{N,J,h,\beta}} \quad \text{tal que} \quad Z_{N,J,h,\beta} = \sum_{\sigma} e^{-\beta \mathcal{H}_{N,J,h}(\sigma)}$$

donde tenemos que

- La expresión $e^{-\beta H}$ representa la **exponencial de Boltzmann** que es la lógica central. Los estados con menor energía (H muy negativo) tienen una probabilidad mucho más alta de ocurrir. La red "cae" en los valles de energía (que son nuestros patrones memorizados).
- La función Z es la **función de Partición**, que matemáticamente es solo una constante de normalización (la suma de todos los pesos exponenciales posibles) para que la probabilidad total sume 1.

4.1.1. Thermodynamic observables

A continuación veremos algunas funciones importantes que nos ayudarán a entender el comportamiento del sistema

- La **Energía Libre** (F) y la **Energía Libre Intensiva** (f), dadas por

  $$F_{N,J,h,\beta} = -\frac{1}{\beta} \log Z_{N,J,h,\beta} \quad \text{y} \quad f_{N,J,h,\beta} = \frac{F_{N,J,h,\beta}}{N}$$

  donde la energía libre F mide el equilibrio entre energía y entropía (desorden) del sistema, y la versión intensiva f nos da esta medida "por neurona", útil para comparar sistemas de diferentes tamaños.

  Cuando hablamos de que la energía libre y libre intesiva, representan el equilibrio entre energía y entropía, nos referimos a que el sistema siempre busca minimizar su energía (bajar H) buscando estados ordenados (como todos los espines hacia arriba o hacia abajo), pero al mismo tiempo, la entropía (que mide el desorden) tiende a maximizarse, ya que aunque estos estados desordenados tienen baja energía, son muchos en la cantidad, por lo que al sumar sobre todas las configuraciones, la estadística empieza a ganar por çantidad"lo que pierde por çalidad"(energía).

- La **Energía Libre en el Límite Termodinámico** ($f_{\beta,J,h}$)

  $$f_{\beta,J,h} = \lim_{N \to \infty} f_{N,J,h,\beta}$$

  que es el valor de la energía libre cuando tenemos una red "infinita". En matemáticas de redes neuronales, solo cuando $N \to \infty$ aparecen los fenómenos de fases (como el paso de "no recordar nada" a "recordar perfectamente"). Estudiamos este límite porque las redes reales tienen miles de millones de conexiones y se comportan de forma muy similar al límite infinito.

- La **Presión Intensiva** ($A_{N,J,h,\beta}$), dada por

$$A_{N,J,h,\beta} = \frac{1}{N} \log Z_{N,J,h,\beta}$$

es simplemente otra forma de escribir la energía libre (fíjate que $A_{N,J,h,\beta} = -\beta f_{N,J,h,\beta}$). Se suele usrr mucho porque es más manejable para usar técnicas como el "interpolatión trick" o el "replica trick". Representa lo mismo: el estado de equilibrio del sistema.
Al igual que antes, definimos la **Presión Intensiva en el Límite Termodinámico** ($A_{\beta,J,h}$)

$$A_{\beta,J,h} = \lim_{N\to\infty} A_{N,J,h,\beta}$$

- El **Promedio Térmico de un Observable** ($\omega_{N,J,h,\beta}$), dada por

$$\omega_{N,J,h,\beta}(g) = \sum_{\sigma} g(\sigma)\rho_{N,J,h,\beta}(\sigma)$$

es una función que nos permite calcular el valor esperado de cualquier cosa que queramos medir (g) en la red bajo la distribución de probabilidad $\rho_{N,J,h,\beta}$. Viene a ser esperanza ponderada por g y la probabilidad P.
Si quieres saber, por ejemplo, cuál es la magnetización promedio (m) de la red, no miras una foto fija, sino que haces un promedio pesado por la probabilidad P de todos los estados. En nuestros apuntes, veremos que a menudo se escribe como $\langle g \rangle$. Es lo que se vería si observaramos la red durante mucho tiempo bajo un nivel de ruido constante (temperatura fija).

Ahora estudiaremos la función de partición $Z_{N,J,h,\beta}$ para reducir la sumatoria, pues si tenemos una red neuronal con muchas neuronas, es decir, N grande, la sumatoria se vuelve inabarcable, ya que existen 2^N configuraciones posibles. Para ello, usaremos la variable macroscópica magnetización $m_N(\sigma)$ que ya hemos definido antes. Notar que muchas configuraciones diferentes σ pueden tener la misma magnetización m, por lo que podemos reagrupar la sumatoria en Z según los valores de m, veámoslo

$$Z_{N,J,h,\beta} = \sum_{\sigma} e^{-\beta \mathcal{H}_{N,J,h}(\sigma)} = \sum_{m\in M_N} e^{-\beta \mathcal{H}_{N,J,h}(\sigma)} \Omega_N(m)$$

donde $\Omega_N(m)$ es el número de configuraciones σ que tienen una magnetización m, es decir, la **densidad de estados** con magnetización m. Esta cantidad se puede calcular combinatoriamente, ya que si $m = 1 - 2i/N$, entonces hay i espines hacia abajo y $N - i$ hacia arriba, por lo que

$$\Omega_N(m) = \binom{N}{i} = \binom{N}{\frac{N(1-m)}{2}}$$

usando la definición del coeficiente binomial (más adelante veremos la demostración de la segunda igualdad). A continuación, definiremos la entropía $S_N(m)$ como el logaritmo natural de la densidad de estados

$$S_N(m) = \log \Omega_N(m)$$

teniendo por tanto

$$Z_{N,J,h,\beta} = \sum_{m \in M_N} e^{-\beta \mathcal{H}_{N,J,h}(\sigma)} e^{S_N(m)} = \sum_{m \in M_N} e^{\beta\left[-\mathcal{H}_{N,J,h}(\sigma)+TS_N(m)\right]} = \sum_{m \in M_N} e^{-\beta F_N(m)}$$

donde hemos definido la **Energía Libre de Helmholtz** ($F_N(m)$) como

$$F_N(m) = \mathcal{H}_{N,J,h}(\sigma) - TS_N(m)$$

que mide el equilibrio entre energía y entropía para una magnetización m dada. La red neuronal vive en una competencia constante. La energía quiere orden y memoria, mientras que la entropía (potenciada por la temperatura T, es decir, $T \cdot S$) quiere caos y ruido. El sistema siempre buscará el estado m que minimice esta competencia (F).

4.1.2. Saddle Point Method(or Laplace's Method)

Lo que intentaremos buscar en esta resolución es encontrar a partir de que valor T (ruido) la red deja de recordar el patrón almacenado. Para ello, usaremos el **Método del Punto de Silla** (o **Método de Laplace**), que nos permite aproximar sumas o integrales de funciones exponenciales cuando N es grande. La idea clave es que, para N grande, la suma está dominada por el término donde el exponente es máximo (o mínimo si es negativo).

En primer lugar, tenemos que la probabilidad de observar una magentización especifica $\tilde{m}$ en la red es

$$\rho_N(\tilde{m}) = \sum_{\sigma} \rho_N(\sigma) \delta(\tilde{m} - m_N(\sigma))$$

donde δ es la delta de Kronecker, que selecciona solo las configuraciones σ que tienen magnetización $m_N(\sigma) = \tilde{m}$. Al final, $\rho_N(\tilde{m})$ es simplemente la suma de las probabilidades de todas aquellas configuraciones que "votan"por el mismo valor de memoria.

En lo siguiente, utilizaremos la notación m para referirnos a una magnetización genérica $\tilde{m}$, y $m_N(\sigma)$ para la magnetización específica de una configuración σ.

Ahora, veremos una igualdad importante que nos permitirá simplificar la expresión de $p_N(m)$, usando una función cualquiera g, veámoslo

$$\sum_{m \in M_N} \rho_N(m) g(m) = \sum_{m \in M_N} g(m) \sum_{\sigma} \rho_N(\sigma) \delta(m - m_N(\sigma)) = \sum_{\sigma} \rho_N(\sigma) \sum_{m \in M_N} g(m) \delta(m - m_N(\sigma)) =$$
$$\overset{(*)}{=} \sum_{\sigma} \rho_N(\sigma) g(m_N(\sigma)) = \omega_N(g(m_N(\sigma)))$$

donde en $(*)$ hemos usado la propiedad de la delta de Kronecker, que selecciona el valor $m = m_N(\sigma)$ en el sumatorio, de manera que $g(m)$ solo contribuye cuando $m = m_N(\sigma)$.

Calculemos finalmente $\rho_N(m)$ usando la igualdad anterior con $g(m) = \delta(m - \hat{m})$, veámoslo

$$\rho_N(m) = \sum_{\sigma} \rho_N(\sigma)\delta(m - m_N(\sigma)) \overset{(*)}{=} \sum_{\sigma} \left[\frac{1}{Z_N} e^{-\beta H_N(\sigma)}\right] \delta(m - m_N(\sigma)) = \frac{Z_n(m)}{Z_N}$$

donde en $(*)$ hemos usado la forma de Boltzmann de la distribución de probabilidad $\rho_N(\sigma)$, es decir

$$\rho_N(\sigma) = \frac{e^{-\beta H_N(\sigma)}}{Z_N}$$

Tenemos por tanto que

$$Z_N(m) = \sum_{\sigma} \delta(m - m_N(\sigma)) e^{-\beta H_N(\sigma)}$$

es la **función de partición restringida** a una magnetización m. Además

$$\rho_N(m) = \frac{Z_N(m)}{Z_N} = e^{-\beta(F_N(m) - F_N)}$$

donde

$$F_N(m) = -\frac{1}{\beta} \log Z_N(m)$$

es la **energía libre restringida** a una magnetización m. Análogamente, definimos la **energía libre intensiva restringida** como

$$f_N(m) = \frac{F_N(m)}{N}$$

Antes de avanzar es importante tener claro el significado físico de estas cantidades:

- $Z_N(m)$: Nos dice cuánto "peso"tienen todas las configuraciones con magnetización m. Es como si filtráramos todas las posibles configuraciones y solo miráramos aquellas que tienen un nivel específico de memoria (magnetización).

- $F_N(m)$: Mide el equilibrio entre energía y entropía para las configuraciones con magnetización m. Nos dice qué tan "favorable.es ese nivel de memoria en términos de energía libre.

- $\rho_N(m)$: Es la probabilidad de encontrar la red en un estado con magnetización m. Nos indica qué tan probable es que la red recuerde un patrón con un cierto nivel de fidelidad.

Observación. Cuando hablamos en este concepto de una función "restringida" nos referimos a que hemos limitado la consideración a un subconjunto específico de todas las configuraciones posibles del sistema, en este caso, aquellas que tienen una magnetización fija m. Esto nos permite analizar el comportamiento del sistema bajo condiciones específicas, en lugar de considerar todas las configuraciones posibles de manera indiscriminada.

A partir de ahora, no necesitaremos calcular Z_N directamente, es decir, todas las configuraciones posibles, sino que nos centraremos en $Z_N(m)$, que es mucho más manejable. La idea de esto, es que podemos representar cuánta importancia tiene un valor concreto de memoria para el sistema, por lo que si $Z_N(0{,}8)$ es mucho más grande que $Z_N(0)$, la red nos está diciendo que "prefiere" con mucha diferencia estar en un estado de alta memoria que en uno de ruido (recordemos que el caso $m = 0$ es ruido total, sin memoria).

Estudiemos ahora el caso más sencillo donde $f_N(m)$ tiene un único mínimo global en $m = m^*$. en este caso, tomando el desarrollo de taylor de $f_N(m)$ alrededor de m^*, tenemos

$$f_N(m) \approx f_N(m^*) + f'_N(m^*)(m - m^*) + \frac{1}{2} f''_N(m^*)(m - m^*)^2 + \dots$$

donde como m^* es un mínimo, $f'_N(m^*) = 0$, por lo que

$$f_N(m) \approx f_N(m^*) + \frac{1}{2} f''_N(m^*)(m - m^*)^2$$

usando esta aproximación en la expresión de $\rho_N(m)$, tenemos

$$\rho_N(m) = e^{-\beta(F_N(m)-F_N)} = e^{-\beta N(f_N(m)-f_N)} \approx e^{-\beta N\left(f_N(m^*)+\frac{1}{2}f''_N(m^*)(m-m^*)^2-f_N\right)} =$$

$$= e^{-\beta N(f_N(m^*)-f_N)} \cdot e^{-\frac{\beta N}{2} f''_N(m^*)(m-m^*)^2} = Constante + e^{-\frac{(m-m^*)^2}{2\sigma^2}}$$

donde la segunda parte es una función gaussiana centrada en m^* con varianza $\sigma^2 = \dfrac{1}{\beta N f''_N(m^*)}$. Tenemos por tanto que la función de probabilidad $\rho_N(m)$ se comporta como una distribución normal alrededor del valor de magnetización m^* que minimiza la energía libre intensiva restringida $f_N(m)$.

Para $N \to \infty$, la varianza se vuelve muy pequeña ($\sigma^2 \to 0$), lo que significa que la distribución $\rho_N(m)$ se concentra cada vez más alrededor de m^*. En el límite termodinámico, la red neuronal "elige" un valor específico de magnetización m^* que minimiza la energía libre, y las fluctuaciones alrededor de este valor se vuelven insignificantes. Esto refleja el comportamiento típico de los sistemas en equilibrio termodinámico, donde las propiedades macroscópicas (como la magnetización) se estabilizan en valores específicos determinados por las condiciones del sistema (temperatura, interacción, etc.).

