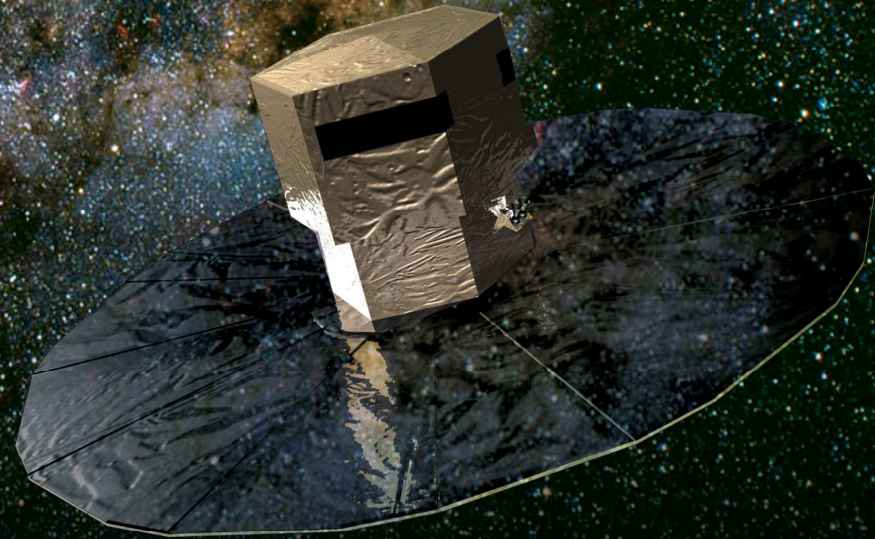


Técnicas de Inteligencia Artificial en Gaia

Bernardino Arcay
Minia Manteiga
Diego Ordoñez
Carlos Dafonte

Universidade da Coruña

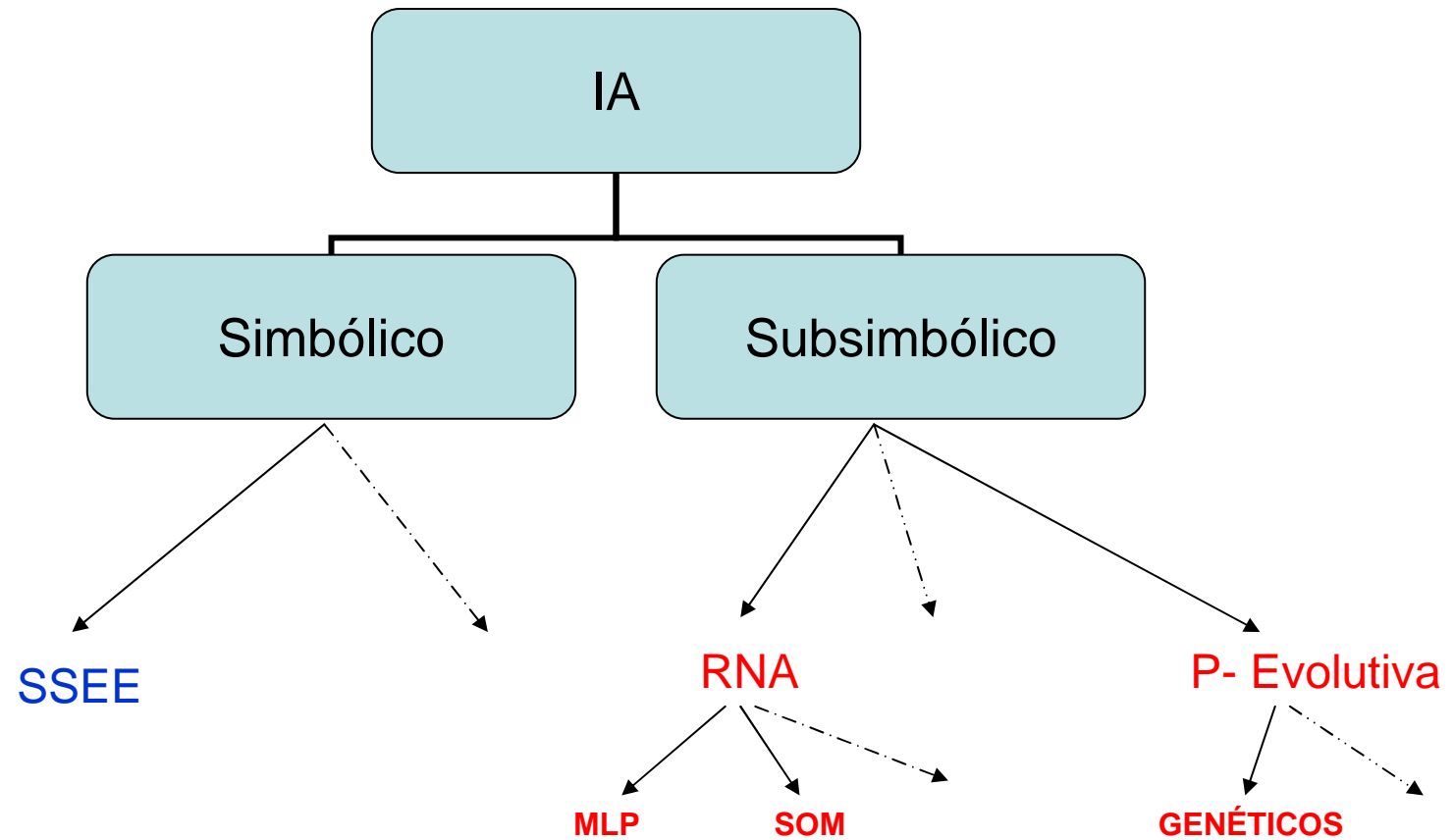


Introducción

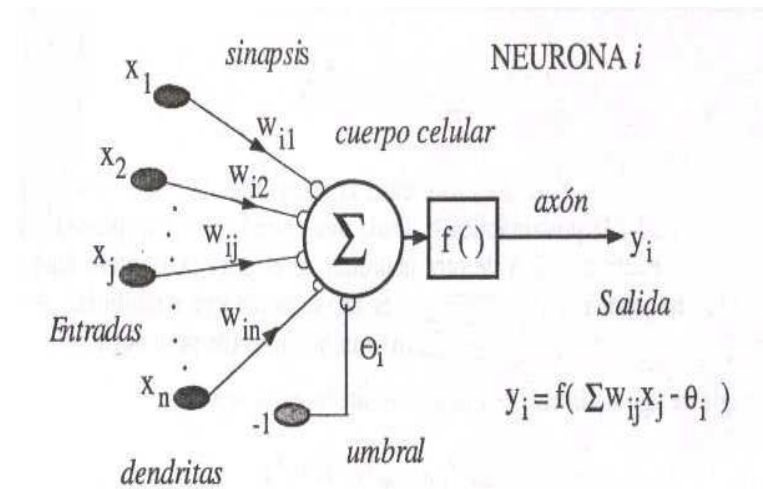
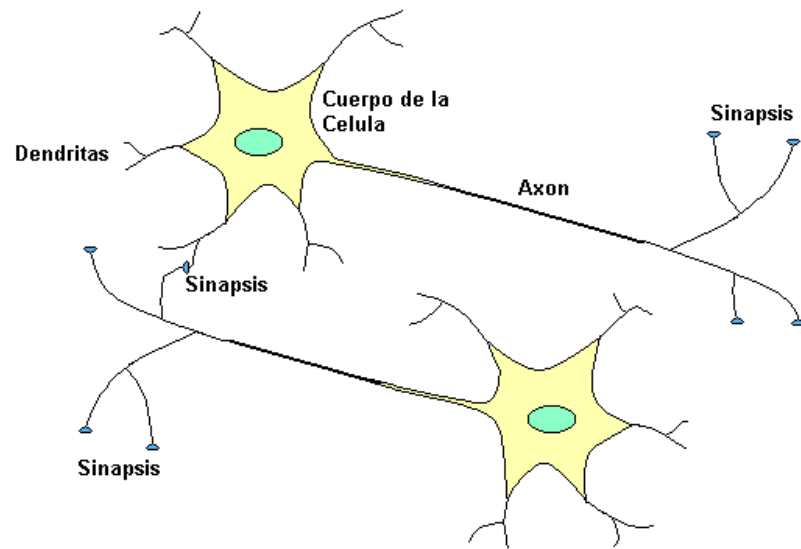
Dos tareas dentro de CU8

- BP/RP → "agrupamiento de Outliers" → No Supervisado
- RVS → obtención parámetros físicos → Supervisado

Inteligencia Artificial



Neurona biológica / Neurona Artificial



Patrón de conectividad o arquitectura

- **Manera de conectarse las neuronas**

$\mathbf{X}=(x_1,x_2,\dots,x_n)$ (vector de entradas)

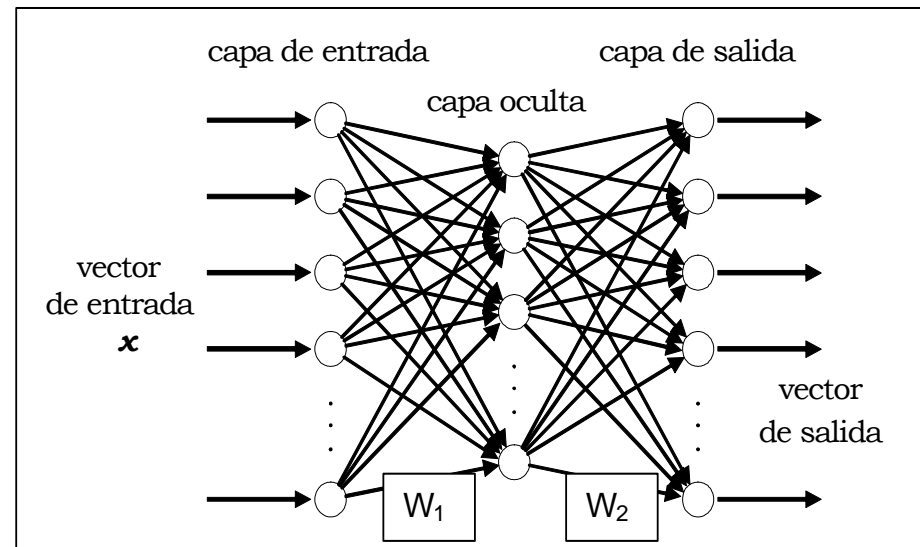
$\mathbf{w}_1, \mathbf{w}_2$ (vectores de pesos)

F (función de activación)

$\mathbf{S}=(y_1,y_2,\dots,y_n)$ (vector de salidas)

$$\mathbf{S}=F(F(\mathbf{X} \cdot \mathbf{W}_1) \cdot \mathbf{W}_2)$$

F es una función NO lineal



Dos tareas dentro de CU8

- BP/RP → "agrupamiento de Outliers" → No Supervisado → SOM
(Genéticos)
- RVS → obtención parámetros físicos → Supervisado → MLP
(Genéticos)

Dominios Transformados

Objetivos

- Disminuir la dimensionalidad de los datos de entrada
- Búsqueda de puntos / regiones de interés en los espectros. Posibilidad de fusión de regiones
- Combinación de técnicas que mejoren resultados de las redes.



