

Guifré de los Rios Blanch

Les Xarxes de Hopfield i el seu abast

Treball de Final de Grau

Dirigit pel
Dr.Alex Arenas Moreno

Grau en Enginyeria Matemàtica i Física



UNIVERSITAT ROVIRA I VIRGILI

Tarragona, 2025

Índex

Resum	vi
Resumen	vi
Abstract	vi
1 Introducció	1
2 <i>Mathematical Background</i>	2
2.1 Àlgebra Lineal i Representació Vectorial	2
2.2 Anàlisi Matemàtica i Sistemes Dinàmics	2
2.3 Mecànica Estadística i Energia Minimitzant	3
2.4 Probabilitat i Estadística	3
2.5 Optimització i Algorísmia	3
2.6 Teoria de la Informació	4
3 Hopfield Networks Clàssiques	5
3.1 <i>Machine Learning</i> i la Neuron Artificial	5
3.2 Conceptes bàsics: Què és una memòria associativa? Història breu . . .	6
3.3 Xarxes de Hopfield	6
3.4 Estructura i Dinàmica de la Xarxa	7
3.5 Funció d'Energia	8
3.6 Algorisme d'Emmagatzematge i Recuperació de Patrons	8
3.7 Capacitat d'Emmagatzematge	8
3.8 Dinàmica de recuperació: Síncrona i Asíncrona	9
3.8.1 Actualització síncrona	9
3.8.2 Actualització asíncrona	9
3.9 Problemes i Limitacions de les Xarxes Hopfield Clàssiques	10
4 Test de recuperació amb soroll amb la regla de Hebb (dígit 8x8)	11
4.1 Test amb 20% de soroll	11
4.2 Test amb 10% de soroll	14
5 Test de recuperació amb soroll amb la pseudoinversa (dígit 8x8)	16
5.1 La pseudoinversa: minimització algebraica de l'energia	16
5.2 Test de recuperació amb la pseudoinversa	17
5.3 Comparativa de resultats entre actualització síncrona i asíncrona	22
5.4 Límit de tolerància al soroll amb la pseudoinversa amb activació síncrona	22
6 Test de recuperació amb dígit 64x64	25
6.1 Recuperació amb la regla de Hebb	25
6.2 Recuperació amb la pseudoinversa i activació síncrona	27
6.3 Recuperació amb la pseudoinversa i activació asíncrona	29
7 Test de recuperació amb dígit amb millor definició	33
7.1 Recuperació amb la regla de Hebb	33
7.2 Recuperació amb la pseudoinversa	34

8	Test de recuperació amb lletres en alta definició	36
8.1	Recuperació amb la regla de Hebb	37
8.2	Recuperació amb la pseudoinversa	38
9	Test de recuperació amb patrons diferenciats	41
9.1	Recuperació amb la regla de Hebb	41
9.2	Recuperació amb la pseudoinversa	42
10	Xarxes de Hopfield amb Energia Exponencial	45
10.1	Nova funció d'energia	45
10.2	Regla d'actualització	45
10.3	Avantatges	46
11	Test de recuperació amb l'energia exponencial	47
11.1	Recuperació amb dígits del 0 al 9 amb alta definició	47
11.2	Recuperació amb patrons ben diferenciats	49
12	Test de recuperació amb la meitat del patró en negre	52
12.1	Recuperació amb la pseudoinversa	52
12.2	Recuperació amb l'energia exponencial	53
13	Conclusions	56
	Apèndix A Recuperacions de patrons	59
	Apèndix B Codis	94
B.1	Dígits del 0 al 9 (8x8)	94
B.2	Dígits del 0 al 9 (64x64)	97
B.3	Lletres de la A a la Z	99
B.4	Patrons ben diferenciats	100
B.5	Recuperació amb l'energia exponencial	103

Índex d'exemples de codi

Índex de figures

1	dígits del 0 al 9 de mida 8x8	12
2	Recuperació amb la regla de Hebb amb un 20% de soroll	12
3	Recuperació amb la regla de Hebb amb un 10% de soroll	14
4	Recuperació amb la pseudoinversa i actualització síncrona, soroll del 20%.	18
5	Recuperació amb la pseudoinversa i actualització asíncrona, soroll del 20%.	19
6	Recuperació amb la pseudoinversa i actualització asíncrona amb més iteracions.	20
7	Recuperació amb la pseudoinversa i actualització asíncrona amb més iteracions. (2)	21
8	Recuperació amb pseudoinversa i activació síncrona amb 30% de soroll.	23
9	Recuperació amb pseudoinversa i activació síncrona amb 40% de soroll.	24
10	Recuperació amb la regla de Hebb amb un 20% de soroll	26
11	Recuperació amb la pseudoinversa i activació síncrona amb un 20% de soroll	27
12	Recuperació amb pseudoinversa i activació síncrona amb 40% de soroll.	28
13	Recuperació amb la pseudoinversa i activació asíncrona amb un 20% de soroll	29
14	Recuperació amb la pseudoinversa i actualització asíncrona amb més iteracions.	30
15	Recuperació amb la pseudoinversa i actualització asíncrona amb més iteracions. (2)	31
16	Dígits del 0 al 9 amb millor definició.	33
17	Recuperació amb regla de Hebb (dígits definits)	34
18	Recuperació amb pseudoinversa (dígits definits)	35
19	Lletres de la A a la Z en alta definició	36
20	Recuperació de patrons de lletres amb la regla de Hebb.	37
21	Recuperació de patrons de lletres amb la pseudoinversa.	39
22	Conjunt de 16 patrons binaris de mida 64×64 ben diferenciats	41
23	Recuperació de patrons diferenciats amb la regla de Hebb.	42
24	Recuperació de patrons diferenciats amb la pseudoinversa.	43
25	Recuperació dels dígits del 0 al 9 amb alta definició amb l'energia exponencial i 20% de soroll.	48
26	Recuperació dels dígits del 0 al 9 amb alta definició amb l'energia exponencial i 40% de soroll.	49
27	Recuperació de patrons ben diferenciats amb l'energia exponencial i 20% de soroll	50
28	Recuperació de patrons ben diferenciats amb l'energia exponencial i 40% de soroll.	51
29	Recuperació de patrons ben diferenciats amb la pseudoinversa amb la meitat esquerra negra.	53
30	Recuperació de patrons ben diferenciats amb l'energia exponencial amb la meitat esquerra negra.	54
31	Recuperació amb la regla de Hebb amb un 20% de soroll	59

32	Recuperació amb la regla de Hebb amb un 10% de soroll	60
33	Recuperació amb la pseudoinversa i actualització síncrona, soroll del 20%.	61
34	Recuperació amb la pseudoinversa i actualització asíncrona, soroll del 20%.	62
35	Recuperació amb la pseudoinversa i actualització asíncrona amb més iteracions.	63
36	Recuperació amb la pseudoinversa i actualització asíncrona amb més iteracions. (2)	64
37	Recuperació amb pseudoinversa i activació síncrona amb 30% de soroll.	65
38	Recuperació amb pseudoinversa i activació síncrona amb 40% de soroll.	66
39	Recuperació amb la regla de Hebb amb un 20% de soroll	67
40	Recuperació amb la pseudoinversa i activació síncrona amb un 20% de soroll	68
41	Recuperació amb pseudoinversa i activació síncrona amb 40% de soroll.	69
42	Recuperació amb la pseudoinversa i activació asíncrona amb un 20% de soroll	70
43	Recuperació amb la pseudoinversa i actualització asíncrona amb més iteracions.	71
44	Recuperació amb la pseudoinversa i actualització asíncrona amb més iteracions. (2)	72
45	Recuperació amb regla de Hebb (dígit definit)	73
46	Recuperació amb pseudoinversa (dígit definit)	74
47	Recuperació de patrons de lletres amb la regla de Hebb. (1)	75
48	Recuperació de patrons de lletres amb la regla de Hebb. (2)	76
49	Recuperació de patrons de lletres amb la regla de Hebb. (3)	77
50	Recuperació de patrons de lletres amb la pseudoinversa. (1)	78
51	Recuperació de patrons de lletres amb la pseudoinversa. (2)	79
52	Recuperació de patrons de lletres amb la pseudoinversa. (3)	80
53	Recuperació de patrons diferenciats amb la regla de Hebb. (1)	81
54	Recuperació de patrons diferenciats amb la regla de Hebb. (2)	82
55	Recuperació de patrons diferenciats amb la pseudoinversa. (1)	83
56	Recuperació de patrons diferenciats amb la pseudoinversa. (2)	84
57	Recuperació dels dígit del 0 al 9 amb alta definició amb l'energia exponencial i 20% de soroll.	85
58	Recuperació dels dígit del 0 al 9 amb alta definició amb l'energia exponencial i 40% de soroll.	86
59	Recuperació de patrons ben diferenciats amb l'energia exponencial i 20% de soroll. (1)	87
60	Recuperació de patrons ben diferenciats amb l'energia exponencial i 20% de soroll. (2)	88
61	Recuperació de patrons ben diferenciats amb l'energia exponencial i 40% de soroll. (1)	89
62	Recuperació de patrons ben diferenciats amb l'energia exponencial i 40% de soroll. (2)	90
63	Recuperació de patrons ben diferenciats amb la pseudoinversa amb la meitat esquerra negra. (1)	91
64	Recuperació de patrons ben diferenciats amb la pseudoinversa amb la meitat esquerra negra. (2)	92

65	Recuperació de patrons ben diferenciats amb l'energia exponencial amb la meitat esquerra negra. (1)	93
66	Recuperació de patrons ben diferenciats amb l'energia exponencial amb la meitat esquerra negra. (2)	94

Índex de taules

1	Resultats de la recuperació dels patrons amb un 20% de soroll.	13
2	Similitud i energia per patró amb actualització síncrona i asíncrona (pseudoinversa, 20% soroll).	22
3	Comparativa de la similitud en la recuperació dels 16 patrons parcialment ocults, utilitzant la pseudoinversa i l'energia exponencial.	55

Resum

Aquest treball analitza en profunditat les xarxes de Hopfield, tant en la seva formulació clàssica com en les extensions modernes basades en energies alternatives. A partir d'una sèrie d'experiments amb patrons binaris de diverses resolucions (8×8 i 64×64), s'estudia la capacitat de recuperació dels patrons emmagatzemats i l'impacte de l'actualització síncrona o asíncrona, així com la influència del soroll i la similitud entre patrons. Es comparen tres estratègies d'aprenentatge: la regla de Hebb, la pseudoinversa i la formulació energètica exponencial. Els resultats mostren les limitacions estructurals del model de Hebb, fins i tot amb patrons no correlacionats, i confirmen la robustesa de la pseudoinversa i de les xarxes modernes de Hopfield. El treball conclou amb una reflexió sobre les diferents formulacions d'energia i com condicionen la dinàmica de recuperació.

Resumen

Este trabajo analiza en profundidad las redes de Hopfield, tanto en su formulación clásica como en las extensiones modernas basadas en energías alternativas. A partir de una serie de experimentos con patrones binarios de distintas resoluciones (8×8 y 64×64), se estudia la capacidad de recuperación de los patrones almacenados y el impacto de la actualización síncrona o asíncrona, así como la influencia del ruido y la similitud entre patrones. Se comparan tres estrategias de aprendizaje: la regla de Hebb, la pseudoinversa y la formulación energética exponencial. Los resultados muestran las limitaciones estructurales del modelo de Hebb, incluso con patrones no correlacionados, y confirman la robustez de la pseudoinversa y de las redes modernas de Hopfield. El trabajo concluye con una reflexión sobre las distintas funciones de energía y su impacto en la dinámica de recuperación.

Abstract

This thesis presents an in-depth analysis of Hopfield networks, addressing both the classical formulation and modern extensions based on alternative energy functions. Through a series of experiments with binary patterns of varying resolutions (8×8 and 64×64), it explores the networks' ability to recover stored patterns, the effects of synchronous or asynchronous updates, and the role of noise and pattern similarity. Three learning strategies are compared: Hebb's rule, the pseudoinverse method, and an exponential energy formulation. The results highlight the structural limitations of Hebb's model—even with uncorrelated patterns—and confirm the robustness of the pseudoinverse and modern Hopfield networks. The thesis concludes with a discussion on how different energy functions influence the recovery dynamics.

1 Introducció

Durant els darrers anys del grau en Enginyeria Matemàtica i Física, he pogut adquirir una formació rigorosa i interdisciplinària que combina coneixements matemàtics avançats amb una comprensió profunda dels fonaments de la física. Aquest entorn formatiu m'ha permès desenvolupar habilitats en la modelització, l'anàlisi i la resolució de problemes complexos, tant des d'un punt de vista teòric com computacional. L'enfocament dual de la titulació m'ha proporcionat les eines necessàries per entendre el comportament de sistemes des de les seves lleis fonamentals fins a la seva representació abstracta i algorítmica.

El treball que es presenta a continuació té com a objectiu posar en pràctica aquest bagatge acadèmic a través d'un projecte que connecta diverses branques de la carrera: des de les matemàtiques aplicades fins a la física estadística, passant per la computació científica i la intel·ligència artificial. Concretament, s'ha estudiat i implementat un conjunt de models de memòria associativa basats en xarxes de Hopfield —tant en la seva formulació clàssica com moderna—, amb l'objectiu de comparar la seva capacitat de recuperació de patrons binaris en diferents escenaris de soroll i arquitectura.

El projecte s'ha estructurat en diferents fases: en primer lloc, s'ha desenvolupat un marc teòric sòlid que inclou conceptes d'energia estadística, optimització, teoria de la informació i dinàmica de sistemes. Posteriorment, s'han implementat xarxes de Hopfield clàssiques utilitzant tant la regla de Hebb com la regla de la pseudoinversa, i s'ha analitzat la seva capacitat d'emmagatzematge i la seva robustesa davant patrons similars. També s'ha comparat el comportament de la recuperació síncrona i asíncrona, observant fenòmens d'atractors espuris i convergència incorrecta en determinades condicions.

Una part substancial del treball ha estat dedicada a l'experimentació amb patrons binaris de resolució 64x64, incloent-hi díigits i lletres majúscules. S'ha fet un estudi empíric exhaustiu de la qualitat de recuperació en funció del soroll injectat i s'han generat taules comparatives de similitud, mostrant clarament les diferències entre els models. Finalment, s'ha explorat la dinàmica de les xarxes Hopfield modernes mitjançant funcions d'energia alternatives, implementant-les i analitzant-ne l'evolució energètica, estabilitat i qualitat de les solucions trobades.

Aquest projecte, doncs, no només ha permès aprofundir en l'estudi d'una classe de models amb gran rellevància històrica i teòrica, sinó que també ha servit per integrar coneixements transversals del grau en un entorn de recerca aplicada. Les conclusions obtingudes posen de manifest tant les possibilitats com les limitacions de cada enfocament, i aporten una visió comparativa útil per a futures línies d'investigació en el camp de les xarxes neuronals i la memòria artificial.

2 *Mathematical Background*

El desenvolupament de models de xarxes neuronals inspirats en el comportament del cervell humà requereix una base matemàtica sòlida i transversal. En aquest apartat, s'exposen les eines teòriques fonamentals que s'utilitzaran en l'estudi i implementació de les Hopfield Networks, tant en la seva formulació clàssica com moderna. Aquestes eines provenen de diverses àrees del grau: **àlgebra lineal, anàlisi matemàtica, probabilitat i estadística, optimització, teoria de sistemes dinàmics i teoria de la informació.**

2.1 Àlgebra Lineal i Representació Vectorial

Les Hopfield Networks modelen patrons com vectors en un espai de dimensió n , i les connexions entre neurones com una matriu de pesos simètrica $W \in R^{n \times n}$. Els conceptes clau que s'utilitzen són:

- **Producte escalar:** $\langle x, y \rangle = \sum_{i=1}^n x_i y_i$, que mesura la similitud entre dos vectors d'estat.
- **Matrius simètriques:** $W = W^T$, garantint l'existència de valors propis reals i que l'energia decreixi.
- **Descomposició espectral:** $W = Q\Lambda Q^T$, on Q és ortogonal i Λ diagonal.
- **Projeccions en subespais:** Usades per recuperar components dominants.

Els patrons es representen com vectors binaris $x \in \{-1, +1\}^n$, tot i que en versions modernes poden ser reals i normalitzats, $x \in R^n$, amb $|x| = 1$.

2.2 Anàlisi Matemàtica i Sistemes Dinàmics

La dinàmica d'una Hopfield Network es pot escriure com:

$$x_i(t+1) = \operatorname{sgn} \left(\sum_j W_{ij} x_j(t) - \theta_i \right)$$

on θ_i és el llindar de la neurona i .

Conceptes clau:

- **Convergència:** La seqüència $x(t)$ tendeix a un punt fix x^* tal que $x(t+1) = x(t)$.
- **Funció de Lyapunov (energia):** $E(x) = -\frac{1}{2} \sum_{i,j} W_{ij} x_i x_j + \sum_i \theta_i x_i$
- **Estabilitat:** Si x^* és punt fix i $E(x)$ no decreix en el seu entorn, és estable.

Això connecta amb assignatures com Equacions Diferencials i Dinàmica de Sistemes.

2.3 Mecànica Estadística i Energia Minimitzant

La funció d'energia d'una Hopfield Network és anàloga a l'Hamiltoniana d'un sistema d'Ising:

$$E(x) = -\frac{1}{2} \sum_{i,j} W_{ij} x_i x_j + \sum_i \theta_i x_i$$

En aquest sistema, l'estat d'equilibri correspon a un mínim local de l'energia. Aquesta analogia permet interpretar l'evolució de la xarxa com una relaxació tèrmica vers un estat d'energia mínima.

En versions estocàstiques, s'introdueix una temperatura T i la probabilitat d'estar en un estat x es dona per:

$$P(x) = \frac{1}{Z} e^{-E(x)/(k_B T)}$$

on k_B és la constant de Boltzmann. Això permet simular transicions de fase i millorar l'exploració de l'espai d'estats.

2.4 Probabilitat i Estadística

Les xarxes Hopfield modernes poden incorporar soroll o funcions d'activació probabilístiques:

- **Funció sigmoide:** $\sigma(u) = \frac{1}{1+e^{-\beta u}}$
- **Esperança:** $E[x_i] = \sigma\left(\sum_j W_{ij} x_j - \theta_i\right)$
- **Distància de Hamming:** $d(x, y) = \sum_i |x_i - y_i|$, per mesurar l'error de recuperació.
- **Capacitat d'emmagatzematge:** $p_{max} \approx 0.138n$ patrons per n neurones.

Coneixements d'Estadística i Probabilitat s'utilitzen per analitzar la robustesa i precisió del model.

2.5 Optimització i Algorísmia

L'actualització de la xarxa pot entendre's com un problema d'optimització:

- **Descens de gradient:** En xarxes modernes, la pèrdua L es minimitza mitjançant $\nabla_W L$.
- **Regularització:** Penalitzacions com $\lambda|W|^2$ per evitar sobreajustament.
- **Funcions d'activació suaus:** tanh, ReLU, etc., per fer diferencials les actualitzacions.

Aquests coneixements provenen d'assignatures com Optimització Numèrica i Algorísmia.

2.6 Teoria de la Informació

L'anàlisi del rendiment d'una Hopfield Network es pot fer amb:

- **Entropia:** $H(X) = -\sum_x P(x) \log P(x)$
- **Informació mútua:** $I(X;Y) = H(X) - H(X|Y)$, quantifica la capacitat de recuperació.
- **Codificació eficient:** Mètodes per optimitzar la representació dels patrons.

Aquestes eines permeten quantificar el flux d'informació i la qualitat de la memòria.

Aquest marc teòric proporcionarà les bases per a les seccions següents del treball, on es descriurà en detall el funcionament de les Hopfield Networks clàssiques (apartat 3) i modernes (apartat 4), i posteriorment es presentarà el desenvolupament i entrenament del model (apartats 5 i 6).

3 Hopfield Networks Clàssiques

Les *Hopfield Networks* van ser proposades per John Hopfield el 1982 com una forma de xarxa neuronal recurrent dissenyada per resoldre problemes de memòria associativa. Aquest tipus de xarxa és una xarxa de neurones unitàries en què la informació es representa mitjançant patrons d'activació que poden ser emmagatzemats i recuperats de manera eficient, tot aprofitant la seva dinàmica d'evolució energètica. Abans però, es farà una breu introducció al *Machine Learning* i a conceptes relacionats.

3.1 *Machine Learning* i la Neurona Artificial

El *machine learning* és una branca de la intel·ligència artificial que es basa en la capacitat dels sistemes informàtics per aprendre a partir de dades, sense ser explícitament programats per a cada tasca específica. L'objectiu és construir models capaços de reconèixer patrons, fer prediccions i prendre decisions en base a l'experiència prèvia. Dins d'aquest àmbit, les xarxes neuronals artificials constitueixen una estructura fonamental per a l'aprenentatge automàtic.

En el context de les xarxes neuronals artificials, una **neurona** és una unitat fonamental de processament que rep un conjunt d'entrades numèriques (anomenades inputs), les pondera mitjançant pesos sinàptics, i produeix una sortida (output) en funció del resultat d'aquesta combinació. Aquesta sortida pot ser binària o contínua, depenent del tipus de funció d'activació utilitzada.

Aquest model s'inspira en el funcionament de les neurones biològiques del cervell humà. En una neurona natural, els impulsos elèctrics arriben a través de les dendrites, es combinen al cos cel·lular, i si la diferència de potencial (és a dir, la suma de totes aquestes senyals) supera un cert llindar (*threshold*), la neurona es dispara enviant un impuls a través de l'axó cap a altres neurones.

De manera anàloga, en una neurona artificial, es calcula una suma ponderada de les entrades x_1, x_2, \dots, x_n amb els seus pesos associats w_1, w_2, \dots, w_n , i es compara el resultat amb un llindar θ . Si aquesta suma supera el llindar, la neurona s'activa i genera una sortida (per exemple, +1), i en cas contrari, roman inactiva (1 o 0). Matemàticament, aquest procés es pot expressar com:

$$y = \begin{cases} 1 & \text{si } \sum_{i=1}^n w_i x_i \geq \theta, \\ -1 & \text{altrament.} \end{cases}$$

Aquest comportament binari és característic de les primeres xarxes neuronals, com les de tipus perceptró o les xarxes de Hopfield, tot i que en models més moderns s'utilitzen funcions d'activació més suaus com la sigmoide o la ReLU, on es tindrà

$$y = f \left(\sum_{i=1}^n w_i x_i + b \right)$$

on b és un terme de biaix i f és una funció com la sigmoide, la tangents hiperbòlica o la ReLU, entre d'altres.

En el cas de les Hopfield Networks clàssiques, la funció d'activació és usualment

una funció signe, i les neurones s'actualitzen de manera asíncrona o síncrona.

Aquest model de neurona permet que la xarxa aprengui patrons mitjançant ajustos dels pesos sinàptics, seguint principis d'aprenentatge supervisat, no supervisat o per reforç. En aquest treball, ens centrarem en les xarxes de tipus Hopfield, que utilitzen un model particularment senzill de neurona, però amb propietats molt interessants pel que fa a memòria i estabilitat.

3.2 Conceptes bàsics: Què és una memòria associativa? Història breu

Les Hopfield Networks pertanyen a una categoria de models coneguts com a **memòries associatives**, també anomenades *content-addressable memories* (CAM). Aquest tipus de memòria no accedeix a la informació per una adreça específica, com en els ordinadors clàssics, sinó que la recupera a partir del contingut parcial o distorsionat d'un patró prèviament emmagatzemat.

En una memòria associativa, l'objectiu és trobar el patró més proper (segons alguna mètrica com la distància de Hamming) a l'entrada donada. Aquest mecanisme és altament inspirat en la forma en què es creu que el cervell humà recupera informació: a partir de fragments incomplets d'experiències prèvies, és capaç d'evocar records complets.

Exemple: si una memòria associativa ha après el patró $x = (1, -1, 1, -1)$, i rep com a entrada $x' = (1, -1, 1, +1)$, pot reconèixer i corregir aquest error, retornant el patró original.

Història breu

La idea de memòries associatives té les seves arrels a la dècada dels 1950, quan es van començar a explorar models computacionals inspirats en la biologia. No obstant això, fou John Hopfield qui, el 1982, va proposar un model matemàtic formal que unificava aquestes idees sota un marc energètic i dinàmic ben definit.

Aquest model, conegut com a **Hopfield Network**, va permetre una formulació clara de les condicions de convergència, estabilitat i capacitat d'emmagatzematge. Es tracta d'una xarxa recurrent i totalment connectada, on cada neurona es connecta amb totes les altres (excepte amb si mateixa), i on l'estat global del sistema evoluciona per minimitzar una funció d'energia.

El model de Hopfield es va convertir en una eina fonamental per estudiar la memòria associativa des del punt de vista de la física estadística, l'optimització i la neurociència teòrica. Tot i les seves limitacions inicials (capacitat reduïda, dificultats amb patrons correlacionats), ha estat la base per a nombroses extensions i reformulacions modernes, especialment gràcies als avenços en *deep learning*.

3.3 Xarxes de Hopfield

El model més senzill de memòria associativa es pot construir a partir de la suma dels productes exteriors dels N patrons $\{\mathbf{x}_i\}_{i=1}^N$ que es volen emmagatzemar. Aquesta regla d'aprenentatge es coneix com a regla de Hebb (Hebbian learning rule), la qual es basa en la idea que "les neurones que s'activen juntes, es connecten juntes". És una formulació simple però fonamental dins del camp de les xarxes neuronals.

En el cas de les xarxes de Hopfield clàssiques, els patrons \mathbf{x}_i que es volen guardar són vectors binaris polars, és a dir, cadascun dels seus components pren valors en

$$\mathbf{x}_i \in \{-1, 1\}^d,$$

on d representa la longitud (o dimensió) dels patrons.

La matriu de pesos W , que defineix les connexions sinàptiques entre les neurones de la xarxa, es construeix de la manera següent:

$$W = \sum_{i=1}^N \mathbf{x}_i \mathbf{x}_i^T. \quad (1)$$

Aquesta matriu W conté la informació associada als patrons emmagatzemats. Un cop la xarxa ha estat entrenada amb aquesta regla, és possible recuperar un patró a partir d'un estat inicial ξ , que pot ser una versió parcial, distorsionada o incompleta d'un dels patrons originals. La dinàmica de la xarxa portarà aquest estat cap al patró més proper emmagatzemat, actuant així com un sistema de memòria associativa.

Aquest mètode permet una implementació senzilla i efectiva d'aprenentatge no supervisat, tot i que presenta certes limitacions que seran analitzades en apartats posteriors, com ara la capacitat màxima d'emmagatzematge, la presència d'atractors espuris i la sensibilitat al soroll.

3.4 Estructura i Dinàmica de la Xarxa

Una Hopfield Network està formada per un conjunt de neurones, on cadascuna d'elles es connecta amb totes les altres. La xarxa és recurrent, ja que les neurones s'actualitzen a través de les seves pròpies connexions, i el sistema evoluciona fins arribar a un estat estable (punt fix), que es correspon amb un patró emmagatzemat.

$$x_i(t+1) = \operatorname{sgn} \left(\sum_j W_{ij} x_j(t) - \theta_i \right)$$

On:

- $x_i(t)$ és l'estat de la neurona i en el temps t .
- W_{ij} és el pes de la connexió entre les neurones i i j .
- θ_i és el llindar de la neurona i .
- La funció $\operatorname{sgn}(u)$ és la funció sign, que retorna $+1$ si $u > 0$ i -1 si $u < 0$.

La xarxa és un sistema dinàmic on l'evolució de l'estat de les neurones depèn dels seus valors anteriors i de les connexions amb les altres neurones. Les neurones es van actualitzant iterativament fins que s'arriba a un estat estable, anomenat punt fix, on el sistema ja no canvia més.

3.5 Funció d'Energia

Una de les característiques més importants de les Hopfield Networks és la seva funció d'energia, que es defineix com:

$$E(x) = -\frac{1}{2} \sum_{i,j} W_{ij} x_i x_j + \sum_i \theta_i x_i$$

Aquesta funció serveix com una mena de “potencial” per la xarxa, i l'objectiu de la dinàmica de les neurones és minimitzar aquesta energia. Quan les neurones arriben a un punt fix, l'energia de la xarxa ha assolit un mínim local.

El comportament d'una Hopfield Network és anàlog al de sistemes físics que tendeixen a un estat de mínima energia, com els sistemes magnètics descrits per l'Hamiltoniana del model d'Ising.

3.6 Algorisme d'Emmagatzematge i Recuperació de Patrons

Una de les aplicacions fonamentals de les Hopfield Networks és l'emmagatzematge i la recuperació de patrons. El model original de Hopfield emmagatzema patrons a través d'una regla d'actualització basada en els pesos sinàptics entre neurones, que es defineixen com:

$$W_{ij} = \frac{1}{n} \sum_{\mu=1}^p \xi_i^{(\mu)} \xi_j^{(\mu)}$$

On:

- $\xi_i^{(\mu)}$ és el valor de l'estat de la neurona i per al patró μ .
- p és el nombre de patrons emmagatzemats.
- n és el nombre de neurones.

Els patrons $\xi^{(\mu)}$ són vectors binaris, on cada entrada de cada vector pot ser $+1$ o -1 . Aquest algorisme crea una matriu de pesos que permet recuperar els patrons originals quan la xarxa és activada amb un patró parcial o corrupte.

3.7 Capacitat d'Emmagatzematge

La capacitat d'emmagatzematge d'una Hopfield Network és limitada, i depèn del nombre de neurones n . La capacitat teòrica màxima de la xarxa és aproximadament $0.138n$ patrons. Això vol dir que, en general, una xarxa amb n neurones pot emmagatzemar i recuperar fins a $0.138n$ patrons diferents de manera efectiva.

Aquesta limitació es deu a l'alta interconnexió de la xarxa i la dependència dels pesos per emmagatzemar múltiples patrons de manera eficaç. Si el nombre de patrons emmagatzemats supera aquest límit, la xarxa pot començar a perdre la capacitat de recuperar correctament els patrons.

3.8 Dinàmica de recuperació: Síncrona i Asíncrona

Un cop entrenada la xarxa de Hopfield, el procés de recuperació consisteix a aplicar una dinàmica iterativa sobre un estat inicial (normalment un patró amb soroll), amb l'objectiu que la xarxa convergeixi cap a un mínim d'energia que correspongui amb un dels patrons memoritzats. Aquesta actualització pot dur-se a terme de dues maneres fonamentals: de forma síncrona o asíncrona. A continuació s'estudien les dues aproximacions, tant des del punt de vista teòric com experimental.

3.8.1 Actualització síncrona

En la recuperació síncrona, totes les neurones de la xarxa s'actualitzen al mateix temps a cada iteració. Formalment, donat un estat actual $x^{(t)}$, el nou estat $x^{(t+1)}$ es calcula com:

$$x^{(t+1)} = \text{sign}(Wx^{(t)} - \theta)$$

on W és la matriu de pesos sinàptics, i θ és un vector llindar (opcional). Aquesta actualització s'aplica de manera simultània a totes les neurones, i el procés es repeteix fins que s'assoleix un punt fix (és a dir, $x^{(t+1)} = x^{(t)}$).

