



Percepción y **Sistemas** Inteligentes



Universidad
del Valle

ENTRENAMIENTO DE RED NEURONAL ARTIFICIAL MORFOLÓGICA DE DENDRITAS CON ALGORITMO DE OPTIMIZACIÓN POR INTELIGENCIA DE ENJAMBRES

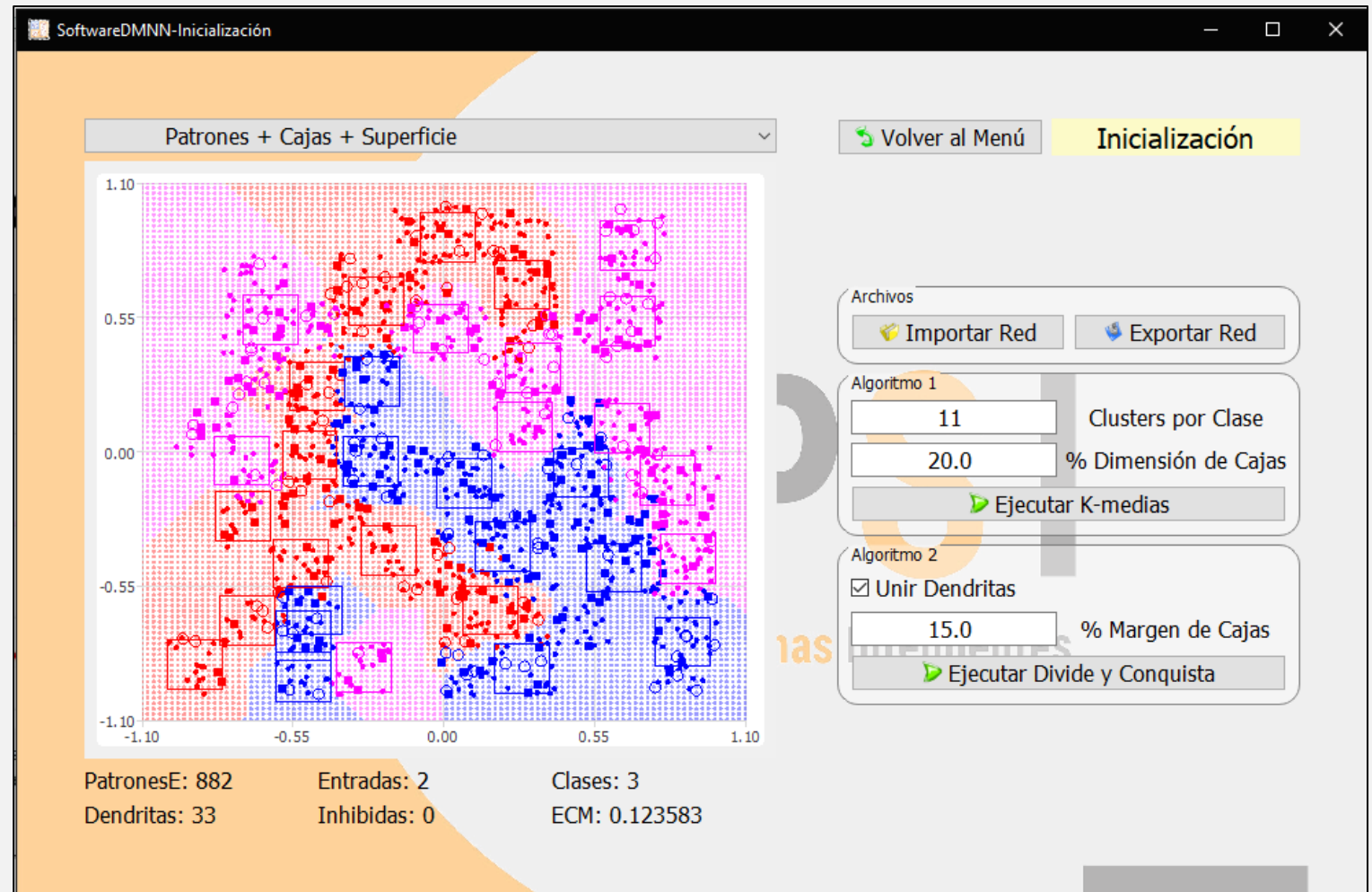


Omar Jordán Jordán

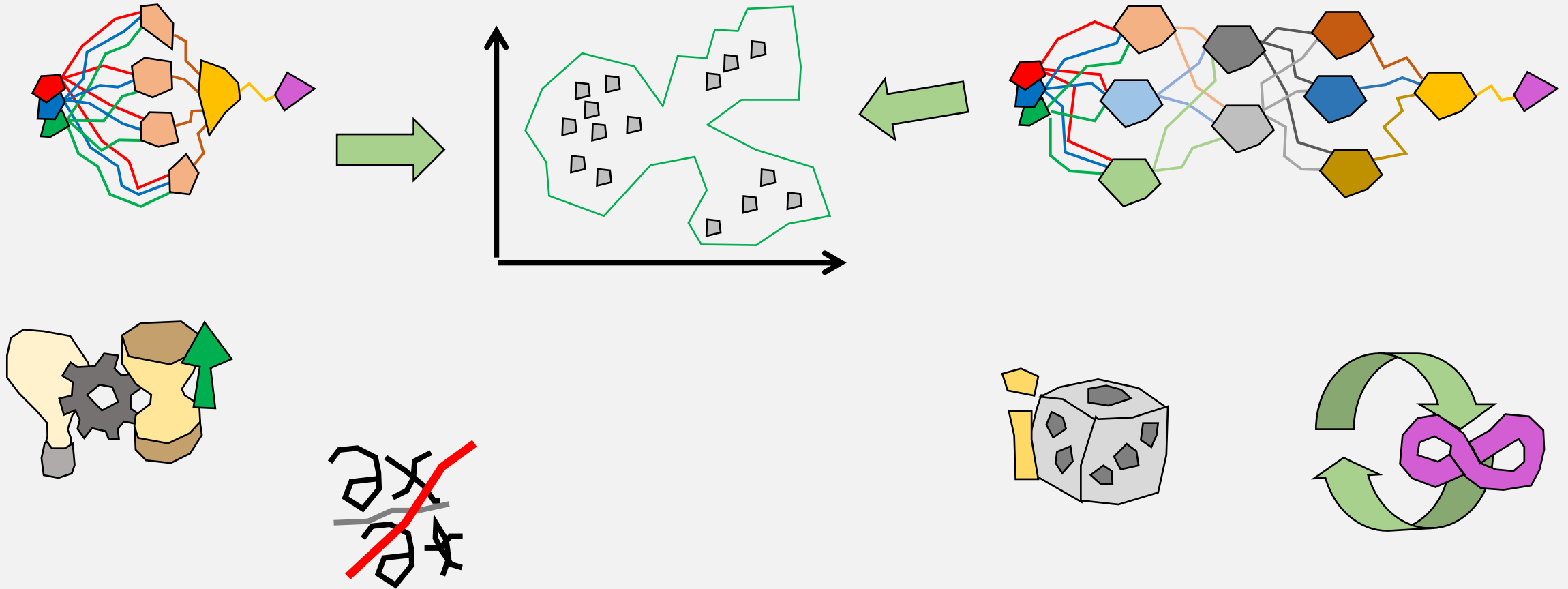
omar.jordan@correounivalle.edu.co

Contenido

1. Introducción
2. Pregunta Problema
3. Objetivos
4. Marco Teórico
5. Solución Propuesta
6. Resultados
7. Conclusiones
8. Trabajos Futuros



1. Introducción



2. Pregunta Problema

¿Qué tan eficiente es la técnica de inteligencia de enjambres *PSO* en la sintonización de parámetros de una red neuronal *DMNN* en comparación con los algoritmos recientemente utilizados (*SGD*, *DE*)?

3. Objetivos

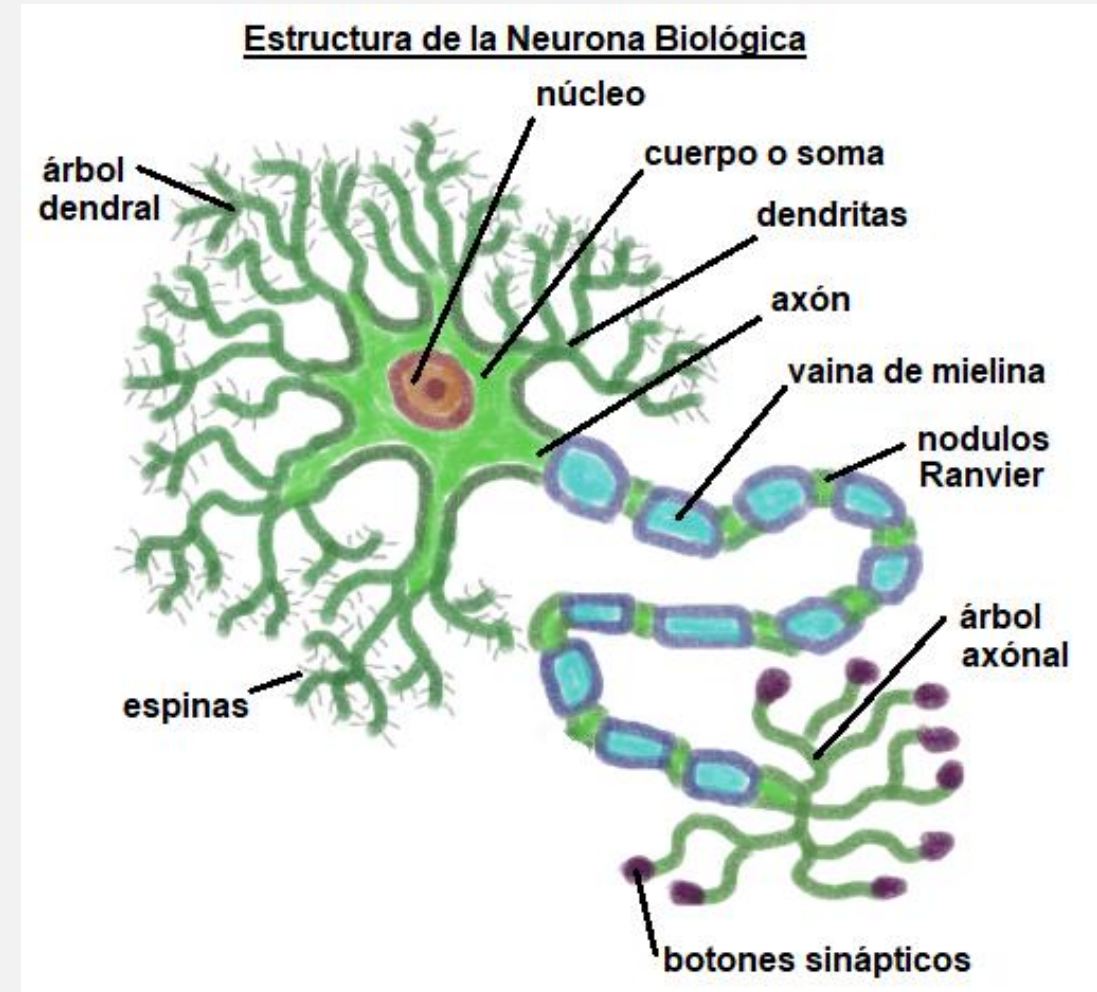
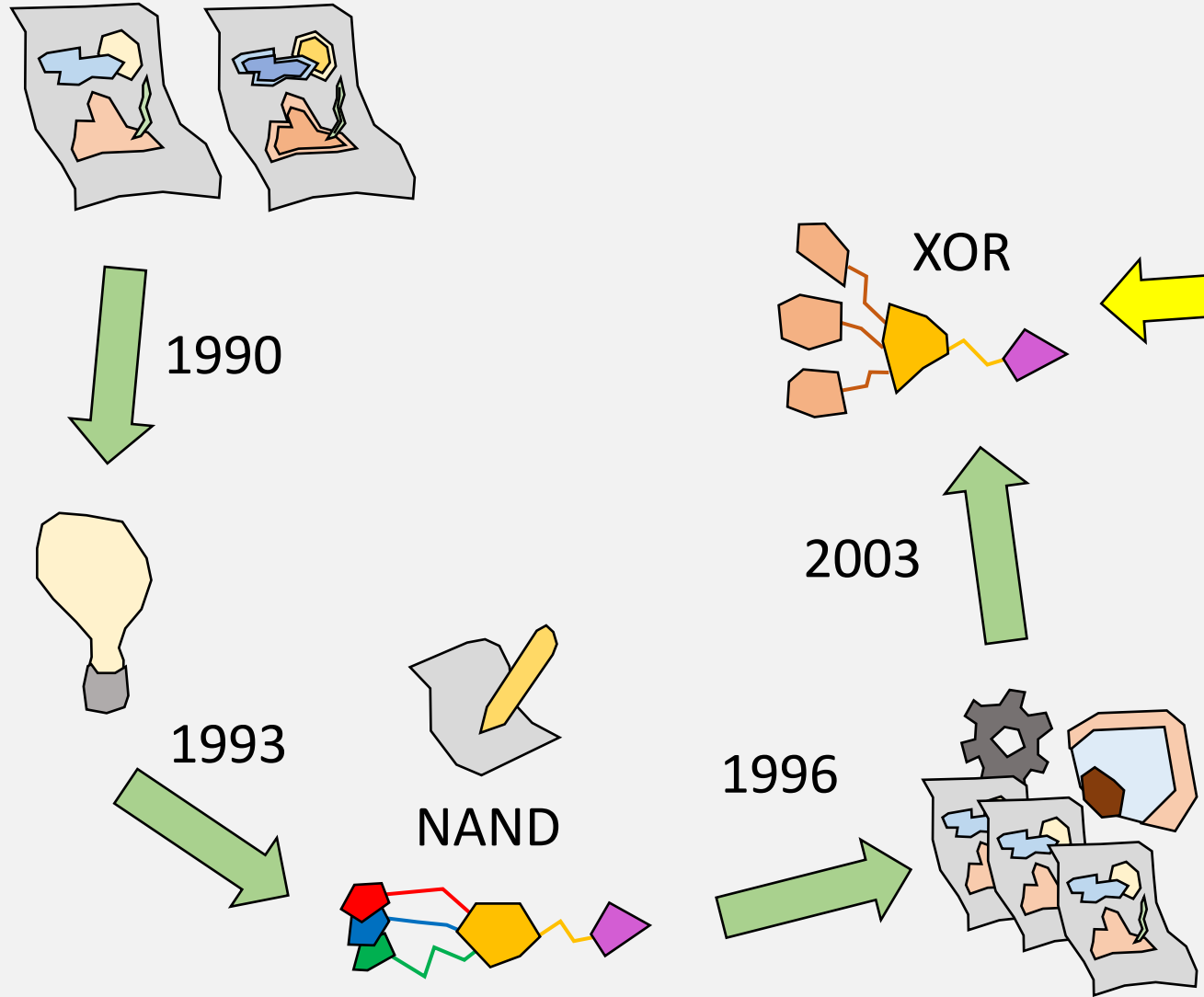
General:

Desarrollar un algoritmo de sintonización de los pesos sinápticos de una *DMNN* mediante un mecanismo de Optimización por Enjambre de Partículas (*PSO*), el cual debe ser comparado con los algoritmos: Gradiente Descendente Estocástico (*SGD*) y Evolución Diferencial (*DE*).

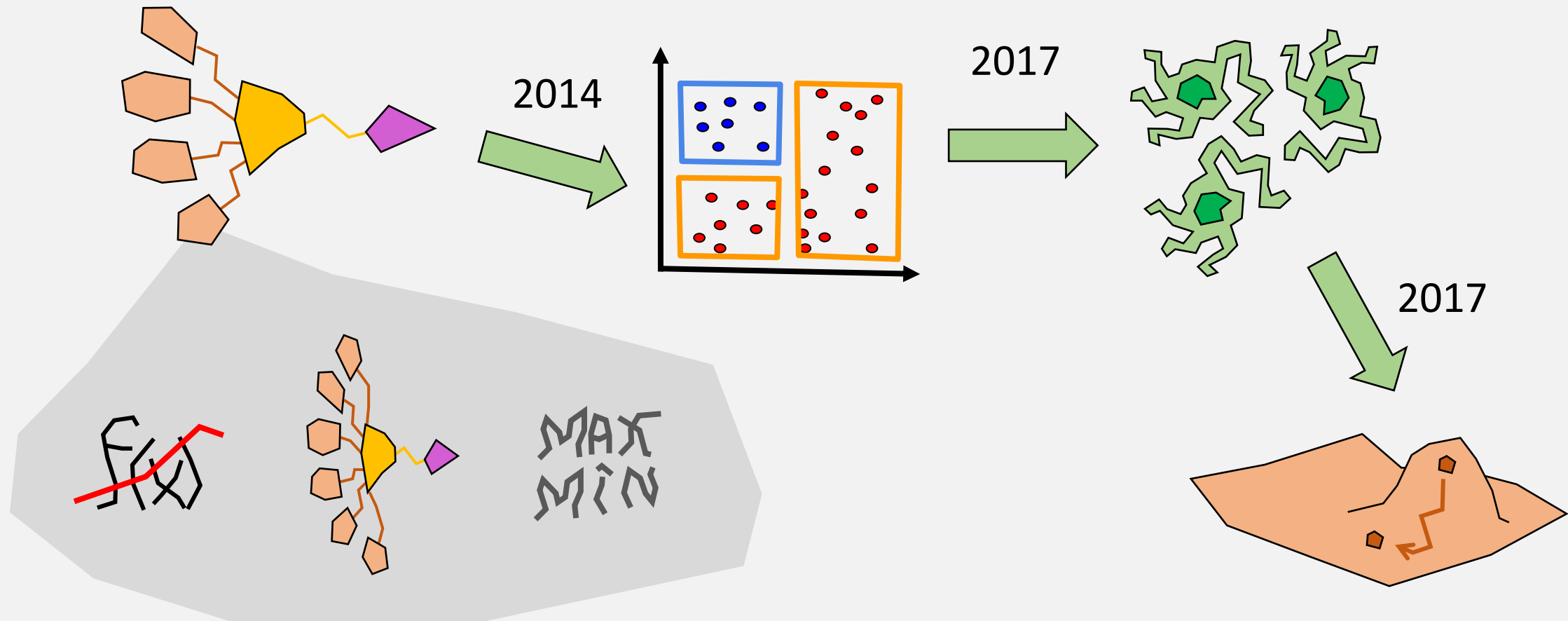
Específicos:

1. Realizar una búsqueda bibliográfica de los algoritmos de aprendizaje utilizados para la sintonización de una *DMNN*.
2. Desarrollar una herramienta software para poner a prueba la sintonización de parámetros de la red *DMNN* mediante los algoritmos *SGD*, *DE* y *PSO*.
3. Evaluar los algoritmos de aprendizaje para la red neuronal morfológica a través de la comparación de métricas de desempeño estandarizadas.