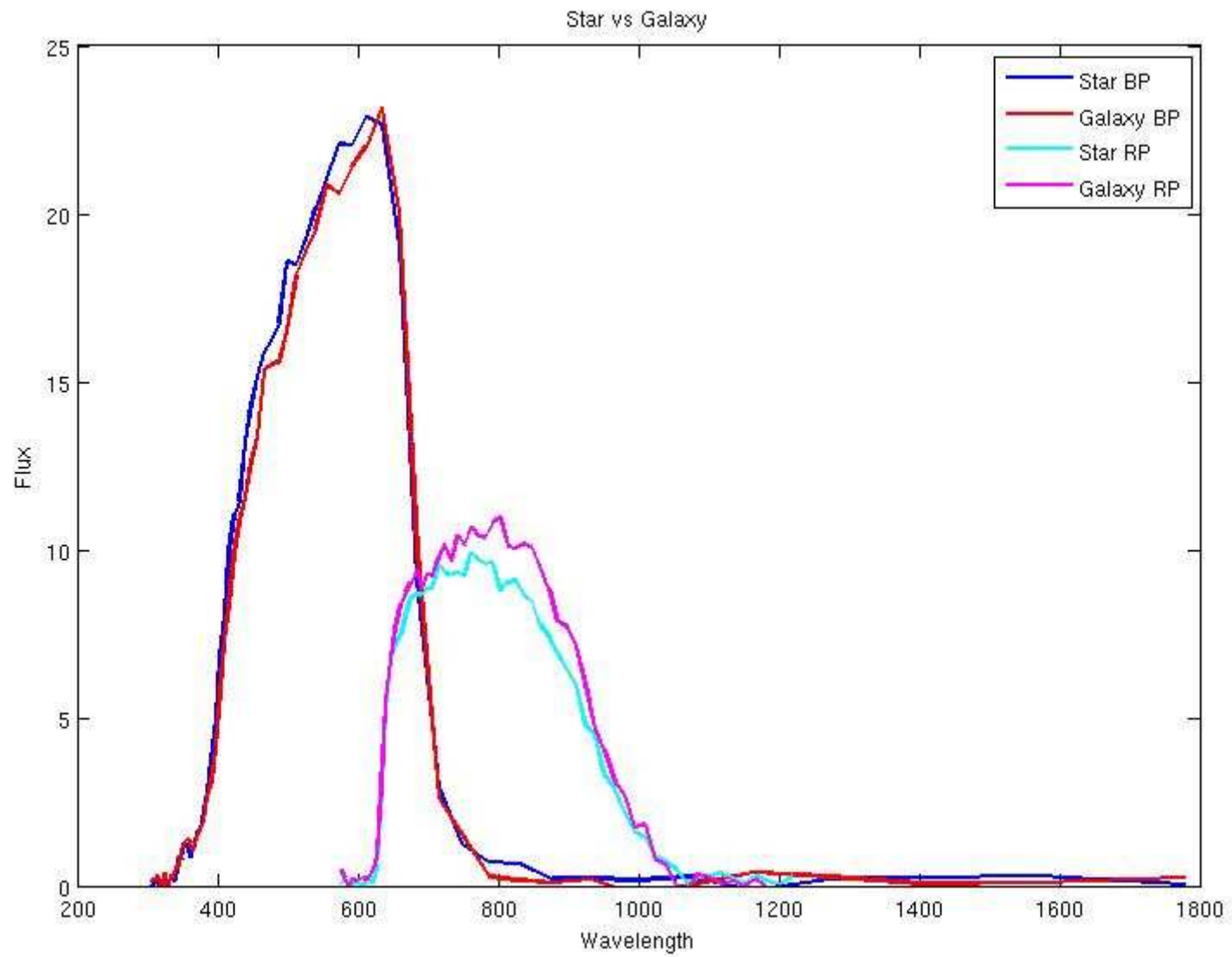
PCA

Fourier

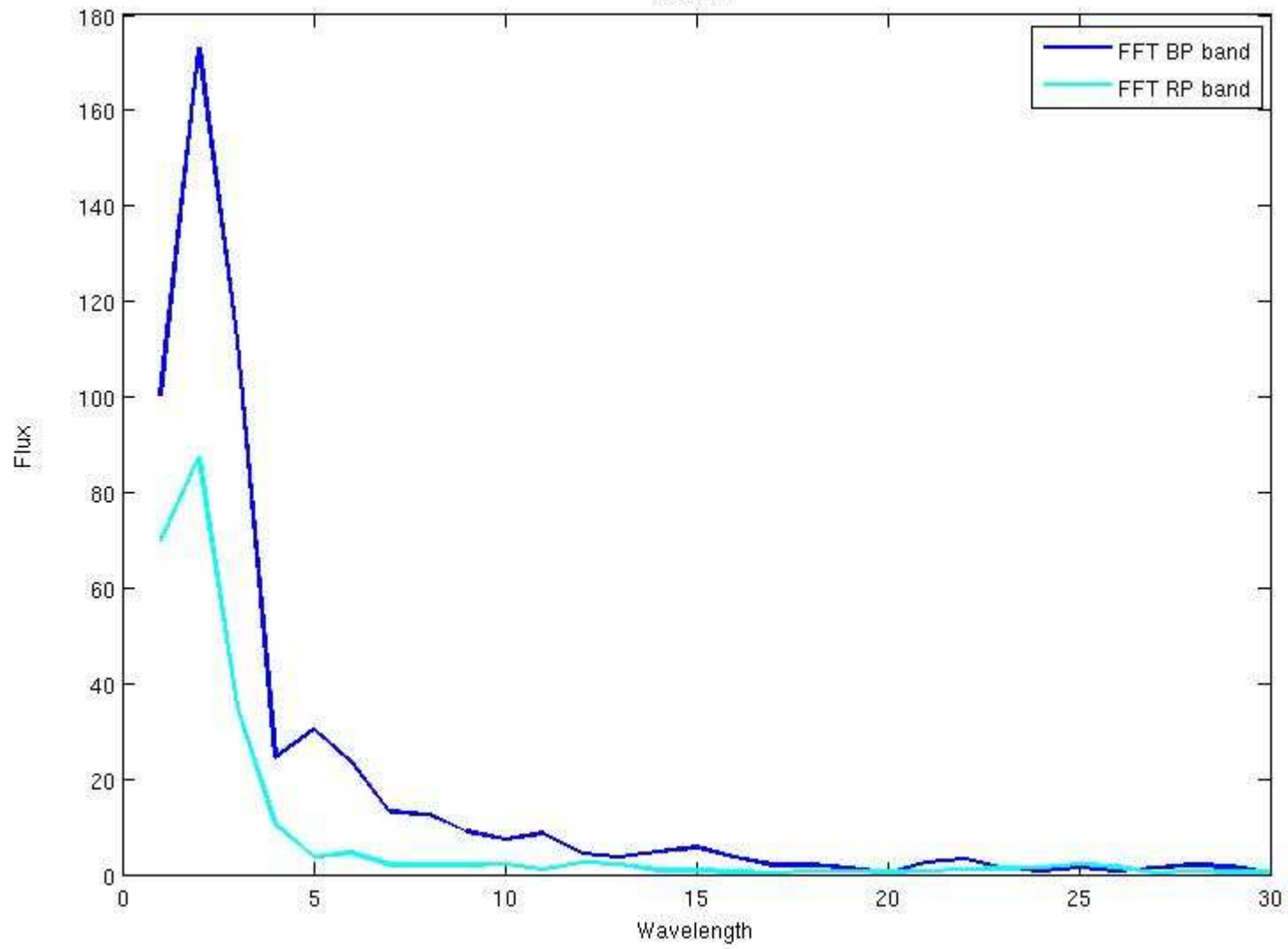
Waveletets

Genéticos

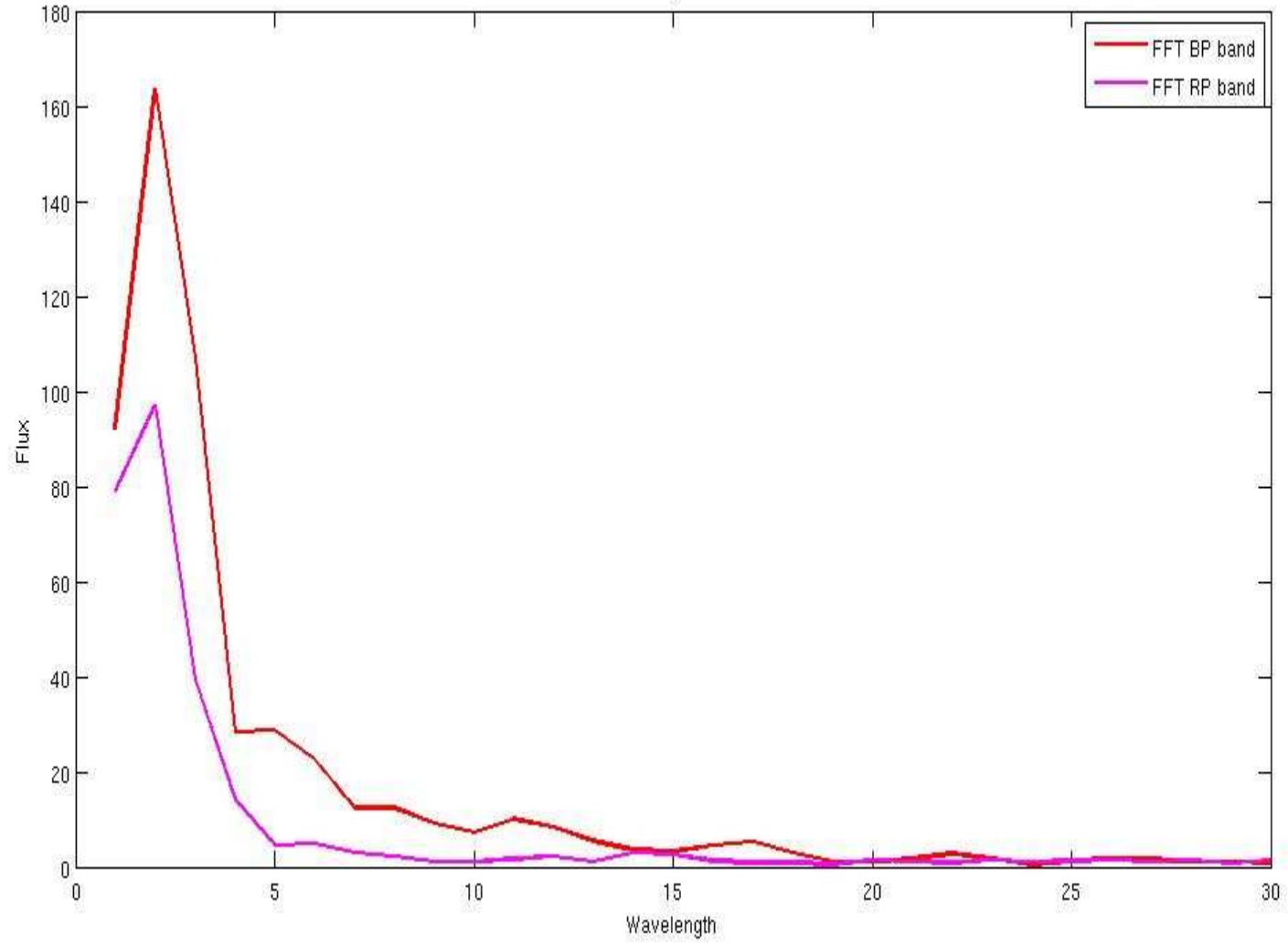
Transformada de Fourier