Aquest mètode és computacionalment eficient, ja que permet aprofitar paral·lelisme, però presenta una limitació important: la naturalesa síncrona pot provocar oscil·lacions o convergència a mínims espuris. En escenaris on els patrons entrenats són molt similars entre si o tenen formes molt dominants, la xarxa pot convergir sistemàticament cap a un patró incorrecte.

A les proves realitzades, s'ha observat que la recuperació síncrona tendeix a portar gairebé tots els patrons corromputs cap a una mateixa forma (en aquest cas, el dígit 3), fet que indica la presència d'un atractor dominant dins l'espai de fases de la xarxa.

3.8.2 Actualització asíncrona

L'actualització asíncrona consisteix a actualitzar una sola neurona per iteració, seleccionada aleatòriament, i deixar la resta inalterada. Això s'ajusta millor al funcionament de xarxes neuronals reals i minimitza la possibilitat d'oscil·lacions. El procés iteratiu és el següent:

1. Escollir una neurona i a l'atzar.
2. Calcular la seva nova activació: $x_i = \text{sign}(h_i)$, amb $h_i = \sum_j W_{ij}x_j - \theta_i$.
3. Repetir fins a convergència (cap neurona canvia d'estat).

Conclusió

La tria entre actualització síncrona o asíncrona pot influir de manera decisiva en el rendiment d'una xarxa de Hopfield. Si bé la primera és més ràpida, també és més susceptible a col·lapsar cap a mínims incorrectes. L'actualització asíncrona, en canvi, ofereix una dinàmica més estable i fidel al comportament desitjat, tot i requerir més iteracions.

Per tant, en aplicacions on l'estabilitat i la precisió són prioritàries, es recomana utilitzar la recuperació asíncrona, especialment quan s'entrenen patrons similars o es treballa amb conjunts de dades densos.

3.9 Problemes i Limitacions de les Xarxes Hopfield Clàssiques

Tot i que les Hopfield Networks clàssiques tenen una gran aplicació en memòria associativa, tenen algunes limitacions importants:

- **Capacitat d'emmagatzematge limitada:** Com hem esmentat abans, la capacitat d'emmagatzematge de patrons és limitada, el que pot restringir la seva utilització en problemes més complexos.
- **Problemes de convergència:** Les xarxes clàssiques poden tenir problemes de convergència en casos on els pesos no són adequats o quan es tracten patrons molt similars.
- **Sensibilitat a la inicialització:** Els punts fixos als quals convergiran les xarxes depenen de l'estat inicial, per la qual cosa la xarxa pot no convergir al patró desitjat si la configuració inicial és inapropiada.
- **Correlació entre patrons:** Si els patrons emmagatzemats són massa semblants entre ells (altament correlacionats), interfereixen en el procés de recuperació i poden aparèixer punts fixos incorrectes. Això limita la capacitat efectiva en aplicacions reals, on els patrons solen estar estructurats i no aleatoris.
- **Precisió en la recuperació:** És important destacar que la recuperació dels patrons no és perfecta. Una hipòtesi comuna és que la capacitat d'emmagatzematge limitada n'és la causa principal. No obstant això, estudis recents (Amit et al., Torres et al.) mostren que la capacitat no és l'únic factor limitant. En particular, s'ha derivat una fórmula per a la capacitat de recuperació de patrons sense error:

$$C \approx \frac{d^2}{\log(d)},$$

on d és la dimensió de l'entrada. Aquesta expressió indica que, si bé la capacitat creix amb d^2 , la recuperació pot seguir sent imperfecta degut a altres factors com l'estructura de l'energia, la presència de mínims locals o la interferència entre patrons.

Aquestes limitacions han portat al desenvolupament de versions més modernes de les Hopfield Networks, que utilitzen tècniques d'aprenentatge més sofisticades per superar aquestes barreres.

4 Test de recuperació amb soroll amb la regla de Hebb (dígits 8x8)

Per posar a prova la capacitat de la xarxa de Hopfield entrenada amb la regla hebbiana, hem realitzat una sèrie d'experiments emprant els deu dígits decimals, és a dir, del 0 al 9. Cada dígit ha estat representat mitjançant una matriu binària de mida 8×8 , on el valor 1 indica un píxel negre (part activa del patró) i el valor 0 un píxel blanc (fons). Aquestes matrius han estat codificades com vectors de 64 components, que han servit com a patrons d'entrenament de la xarxa. Es segueix el següent procediment:

- Es defineixen els 10 patrons corresponents als dígits del 0 al 9 en format binari 8×8 , convertint els valors de 0 a -1 per tal d'ajustar-se al model de Hopfield (que treballa amb estats $\{-1, +1\}$).
- A continuació, es calcula la matriu de pesos W .
- Per a cada patró original, es genera una versió corrupta injectant un percentatge determinat de soroll, invertint l'estat d'un nombre aleatori de píxels.
- Es realitza el procés de recuperació iterativa a partir del patró corruptat, actualitzant els valors de les neurones segons la regla de Hopfield fins a assolir un estat estable.
- Finalment, es calcula la similitud entre el patró recuperat i l'original, així com l'energia final de la xarxa

Per assegurar la comparabilitat entre diferents tècniques i resultats mostrats, s'ha utilitzat una *seed* predeterminada en la generació de soroll. Això permet avaluar el rendiment de cada mètode sota les mateixes condicions exactes.

4.1 Test amb 20% de soroll

Amb l'objectiu d'avaluar la robustesa del sistema, s'ha introduït un nivell de soroll exacte del 20% a cadascun dels patrons, és a dir, s'ha modificat aleatòriament el signe d'un 20% dels píxels del patró original. Aquest tipus de soroll permet estudiar el comportament de la xarxa davant distorsions moderades però significatives, simulant situacions en què la informació d'entrada no sigui precisa o estigui parcialment deteriorada.

A través d'aquesta prova, es busca observar fins a quin punt la xarxa és capaç de recuperar correctament els patrons memoritzats a partir d'una versió corrupta, analitzar la qualitat de la reconstrucció i estudiar les possibles confusions entre patrons similars. És especialment interessant identificar si alguns dígits tenen més tendència a ser confosos que d'altres, i si existeixen patrons atractors cap als quals convergeixen múltiples entrades sorolloses.

Les representacions binàries dels dígits utilitzats, en la seva versió original sense soroll, són les següents:

L'experiment es realitza tant en mode d'actualització síncrona com asíncrona, i els resultats són pràcticament idèntics en ambdós casos tots els patrons tendeixen a un

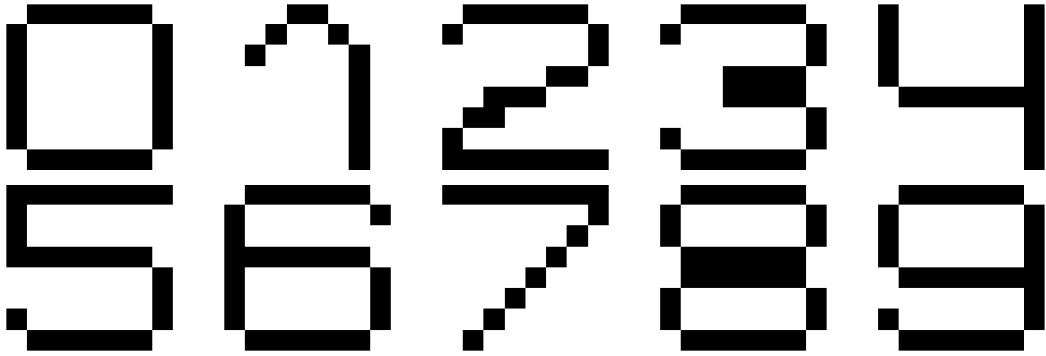


Figura 1: dígits del 0 al 9 de mida 8x8

mateix estat d'atracció, molt similar visualment al dígit "3", però que no coincideix exactament amb el patró original del número 3. Aquest comportament esdevé simptomàtic d'un fenomen característic de les xarxes de Hopfield: la presència de mínims locals espuris dins l'espai d'energia.

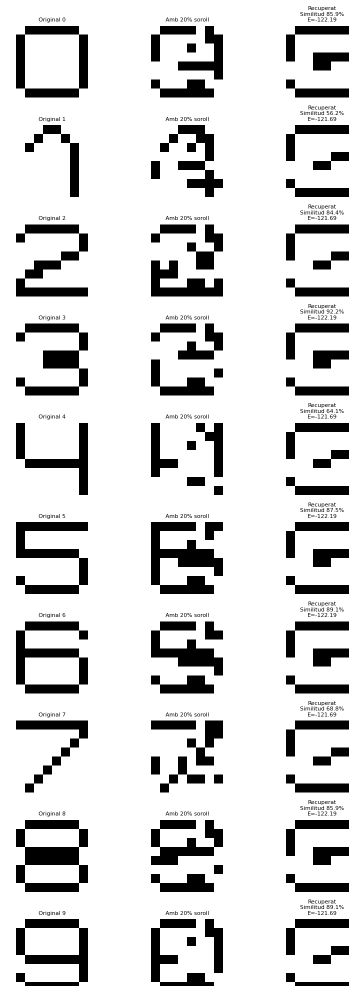


Figura 2: Recuperació amb la regla de Hebb amb un 20% de soroll

Nota: Totes les taules de recuperació es poden consultar ampliades a l'Apèndix, on es pot apreciar millor el seu contingut i les dades que inclouen.

Número	Similitud (%)	Energia final
0	89.1	-121.69
1	53.1	-122.19
2	84.4	-121.69
3	92.2	-122.19
4	64.1	-121.69
5	84.4	-121.69
6	89.1	-122.19
7	68.8	-121.69
8	85.9	-122.19
9	89.1	-121.69

Taula 1: Resultats de la recuperació dels patrons amb un 20% de soroll.

Els resultats del test amb un 20% de soroll mostren que, malgrat que en molts casos la sortida final presenta una percentatge de semblança alt amb el patró original, la xarxa tendeix a convergir de manera reiterada cap a un estat concret —un patró similar al dígit 3— independentment del dígit d’entrada. Aquest comportament és especialment evident en patrons com el 1, el 4 o el 7, on la similitud amb el patró original cau per sota del 70%, arribant al 53.1% en el cas del número 1. Aquestes baixes similituds es poden atribuir tant a la simplicitat o verticalitat d’alguns dígits com al fet que el sistema acaba atrapant-se en un atractor espuri.

Un fet destacat és la presència de només dues energies finals possibles: $E = -121.69$ i $E = -122.19$, que es repeteixen de manera gairebé alterna en totes les execucions. Aquest fenomen indica que la xarxa no està accedint als 10 mínims corresponents als patrons memoritzats, sinó que s’està limitant a només dos mínims locals. Aquest “fals 3” actua com a atractor espuri: un estat estable que no correspon exactament a cap dels patrons entrenats, però que emergeix com una combinació lineal de diversos dígits amb estructures similars (com ara el 3, 5, 6, 8 i 9), especialment a les regions superiors i centrals del patró. Aquestes coincidències estructurals fan que aquestes zones tinguin pesos reforçats a la matriu W , provocant que l’espai d’energia contingui un mínim dominant en aquesta configuració.

Aquest comportament es pot explicar des del punt de vista de la capacitat teòrica de la xarxa de Hopfield. Amb $n = 64$ neurones, la capacitat màxima estimada és $p \approx 0.15 \cdot n \approx 9.6$ patrons. En aquest experiment s’han entrenat 10 patrons, la qual cosa situa la xarxa just al límit de la seva capacitat. Aquest excés condueix a la solapació d’atractors i a la degeneració d’energia, on múltiples patrons acaben col·lapsant en una mateixa vall energètica.

A més, aquesta convergència cap a un atractor espuri no depèn del tipus d’actualització utilitzat (sigui síncrona o asíncrona), cosa que demostra que el problema no es troba en el procés de relaxació sinó en la pròpia estructura de la matriu de pesos generada per la regla de Hebb. Això posa de manifest una limitació estructural d’aquest mètode: no garanteix que els únics estats estables siguin els patrons entrenats, i pot generar estats fixos no desitjats que compleixen la condició de mínim d’energia, però no tenen cap valor funcional en la tasca de memòria associativa.

4.2 Test amb 10% de soroll

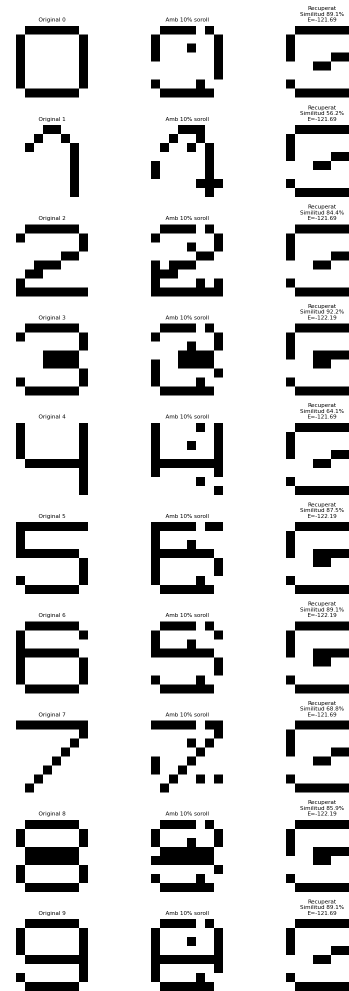


Figura 3: Recuperació amb la regla de Hebb amb un 10% de soroll

Per comprovar si la convergència cap a patrons espuris es devia a un nivell de soroll excessiu, s'ha repetit l'experiment reduint la distorsió al 10%. No obstant això, els resultats obtinguts mostren exactament el mateix comportament observat anteriorment amb un 20% de soroll: la xarxa continua convergint cap a només dos mínims locals d'energia ($E = -121.69$ i $E = -122.19$), i la sortida final tendeix a coincidir amb una configuració espúria molt propera al dígit 3.

Aquest fenomen confirma que el problema no prové del soroll injectat sinó que està intrínsecament relacionat amb la capacitat de la xarxa. Com ja s'ha comentat, amb $n = 64$ neurones, la capacitat teòrica se situa entorn de $p \approx 9.6$ patrons. En aquest cas s'estan memoritzant 10 patrons, fet que força la xarxa a una situació límit. Això comporta que múltiples patrons col·lapsin sobre el mateix atractor, minimitzant l'energia en configuracions intermèdies.

Aquest resultat reforça la conclusió que la xarxa, quan funciona prop del seu límit de capacitat, no conserva la identitat individual dels patrons, i mostra comportaments espuris que no milloren ni tan sols quan el soroll és moderat.

Aquesta situació motiva la recerca de models alternatius amb una codificació més

4 Test de recuperació amb soroll amb la regla de Hebb (digits 8x8)

precisa i controlada dels patrons. En el capítol següent es presentarà una millora basada en la pseudoinversa, una tècnica que permet establir pesos que garanteixen una energia mínima exacta per als patrons entrenats, reduint així l'aparició d'atractors espuris.

5 Test de recuperació amb soroll amb la pseudoinversa (dígits 8x8)

5.1 La pseudoinversa: minimització algebraica de l'energia

En una xarxa de Hopfield, l'objectiu principal és emmagatzemar un conjunt de patrons com a mínims d'energia del sistema. És a dir, es vol construir una matriu de connexions W tal que, quan un patró o una versió alterada d'aquest patró s'introdueix a la xarxa, el sistema evolucioni cap al patró original de manera estable. Aquesta estabilitat es garanteix si el patró és un atractor de la dinàmica, és a dir, un punt fix d'energia mínima dins de l'espai de representació.

Tradicionalment, per construir la matriu de pesos W , s'ha utilitzat la regla de Hebb, una tècnica senzilla i local que suma les correlacions exteriors de cada patró:

$$W = \sum_{i=1}^p x_i x_i^T$$

on x_i és el patró i -èssim expressat en format bipolar (valors -1 i $+1$), i p és el nombre total de patrons. Aquesta fórmula permet emmagatzemar patrons de manera distribuïda, però té una limitació fonamental: només funciona bé quan els patrons són poc nombrosos i suficientment ortogonals entre ells.

Quan els patrons són molt similars entre si, apareix un fenomen conegut com a interferència creuada (cross-talk), en què les contribucions dels diferents patrons es superposen i distorsionen els atractors del sistema. Això fa que el sistema, en lloc de convergir al patró original a partir d'una versió amb soroll, es desplaci cap a un mínim local d'energia incorrecte. Aquest error és particularment freqüent quan el nombre de patrons s'apropa o supera el límit teòric de capacitat de la xarxa (al voltant de $0.14N$, on N és la dimensió dels vectors), o quan hi ha estructures redundants i similars entre els patrons.

Una altra limitació de la regla de Hebb és que la minimització de l'energia no és explícita ni garantida globalment. Durant el procés de recuperació, la xarxa fa una seqüència d'actualitzacions per reduir l'energia $E(x) = -\frac{1}{2}x^T W x$, però no hi ha cap garantia que aquest procés arribi a un mínim global associat al patró correcte. De fet, sovint la dinàmica pot quedar atrapada en mínims espuris, estats que tenen baixa energia però que no corresponen a cap patró entrenat.

Per solucionar aquestes limitacions, s'introdueix la regla de la pseudoinversa com a alternativa algebraicament exacta a la regla de Hebb. Aquesta tècnica parteix del desig que cada patró x_i sigui un punt fix exacte del sistema, és a dir, que compleixi:

$$W x_i = x_i$$

per a tots els patrons x_i . Això és equivalent a voler que cada patró sigui un autovector de la matriu W amb valor propi igual a 1. Per resoldre aquest sistema de manera exacta, es considera el conjunt de patrons com una matriu $P \in R^{N \times p}$, on cada columna és un patró x_i . Llavors, la solució que minimitza l'error quadràtic entre $W x_i$ i x_i , per a tots els i , ve donada per:

$$W = P(P^T P)^{-1} P^T$$

Aquesta és la pseudoinversa de Moore-Penrose aplicada al context de les xarxes de Hopfield. El resultat és una matriu de pesos que projecta qualsevol entrada sobre el subespai generat pels patrons entrenats. Això vol dir que els patrons entrenats són mínims d'energia garantits i exactes del sistema. A diferència de la regla de Hebb, aquí no es tracta d'esperar que l'energia disminueixi iterativament cap a un mínim correcte; el mínim ja s'ha definit algebraicament perquè coincideixi exactament amb el patró.

Aquest enfocament proporciona diversos avantatges:

- Eliminació de patrons espuris: com que només es projecta sobre l'espai generat pels patrons entrenats, no apareixen estats estables no desitjats.
- Recuperació robusta amb soroll: fins i tot quan s'introdueix un percentatge elevat de soroll al patró, la projecció sobre l'espai d'entrenament el porta molt a prop de la versió original.
- Estabilitat garantida: matemàticament, la solució de la pseudoinversa imposa la condició d'estabilitat sense necessitat de comprovacions iteratives.

Tanmateix, la pseudoinversa també té limitacions: requereix més càlcul (inversió de matriu), no és una regla local (no es pot implementar fàcilment amb plasticitat sinàptica biològica), i només funciona si els patrons no són linealment dependents. Tot i així, per a aplicacions computacionals on es busca una recuperació fiable i matemàticament precisa, la pseudoinversa representa una millora significativa respecte a la regla de Hebb.

En conclusió, el test amb la pseudoinversa no només mostra millor rendiment empíric, sinó que es fonamenta en una minimització explícita i algebraicament resolta de l'energia de la xarxa. Això converteix aquesta tècnica en una eina potent per a l'emmagatzematge i la recuperació de patrons en entorns amb alta densitat o similitud entre patrons.

Amb l'objectiu de validar la robustesa de la regla de la pseudoinversa davant escenaris sorollosos, s'ha realitzat una segona prova d'emmagatzematge i recuperació amb una distorsió del 20% sobre els 10 patrons binaritzats corresponents als dígits del 0 al 9 en format 8x8. S'ha mantingut una mateixa *random seed* per controlar la generació de soroll, i s'han tornat a comparar dues estratègies dinàmiques diferents: l'actualització síncrona i l'asíncrona.

5.2 Test de recuperació amb la pseudoinversa

Com es pot observar, els resultats amb actualització síncrona són excel·lents. En els 10 casos analitzats, la xarxa aconsegueix recuperar exactament els patrons originals, amb una similitud del 100% i una energia final constant de $E = -27.00$.

Aquest comportament confirma que la pseudoinversa projecta qualsevol versió distorsionada del patró directament cap al subespai generat per aquest conjunt de patrons,

i que aquest subespai conté únicament els mínims d'energia corresponents. La dinàmica síncrona actua de manera global, actualitzant simultàniament totes les unitats de la xarxa a cada iteració, cosa que permet una relaxació més eficient i directa cap a l'estat d'equilibri. Això evita rutes oscil·lants o desestabilitzadores dins de l'espai d'estats, consolidant una convergència neta cap al patró correcte.

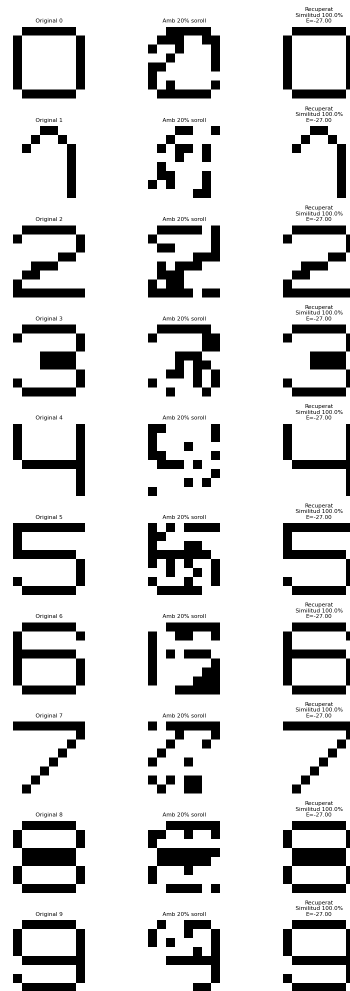


Figura 4: Recuperació amb la pseudoinversa i actualització síncrona, soroll del 20%.

Aquest resultat reforça l'eficàcia teòrica i pràctica de la pseudoinversa: no només assegura que cada patró entrenat sigui un punt fix exacte de la dinàmica, sinó que, quan el procés d'actualització és estable i global (com en el cas síncron), la recuperació és perfecta fins i tot amb un grau elevat de degradació del senyal d'entrada.

Actualització asíncrona

Els resultats obtinguts amb actualització asíncrona, tot i que la majoria de patrons s'han recuperat amb èxit, s'observen petites desviacions en alguns d'ells. En concret, els dígitos 2, 3, 4 i 6 presenten recuperacions parcials, amb valors de similitud inferiors al 100%. Això reflecteix que, tot i que la pseudoinversa ha generat un espai d'atractors pur i exempt de patrons espuris, la dinàmica d'actualització pot afectar el resultat final.

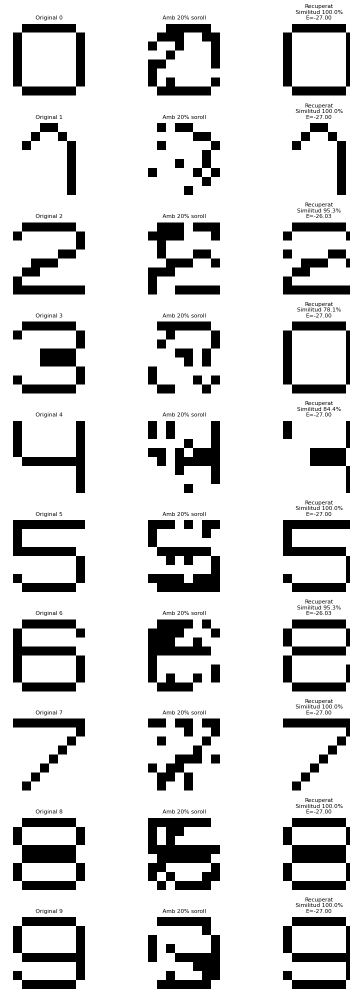


Figura 5: Recuperació amb la pseudoinversa i actualització asíncrona, soroll del 20%.

Aquestes diferències són degudes principalment a la naturalesa estocàstica de l'actualització asíncrona: en aquest enfocament, les neurones s'actualitzen una a una, en ordre aleatori, cosa que pot portar el sistema a trajectòries lleugerament diferents en cada execució. Encara que es parteixi del mateix patró corromput i amb la mateixa matriu de pesos, la variació en la seqüència d'actualització pot provocar que el sistema es relaxi cap a un estat proper, però no exactament igual al patró original.

Aquesta sensibilitat de la dinàmica asíncrona a petits canvis posa de manifest una distinció important: mentre que la pseudoinversa garanteix matemàticament l'existència de mínims correctes per als patrons entrenats, no controla el recorregut que hi fa la dinàmica, especialment quan aquesta incorpora aleatorietat. És a dir, l'òptim global pot no ser sempre assolit si el procés de relaxació no segueix un camí prou estable.

Aquest comportament també pot ser atribuït, en part, a la limitació en el nombre màxim d'iteracions establert per a la recuperació asíncrona. Aquesta restricció s'ha aplicat amb l'objectiu d'assegurar unes condicions comparables entre els dos models d'actualització —síncron i asíncron—, de manera que ambdós disposessin del mateix temps computacional per dur a terme la recuperació.

Tanmateix, si s'incrementa significativament aquest límit d'iteracions en el cas asíncron, els resultats milloren notablement, ja que l'estat té més oportunitats per

estabilitzar-se dins del mínim d'energia corresponent. Això posa de manifest que la inferior precisió observada inicialment en el model asíncron no es deu tant a la seva naturalesa estocàstica, sinó a una qüestió purament de terminació prematura del procés de relaxació.

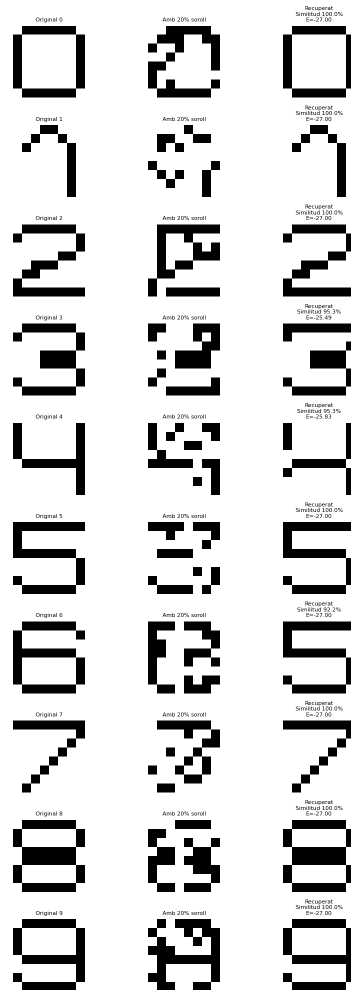


Figura 6: Recuperació amb la pseudoinversa i actualització asíncrona amb més iteracions.

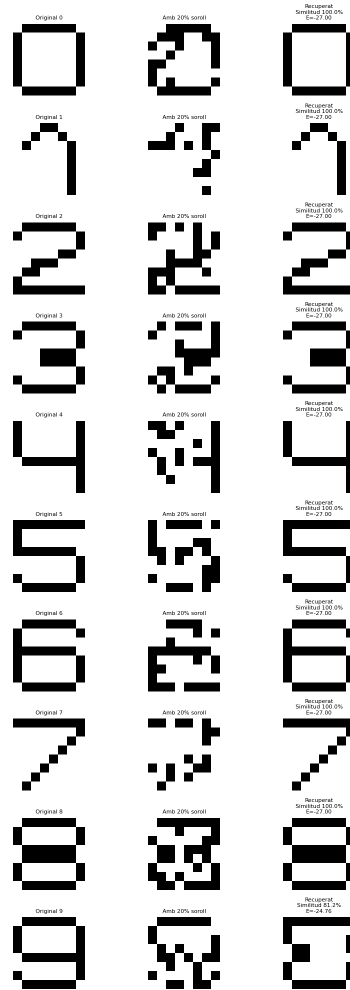


Figura 7: Recuperació amb la pseudoinversa i actualització asíncrona amb més iteracions. (2)

Per tal d'analitzar com afecta aquest paràmetre, s'han realitzat nous tests incrementant progressivament el nombre màxim d'iteracions. Si doblem el nombre d'iteracions (35), la recuperació millora parcialment: patrons que anteriorment no es recuperaven correctament, ara convergeixen al patró correcte. No obstant això, apareix un nou efecte advers: alguns patrons que ja estaven ben recuperats amb el límit inicial comencen a desviar-se del seu estat correcte. És el cas, per exemple, del dígit 9, que inicialment presentava una similitud del 100% i que, amb més iteracions, acaba en un estat amb una similitud inferior.

Aquest efecte es fa més evident quan tornem a doblem el nombre màxim d'iteracions (36). En aquest escenari, patrons com el 3 i el 4, que estaven perfectament estabilitzats, es degraden i acaben derivant cap a altres estats. Aquest comportament es pot atribuir a l'activitat persistent de la dinàmica asíncrona, la qual continua actualitzant neurones individualment encara que l'estat global ja sigui molt proper a un mínim. En absència d'un criteri de parada basat en estabilitat, el sistema pot acabar sortint d'un mínim correcte per la influència acumulada d'actualitzacions locals.

Aquesta observació posa de manifest que incrementar el nombre d'iteracions no sempre condueix a una millor recuperació. De fet, pot ser contraproduent si no es disposa d'un mecanisme de detecció de convergència efectiva. Aquest límit pràctic de

la recuperació asíncrona amb pseudoinversa subratlla la necessitat d'incorporar criteris de parada més sofisticats que no es basin exclusivament en la quantitat d'iteracions, sinó també en la variació global de l'estat o la no modificació de neurones durant un nombre determinat de passos consecutius.

Per tal de dur a terme una comparativa equitativa entre els diferents enfocaments, es farà servir la primera recuperació obtinguda amb actualització asíncrona. Ja que és la que està en condicions similars en quant a temps de computació amb la recuperació síncrona i poder fer una comparació equitativa.

5.3 Comparativa de resultats entre actualització síncrona i asíncrona

Els resultats d'aquesta segona prova reafirmen el potencial de la pseudoinversa per preservar patrons en xarxes d'alta capacitat, eliminant completament l'aparició d'attractors espuris que eren evidents en la versió amb regla hebbiana. Tanmateix, també posen de manifest la importància de la dinàmica d'actualització. L'actualització síncrona ofereix una trajectòria més controlada i predictable cap als patrons memoritzats, mentre que l'asíncrona, malgrat ser més propera a la implementació biològica, introdueix incerteses que poden comprometre parcialment la recuperació.

Per facilitar la comparació dels resultats, a la següent taula es resumeixen les similituds i energies finals obtingudes per cada dígit tant amb actualització síncrona com asíncrona. Es pot observar que, mentre que la versió síncrona recupera tots els patrons de manera perfecta (100% de similitud i energia constant), l'actualització asíncrona presenta petites desviacions en diversos dígit, especialment en els dígit 3 i 4.

Taula 2: Similitud i energia per patró amb actualització síncrona i asíncrona (pseudoinversa, 20% soroll).