Antes para calcular Z_N necesitabamos la suma sobre todas las configuraciones posibles σ (es decir, 2^N términos). Ahora, gracias a relacionar la probabilidad ρ_N con una distribución gaussiana alrededor del mínimo de $f_N(m)$, podemos aproximar Z_N usando una integral, veámoslo

$$Z_N = \sum_{m \in M_N} Z_N(m) = \sum_{m \in M_N} e^{-N\beta f_N(m)} \overset{N\to\infty}{\approx} \int dm\ e^{-\beta F_N(m)} = \int dm\ e^{-\beta N f_N(m)} \approx$$

$$\overset{(*)}{\approx} \int dm\ e^{-\beta N\left(f_N(m^*)+\frac{1}{2}f''_N(m^*)(m-m^*)^2\right)} = e^{-\beta N f_N(m^*)} \int dm\ e^{-\frac{\beta N}{2} f''_N(m^*)(m-m^*)^2} =$$

$$\overset{(**)}{=} e^{-\beta N f_N(m^*)} \sqrt{\frac{2\pi}{\beta N f''_N(m^*)}}$$

donde en $(*)$ hemos usado la aproximación de Taylor de $f_N(m)$ alrededor de su mínimo m^* y la integral gaussiana estándar. Es importante destacar que al tomar N grande (cuando hacemos $N \to \infty$), estamos pasando de una suma discreta a una integral continua, lo que es lógico, pues tenemos ahora una función de probabilidad continua alrededor del mínimo. Lo que nos podríamos preguntar es porque si $m \in [-1, 1]$, la integral va de $-\infty$ a ∞. La respuesta es que para N grande, la función gaussiana está muy concentrada alrededor de m^* (desviación muy pequeña), por lo que los valores fuera del intervalo $[-1, 1]$ contribuyen muy poco a la integral, y podemos extender los límites sin afectar significativamente el resultado. Recalcar que en $(**)$ hemos usado la fórmula de la integral gaussiana

$$\int_{-\infty}^{+\infty} e^{-a(x+b)^2} dx = \sqrt{\frac{\pi}{a}} \quad a \in R^+,\ b \in R$$

Por tanto, la energía libre por neurona es

$$f_N = -\frac{1}{\beta N} \log Z_N \approx f_N(m^*) - \frac{\log\left(\frac{2\pi}{\beta f_N''(m^*)}\right) - \log(N)}{2N\beta} \overset{N\to\infty}{\longrightarrow} f(m^*)$$

así que en el límite termodinámico, la energía libre intensiva f_N converge al valor mínimo de la energía libre intensiva restringida $f_N(m)$, es decir, $f(m^*)$. Esto significa que el comportamiento macroscópico del sistema está dominado por la configuración de magnetización que minimiza la energía libre.

Nuestro objetivo ahora es claro, evaluar $f(m)$ y extremizarlo, es decir, buscar su mínimo. Veámos ambos desarrollos por separado, tomando $h = 0$ (para ver más sencillo los resultados), pues de esta manera los dos estados de memoria (todos +1 o todos -1) tienen la misma energía, y es la red la que "decide" en cuál atraparse.

Cálculo de $f(m)$. Usando la definición de $f(m)$, tenemos

$$f(m) = \lim_{N\to\infty} f_N(m) = -\lim_{N\to\infty} \frac{T}{N} \log Z_N(m)$$

Ahora, tenemos que calcular $Z_N(m)$, veámoslo

$$Z_N(m) = e^{\frac{\beta J N m^2}{2}} \Omega_N(m) \quad \text{donde} \quad \Omega_N(m) := \sum_{\sigma} \delta(m - m_N(\sigma))$$

donde es importante notar que el término $\frac{J}{2}$ del Hamiltoniano lo hemos obviado, pues al ser constante no afecta a la dinámica del sistema, cuando N es grande. Por tanto, tenemos que

$$\begin{aligned} f(m) &= -\lim_{N\to\infty} \frac{T}{N} \log Z_N(m) = -\lim_{N\to\infty} \frac{T}{N} \log\left[e^{\frac{\beta J N m^2}{2}} \Omega_N(m)\right] = \\ &= -\lim_{N\to\infty} \frac{T}{N} \left[\frac{\beta J N m^2}{2} + \log \Omega_N(m)\right] = -\frac{J m^2}{2} - T s(m) \end{aligned}$$

donde

$$s(m) := \lim_{N\to\infty} \frac{S_N(m)}{N} := \lim_{N\to\infty} \frac{1}{N} \log \Omega_N(m)$$

que es la entropía de Boltzmann por neurona asociada a la magnetización m. La primera parte se calcula facilmente del Hamiltoniano, y la segunda parte es la entropía que mide el desorden asociado a tener una magnetización m. Asumiremos que $N = N^+ + N^-$, donde N^+ es el número de neuronas activas ($\sigma_i = +1$) y N^- el número de neuronas inactivas ($\sigma_i = -1$), entonces

$$\Omega_N(m) = \binom{N}{N^+}$$

donde $N^+ = \dfrac{N(1+m)}{2}$ y $N^- = \dfrac{N(1-m)}{2}$, ya que siendo $m = \frac{N^+ - N^-}{N}$, deducimos que

$$(N^+ - N^-) + (N^+ + N^-) = Nm + N \implies 2N^+ = N(1+m) \implies N^+ = \frac{N(1+m)}{2}$$

Usando la aproximación de Stirling para factoriales grandes, que dice lo siguiente

$$\log x! \approx x \log x \qquad \text{para } x \text{ grande}$$

podemos aproximar $\log \Omega_N(m)$ de la siguiente forma

$$\begin{aligned}
\log \Omega_N(m) = \log \binom{N}{N^+} = \log \frac{N!}{N^+!N^-!} &\approx \log N! - \log N^+! - \log N^-! \approx \\
&\overset{(*)}{\approx} N \log N - N^+ \log N^+ - N^- \log N^- = \\
&= N \log N - \frac{N(1+m)}{2} \log \frac{N(1+m)}{2} - \frac{N(1-m)}{2} \log \frac{N(1-m)}{2} = \\
&= -N \left[\frac{1+m}{2} \log \left(\frac{1+m}{2} \right) + \frac{1-m}{2} \log \left(\frac{1-m}{2} \right) \right]
\end{aligned}$$

donde en $(*)$ hemos aplicado la aproximación de Stirling a cada término, y finalmente tenemos que

$$s(m) = -\frac{1+m}{2} \log \left(\frac{1+m}{2} \right) - \frac{1-m}{2} \log \left(\frac{1-m}{2} \right)$$

Realmente lo que hemos hecho aquí es calcular la entropía de una distribución binomial con parámetros N y $p = \frac{1+m}{2}$, que es la probabilidad de que una neurona esté activa. La entropía mide la incertidumbre o el desorden en la distribución de estados de la red. Es importante destacar que hemos puesto $=$ en lugar de $\approx$, pues en el límite $N \to \infty$, la aproximación de Stirling se vuelve exacta para el cálculo de la entropía por neurona.

Extremización de $f(m)$. Ahora que tenemos $f(m)$, buscaremos su mínimo derivando e igualando a cero, pero primero veámos lo que hemos conseguido hasta el momento

$$f(m) = -\frac{Jm^2}{2} + T \left[\frac{1+m}{2} \log \left(\frac{1+m}{2} \right) + \frac{1-m}{2} \log \left(\frac{1-m}{2} \right) \right]$$

Antes de seguir con este desarrollo debemos entender físicamente qué representa cada término

- El término $-\frac{Jm^2}{2}$ representa la energía interna del sistema. Este término favorece estados con alta magnetización (es decir, m cercano a $+1$ o -1), ya que minimiza la energía cuando las neuronas están alineadas (todas activas o todas inactivas). En el contexto de redes neuronales, esto corresponde a la capacidad de la red para recordar patrones específicos.
- El término $Ts(m)$ representa la contribución de la entropía al equilibrio del sistema. La entropía mide el desorden o la incertidumbre en la distribución de estados de la red. Se busca una baja entropía (es decir, estados más ordenados) para maximizar la memoria, pero a medida que la temperatura T aumenta, la entropía se vuelve más significativa, favoreciendo estados con menor magnetización (más desordenados).

  Es importante notar $s(m)$ es negativo, lo cual es totalmente lógico, pues como bien hemos comentado si T es pequeña la batuta la lleva el término de energía, pero si T es grande, la batuta la lleva el término de entropía, y como queremos minimizar $f(m)$, el término de entropía debe ser negativo para que al multiplicarlo por T (positivo) pueda crecer y dominar sobre el término de energía.

Otros aspectos que debemos tener en cuenta es ciertas propiedades de $f(m)$

- $f(m)$ es una función par, es decir, $f(m) = f(-m)$. Esto refleja la simetría del sistema, ya que no hay preferencia entre recordar un patrón o su inverso (todas neuronas activas o todas inactivas), pues en el Hamiltoniano no hay campo externo ($h = 0$), y por tanto ambos estados tienen la misma energía.
- Si $J < 0$ (antiferromagneto), el término de energía favorece estados con baja magnetización (es decir, m cercano a 0), ya que minimiza la energía cuando las neuronas están desalineadas (mezcla de activas e inactivas). En este caso, la red no puede formar patrones estables y tiende a un estado desordenado (ver panel derecho de la Figura 4.1).

- Si $J > 0$ (ferromagneto), el término de energía favorece estados con alta magnetización (es decir, m cercano a $+1$ o -1), ya que minimiza la energía cuando las neuronas están alineadas (todas activas o todas inactivas). En este caso, la red puede formar patrones estables y recordar información.
  En este caso si T es muy alta, el término de entropía domina, favoreciendo estados con baja magnetización (más desordenados). Sin embargo, si T es baja, el término de energía domina, favoreciendo estados con alta magnetización (más ordenados), creando dos estados de recuerdo estables el buscado y su inverso (ver panel izquierdo de la Figura 4.1).

Dejando esto claro, buscaremos derivar $f(m)$ e igualar a cero para encontrar sus puntos críticos, por lo que para ellos usaremos la propiedad siguiente

$$x\log x \implies \frac{d}{dx}(x\log x) = \log x + 1 \implies \frac{d}{dm}\left(\frac{1+m}{2}\log\left(\frac{1+m}{2}\right)\right) = \frac{1}{2}\left[\log\left(\frac{1+m}{2}\right) + 1\right]$$

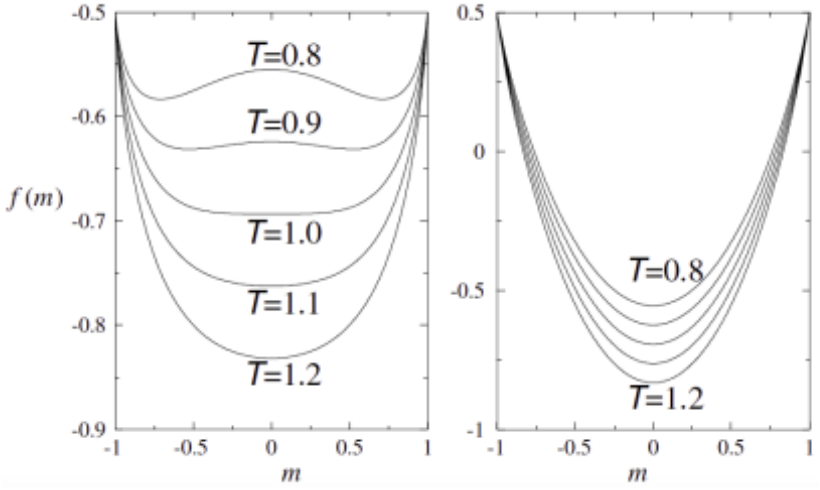


Figura 4.1: Comportamiento de la función de energía libre intensiva restringida $f(m)$ en función de la magnetización m para diferentes valores de la temperatura T y el parámetro de interacción J. Izquierda: $J = 1$. Derecha: $J = -1$.

y aplicandolo análogamente al segundo término de la entropía $s(m)$, tenemos

$$f'(m) = -Jm + \frac{T}{2}\left[\log\left(\frac{1+m}{2}\right) - \log\left(\frac{1-m}{2}\right)\right] =$$
$$= -Jm + \frac{T}{2}\log\left(\frac{1+m}{1-m}\right) = 0 \implies \frac{1}{2}\log\left(\frac{1+m}{1-m}\right) = \frac{Jm}{T} = J\beta m$$

Ahora usaremos la siguiente identidad que define la tangente hiperbólica inversa (atanh) para simplificar la expresión anterior

$$\operatorname{atanh}(x) = \frac{1}{2}\log\left(\frac{1+x}{1-x}\right)$$

y por tanto tenemos la ecuación de estado

$$\operatorname{atanh}(m^*) = J\beta m^* \implies m^* = \tanh(J\beta m^*)$$

Esta es la **ecuación de auto-consistencia** del problema, también llamada la ecuación de Curie-Weiss en el contexto del magnetismo.

Ahora analizaremos las soluciones para $h \neq 0$, pues antes habíamos impuesto dicha restricción para ver ciertas propiedades de la red, como por ejemplo que es par y por tanto simétrica, es decir, la elección entre el patrón y su inverso es equiprobable, y no venía afectada por un campo externo.

Volviendo a la energía libre de la red, tenemos

$$f(m) = -\frac{Jm^2}{2} - hm + T\left[\frac{1+m}{2}\log\left(\frac{1+m}{2}\right) + \frac{1-m}{2}\log\left(\frac{1-m}{2}\right)\right]$$

y la **ecuación de auto-consistencia generalizada** es

$$m^* = \tanh(\beta(Jm^* + h))$$

donde el campo externo h rompe la simetría entre los dos estados de memoria, favoreciendo uno sobre el otro. Esto significa que la red tendrá una preferencia clara por recordar un patrón específico (por ejemplo, todas neuronas activas) en lugar de su inverso (todas inactivas).

Es lógico que m^* son los valores donde la derivada de $f(m)$ es cero, pero esto incluye tanto mínimos como máximos y puntos de inflexión. Nosotros nos interesaremos por los mínimos que son los estados estables de la red. Querremos ahora demostrar que m^* es el valor esperado de la magnetización en el límite termodinámico (cuando $N \to \infty$), es decir

$$\lim_{N\to\infty} \omega_{N,J,h,\beta}(m_N(\sigma)) = m^*$$

donde $\omega_{N,J,h,\beta}$ representa el promedio estadístico de la magnetización bajo la medida de Boltzmann. El límite $\lim_{N\to\infty} \omega(m_N) = m^*$ te asegura que, en una red grande, no verás fluctuaciones locas: verás a la red comportándose de forma sólida y constante en este nivel de fidelidad m^*.

