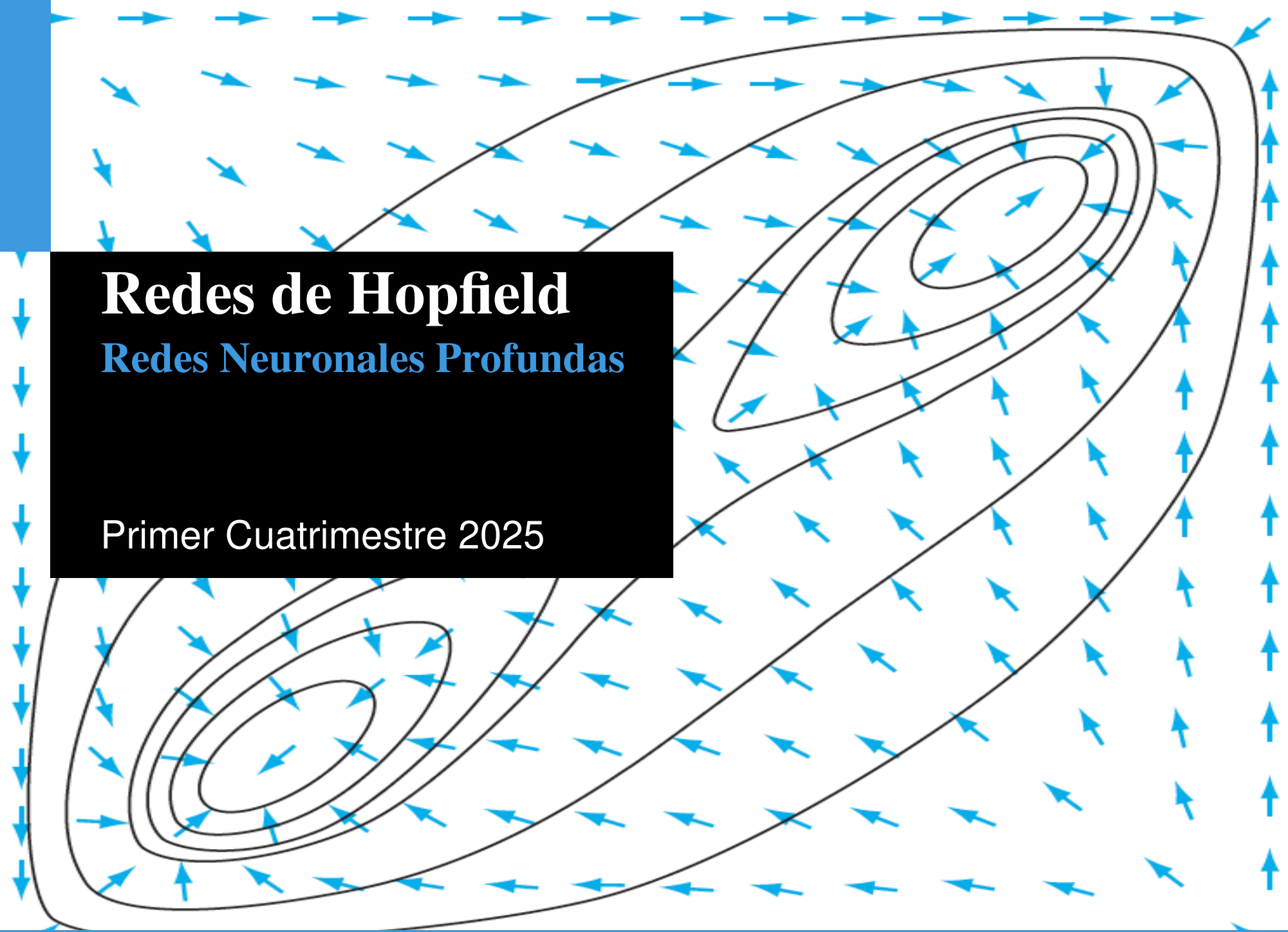
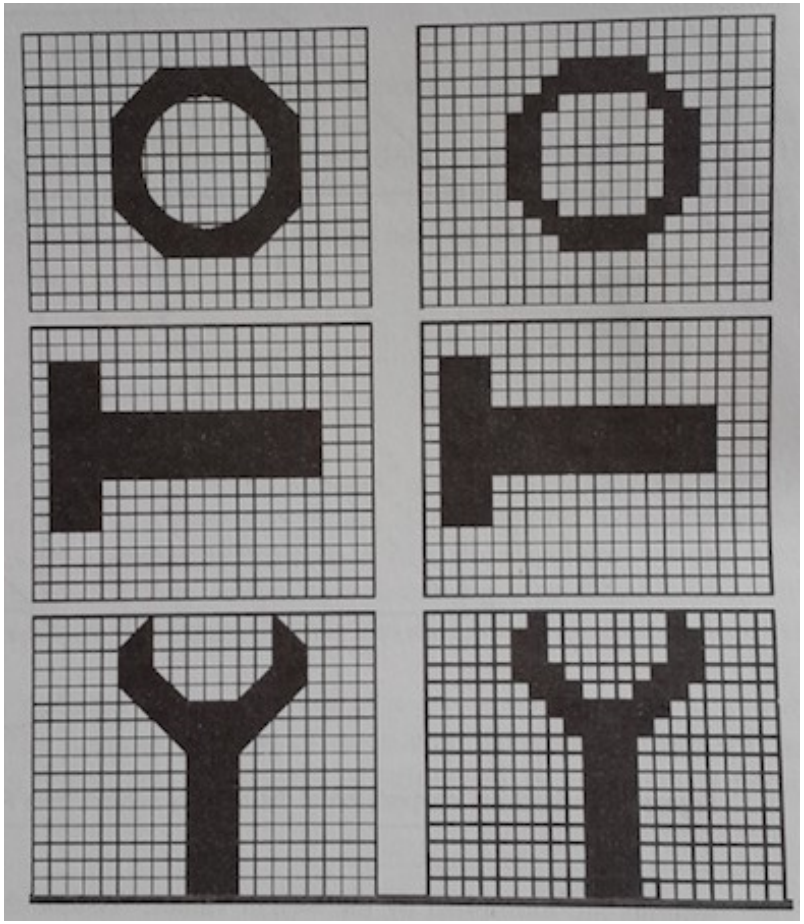