Dígit	Síncrona		Asíncrona	
	Similitud	Energia	Similitud	Energia
0	100.0%	-27.00	100.0%	-27.00
1	100.0%	-27.00	100.0%	-27.00
2	100.0%	-27.00	95.3%	-26.03
3	100.0%	-27.00	78.1%	-27.00
4	100.0%	-27.00	84.4%	-27.00
5	100.0%	-27.00	100.0%	-27.00
6	100.0%	-27.00	95.3%	-26.03
7	100.0%	-27.00	100.0%	-27.00
8	100.0%	-27.00	100.0%	-27.00
9	100.0%	-27.00	100.0%	-27.00

5.4 Límit de tolerància al soroll amb la pseudoinversa amb activació síncrona

Un cop establerta la superioritat de la regla de la pseudoinversa amb activació síncrona en la recuperació de patrons amb soroll, és pertinent explorar fins on és capaç d'arribar aquest mètode abans de perdre la capacitat de reconduir correctament l'estat. En aquest experiment es testa la recuperació sota nivells de soroll elevats, concretament del 30% i del 40%, mantenint en tot moment l'actualització síncrona per garantir una dinàmica estable i sistemàtica.

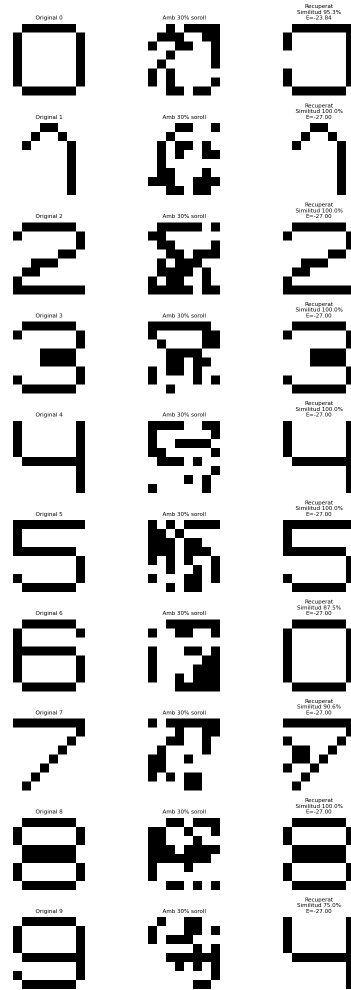


Figura 8: Recuperació amb pseudoinversa i activació síncrona amb 30% de soroll.

Amb un 30% de soroll (és a dir, 19 píxels invertits dels 64 totals), la xarxa encara mostra una capacitat destacable de recuperació: 6 dels 10 dígets es reconstrueixen amb una similitud del 100%, i la resta amb valors propers al 90%. Les imperfeccions apareixen principalment en patrons més simples o lineals, com el 0 i el 9, que tenen menys redundància estructural. Aquest resultat evidencia una tolerància robusta al soroll, gràcies a la capacitat del sistema de projectar l'estat cap al subespai generat pels patrons entrenats.

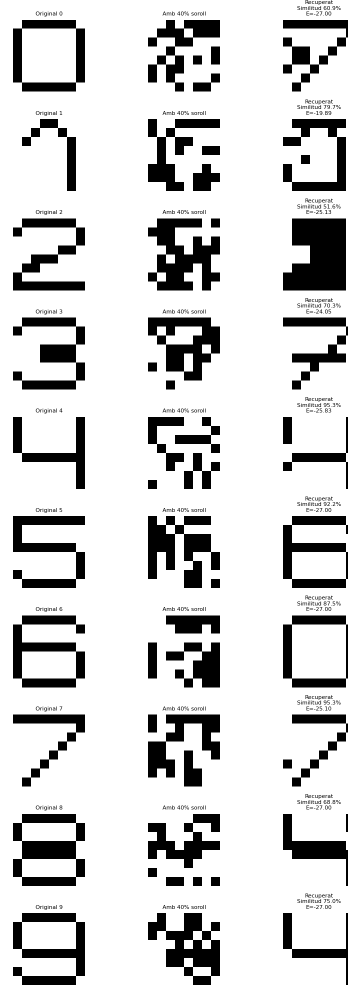


Figura 9: Recuperació amb pseudoinversa i activació síncrona amb 40% de soroll.

Quan el soroll augmenta fins al 40% (26 píxels invertits), el rendiment empitjora significativament. En aquest cas, només 2 dels 10 dígit arriben a valors de similitud superiors al 90%, i diversos patrons —com ara el 2, el 3 o el 8— divergeixen clarament del seu estat original. Es fa evident que la distorsió acumulada supera la capacitat de la xarxa per identificar el patró base, i la projecció resultant pot derivar cap a una configuració incorrecta.

Aquest experiment mostra que, en patrons de 64 dimensions, la pseudoinversa amb actualització síncrona manté una gran resistència al soroll fins al voltant del 30%, però a partir d'aquest llindar la fiabilitat es degrada ràpidament. Aquesta observació pot ajudar a delimitar el marge operatiu segur d'aquest tipus de xarxa quan es treballa amb patrons de baixa dimensionalitat.

6 Test de recuperació amb dígit 64x64

En aquesta secció s'analitza com varia el comportament de la xarxa de Hopfield quan s'augmenta significativament la resolució dels patrons, passant de 8×8 a 64×64 píxels. L'objectiu és observar si l'increment de complexitat i informació dels patrons permet una recuperació més precisa o si, al contrari, introdueix nous problemes relacionats amb la capacitat o la convergència.

S'han realitzat tres experiments de recuperació amb un nivell de soroll del 20%:

- Utilitzant la regla de Hebb amb actualització **síncrona**.
- Utilitzant la pseudoinversa amb actualització **síncrona**.
- Utilitzant la pseudoinversa amb actualització **asíncrona**.

En el cas dels dígit representats amb matrius de 64×64 , la xarxa conté $N = 4096$ neurones. Aplicant la capacitat crítica teòrica per a la regla de Hebb, $\alpha_c \approx 0,138$, podem estimar que el nombre màxim de patrons que la xarxa pot emmagatzemar sense degradació significativa és aproximadament

$$p_{\text{màxim}} \approx \alpha_c \cdot N = 0,138 \cdot 4096 \approx 565 \text{ patrons.}$$

En aquesta secció, només s'han entrenat 10 patrons corresponents als dígit de l'0 al 9, molt per sota del límit teòric. Per tant, la xarxa opera en un règim molt poc saturat i, des del punt de vista de la capacitat.

6.1 Recuperació amb la regla de Hebb

En aquest experiment s'ha incrementat la resolució dels patrons, passant de 8×8 a 64×64 píxels, amb l'objectiu d'analitzar si una major densitat d'informació millora la capacitat de recuperació de la xarxa. El test s'ha realitzat amb soroll del 20% i s'ha aplicat la regla d'aprenentatge de Hebb.

Només s'ha utilitzat actualització síncrona, ja que s'ha comprovat que els resultats amb actualització asíncrona són pràcticament idèntics tant pel que fa a les imatges recuperades com a les mesures de similitud i energia. Això permet simplificar l'anàlisi sense pèrdua de rigor.

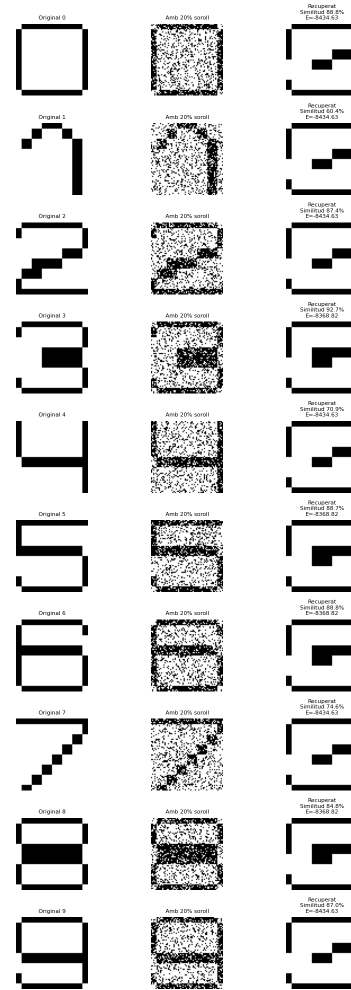


Figura 10: Recuperació amb la regla de Hebb amb un 20% de soroll

Els resultats obtinguts mostren que, malgrat l'augment substancial de dimensions, la xarxa continua tenint serioses dificultats per distingir els patrons. En els deu casos, l'estat recuperat tendeix clarament cap a una versió deformada del dígit 3, amb valors de similitud que en cap cas arriben al 95%. Alguns dígets, com ara el 1, el 4 i el 7, mostren rendiments molt baixos, amb similituds del 60.4%, 70.9% i 74.6% respectivament.

Aquest comportament reitera el fenomen ja observat en baixa resolució: la presència d'un atractor espuri dominant. L'atractor correspon a una estructura que agrupa característiques comunes de múltiples patrons —en aquest cas, una forma central i simètrica pròpia del 3— que actua com a punt de convergència per a la dinàmica de la xarxa.

Pel que fa a l'energia final dels estats recuperats, s'observen dues concentracions clares al voltant de $E = -8434.63$ i $E = -8368.82$, cosa que indica que, malgrat les diferències en el soroll o el patró original, la xarxa tendeix a només dos mínims d'energia. Això confirma que diversos patrons col·lapsen cap als mateixos estats, mostrant una capacitat de diferenciació molt limitada.

En definitiva, aquests resultats evidencien que l'increment del nombre de neurones no implica automàticament una millor capacitat de recuperació quan s'utilitza la regla

de Hebb. Això reforça la necessitat d'emprar altres mètodes, com la pseudoinversa, que proporcionen una codificació més precisa i estable.

6.2 Recuperació amb la pseudoinversa i activació síncrona

Un cop augmentada la resolució dels patrons a 64x64 píxels, el sistema continua mostrant una capacitat excel·lent de recuperació sota soroll quan es fa servir la regla de la pseudoinversa amb actualització síncrona. En aquest experiment, s'ha afegit un 20% de soroll a cadascun dels 10 dígets, invertint de manera aleatòria 819 de les 4096 neurones.

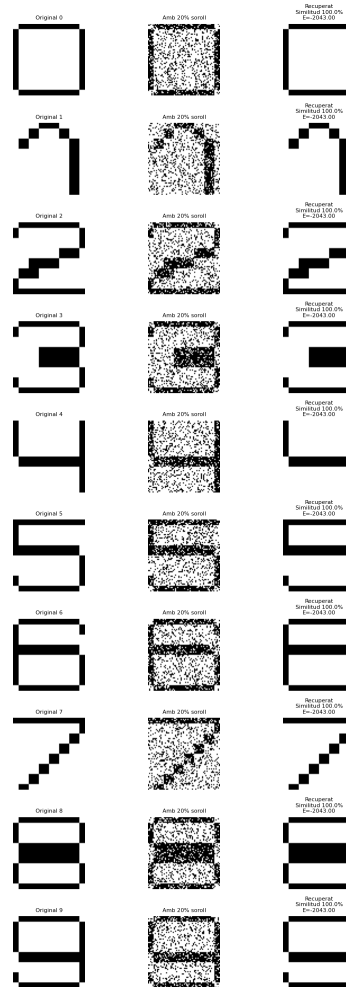


Figura 11: Recuperació amb la pseudoinversa i activació síncrona amb un 20% de soroll

El resultat és impecable: tots els patrons s'han recuperat amb un 100% de similitud, sense cap píxel incorrecte, i amb una energia final de $E = -2043.00$ en tots els casos. Aquesta energia, que és molt més baixa que en el cas dels dígets de 8x8 (on l'energia mínima era -27), reflecteix simplement la nova dimensió del sistema, però el comportament qualitatiu és el mateix: els patrons són mínims d'energia perfectament definits i estables.

Aquesta recuperació perfecta es deu a dos factors fonamentals. En primer lloc, l'ús de la pseudoinversa garanteix algebraicament que cada patró original és un atractor exacte del sistema. En segon lloc, el fet d'utilitzar actualització síncrona permet que

totes les neurones es relaxin simultàniament cap al seu valor òptim, sense les fluctuacions aleatòries de la versió asíncrona. Això assegura que la trajectòria dins de l'espai d'estats segueixi una evolució directa cap al mínim energètic predefinit.

A més, s'ha realitzat una prova addicional amb activació síncrona per comprovar el límit de tolerància al soroll en aquest espai de 4096 dimensions. S'ha incrementat el soroll fins al 40%, invertint aleatòriament més de 1600 píxels de cada patró, i sorprenentment el sistema ha sigut capaç de recuperar tots els dígets amb una similitud perfecta del 100%.

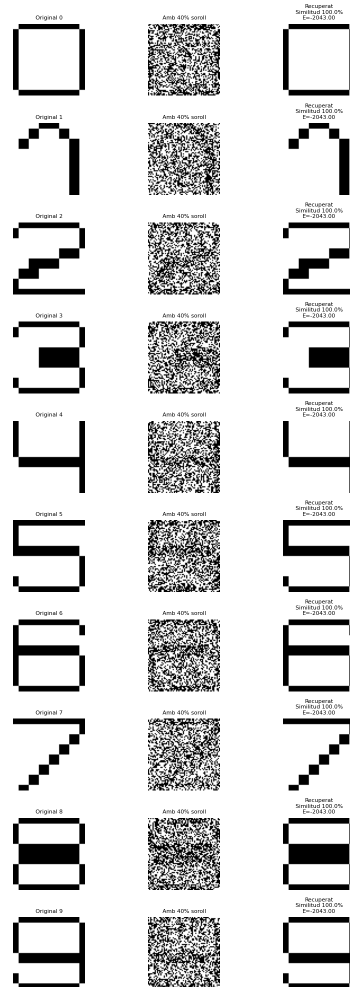


Figura 12: Recuperació amb pseudoinversa i activació síncrona amb 40% de soroll.

Aquest resultat posa de manifest una millora significativa respecte als experiments amb patrons de 8×8 : mentre que amb aquella resolució la recuperació amb un 40% de soroll no era viable —donant lloc a patrons clarament deformats o mal classificats—, en el cas dels patrons de 64×64 , la pseudoinversa amb actualització síncrona ha aconseguit una recuperació perfecta en tots els dígets, amb una similitud del 100%. Això indica que l'augment de dimensionalitat no només incrementa la capacitat representacional del sistema, sinó que també reforça la definició dels mínims d'energia, fent-los més profunds i selectius. La xarxa és així capaç d'absorbir distorsions molt més intenses sense perdre la identitat dels patrons, cosa que no s'assolia en entorns de baixa resolució.

Així doncs, l'augment de resolució no només no empitjora els resultats, sinó que

aporta més estabilitat numèrica i convergència clara, sempre que es mantinguin les condicions adequades d'actualització i estructura de pesos.

6.3 Recuperació amb la pseudoinversa i activació asíncrona

Quan s'aplica la regla de la pseudoinversa amb actualització asíncrona en el cas de dígets de 64×64 píxels amb un 20% de soroll, els resultats continuen sent molt satisfactoris, però amb lleugeres diferències respecte al cas síncron.

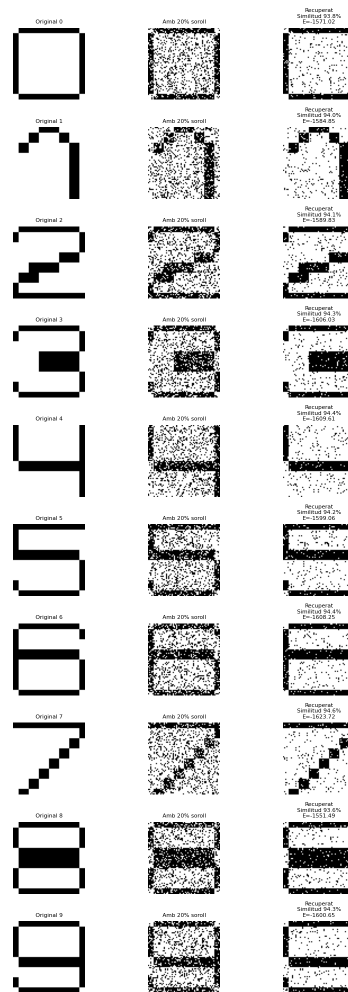


Figura 13: Recuperació amb la pseudoinversa i activació asíncrona amb un 20% de soroll

Com es pot veure en els patrons recuperats, la xarxa és capaç de reconduir l'estat inicial sorollós cap a formes que s'aproximen molt als originals, amb valors de similitud que ronden el 94%. No obstant això, a diferència del cas síncron, on s'aconseguia una similitud pràcticament perfecta (100%), aquí la convergència no arriba sempre al mínim global, sinó a configuracions lleugerament desviades. Aquestes desviacions es tradueixen en petits punts fora de lloc o píxels inconsistents, sovint a zones perifèriques del dígit.

Aquest fenomen es pot atribuir al caràcter estocàstic del procés d'actualització asíncrona. Com que s'actualitzen les neurones de manera aleatòria i independent, la trajectòria de l'estat dins de l'espai d'energia pot seguir camins diferents a cada execució, i quedar-se atrapada en estats quasi-estables abans d'assolir el mínim exacte.

A més, el fet de tenir un espai d'estats molt més gran (4096 dimensions) incrementa les possibilitats que el sistema es desplaci lleugerament del mínim projectat.

Tot i així, el rendiment continua essent excel·lent, i no s'observa cap col·lapse de patrons, ni la presència d'atractors espuris com succeïa amb la regla de Hebb. La identitat individual dels patrons entrenats es manté i cap patró deriva cap a la forma d'un altre, cosa que demostra la robustesa de la pseudoinversa, fins i tot en escenaris amb alta dimensionalitat i soroll significatiu.

Aquest comportament ja havia estat discutit en el cas dels patrons de 8×8 , on es va observar que la recuperació asíncrona podia requerir un nombre elevat d'iteracions per assolir la convergència completa. A diferència de l'actualització síncrona —que modifica totes les neurones alhora i arriba ràpidament a un estat estable—, la versió asíncrona actualitza només una neurona per pas, cosa que pot allargar el procés de relaxació.

En aquest cas, una nova prova amb un nombre d'iteracions significativament superior (4 vegades més gran) ha mostrat que els patrons inicialment mal recuperats convergeixen cap a l'original amb una similitud pràcticament perfecta, mentre que els ja recuperats es mantenen estables. Això demostra que, fins i tot en espais de 4096 dimensions, la regla de la pseudoinversa és capaç de definir atractors nítids i estables, sempre que se li concedeixi prou temps de relaxació dins el procés dinàmic.

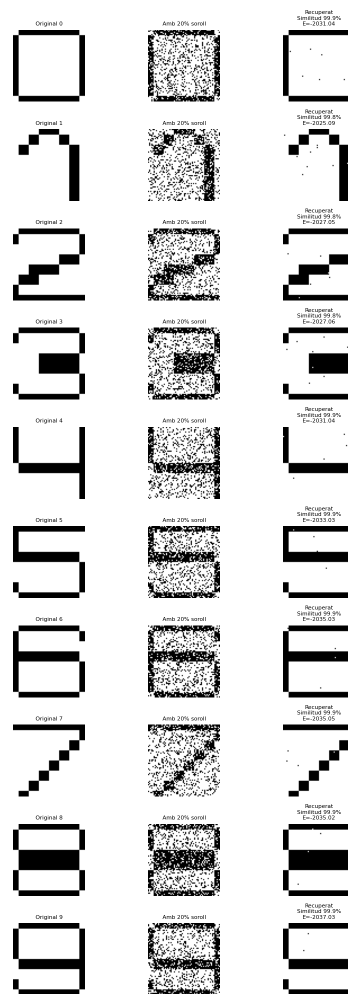


Figura 14: Recuperació amb la pseudoinversa i actualització asíncrona amb més iteracions.

Cal destacar una diferència significativa respecte als resultats obtinguts amb patrons de 8×8 . En aquell cas, incrementar el nombre màxim d'iteracions en la recuperació asíncrona podia tenir efectes adversos: alguns patrons que ja estaven correctament recuperats acabaven desviant-se del seu atractor correcte, probablement per la presència de mínims d'energia espuris en un espai de baixa dimensionalitat.

En canvi, amb patrons de 64×64 , aquest efecte no es produeix. Quan s'augmenta el nombre d'iteracions, les recuperacions milloren progressivament i de manera estable. Els patrons que inicialment presentaven imperfeccions tendeixen a convergir cap als seus estats originals amb més precisió, mentre que els que ja estaven correctes es mantenen inalterats. Aquest comportament reforça la idea que, en espais d'alta dimensionalitat, la projecció realitzada per la pseudoinversa genera mínims d'energia més profunds i ben definits, i que la dinàmica asíncrona, si se li concedeix prou temps, és capaç d'arribar-hi de manera fiable. Si doblem les iteracions respecte la última prova, aconseguim una recuperació del 100% en tots els dígets.

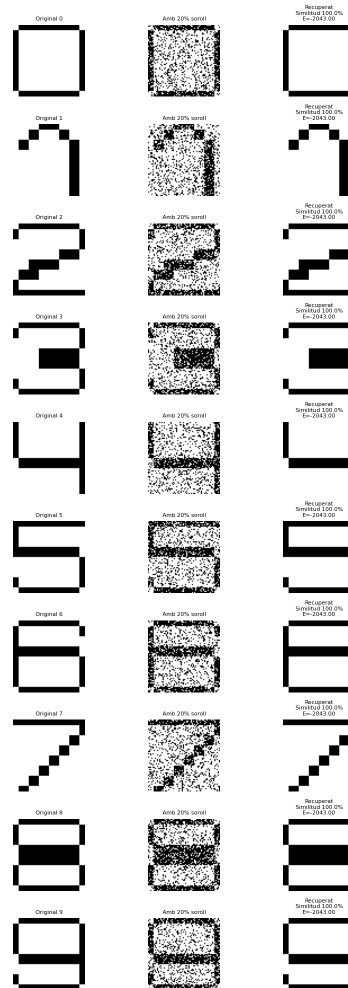


Figura 15: Recuperació amb la pseudoinversa i actualització asíncrona amb més iteracions. (2)

Conclusió

L'augment de la resolució dels patrons de 8×8 a 64×64 ha permès explorar com la dimensionalitat afecta la dinàmica de recuperació de les xarxes de Hopfield. El prin-

El principal resultat observat és que, tot i disposar de molta més informació per patró (4096 neurones en lloc de 64), el problema dels atractors espuris persisteix quan s'utilitza la regla de Hebb. Aquest comportament es manifesta clarament en la tendència de la xarxa a col·lapsar diversos patrons cap a una forma dominant (sempre formes properes L 3), tal com ja passava en baixa resolució. Per tant, l'increment de dimensionalitat no resol per si sol la problemàtica associada als mínims no desitjats generats per Hebb.

En canvi, els resultats amb la pseudoinversa són molt més robustos. Tant amb actualització síncrona (que ja ho era per 8×8) com asíncrona, la xarxa és capaç de recuperar amb gran precisió els patrons originals, fins i tot amb un nivell de soroll elevat (20%). A més, s'ha constatat que amb actualització asíncrona, si es permet un nombre suficientment elevat d'iteracions, les recuperacions milloren progressivament sense empitjorar patrons ja estabilitzats. Aquest comportament contrasta amb els resultats a 8×8 , on un excés d'iteracions podia desestabilitzar patrons correctes. Això indica que en espais d'alta dimensionalitat, els mínims definits per la pseudoinversa són més profunds i ben delimitats, facilitant la convergència fiable fins i tot amb dinàmiques aleatòries.

En resum, l'augment de la resolució no millora els resultats amb la regla de Hebb, però sí potencia la precisió i estabilitat del sistema quan s'utilitza una arquitectura algebraicament òptima com la pseudoinversa, especialment en combinació amb una dinàmica prou relaxada d'actualització (activació asíncrona).

7 Test de recuperació amb dígets amb millor definició

Amb l'objectiu d'analitzar si els problemes de recuperació observats amb la regla de Hebb són deguts, en part, a la poca claredat i definició dels patrons originals, s'ha preparat un nou conjunt de dígets amb una estructura molt més neta, estilitzada i coherent. Aquests nous patrons presenten contorns més suaus, proporcions típiques de tipografies estàndard i diferències més marcades entre els dígets. La seva aparença s'allunya de la construcció geomètrica pixelada inicial, i s'aproxima a una representació visual més realista (Figura 16).



Figura 16: Dígets del 0 al 9 amb millor definició.

Aquest canvi ens permet posar a prova una hipòtesi rellevant: si la dificultat de la regla de Hebb per distingir patrons en tests anteriors prové principalment de la seva limitació teòrica (interferència entre patrons, atractors espuris, etc.), o si també hi juga un paper important la qualitat visual dels dígets utilitzats.

Amb patrons millor definits, esperem observar una disminució de la confusió entre patrons, i veure si la xarxa és capaç de mantenir-los més ben diferenciats. Això ens ajudarà a entendre fins a quin punt el rendiment de la regla de Hebb pot dependre no només del model matemàtic, sinó també de la representació gràfica dels patrons.

Tot i que en aquesta secció s'han utilitzat dígets més ben definits visualment, aquests continuen tenint una representació de 64×64 píxels, per tant el nombre de neurones de la xarxa roman igual: $N = 4096$. La capacitat teòrica màxima, segons la regla de Hebb, continua sent d'uns $p_{\text{màxim}} \approx 0,138 \cdot N \approx 565$ patrons. Atès que només s'han fet servir 10 dígets, la xarxa segueix molt lluny de la seva saturació.

A continuació, s'analitzen els resultats obtinguts utilitzant aquesta nova base visual.

7.1 Recuperació amb la regla de Hebb

En aquesta prova s'ha aplicat la regla de Hebb amb actualització síncrona (els resultats amb activació asíncrona i nombre elevat d'iteracions són idèntics). Malgrat l'alta definició dels dígets, la xarxa mostra novament una clara tendència cap a un únic atractor espuri. Tots els patrons recuperats col·lapsen cap a una mateixa forma semblant al dígit 3, amb independència del dígit original o de la distribució del soroll.

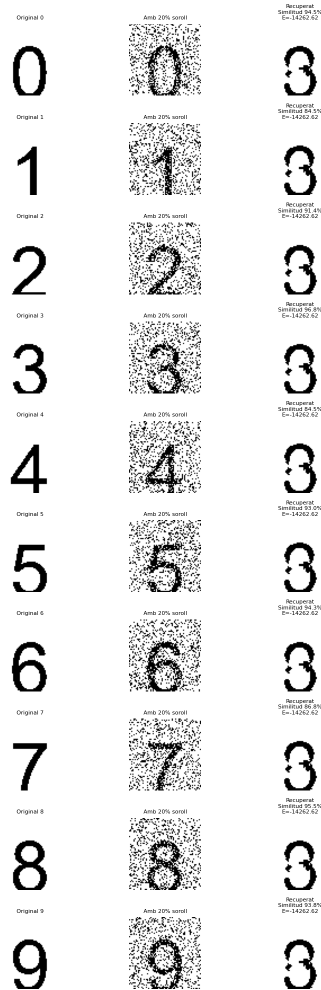


Figura 17: Recuperació amb regla de Hebb (dígets definites)

Aquest comportament coincideix plenament amb el que ja s'havia observat amb patrons menys definits, tant de resolució 8×8 com 64×64 . Això reforça la idea que l'origen del problema no rau en la qualitat gràfica o la resolució dels patrons, sinó en que son molt "semblants" els dígets. Com que molts nombres comparteixen estructures gràfiques similars (segments verticals, bucles, simetries centrals, etc.), la dinàmica hebbiana propicia que la xarxa convergeixi cap a una combinació dominant d'aquestes formes, independentment del patró d'origen.

7.2 Recuperació amb la pseudoinversa

A diferència de la regla de Hebb, la pseudoinversa continua mostrant un comportament excel·lent. Tant amb actualització síncrona com asíncrona (amb prou iteracions), tots els patrons s'han recuperat amb una similitud del 100% i una energia mínima igual per a tots els casos.

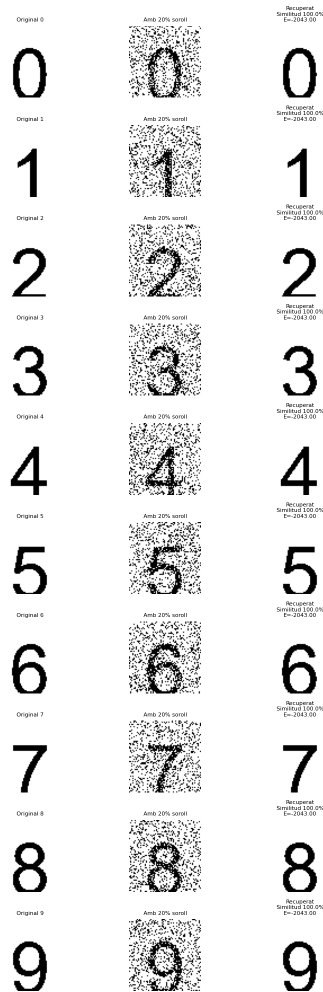


Figura 18: Recuperació amb pseudoinversa (dígets definits)

Aquest resultat indica que la qualitat de la definició gràfica no és una condició necessària per a l'èxit de la pseudoinversa, però sí que confirma la seva robustesa: fins i tot amb més detall, la regla és capaç de mantenir els patrons originals com a mínims d'energia ben definits i estables.

Conclusió

Els resultats obtinguts amb dígets d'alta definició mostren que, tot i la qualitat gràfica dels patrons, la regla de Hebb continua generant col·lapse cap a un atractor espuri. A diferència d'altres conjunts on la confusió es podia atribuir a una baixa resolució, en aquest cas queda clar que el problema s'explica pel fet que els dígets comparteixen molts trets estructurals. Això fa que els patrons siguin massa similars entre si i que la regla de Hebb, en reforçar les coincidències, acabi construint un estat intermig que combina fragments de diversos dígets. De nou, l'atractor dominant adopta una forma pròpia al dígit 3, com ja s'havia observat en proves anteriors. Per contra, la pseudoinversa continua oferint una recuperació precisa i robusta, independentment de la naturalesa dels patrons utilitzats.

8 Test de recuperació amb lletres en alta definició

Després d’haver analitzat la capacitat de recuperació de la xarxa amb conjunts de 10 patrons (els dígit), ens proposem estudiar com respon el sistema quan s’incrementa significativament el nombre de patrons emmagatzemats. En aquest capítol, utilitzarem les 26 lletres majúscules de l’alfabet, generades a una resolució de 64×64 píxels, ocupant tot l’espai disponible a la imatge per maximitzar la informació i claredat de cada patró.