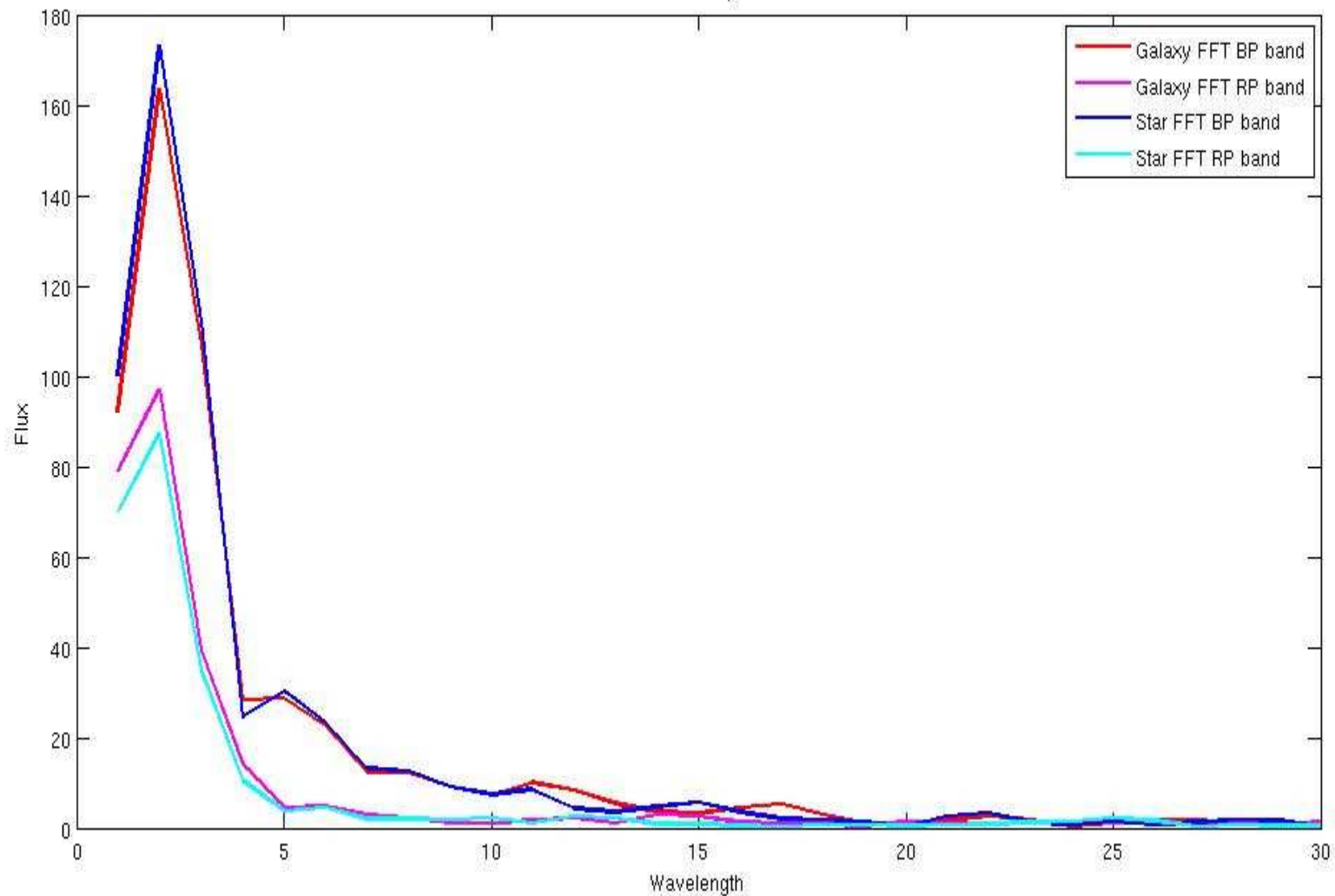
Star FFT



Galaxy FFT



Star vs Galaxy FFT

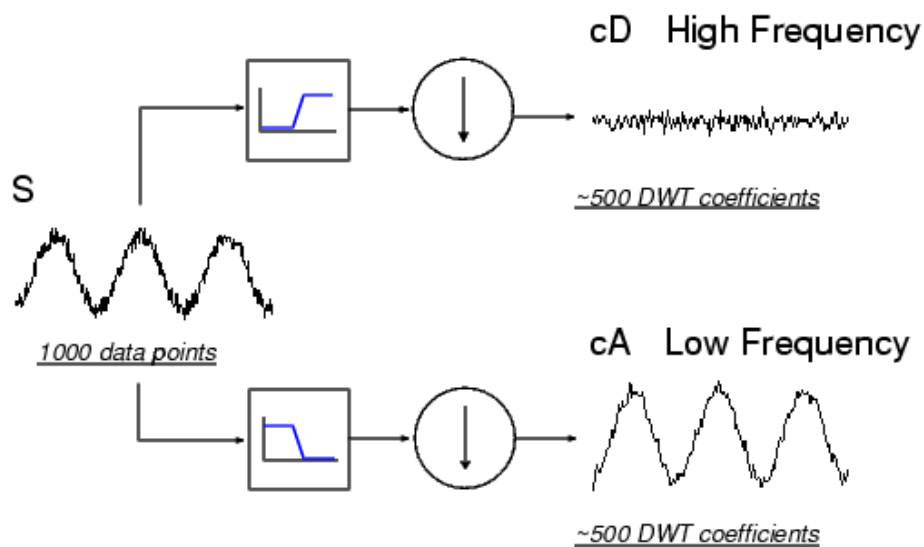


Wavelets

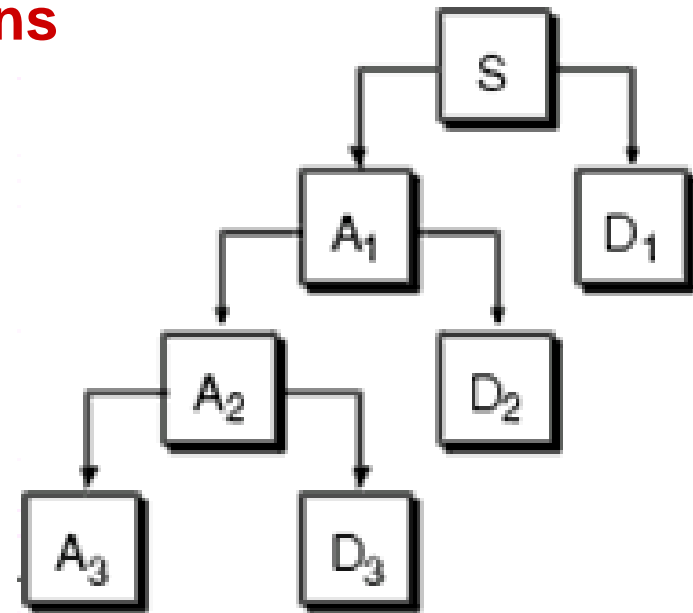
Wavelet discreta

