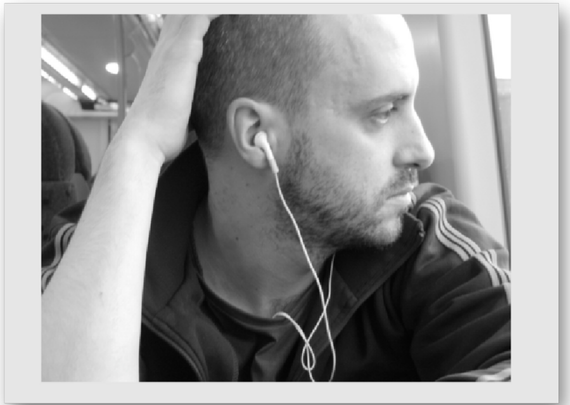
FÍSICA Y AI: EL PREMIO NOBEL DE HOPFIELD Y HINTON Y EL FUTURO TECNOLÓGICO

Isaac Pérez Castillo



UNIVERSIDAD AUTÓNOMA METROPOLITANA
Casa abierta al tiempo
Unidad Iztapalapa
BOURBAKI
COLEGIO DE MATEMÁTICAS

OUTLINE

- Introducción
- Emergencia
- Estados Multiples y vidrios de espín
- Redes Neuronales: diseñando atractores
- Máquinas de Boltzmann
- Cornucopia

Introducción

Ill. Niklas Elmehed © Nobel Prize Outreach

John J. Hopfield

Prize share: 1/2

Ill. Niklas Elmehed © Nobel Prize Outreach

Geoffrey Hinton

Prize share: 1/2

The Nobel Prize in Physics 2024 was awarded jointly to John J. Hopfield and Geoffrey E. Hinton "for foundational discoveries and inventions that enable machine learning with artificial neural networks"

The Nobel Prize in Chemistry 2024 was divided, one half awarded to David Baker "for computational protein design", the other half jointly to Demis Hassabis and John Jumper "for protein structure prediction"