Redes de Hopfield

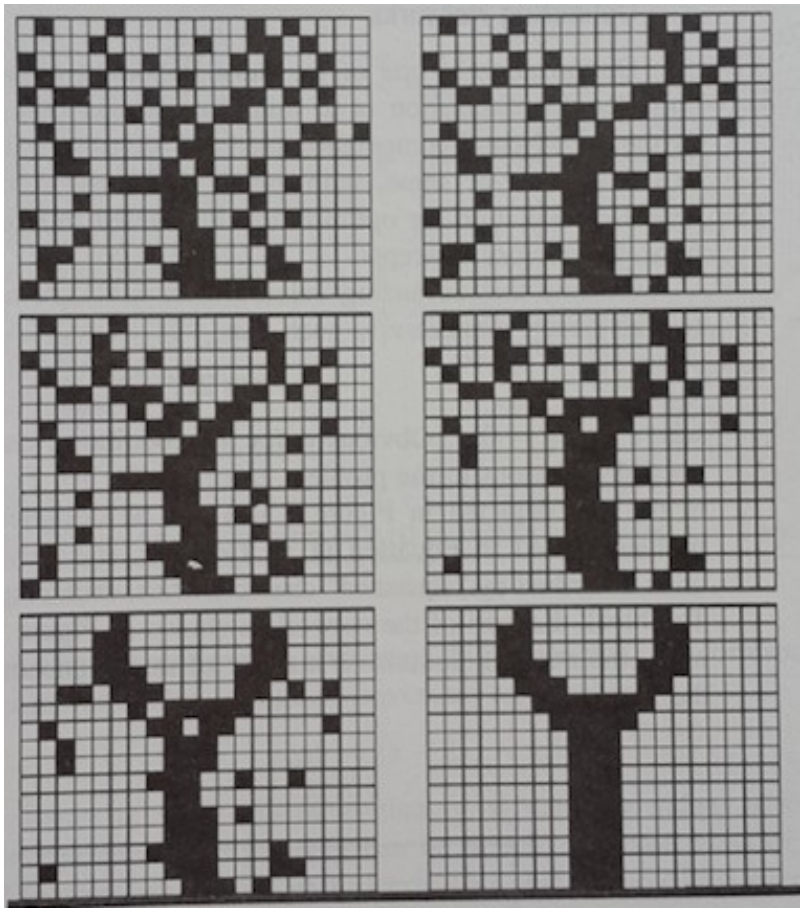
Redes Neuronales Profundas

Primer Cuatrimestre 2025

Problema: reconocer objetos en una fábrica

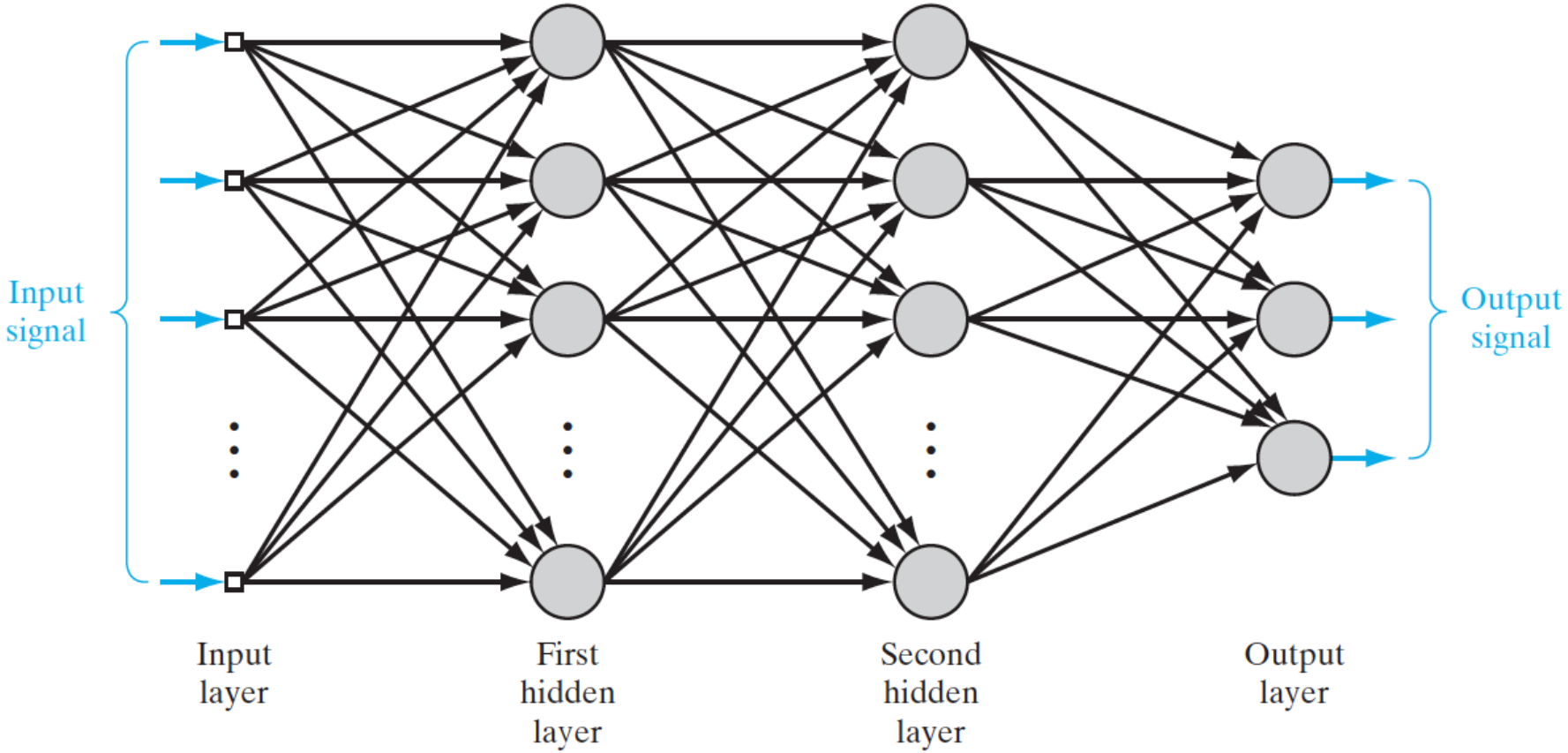


(a) Imagen digitalizada



(b) Imagen difusa

Solución 1: Redes neuronales feedforward



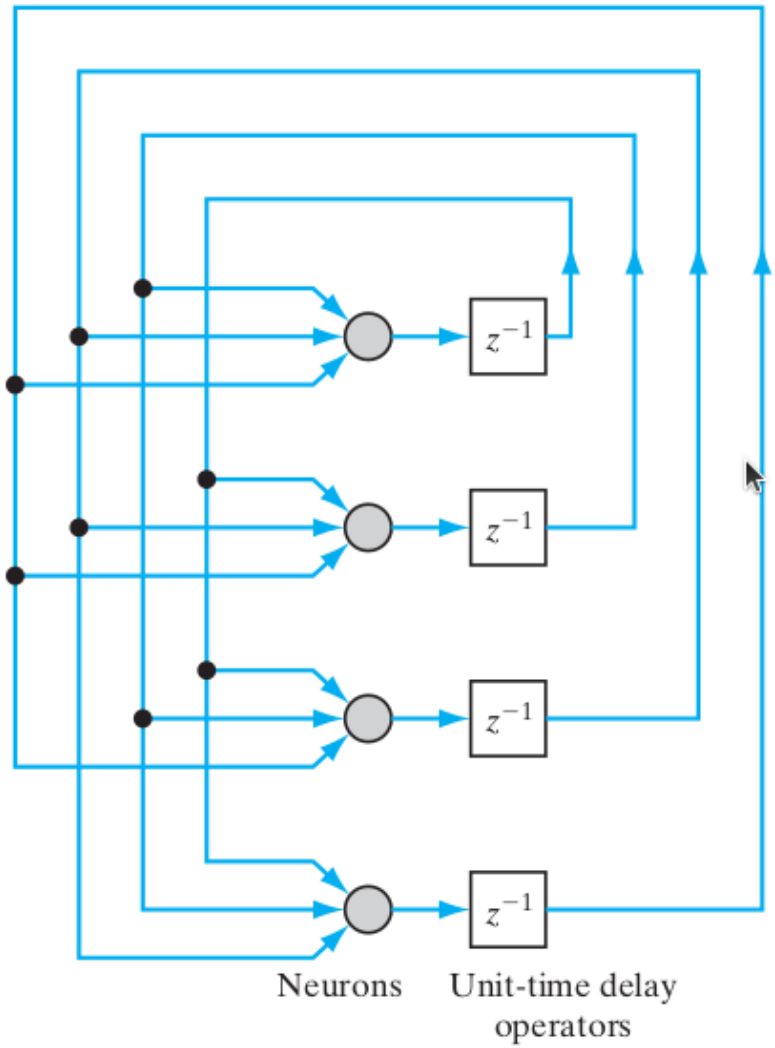
Durante el entrenamiento, hacen feedback, pero una vez entrenadas, siempre van hacia adelante.