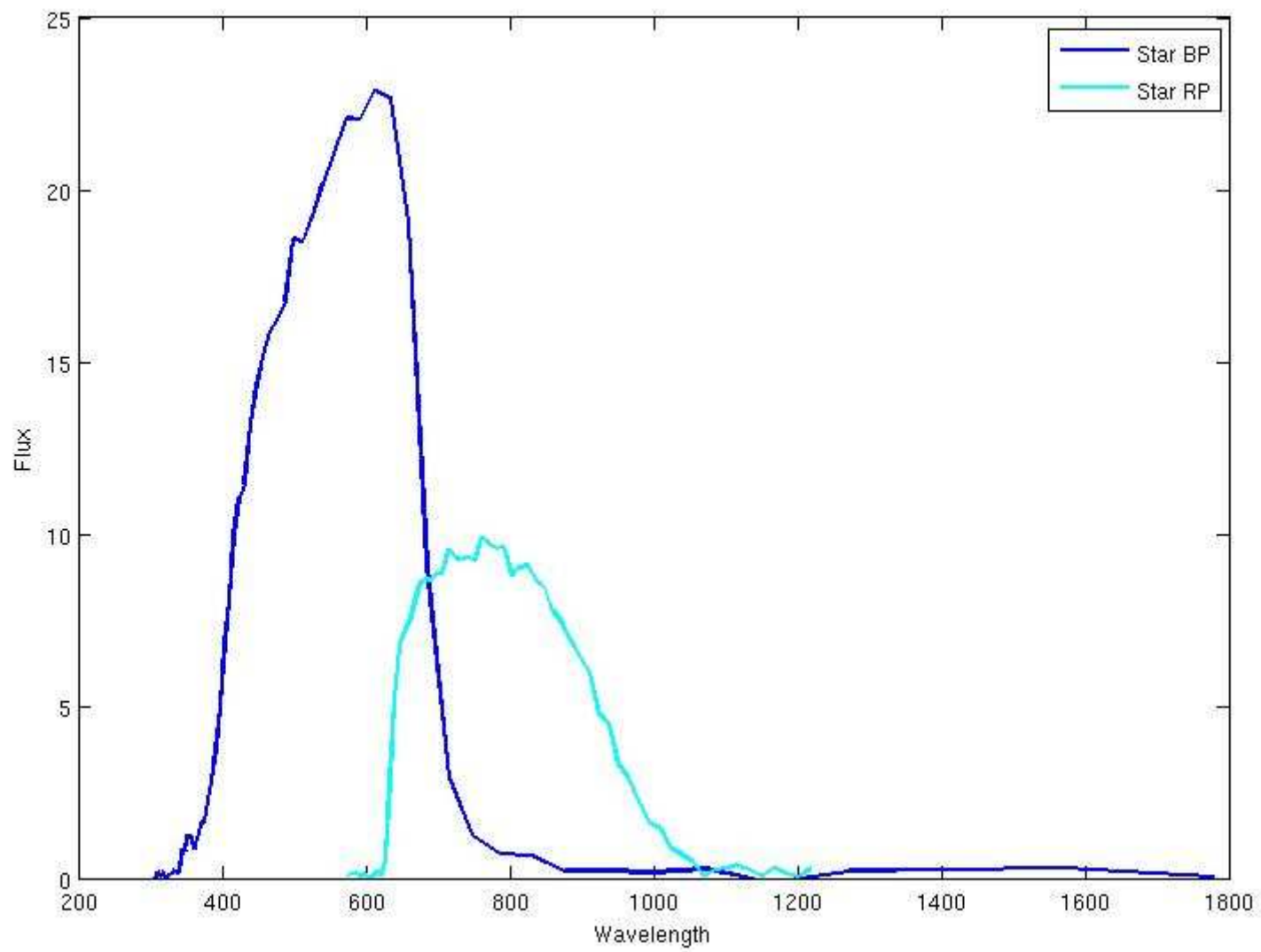
- Analisis en el “dominio” Wavelet .
- DWT . Analisis Multinivel .

Low-pass filter - Approximations



High-pass filters – Details