4. Marco Teórico: Origen *DMNN*



4. Marco Teórico: Origen *DMNN*



4. Marco Teórico: Modelo *DMNN*

$$S_{mk} = \bigvee_{i=0}^I \bigvee_{l=0:L}^{1:H} (-1)^l \cdot (x_i + w_{mki}^{li})$$

$$Z_m = \bigwedge_{k=0}^{K_m} S_{mk}$$

$$y = \operatorname{argmax}(z_m)$$

I = número de entradas.

M = número de neuronas.

$\ell = 0-1$, $\ell := L-H$ como etiqueta.

L = low (bajo), **H** = high (alto).

X = valores de las entradas.

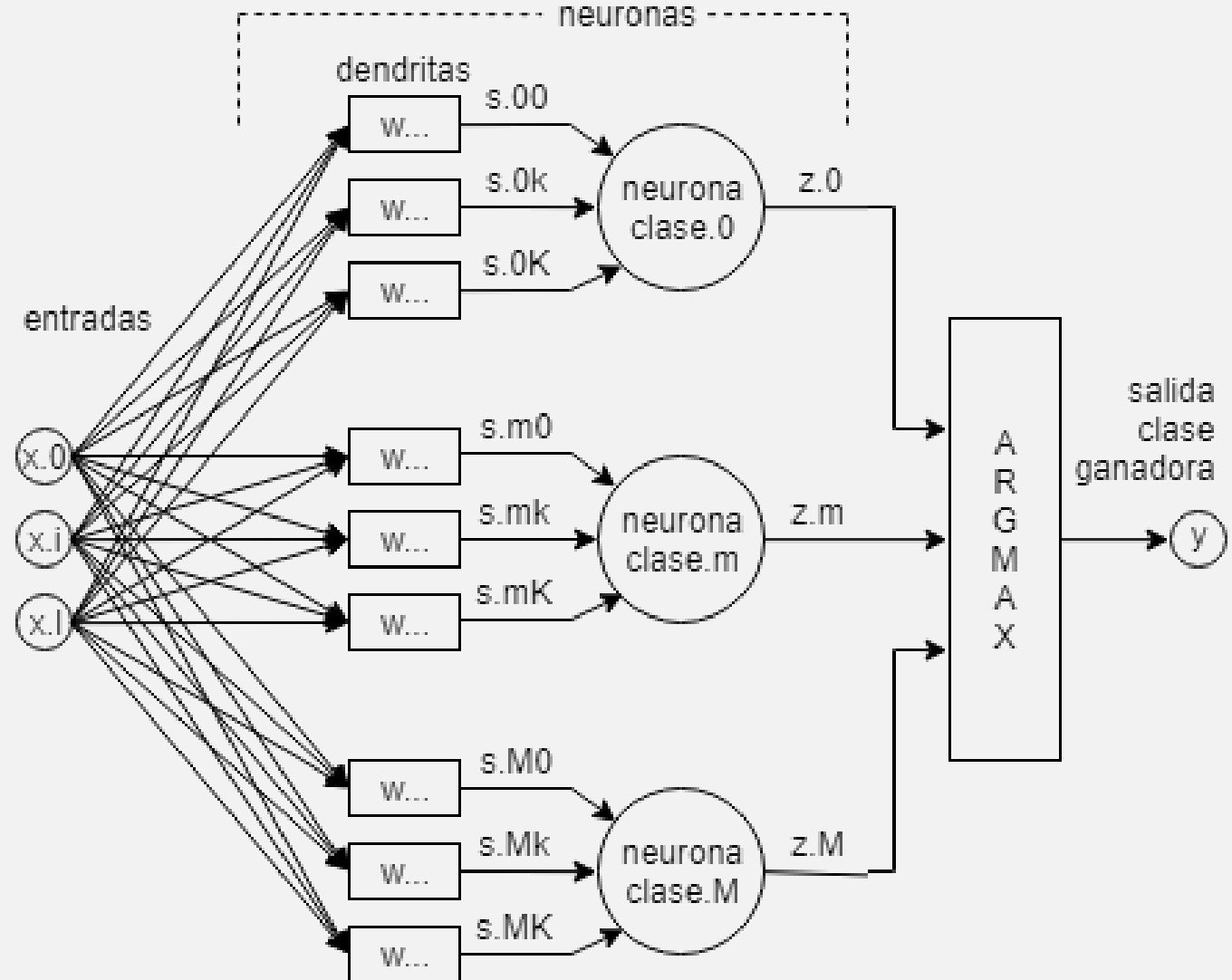
W = valores de pesos sinápticos.

K = valores de número de dendritas.

S = salidas de las dendritas.

Z = salidas de las neuronas.

y = clase / neurona ganadora.

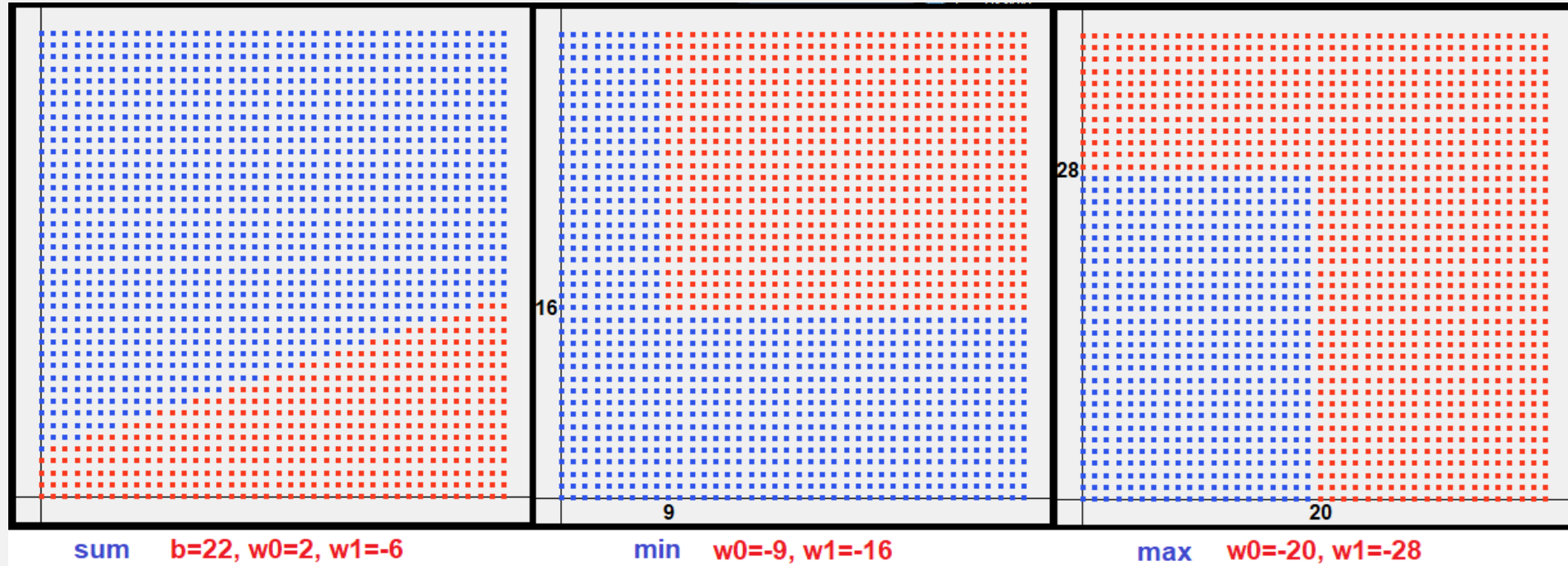
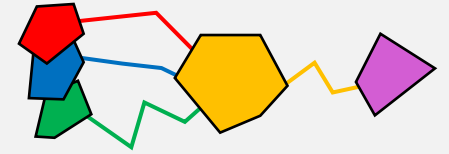


4. Marco Teórico: Modelo *DMNN*

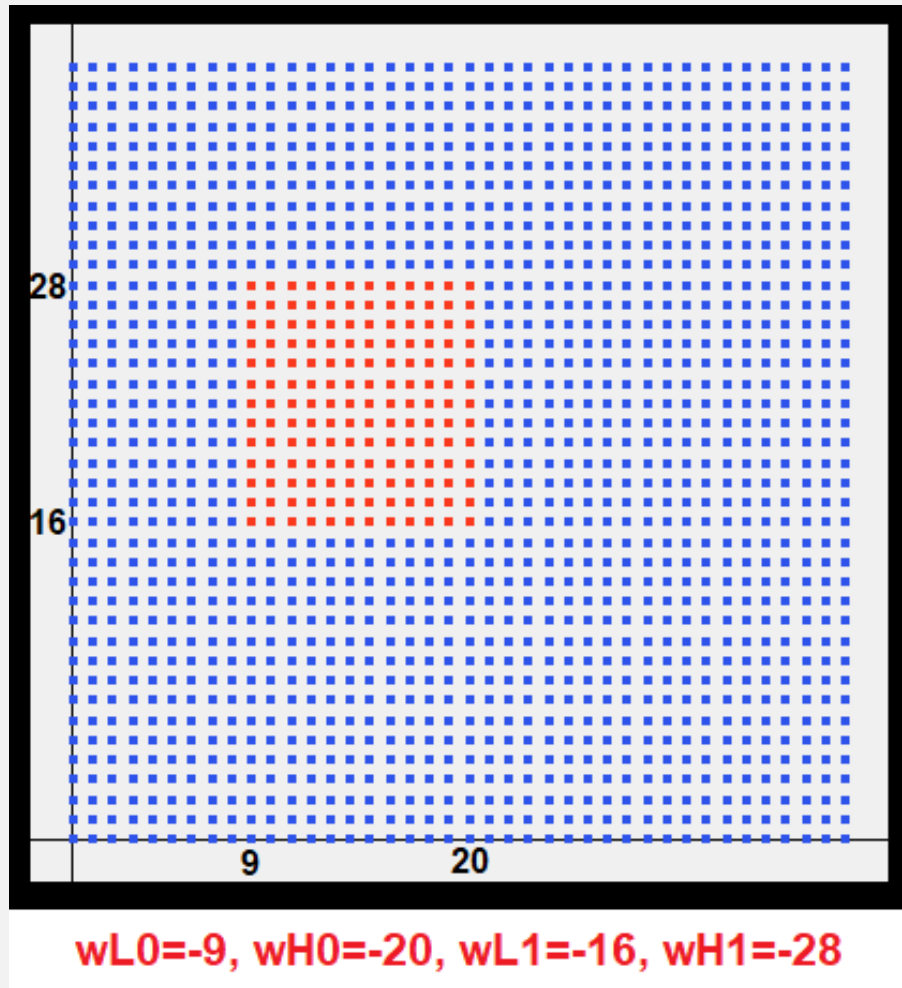
$$s = b + \sum_{i=0}^I (x_i \cdot w_i)$$



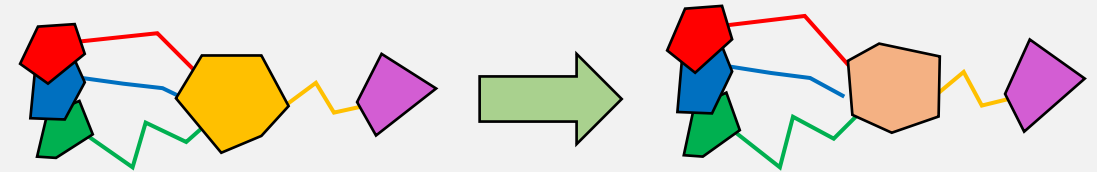
$$s = \bigvee_{i=0}^I (x_i + w_i)$$



4. Marco Teórico: Modelo *DMNN*

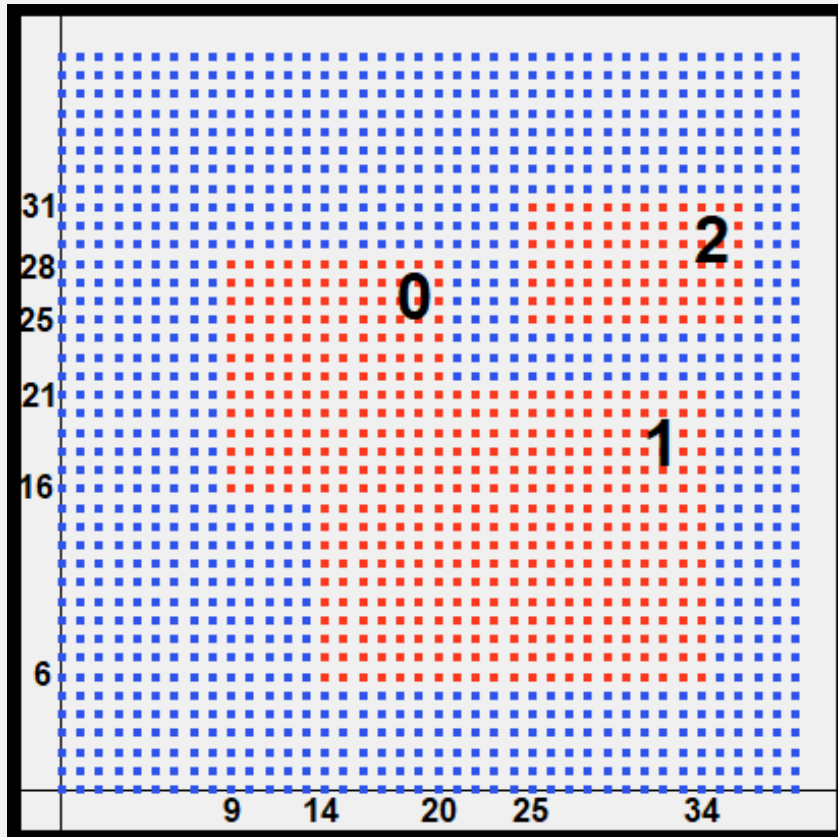


$$s = \bigvee_{i=0}^I \bigvee_{l=0:L}^{1:H} (-1)^l \cdot (x_i + w_i^{l:})$$



rojo (+), azul (-)

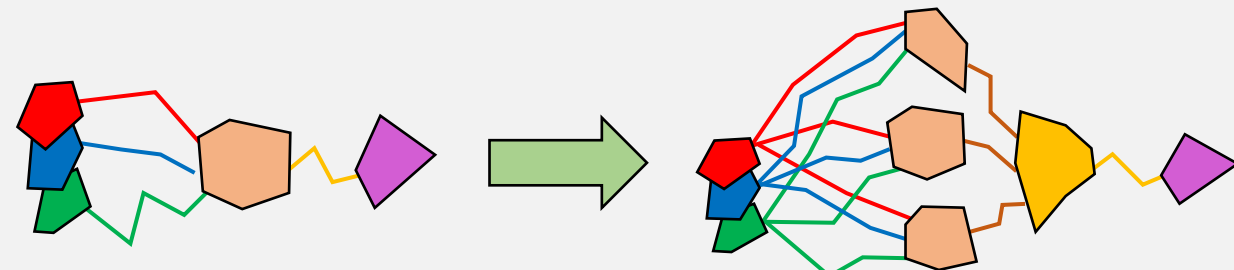
4. Marco Teórico: Modelo *DMNN*



$w_{L00}=-9$, $w_{H00}=-20$, $w_{L01}=-16$, $w_{H01}=-28$
 $w_{L10}=-14$, $w_{H10}=-34$, $w_{L11}=-6$, $w_{H11}=-21$
 $w_{L20}=-25$, $w_{H20}=-36$, $w_{L21}=-25$, $w_{H21}=-31$