Solución 1: Redes neuronales feedforward

Necesitaremos un dataset de imágenes etiquetadas difusas.

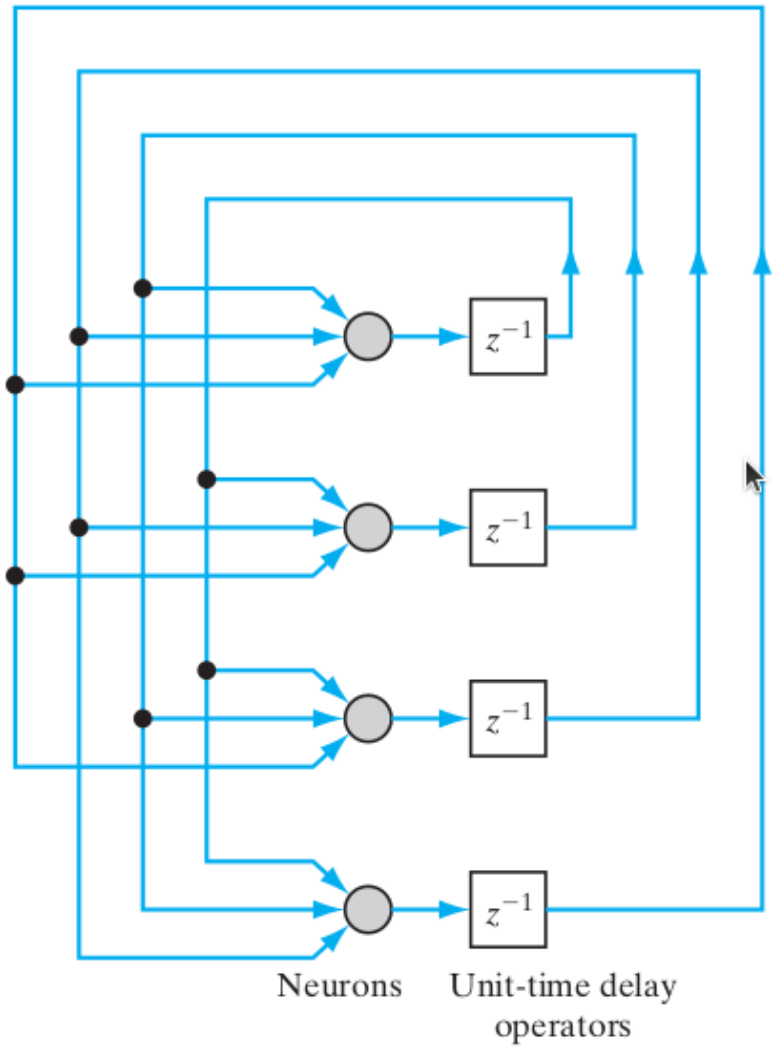
Mientras más grande el dataset, mejor será el reconocedor.

Solución 2: red de Hopfield (memorias asociativas)



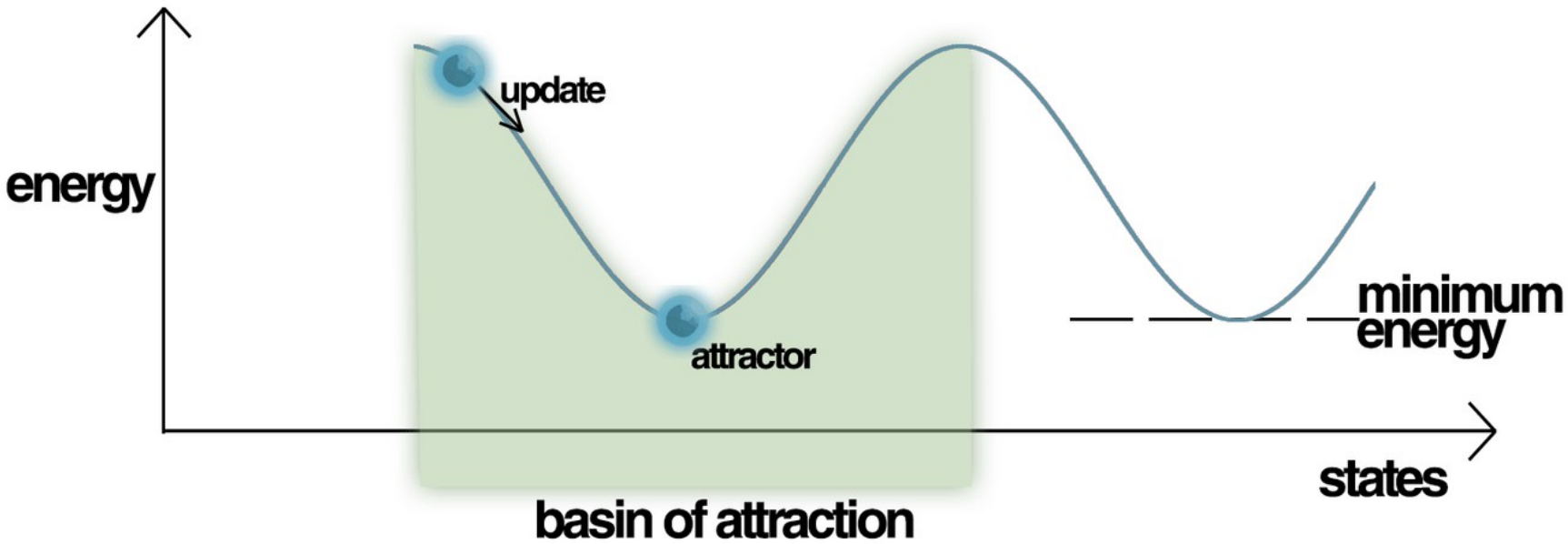
Arquitectura de una red de Hopfield de N = 4 neuronas (Recurrent Network)

Solución 2: red de Hopfield (memorias asociativas)



Modelo Continuo: auto-retroalimentación
Modelo Discreto: sin auto-retroalimentación ($w_{jj} = 0$)
Convergencia: los patterns deben ser mínimos de la función de Lyapunov