Vamos a demostrar esto usando el **método del punto de silla**, para ellos necesitamos usar la siguiente identidad

$$e^{\frac{A}{2}O^2} = C' \int_{-\infty}^{\infty} e^{-\frac{x^2}{2A}+Ox} dx$$

donde C' es una constante de normalización. Usando esta identidad, podemos reescribir la función de partición $Z_{N,h,\beta}$ de la siguiente manera, donde usaremos $J = 1$ para simplificar la notación

$$\begin{aligned}
Z_{N,h,\beta} = \sum_{\sigma} e^{-\beta \mathcal{H}_{N,h}(\sigma)} &= \sum_{\sigma} \exp\left[\frac{\beta}{2N}\left(\sum_i \sigma_i\right)^2 + \beta h \left(\sum_i \sigma_i\right)\right] = \\
&\overset{(*)}{=} \sum_{\sigma} \exp\left[\beta h \left(\sum_i \sigma_i\right)\right] C' \int_{-\infty}^{+\infty} \exp\left[-\frac{y^2}{2(\beta/N)} + y \sum_i \sigma_i\right] dy = \\
&\overset{(**)}{=} \sum_{\sigma} \exp\left[\beta h \left(\sum_i \sigma_i\right)\right] C \int_{-\infty}^{+\infty} \exp\left[-\frac{N\beta x^2}{2} + \beta x \sum_i \sigma_i\right] dx = \\
&= C \int_{-\infty}^{+\infty} \sum_{\sigma} \exp\left[-N\beta\frac{x^2}{2} + \beta(h+x)\sum_i \sigma_i\right] dx = \\
&= C \int_{-\infty}^{+\infty} \exp\left(-N\beta\frac{x^2}{2}\right) \left[\sum_{\sigma} \prod_{i=1}^{N} \exp\left(\beta(h+x)\sigma_i\right)\right] dx
\end{aligned}$$

donde en $(*)$ hemos aplicado la identidad anterior con $A = \beta/N$ y $O = \sum_i \sigma_i$, y en $(**)$ hemos hecho un cambio de variable $y = \beta x$. Resaltar que la expresión del

Hamiltoniano usada es la siguiente

$$\mathcal{H}_{N,J,h}(\sigma) = -\frac{NJ}{2}[m_N(\sigma)]^2 - hNm_N(\sigma) = -\frac{N}{2}\left(\frac{1}{N}\sum_{i=1}^{N}\sigma_i\right)^2 - hN\left(\frac{1}{N}\sum_{i=1}^{N}\sigma_i\right) = \\ = -\left[\frac{1}{2N}\left(\sum_{i=1}^{N}\sigma_i\right)^2 + h\sum_{i=1}^{N}\sigma_i\right]$$

Para seguir con el desarrollo anterior, vamos a ver que

$$\sum_{\sigma}\cdots = \sum_{\sigma_1=\pm1}\sum_{\sigma_2=\pm1}\cdots\sum_{\sigma_N=\pm1}\cdots$$

donde lo que hacemos en la primera sumatoria es recorrer todas las posibles configuraciones de la red, y en las siguientes sumatorias recorremos cada neurona individualmente, dandole a la primera neurona el valor $+1$ y luego -1, y así sucesivamente con todas las neuronas, mientras que cuando avanzamos en la sumatoria de la primera neurona, las demás permanecen fijas.

Veámos esta idea un poco más al detalle, pues lo que estamos haciendo es recorrer el estado individual de cada neurona por separado.

- $\sum_{\sigma_1=\pm1}$: Esto significa "mira la neurona 1; primero ponla en $+1$ y luego en -1".
- $\sum_{\sigma_2=\pm1}$:Mientras la neurona 1 está fija, haz lo mismo con la neurona 2.
- Al poner todos estos sumatorios juntos, lo que haces es recorrer de forma sistemática todas las combinaciones posibles de los N elementos.

De esta manera, recorremos todas las configuraciones posibles de la red, y tenemos

$$\sum_{\sigma}\prod_{i=1}^{N}\exp\left(\beta(h+x)\sigma_i\right) = \sum_{\sigma_1=\pm1}\sum_{\sigma_2=\pm1}\cdots\sum_{\sigma_N=\pm1}\left[e^{\beta(h+x)\sigma_1}\cdot e^{\beta(h+x)\sigma_2}\cdot\ldots\cdot e^{\beta(h+x)\sigma_N}\right] = \\ = \left(\sum_{\sigma_1=\pm1}e^{\beta(h+x)\sigma_1}\right)\cdot\left(\sum_{\sigma_2=\pm1}e^{\beta(h+x)\sigma_2}\right)\cdots\left(\sum_{\sigma_N=\pm1}e^{\beta(h+x)\sigma_N}\right) = \\ \overset{(*)}{=} \left[2\cosh(\beta(h+x))\right]^N$$

donde en $(*)$ hemos usado que cada sumatorio es idéntico, y se calcula fácilmente como

$$\sum_{\sigma_i=\pm1}e^{\beta(h+x)\sigma_i} = e^{\beta(h+x)} + e^{-\beta(h+x)} = 2\cosh(\beta(h+x))$$

Siguiendo con el desarrollo de $Z_{N,h,\beta}$, tenemos

$$Z_{N,h,\beta} = C\int_{-\infty}^{+\infty}\exp\left[-N\left(\beta\frac{x^2}{2} - \log 2\cosh(\beta(h+x))\right)\right]dx = C\int_{-\infty}^{+\infty}\exp(-N\phi(x))\,dx \approx \\ \overset{(*)}{\approx} C\exp(-N\phi(x^*))$$

donde en $(*)$ hemos aplicado el **método del punto de silla**, que nos dice que para N grande, la integral está dominada por el valor mínimo de la función $\phi(x)$ dado por x^*, pues para el resto de puntos x la exponencial decae muy rápido (queda como la función de dirac). Calculemos dicho mínimo

$$\begin{aligned}\phi(x) = \beta\frac{x^2}{2} - \log 2\cosh(\beta(h+x)) \implies \phi'(x) = \beta x - \beta\frac{2\sinh(\beta(h+x))}{2\cosh(\beta(h+x))} = \\ = \beta x - \beta\tanh(\beta(h+x)) = 0 \implies x^* = \tanh(\beta(h+x^*))\end{aligned}$$

donde hemos usado que la derivada de $\cosh(x)$ es $\sinh(x)$.

Por definición y propiedades de las observaciones térmicas, tenemos que

$$\begin{aligned}\omega_N(m_N) = -\frac{\partial f_N}{\partial h} = \frac{1}{N\beta}\frac{\partial \log Z_{N,h,\beta}}{\partial h} = \frac{1}{N\beta}\frac{\partial}{\partial h}\log\left(C\exp(-N\phi(x^*))\right) = \\ = \frac{1}{N\beta}\left[\frac{\partial}{\partial h}\log(C) - \frac{\partial}{\partial h}N\phi(x^*)\right] = -\frac{1}{\beta}\frac{\partial\phi(x^*)}{\partial h}\end{aligned}$$

por lo que calculemos ahora la derivada de $\phi(x^*)$ respecto a h

$$\begin{aligned}\frac{\partial\phi(x^*)}{\partial h} &= \frac{\partial}{\partial h}\left[\beta\frac{(x^*)^2}{2} - \log 2\cosh(\beta(h+x^*))\right] = \\ &= \beta x^*\frac{\partial x^*}{\partial h} - \beta\frac{2\sinh(\beta(h+x^*))}{2\cosh(\beta(h+x^*))}\left(1+\frac{\partial x^*}{\partial h}\right) = \\ &= \beta x^*\frac{\partial x^*}{\partial h} - \beta\tanh(\beta(h+x^*))\left(1+\frac{\partial x^*}{\partial h}\right) = \\ &= \frac{\partial x^*}{\partial h}\beta\left(x^* - \tanh(\beta(h+x^*))\right) - \beta\tanh(\beta(h+x^*)) = \\ &\overset{(*)}{=} -\beta\tanh(\beta(h+x^*))\end{aligned}$$

donde hemos usado la regla de la cadena para derivar el segundo término, y en $(*)$ hemos usado que x^* satisface la ecuación de auto-consistencia $x^* = \tanh(\beta(h+x^*))$, por lo que el primer término se anula. Finalmente, tenemos lo esperado

$$\omega(m) = \tanh(\beta(h+x^*)) = x^*$$

que nos indica que el valor esperado de la magnetización en el límite termodinámico es igual a la solución de la ecuación de auto-consistencia.

Aquí tenemos un resumen de lo que hemos logrado hasta ahora

- El punto de mínima energía (m^*): representa el estado de mayor estabilidad termodinámica. Es el valor hacia el cual el sistema "quiereïr naturalmente para minimizar su energía libre, pues es el punto mínimo de $f(m)$.
- El valor más esperado ($\omega(m) = x^*$): representa lo que realmente observarías si midieras la red. En el límite de muchas neuronas ($N \to \infty$), la probabilidad se concentra tanto en el punto de silla x^* que cualquier fluctuación desaparece.

- La coincidencia ($\omega(m) = x^* = m^*$): al demostrar que estos valores son el mismo, estás probando que la realidad física (lo que mides) coincide con la estabilidad teórica (el mínimo de energía).

Ahora, sabiendo que la magentización converge a $x^* = m^*$, estudiaremos las soluciones de la ecuación de auto-consistencia para diferentes valores de la temperatura T y el campo externo h, es decir, para que valores de T y h la red puede recordar patrones (tener $m^* \neq 0$) o no (tener $m^* = 0$).

Primero, veamos el caso sin campo externo ($h = 0$), para centrarnos en el ruido. En este caso, usaremos la aproximación de $\tanh(x)$ por la serie de Taylor alrededor de $x = 0$, que es

$$\tanh(x) \approx x - \frac{x^3}{3} + O(x^5)$$

y aplicándola a la ecuación de auto-consistencia, tenemos

$$m = \tanh(\beta J m) \approx \beta J m - \frac{(\beta J m)^3}{3} + O(m^5)$$

donde tomamos $m^* = m$ por simplicidad y obviando los términos de orden superior, tenemos

$$m = \beta J m - \frac{(\beta J m)^3}{3} \implies m(J\beta - 1) = \frac{(\beta J)^3}{3} m^3 \implies m^2 = \frac{3(J\beta - 1)}{(\beta J)^3}$$

Llegados a este punto, tenemos dos casos

- Si $J\beta < 1$ (es decir, $T > J$), entonces $m^2 < 0$, lo cual es imposible. Por lo tanto, la única solución física es $m = 0$. Esto significa que la red no puede recordar ningún patrón, ya que la magnetización es nula.
- Si $J\beta > 1$ (es decir, $T < J$), entonces $m^2 > 0$, y por lo tanto hay dos soluciones físicas posibles

  $$m = \pm\sqrt{\frac{3(J\beta - 1)}{(\beta J)^3}}$$

  Esto significa que la red puede recordar dos patrones (uno y su inverso), ya que la magnetización es diferente de cero.

Por tanto, hemos encontrado un punto crítico en la temperatura $T_c = \frac{1}{\beta_c} = J$, que separa dos fases distintas del sistema. Para que la fórmula sea más fácil de interpretar, aproximamos el término $(\beta J - 1)$ cerca de T_c

$$\beta J - 1 = \frac{J}{T} - 1 = \frac{J - T}{T} \approx \frac{T_c - T}{T_c}$$

Sustituyendo esto en la fórmula de m y asumiendo que cerca de T_c, $\beta J \approx 1$, entonces

$$m \approx \pm\sqrt{\frac{3(T_c - T)}{T_c}}$$

esta ecuación nos dice que la magnetización m crece como la raíz cuadrada de la diferencia entre la temperatura crítica T_c y la temperatura actual T, siempre que $T_c < T$. Al ser una raíz cuadrada, la pendiente en T_c es infinita. Esto significa que en cuanto el ruido baja un milímetro por debajo del punto crítico, la red reacciona violentamente y empieza a formar un recuerdo muy rápido.

Finalmente, hemos encontrado los valores de temperatura T que definen las dos fases del sistema

- **Fase ordenada (memoria):** para $T < T_c$, la red puede recordar patrones, ya que la magnetización es diferente de cero ($m \neq 0$).
- **Fase desordenada (caos):** para $T > T_c$, la red no puede recordar ningún patrón, ya que la magnetización es nula ($m = 0$).

Este comportamiento es característico de una **transición de fase** en sistemas físicos, donde un cambio en un parámetro (en este caso, la temperatura) provoca un cambio abrupto en las propiedades del sistema (en este caso, la capacidad de memoria de la red neuronal).

4.2. Phase transition

Definición 4.2.2. En física, una **transición de fase** es cuando el agua se congela. En nuestra asignatura, ocurre cuando hay una singularidad (un "salto.º un cambio brusco) en la Energía Libre (F). Imagina que vas subiendo el ruido (temperatura T) en tu red. De repente, en un valor exacto de T, la red deja de reconocer el patrón. No lo hace poco a poco, sino que "colapsa". Ese punto exacto es la transición de fase.

Para la clasificación de las transiciones de fase, usamos la **Clasificación de Ehrenfest**, que se basa en la continuidad de las derivadas de la energía libre (F):

- **Primer orden**: hay una discontinuidad en la primera derivada de F. En la práctica: la memoria desaparece de golpe (como el hielo fundiéndose).
- **Segundo orden (o continua)**: las primeras derivadas son continuas, pero las segundas fallan. En la práctica: la capacidad de recordar va bajando suavemente hasta que llega a cero de forma nítida.

El **parámetro de orden** es una variable que nos dice en qué fase estamos. En nuestro caso, es la magnetización (m). En nuestro caso el solapamiento (m), puede ser

- $m = 0$: Fase paramagnética (ruido total, no hay memoria).
- $m \neq 0$: Fase ferromagnética (la red recuerda, hay orden).

Symmetry Breaking

En el modelo de Hopfield, si inviertes todas las neuronas ($\sigma_i \to -\sigma_i$), la energía es la misma. El sistema "no sabe"si prefiere el patrón o su inverso. Esto se llama **simetría**.

En baja temperatura (Orden): El sistema .[el]igeüno de los dos estados (el patrón o su inverso) y se queda ahí. Al elegir, rompe la simetría.
Aplicación: Aprender es, esencialmente, romper la simetría del desorden para çaer.[en] un estado concreto (el recuerdo).

En alta temperatura (Caos): La red salta tanto de un estado a otro que el promedio es cero. El sistema es simétrico.

4.3. Ergodicity breaking

Cuando la temperatura (ruido) es alta ($T > T_c$), el sistema tiene tanta energía térmica que nada lo detiene. La red neuronal salta de un estado a otro sin parar. No se queda fija en ningún recuerdo, lo que significaría que es una red **ergódica**, por lo que decimos que el sistema "explora todo el espacio Ω".