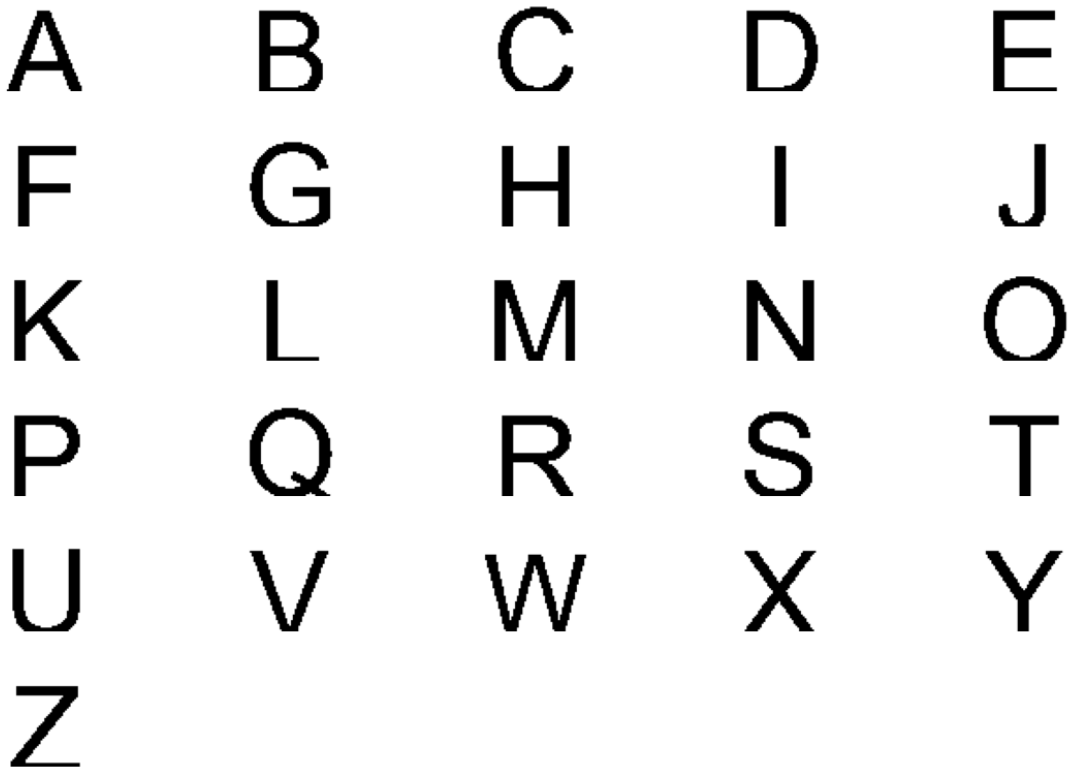


Figura 19: Lletres de la A a la Z en alta definició

Aquesta ampliació permet analitzar com afecta la càrrega de memòria —és a dir, el nombre de patrons simultanis— a la capacitat de recuperació i estabilitat del sistema. A diferència dels experiments previs, aquí es treballarà amb un conjunt de patrons més dens i variat, amb formes tipogràfiques més complexes i menys simetries internes. La naturalesa més diversa de les lletres permet posar a prova la robustesa de les regles d’aprenentatge i la capacitat de discriminació de la xarxa.

En el cas de les lletres majúscules de l’alfabet, s’han utilitzat 26 patrons, cadascun representat en una matriu de 64×64 píxels, és a dir, amb $N = 4096$ neurones. Això implica que la capacitat teòrica màxima de la xarxa, segons la regla de Hebb, es manté al voltant de $p_{\text{màxim}} \approx 0,138 \cdot N \approx 565$ patrons. Tot i haver més patrons que en els casos anteriors (dígit), la xarxa continua molt per sota del seu límit de saturació.

L’objectiu principal és observar si l’increment del nombre de patrons:

- Afecta negativament la recuperació amb la regla de Hebb, incrementant la presència d’atractors espuris o patrons deformats.

- És manejat adequadament per la pseudoinversa, mantenint la capacitat de recuperar patrons individuals sense interferències.

Aquest escenari permetrà veure fins a quin punt les dues regles escalen en entorns amb més càrrega de memòria, i si es manté la tendència observada als capítols anteriors: limitacions severes per la regla de Hebb i una alta precisió de recuperació amb la pseudoinversa.

Com en els casos anteriors, s'ha aplicat un 20% de soroll binari a cada patró abans d'iniciar el procés de recuperació, i s'han realitzat proves tant amb actualització síncrona com asíncrona.

8.1 Recuperació amb la regla de Hebb

Com ja s'ha observat en altres capítols, la regla de Hebb mostra dificultats per establir patrons individuals quan hi ha moltes similituds entre ells. Aquest comportament es reitera clarament en el test amb les 26 lletres.

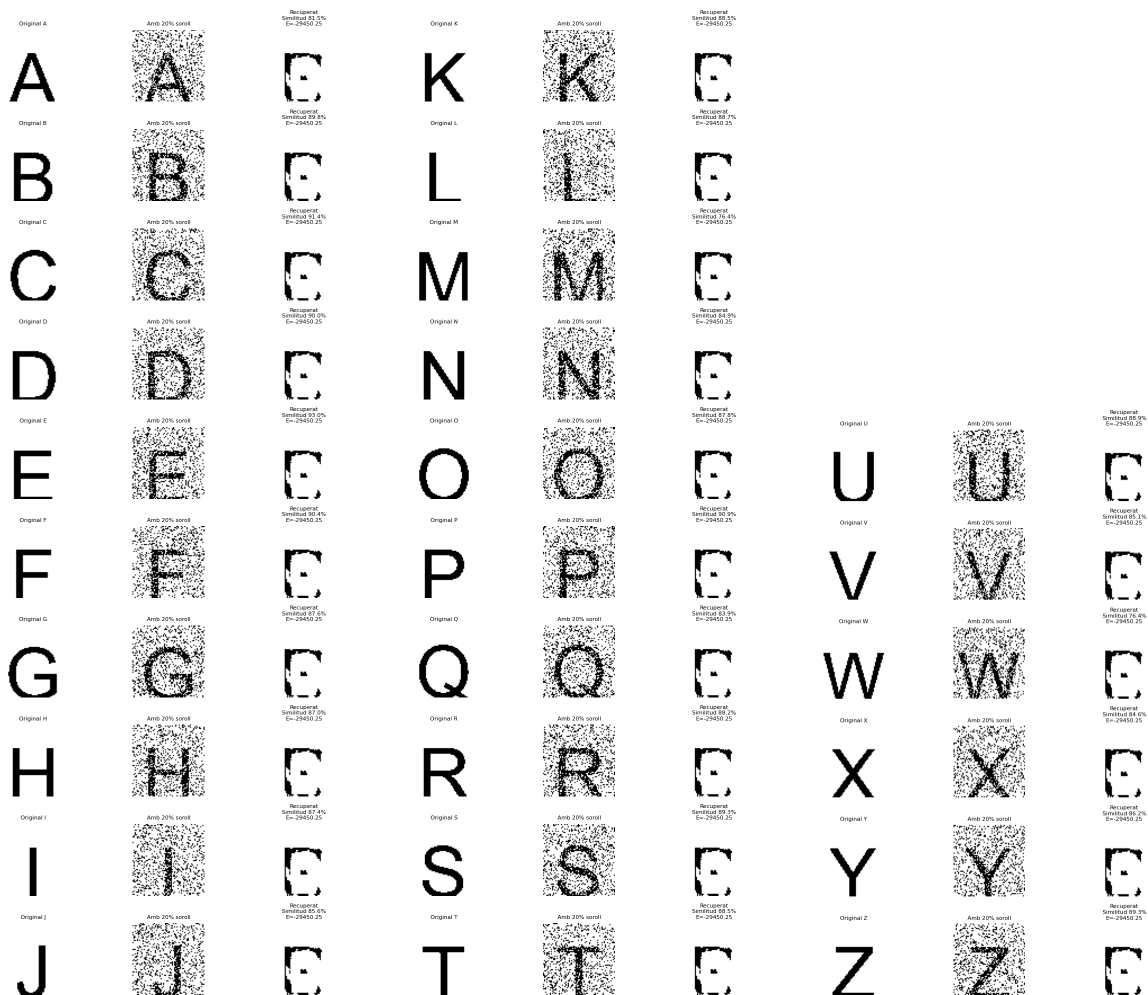


Figura 20: Recuperació de patrons de lletres amb la regla de Hebb.

Un cop més, la regla de Hebb torna a mostrar els seus límits. Tot i que els patrons originals tenen una alta qualitat visual i estan ben centrats dins de la matriu, el re-

sultat de la recuperació és força deficient. En pràcticament tots els casos, els patrons recuperats col·lapsen cap a una figura deformada, que recorda clarament una **lletra E**. Aquest fenomen ja havia estat detectat en les proves amb dígit, on els atractors espuris tendien cap a un dígit 3 o una forma similar. Ara, amb les lletres, es repeteix l'efecte: molts patrons comparteixen línies verticals i horitzontals, cosa que provoca que la dinàmica de Hebb reforci aquestes estructures comunes i generi una forma que les sintetitza.

Això confirma que, quan els patrons són visualment semblants —com és el cas de les lletres—, la regla de Hebb no té prou capacitat de discriminació i acaba agrupant trets compartits en un atractor únic, espuri i dominant.

8.2 Recuperació amb la pseudoinversa

Quan s'utilitza la pseudoinversa com a regla d'aprenentatge, el comportament de la xarxa és impecable. Amb 26 lletres diferents i un 20% de soroll, la recuperació és perfecta en tots els casos, amb una similitud del 100% i sense cap error estructural. Tant amb activació síncrona com asíncrona —sempre que es permeti un nombre suficient d'iteracions—, el sistema reconstrueix amb precisió cada patró original.

Aquest resultat és coherent amb les observacions fetes en proves prèvies amb dígit: en espais de gran dimensionalitat, la pseudoinversa genera atractors nítids i profunds, capaços de resistir distorsions intenses sense degradar-se. A diferència del que es veia en espais més petits, l'activació asíncrona aquí no només manté el rendiment, sinó que millora progressivament la convergència, corregint gradualment els patrons mal recuperats i mantenint estables els ja correctes.



Figura 21: Recuperació de patrons de lletres amb la pseudoinversa.

Aquest resultat confirma que la pseudoinversa genera atractors energètics nítids i ben definits, fins i tot en espais de molta càrrega i alta dimensionalitat. No es detecta cap col·lapse, ni s'observen patrons desplaçats o deformats, cosa que demostra una gran estabilitat interna del sistema.

Conclusió

Aquest experiment aporta una confirmació important: el principal factor que dificulta la recuperació en una xarxa de Hopfield amb la regla de Hebb no és la qualitat gràfica dels patrons, sinó la seva semblança estructural. Les lletres, tot i estar perfectament definides, comparteixen moltes formes bàsiques (línies verticals, barres horitzontals, simetries centrals), i això provoca que la regla de Hebb col·lapsi cap a un atractor espuri comú, en aquest cas proper a la lletra E —de la mateixa manera que passava amb el dígit 3 en els casos anteriors.

En canvi, la pseudoinversa torna a demostrar la seva superioritat: amb 26 patrons i un espai de 4096 dimensions, continua oferint una recuperació perfecta i estable. A més, es confirma el comportament observat en altres proves: en espais d'alta dimensionalitat, l'activació asíncrona esdevé tan robusta com la síncrona, sempre que se li permeti prou temps de relaxació.

Aquest capítol reforça la idea que, per evitar col·lapses espuris en xarxes amb molts patrons semblants, no n'hi ha prou amb augmentar la resolució. Cal emprar una representació algebraicament òptima, com la que proporciona la pseudoinversa.

9 Test de recuperació amb patrons diferenciats

En aquest apartat s'analitza novament el comportament de la xarxa de Hopfield clàssica utilitzant la regla de Hebb i la pseudoinversa, però aquesta vegada es modifica l'estratègia de prova. A diferència dels experiments anteriors —on molts patrons compartien estructura visual i això afavoria la presència d'atractors espuris—, aquí es busca afavorir la recuperació dissenyant un conjunt de dades format per patrons binaris de 64×64 píxels clarament diferenciats entre ells.

L'objectiu d'aquesta configuració és comprovar si, eliminant la similitud entre patrons, la regla de Hebb és capaç de recuperar correctament els estats memoritzats. En certa manera, s'està “ajudant” la dinàmica hebbiana a superar les seves limitacions estructurals per tal de verificar fins on pot arribar si se li redueix l'ambigüitat entre patrons. I verificar que la pseudoinversa és molt efectiva.



Figura 22: Conjunt de 16 patrons binaris de mida 64×64 ben diferenciats

S'entrenen dues xarxes de Hopfield amb aquest conjunt: una amb la regla de Hebb i una altra amb la regla de la pseudoinversa. A continuació es presenten els resultats de recuperació dels patrons alterats amb un 20% de soroll.

9.1 Recuperació amb la regla de Hebb

Tot i l'intent de facilitar la tasca de recuperació amb patrons ben diferenciats, els resultats amb la regla de Hebb són clarament deficients. En aquest experiment, tots els patrons convergeixen cap a un mateix atractor espuri, totalment desvinculat dels patrons originals. A diferència d'altres casos, on alguns patrons podien mantenir-se

prop de l'original, en aquesta prova es constata un col·lapse total de la dinàmica: independentment del punt d'inici, la xarxa tendeix sempre cap a una mateixa configuració errònia.

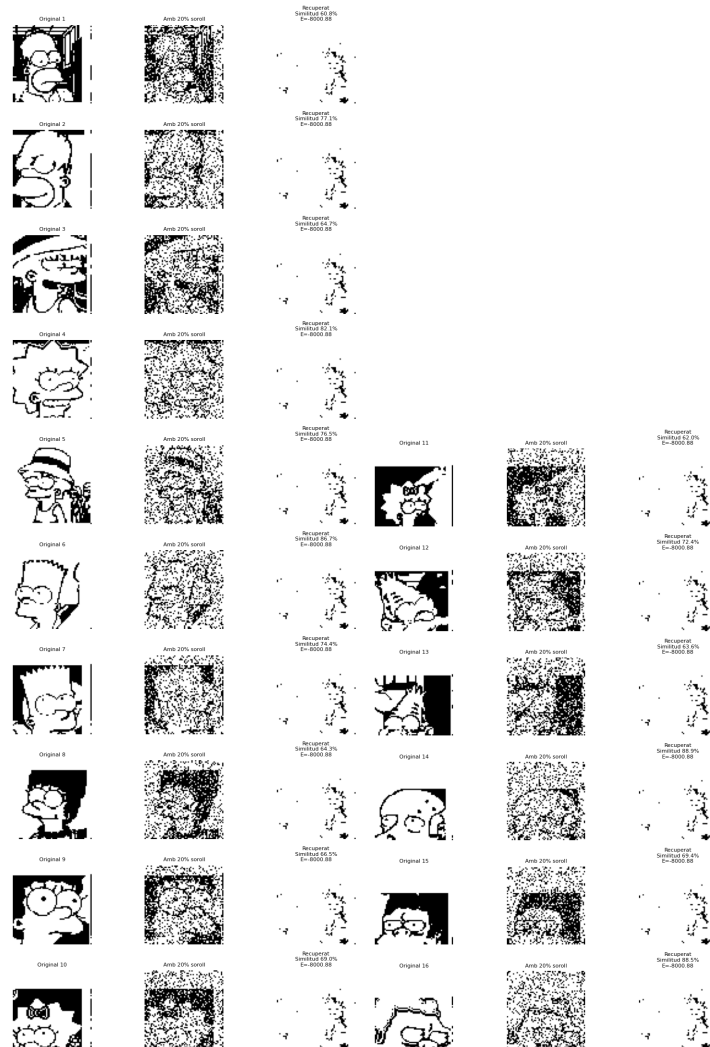


Figura 23: Recuperació de patrons diferenciats amb la regla de Hebb.

Aquest comportament reflecteix una de les limitacions estructurals de la regla de Hebb: fins i tot amb patrons molt diferenciats, el solapament entre productes interns pot induir a convergència cap a estats no desitjats. Recordem que la capacitat teòrica de la regla de Hebb és:

$$p_{max} \approx 0.138 \cdot n = 0.138 \cdot 4096 \approx 565 \text{ patrons}$$

Per tant, en aquest cas no tenim problemes de capacitat ja que tenim 16 patrons.

9.2 Recuperació amb la pseudoinversa

Tal com ja s'ha vist en altres capítols, la regla de la pseudoinversa continua oferint una recuperació impecable. Amb un 20% de soroll introduït artificialment en cada

patró i una dinàmica d'activació síncrona, el sistema recupera exactament els patrons originals en tots els casos, amb una similitud propera al 100%.



Figura 24: Recuperació de patrons diferenciats amb la pseudoinversa.

Aquest comportament reforça l'observació feta en experiments previs: la pseudoinversa genera un conjunt d'atractors ben separats, energèticament profunds i estables, fins i tot quan s'afegeix soroll considerable. La robustesa d'aquest enfocament algebraic permet una discriminació precisa entre patrons, independentment de la seva semblança o de la seva complexitat gràfica.

Conclusió

Tot i l'ús de patrons fortament diferenciats, la regla de Hebb no és capaç de garantir la recuperació correcta de tots els patrons. La pseudoinversa, en canvi, mostra una robustesa total. No obstant això, el càlcul de la pseudoinversa té un cost computacional elevat i no pot ser generalitzat com una dinàmica implícita de xarxa.

Aquesta observació ens porta a explorar models alternatius que puguin:

- Recuperar patrons sense requerir càlculs explícits de pseudoinverses.

- Tenir una dinàmica definida per una energia que es pugui minimitzar.
- Assegurar la recuperació correcta fins i tot amb soroll alt.

Aquests requisits són els que motiven l'estudi de les xarxes modernes de Hopfield, que s'introduiran a la següent secció.

10 Xarxes de Hopfield amb Energia Exponencial

Els resultats de les seccions anteriors han deixat clar que la regla de Hebb presenta limitacions greus, fins i tot en condicions favorables on els patrons són clarament diferenciats entre si. Tot i reduir la interferència visual entre els patrons, la xarxa col·lapsa igualment cap a un atractor espuri comú, cosa que evidencia una manca de capacitat de discriminació robusta.

Aquesta observació motiva la necessitat d'explorar noves formulacions que permetin una recuperació més precisa i estable dels patrons memoritzats, sense haver d'utilitzar la pseudoinversa. En aquesta línia, una proposta rellevant és la que fa *Demircigil et al.* amb una redefinició de la funció d'energia i de la dinàmica associada. Aquesta nova formulació, coneguda com a *Modern Hopfield Network*, busca superar les limitacions estructurals dels models clàssics, augmentant la capacitat d'emmagatzematge i reduint la presència d'atractors espuris. En la secció següent es presenta aquesta formulació amb detall.

10.1 Nova funció d'energia

La nova funció d'energia es defineix com:

$$E(\xi) = - \sum_{i=1}^N \exp(\beta \cdot x_i^\top \xi)$$

on:

- $\xi \in \{-1, +1\}^d$ és l'estat actual de la xarxa (vector bipolar),
- $\{x_i\}_{i=1}^N$ són els patrons memoritzats (també vectors bipolars),
- β és un paràmetre d'escala que controla la sensibilitat de l'energia als productes interns (normalment petit, per exemple $\beta = 0.005$).

Aquesta funció d'energia no es basa en una matriu de pesos W , sinó en una suma ponderada per $\exp(\beta \cdot x_i^\top \xi)$, que fa que els patrons similars a l'estat actual tinguin un impacte exponencialment més gran.

10.2 Regla d'actualització

A diferència de les xarxes clàssiques, aquí no es pot fer una actualització directa a través de $W\xi$. En lloc d'això, es considera com canvia l'energia si es modifica un sol component $\xi[l]$ de l'estat:

$$\xi^{\text{new}}[l] = \text{sign} [-E(\xi^{(l+)}) + E(\xi^{(l-)})]$$

on:

- $\xi^{(l+)}$ és el vector ξ amb el component l posat a $+1$,
- $\xi^{(l-)}$ és el mateix vector amb el component l posat a -1 ,

- i s'escull el valor que minimitza l'energia.

Aquest procés es repeteix iterativament per cada component l de manera asíncrona, fins que l'estat convergeix a un mínim local d'energia.

10.3 Avantatges

Aquesta formulació ofereix diversos avantatges:

- **Capacitat d'emmagatzematge exponencial:** es pot demostrar que la xarxa pot emmagatzemar fins a $\mathcal{O}(2^{d/2})$ patrons (on d és la dimensió dels vectors), superant àmpliament el límit clàssic de $0.138 \cdot d$.
- **Evita atractors espuris:** l'energia exponencial afavoreix patrons memoritzats en detriment de configuracions intermèdies.

11 Test de recuperació amb l'energia exponencial

Després de constatar les limitacions de la regla de Hebb, fins i tot en condicions òptimes, en aquesta secció s'analitza una nova proposta energètica aplicada a la recuperació de patrons: la formulació exponencial introduïda per *Demircigil et al.*. Aquesta proposta redefineix la funció d'energia per incorporar una interacció exponencial entre els patrons emmagatzemats i l'estat actual de la xarxa, afavorint atractors més estables i discriminatius.

Per validar l'efectivitat d'aquesta formulació, es realitzaran dues proves diferenciades:

- Recuperació dels dígit binaris del 0 al 9, en resolució alta, amb formes ben definides i centrades.
- Recuperació del conjunt de 16 patrons clarament diferenciats, emprats prèviament per "ajudar-la regla de Hebb.

En tots dos casos, els patrons tenen una resolució de 64×64 píxels, és a dir, un total de $d = 4096$ neurones. A diferència de la regla de Hebb, que presenta una capacitat teòrica limitada a uns $0,138 \cdot d \approx 565$ patrons, la formulació energètica de la Modern Hopfield Network pot arribar a emmagatzemar un nombre exponencial de patrons, fins a l'ordre de $p_{\text{màxim}} \sim 2^{d/2}$.

En ambdós casos aplicarem diferents percentatges de soroll. S'ha utilitzat una actualització asíncrona per facilitar una relaxació progressiva del sistema.

11.1 Recuperació amb dígit del 0 al 9 amb alta definició

En aquesta primera prova s'han emprat patrons binaris corresponents als dígit del 0 al 9, cadascun representat en una imatge de 64×64 píxels amb alta definició que ja han estat utilitzats anteriorment. Tots els patrons han estat alterats amb un 20% i un 40% de soroll binari, i posteriorment s'ha aplicat la dinàmica de recuperació associada a l'energia exponencial.

Amb un 20% de soroll, la recuperació ha estat perfecta en tots els casos: els patrons recuperats coincideixen exactament amb els originals i no es detecta cap forma espúria o distorsió. Això demostra que, en condicions moderades de soroll, la formulació exponencial és capaç de generar atractors energètics prou profunds i diferenciats per garantir una convergència robusta.

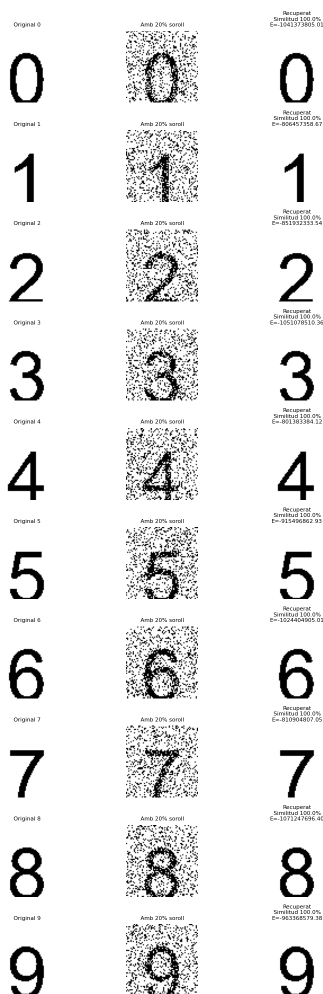


Figura 25: Recuperació dels dígets del 0 al 9 amb alta definició amb l'energia exponencial i 20% de soroll.

Quan el soroll augmenta fins al 40%, la recuperació ja no és perfecta, però continua essent funcional: si bé alguns patrons es recuperen amb errors (confusions entre dígets propers), tots els resultats convergeixen cap a patrons reals del conjunt emmagatzemat. És a dir, la xarxa no col·lapsa cap a atractors espuris, com sí que feia amb la regla de Hebb, sinó que es manté dins del conjunt de solucions vàlides.

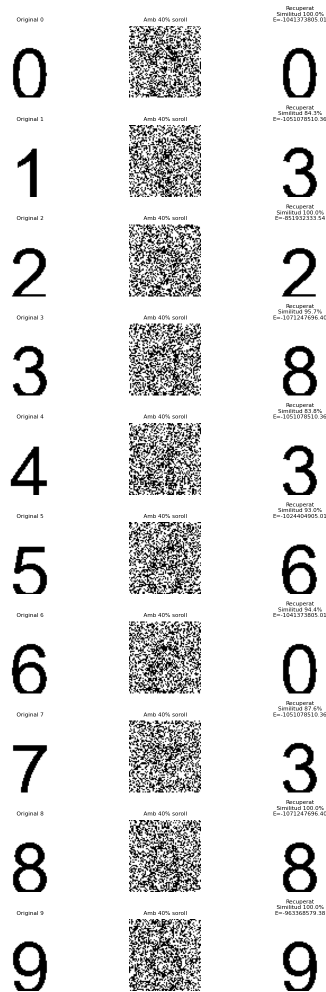


Figura 26: Recuperació dels dígitos del 0 al 9 amb alta definició amb l'energia exponencial i 40% de soroll.

Aquests resultats reforcen la idea que els dígitos, tot i estar ben definits visualment, comparteixen moltes estructures geomètriques entre ells (com barres verticals, corbes centrals o simetries verticals). Això que provoca que les recuperacions amb xarxes de Hopfield amb aquest tipus de patrons no sigui tan efectiva com amb patrons diferenciats.

11.2 Recuperació amb patrons ben diferenciats

En aquesta segona prova s'ha utilitzat el conjunt de 16 patrons clarament diferenciats que ja havien estat emprats anteriorment per estudiar el comportament de la regla de Hebb. A diferència dels dígitos, aquests patrons estan dissenyats per tenir formes visuals i estructures molt diferenciades entre si, amb l'objectiu de minimitzar la seva interferència interna dins la xarxa.

Amb un 20% de soroll binari afegit, la recuperació ha estat perfecta en tots els casos. La dinàmica associada a l'energia exponencial ha permès recuperar els patrons originals amb una precisió absoluta, sense desviacions ni formes espúries.

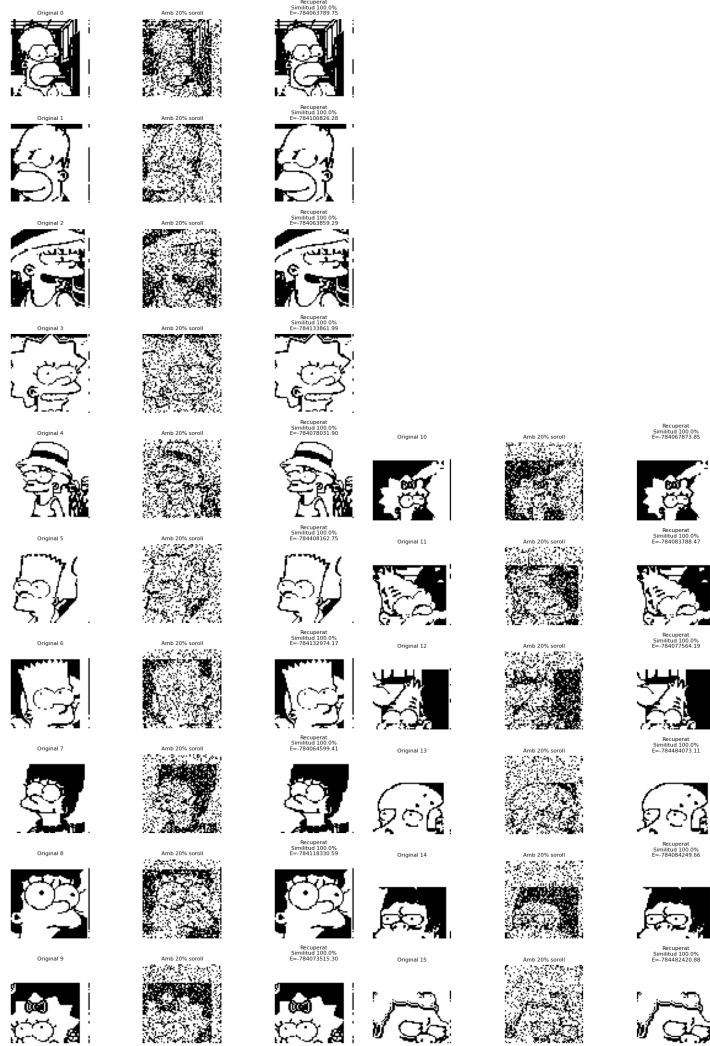


Figura 27: Recuperació de patrons ben diferenciats amb l'energia exponencial i 20% de soroll

Quan el nivell de soroll augmenta fins al 40%, la qualitat de la recuperació es manté sorprenentment intacta: tots els patrons es reconstrueixen de forma exacta, mantenint tant l'estructura com la seva identitat visual original. Aquest resultat contrasta amb el comportament observat amb els dígit, on el soroll elevat provocava confusions entre patrons. En aquest cas, però, la gran distància energètica entre els estats emmagatzemats evita que la dinàmica convergeixi cap a solucions incorrectes.

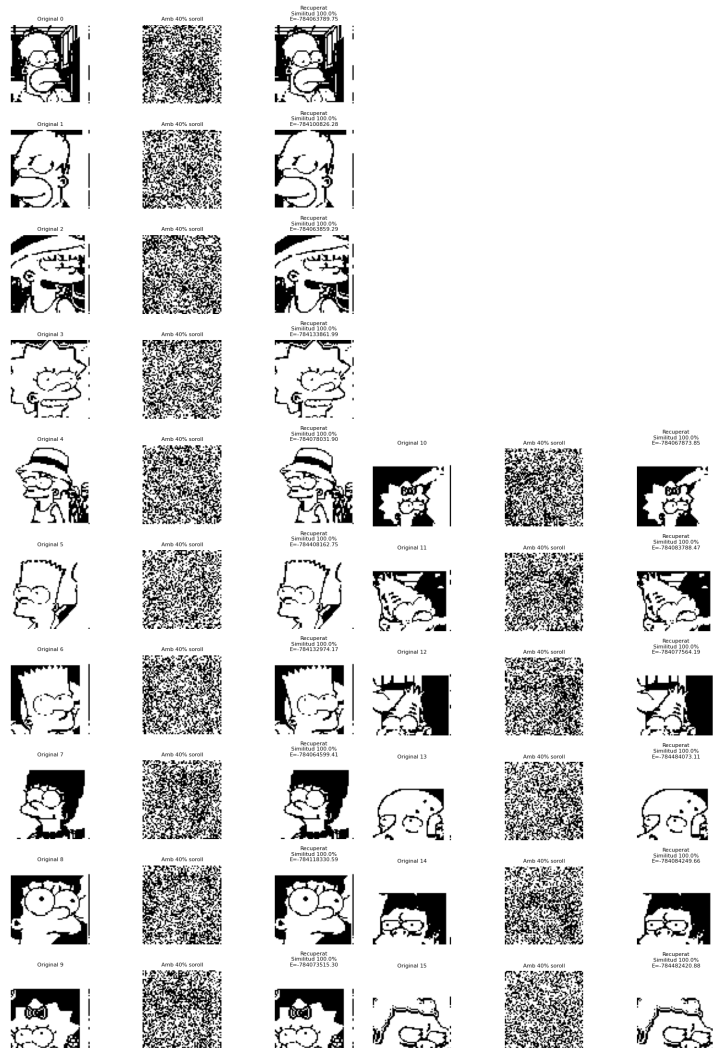


Figura 28: Recuperació de patrons ben diferenciats amb l'energia exponencial i 40% de soroll.