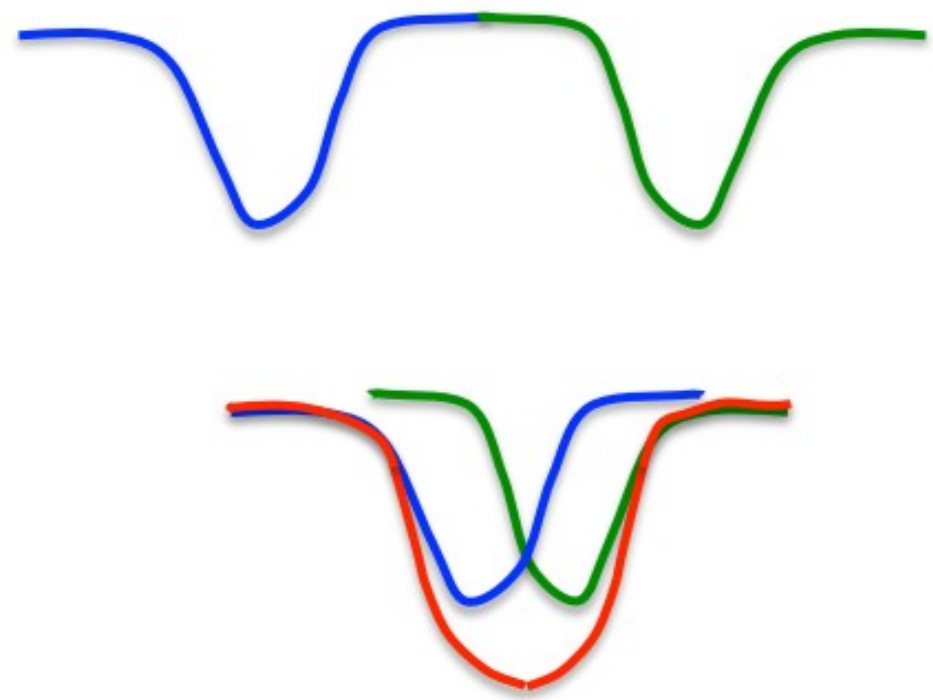
Energía en Hopfield



Gradiente de la función de Lyapunov: $\nabla V(a) = [- Wa + n - b]$

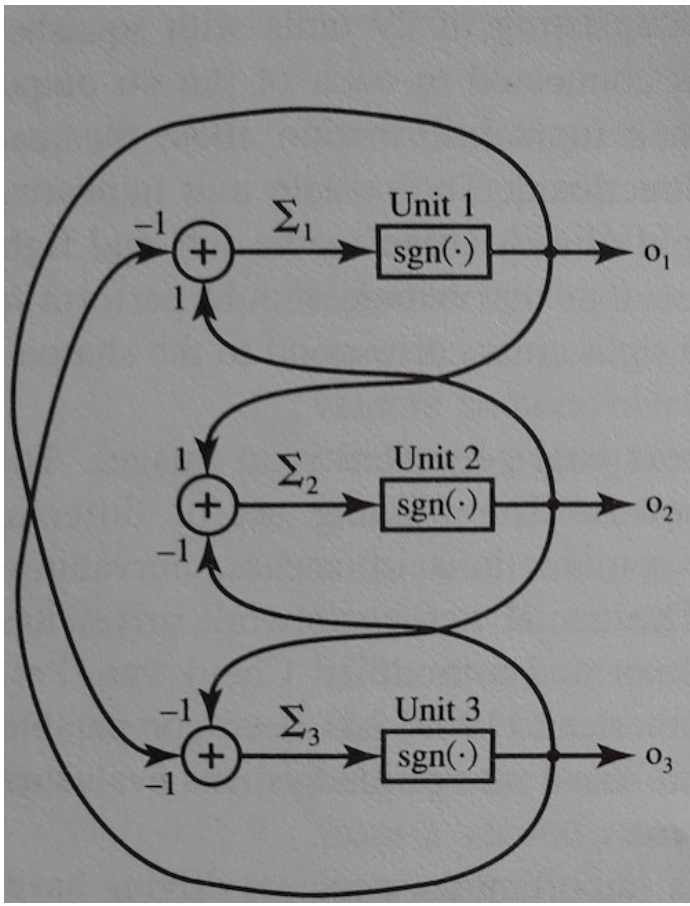
Mínima Energía $\nabla V(a) = 0$

Memorias espúreas



Los estados espúreos son también mínimos de energía, que se forman a partir de las memorias. Ejemplo: Si *m* es una memoria, *-m* será un mínimo de energía (espúreo). Combinaciones lineales de las memorias también podrían ser estados espúreos.