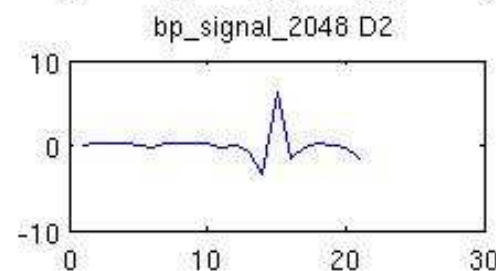
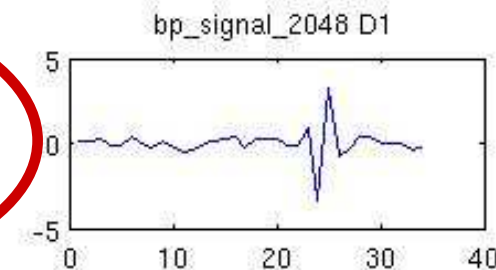
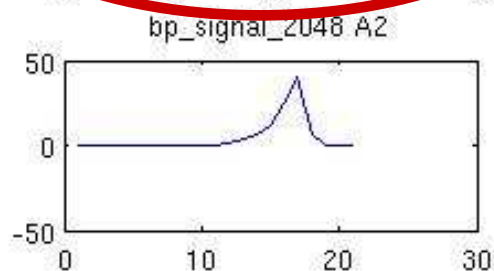
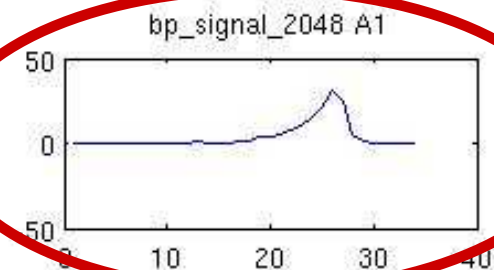
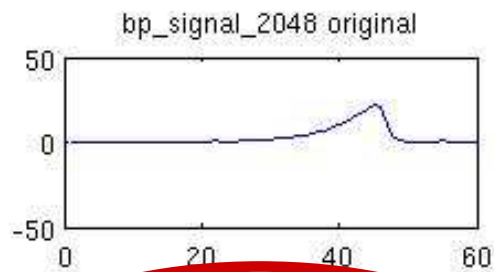