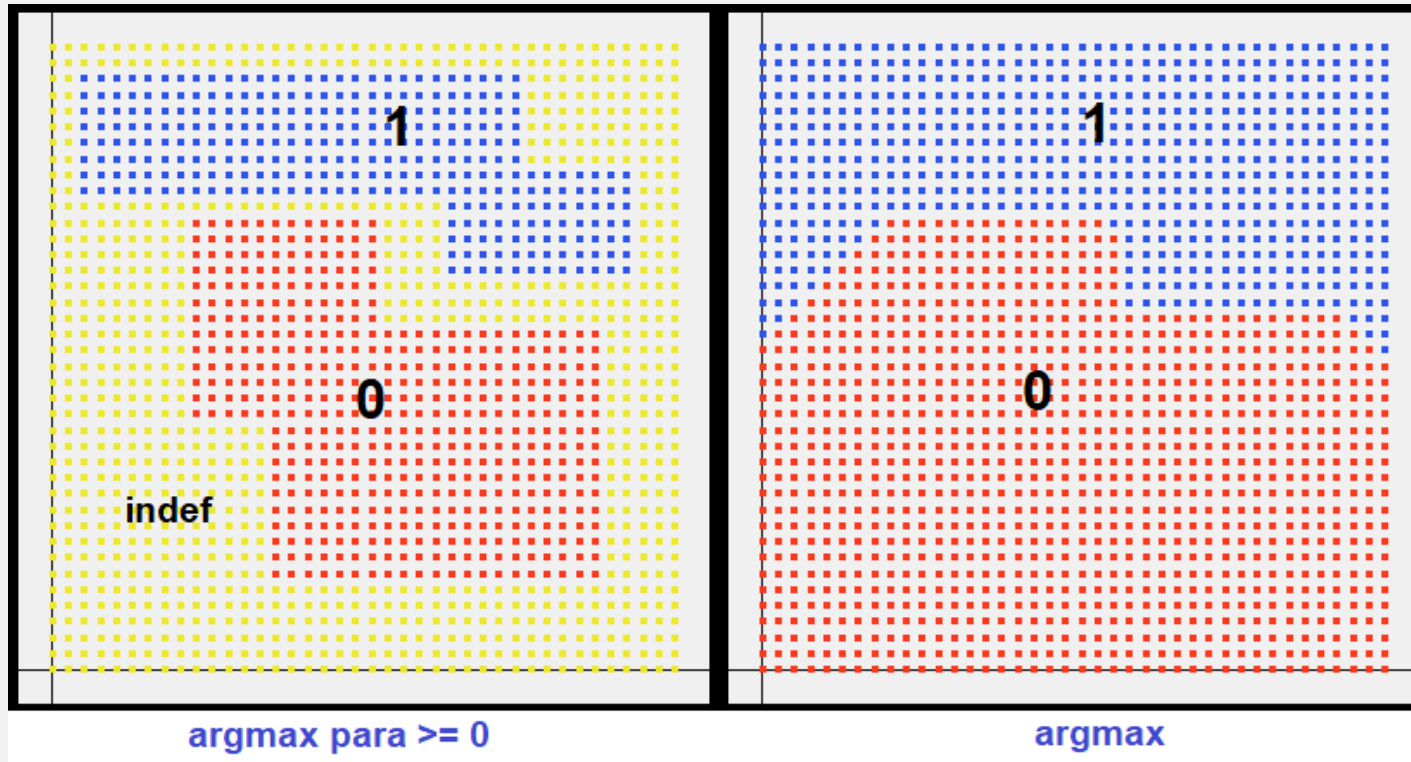
$$s_k = \bigvee_{i=0}^I \bigvee_{l=0:L}^{1:H} (-1)^l \cdot (x_i + w_{ki}^l)$$

$$z = \bigwedge_{k=0}^K s_k$$



rojo (+), azul (-)

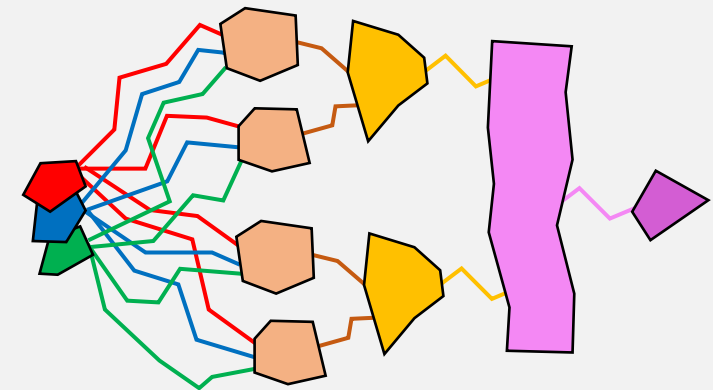
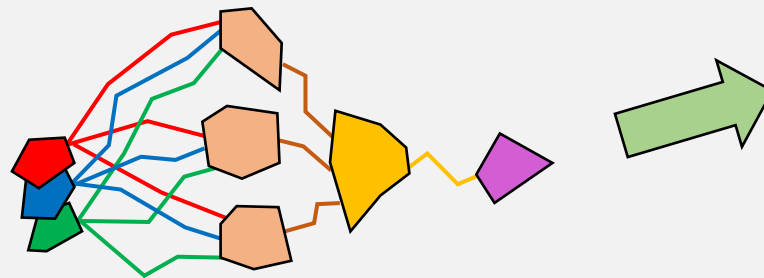
4. Marco Teórico: Modelo *DMNN*



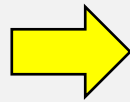
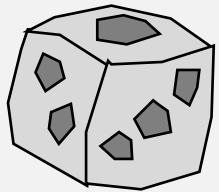
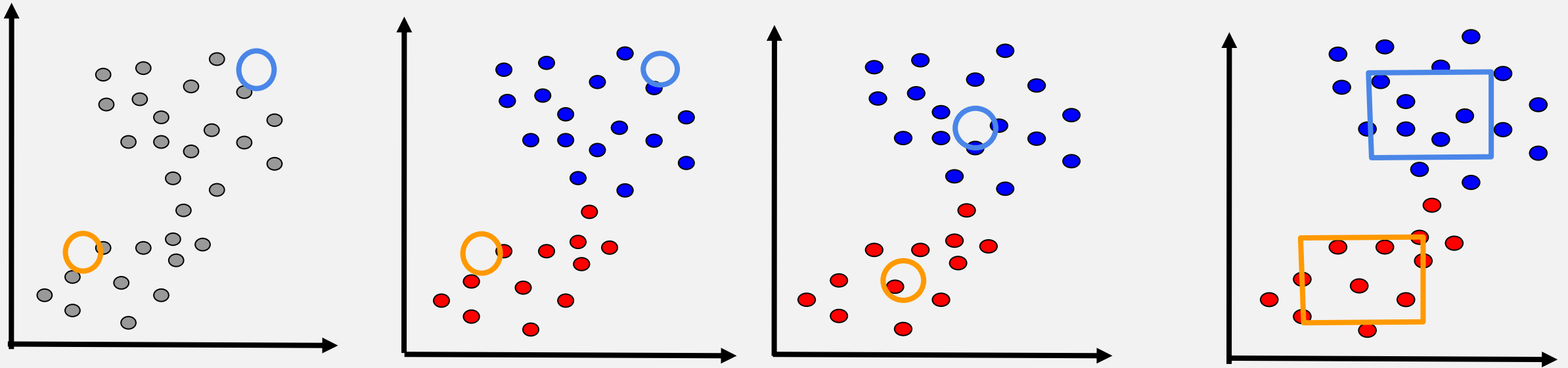
$$s_{mk} = \bigvee_{i=0}^I \bigvee_{l=0:L}^{1:H} (-1)^l \cdot (x_i + w_{mki}^l)$$

$$z_m = \bigwedge_{k=0}^{K_m} s_{mk}$$

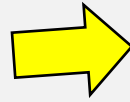
$$y = \operatorname{argmax}(z_m)$$



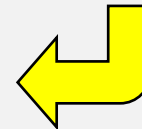
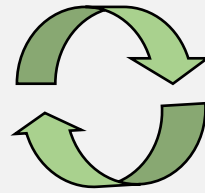
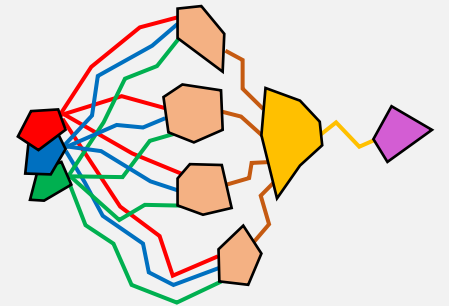
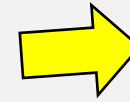
4. Marco Teórico: K-medias



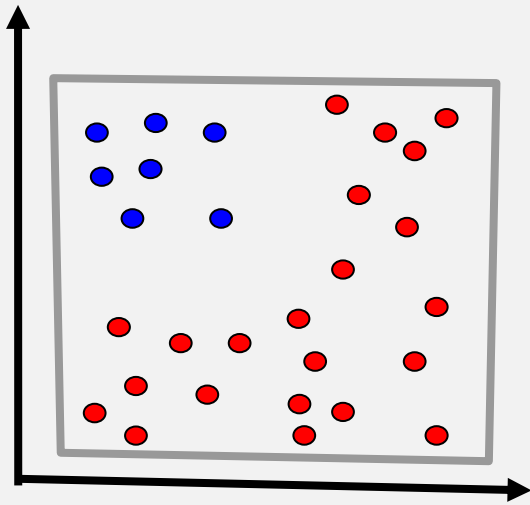
Asociar



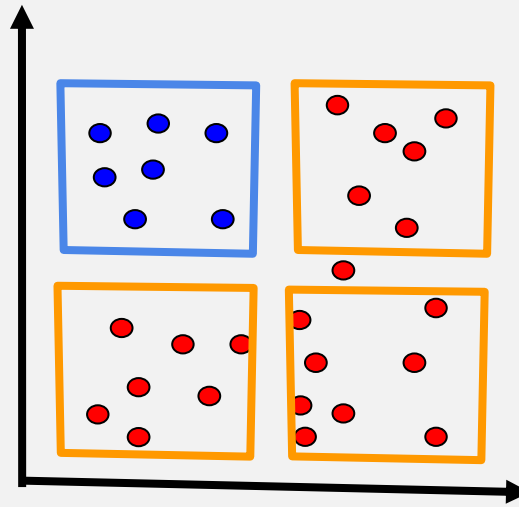
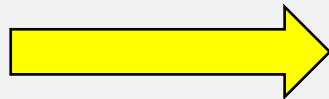
Mover



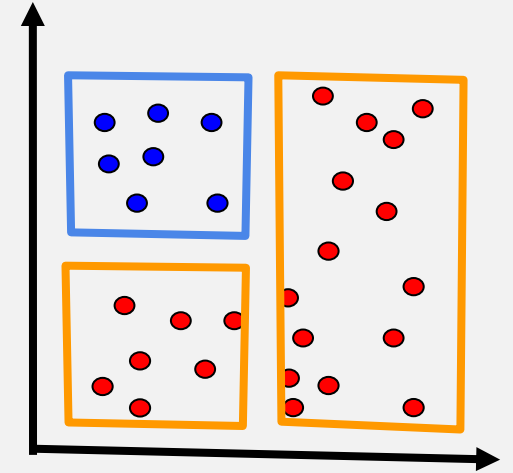
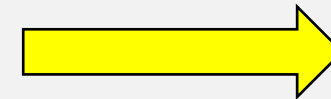
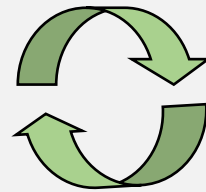
4. Marco Teórico: Divide y Conquista



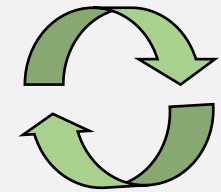
1 hiper-caja
+10%



Dividir (2^i)



Unir



4. Marco Teórico: *Softmax* y Funciones de Error

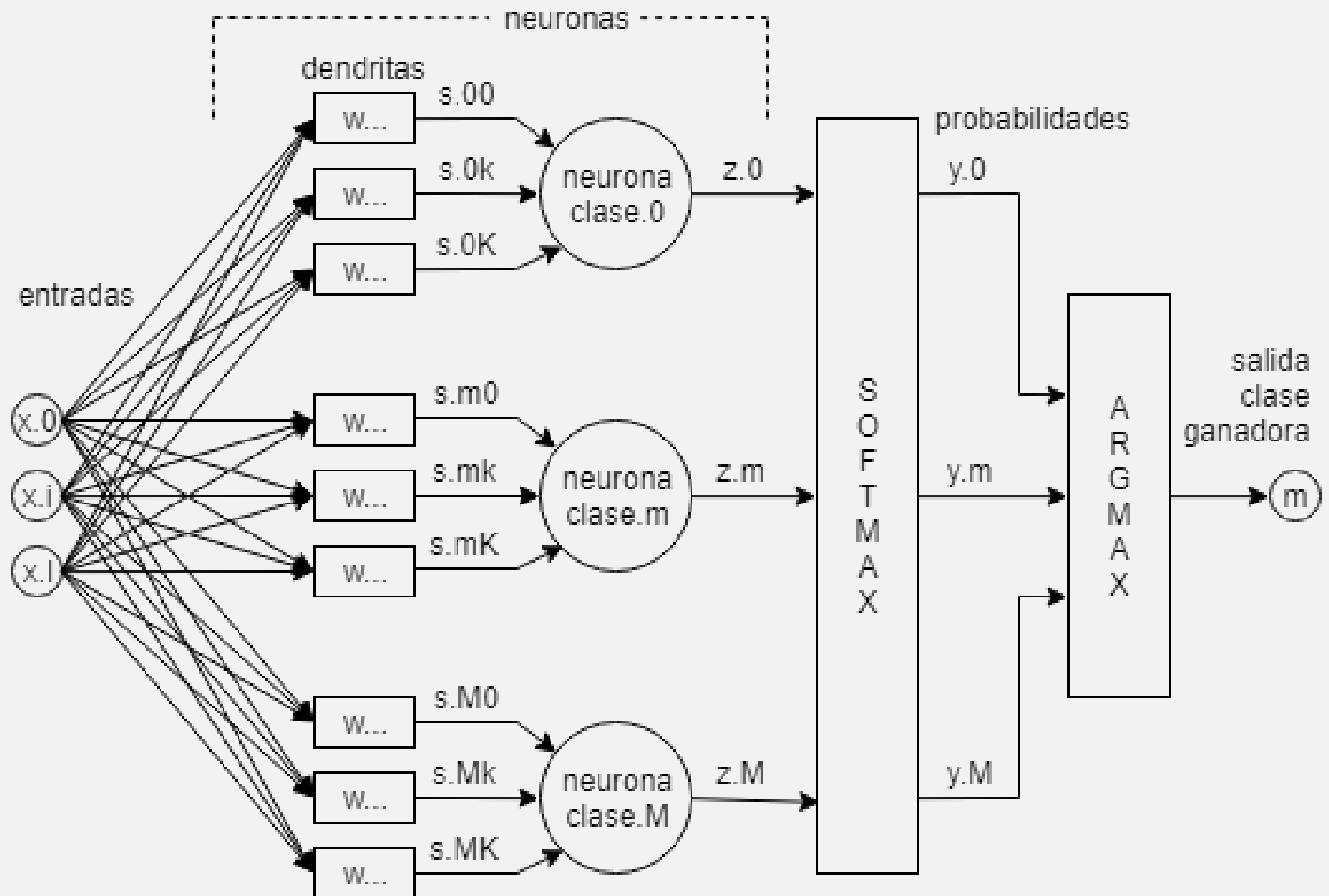
$$y_m = e^{z_m} / \sum_{n=0}^M e^{z_n}$$

$$\frac{\partial y_m}{\partial z_m} = y_m \cdot (1 - y_m)$$

Precisión y Logístico

$$\varepsilon = \frac{1}{P} \cdot \sum_{p=1}^P \begin{cases} 0, & \text{if } \operatorname{argmax}(y_{mp}) = y_{d_p} \\ 1, & \text{otherwise} \end{cases}$$

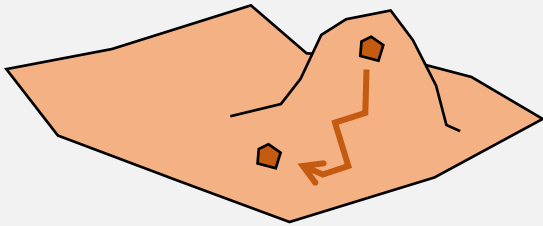
$$\varepsilon = \frac{-1}{P} \cdot \sum_{p=1}^P y_{d_p} \cdot \log(y_p) + (1 - y_{d_p}) \cdot \log(1 - y_p)$$