Red de memorias

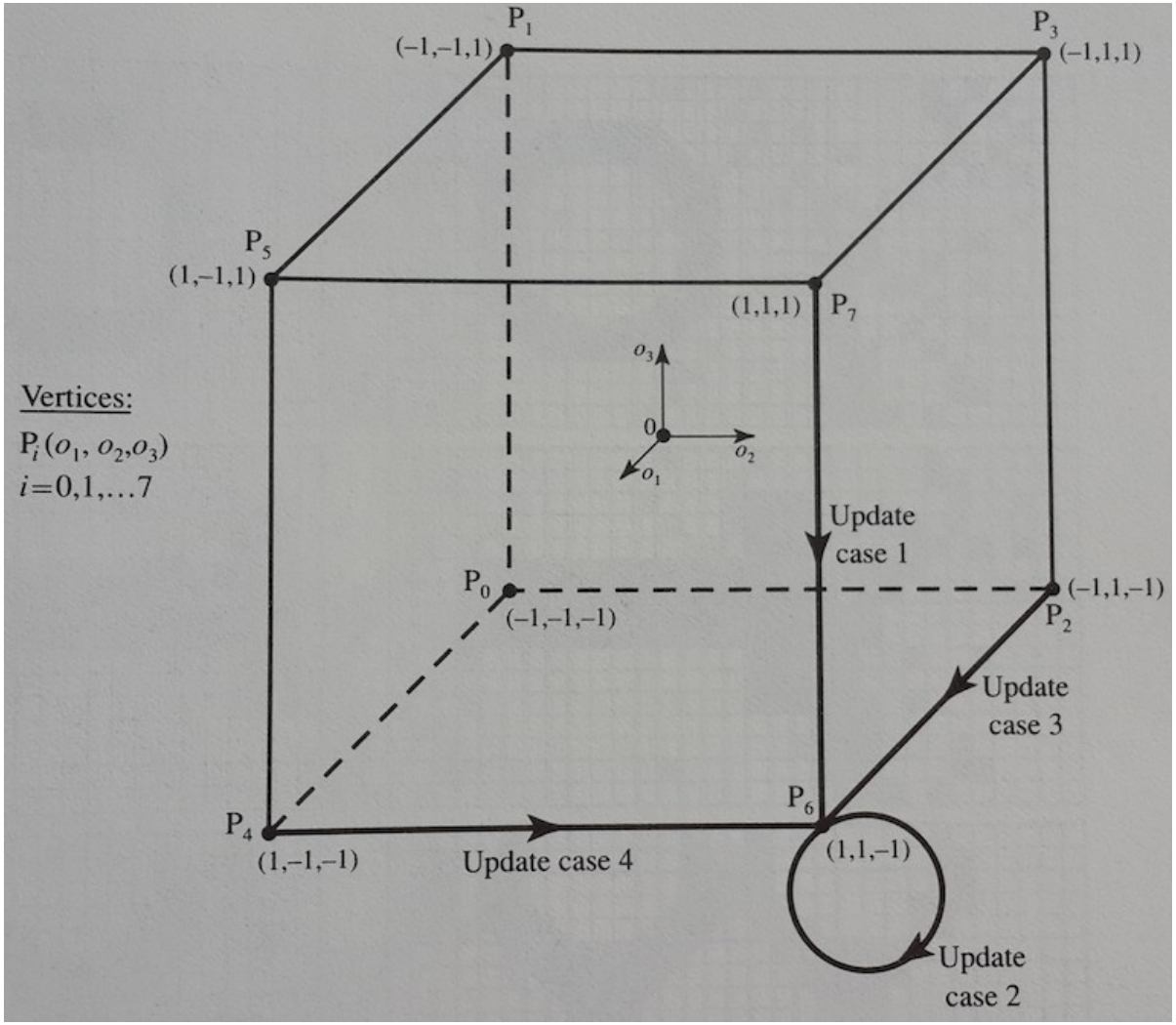


Case	Unit Number	Present Output O	Σ	sgn Σ	Next Output O
1	1	1	0	x	1
	2	1	0	x	1
	3	1	−2	−1	(−1)
2	1	1	2	1	1
	2	1	2	1	1
	3	−1	−2	−1	−1
3	1	−1	2	1	(1)
	2	1	0	x	1
	3	−1	0	x	−1
4	1	1	0	x	1
	2	−1	2	1	(1)
	3	−1	0	x	−1

x stands for sgn (0).
Encircled are updated outputs.

Tres neuronas y cuatro patterns.

Redes neuronales recurrentes



Caso discreto, proyecta el pattern en el espacio de las memorias fundamentales.
En un vértice del hipercubo unitario cuando se minimiza la Energía (P_6)

Modelo de Hopfield: Algoritmo

1. *Learning.* Let $\xi_1, \xi_2, ..., \xi_\mu$ denote a known set of N-dimensional fundamental memories. Use the outer-product rule (i.e., Hebb's postulate of learning) to compute the synaptic weights of the network as

$$w_{ji} = \begin{cases} \frac{1}{N}\sum_{\mu=1}^{M} \xi_{\mu,j}\,\xi_{\mu,i}, & j \neq i \\ 0, & j = i \end{cases}$$

where w_{ji} is the synaptic weight from neuron i to neuron j. The elements of the vector ξ_μ equal ± 1. Once they are computed, the synaptic weights are kept fixed.

2. *Initialization.* Let ξ_{probe} denote an unknown N-dimensional input vector (probe) presented to the network. The algorithm is initialized by setting

$$x_j(0) = \xi_{j,\,\text{probe}}, \qquad j = 1, ..., N$$

where $x_j(0)$ is the state of neuron j at time $n = 0$ and $\xi_{j,\text{probe}}$ is the jth element of the probe ξ_{probe}.

3. *Iteration Until Convergence.* Update the elements of state vector $\mathbf{x}(n)$ asynchronously (i.e., randomly and one at a time) according to the rule

$$x_j(n+1) = \text{sgn}\left(\sum_{i=1}^{N} w_{ji}x_i(n)\right), \qquad j = 1, 2, ..., N$$

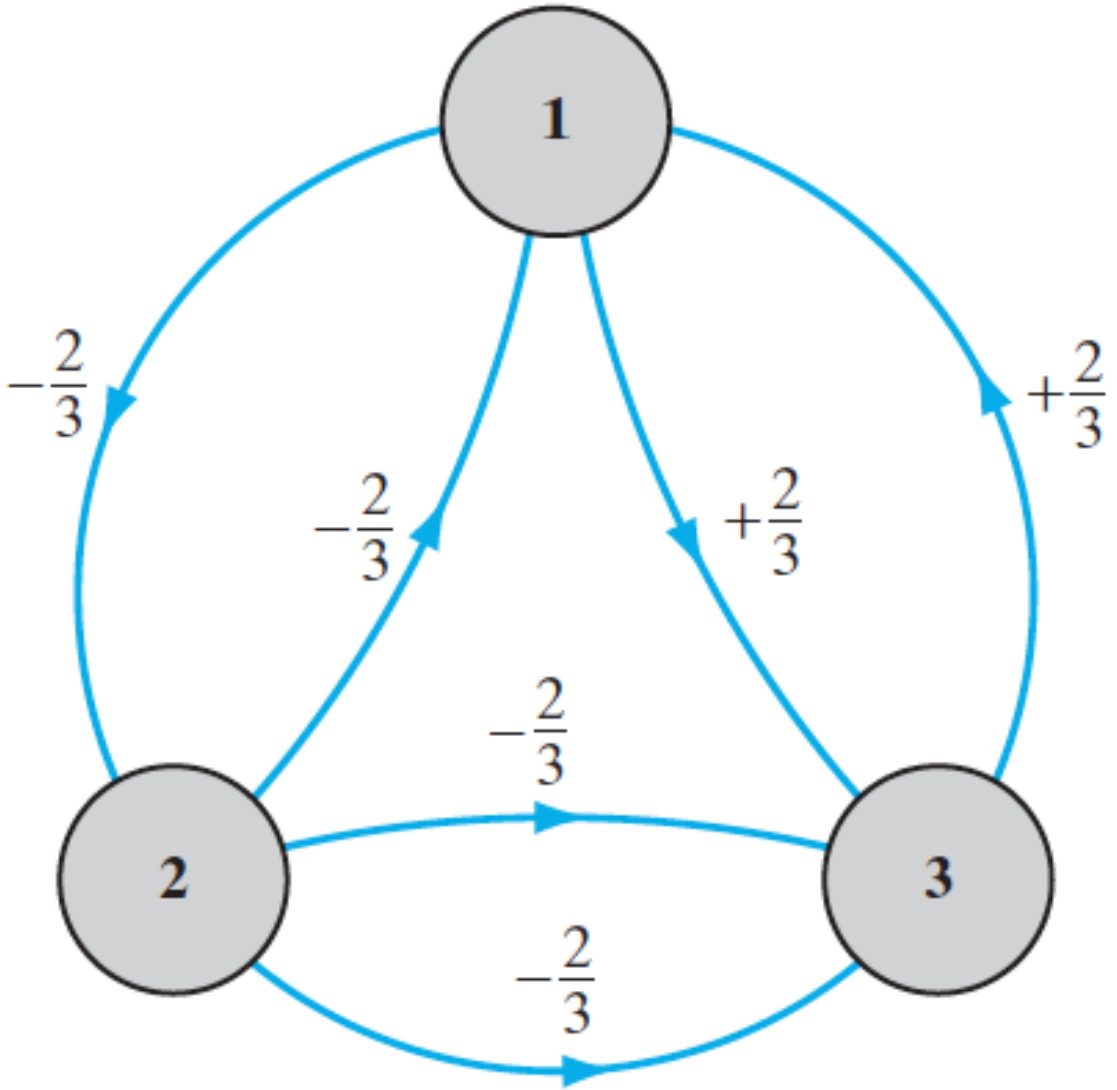
Repeat the iteration until the state vector $\mathbf{x}$ remains unchanged.