Sin embargo, cuando la temperatura baja ($T < T_c$) (el sistema se enfría), el sistema pierde energía térmica y se "atasca" en ciertos estados (recuerdos). La red ya no puede explorar todo el espacio de configuraciones posibles, sino que se queda atrapada en ciertos atractores. Esto se llama **ruptura de ergodicidad**. Lo que ocurre es que, el espacio de fases Ω se "rompe" en pedazos aislados (por ejemplo, Ω_+ y Ω_-), donde Ω_+ podría representar el recuerdo de una "Cara", y Ω_- podría representar el recuerdo inverso (o un "Gato").

En este sentido, decimos que el sistema se queda **confinado** en uno de estos subespacios, dependiendo de su estado inicial y de las fluctuaciones térmicas. La red ya no puede saltar libremente entre todos los estados posible (por ejemplo de Ω_+ a Ω_-), porque hay una barrera de energía gigante (proporcional al número de neuronas N).

Para que esto sea "memoria", necesitamos comparar dos tiempos:

- τ_0 (**Tiempo de observación**): el tiempo que tardamos en usar la red.
- τ_{flip} (**Tiempo de salto**): el tiempo que tardaría la red en saltar de un recuerdo a otro por puro azar.
- **Ruptura de la Ergodicidad** ($\tau_0 \ll \tau_{flip}$): el salto es tan improbable (tardaría "tiempos astronómicos", más que la edad del universo) que, para nosotros, el sistema está en equilibrio local. ¡Esto es la memoria! La red se queda fija en el patrón que le hemos pedido recuperar.

Un punto a notar importante es que, en teoría, si esperasemos trillones de años, el sistema acabaría saltando de un valle a otro y recuperaría la ergodicidad. Sin embargo, en la práctica, como nosotros usamos la red neuronal durante unos segundos

o minutos (un tiempo τ_0 muy pequeño), para nosotros el sistema está en equilibrio local y no vemos esos saltos. Esta es la justificación matemática de por qué podemos decir que una red neuronal ha "convergido" a una solución, aunque técnicamente el universo siga evolucionando.

¿Por qué es importante esto en el modelo de Hopfield? Si la red fuera siempre ergódica, tú le darías una imagen borrosa y ella empezaría a oscilar entre todos los patrones que conoce sin detenerse en ninguno. La ruptura de la ergodicidad es lo que permite que la red "tome una decisión" y se quede en un valle de energía (atractor).

4.4. Hopfield model

El modelo de Hopfield es la evolución natural del modelo de Curie-Weiss (CW) que hemos estado analizando. Mientras que en el modelo CW las neuronas solo aprendían a "ponerse todas de acuerdo" (almacenando solo 1 bit de información), el modelo de Hopfield permite que la red actúe como una memoria asociativa capaz de almacenar y recuperar múltiples patrones complejos.

La tarea principal de esta red es la reconstrucción de P "piezas de información". Estas piezas son vectores binarios $\{\xi^\mu\}$ de longitud N. Imagina que cada vector ξ^μ es una imagen o un concepto (por ejemplo, una foto de un gato). La red debe ser capaz de "recordar" esa imagen completa incluso si solo le das una parte borrosa o con ruido. En lugar de tener solo dos valles en $+1$ y -1, el modelo de Hopfield diseña el paisaje para que haya un "valle" (mínimo de energía) en cada uno de los patrones ξ^μ que queremos recordar.

En el modelo CW, las conexiones entre neuronas eran uniformes (todas las neuronas se influían igual), es decir, $J_{ij} = \frac{J}{N}$. En el modelo de Hopfield, las conexiones son específicas y se diseñan para almacenar los patrones deseados. La **regla de aprendizaje de Hebb** nos dice cómo ajustar estas conexiones de la siguiente manera

$$J_{ij} = \frac{1}{N} \sum_{\mu=1}^{P} \xi_i^\mu \xi_j^\mu$$

donde ξ_i^μ es el estado (activo o inactivo) de la neurona i en el patrón μ. Esta regla asegura que las neuronas que tienden a activarse juntas (en el mismo patrón) refuercen su conexión, mientras que las que no lo hacen debilitan su vínculo. Ahora el hamiltoniano del sistema es

$$\mathcal{H}_{N,\xi,h}(\sigma) = -\frac{1}{N} \sum_{i,j<i} \sum_{\mu} \xi_i^\mu \xi_j^\mu \sigma_i \sigma_j - \sum_i h_i \sigma_i$$

donde σ_i es el estado actual de la neurona i.

Nos centraremos en estudiar como la red puede recuperar un patrón específico ξ^ν cuando se le presenta una versión ruidosa o incompleta del mismo, por lo que definiremos $h = 0$, ya que queremos estudiar la capacidad de memoria pura de la red

sin influencias externas, pues estas pueden ser beneficiosas para la clasificación, y es eso justo lo que no queremos, intentando estudiar solo como reacciona la red en sí.

4.4.1. Low-load regime

En esta sección nos centraremos en el **régimen de baja carga** (low-load), donde el número de patrones almacenados P es mucho menor que el número de neuronas N (es decir, $\alpha = P/N \to 0$ cuando $N \to \infty$). En este régimen, la red puede manejar múltiples patrones sin que interfieran entre sí, lo que facilita el análisis.

Definición 4.4.3. Consider the Hopfield model with N neurons σ_i, $i \in \{1, \dots, N\}$ and P patterns ξ^μ, $\mu \in \{1, \dots, P\}$, the **Hamiltonian** is defined as

$$H_{N,\xi}(\sigma) = -\frac{1}{N}\sum_{i,j<i}^{N}\sum_{\mu=1}^{P}\xi_i^\mu\xi_j^\mu\sigma_i\sigma_j = -\frac{1}{2N}\sum_{i,j,\mu}\xi_i^\mu\xi_j^\mu\sigma_i\sigma_j + \frac{1}{2N}\sum_{i,\mu}\left(\xi_i^\mu\right)^2\sigma_i^2 =$$
$$= -\frac{1}{2N}\sum_{i,j,\mu}\xi_i^\mu\xi_j^\mu\sigma_i\sigma_j + \frac{P}{2}$$

Definimos las siguientes funciones

- the **(pattern realization dependent) partition function** as

$$Z_{N,\beta,\xi} = \sum_{\sigma} e^{-\beta H_{N,\xi}(\sigma)}$$

- the **Boltzmann factor** as

$$B_{N,\beta,\xi}(\sigma) = e^{-\beta H_{N,\xi}(\sigma)}$$

- the **Boltzmann-Gibbs average** as

$$\omega_{N,\xi}(F) = \frac{\sum_\sigma F(\sigma)B_{N,\beta,\xi}(\sigma)}{\sum_\sigma B_{N,\beta,\xi}(\sigma)}$$

  for any arbitrary function $F(\sigma)$ of the spins.

- the **quenched statistical pressure** as

$$A_{N,\beta} = -\beta f_{N,\beta}^Q, \quad f_{N,\beta}^Q = -\frac{1}{\beta N}\mathbb{E}\log Z_{N,\beta,\xi}$$

  where $f_{N,\beta}^Q = -\frac{1}{\beta N}\mathbb{E}\log Z_{N,\beta,\xi}$ is the **(quenched) free energy**.

Para resolver el modelo de Hopfield, lo que buscamos es querer demostrar que la red es capaz de recordar los patrones almacenados $\{\xi^\mu\}$, es decir, que la red puede converger a uno de estos patrones cuando se le presenta una versión ruidosa o incompleta del mismo. Para ello, debemos determinar el valor de los **solapamientos**

u **order parameters** de m_μ, que miden la similitud entre el estado actual de la red σ y cada patrón almacenado ξ^μ. Estos solapamientos se definen como

$$m_\mu(\sigma) = \frac{1}{N}\sum_{i=1}^{N} \xi_i^\mu \sigma_i$$

donde m_μ toma valores entre -1 y $+1$. Un valor de m_μ cercano a $+1$ indica que la red está en un estado muy similar al patrón ξ^μ, mientras que un valor cercano a -1 indica que la red está en el patrón inverso. Un valor cercano a 0 significa que la red no tiene correlación con ese patrón.

The main interest of this section and of the two following ones (for which the same definitions hold) is to find the expression of the free energy in the thermodynamic limit $f_\beta^Q = \lim_{N\to\infty} f_{N,\beta}^Q$ in terms of the order parameters, both in the low and high storage regimes.

Saddle Point Method

Antes de proceder con el desarrollo del modelo de Hopfield, repasemos brevemente el **método del punto de silla** (saddle point method), que ya hemos usado en la sección anterior. Este método es una técnica matemática utilizada para aproximar integrales de la forma

$$I(N) = \int e^{-N\phi(x)}\, dx$$

cuando N es grande. La idea principal es que, para valores grandes de N, la integral está dominada por las regiones donde la función $\phi(x)$ es mínima, ya que la exponencial decae rápidamente en otras regiones.

También, es importante destacar esta igualdad que usaremos en el desarrollo posterior procedente de la delta de Dirac

$$\int_{-\infty}^{+\infty} f(x)\delta(x-a)\, dx = f(a) \implies \int_{-\infty}^{+\infty} g(m_\mu)\delta\left(m_\mu - \frac{1}{N}\sum_i \xi_i^\mu \sigma_i\right)\, dm_\mu = 1$$

donde $g(x) = 1$. Aplicando esto a todos los patrones $\mu \in \{1, \dots, P\}$, tenemos

$$\prod_{\mu=1}^{P}\int_{-\infty}^{+\infty} \delta\left(m_\mu - \frac{1}{N}\sum_i \xi_i^\mu \sigma_i\right)\, dm_\mu = 1$$

Usando esta identidad, y el siguiente desarrollo

$$\sum_{i,j} \xi_i^\mu \xi_j^\mu \sigma_i \sigma_j = \left(\sum_i \xi_i^\mu \sigma_i\right)^2 = N^2 m_\mu^2 \implies \frac{\beta}{2N}\sum_{\mu=1}^{P}\left(\sum_i \xi_i^\mu \sigma_i\right)^2 = \frac{\beta N}{2}\sum_{\mu=1}^{P} m_\mu^2$$

podemos reescribir la función de partición del modelo de Hopfield como

$$Z_{N,\beta,\xi} = \sum_\sigma \exp\left[\frac{\beta}{2N}\sum_{i,j,\mu}\xi_i^\mu \xi_j^\mu \sigma_i \sigma_j\right] = \sum_\sigma \int \exp\left(\frac{\beta N}{2}\sum_{\mu=1}^{P} m_\mu^2\right)\prod_{\mu=1}^{P}\left[\delta\left(m_\mu - \frac{1}{N}\sum_i \xi_i^\mu \sigma_i\right)\, dm_\mu\right]$$

donde hemos obviado el término constante $\exp(-\beta P/2)$, ya que no afecta a las observables térmicas, cuando N es grande. Ahora buscaremos aplicar la definición formal de la delta de Dirac

$$\delta(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{ikx}dk \implies \delta(x-y) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{ik(x-y)}dk \implies$$
$$\implies \delta\left(m_\mu - \frac{1}{N}\sum_i \xi_i^\mu \sigma_i\right) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} \exp\left(ik\left(m_\mu - \frac{1}{N}\sum_i \xi_i^\mu \sigma_i\right)\right) dk$$

además de hacer un cambio de variable $k = N\bar{m}_\mu$. Veamos todos los pasos al aplicar estos conceptos

$$\begin{aligned}
Z_{N,\beta,\xi} &= \sum_\sigma \int \exp\left(\frac{\beta N}{2}\sum_{\mu=1}^P m_\mu^2\right)\prod_{\mu=1}^P\left[\exp\left(ik_\mu\left(m_\mu - \frac{1}{N}\sum_i \xi_i^\mu\sigma_i\right)\right)\ \frac{dm_\mu dk_\mu}{2\pi}\right] = \\
&\overset{(*)}{=} \sum_\sigma \int \exp\left(\frac{\beta N}{2}\sum_{\mu=1}^P m_\mu^2\right)\prod_{\mu=1}^P\left[\exp\left(iN\bar{m}_\mu\left(m_\mu - \frac{1}{N}\sum_i \xi_i^\mu\sigma_i\right)\right)\ \frac{Ndm_\mu d\bar{m}_\mu}{2\pi}\right] = \\
&= \sum_\sigma \int \exp\left(\frac{\beta N}{2}\sum_{\mu=1}^P m_\mu^2\right)\prod_{\mu=1}^P\left[\frac{Ndm_\mu d\bar{m}_\mu}{2\pi}\right]\exp\left[iN\sum_\mu \bar{m}_\mu\left(m_\mu - \frac{1}{N}\sum_i \xi_i^\mu\sigma_i\right)\right] = \\
&= \sum_\sigma \int \prod_{\mu=1}^P\left[\frac{Ndm_\mu d\bar{m}_\mu}{2\pi}\right]\exp\left(iN\sum_\mu \bar{m}_\mu m_\mu - i\sum_{i,\mu}\bar{m}_\mu\xi_i^\mu\sigma_i + \frac{\beta N}{2}\sum_\mu m_\mu^2\right)
\end{aligned}$$

donde en $(*)$ hemos hecho el cambio de variable $k_\mu = N\bar{m}_\mu$. Ahora volveremos a usar una idea similar a la que usamos en la sección anterior, es decir, que

$$\begin{aligned}
\sum_\sigma \exp\left(-i\sum_{i,\mu}\bar{m}_\mu\xi_i^\mu\sigma_i\right) &= \prod_{i=1}^N\sum_{\sigma_i=\pm1}\exp\left(-i\sigma_i\sum_\mu \bar{m}_\mu\xi_i^\mu\right) = \\
&\overset{(*)}{=} \prod_{i=1}^N 2\cos\left(\sum_\mu \bar{m}_\mu\xi_i^\mu\right) = \exp\left(\ln\left[\prod_{i=1}^N 2\cos\left(\sum_\mu \bar{m}_\mu\xi_i^\mu\right)\right]\right) = \\
&= \exp\left(\sum_{i=1}^N \ln\left[2\cos\left(\sum_\mu \bar{m}_\mu\xi_i^\mu\right)\right]\right)
\end{aligned}$$

o más desarrollado la primera parte

$$\begin{aligned}\sum_{\sigma}\exp\left(-i\sum_{i,\mu}\bar{m}_\mu\xi_i^\mu\sigma_i\right) &= \sum_{\sigma}\exp\left(-i\sum_{i}\sigma_i\sum_{\mu}\bar{m}_\mu\xi_i^\mu\right) = \\ &= \sum_{\sigma}\exp\left(-i\sigma_1\sum_{\mu}\bar{m}_\mu\xi_1^\mu\right)\ldots\exp\left(-i\sigma_N\sum_{\mu}\bar{m}_\mu\xi_N^\mu\right) = \\ &= \sum_{\sigma_1=\pm1}\exp\left(-i\sigma_1\sum_{\mu}\bar{m}_\mu\xi_1^\mu\right)\ldots\sum_{\sigma_N=\pm1}\exp\left(-i\sigma_N\sum_{\mu}\bar{m}_\mu\xi_N^\mu\right) = \\ &\overset{(*)}{=} 2\cos\left(\sum_{\mu}\bar{m}_\mu\xi_1^\mu\right)\ldots 2\cos\left(\sum_{\mu}\bar{m}_\mu\xi_N^\mu\right) = \\ &= \prod_{i=1}^{N}2\cos\left(\sum_{\mu}\bar{m}_\mu\xi_i^\mu\right)\end{aligned}$$

donde en $(*)$ hemos usado que $e^{ix}+e^{-ix}=2\cos(x)$. Tenemos ahora la función de partición como

$$Z_{N,\beta,\xi} = \int\prod_{\mu=1}^{P}\left[\frac{N dm_\mu d\bar{m}_\mu}{2\pi}\right]\exp\left(iN\sum_{\mu}\bar{m}_\mu m_\mu + \sum_{i}\ln\left[2\cos\left(\sum_{\mu}\bar{m}_\mu\xi_i^\mu\right)\right] + \frac{\beta N}{2}\sum_{\mu}m_\mu^2\right) \tag{4.1}$$