Ill. Niklas Elmehed © Nobel Prize Outreach

David Baker

Prize share: 1/2

Ill. Niklas Elmehed © Nobel Prize Outreach

Demis Hassabis

Prize share: 1/4

Ill. Niklas Elmehed © Nobel Prize Outreach

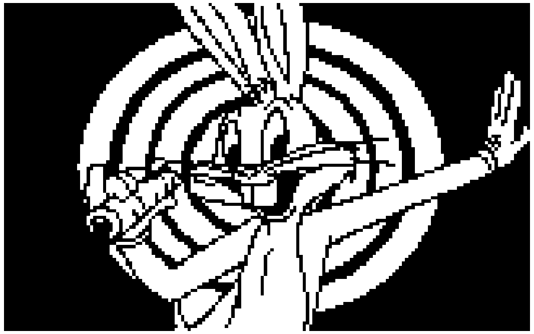
John Jumper

Prize share: 1/4

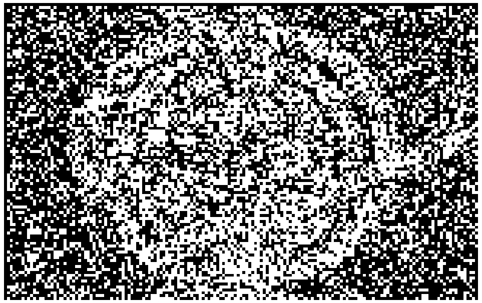
Introducción

Memoria asociativa

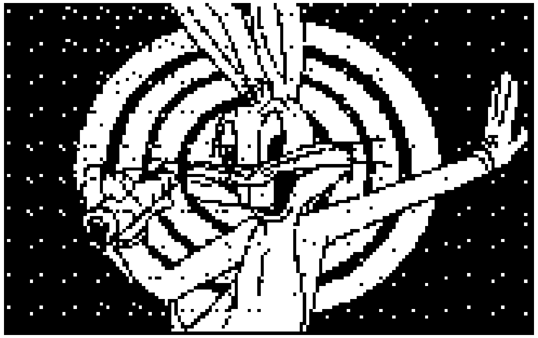
Original

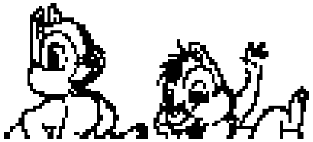


Degraded



Reconstruction



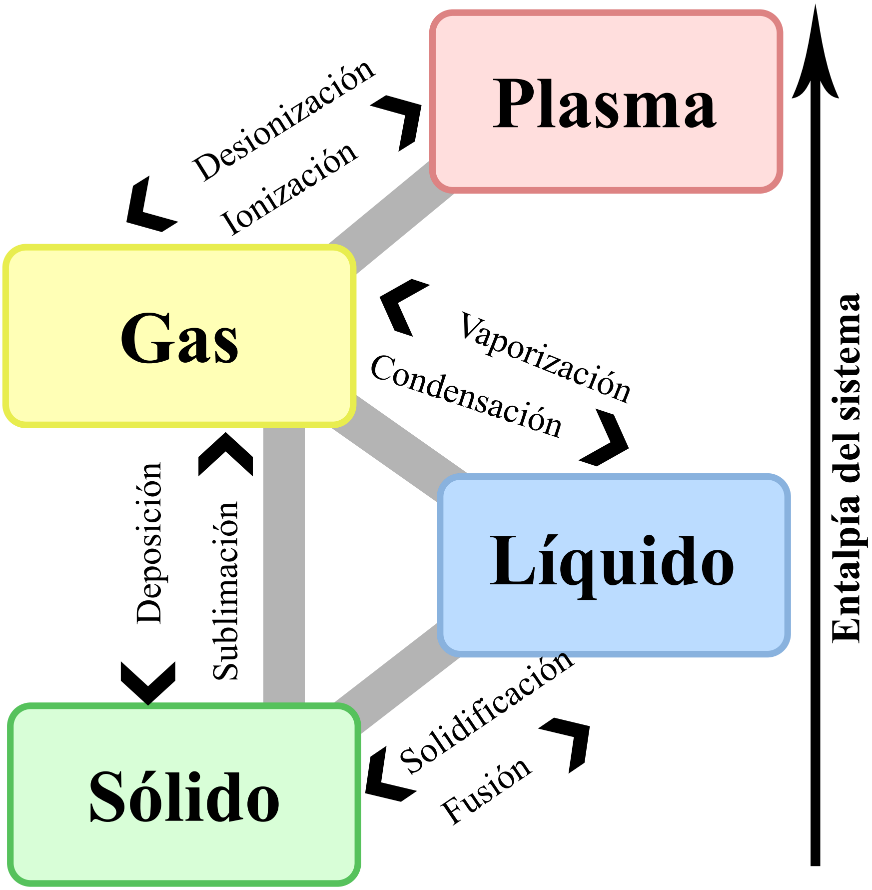


Patrones

estado neuronal inicial → evolución → **estado neuronal final**

Emergencia

Transiciones de fase



Descripción microscópica

$$E(\boldsymbol{s}) = \sum_{i,j} J_{ij} s_i s_j$$

s_j — Grado de libertad

J_{ij} — Interacciones

$$P(\boldsymbol{s}) = e^{-\beta E(\boldsymbol{s})} / Z(\beta)$$

Emergencia

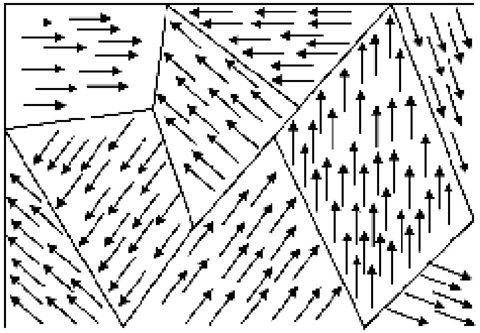
Descripción microscópica

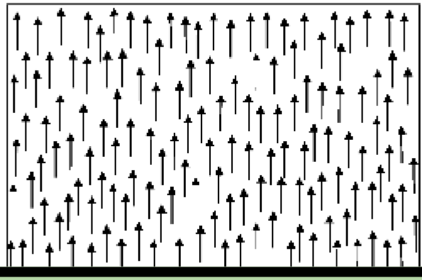
$$E(\boldsymbol{s}) = \sum_{i,j} J_{ij} s_i s_j$$