4. *Outputting.* Let $\mathbf{x}_{\text{fixed}}$ denote the fixed point (stable state) computed at the end of step 3. The resulting output vector $\mathbf{y}$ of the network is

$$\mathbf{y} = \mathbf{x}_{\text{fixed}}$$

Step 1 is the storage phase, and steps 2 through 4 constitute the retrieval phase.

Ejemplo Hopfield (ejemplo)

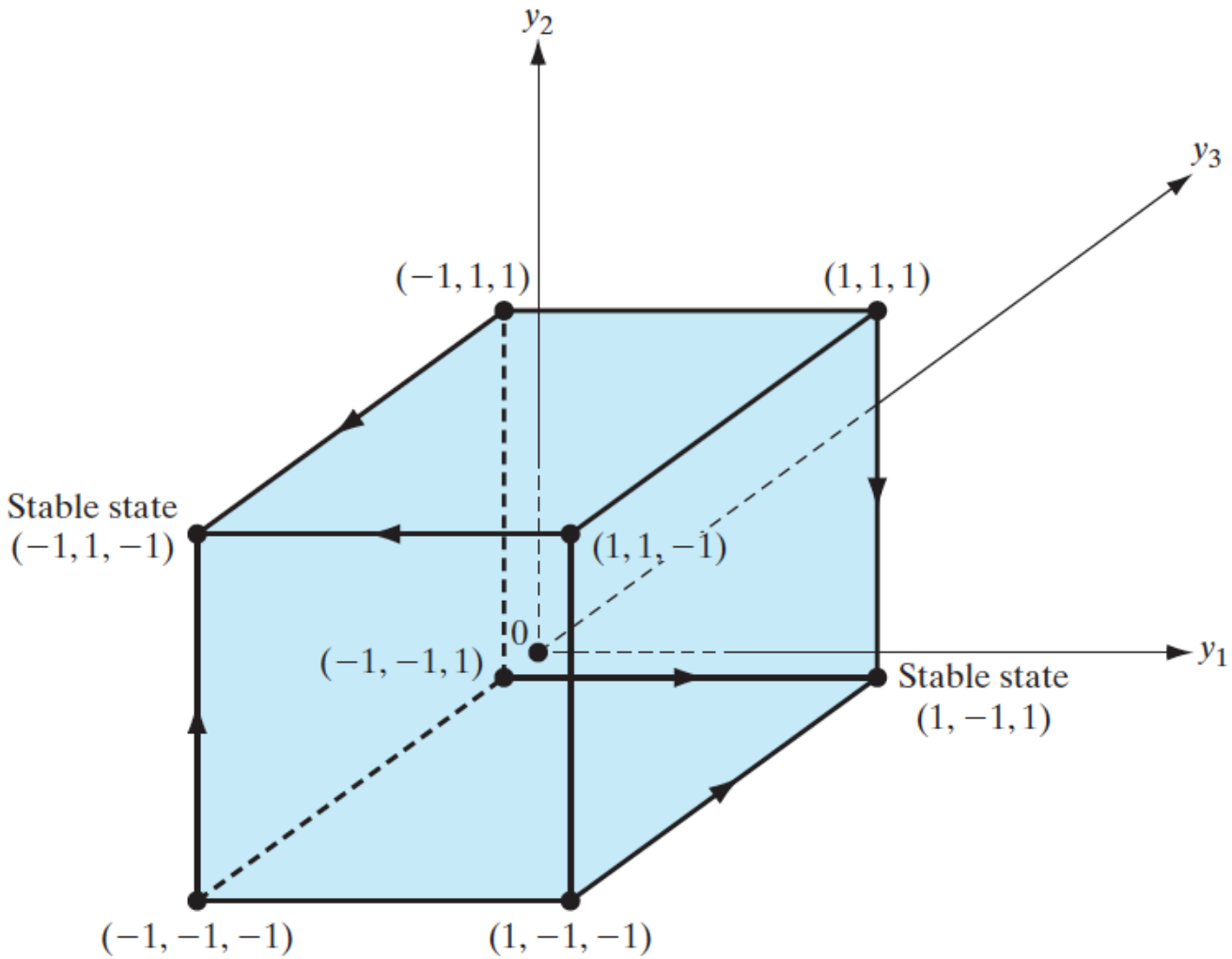


Ejemplo Hopfield (ejemplo)
Matriz de pesos W.

$$\mathbf{W} = \frac{1}{3}\begin{bmatrix} +1 \\ -1 \\ +1 \end{bmatrix}[+1, -1, +1] + \frac{1}{3}\begin{bmatrix} -1 \\ +1 \\ -1 \end{bmatrix}[-1, +1, -1] - \frac{2}{3}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \frac{1}{3}\begin{bmatrix} 0 & -2 & +2 \\ -2 & 0 & -2 \\ +2 & -2 & 0 \end{bmatrix}$$

Ejemplo Hopfield



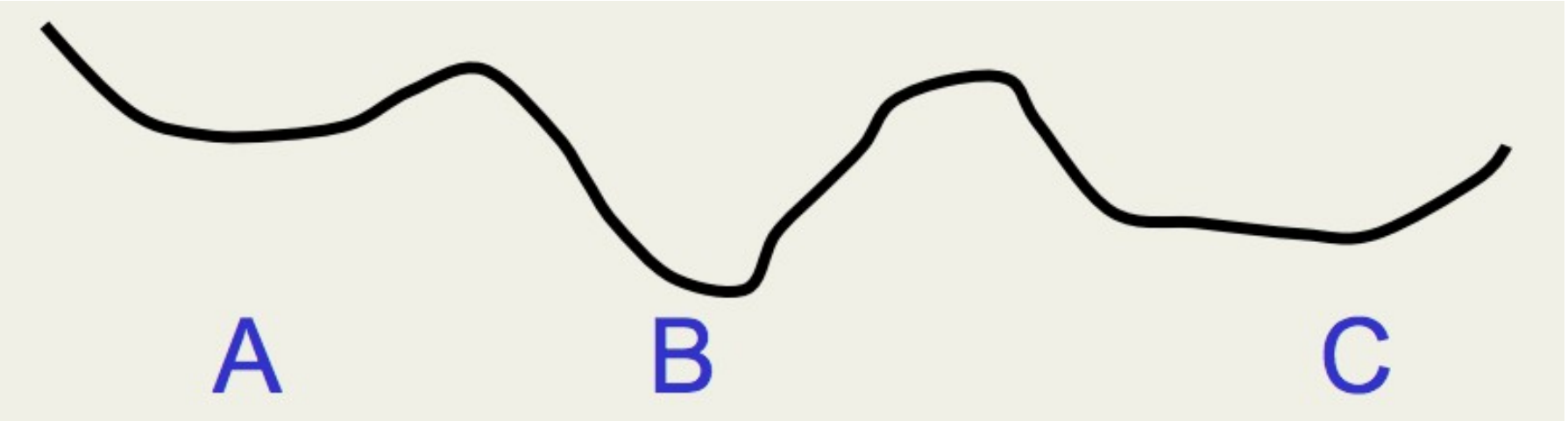
Sólo dos estados estables que cumplen sgn(Wy) = y

$2^3 = 8$ posibles inputs ¿a cuál estado estable converge cada uno?