Cuando N es grande, podemos aplicar el **método del punto de silla** para aproximar la integral, donde buscaremos el mínimo primero para m_μ y luego para $\bar{m}_\mu$. Primero nos centraremos en minimizar respecto a m_μ , por lo que derivaremos el exponente respecto a m_μ (debemos fijar μ pues el resto de m_ν son distintas para $\mu\neq\nu$) y lo igualaremos a cero

$$\frac{\partial}{\partial m_\mu}\left(iN\sum_{\mu}\bar{m}_\mu m_\mu + \sum_{i}\ln\left[2\cos\left(\sum_{\mu}\bar{m}_\mu\xi_i^\mu\right)\right] + \frac{\beta N}{2}\sum_{\mu}m_\mu^2\right) = iN\bar{m}_\mu + \beta N m_\mu = 0 \implies$$

$$\implies \bar{m}_\mu = \frac{-\beta m_\mu}{i} = i\beta m_\mu$$

notar que aunque hemos derivado respecto a m_μ, hemos despejado para $\bar{m}_\mu$, pues buscamos tener la expresión de la función de partición en función de los ordenes de parámetro m_μ. Ahora, sustituyendo este valor de $\bar{m}_\mu$ en la función de partición, y aplicando el método del punto de silla (donde eliminamos por tanto las integrales respecto a $\bar{m}_\mu$), tenemos

$$Z_{N,\beta,\xi}\underset{N\to\infty}{\sim}\int\left(\prod_{\mu}dm_\mu\right)\exp\left(-\frac{\beta N}{2}\sum_{\mu=1}^{P}m_\mu^2 + \sum_{i}\ln\left[2\cosh\left(\beta\sum_{\mu}m_\mu\xi_i^\mu\right)\right]\right)$$

debemos destacar que hemos usado que $\cos(ix)=\cosh(x)$.

Hemos obtenido una expresión de la función de partición en términos de los órdenes de parámetro m_μ, pero realmente queremos poder quitar la integral, por lo que para

ello aplicaremos el método del punto de silla a la vez, tanto para m_μ como para $\bar{m}_\mu$. Por tanto, derivamos el exponente que teníamos en 4.1 respecto a $\bar{m}_\mu$ y lo igualamos a cero

$$iNm_\mu + \sum_i \frac{-\sin\left(\sum_\mu \bar{m}_\mu \xi_i^\mu\right)}{\cos\left(\sum_\mu \bar{m}_\mu \xi_i^\mu\right)} \xi_i^\mu = 0 \implies iNm_\mu - \sum_i \xi_i^\mu \tan\left(\sum_\mu \bar{m}_\mu \xi_i^\mu\right) = 0 \implies$$

$$\implies m_\mu = -\frac{i}{N} \sum_j \xi_i^\mu \tan\left(\sum_\nu \bar{m}_\nu \xi_j^\nu\right)$$

Ahora tomando el mínimo m_μ^* para este valor de m_μ y el valor de $\bar{m}_\mu$ que habíamos obtenido antes, tenemos el sistema de ecuaciones

$$\begin{cases} \bar{m}_\mu = i\beta m_\mu^* \\ m_\mu = m_\mu^* = -\dfrac{i}{N} \displaystyle\sum_j \xi_i^\mu \tan\left(i\beta \sum_\mu m_\mu^* \xi_j^\mu\right) \end{cases} \implies m_\mu^* = \frac{1}{N} \sum_j \xi_i^\mu \tanh\left(\beta \sum_\nu m_\nu \xi_j^\nu\right)$$

donde hemos usado que $\tan(ix) = i\tanh(x)$. Finalmente, aplicamos el método del punto de silla en la función de partición, sustituyendo el valor m_μ^* en la expresión, obteniendo

$$Z_{N,\beta,\xi} \underset{N\to\infty}{\sim} \left(\frac{N}{2\pi}\right)^{\frac{P}{2}} \exp\left(-\frac{\beta N}{2} \sum_{\mu=1}^{P} (m_\mu^*)^2 + \sum_i \ln\left[2\cosh\left(\beta \sum_\mu m_\mu^* \xi_i^\mu\right)\right]\right)$$

donde la parte del productorio $\left(\frac{N}{2\pi}\right)^{\frac{P}{2}}$ es irrelevante en el límite termodinámico, pues al calcular la free energy, esta parte se anula al dividir por N y hacer $N \to \infty$. Entonces, the (quenched) free energy in the thermodynamic limit is

$$\begin{aligned} f_\beta^Q &= \lim_{N\to\infty} f_{N,\beta}^Q = \lim_{N\to\infty} \left[-\frac{1}{\beta N} \mathbb{E} \log Z_{N,\beta,\xi}\right] = \\ &= \lim_{N\to\infty} \mathbb{E}_\xi \left[\frac{1}{2} \sum_{\mu=1}^{P} (m_\mu^*)^2 - \frac{1}{\beta N} \sum_{i=1}^{N} \ln\left(2\cosh\left(\beta \sum_{\mu=1}^{P} m_\mu^* \xi_i^\mu\right)\right)\right] = \\ &= \min_m \left[\frac{1}{2} \sum_{\mu=1}^{P} m_\mu^2 - \frac{1}{\beta} \mathbb{E} \log 2\cosh\left(\beta \sum_{\mu=1}^{P} m_\mu \xi^\mu\right)\right] \end{aligned}$$

donde lo que hemos aplicado es definir la variable aleatoria X_i para cada neurona i como

$$X_i = \ln\left(2\cosh\left(\beta \sum_{\mu=1}^{P} m_\mu^* \xi_i^\mu\right)\right)$$

que por la ley de los grandes números (cumple las hipótesis trivialmente), cuando $N \to \infty$, tenemos que

$$\frac{1}{N} \sum_{i=1}^{N} X_i \underset{N\to\infty}{\to} \mathbb{E}[X] = \mathbb{E}_\xi \left[\ln\left(2\cosh\left(\beta \sum_{\mu=1}^{P} m_\mu^* \xi^\mu\right)\right)\right] \quad \text{c.s.}$$

Ahora bien, necesitamos ver como calcular la esperenza respecto a los patrones ξ. En nuestro caso, el "experimento" es elegir los P bits de una neurona al azar. Definimos un vector aleatorio $\xi = (\xi^1, \xi^2, \ldots, \xi^P)$ que representa los bits de una neurona genérica (puede tomar ± 1 con probabilidad $1/2$ cada uno). Entonces, la esperanza que necesitamos calcular es

$$\mathbb{E}[g(\xi)] = \sum_{\text{todas las configuraciones } \xi} P(\xi) \cdot g(\xi)$$

donde sustituyendo nuestra $P(\xi) = 1/2^P$ y expandiendo la suma para cada componente del vector, obtenemos la siguiente ecuación

$$\mathbb{E}[g(\xi)] = \frac{1}{2^P} \sum_{\xi^1=\pm 1} \sum_{\xi^2=\pm 1} \cdots \sum_{\xi^P=\pm 1} g(\xi^1, \xi^2, \ldots, \xi^P)$$

y por tanto sabemos como calcular la esperanza respecto a los patrones ξ.

Ahora para demostrar el siguiente enunciado, debemos recordar que los valores de m_μ que minimizan la quenched free-energy son los obtenidos al derivar la quenched free-energy respecto a m_μ y igualar a cero

$$\begin{aligned}
\frac{\partial f}{\partial m_\mu} &= \frac{\partial}{\partial m_\mu}\left[\frac{1}{2}\sum_{\nu=1}^{P} m_\nu^2 - \frac{1}{\beta}\mathbb{E}\ln\left(2\cosh\left(\beta\sum_{\nu=1}^{P} m_\nu \xi^\nu\right)\right)\right] = \\
&= m_\mu - \frac{1}{\beta}\mathbb{E}\left[\tanh\left(\beta\sum m_\nu \xi^\nu\right)\cdot \beta\xi^\mu\right] = 0 \\
&\implies m_\mu = \mathbb{E}_\xi\left[\xi^\mu \tanh\left(\beta\sum_{\nu=1}^{P} m_\nu \xi^\nu\right)\right]
\end{aligned}$$

Teorema 4.4.1. *The quenched free-energy of the Hopfield model in the thermodynamic limit of the low-storage regime is*

$$f_\beta^Q = \min_m \left[\frac{1}{2}\sum_{\mu=1}^{P} m_\mu^2 - \frac{1}{\beta}\mathbb{E}\log 2\cosh\left(\beta\sum_{\mu=1}^{P} m_\mu \xi^\mu\right)\right]$$

where the Mattis magnetizations satisfy the ***self-consistency equations***

$$m_\mu^* = \mathbb{E}_\xi\left[\xi^\mu \tanh\left(\beta\sum_{\nu=1}^{P} m_\nu^* \xi^\nu\right)\right]$$

at the equilibrium states.

Ahora veremos ciertos comentarios muy importantes respecto a la solución obtenida.

Notación Vectorial y el Estado de Equilibrio. Usando una notación vectorial más compacta, podemos escribir la energía libre y las ecuaciones de auto-consistencia como

$$f_\beta = \frac{1}{2}\mathbf{m}^2 - \frac{1}{\beta}\mathbb{E}\log 2\cosh(\beta\mathbf{m}\cdot\xi) \quad \text{tal que } \mathbf{m} = \mathbb{E}\xi\tanh(\beta\mathbf{m}\cdot\xi)$$

donde $\mathbf{m}$ es la **ecuación de auto-consistencia**, que indica que el estado de equilibrio de la red (su magnetización) debe ser igual al promedio de la respuesta de las neuronas ante el campo creado por los patrones.

El Límite de Recuperación Pura (Curie-Weiss). Si suponemos que la red solo intenta recuperar el patrón 1 ($m_1 \neq 0$ y los demás $m_\mu = 0$), ocurre algo muy elegante: La ecuación se reduce a

$$m_1 = \mathbb{E}[\xi^1 \tanh(\beta m_1 \xi^1)]$$

como la función tanh es impar, $\tanh(\beta m_1 \xi^1) = \xi^1 \tanh(\beta m_1)$, y $\xi^1 = \pm 1$, entonces al multiplicar por la ξ^1 de fuera, obtenemos $(\xi^1)^2 = 1$, lo que nos da lo siguiente

$$m_1 = \tanh(\beta m_1)$$

Esta es la **Ley de Curie-Weiss** que describe a los imanes. Esto demuestra que, si solo hay un patrón, la red de Hopfield se comporta exactamente como un material ferromagnético donde los "espines" se alinean para formar el recuerdo.

Simetría de las Variables Auxiliares. Al derivar respecto a m_μ y $\bar{m}_\mu$ simultáneamente, realmente no se calculó la "energía libre restringida" (que es más difícil), pero que matemáticamente no importa: los puntos de equilibrio (mínimos) son los mismos.

Analysis of the self-consistency equation

El análisis de la ecuación de auto-consistencia

$$m_\mu = \mathbb{E}_\xi \left[\xi^\mu \tanh \left(\beta \sum_{\nu=1}^{P} m_\nu \xi^\nu \right) \right]$$

muestra que existe un umbral $\beta_c = 1$ tal que

- si $\beta < \beta_c$, $m = 0$ es la única solución;
- si $\beta > \beta_c$, resulta $m \neq 0$ y hay muchas soluciones admisibles.

Para probar esto, sabiendo donde esta el umbral β_c, tomaremos una pequeña desviación $\epsilon = \beta - 1 \ll 1$, cerca de este punto, los valores de m_μ son muy pequeños (cuanto más crezca β respecto a β_c más irá creciendo m_μ), lo que nos permite usar la aproximación de Taylor para la tangente hiperbólica

$$\tanh(x) = x - \frac{1}{3}x^3 + \frac{2}{15}x^5 - \dots$$

Sustituimos esto en nuestra ecuación original

$$m_\mu \approx \mathbb{E}_\xi \left[\xi^\mu \left(\beta \sum_\nu m_\nu \xi^\nu - \frac{1}{3} \left(\beta \sum_\nu m_\nu \xi^\nu \right)^3 \right) \right]$$

Debemos resolver la esperanza término a término, recordando que $\mathbb{E}[\xi^\mu \xi^\nu] = \delta_{\mu\nu}$ (son independientes) y que $(\xi^\mu)^2 = 1$.