4. Marco Teórico: Gradiente Descendente Estocástico

$$u = \beta \cdot u + \frac{\partial \varepsilon}{\partial w}$$

$$w = w - \alpha \cdot u$$



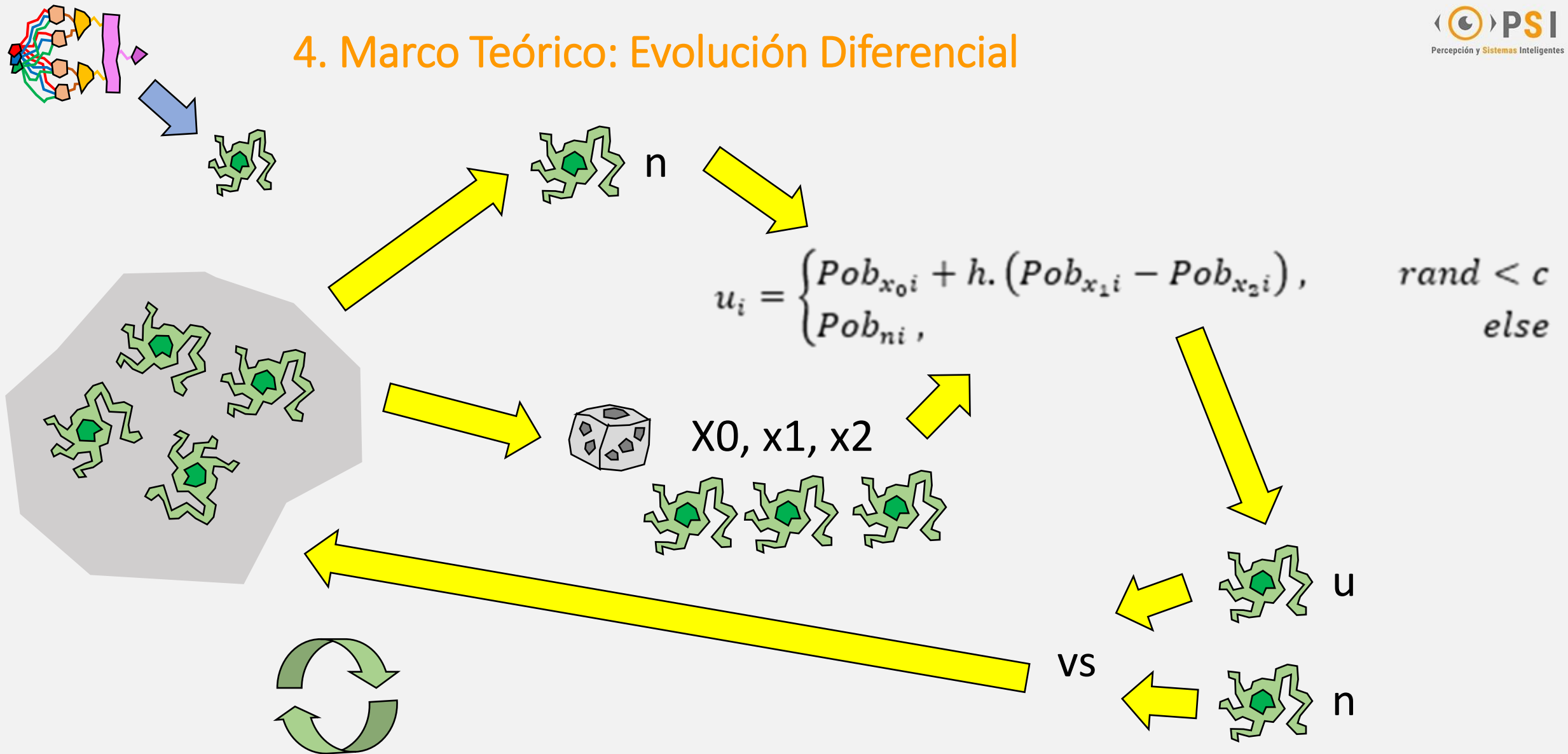
$$s_{mk} = \bigvee_{i=0}^I \bigvee_{l=0:L}^{1:H} (-1)^l \cdot (x_i + w_{mki}^l)$$

$$z_m = \bigwedge_{k=0}^{K_m} s_{mk}$$

$$y = \operatorname{argmax}(z_m)$$

$$\frac{\partial \varepsilon}{\partial w_{mki}^l} = \frac{1 - y_d}{\ln(10)} \cdot \begin{cases} 1, & l = L \\ -1, & l = H \end{cases}$$

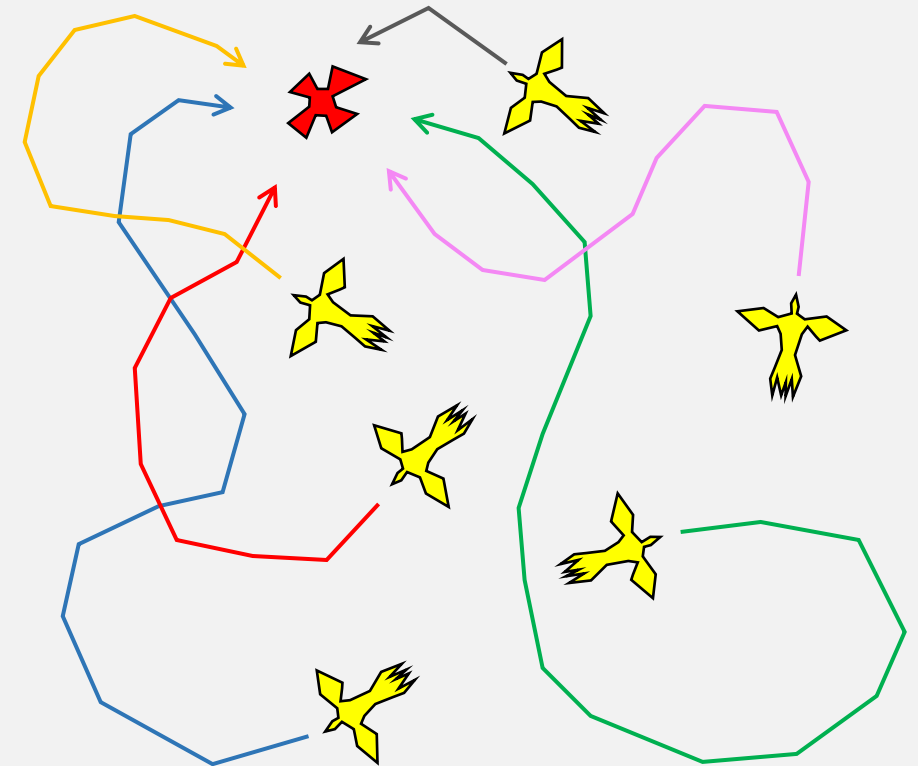
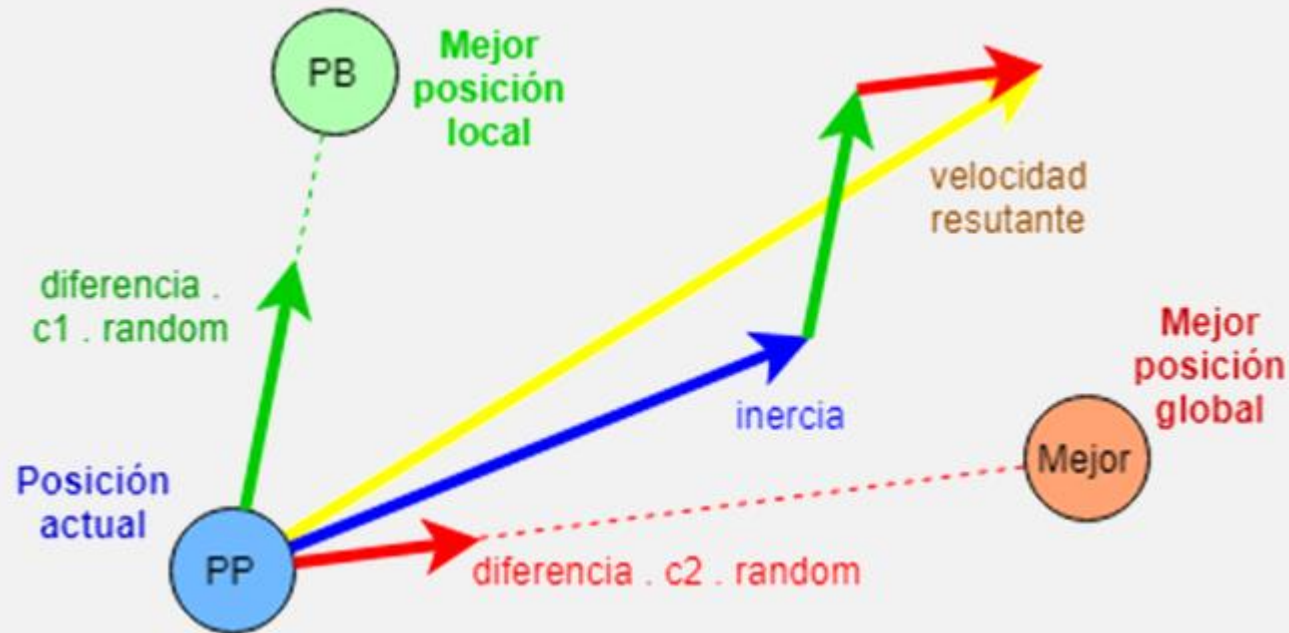
4. Marco Teórico: Evolución Diferencial



4. Marco Teórico: Optimización por Enjambre de Partículas

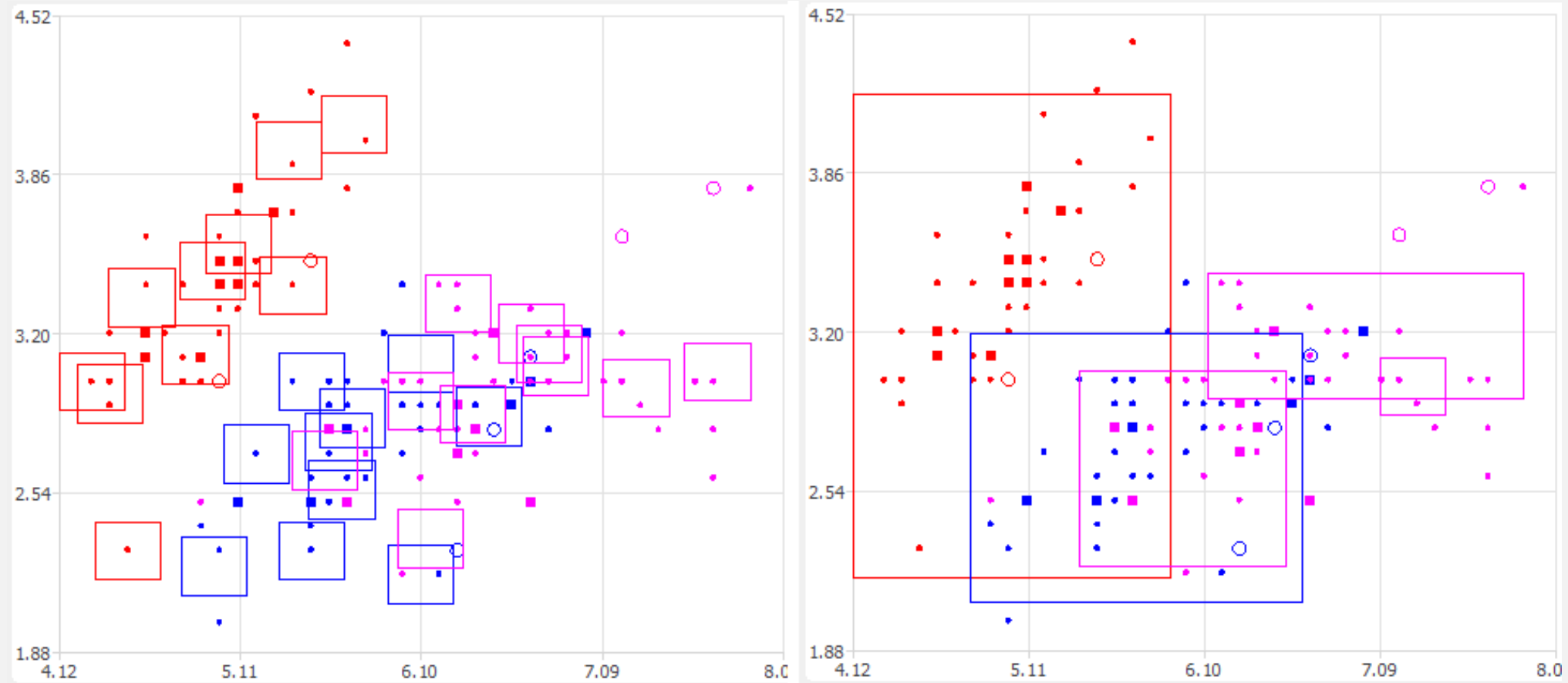
$$PV_n = c3.PV_n + c1.rand.(PB_n - PP_n) + c2.rand.(Mejor - PP_n)$$

$$PP_n = PP_n + PV_n$$



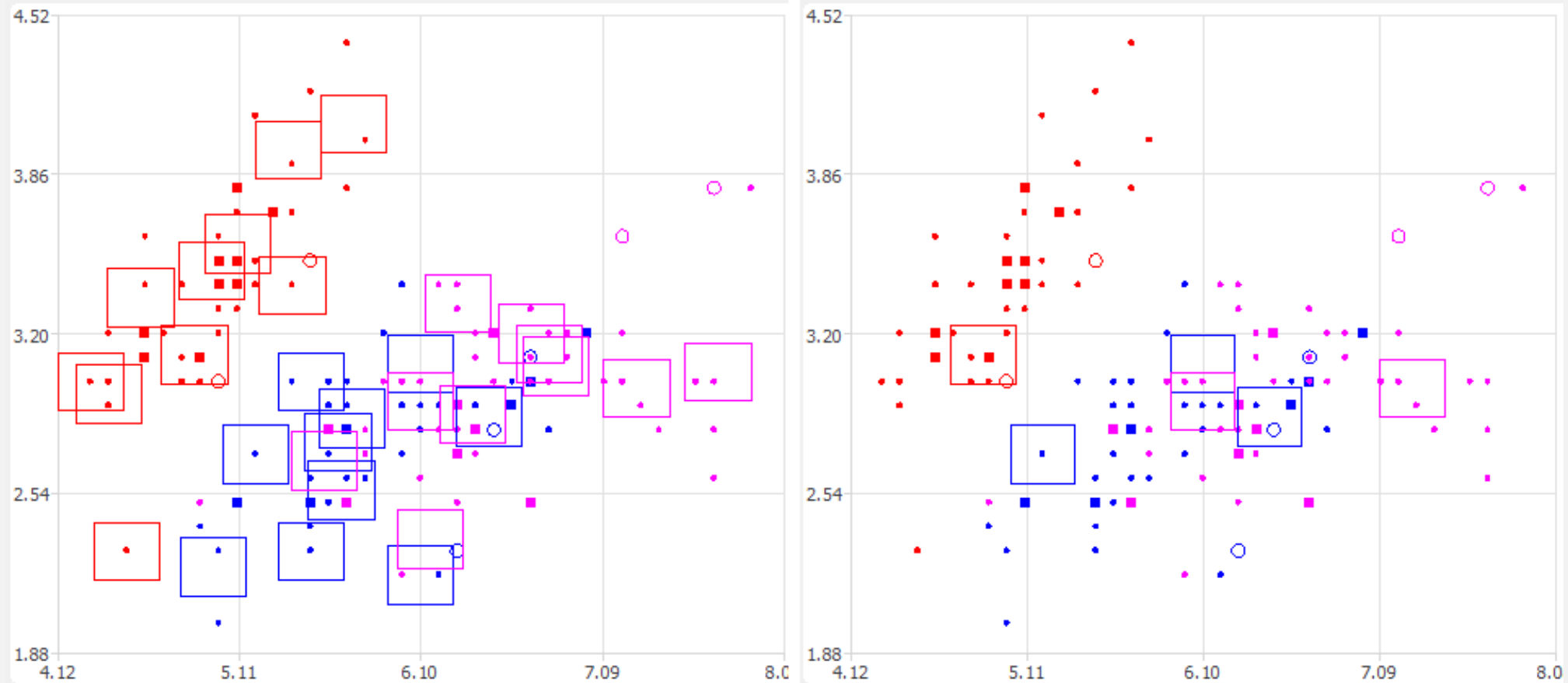
4. Marco Teórico: Optimización Post-Entreno

Unión



4. Marco Teórico: Optimización post-entreno

Eliminación

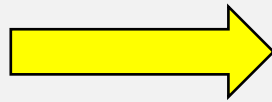


4. Marco Teórico: Procesamiento de Patrones

Matriz

```
#, #, #, 0
#, #, #, 0
#, #, #, 0
#, #, #, 0
#, #, #, 1
#, #, #, 1
#, #, #, 1
#, #, #, 1
#, #, #, 1
#, #, #, 1
#, #, #, 2
#, #, #, 2
#, #, #, 2
```

Mezcla

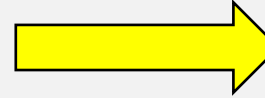


Matriz

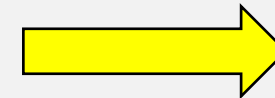
```
#, #, #, 1
#, #, #, 1
#, #, #, 2
#, #, #, 0
#, #, #, 1
#, #, #, 2
#, #, #, 0
#, #, #, 2
#, #, #, 0
#, #, #, 1
#, #, #, 1
#, #, #, 1
#, #, #, 0
```

Entreno

Baches



Validación

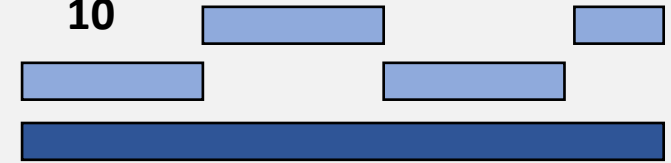


Testeo

10

35

Baches



4. Marco Teórico: Procesamiento de Patrones

Z-score

$$v = \frac{v - prom}{desvstd}$$

Min-Max

$$v = 2 \cdot \frac{v - min}{max - min} - 1$$

```
Patrones: TiposDeRanas
Salidas: RanaAgridulce, RanaMaliciosa
Entradas: Longitud(cm), Grosor(cm), Peso(g)
4.5, 6.2, 9.8, 0
4.7, 5.8, 6.9, 1
5.1, 5.9, 7.7, 0
4.2, 6.0, 7.0, 1
```

 Relativo

```
Patrones: TiposDeRanas
Salidas: RanaAgridulce, RanaMaliciosa
Entradas: Longitud(cm), Grosor(cm), Peso(g)
4.5, 6.2, 9.8, 0
4.7, 5.8, 6.9, 1
5.1, 5.9, 7.7, 0
4.2, 6.0, 7.0, 1
```

 Absoluto

4. Marco Teórico: Matrices de confusión

	1	2	3	4	5	6	7
1 PREDICHOS →	P0	P1	P2	Total	Sensibili.	Valor-F	
2 R0: ZancudoAgrio	39	5	2	46	84%	75%	
3 R1: Azuladito	7	32	1	40	80%	78%	
4 R2: Purpuroso	11	5	24	40	60%	71%	
5 Total	57	42	27	126			
6 Exactitud	68%	76%	88%		Precisión	75%	
7 Especificidad	77%	88%	96%		Kappa	0.628	

Porcentaje de los datos de dicha clase que fueron correctamente clasificados.