s_j **espín**

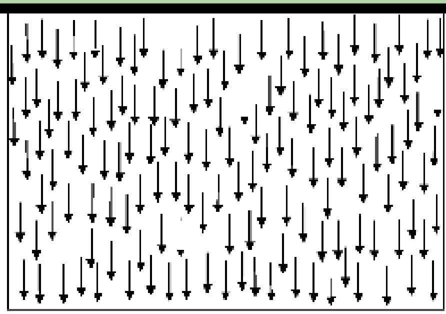
J_{ij} **Interacción magnética**

Ferromagnetismo





Un bit the información

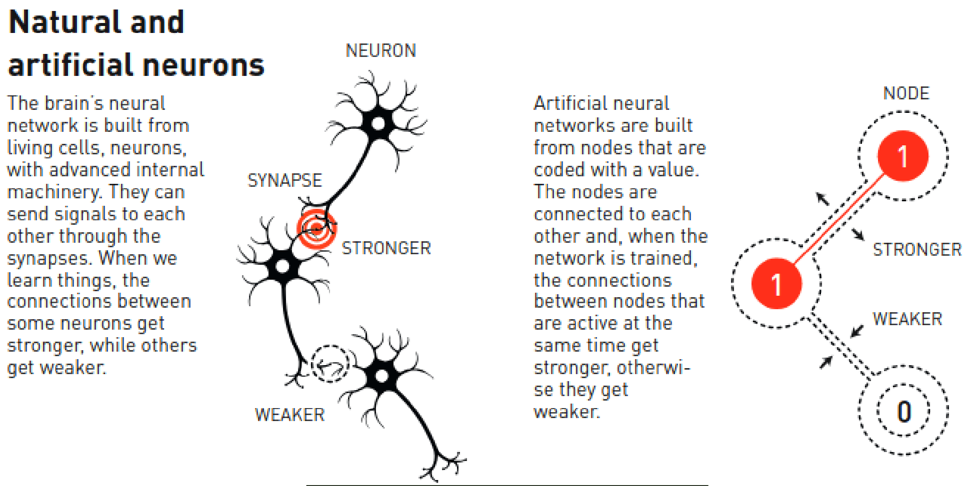


Ferromagnetismo

J_{ij} Interacción magnética

s_j espín

Un bit the información



Red Neuronal

¿Es posible tener interacciones J_{ij} de tal forma que guardamos mas de un bit de información?