Capacidad de una red de Hopfield

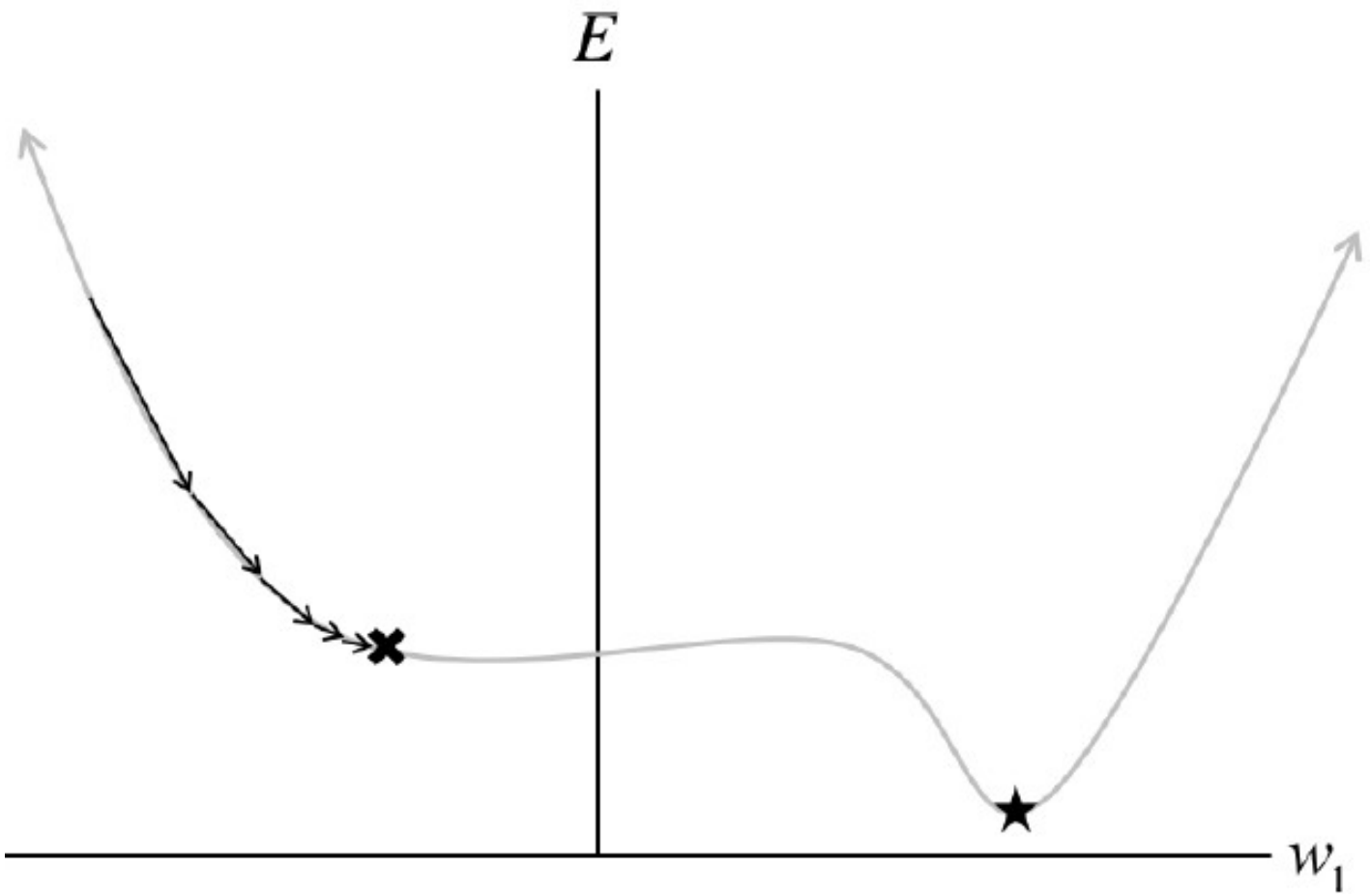
El número de memorias que puede almacenar una red de Hopfield depende del número de neuronas y conexiones (cada 1000 nodos podemos almacenar aproximadamente 138 memorias). Esta capacidad puede ser aumentada por otros métodos.

Energía de Mínimos espureos



Los mínimos espúreos tienen más energía que las memorias verdaderas.

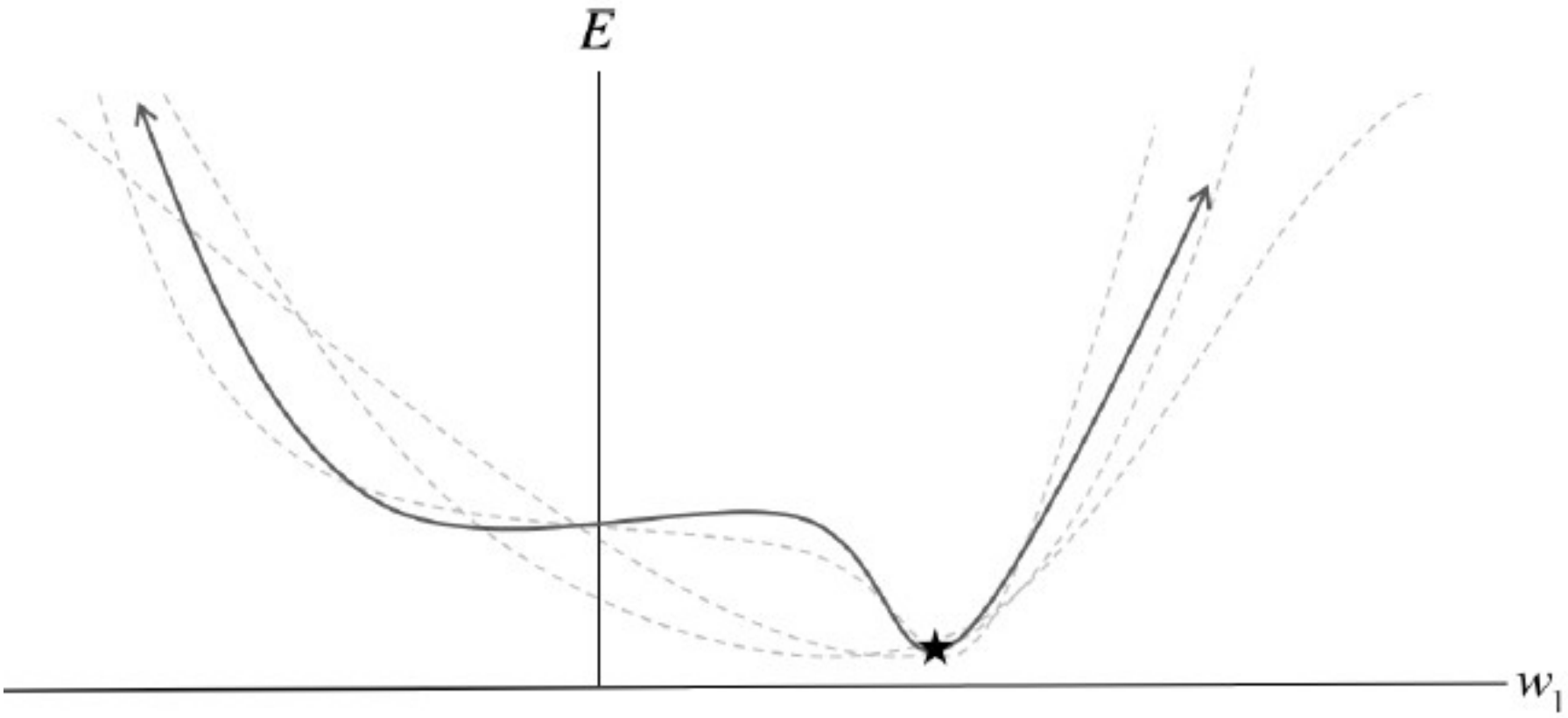
Volviendo al feedforward multicapa... ¿Cómo saltear llanuras de la energía?