Media armónica entre la Exactitud y la Sensibilidad, relaciona ambas mediciones.

Probabilidad de que la predicción entregada por el detector pertenezca realmente a dicha clase.

Grado de acuerdo entre dos mediciones (reales vs predichos) tomando en consideración al azar.

Porcentaje de datos correctamente clasificados.

5. Solución Propuesta: Programación de la *DMNN*

```

1  import numpy as np
2  def EjecutarRed(entradas, pesW, numK):
3      X = entradas
4      while X.size < pesW.size / 2:
5          X = np.hstack((X, entradas))
6      W = pesW.copy().reshape(-1, 2)
7      WH = W[:, 0] - X
8      WL = X - W[:, 1]
9      Wmki = np.minimum(WH, WL)
10     Wmki = Wmki.reshape(-1, entradas.size)
11     Smk = Wmki.min(axis=1)
12     Zm = np.zeros(numK.size)
13     n = 0
14     for m in range(Zm.size):
15         Zm[m] = Smk[n:(n + numK[m])].max()
16         n += numK[m]
17     y = np.argmax(Zm)
18     return y

```



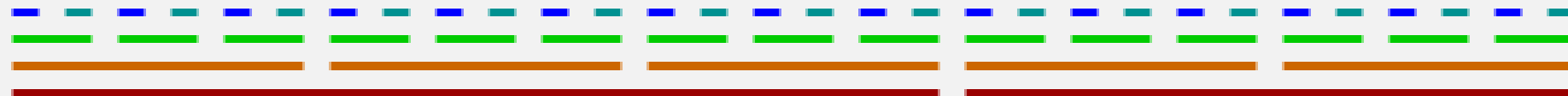
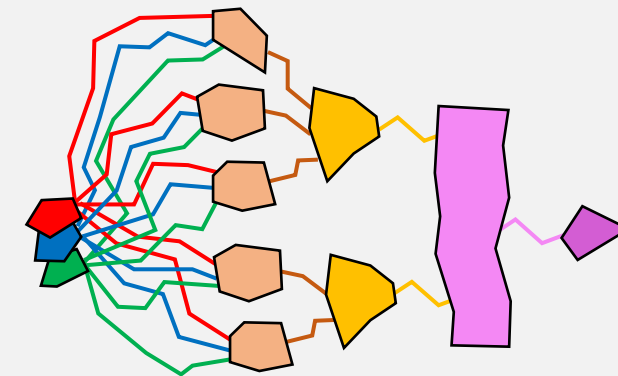
Python
3.7.2



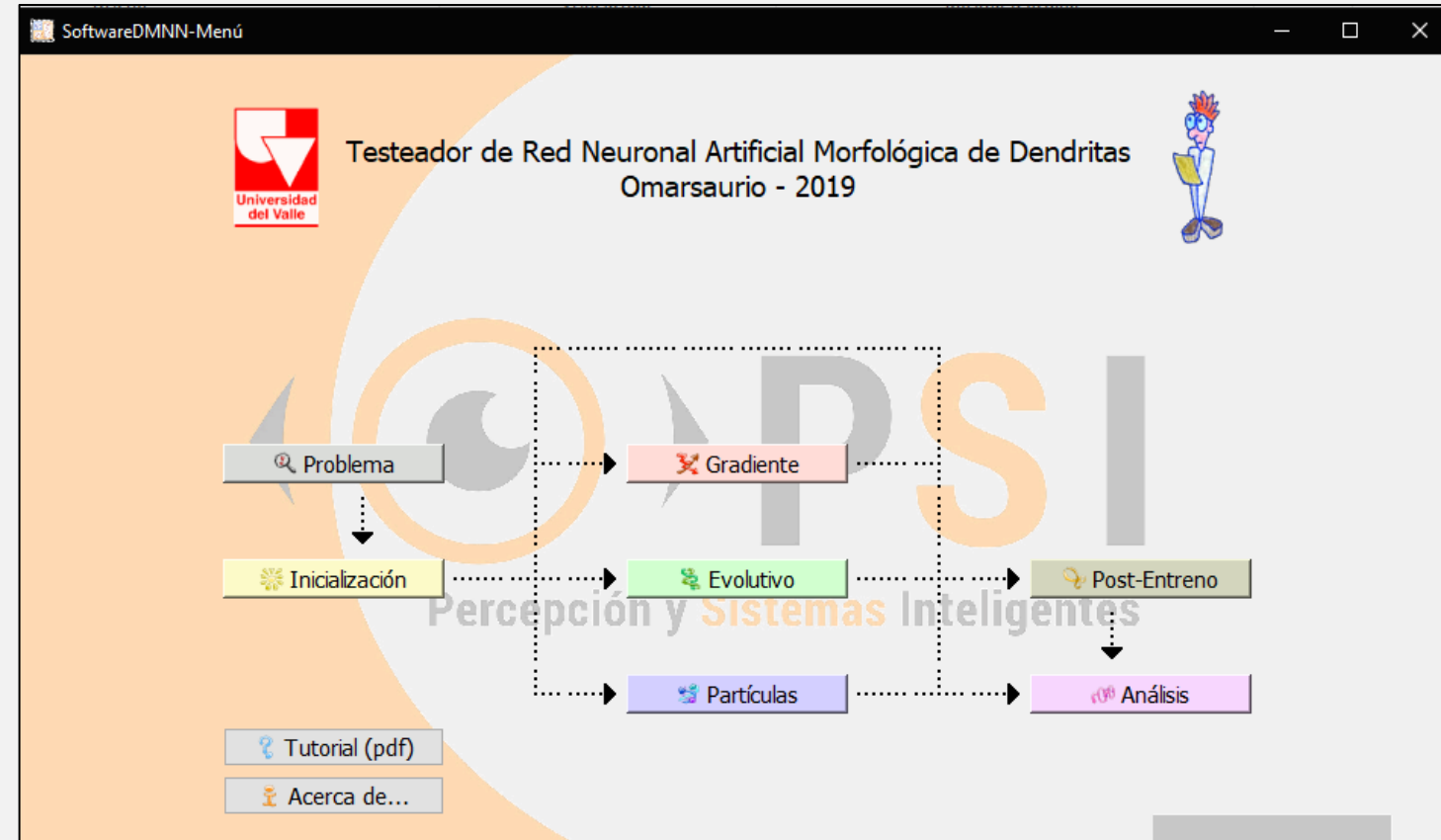
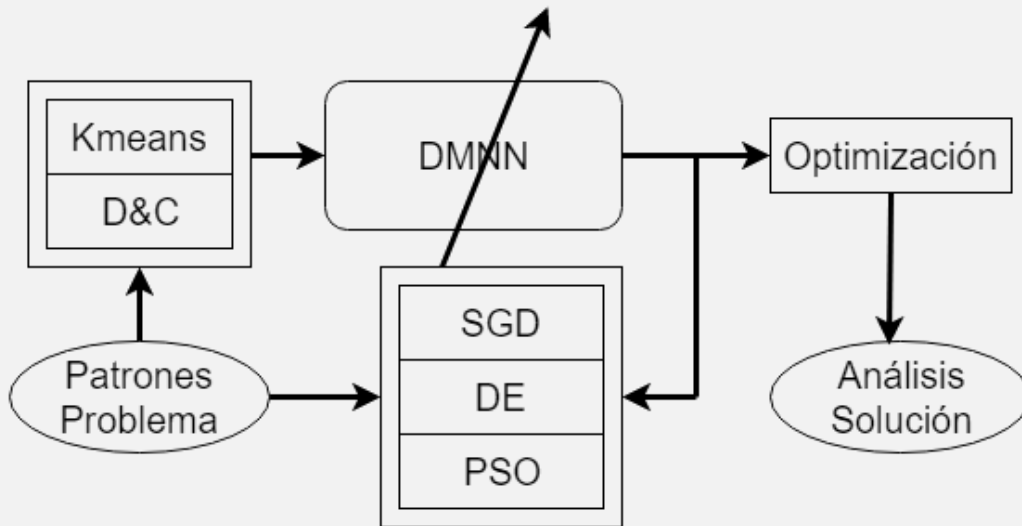
Numpy
1.16.1



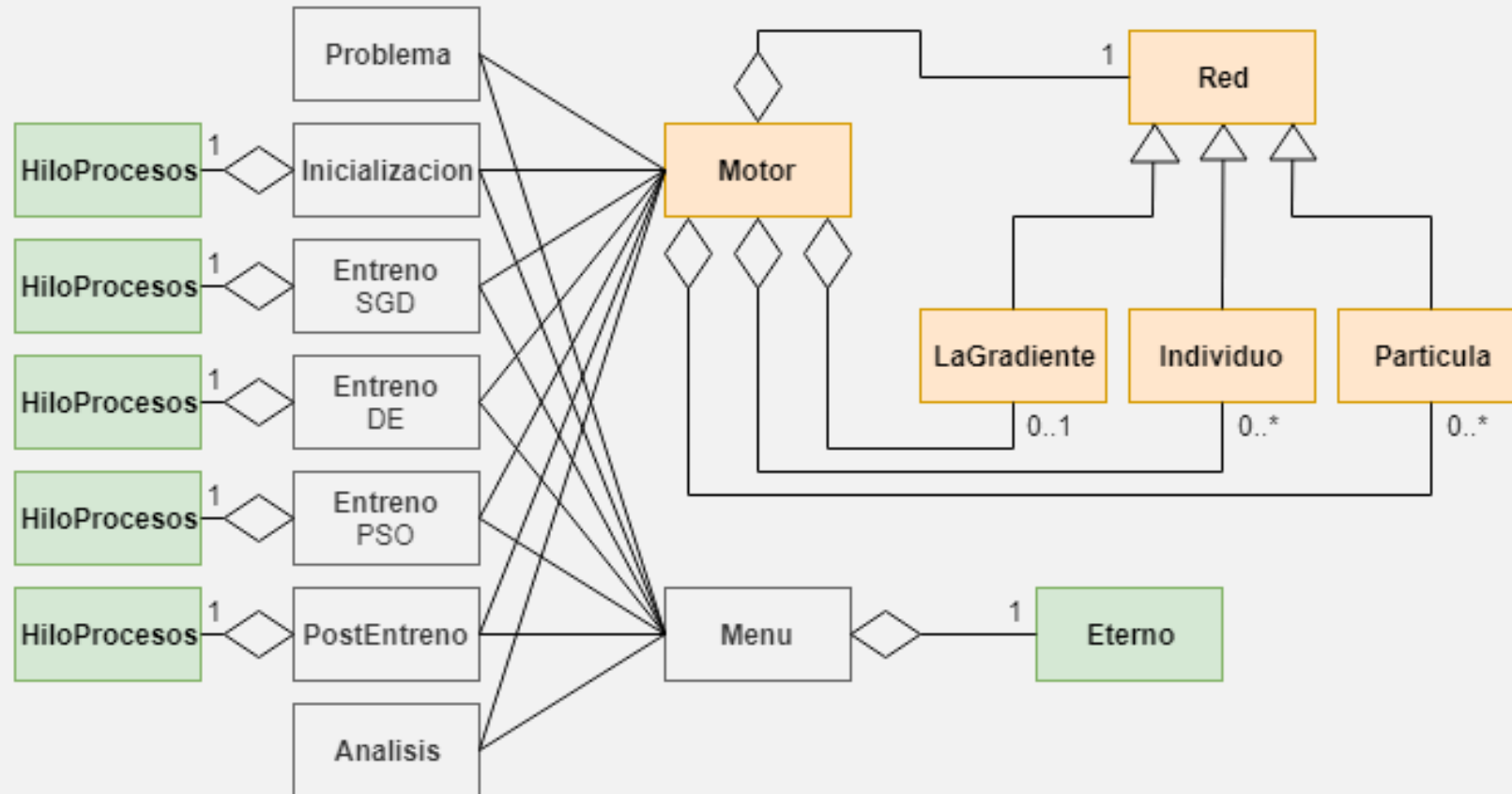
PyQt5
5.12.1



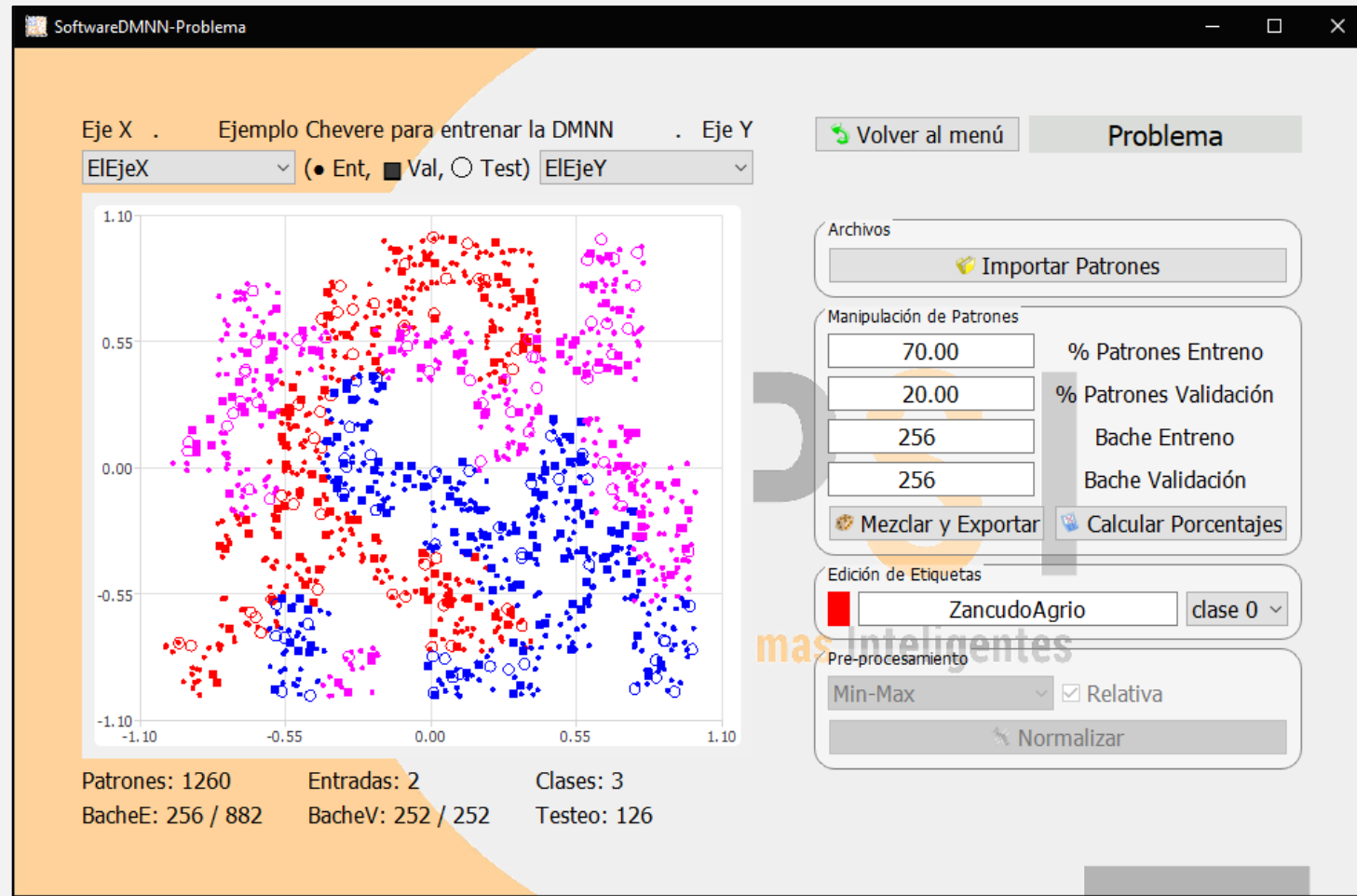
5. Solución Propuesta: Implementación del *software*