J_{ij} Interacción entre neuronas

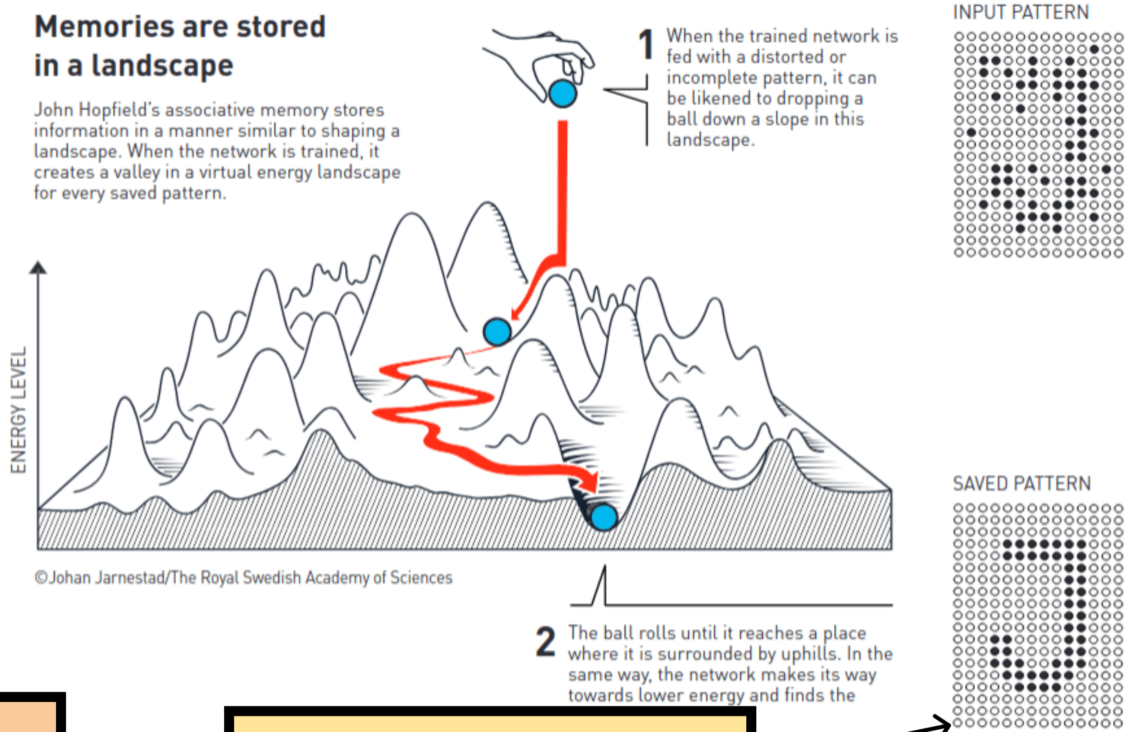
s_j estado neuronal

Redes Neuronales: Diseñando Atractores

$$J_{ij} = \sum_{\mu=1}^{p} \xi_i^{\mu} \xi_j^{\mu}$$

Regla de Hebb

$$\boldsymbol{\xi}^{\mu} = (\xi_1^{\mu}, \ldots \xi_N^{\mu})$$



$$E(\boldsymbol{s}) = \sum_{i,j} J_{ij} \boldsymbol{s}_i \boldsymbol{s}_j$$

Modelo de Hopfield

$$\boldsymbol{\xi}^{\mu} = (\xi_1^{\mu}, \ldots \xi_N^{\mu})$$

$$\boldsymbol{s}(t) \xrightarrow{\text{Evolución temporal}} \boldsymbol{\xi}^{\mu} = (\xi_1^{\mu}, \ldots \xi_N^{\mu})$$

Máquinas de Boltzmann

$$E(\boldsymbol{s}) = \sum_{i,j} J_{ij} s_i s_j + \sum_i \theta_i s_i$$

$$P(\boldsymbol{s}) = e^{-\beta E(\boldsymbol{s})} / Z(\beta)$$

Máquina de Boltzmann

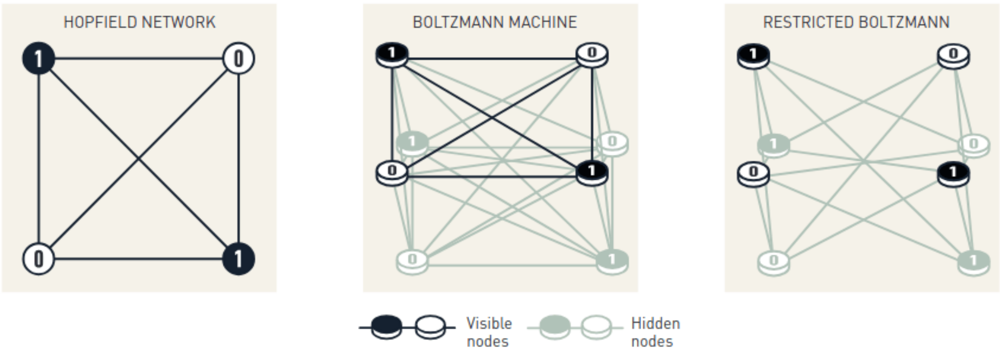
$$\boldsymbol{s} = (s_1, \ldots, s_N)$$

Neuronas ocultas $\boldsymbol{h}$

Neuronas visibles $\boldsymbol{v}$

$$P(\boldsymbol{v}) = \sum_{\boldsymbol{h}} P(\boldsymbol{v}, \boldsymbol{h})$$

Different types of network



John Hopfield's associative memory is built so that all the nodes are connected to each other. Information is fed in and read out from all the nodes.

Geoffrey Hinton's Boltzmann machine is often constructed in two layers, where information is fed in and read out using a layer of *visible* nodes. They are connected to *hidden* nodes, which affect how the network functions in its entirety.

In a restricted Boltzmann machine, there are no connections between nodes in the same layer. The machines are frequently used in a chain, one after the other. After training the first restricted Boltzmann machine, the content of the hidden nodes is used to train the next machine, and so on.