Aquests resultats posen de manifest la importància de la dissimilitud entre patrons per garantir una bona recuperació. Com més diferents són entre si, més profunds i separats són els seus atractors en l'espai energètic, i per tant, més robusta és la convergència del sistema. En aquest escenari, l'energia exponencial demostra la seva capacitat per mantenir intacta la qualitat dels patrons fins i tot en condicions de soroll elevat.

12 Test de recuperació amb la meitat del patró en negre

Fins ara, totes les proves de recuperació s'han centrat en la capacitat de la xarxa per restaurar patrons alterats per soroll additiu —és a dir, inversions aleatòries d'un percentatge de píxels. En aquesta nova secció s'examina un escenari molt més exigent: la recuperació a partir de patrons amb una part substancial completament esborrada. Concretament, s'ha substituït tota la meitat esquerra de cada imatge per negre (valor -1 en bipolar), deixant només la part dreta com a entrada a la xarxa.

L'objectiu és analitzar com es comporten dues aproximacions diferents —la pseudoinversa i l'energia exponencial— davant aquest tipus de pèrdua d'informació sistemàtica i estructural, no s'utilitza la regla de Hebb ja que les recuperacions menys exigents no les recuperava amb èxit, llavors queda descartada. Aquesta prova posa a l'extrem la capacitat de generalització de la xarxa, i permet detectar si és capaç de completar patrons a partir d'una part significativa de la seva estructura. S'utilitzaran els 16 patrons ben diferenciats.

12.1 Recuperació amb la pseudoinversa

Els resultats obtinguts amb la pseudoinversa són, en general, decebedors. Tot i que aquesta regla havia demostrat una gran precisió en situacions de soroll, en aquest cas la seva resposta és inadequada: la majoria de patrons recuperats són visualment deformats, alguns fins al punt de no poder-se identificar clarament amb cap dels patrons originals.

Això indica que la pseudoinversa —basada en una aproximació algebraica òptima en mitjana— no és prou flexible davant pèrdues de components estructurals senceres. En absència de la part esquerra, la xarxa no troba prou informació per reconstruir el conjunt del patró de manera fidel, i la dinàmica tendeix a generar combinacions incorrectes de trets memoritzats. A més, es detecta un efecte de saturació energètica: molts dels patrons recuperats acaben amb una energia idèntica ($E = -2040.00$), suggerint un col·lapse cap a estats límit amb poca capacitat de diferenciació interna.



Figura 29: Recuperació de patrons ben diferenciats amb la pseudoinversa amb la meitat esquerra negra.

12.2 Recuperació amb l'energia exponencial

Quan es repeteix la mateixa prova utilitzant la formulació d'energia exponencial, els resultats canvien radicalment. La xarxa és capaç de reconstruir correctament la majoria dels patrons, malgrat haver rebut només la meitat dreta com a estímul. Tant la forma global com els detalls dels patrons originals es restauren amb una precisió sorprenent, assolint similituds del 100% en molts casos (13 de 16).

Aquest comportament suggereix que l'energia exponencial actua com un model molt més discriminatiu i tolerant a la pèrdua parcial d'informació. En els casos on no s'assoleix la recuperació del original, es recupera un altre patró dels originals, és a dir, amb l'energia exponencial sempre s'obté a un dels patrons originals com a resultat de la recuperació.

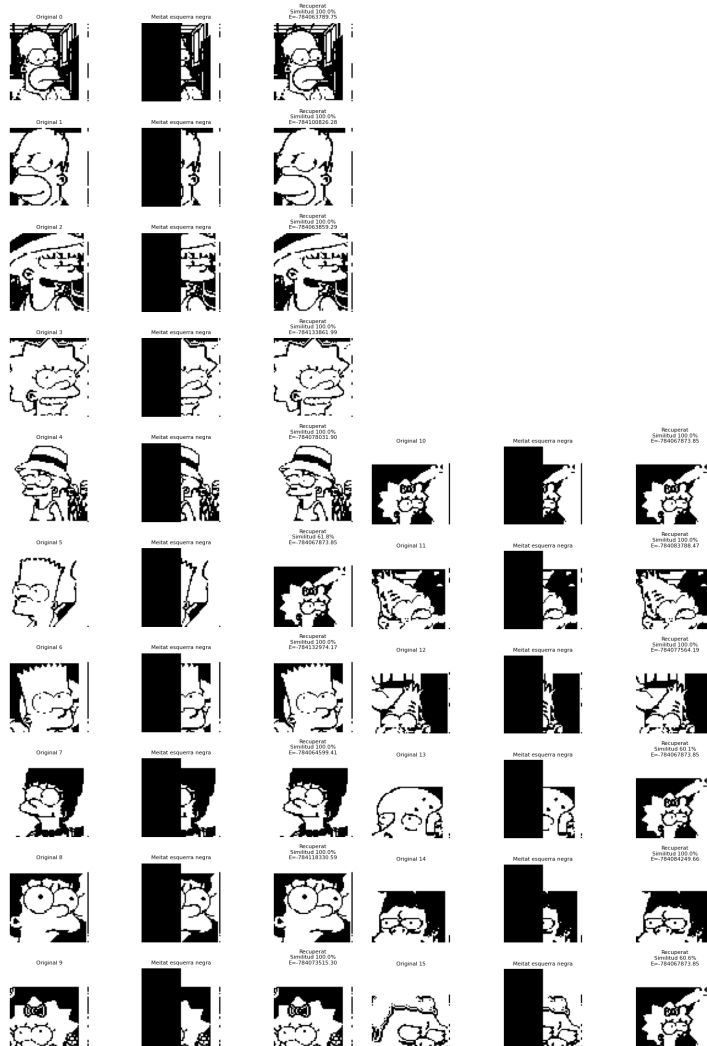


Figura 30: Recuperació de patrons ben diferenciats amb l'energia exponencial amb la meitat esquerra negra.

Conclusió

Aquest experiment permet extreure conclusions valuoses. Primer, es confirma que la pseudoinversa és molt eficaç amb soroll dispers, però no pot recuperar patrons quan una part significativa de l'entrada és totalment nul·la. En canvi, la formulació energètica exponencial és molt més robusta davant pèrdues sistemàtiques: la convergència cap a patrons originals es manté estable fins i tot amb només la meitat de la informació.

Aquestes observacions reforcen la idea que l'energia exponencial no només augmenta la capacitat d'emmagatzematge, sinó que també dota la xarxa d'una millor capacitat de completació estructural. Això obre la porta a aplicacions més exigents, on calgui reconstruir imatges o patrons a partir de fragments parcials, com en el cas de dades incompletes o censurades.

La taula següent recull els valors de similitud percentual obtinguts per a cadascun dels 16 patrons diferenciats, comparant els resultats de la recuperació amb la pseudoinversa i amb la funció d'energia exponencial després d'ocultar completament la meitat esquerra de cada imatge.

Patró	Similitud Pseudoinversa (%)	Similitud Exponencial (%)
1	67.4	100.0
2	49.0	100.0
3	47.2	100.0
4	34.4	100.0
5	40.8	100.0
6	38.0	61.8
7	41.2	100.0
8	38.6	100.0
9	39.2	100.0
10	41.8	100.0
11	33.3	100.0
12	34.6	100.0
13	39.6	60.1
14	43.1	100.0
15	40.2	100.0
16	38.8	60.6

Taula 3: Comparativa de la similitud en la recuperació dels 16 patrons parcialment ocults, utilitzant la pseudoinversa i l'energia exponencial.

13 Conclusions

Aquest treball ha dut a terme una anàlisi exhaustiva de les xarxes de Hopfield, tant en la seva formulació clàssica com en variants modernes, per a la recuperació de patrons binaris. A través de diversos experiments, s'han avaluat els límits, capacitats i comportaments d'aquestes xarxes en escenaris amb diferents nivells de complexitat, soroll i dimensionalitat.

Inicialment, s'ha verificat que la regla de Hebb, tot i la seva simplicitat i interès històric, presenta limitacions severes en entorns realistes. Quan els patrons a emmagatzemar són visualment similars —com en el cas dels dígitos o les lletres majúscules—, la dinàmica de la xarxa tendeix a col·lapsar en atractors espuris, sovint dominats per trets compartits entre els patrons. Aquest fenomen posa de manifest la manca de discriminació de la regla de Hebb en espais amb correlacions internes elevades.

Tanmateix, fins i tot en condicions aparentment favorables —amb patrons clarament diferenciats i no correlacionats—, la recuperació amb la regla de Hebb continua sent problemàtica: la xarxa mostra una tendència recurrent a convergir cap a un únic atractor espuri, independentment del patró inicial. Aquest resultat contradiu les expectatives teòriques i reforça la idea que, a la pràctica, la regla de Hebb no garanteix estabilitat ni tan sols en escenaris òptims.

En contrast, l'ús de la pseudoinversa ha permès obtenir una recuperació gairebé perfecta en la majoria de situacions, tant amb activació síncrona com asíncrona. Aquesta regla ofereix una aproximació algebraicament òptima al problema de la memòria associativa, eliminant les interaccions no desitjades entre patrons. Això es tradueix en atractors ben definits, capaços de resistir nivells considerables de soroll i deformació.

Tanmateix, en escenaris d'alteració extrema —com la ocultació completa de mitja imatge—, fins i tot la pseudoinversa mostra signes de debilitament: en alguns casos, la recuperació resulta incompleta o deriva cap a deformacions parcialment reconeixibles. És en aquest context on pren especial rellevància la introducció de l'energia exponencial.

Mitjançant aquesta nova energia, s'ha aconseguit una recuperació sorprenentment robusta, fins i tot davant d'alteracions severes. El sistema no només evita els atractors espuris, sinó que, en molts casos, és capaç de reconstruir patrons a partir d'informació parcial extrema. Aquesta estabilitat es deu al fet que l'energia exponencial genera paisatges energètics amb mínims més pronunciats i selectius, que reforcen l'atracció cap als patrons originals.

Com a síntesi, aquest treball conclou que:

- La regla de Hebb no és fiable per a la recuperació de patrons, ni tan sols en condicions ideals amb patrons clarament diferenciats.
- La pseudoinversa representa una millora substancial en termes de capacitat, estabilitat i absència d'atractors espuris, tot i que no és invulnerable a danys extrems.
- La funció d'energia exponencial constitueix una de les alternatives més prometedores, especialment en situacions amb poca informació o amb danys greus als patrons.

Aquestes conclusions obren la porta a futures línies de recerca, com ara l'aplicació d'energies alternatives (polinòmiques, combinades), l'estudi formal del paisatge energètic d'aquestes xarxes, o la seva implementació en entorns neuromòrfics per aplicacions reals de recuperació de memòria associativa.

Referències

- [1] M. Ramsauer et al., *Hopfield Layers is All You Need*, Institut de Machine Learning, Johannes Kepler University, 2021. https://ml-jku.github.io/hopfield-layers/#mjax-eqn-eqrestorage_demircigil
- [2] D. Krotov and J. J. Hopfield, *Dense Associative Memory for Pattern Recognition*, *Advances in Neural Information Processing Systems*, 29, 2016. <https://arxiv.org/abs/1606.01164>
- [3] M. Demircigil, J. Heusel, M. Löwe, S. Upgang, and F. Vermet, *On a model of associative memory with huge storage capacity*, *Journal of Statistical Physics*, 168(2), 288–299, 2017. <https://arxiv.org/abs/1702.01929>
- [4] J. J. Hopfield, *Neural networks and physical systems with emergent collective computational abilities*, *Proceedings of the National Academy of Sciences*, 79(8), 2554–2558, 1982. <https://www.pnas.org/doi/10.1073/pnas.79.8.2554>
- [5] R. Meng, *Modern Hopfield Networks*, 2021. <https://modernhopfield.com>
- [6] H. Ramsauer et al., *Hopfield Networks is All You Need*, *International Conference on Learning Representations*, 2021. <https://arxiv.org/abs/2008.02217>
- [7] D. Krotov and J. J. Hopfield, *Dense Associative Memory is Robust to Adversarial Inputs*, *arXiv preprint*, arXiv:1701.00939, 2017. <https://arxiv.org/abs/1701.00939>
- [8] D. Krotov and J. J. Hopfield, *Large Associative Memory Problem in Neurobiology and Machine Learning*, *arXiv preprint*, arXiv:2008.06996, 2020. <https://arxiv.org/abs/2008.06996>
- [9] *Modern Hopfield Network*, *Wikipedia*, 2024. https://en.wikipedia.org/wiki/Modern_Hopfield_network
- [10] *Hopfield Network*, *Wikipedia*, 2024. https://en.wikipedia.org/wiki/Hopfield_network
- [11] D. J. Amit, H. Gutfreund, and H. Sompolinsky, *Storing Infinite Numbers of Patterns in a Spin-Glass Model of Neural Networks*, *Physical Review Letters*, 55(14), 1985. <https://doi.org/10.1103/PhysRevLett.55.1530>
- [12] D. J. Amit, *Modeling Brain Function: The World of Attractor Neural Networks*, Cambridge University Press, 1989.
- [13] R. Rojas, *Neural Networks: A Systematic Introduction*, Springer-Verlag, 1996. <https://doi.org/10.1007/978-3-642-61068-4>

Apèndix A: Recuperacions de patrons

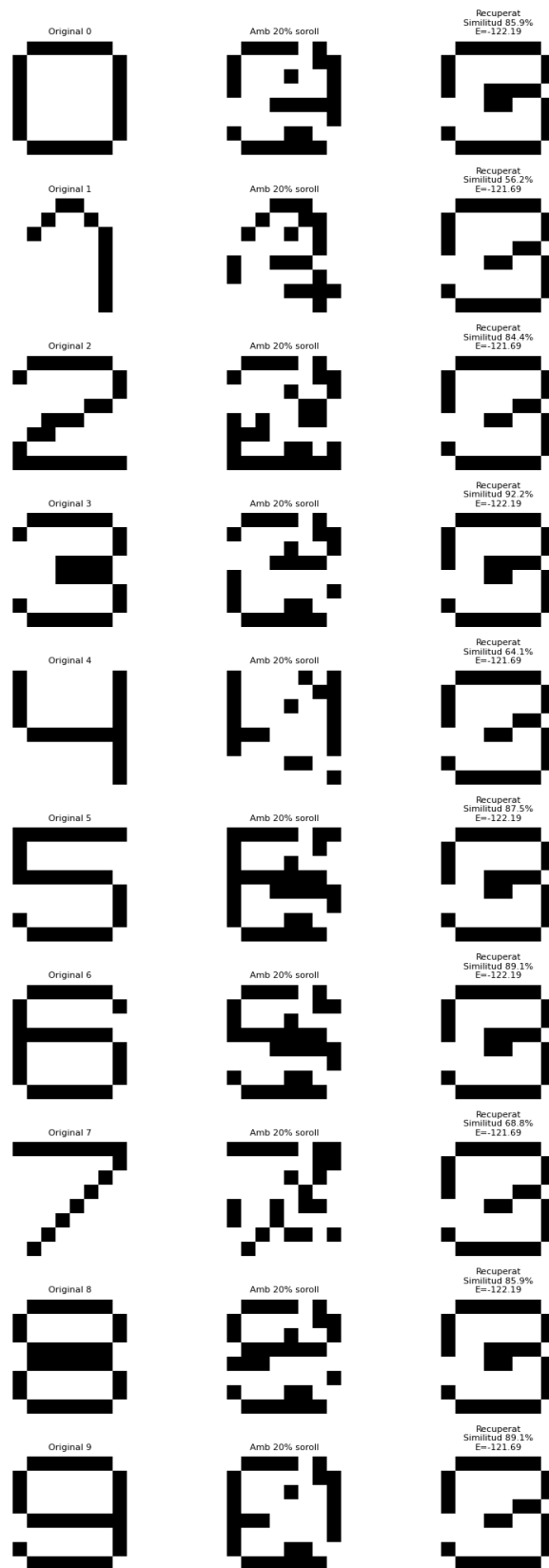


Figura 31: Recuperació amb la regla de Hebb amb un 20% de soroll

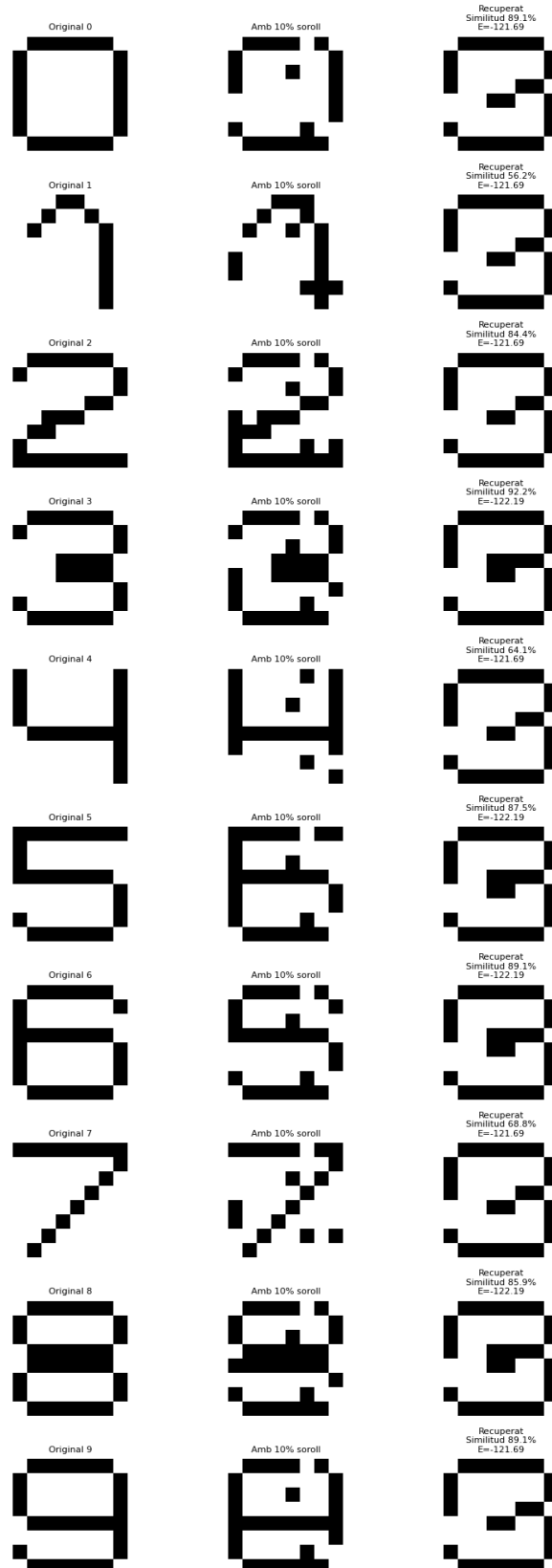


Figura 32: Recuperació amb la regla de Hebb amb un 10% de soroll

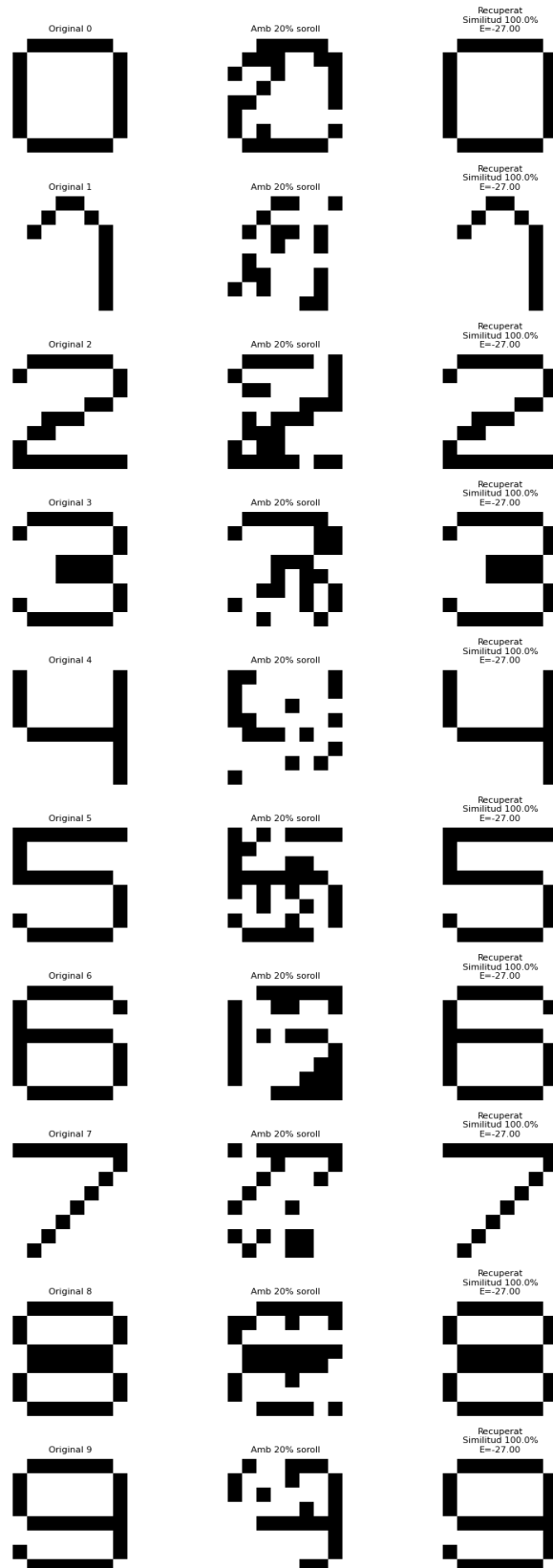


Figura 33: Recuperació amb la pseudoinversa i actualització síncrona, soroll del 20%.

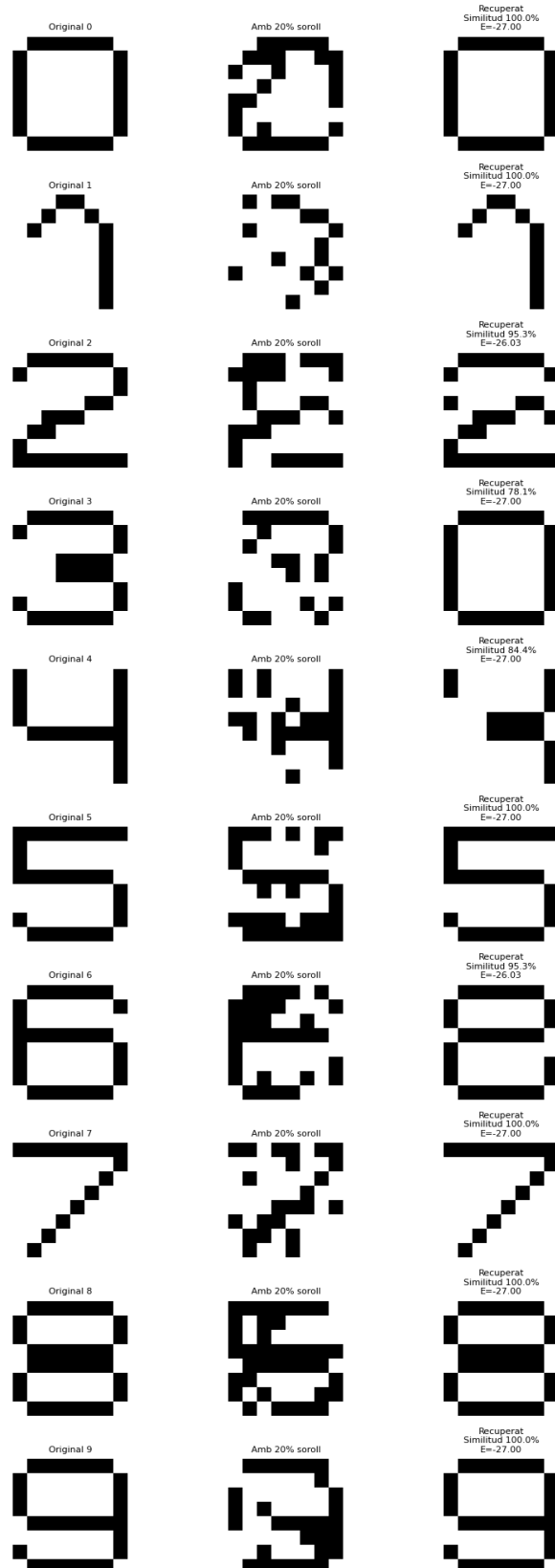


Figura 34: Recuperació amb la pseudoinversa i actualització asíncrona, soroll del 20%.

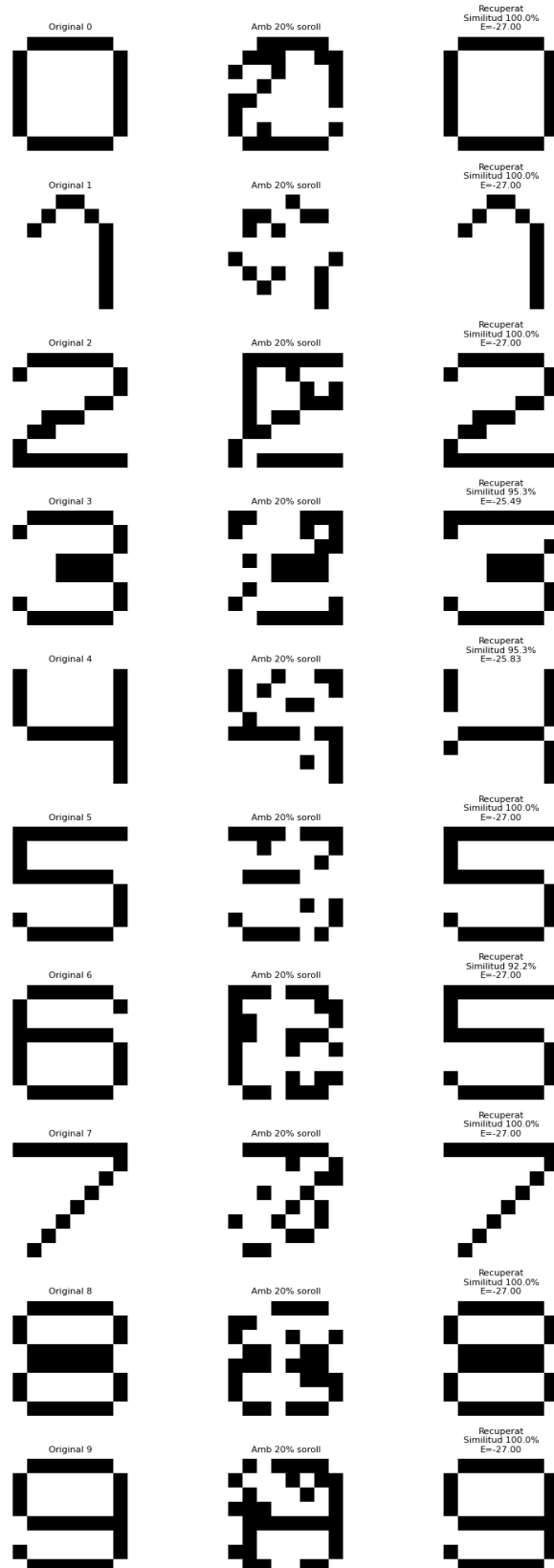


Figura 35: Recuperació amb la pseudoinversa i actualització asíncrona amb més iteracions.

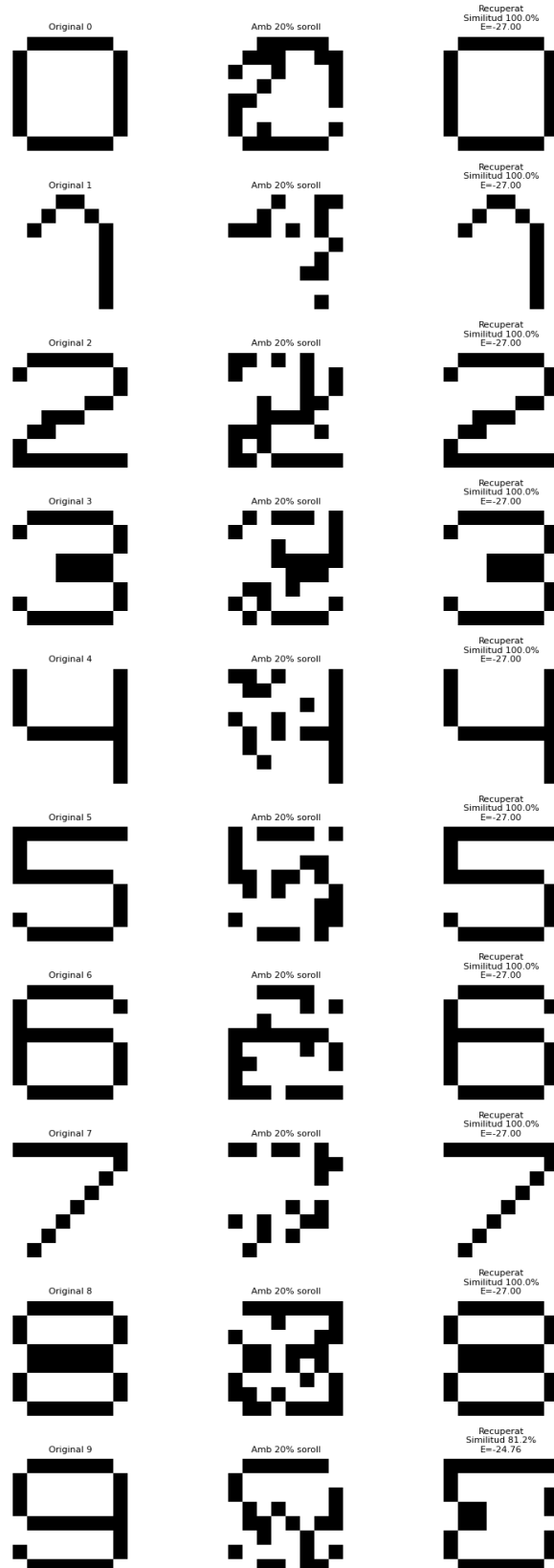


Figura 36: Recuperació amb la pseudoinversa i actualització asíncrona amb més iteracions.
(2)

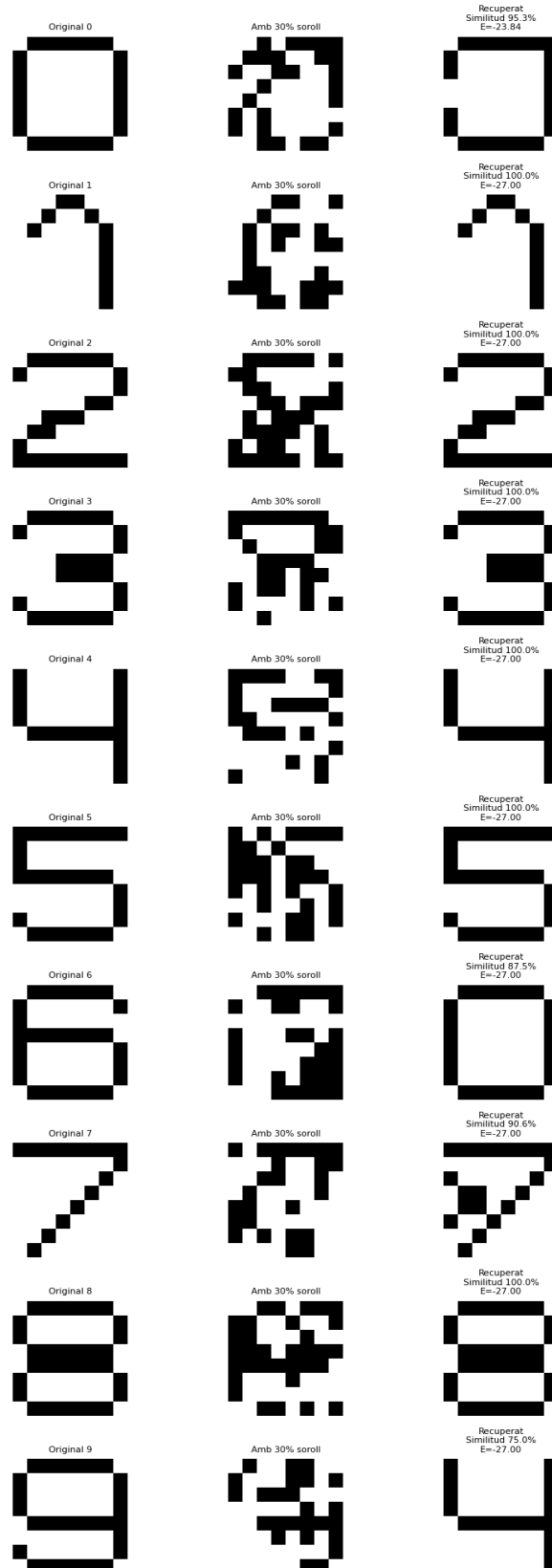


Figura 37: Recuperació amb pseudoinversa i activació síncrona amb 30% de soroll.