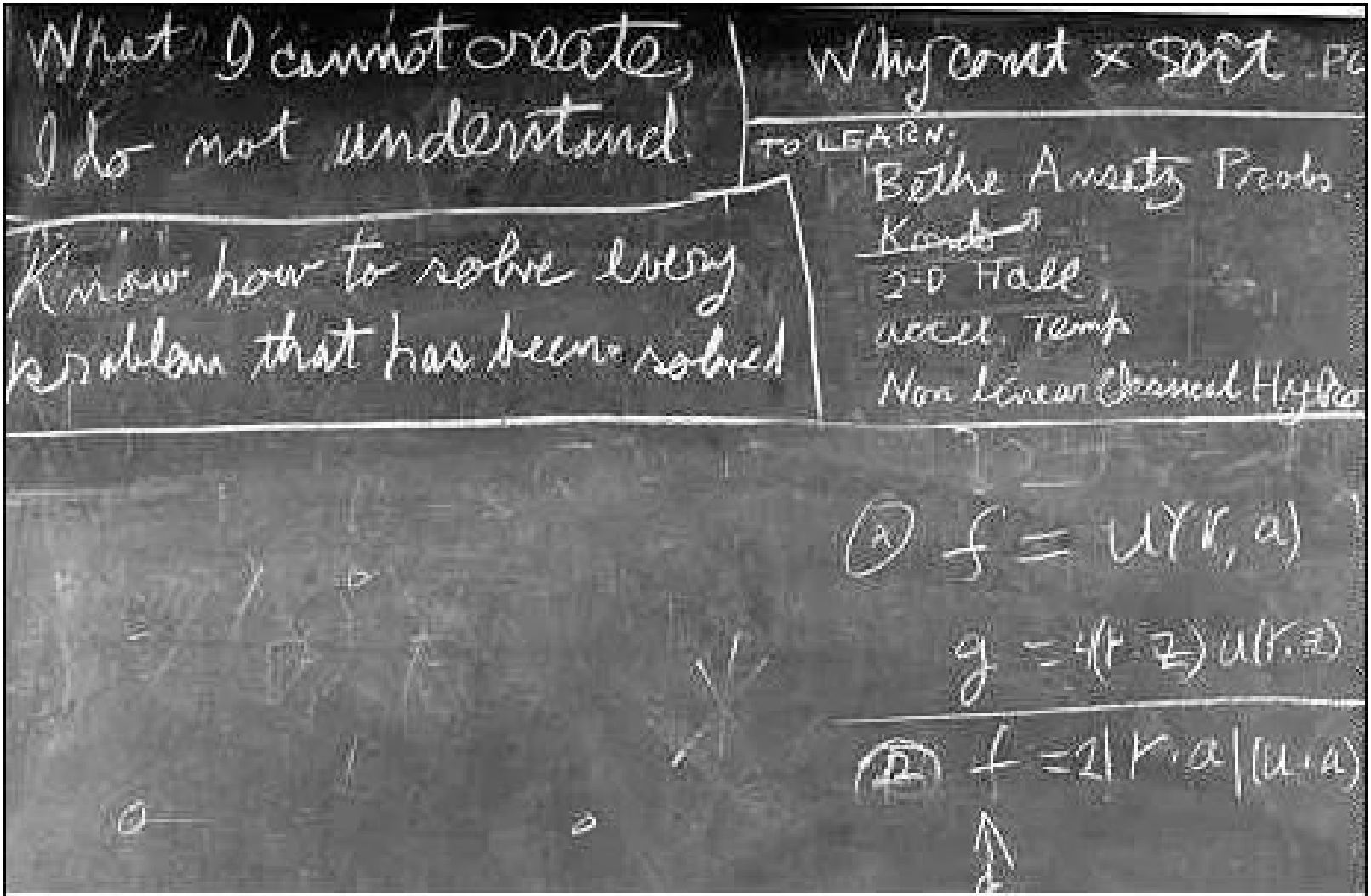
Entrenamiento estocástico, batch, y mini-batch

Se diferencian en el cálculo de la dirección de descenso. En el entrenamiento estocástico, la dirección de descenso del error se calcula en base al error del patrón actual y se actualizan los pesos. Esto puede hacer que disminuya el error del patrón actual, pero se incrementen los errores de los otros patrones. En el entrenamiento por batch, la dirección de descenso se calcula en base a los errores de todos los patrones del dataset. Como hay que pasar a todos los patrones del dataset, antes de actualizar los pesos, el entrenamiento por batch es el más lento. Una solución intermedia, es el entrenamiento por mini-batch, donde se usa un porcentaje del conjunto de entrenamiento para calcular el gradiente descendiente de la función de costo.

Energía en minibatch



Laboratorio: *What I cannot create, I do not understand.* Richard P. Feynman



Laboratorio

Misión 0: Tutorial de Python Numpy:

http://cs231n.github.io/python-numpy-tutorial

Misión 1a: Hopfield

Leer los ejercicios resueltos del Capítulo 21 del libro Neural Networks Design (Hagan et al.).

Misión 1b: Hopfield básico

- Programar una red de Hopfield, con las dos memorias del ejemplo, y ver si converge a las memorias.
- Resolver el ejercicio E21.11 del libro Neural Networks Design, Sólo el inciso i).
  Considerar lo aprendido en el ejercicio 1.2 de la Práctica 1 (OCR).

Misión 2: Hopfield ejemplo de Image Reconstruction

- https://github.com/crypto-code/Hopfield-Network
- Utilizarlo con un conjunto propio de imágenes y analizar diferencias.

Misión 3: Feedforward multicapa con backprop

Estudiar el código del multicapa por minibatch para entender el comportamiento. Modificarle los parámetros y estudiar el comportamiento.