- **Término Lineal:**

$$\mathbb{E}[\xi^\mu \beta \sum_\nu m_\nu \xi^\nu] = \beta \sum_\nu m_\nu \mathbb{E}[\xi^\mu \xi^\nu] = \beta m_\mu$$

- **Término Cúbico:** Sea $X = \sum_\nu m_\nu \xi^\nu$. El término es $-\frac{1}{3}\beta^3 \mathbb{E}[\xi^\mu X^3]$. Al expandir X^3 y promediar sobre el ruido, los únicos términos que sobreviven son aquellos donde los índices de ξ se emparejan. El resultado de esta combinatoria es:

$$\begin{aligned}\mathbb{E}[\xi^\mu X^3] &= \sum_{\alpha,\gamma,\delta} m_\alpha m_\gamma m_\delta \mathbb{E}[\xi^\mu \xi^\alpha \xi^\gamma \xi^\delta] = m_\mu^3 + 3\sum_{\nu\neq\mu} m_\mu m_\nu^2 = \\ &= m_\mu^3 + 3\left(\sum_{\nu=1}^{P} m_\mu m_\nu^2 - m_\mu m_\mu^2\right) = m_\mu^3 + 3m_\mu\left(\sum_{\nu=1}^{P} m_\nu^2\right) - 3m_\mu^3 = \\ &= 3m_\mu\left(\sum_{\nu=1}^{P} m_\nu^2\right) - 2m_\mu^3 = 3m_\mu \mathbf{m}^2 - 2m_\mu^3\end{aligned}$$

  donde hemos usado que

$$\mathbb{E}[\xi^\mu] = P(\xi^\mu = +1)(\xi^\mu = +1) + P(\xi^\mu = -1)(\xi^\mu = -1) = \frac{1}{2}(+1) + \frac{1}{2}(-1) = 0$$
$$\mathbb{E}[(\xi^\mu)^2] = P(\xi^\mu = +1)(\xi^\mu = +1)^2 + P(\xi^\mu = -1)(\xi^\mu = -1)^2 = \frac{1}{2}(+1)^2 + \frac{1}{2}(-1)^2 = 1$$

  de manera que

  - **Regla de imparidad**: si algún índice ξ aparece un número impar de veces (como 1 o 3), el promedio es 0.
  - **Regla de paridad**: si todos los índices aparecen un número par de veces (como 2 o 4), el promedio es 1 (porque $(\xi)^2 = 1$ y $(\xi)^4 = 1$).

Agrupando los resultados, tenemos

$$\begin{aligned}m_\mu &\approx \beta m_\mu - \frac{1}{3}\beta^3(3m_\mu\mathbf{m}^2 - 2m_\mu^3) = m_\mu(1+\epsilon) - \frac{(1+\epsilon)^3}{3}(3m_\mu\mathbf{m}^2 - 2m_\mu^3) \implies \\ &\implies m_\mu - m_\mu = m_\mu\epsilon - (1+\epsilon)^3 m_\mu \mathbf{m}^2 + \frac{2}{3}(1+\epsilon)^3 m_\mu^3 \implies \\ &\implies m_\mu\left[\epsilon - (1+\epsilon)^3\mathbf{m}^2 + \frac{2}{3}(1+\epsilon)^3 m_\mu^2\right] = 0\end{aligned}$$

entonces tenemos dos posibilidades:

- Si $m_\mu = 0$ para todo μ, tenemos la solución trivial (estado desordenado).
- Si $m_\mu \neq 0$ para algún μ, entonces el término entre corchetes debe ser cero, entonces

$$\begin{aligned}0 &= \epsilon - (1+\epsilon)^3\mathbf{m}^2 + \frac{2}{3}(1+\epsilon)^3 m_\mu^2 = \\ &= \epsilon - (1+3\epsilon+3\epsilon^2+\epsilon^3)\mathbf{m}^2 + \frac{2}{3}(1+3\epsilon+3\epsilon^2+\epsilon^3)m_\mu^2 = \\ &\overset{(**)}{=} \epsilon - \mathbf{m}^2 - 3\epsilon\mathbf{m}^2 + \frac{2}{3}m_\mu^2 + \frac{6}{3}\epsilon m_\mu^2 = \\ &\overset{(*)}{=} \epsilon - \mathbf{m}^2 + \frac{2}{3}m_\mu^2 \implies m_\mu = \pm\sqrt{\frac{3}{2}(\mathbf{m}^2 - \epsilon)}\end{aligned}$$

donde en $(**)$ hemos decidido quedarnos solo con los términos de "primer orden" (los que no tienen potencia en ϵ o solo tienen ϵ^1) y eliminar todo lo que tenga ϵ^2 o ϵ^3 y en $(*)$ hemos despreciado los términos $\epsilon\mathbf{m}^2$ o ϵm_μ^2, pues en ambos casos son números pequeños multiplicando números más pequeños pequeños, por lo que son despreciables.

Si solo existe un $m_\mu \neq 0$ (digamos m_1), tenemos la solución de patrón puro y se llamaría **estado de Mattis**, y tendríamos que toda la magnetización total $\mathbf{m}^2$ se concentra en ese patrón, es decir,

$$\mathbf{m} = (m_1, 0, 0, \ldots, 0) \implies \mathbf{m}^2 = m_\mu^2$$

por lo que la intensidad del recuerdo es

$$m_1 = \pm\sqrt{\frac{3}{2}(m_1^2 - \epsilon)} \implies m_1^2 = \frac{3}{2}m_1^2 - \frac{3}{2}\epsilon \implies m_1^2 = 3\epsilon \implies m_1 = \pm\sqrt{3\epsilon}$$

de manera que claramente si $\epsilon = \beta - 1$ es grande (equivale a $\beta \gg 1$, es decir, $T \ll 1$ poco ruido), el patrón se recuerda con mayor intensidad.

Puede ser que la red intente recuperar varios patrones a la vez, es decir, que varios $m_\mu \neq 0$. Si suponemos que n de los P patrones son recuperados simultáneamente con la misma intensidad $\mathbf{m}$, tenemos que

$$\mathbf{m}^{(\mathbf{n})} = m_n(\underbrace{1, 1, \ldots, 1}_{n}, \underbrace{0, 0, \ldots, 0}_{P-n}) \implies \mathbf{m}^2 = nm_n^2$$

y aplicando esto en la expresión anterior, tenemos

$$m_n = \pm\sqrt{\frac{3}{2}(nm_n^2 - \epsilon)} \implies m_n^2 = \frac{3}{2}(nm_n^2 - \epsilon) \implies m_n^2\left(1 - \frac{3}{2}n\right) + \frac{3}{2}\epsilon = 0$$

$$\implies m_n^2 = \frac{\frac{3}{2}\epsilon}{\frac{3}{2}n - 1} = \frac{3\epsilon}{3n-2} \implies m_n = \pm\sqrt{\frac{3\epsilon}{3n-2}}$$

de manera que para n mas grande se debilita la intensidad del recuerdo m_n, para cada patron. Para finalizar remarquemos dos puntos importantes:

- **Estabilidad:** de todas las soluciones que existen ($n = 1, 2, 3, \ldots$), la de $n = 1$ (patrón puro) es la que tiene la energía libre más baja y, por tanto, es la que la red "prefiereüsar.

- **Límite de Capacidad:** todo esto es válido mientras el número de patrones P no crezca demasiado. Si P se acerca al 14 % de N ($P \approx 0{,}14N$), el ruido de interferencia entre patrones destruiría incluso estas soluciones que hemos calculado.

4.4.2. High-load regime

En esta sección analizaremos el modelo de Hopfield en el **régimen de alta carga**, es decir, cuando el número de patrones P crece linealmente con el tamaño del sistema N, es decir, $P/N \to \alpha$ con $\alpha > 0$ cuando $N \to \infty$ y $P \to \infty$., donde α (**parámetro de carga**) mide qué tan "llena" está la memoria de la red. En este caso, la técnica usada en la sección anterior no funciona, ya que el número de órdenes de parámetro m_μ crece con N, y no podemos aplicar el método del punto de silla como antes. Para resolver este problema, usaremos la técnica del **replica trick**, que nos permitirá calcular la presión estadística promediada sobre el ruido.

En baja carga, los patrones son casi ortogonales y no se molestan entre sí. En carga alta, el sumatorio de los patrones que no estás intentando recordar actúa como un ruido de interferencia (crosstalk). Si intentas recordar el patrón 1, los otros $P-1$ patrones generan un ruido que tiene una varianza proporcional a α. Este ruido actúa como una temperatura efectiva adicional que puede desestabilizar el recuerdo.

En baja carga solo usábamos m (el solapamiento). En carga alta necesitamos dos parámetros para describir el sistema:

- m (**Magnetización/Memoria**): mide cuánto se parece la red al patrón que queremos recuperar.
- q (**Parámetro de Edwards-Anderson**): mide cuánto se parecen dos réplicas distintas entre sí. Si $q > 0$ están correladas. Si además, $m = 0$, la red está çongelada.[en] una configuración aleatoria que no es un recuerdo (esto se llama **fase de Spin Glass o Vidrio de Espín**).

Para resolver las ecuaciones, solemos aplicar la **hipótesis de Simetría de Réplicas**: suponemos que todas las copias del sistema se comportan de la misma manera y tienen el mismo solapamiento entre ellas ($q_{ab} = q$ para todo a, b).

Observación. Aunque esta hipótesis no es exacta (existe algo llamado "Replica Symmetry Breaking"), es la que se usa para obtener la solución estándar del modelo.

El análisis muestra que la memoria no desaparece poco a poco, sino que hay un colapso brusco:

- Para una temperatura $T > 0$, si $\alpha < \alpha_c(T)$, la red puede recordar (fase de recuperación).
- Si $\alpha > \alpha_c(T)$, el ruido de los patrones almacenados es tan fuerte que la red colapsa y ya no puede recuperar ningún patrón, cayendo en un estado de confusión total (**Spin Glass**).

Replica Trick

Como el desorden está congelado (quenched), necesitamos calcular $\mathbb{E}[\ln Z]$. Matemáticamente, promediar un logaritmo es una pesadilla. La estrategia consiste en usar la identidad:

$$\mathbb{E}[\ln Z] = \lim_{n \to 0} \frac{\mathbb{E}[Z^n] - 1}{n}$$

¿Cómo se interpreta esto?

- **Réplicas:** imaginamos n copias idénticas del sistema (réplicas) que comparten los mismos patrones almacenados (ξ) pero que pueden estar en estados distintos.
- **Promedio:** calculamos el promedio de Z^n (que es más fácil porque es un polinomio) y luego hacemos el límite matemático cuando el número de copias tiende a cero.

Para calcular $\mathbb{E}[Z^n]$ de forma analítica, se asume inicialmente que n es un número entero positivo. Físicamente, Z^n representa la función de partición de n copias idénticas (réplicas) del mismo sistema:

- Cada réplica tiene su propia configuración de neuronas S_i^a (donde $a = 1, \ldots, n$).
- La **hipótesis clave** es que todas las réplicas comparten exactamente los mismos patrones almacenados ξ.

Al promediar $\mathbb{E}[Z^n]$, las réplicas que antes eran independientes "se hablan" entre sí porque todas están intentando recordar los mismos patrones. Esto genera una interacción entre las copias que nos permite ver cómo el desorden afecta al sistema global.

Al resolver $\mathbb{E}[Z^n]$, aparecen términos que miden la correlación entre dos réplicas distintas, a y b

$$q_{ab} = \frac{1}{N} \sum_{i=1}^{N} S_i^a S_i^b$$

Para simplificar el problema y poder tomar el límite $n \to 0$, se introduce la **Hipótesis de Simetría de Réplicas** (RS): se asume que todas las copias del sistema son equivalentes, por lo que el solapamiento entre cualquier par de réplicas es el mismo

$$q_{ab} = q \quad \text{para todo } a \neq b$$

5. Machine Learning

5.1. Restricted Boltzmann Machines (RBMs)

Las máquinas de Boltzmann restringidas (RBMs) son modelos generativos estocásticos que pueden aprender una distribución de probabilidad sobre su conjunto de entradas. Fueron introducidas por Geoffrey Hinton y se utilizan ampliamente en el campo del aprendizaje automático, especialmente para tareas de reducción de dimensionalidad, clasificación y generación de datos.
Desde el punto de vista de la física estadística, una RBM es un sistema de espines con dos capas: una capa visible y una capa oculta. Los espines en la capa visible representan las variables observables (datos de entrada), mientras que los espines en la capa oculta capturan las características latentes o patrones en los datos. Las conexiones entre las capas visibles y ocultas son ponderadas, y no hay conexiones entre espines dentro de la misma capa (de ahí el término restringidas").

La representación mínima de una RBM consta de:

- **Capa visible** (σ^I): Contiene las unidades visibles que representan los datos de entrada.
- **Capa oculta** (σ^{II}): Contiene las unidades ocultas que capturan las características latentes de los datos.
- **Salida** (σ^{III}): En algunas arquitecturas, puede haber una capa de salida para tareas específicas.
- **Pesos** (J_{ij}): Conexiones ponderadas entre las unidades visibles y ocultas.

Hay también arquitecturas más complejas, como las Deep Boltzmann Machines, que contienen múltiples capas ocultas, que lleven el nombre de **Redes Neuronales Profundas** (Deep Neural Networks, DNNs), de ahí el nombre de **Deep Learning**.

Notación. Cuando no haya peligro de confusión, usaremos una sola variable s_i ($i = 1, \ldots, N_I + N_H + N_O$) para denotar a todas las neuronas de la red.

Para que estas máquinas funcionen y "aprendan" correctamente, deben cumplir con ciertas condiciones de los sistemas físicos en equilibrio:

- **Pesos Simétricos**: Las conexiones entre neuronas deben ser iguales en ambos sentidos ($J_{ij} = J_{ji}$).

- **Sin Auto-interacción**: Se excluye que una neurona influya sobre sí misma ($J_{ii} = 0$).
- **Función de Energía (Hamiltoniano)**: El estado de la red se describe mediante una energía total definida por

$$H_N(s|w) = -\frac{1}{2}\sum_{i,j} J_{ij}s_i s_j - \sum_i b_i s_i \tag{5.1}$$

- **Distribución de Boltzmann**: Bajo estas premisas, la red converge a una distribución de probabilidad única donde los estados de menor energía son los más probables

$$P(s) = Z^{-1}\exp(-\beta H_N(s|w))$$

A diferencia de los modelos deterministas, aquí el cambio de estado de una neurona es estocástico (probabilístico). La regla de actualización viene dada por

$$P(s_i \to s_i) = \frac{1}{2}[1 + \tanh(\beta s_i h_i(s))]$$

esta regla asegura que el sistema alcance eventualmente el equilibrio térmico descrito por la función de partición Z.