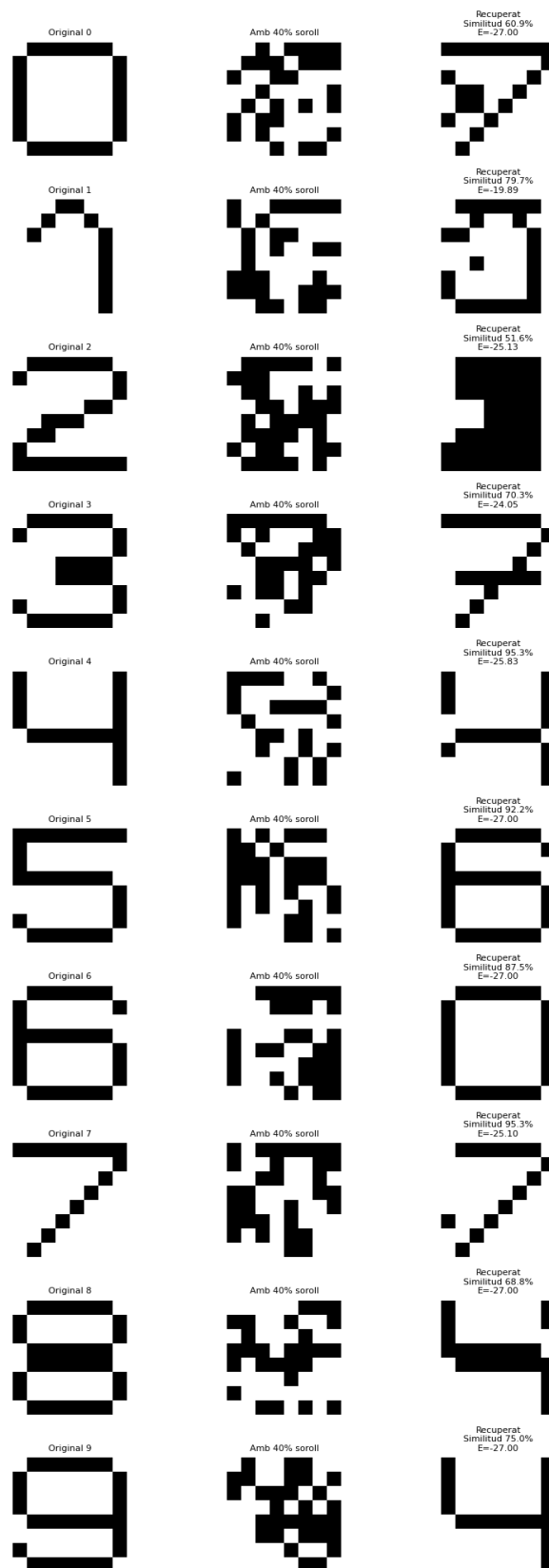


Figura 38: Recuperació amb pseudoinversa i activació síncrona amb 40% de soroll.

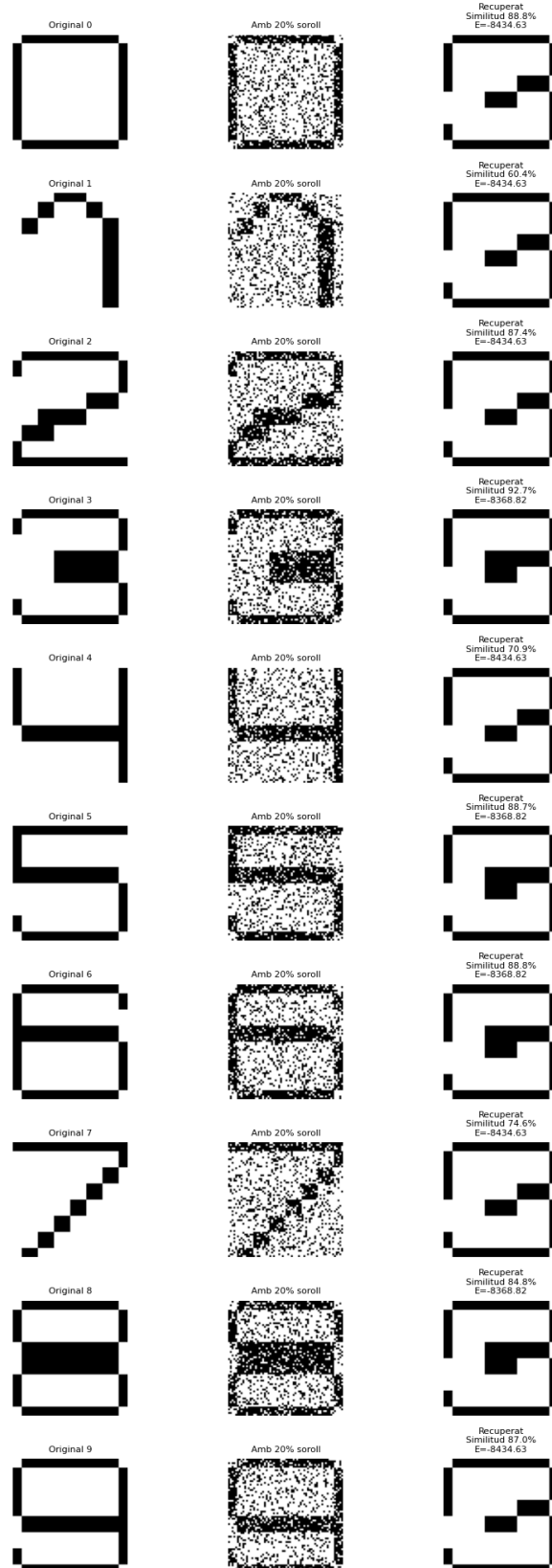


Figura 39: Recuperació amb la regla de Hebb amb un 20% de soroll

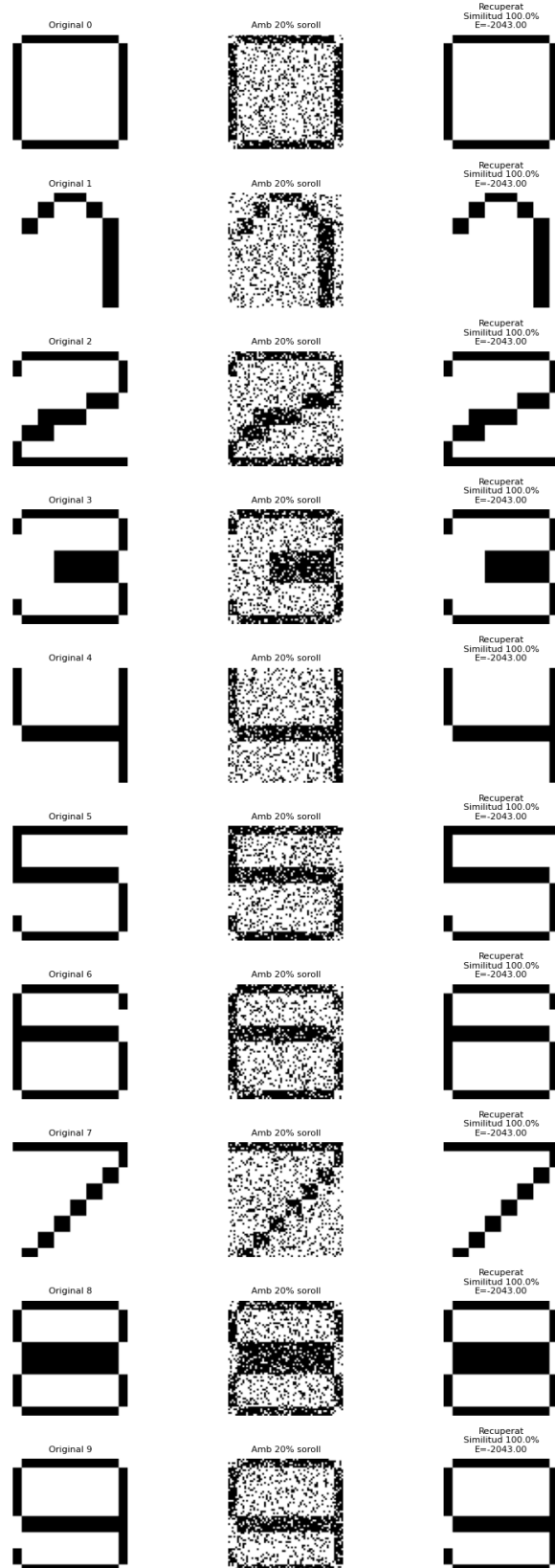


Figura 40: Recuperació amb la pseudoinversa i activació síncrona amb un 20% de soroll

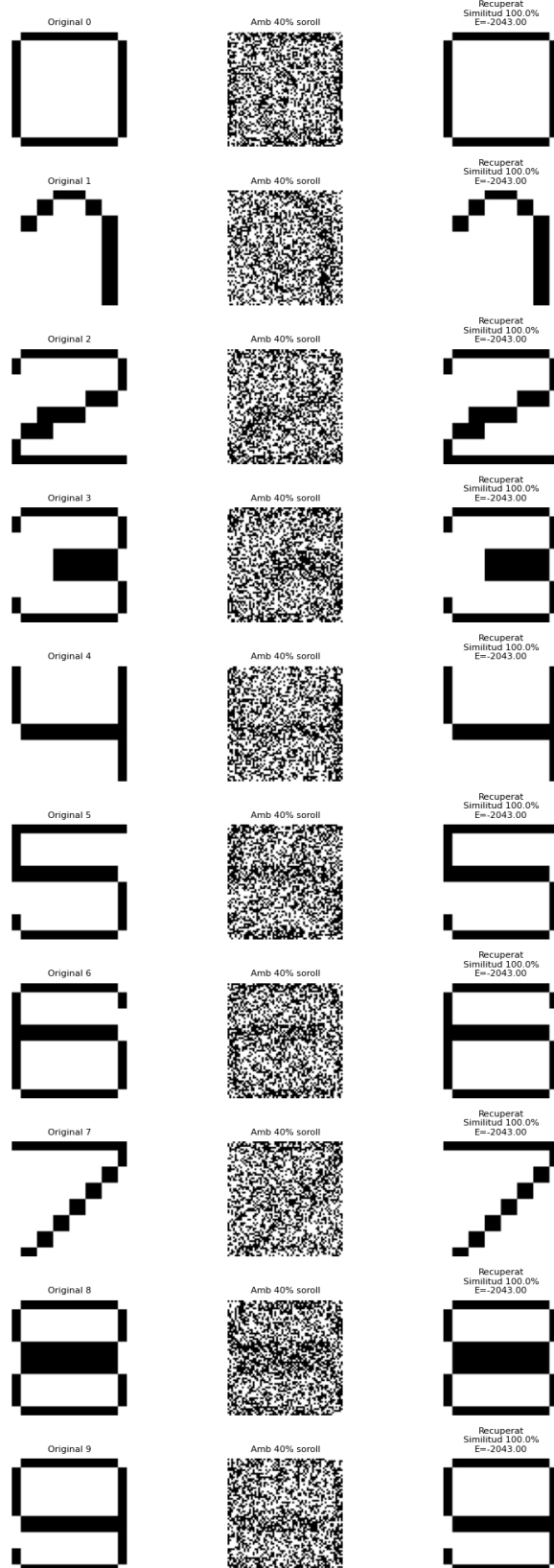


Figura 41: Recuperació amb pseudoinversa i activació síncrona amb 40% de soroll.

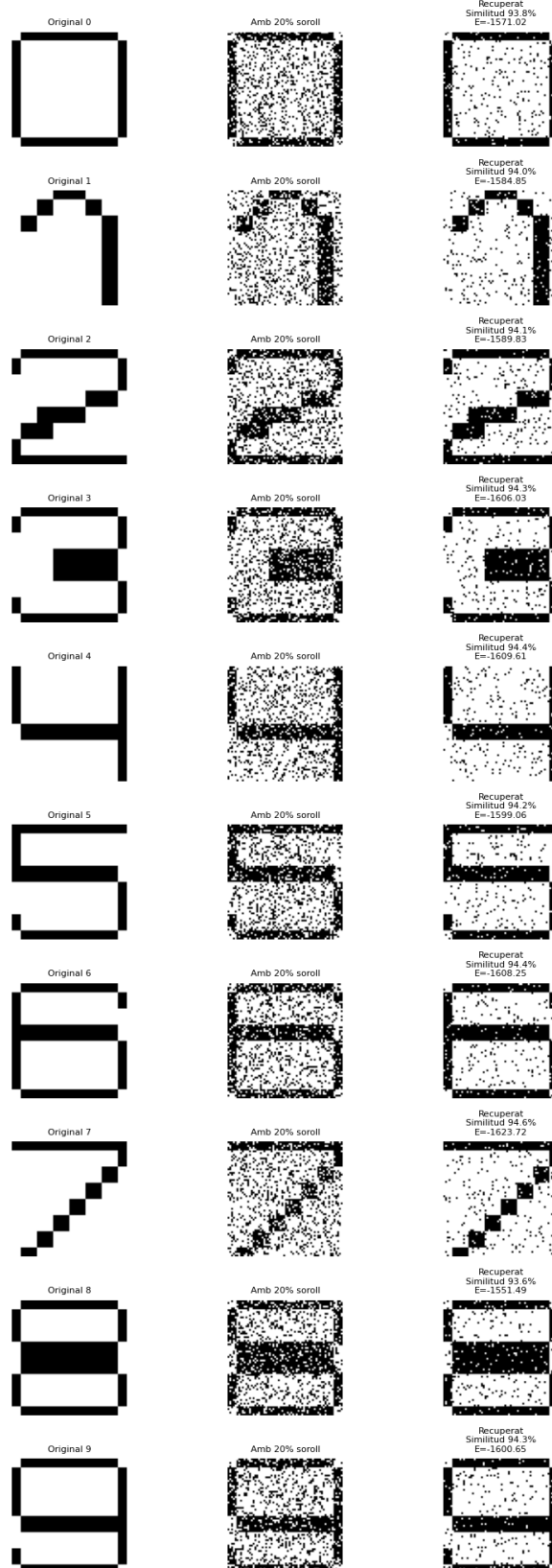


Figura 42: Recuperació amb la pseudoinversa i activació asíncrona amb un 20% de soroll

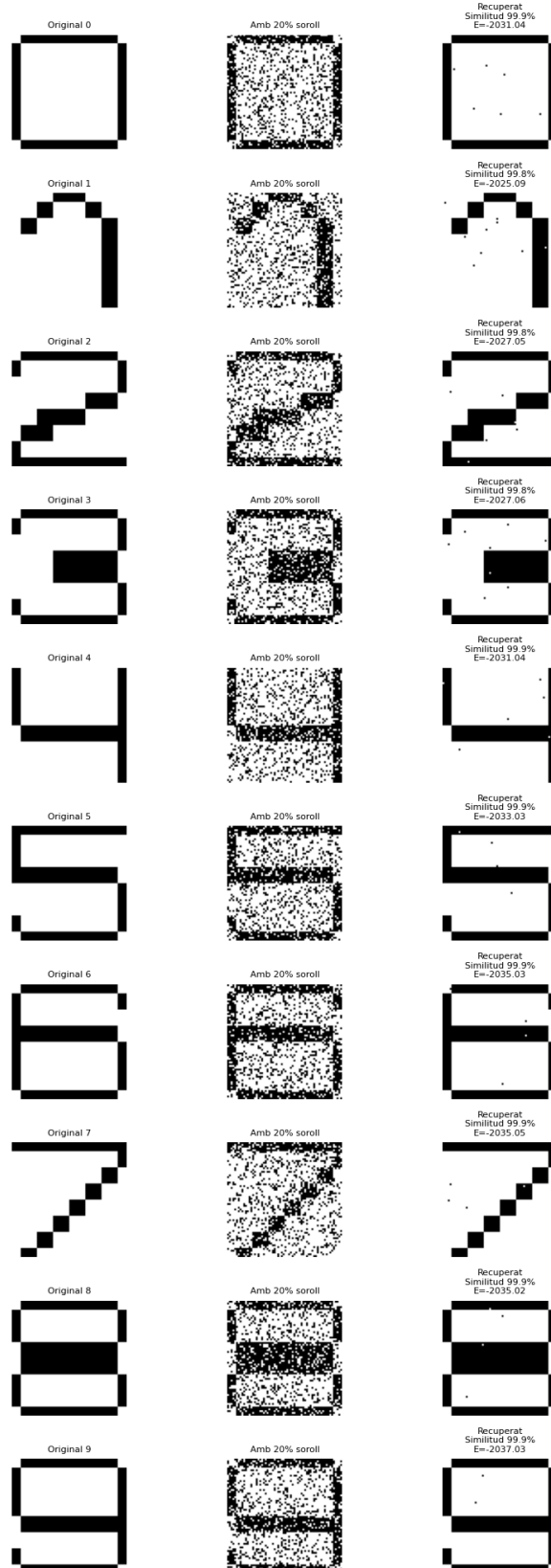


Figura 43: Recuperació amb la pseudoinversa i actualització asíncrona amb més iteracions.

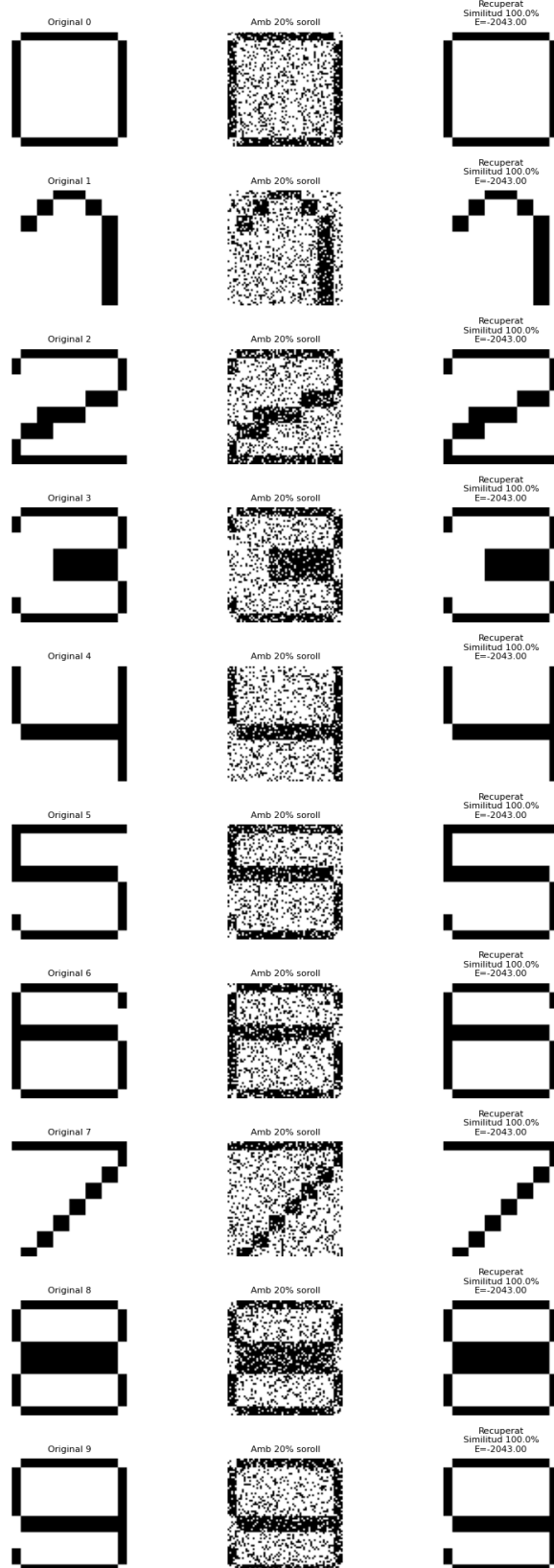


Figura 44: Recuperació amb la pseudoinversa i actualització asíncrona amb més iteracions.
(2)

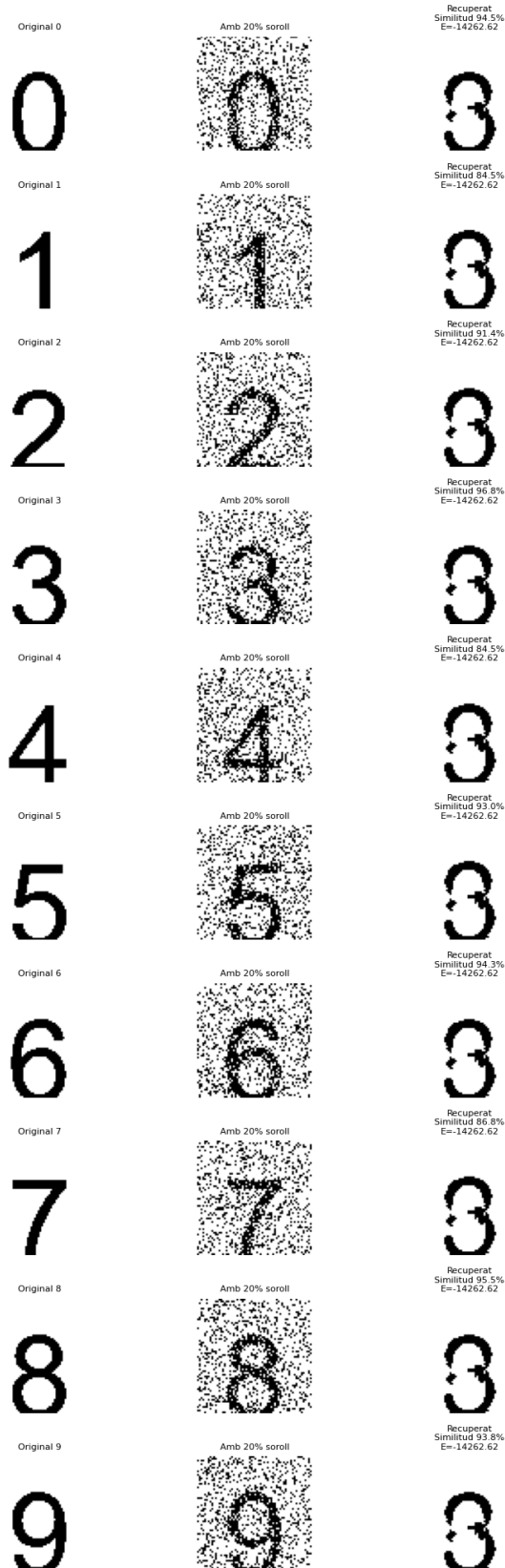


Figura 45: Recuperació amb regla de Hebb (dígit definit)

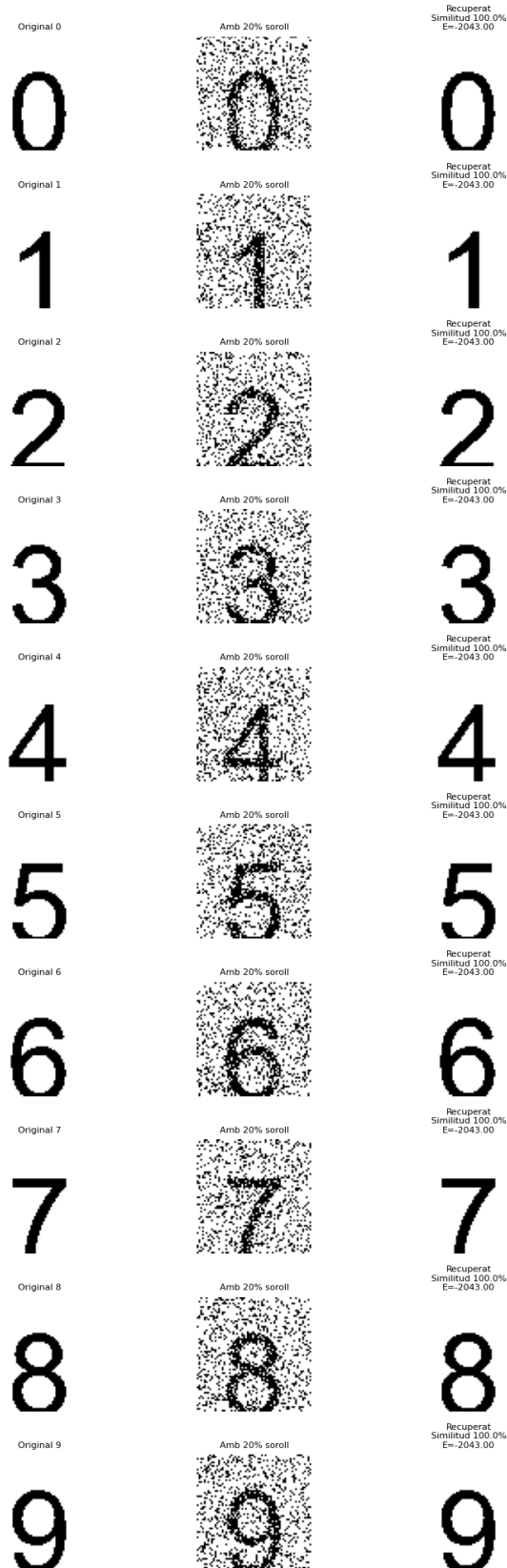


Figura 46: Recuperació amb pseudoinversa (dígit definit)

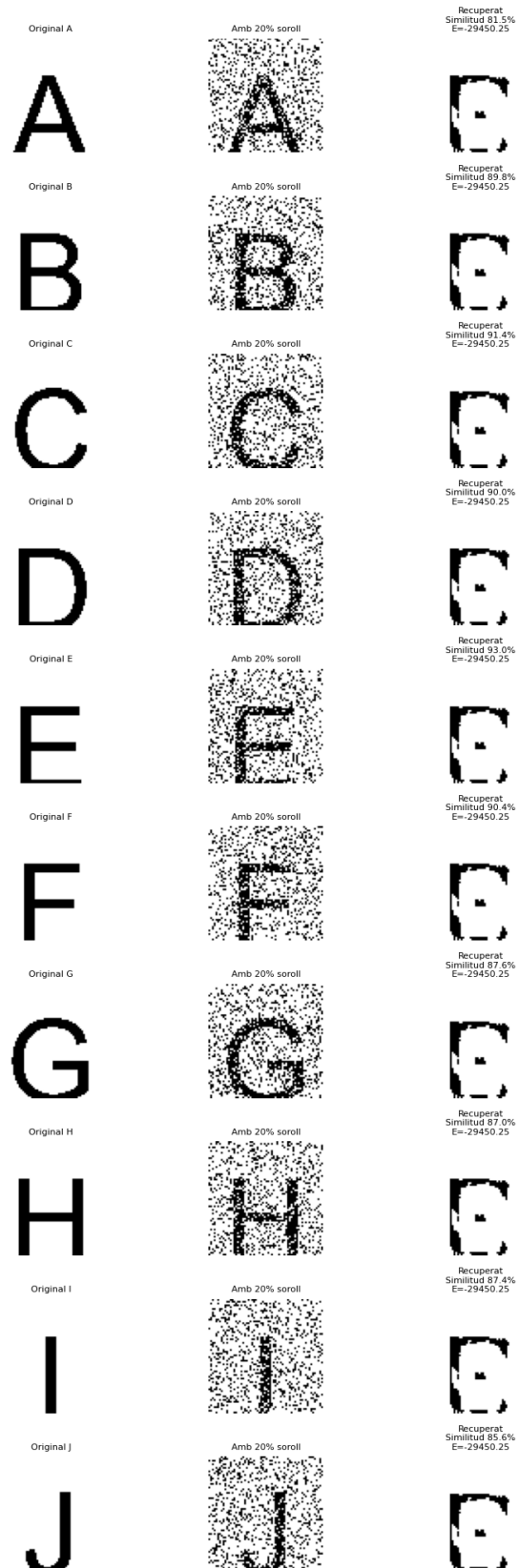


Figura 47: Recuperació de patrons de lletres amb la regla de Hebb. (1)

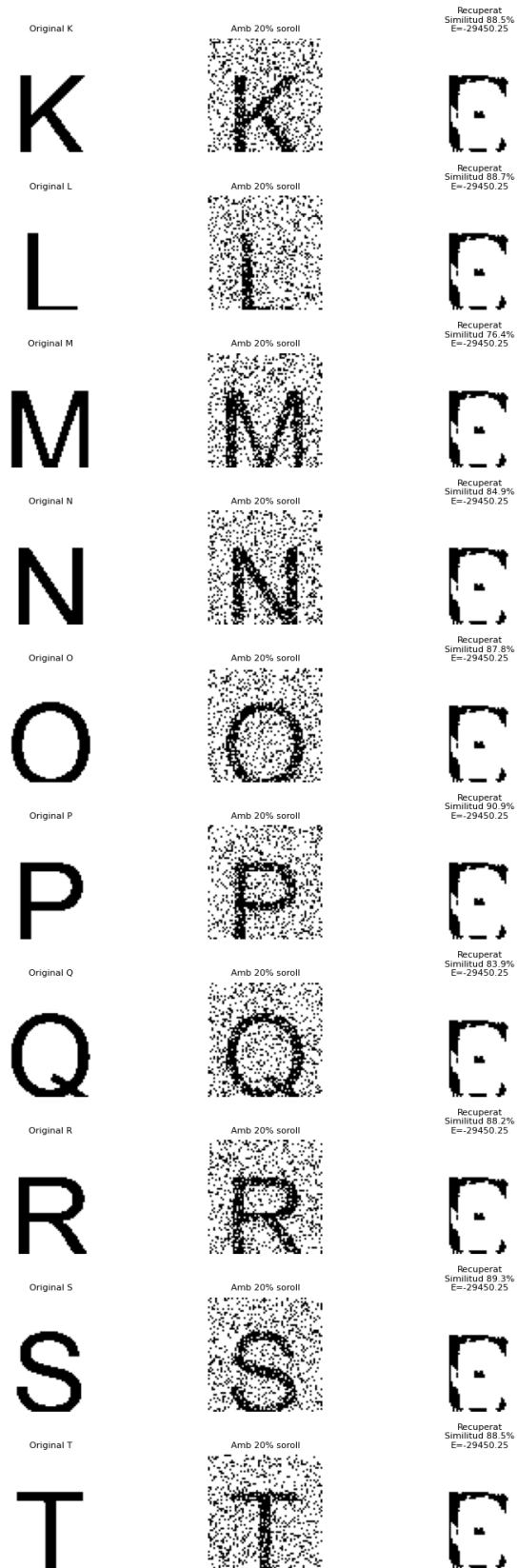


Figura 48: Recuperació de patrons de lletres amb la regla de Hebb. (2)