5. Solución Propuesta: Implementación del *software*

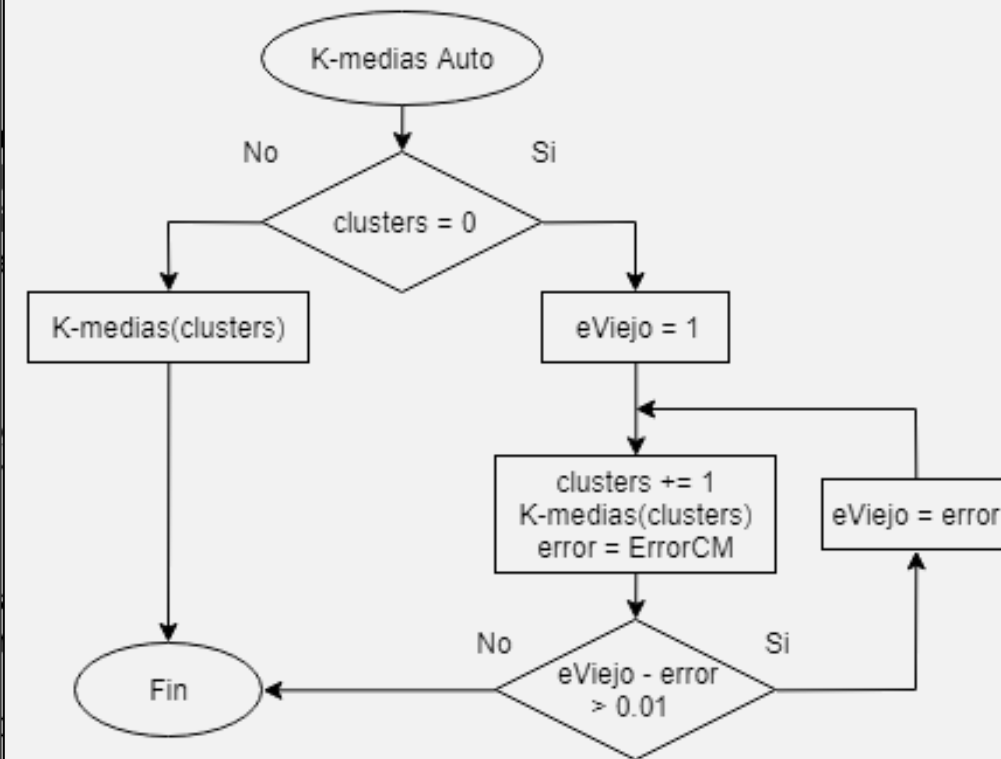
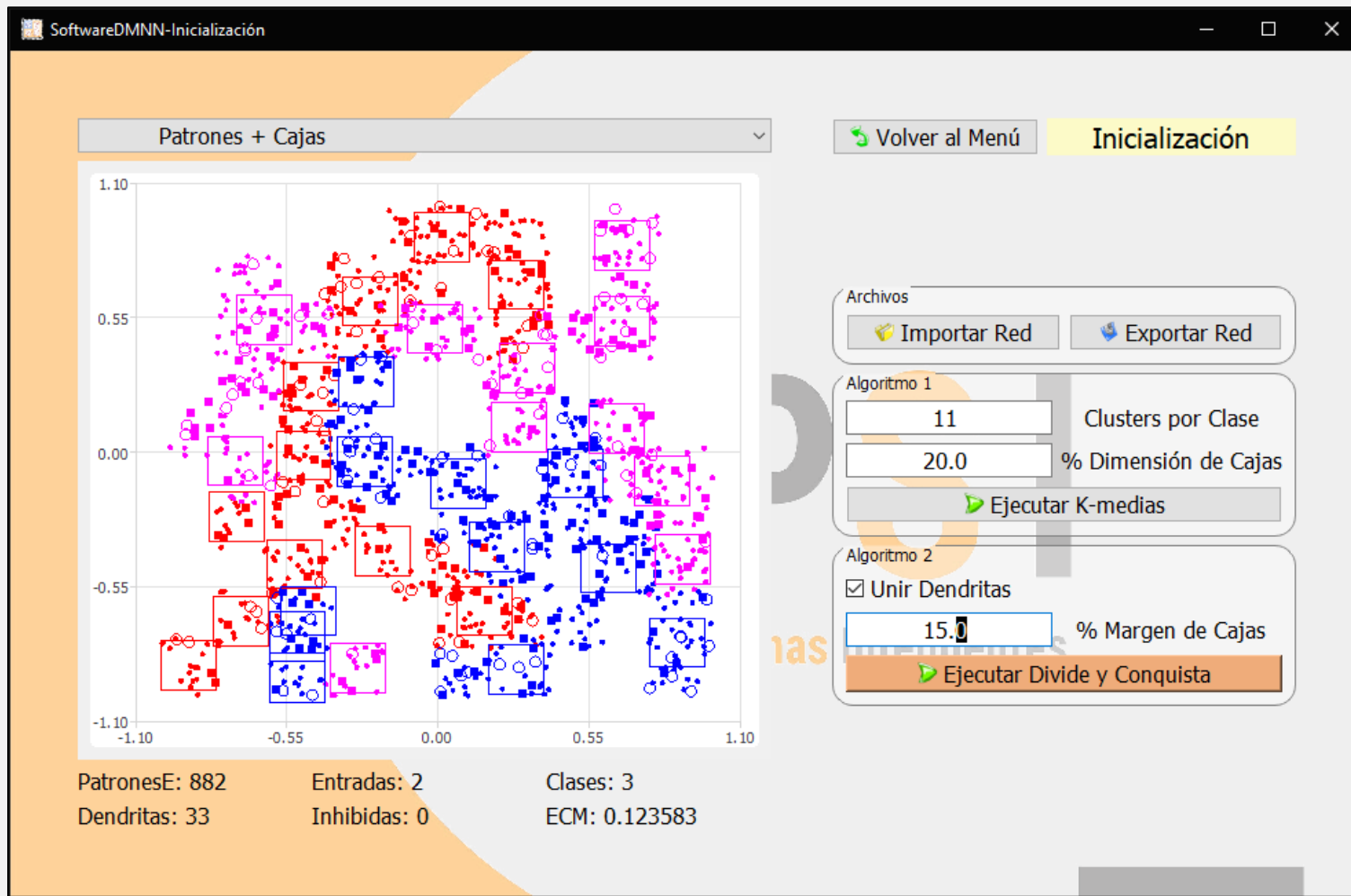


5. Solución Propuesta: Implementación del *software*

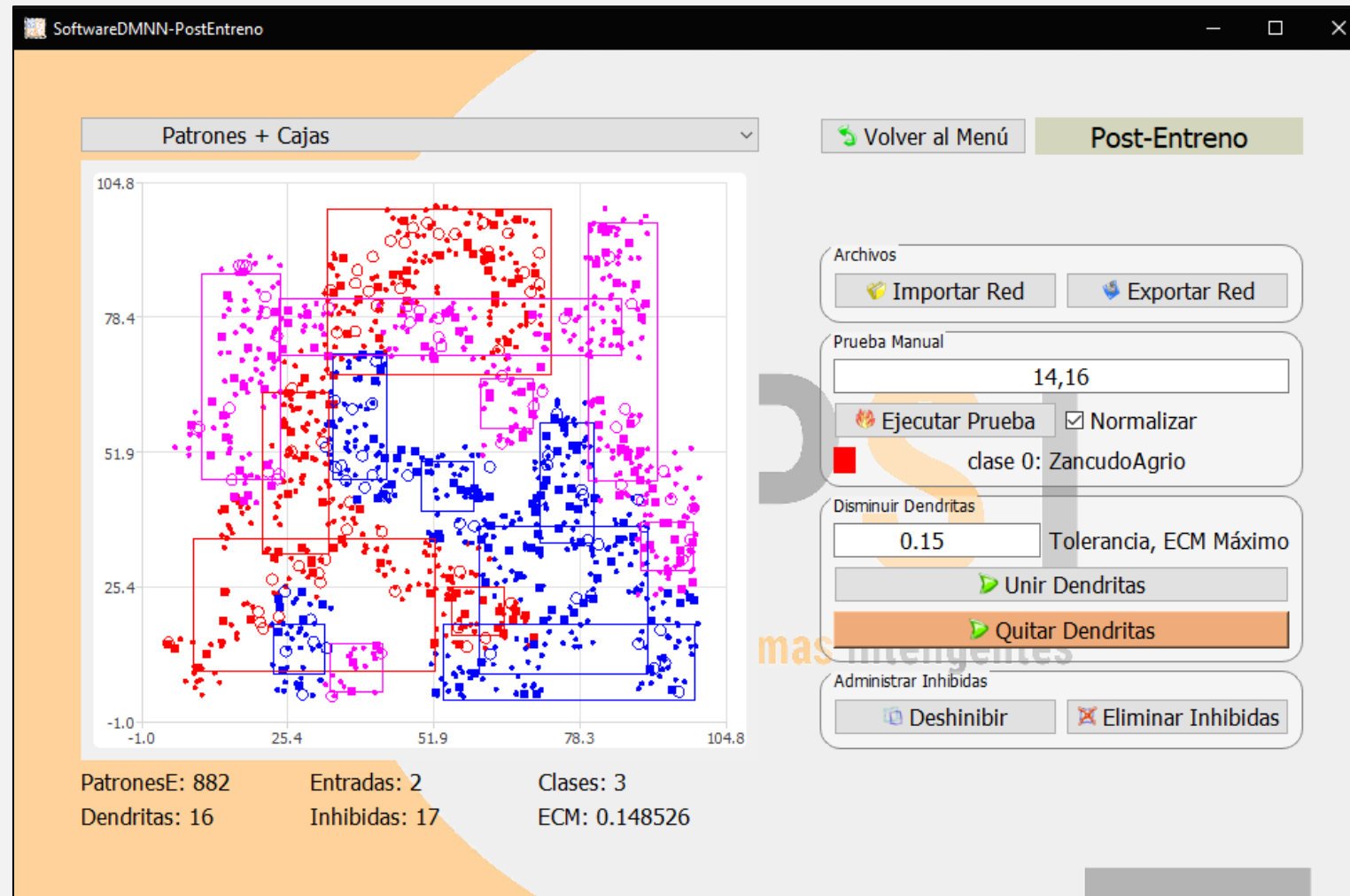


TXT
XML

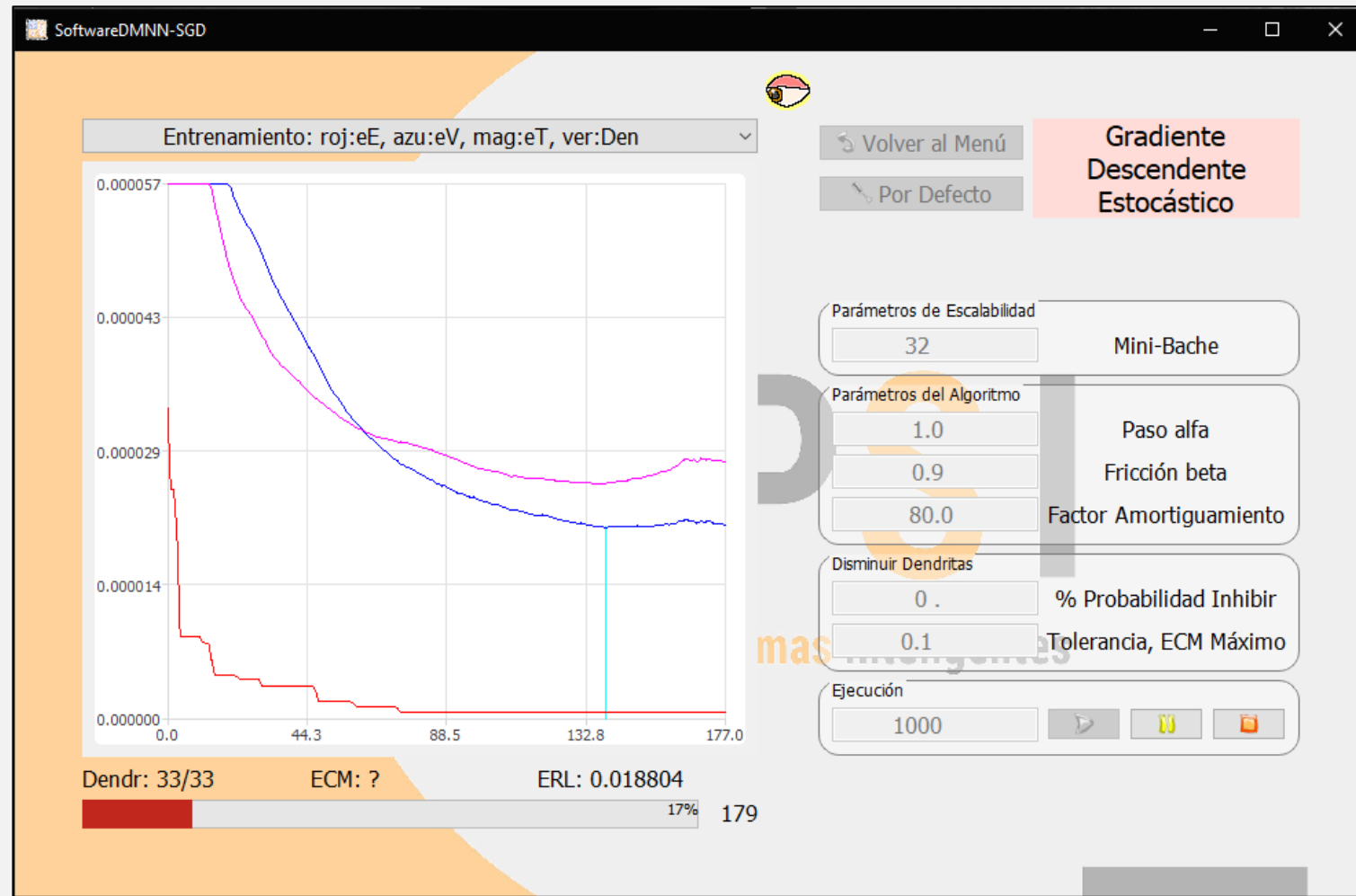
5. Solución Propuesta: Implementación del *software*



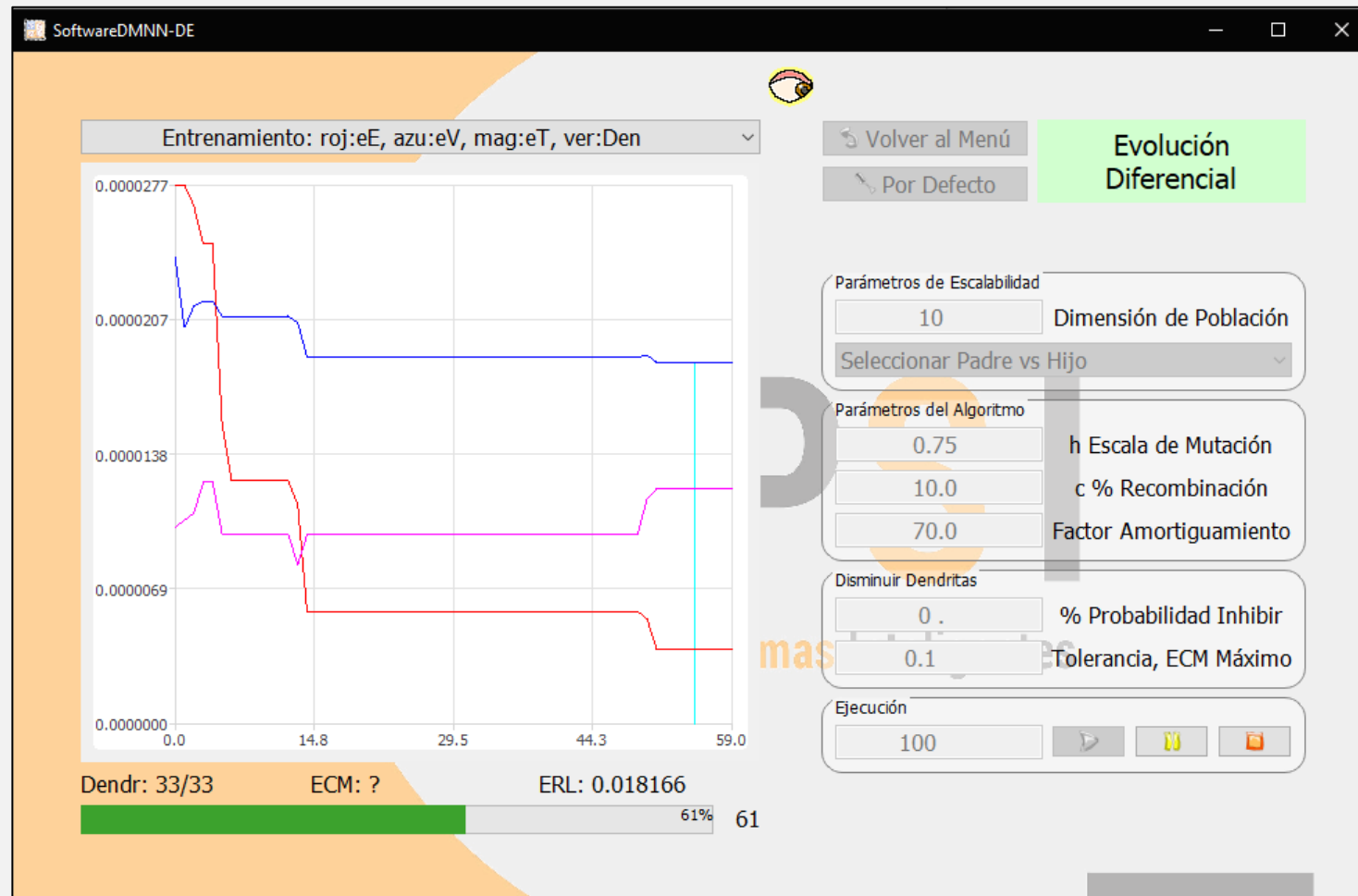
5. Solución Propuesta: Implementación del *software*