Aprendizaje en RBMs

El objetivo del aprendizaje en una RBM es ajustar los pesos J_{ij} y los sesgos b_i para que la distribución de probabilidad de la red ($P(\sigma^I, \sigma^{II})$) se parezca lo más posible a la distribución de los datos reales ($Q(\sigma^I, \sigma^{II})$). Consideramos el **Principio de Máxima Verosimilitud**, el cual establece que los parámetros del modelo (J_{ij} y b_i) deben ajustarse para maximizar la probabilidad de observar los datos dados. En términos prácticos, esto se traduce en minimizar la divergencia entre las dos distribuciones (**divergencia de Kullback-Leibler**).

Definición 5.1.1. La **divergencia Kullback-Leibler (KL)** entre dos distribuciones de probabilidad, Q (la real) y P (la del modelo), se define siempre como sigue

$$\Delta(Q, P) = \sum_x Q(x)\log\left(\frac{Q(x)}{P(x)}\right)$$

Dada la definicón anterior, el procedimiento de entrenamiento se describe mediante la divergencia de Kullback-Leibler entre las distribuciones Q y P:

$$\Delta(Q, P) = \sum_{\sigma^I, \sigma^{III}} Q(\sigma^I, \sigma^{III})\log\left(\frac{Q(\sigma^I, \sigma^{III})}{P(\sigma^I, \sigma^{III})}\right)$$

Minimizar esta divergencia con respecto a los pesos y sesgos nos lleva a las siguientes reglas de actualización:

$$\Delta J_{ij} = -\epsilon\frac{\partial\Delta(Q, P)}{\partial J_{ij}}, \qquad \Delta b_i = -\epsilon\frac{\partial\Delta(Q, P)}{\partial b_i}$$

donde $\epsilon \ll 1$ es la tasa de aprendizaje. La variación de la distancia KL (o divergencia de Kullack-Leibler) es (hasta el orden cuadrático)

$$\delta\Delta(Q,P) = \sum_{ij} \frac{\partial\Delta(Q,P)}{\partial J_{ij}}\Delta J_{ij} + \sum_{i} \frac{\partial\Delta(Q,P)}{\partial b_i}\Delta b_i + O(\epsilon^2) = \\ = -\left[\sum_{ij}\left(\frac{\partial\Delta(Q,P)}{\partial J_{ij}}\right)^2 + \sum_{i}\left(\frac{\partial\Delta(Q,P)}{\partial b_i}\right)^2\right] + O(\epsilon^2)$$

En el aprendizaje supervisado, el docente proporciona un input (σ^I) y un output deseado (σ^{III}). Durante el entrenamiento, las neuronas de la primera capa (σ^I) no fluctúan libremente; se mantienen fijas (fixed) con los valores de los datos de entrada. El sistema ya no busca cualquier configuración aleatoria, sino que busca la mejor configuración de las capas ocultas (σ^{II}) y de salida (σ^{III}) dada esa entrada específica.

Definición 5.1.2. Originalmente, tenemos que la **probabilidad conjunta de un estado** $s = (\sigma^I, \sigma^{II}, \sigma^{III})$ viene dado por la distribución de Boltzmann de la siguiente forma

$$P(s) = \frac{e^{-\beta H_N(s)}}{Z}, \quad \text{donde } Z = \sum_{\sigma^I,\sigma^{II},\sigma^{III}} e^{-\beta H_N(s)}$$

Aquí, Z es la suma sobre todas las combinaciones posibles de todas las capas.

La probabilidad conjunta de lo que "vemos" (entrada y salida) se obtiene sumando sobre todos los estados posibles de lo que "no vemos" (la capa oculta)

$$P(\sigma^I, \sigma^{III}) = \sum_{\sigma^{II}} P(\sigma^I, \sigma^{II}, \sigma^{III})$$

de hecho, como la capa de entrada está fija, podemos considerar la distribución condicional $P(\sigma^{II}, \sigma^{III}|\sigma^I)$, teniendo por tanto

$$P(s) = P(\sigma^{II}, \sigma^{III}|\sigma^I)P(\sigma^I) = P(\sigma^{II}, \sigma^{III}|\sigma^I)Q(\sigma^I)$$

donde dado que estamos en entrenamiento, la probabilidad de observar una entrada específica $P(\sigma^I)$ es exactamente la distribución de los datos reales $Q(\sigma^I)$. Tenemos entonces

$$P(\sigma^I, \sigma^{III}) = Q(\sigma^I)\sum_{\sigma^{II}} P(\sigma^{II}, \sigma^{III}|\sigma^I)$$

pero

$$P(\sigma^{II}, \sigma^{III} \mid \sigma^I) = \frac{e^{-\beta H_N(s)}}{Z(\sigma^I)}$$

donde tenemos la siguiente definición.

Definición 5.1.3. Definimos la **función de partición restringida o condicionada**, como

$$Z(\sigma^I) = \sum_{\sigma^{II},\sigma^{III}} e^{-\beta H_N(s)}$$

que representa la suma de los "pesos estadísticos" de la red, pero solo para aquellas configuraciones que son compatibles con la entrada σ^I que hemos fijado.

Finalmente, tenemos que la probabildiad conjunta de observar una entrada y una salida concretas es

$$P(\sigma^I, \sigma^{III}) = Q(\sigma^I)\frac{Z(\sigma^I, \sigma^{III})}{Z(\sigma^I)}$$

donde claramente $Z(\sigma^I, \sigma^{III}) = \sum_{\sigma^{II}} e^{-\beta H_N(s)}$ es la función de partición restringida a las configuraciones compatibles con la entrada σ^I y la salida σ^{III}.

Para calcular las reglas de actualización, necesitamos las derivadas parciales de la divergencia KL con respecto a los pesos y sesgos, por lo que primero veremos un desarrollo de dicha divergencia

$$\Delta(Q,P) = \sum_x Q(x)\log\left(\frac{Q(x)}{P(x)}\right) = \sum_x Q(x)\log(Q(x)) - \sum_x Q(x)\log(P(x)) = -S(Q) - \sum_x Q(x)\log$$

donde como $S(Q)$ es la entropía de la distribución real Q, y al ser independiente de los parámetros del modelo, al derivar solo necesitamos preocuparnos por la segunda parte. Por tanto, tenemos

$$\begin{aligned}
\Delta(Q,P) &= -\sum_{\sigma^I,\sigma^{III}} Q(\sigma^I,\sigma^{III})\log\big(P(\sigma^I,\sigma^{III})\big) = -\sum_{\sigma^I,\sigma^{III}} Q(\sigma^I,\sigma^{III})\log\left(Q(\sigma^I)\frac{Z(\sigma^I,\sigma^{III})}{Z(\sigma^I)}\right) = \\
&= -\sum_{\sigma^I,\sigma^{III}} Q(\sigma^I,\sigma^{III})\log\big(Q(\sigma^I)\big) - \sum_{\sigma^I,\sigma^{III}} Q(\sigma^I,\sigma^{III})\log\left(\frac{Z(\sigma^I,\sigma^{III})}{Z(\sigma^I)}\right) = \\
&\overset{(*)}{=} -\sum_{\sigma^I,\sigma^{III}} Q(\sigma^I,\sigma^{III})\left[\log\big(Z(\sigma^I,\sigma^{III})\big) - \log\big(Z(\sigma^I)\big)\right] = \\
&\overset{(**)}{=} -\frac{1}{\beta}\sum_{\sigma^I,\sigma^{III}} Q(\sigma^I,\sigma^{III})\left[\log\big(Z(\sigma^I,\sigma^{III})\big) - \log\big(Z(\sigma^I)\big)\right]
\end{aligned}$$

donde en $(*)$ hemos usado que es una constante (independiente de los parámetros del modelo) y la podemos omitir (no afectará al cambio de parámetros) y en $(**)$ hemos introducido el factor $1/\beta$ para facilitar los cálculos posteriores.

Como bien hemos comentado, buscamos minimizar la divergencia KL, por lo que necesitamos obtener los mínimos de $\Delta(Q,P)$ con respecto a los pesos y sesgos. Para encontrar el mínimo de una función compleja, utilizamos el algoritmo de **gradiente descendente**. La derivada parcial

$$\frac{\partial \Delta}{\partial J_{ij}}$$

nos indica en qué dirección y con qué intensidad cambia el error cuando modificamos un peso específico. Destacar que lo mismo ocurre para los sesgos b_i. Para ello, primero calcularemos estas derivadas parciales

$$\begin{aligned}
\frac{\partial}{\partial J_{ij}}\log Z(\sigma^I,\sigma^{II}) &= -\beta\sum_{\sigma^{II}}\frac{\partial H_N(s)}{\partial J_{ij}}\frac{e^{-\beta H_N}}{Z(\sigma^I,\sigma^{III})} = -\beta\sum_{\sigma^{II}}\frac{\partial H_N}{\partial J_{ij}}P(\sigma^{II}|\sigma^I,\sigma^{III}), \\
\frac{\partial}{\partial J_{ij}}\log Z(\sigma^I) &= -\beta\sum_{\sigma^{II},\sigma^{III}}\frac{\partial H_N(s)}{\partial J_{ij}}\frac{e^{-\beta H_N(s)}}{Z(\sigma^I)} = -\beta\sum_{\sigma^{II},\sigma^{III}}\frac{\partial H_N(s)}{\partial J_{ij}}P(\sigma^{II},\sigma^{III}|\sigma^I)
\end{aligned}$$

Poniendo esto todo junto en la expresión de la divergencia KL, obtenemos

$$\frac{\partial \Delta(Q,P)}{\partial J_{ij}} = -\frac{1}{\beta}\sum_{\sigma^I,\sigma^{III}} Q(\sigma^I,\sigma^{III})\left[\frac{\partial}{\partial J_{ij}}\log Z(\sigma^I,\sigma^{III}) - \frac{\partial}{\partial J_{ij}}\log Z(\sigma^I)\right] =$$

$$= -\frac{1}{\beta}\sum_{\sigma^I,\sigma^{III}} Q(\sigma^I,\sigma^{III})\left[-\beta\sum_{\sigma^{II}}\frac{\partial H_N(s)}{\partial J_{ij}}P(\sigma^{II}|\sigma^I,\sigma^{III}) + \beta\sum_{\sigma^{II},\sigma^{III}}\frac{\partial H_N(s)}{\partial J_{ij}}P(\sigma^{II},\sigma^{III}|\sigma^I\right.$$

$$= \sum_{\sigma^I,\sigma^{III}} Q(\sigma^I,\sigma^{III})\left[\sum_{\sigma^{II}}\frac{\partial H_N(s)}{\partial J_{ij}}P(\sigma^{II}|\sigma^I,\sigma^{III}) - \sum_{\sigma^{II},\sigma^{III}}\frac{\partial H_N(s)}{\partial J_{ij}}P(\sigma^{II},\sigma^{III}|\sigma^I)\right]$$

y finalmente queda como

$$\frac{\partial \Delta(Q,P)}{\partial J_{ij}} = \sum_{\sigma^I,\sigma^{II},\sigma^{III}}\frac{\partial H_N(s)}{\partial J_{ij}}P(\sigma^{II}|\sigma^I,\sigma^{III})Q(\sigma^I,\sigma^{III})$$
$$- \sum_{\sigma^I,\sigma^{II},\sigma^{III},\bar{\sigma}^{III}}\frac{\partial H_N(\bar{s})}{\partial J_{ij}}P(\sigma^{II},\bar{\sigma}^{III}|\sigma^I)Q(\sigma^I,\sigma^{III})$$

donde hemos usado la notación $\bar{s} = (\sigma^I, \sigma^{II}, \bar{\sigma}^{III})$ para distinguir las dos sumas. Podemos simplificar la expresión anterior, para ellos veámos la simplificación del segundo término como sigue

$$\sum_{\sigma^I,\sigma^{II},\sigma^{III},\bar{\sigma}^{III}}\frac{\partial H_N(\bar{s})}{\partial J_{ij}}P(\sigma^{II},\bar{\sigma}^{III}|\sigma^I)Q(\sigma^I,\sigma^{III}) = \sum_{\sigma^I,\sigma^{II},\bar{\sigma}^{III}}\left[\frac{\partial H_N(\bar{s})}{\partial J_{ij}}P(\sigma^{II},\bar{\sigma}^{III}|\sigma^I)\left(\sum_{\sigma^{III}}Q(\sigma^I,\sigma^{III})\right.\right.$$
$$\overset{(*)}{=} \sum_{\sigma^I,\sigma^{II},\bar{\sigma}^{III}}\frac{\partial H_N(\bar{s})}{\partial J_{ij}}P(\sigma^{II},\bar{\sigma}^{III}|\sigma^I)Q(\sigma^I)$$

donde en $(*)$ hemos usado que por definición de probabilidad marginal, si sumamos la distribución conjunta de los datos $Q(\sigma^I, \sigma^{III})$ sobre todos los posibles estados de la salida σ^{III}, obtenemos la probabilidad de la entrada. Por tanto, la expresión de la derivada parcial queda como

$$\frac{\partial \Delta(Q,P)}{\partial J_{ij}} = \sum_{\sigma^I,\sigma^{II},\sigma^{III}}\frac{\partial H_N(s)}{\partial J_{ij}}P(\sigma^{II}|\sigma^I,\sigma^{III})Q(\sigma^I,\sigma^{III})$$
$$- \sum_{\sigma^I,\sigma^{II},\sigma^{III}}\frac{\partial H_N(s)}{\partial J_{ij}}P(\sigma^{II},\sigma^{III}|\sigma^I)Q(\sigma^I)$$

Recordando la expresión del Hamiltoniano (5.1), vamos a deducir un resultado importante para el aprendizaje en RBMs. Primero observamos que

$$\frac{\partial H_N(s)}{\partial J_{ij}} = -s_i s_j$$

Definición 5.1.4. En física estadística, el **valor esperado o promedio** de una cantidad A bajo una distribución $P(x)$ se define como

$$\langle A\rangle = \sum_x A(x)P(x)$$

Dada la anteriore definición, podemos interpretar las sumas ponderadas en la expresión de la derivada parcial como valores esperados bajo diferentes distribuciones, teniendo por tanto

$$\sum_{\sigma^{II}} \frac{\partial H_N(s)}{\partial J_{ij}} P(\sigma^{II} \mid \sigma^I, \sigma^{III}) = \langle \frac{\partial H_N}{\partial J_{ij}} \rangle_{cond_1} = -\langle s_i s_j \rangle_{cond_1}$$
$$\sum_{\sigma^{II},\sigma^{III}} \frac{\partial H_N(s)}{\partial J_{ij}} P(\sigma^{II}, \sigma^{III} \mid \sigma^I) = \langle \frac{\partial H_N}{\partial J_{ij}} \rangle_{cond_2} = -\langle s_i s_j \rangle_{cond_2}$$

donde las etiquetas $cond_1$ y $cond_2$ significan lo siguiente

- $cond_1$ (**Fase clamped**): el promedio se hace solo sobre σ^{II}. Esto ocurre porque tanto la entrada (σ^I) como la salida (σ^{III}) están fijas por los datos del maestro.
- $cond_2$ (**Fase free**): el promedio se hace sobre σ^{II} y σ^{III}. Aquí solo la entrada (σ^I) está fija; la red es "libre" de decidir qué salida generar.