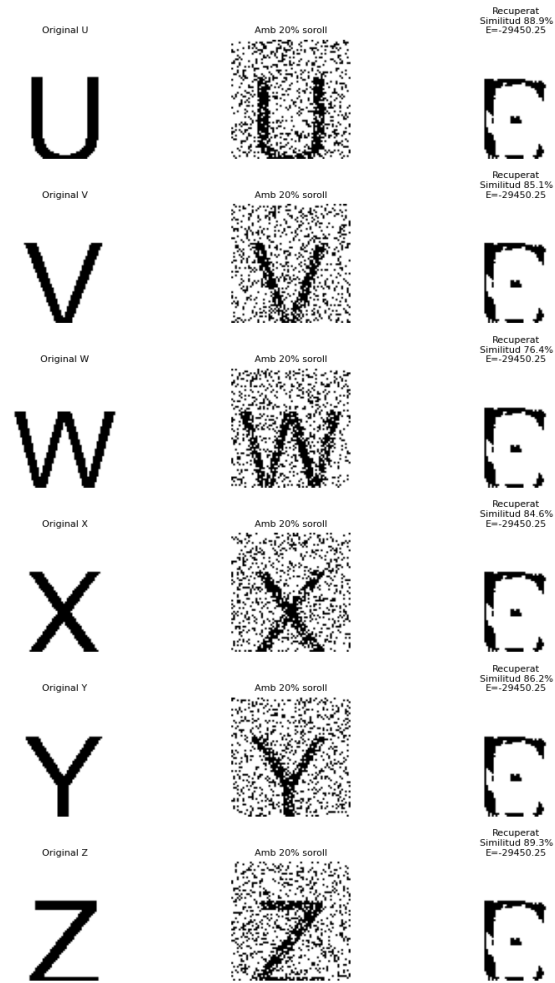


Figura 49: Recuperació de patrons de lletres amb la regla de Hebb. (3)

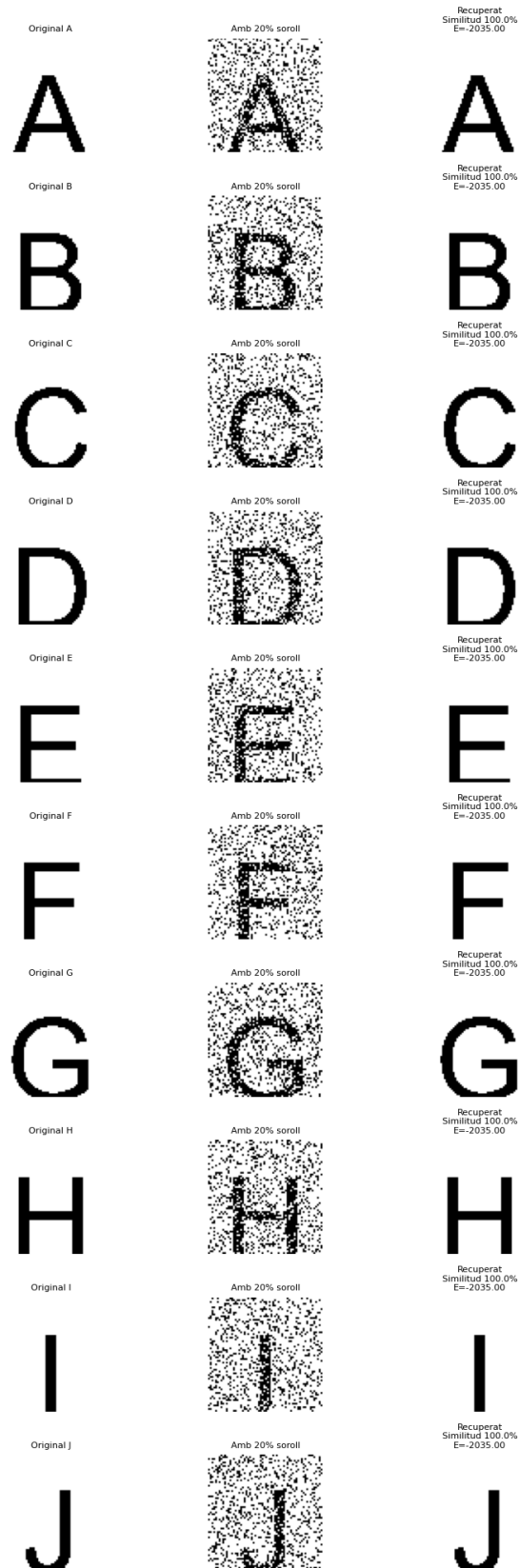


Figura 50: Recuperació de patrons de lletres amb la pseudoinversa. (1)



Figura 51: Recuperació de patrons de lletres amb la pseudoinversa. (2)

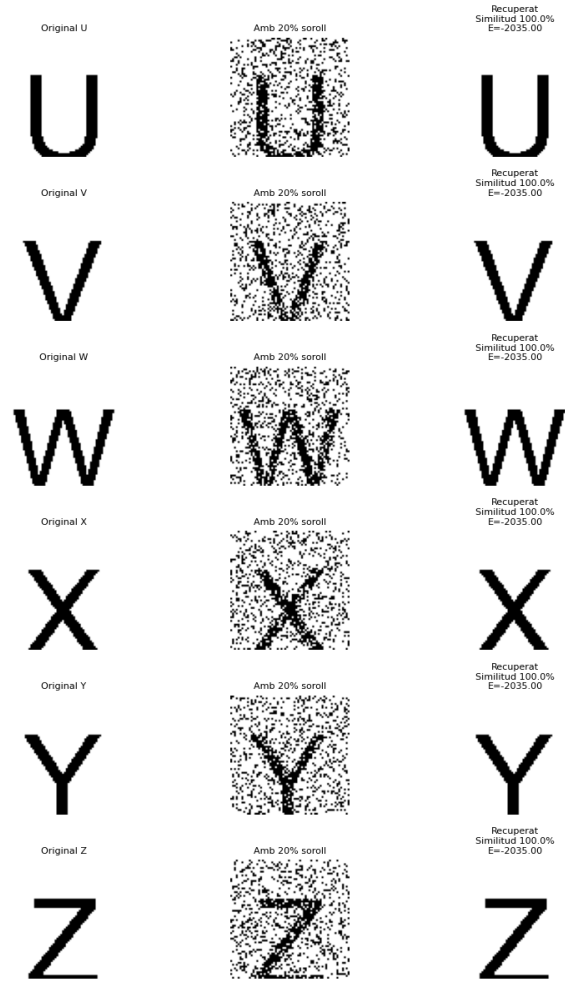


Figura 52: Recuperació de patrons de lletres amb la pseudoinversa. (3)

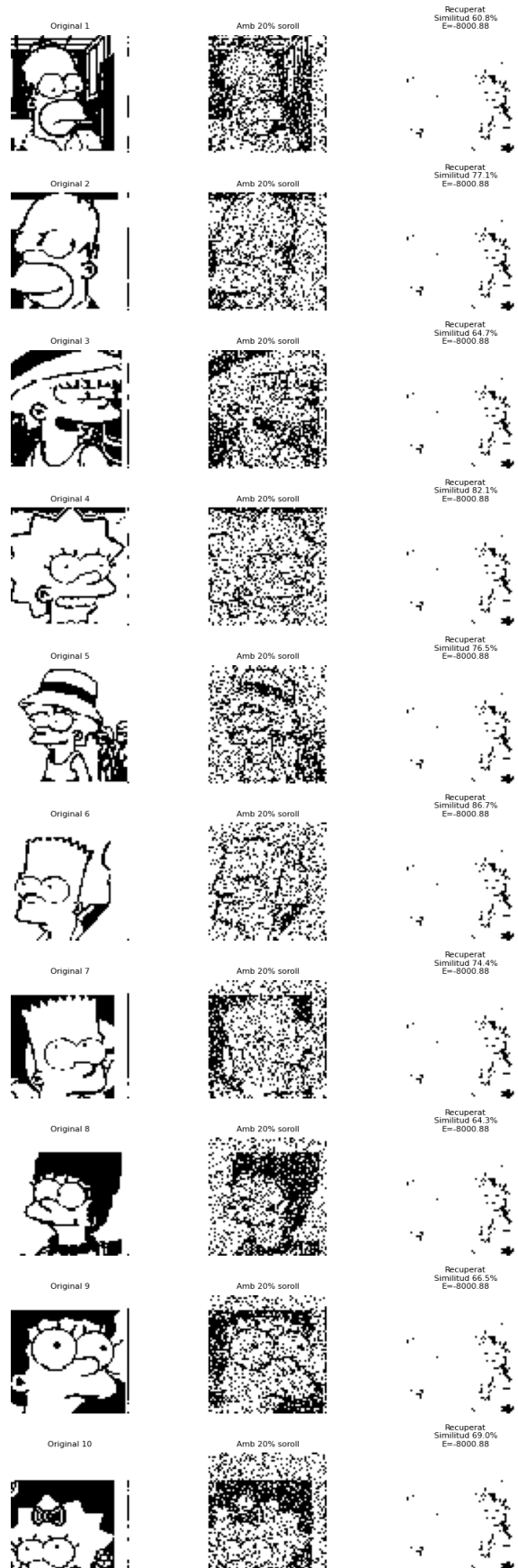


Figura 53: Recuperació de patrons diferenciats amb la regla de Hebb. (1)

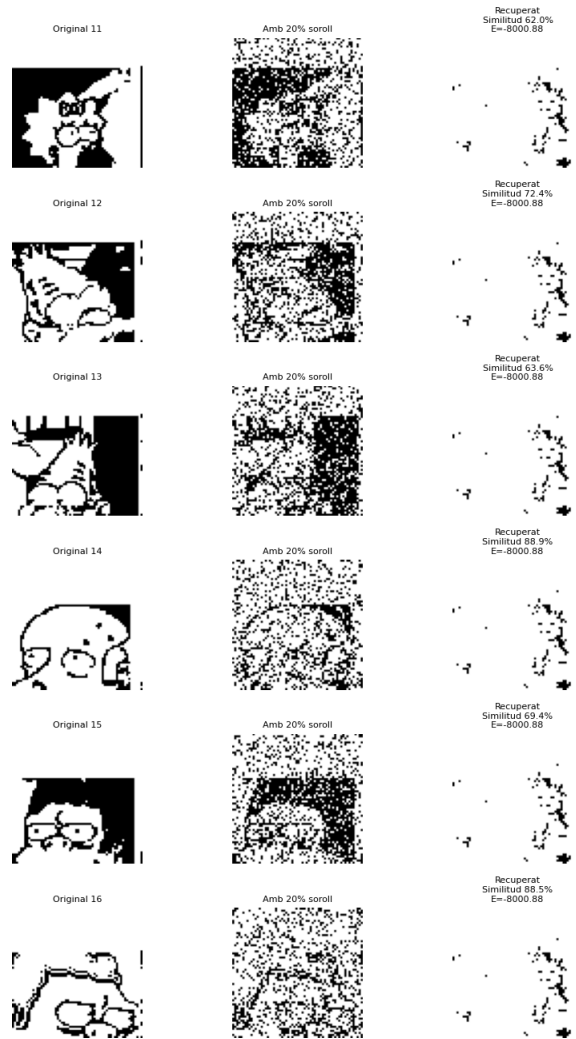


Figura 54: Recuperació de patrons diferenciats amb la regla de Hebb. (2)

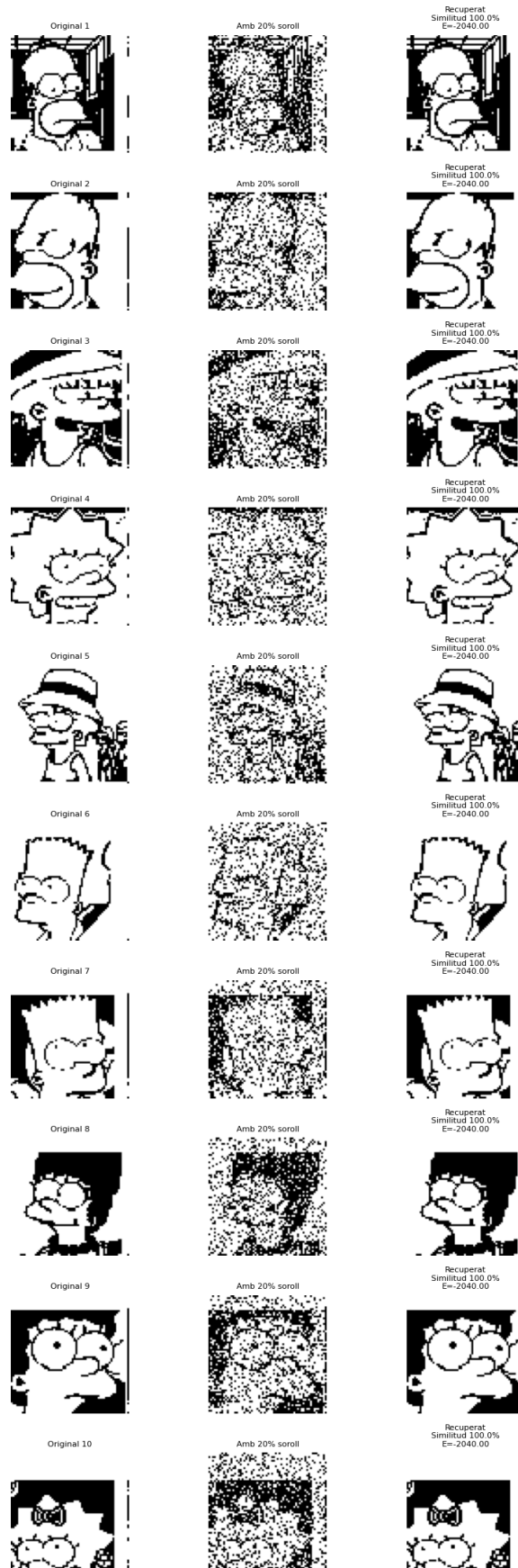


Figura 55: Recuperació de patrons diferenciats amb la pseudoinversa. (1)

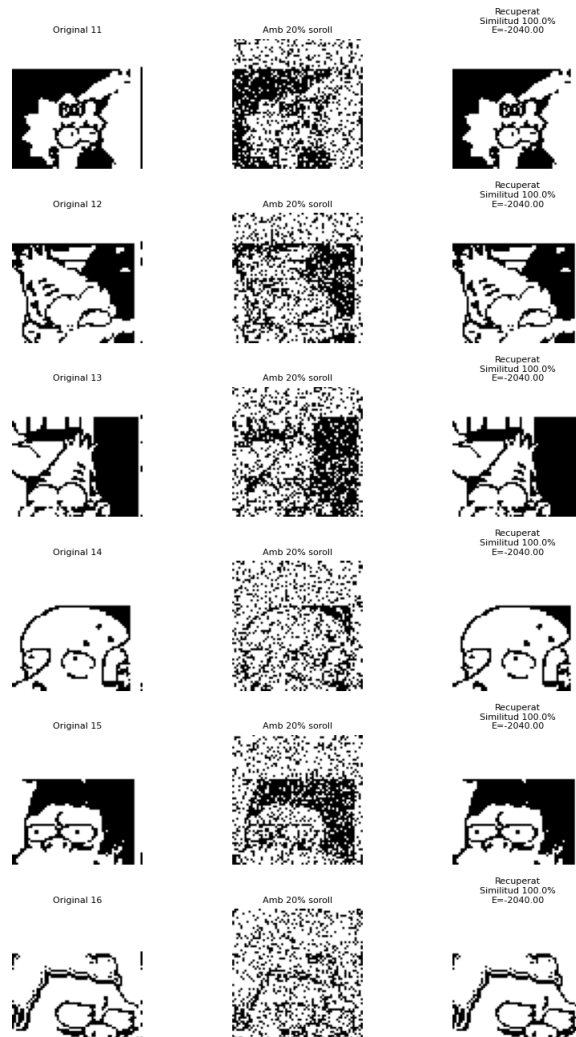


Figura 56: Recuperació de patrons diferenciats amb la pseudoinversa. (2)

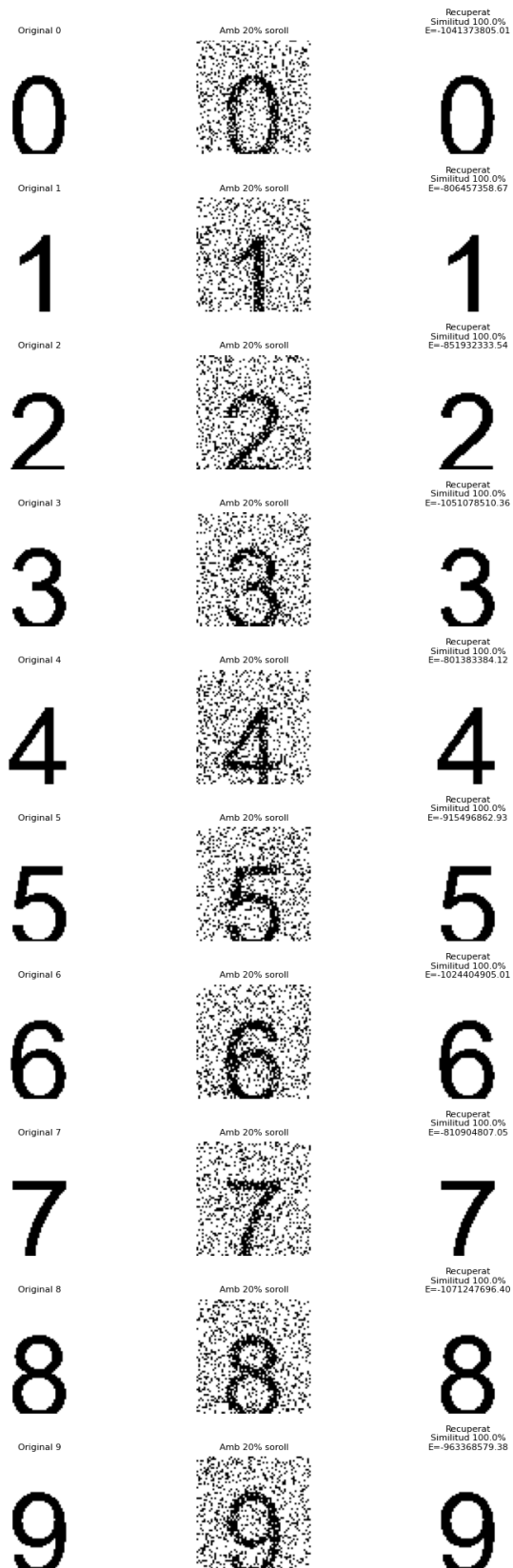


Figura 57: Recuperació dels dígets del 0 al 9 amb alta definició amb l'energia exponencial i 20% de soroll.

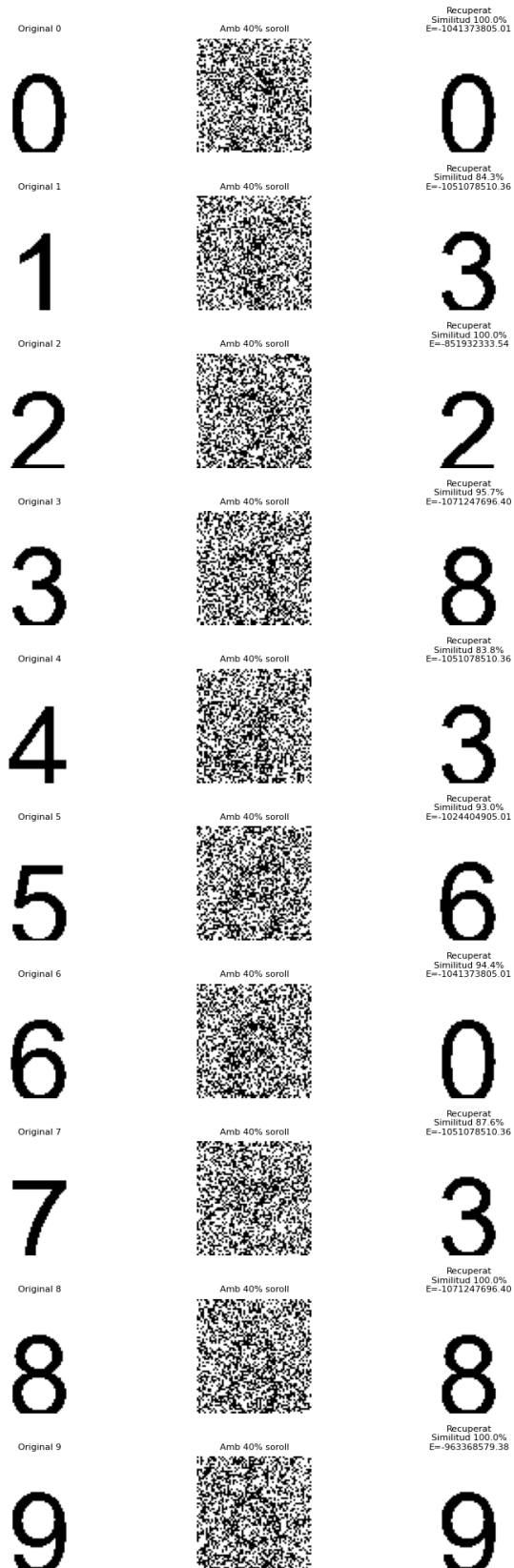


Figura 58: Recuperació dels dígets del 0 al 9 amb alta definició amb l'energia exponencial i 40% de soroll.



Figura 59: Recuperació de patrons ben diferenciats amb l'energia exponencial i 20% de soroll. (1)

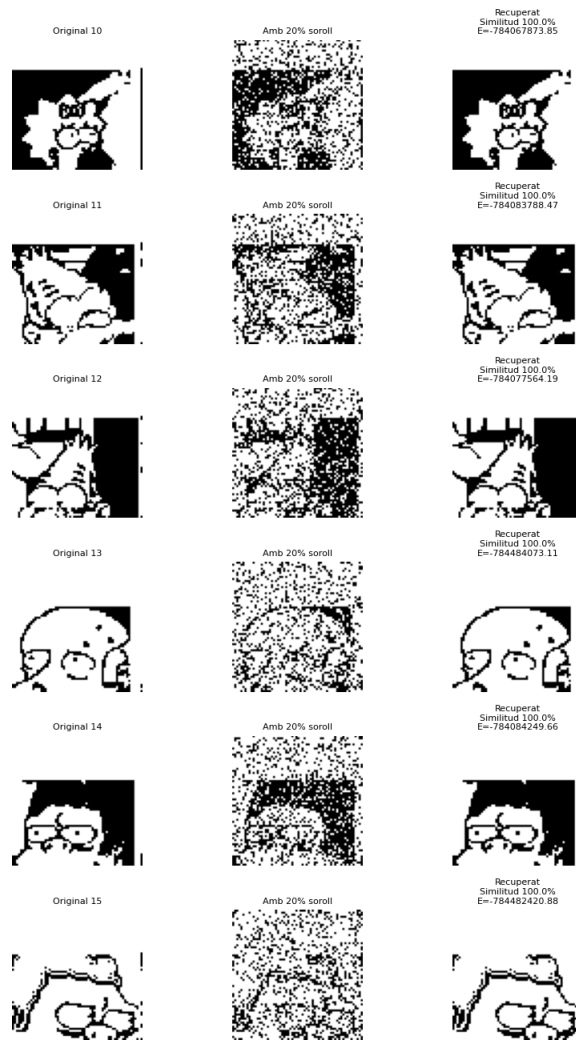


Figura 60: Recuperació de patrons ben diferenciats amb l'energia exponencial i 20% de soroll. (2)

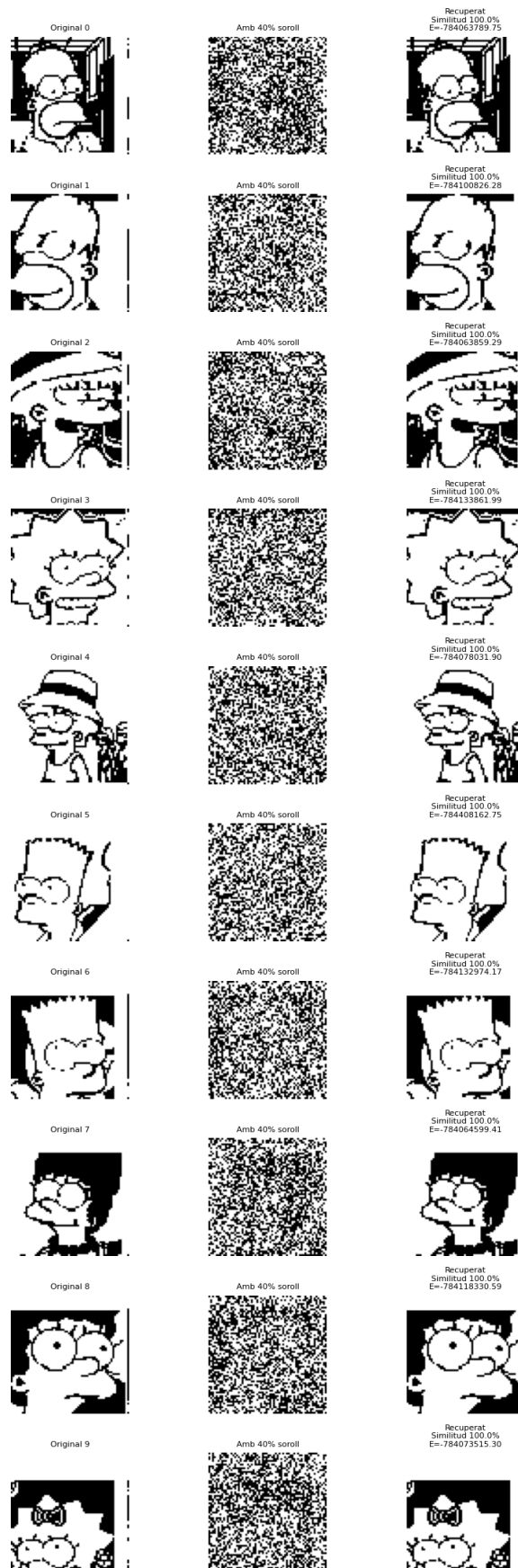


Figura 61: Recuperació de patrons ben diferenciats amb l'energia exponencial i 40% de soroll. (1)

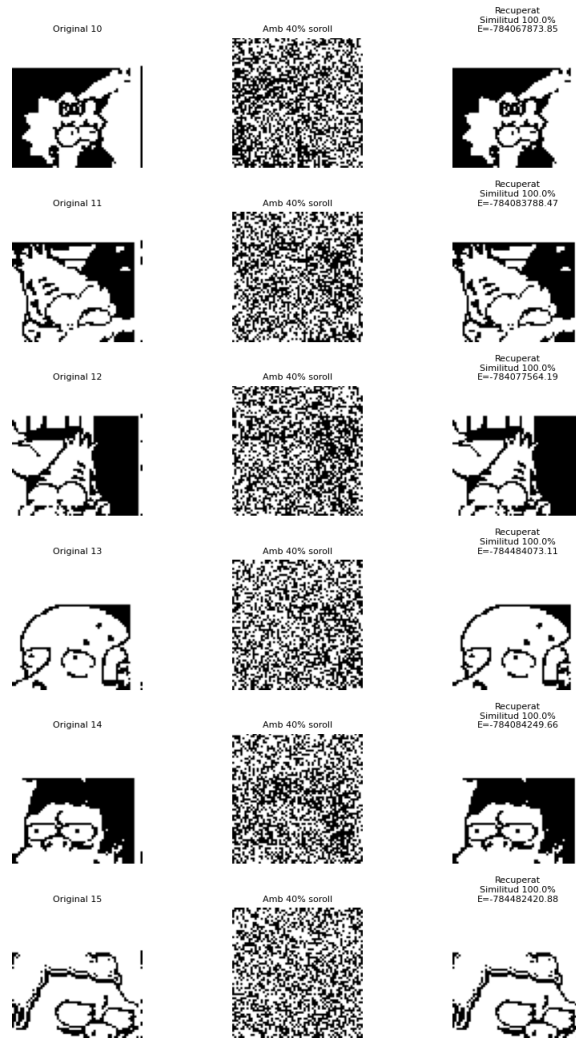


Figura 62: Recuperació de patrons ben diferenciats amb l'energia exponencial i 40% de soroll. (2)

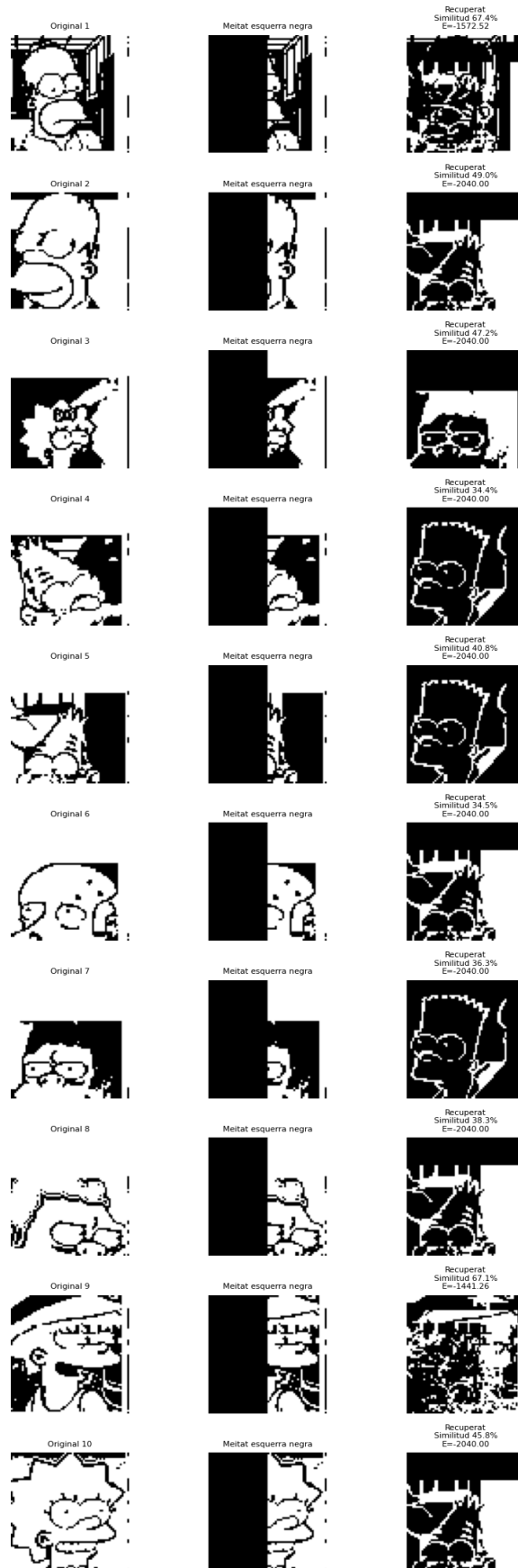


Figura 63: Recuperació de patrons ben diferenciats amb la pseudoinversa amb la meitat esquerra negra. (1)