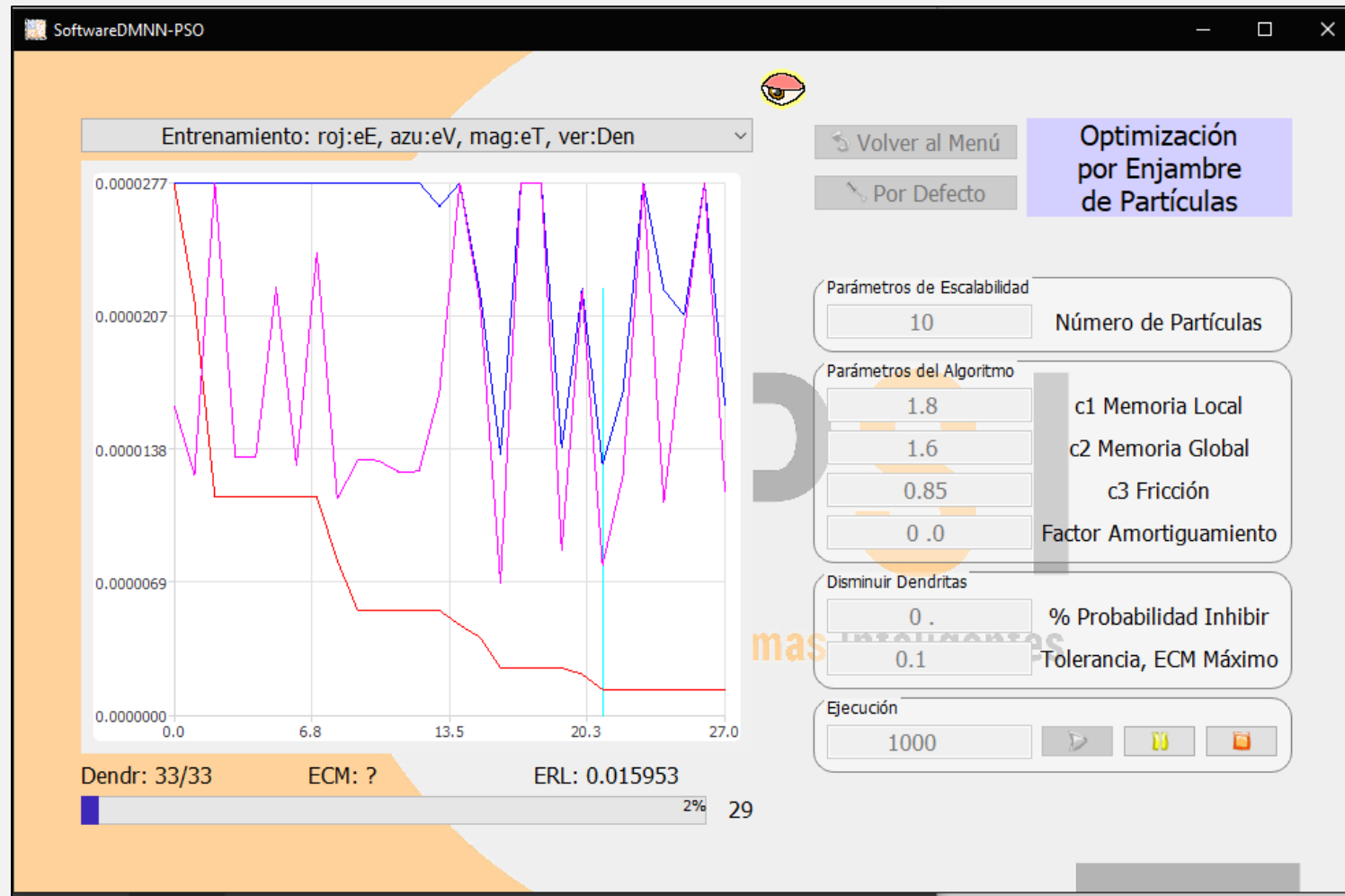
5. Solución Propuesta: Implementación del *software*



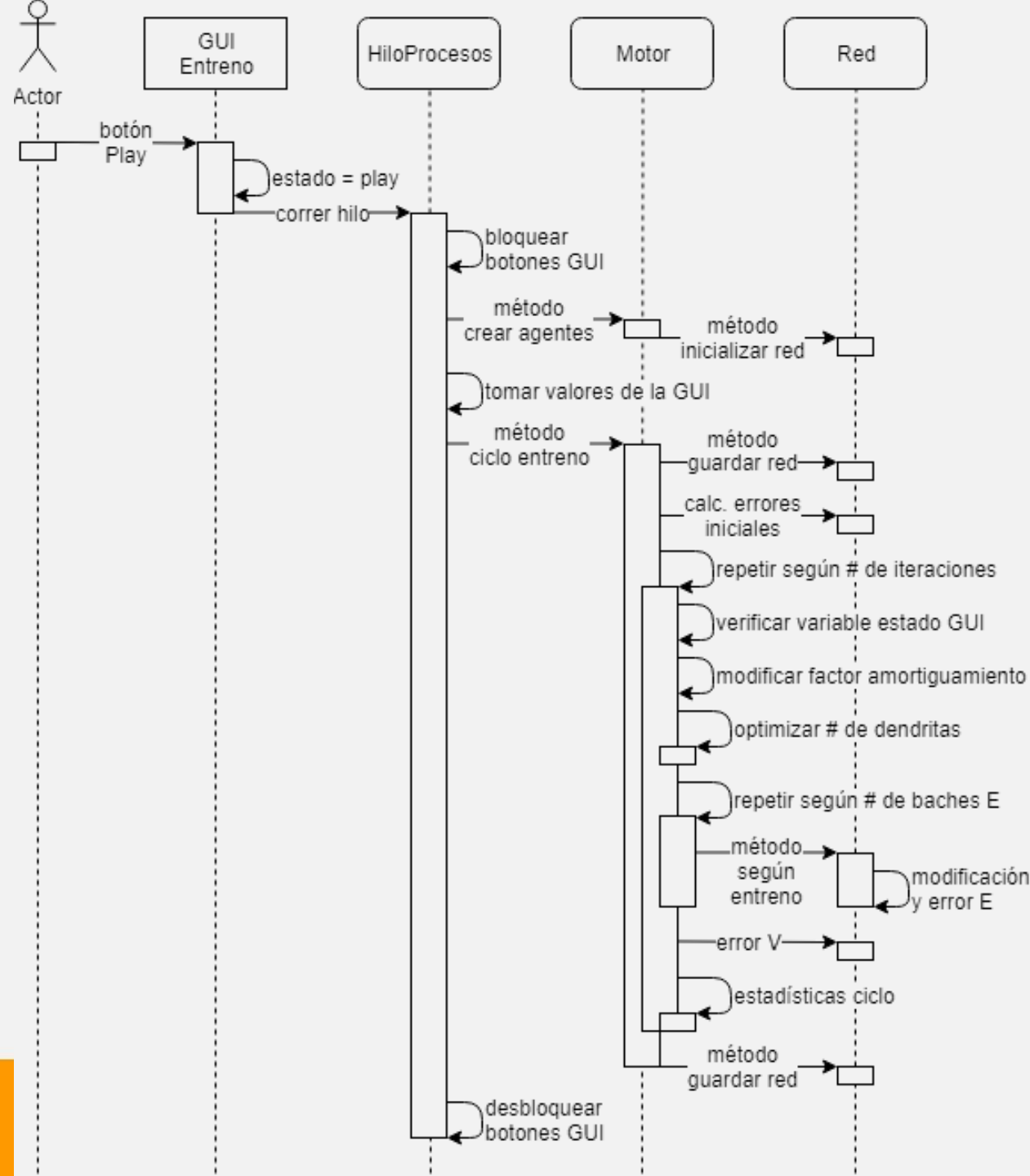
5. Solución Propuesta: Implementación del *software*



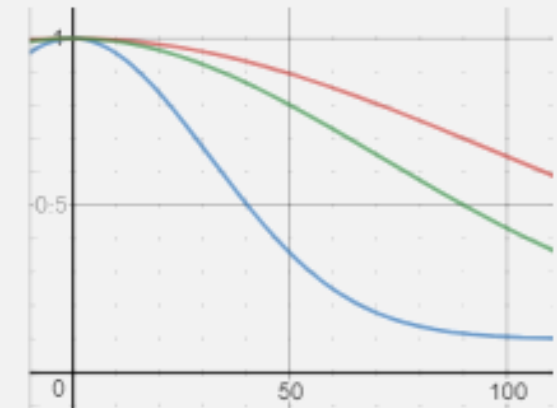
5. Solución Propuesta: Implementación del *software*



5. Solución Propuesta: Implementación del *software*



$$\alpha = 0.1 + 0.9 \cdot \exp \left(\frac{-\text{épocas}^2}{\text{factor} \cdot 2 \cdot \text{total_épocas}^2} \right)$$



5. Solución Propuesta: Implementación del *software*

SoftwareDMNN-Análisis

Matriz de Confusión

	1	2	3	4	5	6	7
1 PREDICHOS →	P0	P1	P2	Total	Sensibili.	Valor-F	
2 R0: ZancudoAgrio	39	5	2	46	84%	75%	
3 R1: Azuladito	7	32	1	40	80%	78%	
4 R2: Purpuroso	11	5	24	40	60%	71%	
5 Total	57	42	27	126			
6 Exactitud	68%	76%	88%		Precisión	75%	
7 Especificidad	77%	88%	96%		Kappa	0.628	

Patrones: 1260 Entradas: 2 Clases: 3
Entreno: 882 Validación: 252 Testeo: 126

Volver al Menú **Análisis**

Resultados Básicos

Dendritas: 12

ECM Testeo: 0.246032

Calculos Estadísticos

Estadísticas Testeo

Precisión: 0.753968
Kappa: 0.627787

Clase 0: ZancudoAgrio

Exactitud: 0.684211
Sensibilidad: 0.847826
Especificidad: 0.775
Valor-F: 0.757282

Archivos

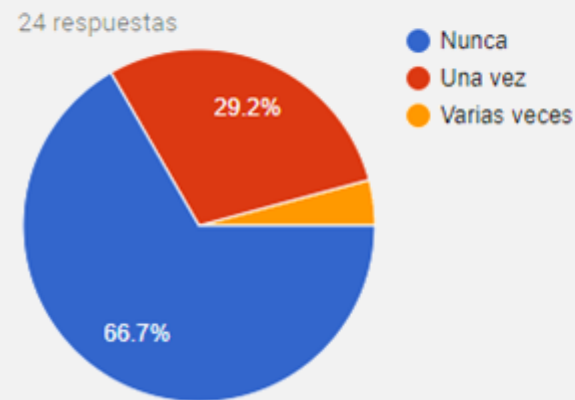
Exportar Matrices de Confusión
Exportar Resultados de Entradas

6. Resultados: Prueba de Usabilidad

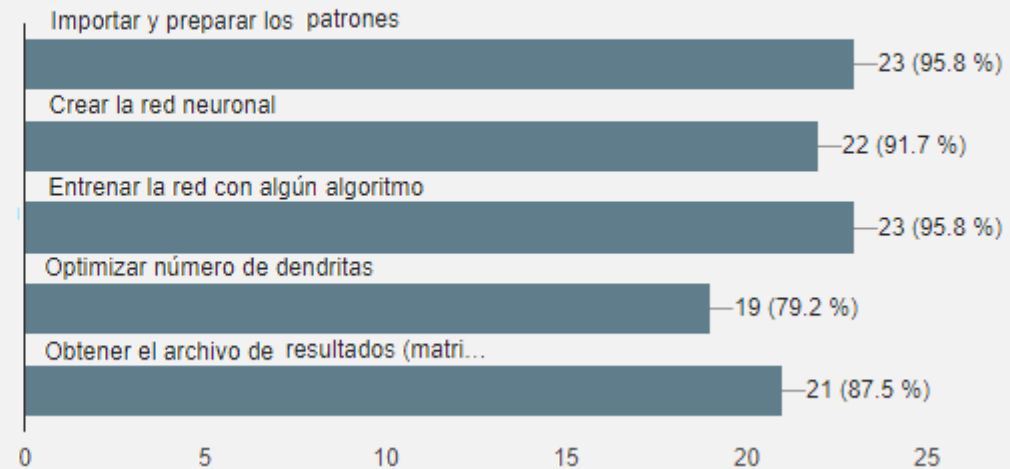
1-¿Nivel de conocimiento sobre problemas de clasificación?



3-¿Se presentó algún error que requirió re-ejecutar el software?



2-Marque las tareas que logró realizar 24 respuestas



4-¿Utilizó el tutorial (pdf) que incluye el software?



6. Resultados: Prueba de Usabilidad

5-¿Qué tan intuitiva fue la GUI llamada Problema?

24 respuestas



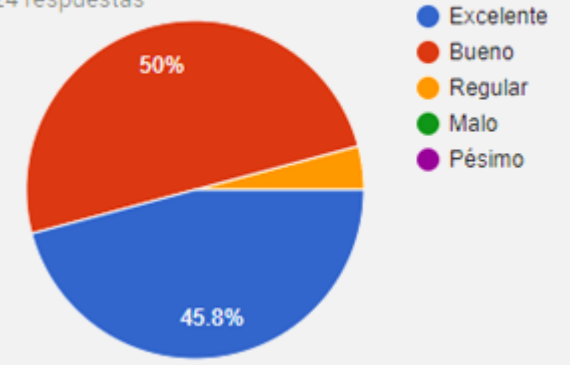
6-¿Qué tan intuitiva fue la GUI llamada Inicialización?

24 respuestas



10-¿Cómo calificarías al software?

24 respuestas



9-¿Qué tan intuitiva fue la GUI llamada Análisis?

24 respuestas



7-¿Qué tan intuitiva fue alguna de las GUI: Gradiente, Evolutivo, Partículas?

24 respuestas



8-¿Qué tan intuitiva fue la GUI llamada Post-Entreno?