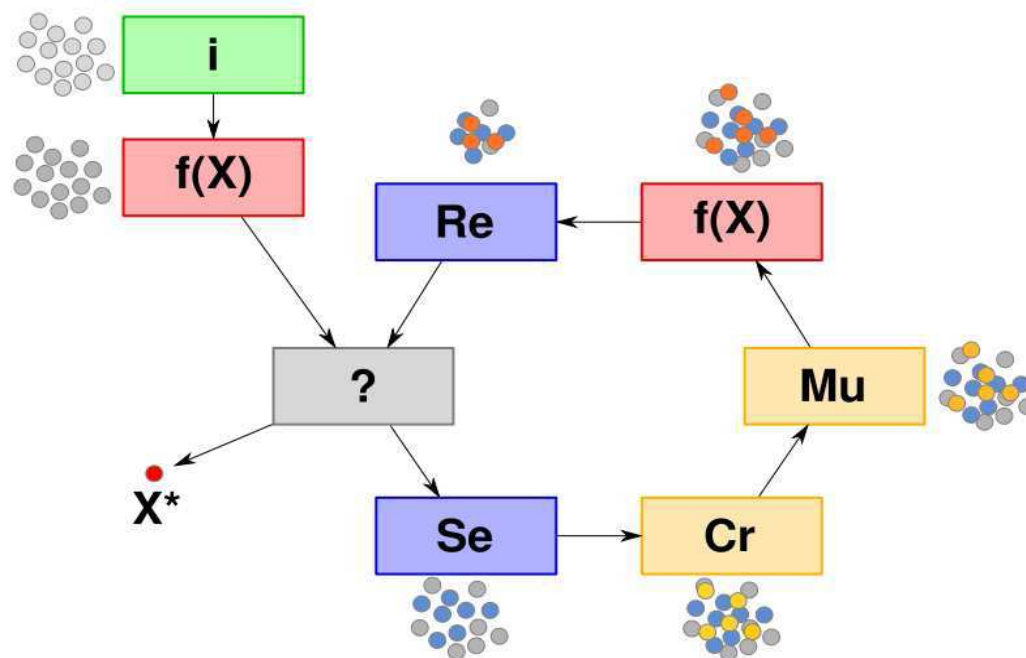
Ejemplo de Wavelets de un espectro

- Wavelet del espectro BP :

Mejor resultado

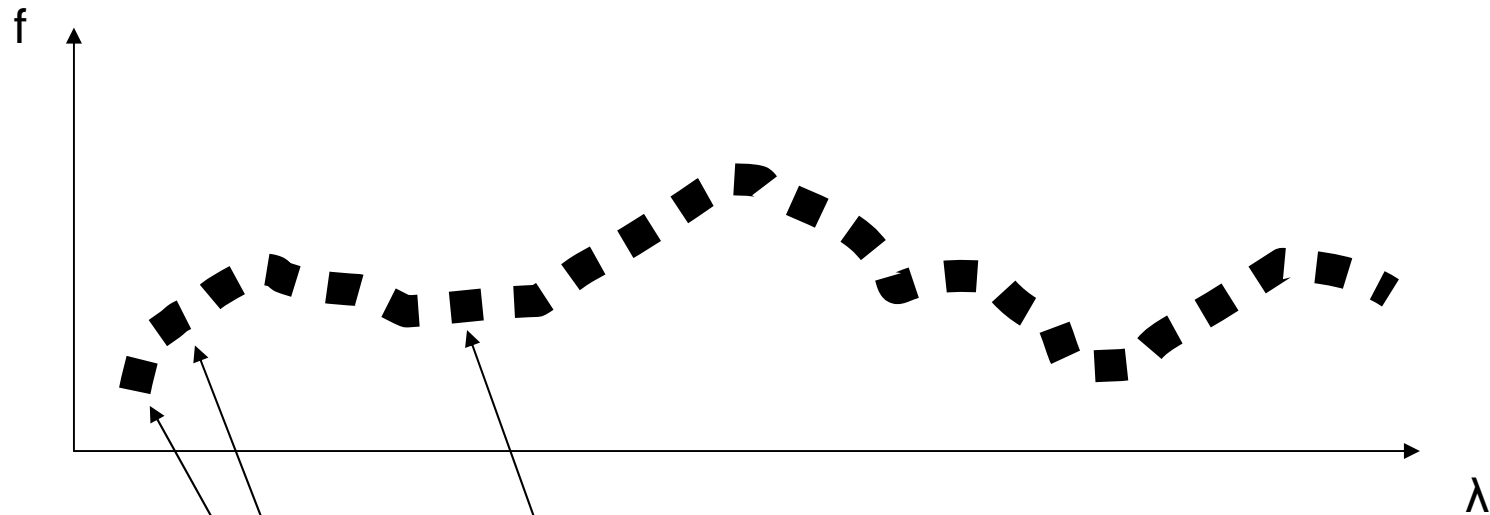


Algoritmos Genéticos



Algoritmo genético. i: inicialización; f(x): evaluación; ?: condición de término; Se: selección; Cr: cruzamiento; Mu: mutación; Re: reemplazo; X*: mejor solución.