Tenemos ahora las siguientes expresiones para los dos términos de la derivada parcial, veámoslo por partes separadas

$$\sum_{\sigma^I} Q(\sigma^I) \sum_{\sigma^{II},\sigma^{III}} \frac{\partial H_N(s)}{\partial J_{ij}} P(\sigma^{II}, \sigma^{III} \mid \sigma^I) = -\sum_{\sigma^I} Q(\sigma^I) \langle s_i s_j \rangle_{cond_2} = -\langle s_i s_j \rangle_{cond_2}$$

$$\begin{aligned}\sum_{\sigma^I,\sigma^{III}} Q(\sigma^I, \sigma^{III}) \sum_{\sigma^{II}} \frac{\partial H_N(s)}{\partial J_{ij}} P(\sigma^{II} \mid \sigma^I, \sigma^{III}) &= -\sum_{\sigma^I,\sigma^{III}} Q(\sigma^I, \sigma^{III}) \langle s_i s_j \rangle_{cond_1} = \\ &= -\sum_{\sigma^I} \left[\sum_{\sigma^{III}} Q(\sigma^I, \sigma^{III}) \right] \langle s_i s_j \rangle_{cond_2} = \\ &= -\sum_{\sigma^I} Q(\sigma^I) \langle s_i s_j \rangle_{cond_1} = -\langle s_i s_j \rangle_{cond_1}\end{aligned}$$

Con lo ya obtenido, se obtiene de forma directa la siguiente proposición.

Proposición 5.1.1. *The* ***Contrastive Divergence learning algorithm*** *for Restricted Boltzmann Machines is given by the following gradient of the Kullback-Leibler divergence with respect to the weights:*

$$\frac{\partial \Delta(Q, P)}{\partial J_{ij}} = -\left(\langle s_i s_j \rangle_{clamped} - \langle s_i s_j \rangle_{free} \right)$$

Por supuesto, un resultado completamente análogo se obtiene para los sesgos, por lo que finalmente llegamos a la regla de actualización de los parámetros

$$\Delta J_{ij} = \epsilon \left(\langle s_i s_j \rangle_{clamped} - \langle s_i s_j \rangle_{free} \right),$$
$$\Delta b_i = \epsilon \left(\langle s_i \rangle_{clamped} - \langle s_i \rangle_{free} \right).$$

Observación. Como las neuronas s_i y s_j son variables que fluctúan (pueden ser $+1$ o -1), el término $s_i s_j$ es el producto de sus estados en un momento dado. El valor

$$\langle s_i s_j \rangle$$

es el promedio de esos productos sobre muchas configuraciones posibles. Un valor cercano a 1 significa que las neuronas casi siempre están de acuerdo (se activan juntas), mientras que un valor cercano a 0 significa que no tienen relación entre sí. Este valor esperado es crucial para entender cómo las conexiones entre neuronas se fortalecen o debilitan durante el proceso de aprendizaje en una RBM.

- $\langle s_i s_j \rangle_{\text{clamped}}$: es la correlación que el "maestro" impone a la red. Indica qué neuronas deberían activarse juntas según los datos reales.

- $\langle s_i s_j \rangle_{\text{free}}$: es la correlación que la red genera por sí sola cuando "sueña" o funciona libremente. Indica qué neuronas cree la red que deben ir juntas basándose en sus pesos actuales.

5.2. Duality between Hopfield model and RBMs

Hemos mencionado cómo la red de Hopfield es un modelo representativo para la **recuperación** y las máquinas de Boltzmann son sistemas fundamentales para el **aprendizaje**. Desde un punto de vista tanto intuitivo como formal, el aprendizaje y la recuperación no son dos operaciones independientes, sino dos aspectos complementarios de la cognición. Por lo tanto, debe ser posible recuperar la regla de Hebb para el aprendizaje también partiendo de las máquinas de Boltzmann (restringidas).

El objetivo principal de esta sección es demostrar la dualidad matemática entre las Máquinas de Boltzmann Restringidas (RBM) y el modelo de Hopfield. En esencia, buscamos probar que lo que una RBM "aprende" es equivalente a lo que una red de Hopfield "almacena". Lo que buscaremos es demostrar que partiendo de una RBM simple de dos capas, se puede recuperar la regla de Hebb del modelo de Hopfield.

Para ver esto, por simplicidad, a partir de ahora las usaremos en su representación más simple, es decir, como redes básicas de dos capas. Usamos el símbolo σ_i, $i \in \{1, \dots, N\}$ para las neuronas en la capa visible, z_μ, $\mu \in \{1, \dots, P\}$ para las de la capa oculta y w_i^μ para etiquetar los enlaces (o pesos) entre la neurona i en una capa y la neurona μ en la otra capa.

Podemos entonces escribir la función de costo para la máquina de Boltzmann como

$$H_N(\sigma, z, w) = -\frac{1}{\sqrt{N}} \sum_{i=1}^{N} \sum_{\mu=1}^{P} w_i^\mu \sigma_i z_\mu.$$

Este Hamiltoniano representa, en la jerga de la mecánica estadística, un vidrio de espines bipartido (bipartite spin-glass). Para estudiar el diagrama de fases relacionado, trabajamos la maquinaria de la mecánica estadística, comenzando por escribir la **presión promediada (quenched) de la máquina de Boltzmann** como

$$A_{N,\beta} = \frac{1}{N} \mathbb{E} \ln \sum_{\sigma} \sum_{z} \exp\left(\beta \frac{1}{\sqrt{N}} \sum_{i=1}^{N} \sum_{\mu=1}^{P} w_i^\mu \sigma_i z_\mu \right).$$

La propiedad fundamental de las RBM es que no hay conexiones entre las neuronas de la misma capa (capa oculta en este caso). Esto permite intercambiar el orden de la sumatoria y el producto en la exponencial, obteniendo

$$\exp\left(\frac{\beta}{\sqrt{N}}\sum_{\mu=1}^{P} z_\mu\left(\sum_{i=1}^{N} w_i^\mu \sigma_i\right)\right) = \prod_{\mu=1}^{P}\exp\left(z_\mu\cdot\frac{\beta}{\sqrt{N}}\sum_{i=1}^{N} w_i^\mu\sigma_i\right)$$

Al aplicar la suma sobre todas las configuraciones posibles de las neuronas ocultas $z_\mu \in \{-1, 1\}$, la suma de un producto se convierte en el producto de las sumas

$$\sum_{z}\prod_{\mu}(\dots) = \prod_{\mu=1}^{P}\sum_{z_\mu=\pm1}\exp\left(z_\mu\cdot\frac{\beta}{\sqrt{N}}\sum_{i=1}^{N} w_i^\mu\sigma_i\right) \overset{(*)}{=} \prod_{\mu}^{P} 2\cosh\left(\frac{\beta}{\sqrt{N}}\sum_{i=1}^{N} w_i^\mu\sigma_i\right)$$

donde en $(*)$ hemos usado que $\sum_{x=\pm1} e^{ax} = e^a + e^{-a} = 2\cosh(a)$. Por ahora, tenemos la siguiente expresión para la presión promediada

$$A_{N,\beta} = \frac{1}{N}\mathbb{E}\ln\sum_{\sigma}\prod_{\mu=1}^{P} 2\cosh\left(\frac{\beta}{\sqrt{N}}\sum_{i=1}^{N} w_i^\mu\sigma_i\right)$$

Para conectar con el modelo de Hopfield, expandimos el término $\ln\cosh(x)$ usando su serie de Taylor siguiente

$$\ln\cosh(x) = \frac{x^2}{2} - \frac{x^4}{12} + \frac{x^6}{45} - \dots \implies \ln\cosh(x) \approx \frac{x^2}{2}$$

para valores pequeños de x, lo que se cumple en nuestro caso en el límite termodinámico donde $N \to \infty$ debido al factor $1/\sqrt{N}$, y por tanto tenemos

$$\ln\cosh\left(\frac{\beta}{\sqrt{N}}\sum_{i} w_i^\mu\sigma_i\right) \approx \frac{1}{2}\left(\frac{\beta}{\sqrt{N}}\sum_{i=1}^{N} w_i^\mu\sigma_i\right)^2 = \frac{\beta^2}{2N}\left(\sum_{i=1}^{N} w_i^\mu\sigma_i\right)^2$$

Aclarada esta idea, buscamos poder expreser el productorio de μ de una forma que podamos aplicar esta idea, para ello simplemente aplicaremos las funciones inversas exponencial y logaritmo de la siguiente manera

$$\begin{aligned}\prod_{\mu=1}^{P} 2\cosh\left(\frac{\beta}{\sqrt{N}}\sum_{i=1}^{N} w_i^\mu\sigma_i\right) &= \exp\left(\sum_{\mu=1}^{P}\ln\left[2\cosh\left(\frac{\beta}{\sqrt{N}}\sum_{i=1}^{N} w_i^\mu\sigma_i\right)\right]\right) = \\ &\overset{(*)}{=} \exp\left(\sum_{\mu=1}^{P}\frac{\beta^2}{2N}\sum_{i,j} w_i^\mu w_j^\mu\sigma_i\sigma_j\right) = \\ &= \exp\left(\frac{\beta^2}{2N}\sum_{i,j}\left[\sum_{\mu} w_i^\mu w_j^\mu\right]\sigma_i\sigma_j\right)\end{aligned}$$

donde en $(*)$ hemos usado la idea demostrada anteriormente y hemos obviado el factor 2 dentro del logaritmo porque genera una constante aditiva que no afecta a la

física del sistema cuando $N \to \infty$. Tenemos finalmente la expresión que buscabamos de la presión promediada

$$A_{N,\beta} \underset{N\to\infty}{\sim} \frac{1}{N} \ln \sum_{\sigma} \exp\left(\frac{\beta}{2N} \sum_{i,j=1}^{N} \left(\sum_{\mu=1}^{P} w_i^\mu w_j^\mu\right) \sigma_i \sigma_j\right)$$

donde la $\mathbb{E}$ desaparece porque la expresión final describe el Hamiltoniano efectivo que sienten las neuronas visibles. En el límite termodinámico, el promedio sobre el desorden ($\mathbb{E}$) y el comportamiento de un sistema con pesos fijos coinciden, es decir, al tomar el límite termodinámico, la esperanza sobre el desorden y el comportamiento de un sistema con pesos fijos convergen.

Comparando la expresión anterior con la del modelo de Hopfield, podemos identificar que los patrones almacenados en la red de Hopfield están dados por los pesos de la máquina de Boltzmann, es decir,

$$\xi_i^\mu = w_i^\mu$$

y por tanto, la regla de Hebb para el modelo de Hopfield

$$J_{ij} = \frac{1}{N} \sum_{\mu=1}^{P} \xi_i^\mu \xi_j^\mu$$

se recupera directamente como

$$J_{ij} = \frac{1}{N} \sum_{\mu=1}^{P} w_i^\mu w_j^\mu$$

esto demuestra la dualidad matemática entre las máquinas de Boltzmann restringidas y el modelo de Hopfield.

Observación. En física estadística, si dos sistemas tienen la misma Presión ($A_{N,\beta}$) o Energía Libre, significa que son macroscópicamente idénticos: tendrán los mismos estados estables, las mismas transiciones de fase y el mismo comportamiento de memoria.

Aunque una Máquina de Boltzmann Restringida (RBM) tiene dos capas (visible y oculta), su comportamiento efectivo para las neuronas que vemos es el de una red de una sola capa. Esto se debe a lo siguiente:

- **RBM (Bipartita)**: originalmente, las neuronas visibles (σ_i) solo interactúan con las ocultas (z_μ) a través de los pesos w_i^μ.
- **Suma sobre z (Marginalización)**: al realizar la suma estadística sobre todos los estados posibles de la capa oculta, esta capa "desaparece"matemáticamente.
- **Resultado**: lo que queda es una interacción directa entre las neuronas visibles (σ_i y σ_j) que tiene la misma forma que el Hamiltoniano de Hopfield.

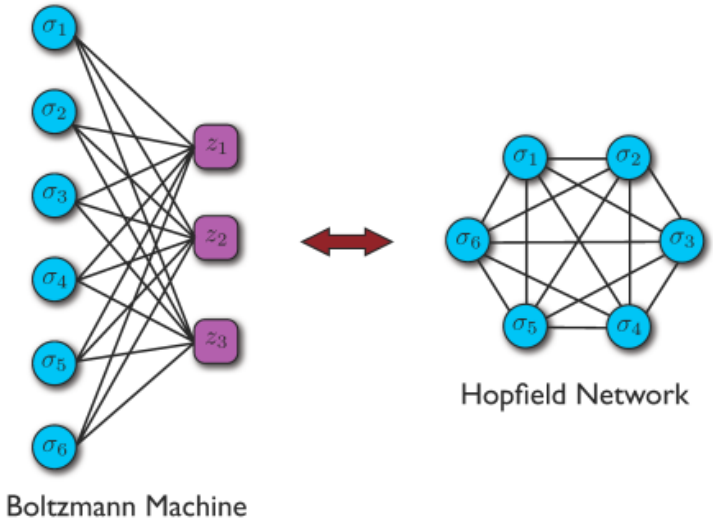


Figura 5.1: Correspondence between a two-layer restricted Boltzmann machine and an Hopfield neural network.

Podemos ver en la Figura 5.1 un esquema que ilustra esta dualidad entre una RBM de dos capas y una red de Hopfield.

Observación. La proporción entre la capa oculta (P) y la visible (N), definida matemáticamente como $\alpha = \lim_{N\to\infty} P/N$, equivale exactamente a la capacidad de almacenamiento de patrones en el modelo de Hopfield. Si la capa oculta es demasiado grande en relación con la visible (superando un umbral crítico), la red no aprende a generalizar, sino que cae en el overfitting (sobreajuste), al igual que una red de Hopfield colapsa si intentas "forzarla" a recordar demasiados patrones.