24 respuestas



6. Resultados: Pruebas con *datasets*

Información datasets

Dataset	Total	Dimensión	Clases
Iris	150	4	3
Wine	178	13	3
Glass	214	9	6
Mammographic Mass	961	5	2
Letter Recognition	20000	16	26
Espiral2	2028	2	2
Ejemplito	1260	2	3
PayasoLander	2032	5	6

Z-score relativo

Mejor solución alcanzada

Dataset	#den	%preciE	%preciT
Iris	5	98.33	96.67
Wine	5	88.03	80.56
Glass	23	95.32	60.47
Mammographic Mass	10	78.31	71.72
Letter Recognition	254	91.81	82.98
Espiral2	40	99.82	97.78
Ejemplito	39	97.02	90.48
PayasoLander	68	40.10	35.29

Naranja > 80%

Rojo > 90%

6. Resultados: Pruebas con *datasets*

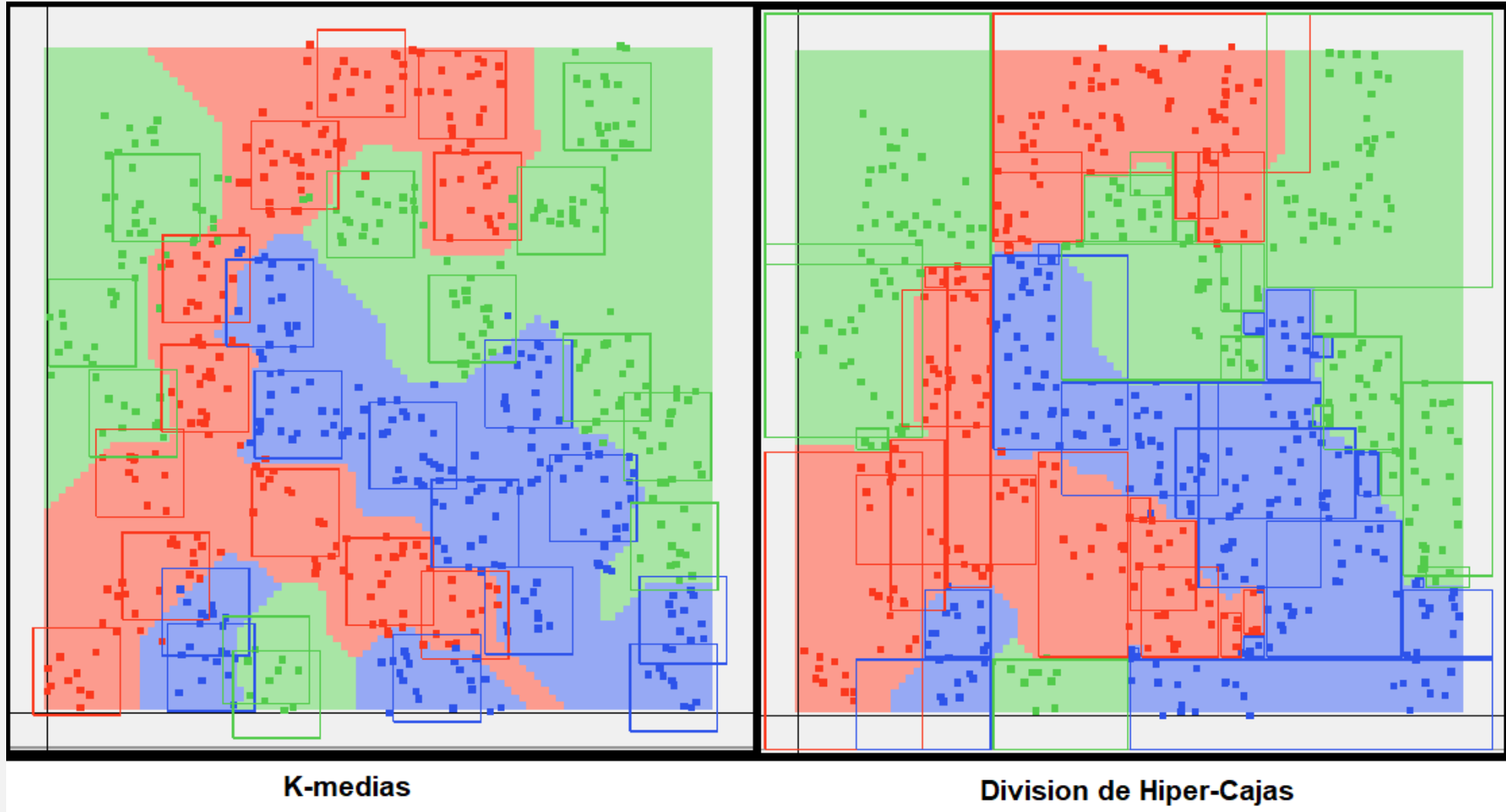
Inicialización

Dataset	K-medias			D&C		
	#den	%preciE	%preciT	#den	%preciE	%preciT
Iris	12	96.67	93.33	12	97.50	96.67
Wine	31	73.94	72.22	24	96.48	77.78
Glass	40	78.82	63.64	44	84.73	63.64
Mammographic Mass	21	71.76	68.04	96	84.84	73.20
Letter Recognition	299	75.83	71.69
Espiral2	40	94.64	94.09	57	98.83	97.29
Ejemplito	45	91.37	88.49	60	94.94	86.51
PayasoLander	70	34.03	32.35	436	52.84	24.02

K-medias: auto, 10%, 3 intentos

DyC: 10% (... 3 días sin solución)

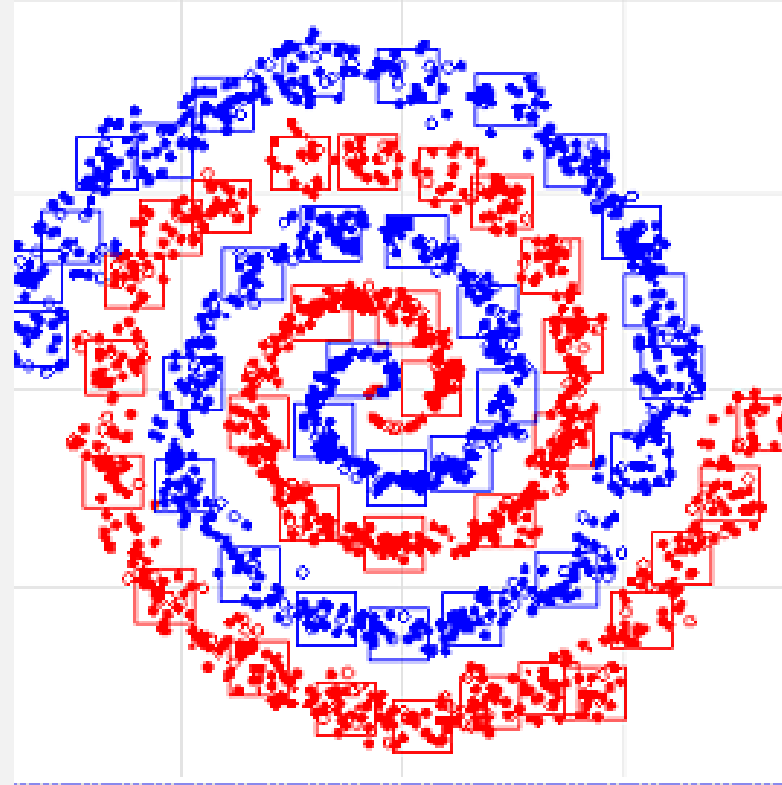
6. Resultados: Pruebas con *datasets*



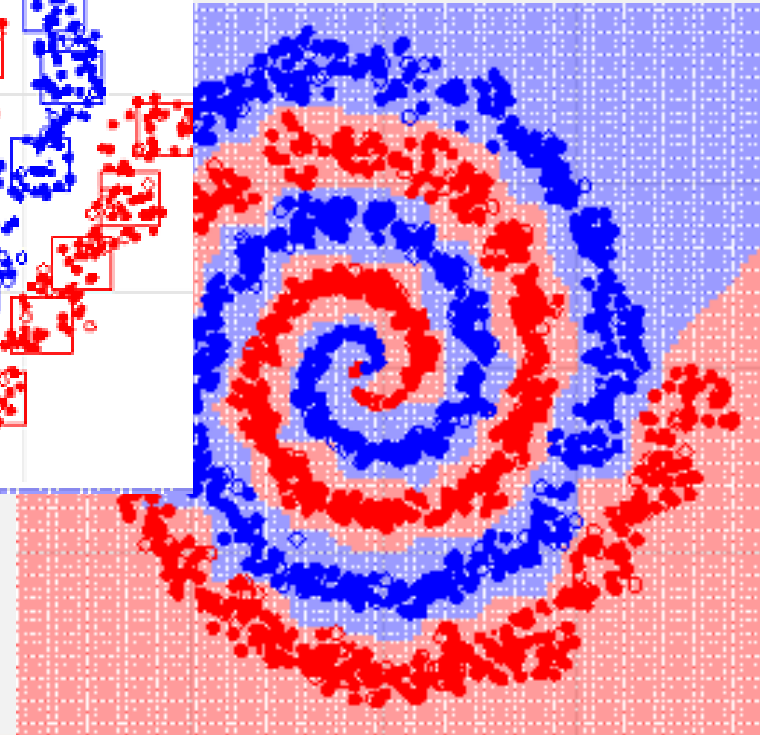
6. Resultados: Pruebas con *datasets*

Mammographic Mass

	1	2	3	4	5	6
1 PREDICHOS →	P0	P1	Total	Sensibili.	Valor-F	
2 R0: Benign	391	125	516	75%	77%	
3 R1: Malignant	108	337	445	75%	74%	
4 Total	499	462	961			
5 Exactitud	78%	72%		Precisión	75%	
6 Especificidad	75%	75%		Kappa	0.514	



Espiral2



6. Resultados: Pruebas con *datasets*

Comparación Algoritmos

Dataset	K-medias			SGD		DE		PSO	
	#den	%preciE	%preciT	%preciE	%preciT	%preciE	%preciT	%preciE	%preciT
Iris	9	95.00	96.67	98.33	96.67	97.50	96.67	96.67	100.0
Wine	41	85.21	83.33	86.62	83.33	85.92	83.33	86.62	83.33
Glass	29	71.92	36.36	95.57	63.64	72.91	45.45	73.40	45.45
Mammogra. Mass	10	73.96	72.16	73.96	72.16	73.73	73.20	76.39	77.32
Letter Recognition	130	64.15	61.97	83.80	75.66	64.09	61.76	64.00	62.03
Espiral2	40	95.50	93.60	97.35	94.83	95.62	94.09	95.56	94.83
Ejemplito	15	79.76	77.78	89.38	84.52	80.26	78.17	81.05	78.57
PayasoLander	106	36.54	29.90	40.54	29.90	36.65	30.39	37.25	29.90

6. Resultados: Pruebas con *datasets*

iris SGD

	1	2	3	4	5	6	7
1 PREDICHOS →	P0	P1	P2	Total	Sensibili.	Valor-F	
2 R0: Setosa	41	0	0	41	100%	100%	
3 R1: Versicolor	0	36	1	37	97%	97%	
4 R2: Virginica	0	1	41	42	97%	97%	
5 Total	41	37	42	120			
6 Exactitud	100%	97%	97%		Precisión	98%	
7 Especificidad	100%	98%	98%		Kappa	0.975	

iris PSO

	1	2	3	4	5	6	7
1 PREDICHOS →	P0	P1	P2	Total	Sensibili.	Valor-F	
2 R0: Setosa	50	0	0	50	100%	100%	
3 R1: Versicolor	0	48	2	50	96%	96%	
4 R2: Virginica	0	2	48	50	96%	96%	
5 Total	50	50	50	150			
6 Exactitud	100%	96%	96%		Precisión	97%	
7 Especificidad	100%	98%	98%		Kappa	0.96	

Glass SGD

	1	2	3	4	5	6	7	8	9	10
1 PREDICHOS →	P0	P1	P2	P3	P4	P5	Total	Sensibili.	Valor-F	
2 R0: Tipo1	66	4	0	0	0	0	70	94%	94%	
3 R1: Tipo2	2	71	1	1	1	0	76	93%	92%	
4 R2: Tipo3	0	2	15	0	0	0	17	88%	90%	
5 R3: Tipo4	1	0	0	12	0	0	13	92%	92%	
6 R4: Tipo5	1	0	0	0	8	0	9	88%	88%	
7 R5: Tipo6	0	0	0	0	0	29	29	100%	100%	
8 Total	70	77	16	13	9	29	214			
9 Exactitud	94%	92%	93%	92%	88%	100%		Precisión	93%	
10 Especificidad	97%	95%	99%	99%	99%	100%		Kappa	0.917	

Glass PSO

	1	2	3	4	5	6	7	8	9	10
1 PREDICHOS →	P0	P1	P2	P3	P4	P5	Total	Sensibili.	Valor-F	
2 R0: Tipo1	54	8	8	0	0	0	70	77%	71%	
3 R1: Tipo2	17	48	9	0	2	0	76	63%	71%	
4 R2: Tipo3	7	1	9	0	0	0	17	52%	40%	
5 R3: Tipo4	0	0	1	11	0	1	13	84%	91%	
6 R4: Tipo5	2	1	0	0	5	1	9	55%	62%	
7 R5: Tipo6	2	0	0	0	0	27	29	93%	93%	
8 Total	82	58	27	11	7	29	214			
9 Exactitud	65%	82%	33%	100%	71%	93%		Precisión	71%	
10 Especificidad	80%	92%	90%	100%	99%	98%		Kappa	0.624	

6. Resultados: Problema *Payasolander*



6. Resultados: Problema *Payasolander*

```
// físicas giro
ag = cg * m_giro
vg = limitar(vg * m_fg + ag * dt, -m_vg, m_vg)
pg = limitar(pg + vg * dt, -m_pg, m_pg)
// físicas movimiento
ax = -sen(pg) * cp * m_propulsion
ay = m_gravedad - cos(pg) * cp * m_propulsion
vx = limitar(vx * m_fv + vv + ax * dt, -m_vxy, m_vxy)
vy = limitar(vy * m_fv + ay * dt, -m_vxy, m_vxy)
px = limitar(px + vx * dt, -m_px, m_px)
py = limitar(py + vy * dt, -m_py, 0)
```



Algoritmos	#den	%preciE	%preciT	%Éxitos	%Maniobra
K-medias	94	34.20	32.32	36.67	70.00
K-medias + DE	106	36.65	30.39	26.67	56.67
K-medias	106	36.54	29.90	23.33	63.33
K-medias	208	42.56	28.92	20.00	80.00
D&C	834	87.31	21.08	10.00	36.67

6. Resultados: Problema *Payasolander*

	1	2	3	4	5	6	7	8	9	10
1	PREDICHOS →	P0	P1	P2	P3	P4	P5	Total	Sensibili.	Valor-F
2	R0: GiroD	18	82	4	39	31	4	178	10%	12%
3	R1: Nula	27	237	60	53	170	61	608	38%	36%
4	R2: Girol	3	72	39	4	44	36	198	19%	21%
5	R3: ImpGiroD	25	79	8	53	61	9	235	22%	23%
6	R4: Impulso	35	155	36	55	197	76	554	35%	34%
7	R5: ImpGirol	12	74	25	21	79	48	259	18%	19%
8	Total	120	699	172	225	582	234	2032		
9	Exactitud	15%	33%	22%	23%	33%	20%		Precisión	29%
10	Especificidad	94%	67%	92%	90%	73%	89%		Kappa	0.089

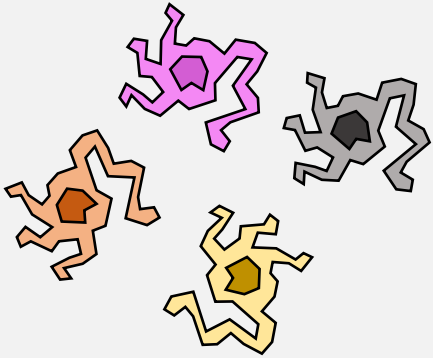
real.	GD	N	GI		sens.
rGD	135	253	25	413	33%
rN	170	759	233	1162	65%
rGI	40	269	148	457	32%
	345	1281	406	2032	
exac.	39%	59%	36%	preci.	51%

real.	N	I		sens.
rN	542	442	984	55%
rl	449	599	1048	57%
	991	1041	2032	
exac.	55%	58%	preci.	56%

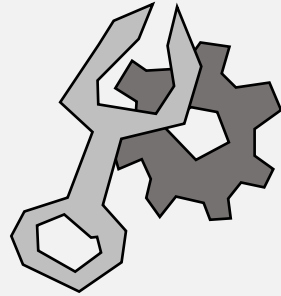
7. Conclusiones

- Reafirmada la superioridad de K-medias sobre Divide y Conquista.
- SGD brinda la mejor solución para la mayoría de los casos.
- PSO puede en algunos casos (como problemas muy discretos) superar a SGD, en general busca soluciones donde SGD y DE se estancan; ha mostrado tener también una buena capacidad de generalización, aunque una mala escogencia de sus parámetros fácilmente daña la estructura de las hiper-cajas.

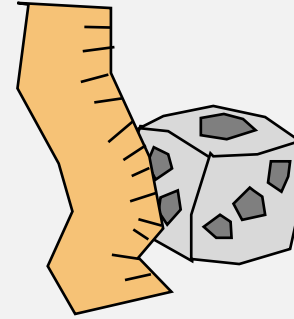
8. Trabajos futuros



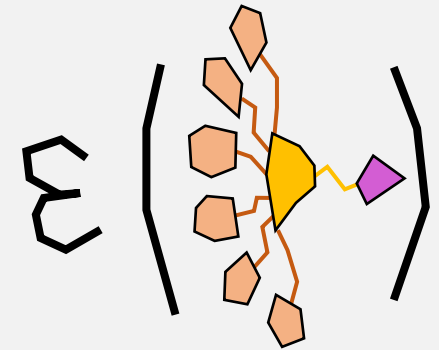
+ Algoritmos



Parámetros



Pruebas + Métricas

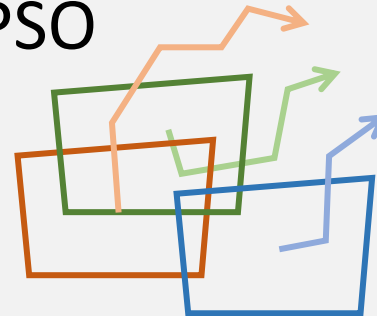


Error + Dendritas

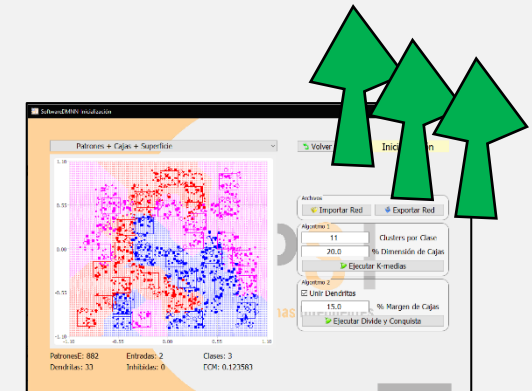


Regresión

PSO



Referencia



Mejoras + Paralelismo



Gracias

Anexo Video