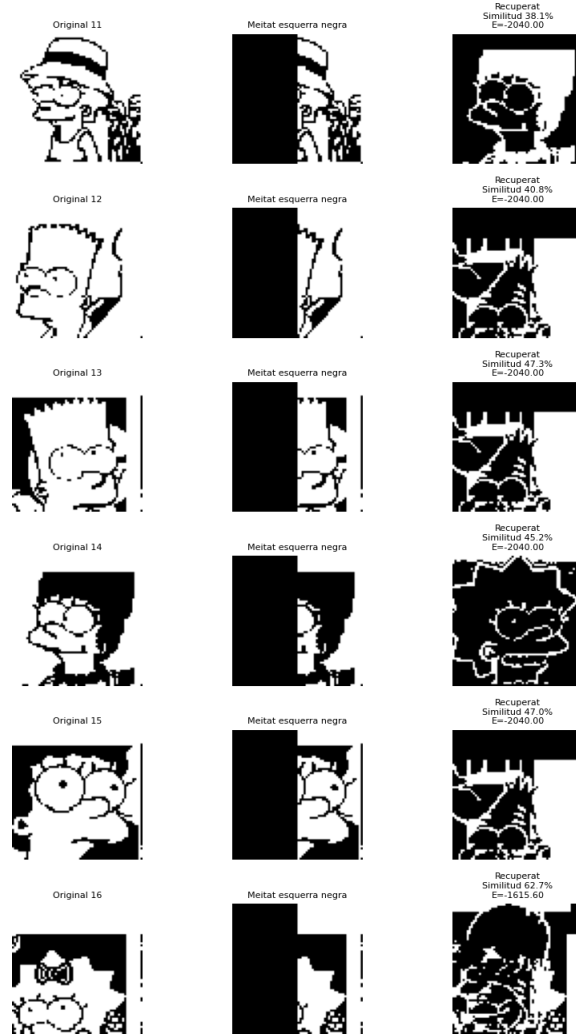


Figura 64: Recuperació de patrons ben diferenciats amb la pseudoinversa amb la meitat esquerra negra. (2)

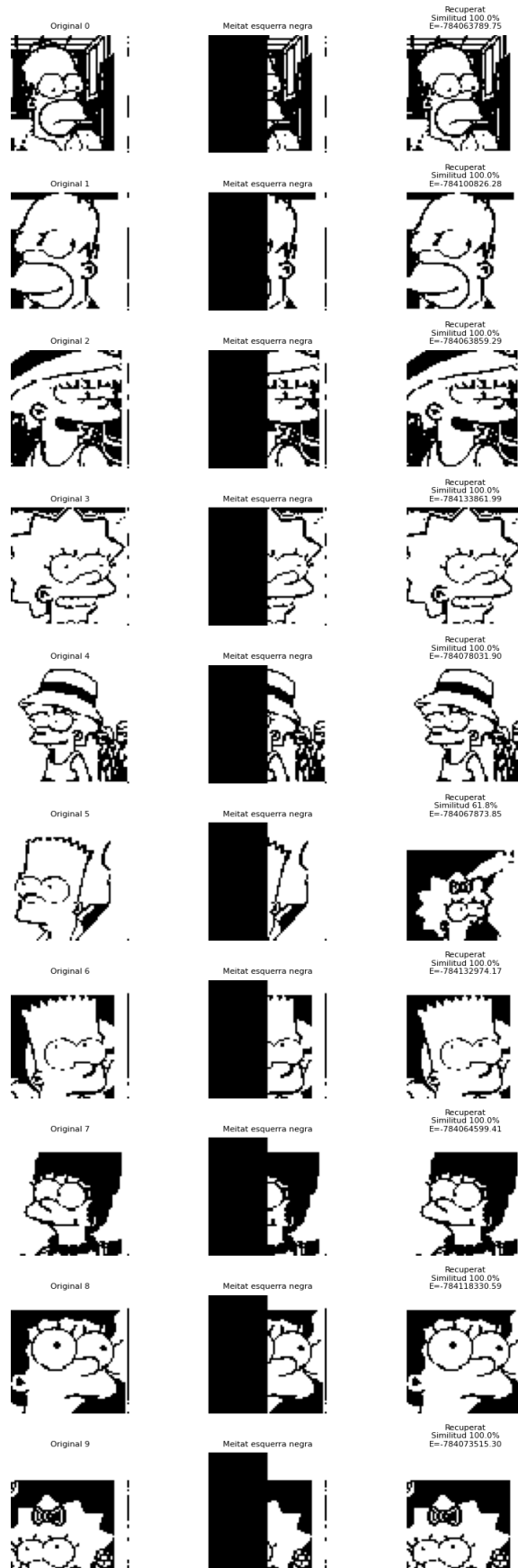


Figura 65: Recuperació de patrons ben diferenciats amb l'energia exponencial amb la meitat esquerra negra. (1)

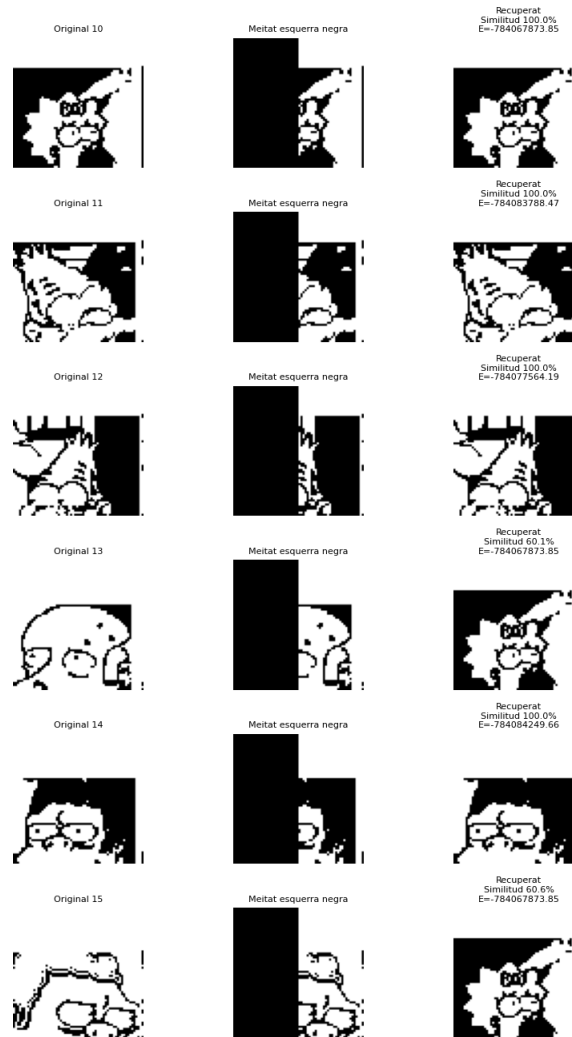


Figura 66: Recuperació de patrons ben diferenciats amb l'energia exponencial amb la meitat esquerra negra. (2)

Apèndix B: Codis

B.1 Dígits del 0 al 9 (8x8)

```

1
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 # Definim cada número com una llista plana de 64 elements (8x8)
6 # 0 = blanc, 1 = negre
7
8 numeros = {
9     0: np.array([
10         0, 1, 1, 1, 1, 1, 1, 0,
11         1, 0, 0, 0, 0, 0, 0, 1,
12         1, 0, 0, 0, 0, 0, 0, 1,

```

```

13         1, 0, 0, 0, 0, 0, 0, 0, 1,
14         1, 0, 0, 0, 0, 0, 0, 0, 1,
15         1, 0, 0, 0, 0, 0, 0, 0, 1,
16         1, 0, 0, 0, 0, 0, 0, 0, 1,
17         0, 1, 1, 1, 1, 1, 1, 1, 0
18     ]),
19
20     1: np.array([
21         0, 0, 0, 1, 1, 0, 0, 0,
22         0, 0, 1, 0, 0, 1, 0, 0,
23         0, 1, 0, 0, 0, 0, 1, 0,
24         0, 0, 0, 0, 0, 0, 1, 0,
25         0, 0, 0, 0, 0, 0, 1, 0,
26         0, 0, 0, 0, 0, 0, 1, 0,
27         0, 0, 0, 0, 0, 0, 1, 0,
28         0, 0, 0, 0, 0, 0, 1, 0
29     ]),
30
31     2: np.array([
32         0, 1, 1, 1, 1, 1, 1, 0,
33         1, 0, 0, 0, 0, 0, 0, 1,
34         0, 0, 0, 0, 0, 0, 0, 1,
35         0, 0, 0, 0, 0, 1, 1, 0,
36         0, 0, 1, 1, 1, 0, 0, 0,
37         0, 1, 1, 0, 0, 0, 0, 0,
38         1, 0, 0, 0, 0, 0, 0, 0,
39         1, 1, 1, 1, 1, 1, 1, 1
40     ]),
41
42     3: np.array([
43         0, 1, 1, 1, 1, 1, 1, 0,
44         1, 0, 0, 0, 0, 0, 0, 1,
45         0, 0, 0, 0, 0, 0, 0, 1,
46         0, 0, 0, 1, 1, 1, 1, 0,
47         0, 0, 0, 1, 1, 1, 1, 0,
48         0, 0, 0, 0, 0, 0, 0, 1,
49         1, 0, 0, 0, 0, 0, 0, 1,
50         0, 1, 1, 1, 1, 1, 1, 0
51     ]),
52
53     4: np.array([
54         1, 0, 0, 0, 0, 0, 0, 1,
55         1, 0, 0, 0, 0, 0, 0, 1,
56         1, 0, 0, 0, 0, 0, 0, 1,
57         1, 0, 0, 0, 0, 0, 0, 1,
58         0, 1, 1, 1, 1, 1, 1, 1,
59         0, 0, 0, 0, 0, 0, 0, 1,
60         0, 0, 0, 0, 0, 0, 0, 1,
61         0, 0, 0, 0, 0, 0, 0, 1
62     ]),
63

```

```

64     5: np.array([
65         1, 1, 1, 1, 1, 1, 1, 1,
66         1, 0, 0, 0, 0, 0, 0, 0,
67         1, 0, 0, 0, 0, 0, 0, 0,
68         1, 1, 1, 1, 1, 1, 1, 0,
69         0, 0, 0, 0, 0, 0, 0, 1,
70         0, 0, 0, 0, 0, 0, 0, 1,
71         1, 0, 0, 0, 0, 0, 0, 1,
72         0, 1, 1, 1, 1, 1, 1, 0
73     ]),
74
75     6: np.array([
76         0, 1, 1, 1, 1, 1, 1, 0,
77         1, 0, 0, 0, 0, 0, 0, 1,
78         1, 0, 0, 0, 0, 0, 0, 0,
79         1, 1, 1, 1, 1, 1, 1, 0,
80         1, 0, 0, 0, 0, 0, 0, 1,
81         1, 0, 0, 0, 0, 0, 0, 1,
82         1, 0, 0, 0, 0, 0, 0, 1,
83         0, 1, 1, 1, 1, 1, 1, 0
84     ]),
85
86     7: np.array([
87         1, 1, 1, 1, 1, 1, 1, 1,
88         0, 0, 0, 0, 0, 0, 0, 1,
89         0, 0, 0, 0, 0, 0, 1, 0,
90         0, 0, 0, 0, 0, 1, 0, 0,
91         0, 0, 0, 0, 1, 0, 0, 0,
92         0, 0, 0, 1, 0, 0, 0, 0,
93         0, 0, 1, 0, 0, 0, 0, 0,
94         0, 1, 0, 0, 0, 0, 0, 0
95     ]),
96
97     8: np.array([
98         0, 1, 1, 1, 1, 1, 1, 0,
99         1, 0, 0, 0, 0, 0, 0, 1,
100        1, 0, 0, 0, 0, 0, 0, 1,
101        0, 1, 1, 1, 1, 1, 1, 0,
102        0, 1, 1, 1, 1, 1, 1, 0,
103        1, 0, 0, 0, 0, 0, 0, 1,
104        1, 0, 0, 0, 0, 0, 0, 1,
105        0, 1, 1, 1, 1, 1, 1, 0
106     ]),
107
108     9: np.array([
109        0, 1, 1, 1, 1, 1, 1, 0,
110        1, 0, 0, 0, 0, 0, 0, 1,
111        1, 0, 0, 0, 0, 0, 0, 1,
112        1, 0, 0, 0, 0, 0, 0, 1,
113        0, 1, 1, 1, 1, 1, 1, 1,
114        0, 0, 0, 0, 0, 0, 0, 1,

```

```

115         1, 0, 0, 0, 0, 0, 0, 1,
116         0, 1, 1, 1, 1, 1, 1, 0
117     ])
118 }
119
120 # Visualització conjunta
121 fig, axs = plt.subplots(2, 5, figsize=(12, 4))
122 for i in range(10):
123     ax = axs[i // 5][i % 5]
124     ax.imshow(numeros[i].reshape(8, 8), cmap='binary')
125     ax.axis('off')
126
127 plt.tight_layout()
128 plt.show()

```

B.2 Dígits del 0 al 9 (64x64)

```

1     from scipy.ndimage import zoom
2
3     def escalar_a_64x64(patro_8x8):
4         # Convertim a matriu 8x8
5         matriu_8x8 = patro_8x8.reshape(8, 8)
6         # Escalem a 64x64 (factor 8)
7         matriu_64x64 = zoom(matriu_8x8, 8, order=0) # order=0 per interpolació nearest-ne
8         return matriu_64x64.flatten()
9
10    # Creem un diccionari amb els números 64x64
11    numeros_64x64 = {k: escalar_a_64x64(v) for k, v in numeros.items()}
12    # Funció per visualitzar un número
13    def mostrar_numero(numero):
14        matriu = numeros_64x64[numero].reshape(64, 64)
15        for fila in matriu:
16            print(' '.join(' ' if pixel == 1 else ' ' for pixel in fila))
17        print()
18
19    # Visualització dels números
20    for i in range(10):
21        print(f"Número {i}:")
22        mostrar_numero(i)
23 \end{lstlisting}
24
25 \subsection{Dígits del 0 al 9 amb alta resolució}
26
27 \begin{listing}[H]
28     \begin{minted}
29
30     import numpy as np
31     from PIL import Image, ImageDraw, ImageFont

```

```

32 import matplotlib.pyplot as plt
33
34 def generar_patrons_64_autoscale(marge=4):
35     """
36     Genera els digits 0-9 en 64x64 pixels, ajustant
37     automàticament la mida de la font perquè no talli.
38     - marge: nombre de píxels lliures al contorn
39     """
40     patrons = {}
41     for digit in range(10):
42         # Busquem la mida de font màxima que càpiga
43         font_size = 64 # punt de partida
44         while font_size > 1:
45             try:
46                 font = ImageFont.truetype("arial.ttf", font_size)
47             except IOError:
48                 font = ImageFont.load_default()
49             # mesurem caixa del text
50             mask = font.getmask(str(digit))
51             w, h = mask.size
52             if w <= 64 - 2*marge and h <= 64 - 2*marge:
53                 break
54             font_size -= 1
55
56         # Creem el canvas i dibuixem el dígit
57         img = Image.new("L", (64,64), 255)
58         draw = ImageDraw.Draw(img)
59         # recentrem dins el marge
60         x = (64 - w) // 2
61         y = (64 - h) // 2
62         draw.text((x, y), str(digit), font=font, fill=0)
63
64         # Binaritzem i desplaquem a vector
65         arr = np.array(img)
66         binar = (arr < 128).astype(int).flatten()
67         patrons[digit] = binar
68
69     return patrons
70
71
72 nums64 = generar_patrons_64_autoscale(marge=4)
73 numeros = nums64
74
75 # Visualització conjunta
76 fig, axs = plt.subplots(2, 5, figsize=(12, 4))
77 for i in range(10):
78     ax = axs[i // 5][i % 5]
79     ax.imshow(numeros[i].reshape(64, 64), cmap='binary')
80     ax.axis('off')
81
82 plt.tight_layout()

```

83 plt.show()

B.3 Lletres de la A a la Z

```

1  \begin{minted}
2  [
3  autogobble,
4  breaklines,
5  breakautoindent,
6  % highlightlines={1, 3-4},
7  linenos
8  ]
9  {python}
10
11  import numpy as np
12  from PIL import Image, ImageDraw, ImageFont
13  import matplotlib.pyplot as plt
14
15  # ----- GENERACIÓ DE LLETRES 64x64 -----
16  def generar_lletres_64_autoscale(marge=4):
17      lletres = {}
18      for char_code in range(ord('A'), ord('Z') + 1):
19          cc = chr(char_code)
20          font_size = 64
21          while font_size > 1:
22              try:
23                  font = ImageFont.truetype("arial.ttf", font_size)
24              except IOError:
25                  font = ImageFont.load_default()
26              mask = font.getmask(cc)
27              w, h = mask.size
28              if w <= 64 - 2 * marge and h <= 64 - 2 * marge:
29                  break
30              font_size -= 1
31
32          img = Image.new("L", (64, 64), 255)
33          draw = ImageDraw.Draw(img)
34          x = (64 - w) // 2
35          y = (64 - h) // 2
36          draw.text((x, y), cc, font=font, fill=0)
37
38          arr = np.array(img)
39          binar = (arr < 128).astype(int).flatten()
40          lletres[cc] = binar
41
42      return lletres
43
44  # ----- MOSTRAR TOTES LES LLETRES EN EL NOTEBOOK -----

```

```

45 def mostrar_lletres(lletres):
46     fig, axs = plt.subplots(6, 5, figsize=(12, 8))
47     axs = axs.flatten()
48
49     for i, lletra in enumerate(sorted(lletres.keys())):
50         matriu = lletres[lletra].reshape(64, 64)
51         axs[i].imshow(matriu, cmap='gray_r')
52         axs[i].axis('off')
53
54         # Amaga caselles sobrants
55         for j in range(i+1, len(axs)):
56             axs[j].axis('off')
57
58     plt.tight_layout()
59     plt.show()
60
61     # ----- EXECUCIÓ -----
62     lletres64 = generar_lletres_64_autoscale()
63     mostrar_lletres(lletres64)

```

B.4 Patrons ben diferenciats

```

1     import pandas as pd
2     import numpy as np
3     import matplotlib.pyplot as plt
4
5     # Llegeix el CSV
6     df = pd.read_csv('vectors_personatges_64x64.csv', index_col=0)
7
8     # Diccionari per guardar els vectors
9     personatges = {}
10
11    # Itera per cada fila (cada patró)
12    for i, (idx, fila) in enumerate(df.iterrows(), start=1):
13        vector = fila.to_numpy()
14        personatges[i] = vector
15
16        # Mostra la imatge (opcional)
17        matriu = vector.reshape(64, 64)
18        plt.imshow(matriu, cmap='gray_r')
19        plt.title(f'Personatge {i}')
20        plt.axis('off')
21        plt.show()
22
23    # Visualització conjunta
24    fig, axs = plt.subplots(4, 4, figsize=(10, 10)) # 4x4 = 16 patrons
25
26    for i in range(16):

```

```

27     ax = axs[i // 4, i % 4] # correcte accés a array 2D
28     matriu = personatges[i + 1].reshape(64, 64) # l'índex comença a 1
29     ax.imshow(matriu, cmap='gray_r')
30     ax.set_title(f'Personatge {i+1}', fontsize=8)
31     ax.axis('off')
32 \end{lstlisting}
33
34 \subsection{Recuperació amb regla de Hebb i pseudoinversa}
35
36 \begin{lstlisting}[language=Python, caption={}]
37     import numpy as np
38     import matplotlib.pyplot as plt
39
40     # ----- PREPROCESSAMENT DE PATRONS -----
41
42     def binari_a_bipolar(patrons_bin):
43         """
44         Converteix un conjunt de patrons binaris (0,1) a codificació bipolar (-1,+1).
45         És el format necessari per operar amb les xarxes de Hopfield clàssiques.
46         """
47         return {k: np.where(v == 0, -1, +1) for k, v in patrons_bin.items()}
48
49     # ----- REGLA D'APRENTATGE HEBBIANA -----
50
51     def entrenar_hebb(patrons):
52         """
53         Entrena una xarxa de Hopfield amb la regla hebbiana.
54         Calcula la matriu de pesos W com la suma dels productes exteriors de cada patró.
55         """
56         n = next(iter(patrons.values())).size # Nombre de neurones (longitud dels vectors)
57         W = sum(np.outer(x, x) for x in patrons.values()) # Suma de productes exteriors
58         np.fill_diagonal(W, 0) # S'anul·len les diagonals per evitar autointeraccions
59         return W / n # Normalització per n
60
61     # ----- REGLA D'APRENTATGE PSEUDO-INVERA -----
62
63     def entrenar_pseudoinversa(patrons):
64         """
65         Entrenament per pseudoinversa de Moore-Penrose.
66         Genera una matriu W òptima que minimitza les interferències entre patrons.
67         """
68         P = np.stack(list(patrons.values()), axis=1) # Matriu de patrons: cada columna és
69         inv = np.linalg.pinv(P.T @ P) # Inversa generalitzada de la matriu de Gram
70         W = P @ inv @ P.T # Fórmula:  $W = P (P^T P)^{-1} P^T$ 
71         np.fill_diagonal(W, 0)
72         return W
73
74     # ----- FUNCIONS DE LA XARXA -----
75
76     def energia(W, x, theta=None):
77         """

```

```

78     Calcula l'energia de l'estat x en la xarxa amb pesos W i llindars theta.
79     Aquesta energia decreix a mesura que s'estabilitza la xarxa.
80     """
81     if theta is None:
82         theta = np.zeros_like(x)
83     return -0.5 * x @ W @ x + theta @ x # Fórmula de Lyapunov
84
85 def recuperar(W, x_inicial, theta=None, max_iter=40000, sincron=True):
86     """
87     Algorisme iteratiu per recuperar un patró a partir d'una versió corrompuda.
88     Es pot fer de manera síncrona (totes les neurones a la vegada)
89     o asíncrona (una a una, de manera aleatòria).
90     """
91     x = x_inicial.copy()
92
93     if sincron:
94         for _ in range(max_iter):
95             x_new = np.sign(W @ x - (theta if theta is not None else 0)) # Actualitza
96             x_new[x_new == 0] = 1 # Resolució de zeros (es posen a 1)
97             if np.array_equal(x_new, x): break # Parada si es convergeix
98             x = x_new
99     else:
100         for _ in range(max_iter):
101             i = np.random.randint(x.size) # Tria una neurona aleatòriament
102             h = W[i] @ x - (theta[i] if theta is not None else 0) # Potencial local
103             x[i] = 1 if h >= 0 else -1 # Actualització individual
104
105     return x, energia(W, x)
106
107     # ----- CORRUPCIÓ PER MEITAT ESQUERRA -----
108
109 def corrupte_meitat_esquerra(original):
110     """
111     Versió de corrupció especial: pinta tota la meitat esquerra a blanc (0).
112     S'utilitza per provar recuperacions més exigents.
113     """
114     mat = original.copy().reshape(64, 64)
115     mat[:, :32] = 0 # Esborrar la meitat esquerra
116     return mat.flatten()
117
118     # ----- TEST AMB OPCIONS DE REGLA -----
119
120 def test_recuperacio(regla='hebb', sincron=True):
121     """
122     Funció principal per testar la recuperació de patrons amb:
123     - La regla hebbiana o la pseudoinvertida
124     - Actualització síncrona o asíncrona
125     Mostra els resultats en una figura amb 10 files i 3 columnes:
126     (original, corrupte, recuperat)
127     """
128     np.random.seed(48) # Per garantir la mateixa corrupció cada cop

```

```

129
130 bipolar = binari_a_bipolar(numeros)
131 patrons = [bipolar[i] for i in sorted(bipolar)]
132 W = entrenar_hebb(bipolar) if regla == 'hebb' else entrenar_pseudoinversa(bipolar)
133
134 fig, axs = plt.subplots(10, 3, figsize=(9, 20))
135
136 for num, original in enumerate(patrons):
137     # corrupte = corrupte_meitat_esquerra(original) # Opció alternativa
138     corrupte = original.copy()
139     idx = np.random.choice(original.size, size=int(0.2 * original.size), replace=False)
140     corrupte[idx] *= -1 # Aplica un 20% de soroll aleatori
141
142     recuperat, E = recuperar(W, corrupte, sincron=sincron) # Recuperació
143     simil = np.mean(recuperat == original) # Percentatge de bits iguals
144
145     dades = [original, corrupte, recuperat]
146     titols = [
147         f"Original {num}",
148         "Amb 20% soroll",
149         f"Recuperat\nSimilitud {simil:.1%}\nE={E:.2f}"
150     ]
151
152     for col in range(3):
153         ax = axs[num, col]
154         mat = (dades[col] > 0).reshape(64, 64)
155         ax.imshow(mat, cmap='binary') # Visualització en blanc i negre
156         ax.set_title(titols[col], fontsize=8)
157         ax.axis('off')
158
159     plt.tight_layout()
160     plt.show()
161
162 # ----- EXECUCIÓ DE TEST -----
163
164 if __name__ == "__main__":
165     print("--- Test Hebbià ---")
166     test_recuperacio('hebb', True)
167     test_recuperacio('hebb', False)
168
169     print("--- Test Pseudo-inversa ---")
170     test_recuperacio('pseudo', True)
171     test_recuperacio('pseudo', False)
172

```

B.5 Recuperació amb l'energia exponencial

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3 import pandas as pd
4
5 # ----- LECTURA DE DADES -----
6
7 # Llegeix el fitxer CSV amb patrons 64x64 (1 fila per patró)
8 df = pd.read_csv('vectors_personatges_64x64.csv', index_col=0)
9
10 # Converteix el DataFrame a diccionari, cada vector binari és una entrada
11 personatges = {str(i): df.iloc[i].to_numpy() for i in range(len(df))}
12
13 # ----- CONVERSIÓ A CODIFICACIÓ BIPOLAR -----
14
15 def binari_a_bipolar(patrons_bin):
16     """
17     Converteix patrons binaris (0,1) a bipolar (-1,+1),
18     necessari per a la dinàmica de les xarxes de Hopfield.
19     """
20     return {k: np.where(v == 0, -1, +1) for k, v in patrons_bin.items()}
21
22 # ----- FUNCIÓ DE CORRUPCIÓ -----
23
24 def corrupte_meitat_esquerra(original):
25     """
26     Degrada el patró establint tota la meitat esquerra a 1 (negre).
27     Simula una pèrdua estructural forta.
28     """
29     mat = original.copy().reshape(64, 64)
30     mat[:, :32] = 1 # Meitat esquerra a 1
31     return mat.flatten()
32
33 # ----- ENERGIA EXPONENCIAL -----
34
35 def energia_exponencial(patrons, x, beta=0.005):
36     """
37     Calcula l'energia exponencial proposada per Demircigil et al.
38     Aquesta energia depèn exponencialment de la projecció entre estat i patrons.
39     """
40     return -sum(np.exp(beta * (p @ x)) for p in patrons)
41
42 # ----- ACTUALITZACIÓ D'UNA COMPONENT -----
43
44 def actualitzar_component_exponencial(patrons, xi, l):
45     """
46     Actualitza una neurona (component l) segons quin valor dona menor energia.
47     Compara xi[l] = +1 i xi[l] = -1 i es queda amb el millor.
48     """
49     xi_plus = xi.copy()
50     xi_plus[l] = 1

```

```

51     E_plus = energia_exponencial(patrons, xi_plus)
52
53     xi_minus = xi.copy()
54     xi_minus[1] = -1
55     E_minus = energia_exponencial(patrons, xi_minus)
56
57     delta = E_minus - E_plus
58     if delta == 0:
59         return xi[1] # Manté el valor si no canvia energia
60     else:
61         return 1 if delta > 0 else -1 # Tria el que tingui energia més baixa
62
63 # ----- RECUPERACIÓ ASÍNCRONA -----
64
65 def recuperar_exponencial_asincron(patrons, x_inicial, max_iter=50):
66     """
67     Recupera un patró utilitzant l'energia exponencial.
68     La dinàmica és asíncrona: s'actualitzen neurones una a una.
69     Es guarda un històric de l'energia per traçar-ne l'evolució.
70     """
71     x = x_inicial.copy()
72     x[x == 0] = -1 # S'assegura que x és bipolar
73
74     energia_hist = [energia_exponencial(patrons, x)]
75     for _ in range(max_iter):
76         canviat = False
77         for l in range(len(x)):
78             nova_xl = actualitzar_component_exponencial(patrons, x, l)
79             if nova_xl != x[l]:
80                 x[l] = nova_xl
81                 canviat = True
82         energia_hist.append(energia_exponencial(patrons, x))
83         if not canviat:
84             break # Para si no hi ha cap canvi (convergència)
85
86     return x, energia_hist # Retorna l'estat final i l'històric d'energia
87
88 # ----- TEST VISUAL DE LA RECUPERACIÓ -----
89
90 def test_recuperacio():
91     """
92     Visualitza la recuperació dels patrons binaris amb corrupció severa (meitat esquerre)
93     Es mostren 3 imatges per patró: original, corrupte i recuperat.
94     """
95     np.random.seed(48)
96
97     bipolar = binari_a_bipolar(personatges)
98     patrons = [bipolar[i] for i in sorted(bipolar, key=int)] # Ordena els patrons pel
99
100     for bloc in range(2): # Bloc 0: primers 10 patrons. Bloc 1: últims 6
101         n_files = 10 if bloc == 0 else 6

```

```

102     fig, axs = plt.subplots(n_files, 3, figsize=(9, 2.2 * n_files)) # 3 columnes:
103
104     rang = range(bloc * 10, bloc * 10 + n_files) # Índex dels patrons d'aquest bl
105
106     for fila, num in enumerate(rang):
107         original = patrons[num]
108         corrupte = corrupte_meitat_esquerra(original) # Aplica la corrupció
109
110         # Recuperació amb energia exponencial
111         recuperat, E_hist = recuperar_exponencial_asincron(patrons, corrupte)
112         E_final = E_hist[-1]
113         simil = np.mean(recuperat == original) # Percentatge de bits encertats
114
115         # Prepara les dades per mostrar
116         dades = [original, corrupte, recuperat]
117         titols = [
118             f"Original {num}",
119             "Meitat esquerra negra",
120             f"Recuperat\nSimilitud {simil:.1%}\nE={E_final:.2f}"
121         ]
122
123         for col in range(3):
124             ax = axs[fila, col]
125             mat = (dades[col] > 0).reshape(64, 64)
126             ax.imshow(mat, cmap='binary')
127             ax.set_title(titols[col], fontsize=8)
128             ax.axis('off')
129
130     plt.tight_layout()
131     plt.show() # Mostra cada bloc de 10 o 6 patrons
132
133     # ----- EXECUCIÓ DEL TEST -----
134
135     test_recuperacio()
136

```