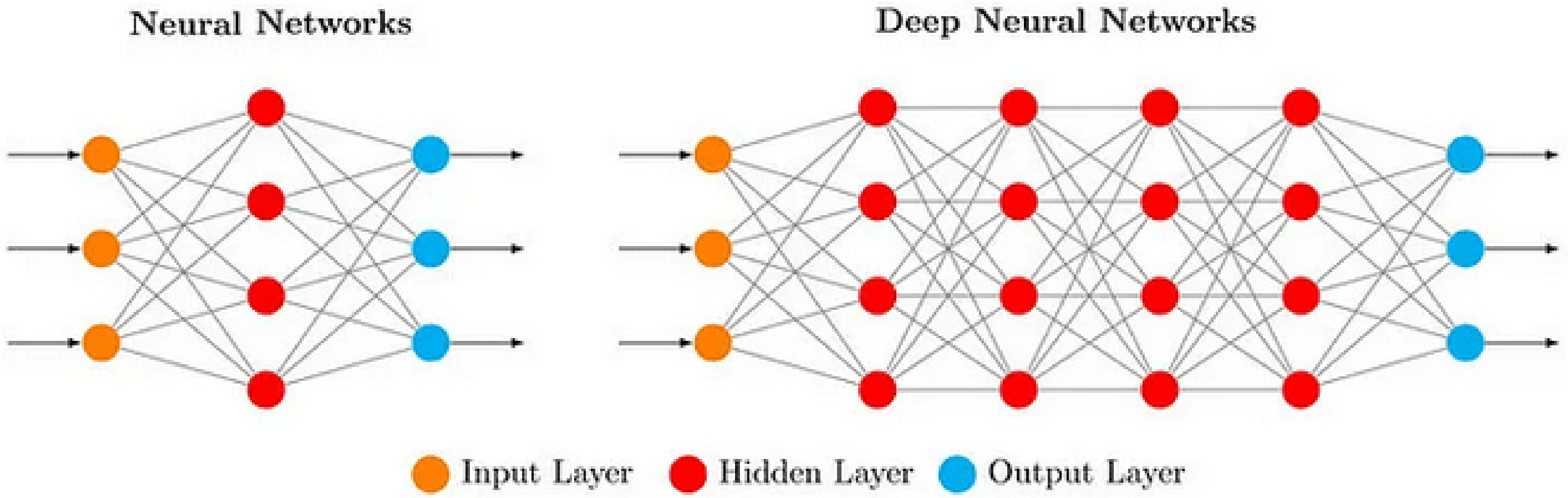
$$P_{\text{training}}(\boldsymbol{v})$$

Entrenamiento: encontrar las $\{J_{ij}, \theta_i\}$ tal que:

$$P(v) \to P_{\text{training}}(v)$$

$$J^\star_{ij}, \theta^\star_i = \arg\min_{\{J_{ij},\theta_i\}} \mathrm{KL}(P_{\text{training}}(\boldsymbol{v}) || P(\boldsymbol{v}))$$

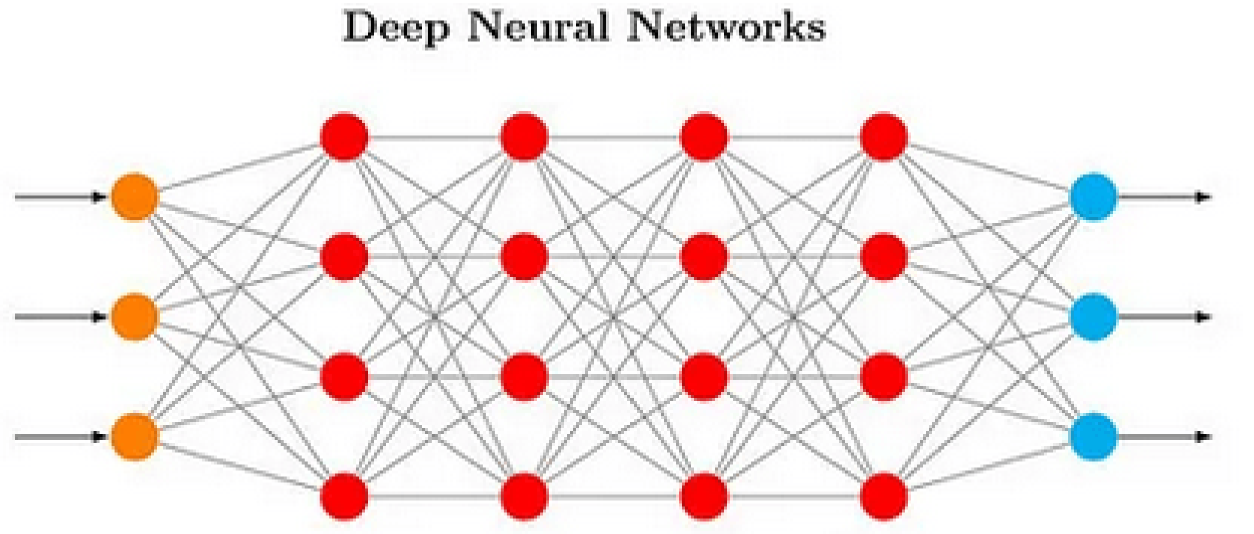
Deep NNs



- Reconocimiento de imágenes
- Procesamiento de lenguaje natural
- Síntesis de voz
- Sistemas de recomendación
- Aplicaciones en salud

- Aplicaciones en finanzas y trading
- Aplicaciones en comercio minorista
- Vehículos autónomos
- Monitoreo ambiental

Deep NNs: Backpropagation



x y

$$g(x) := f^L(W^L f^{L-1}(W^{L-1} \cdots f^1(W^1 x) \cdots))$$

Forward Propagation

Aprendizaje

Minimizar función coste:

$\mathcal{C}(\{W\})$

$$\boxed{W^{t+1} = W^t - \eta \nabla_W \mathcal{C}(\{W\})}$$

Regla de la cadena

Back Propagation

¡Gracias!