Genéticos



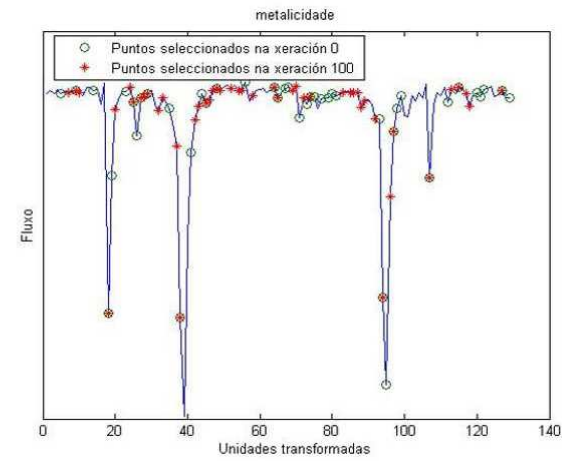
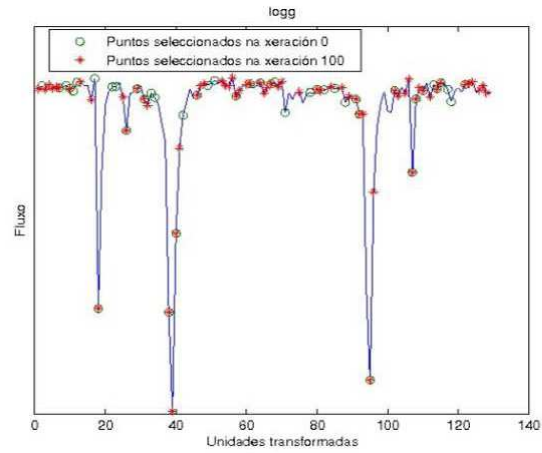
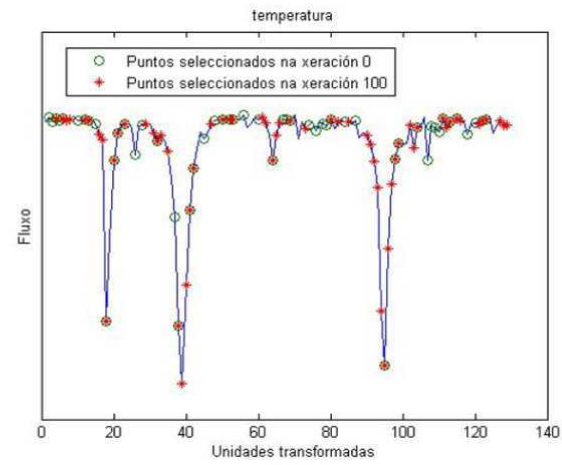
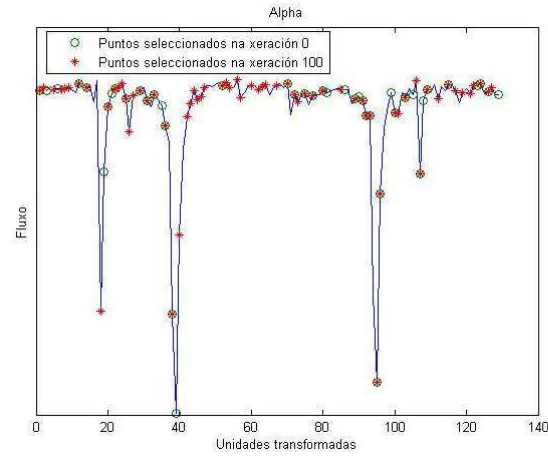
S_i (0 1 0 0 1 1 1 1 0 0 0 1 1 1 0 0 0 1 0 1 0 0 0)

S_k (1 1 0 0 0 0 0 1 0 0 0 1 1 1 0 0 0 1 1 1 0 1 0)

CRUCE + MUTACIÓN

S_{ik} (0 1 0 0 1 1 1 1 0 0 1 0 1 1 0 0 0 1 1 1 0 1 0)

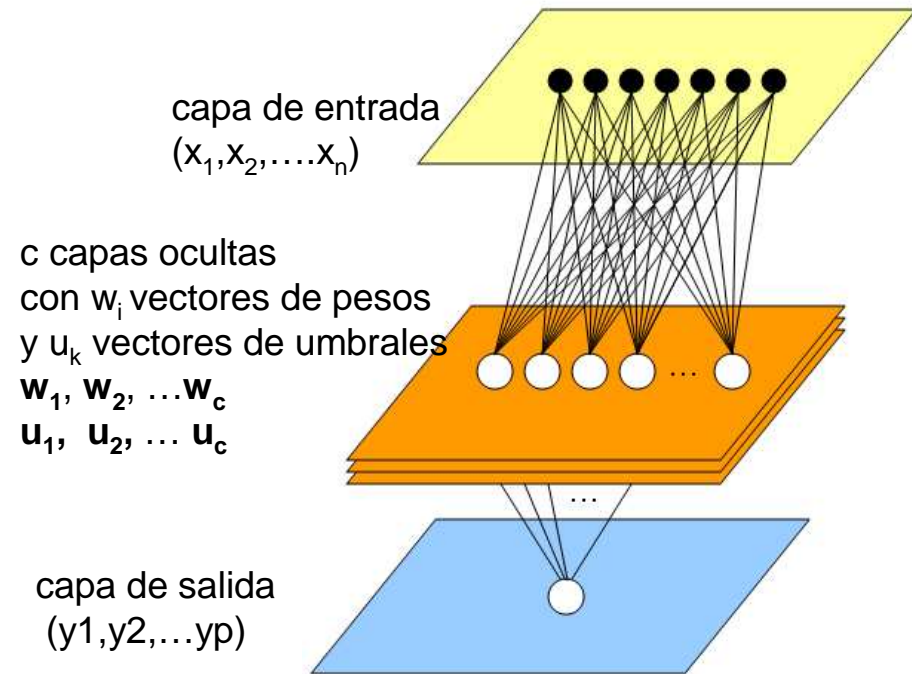
Genéticos



Obtención parámetros físicos en RVS

Perceptron multicapa (MLP)

- Rumelhart, Himton y Willians (1986)
- Conexiones dirigidas hacia delante (redes feedforward)
- Las conexiones entre neuronas tienen asociado un peso (número real)
- Cada neurona tiene un umbral.



Diseño de la arquitectura de red

- Definir la función de activación

Por el recorrido deseado (no influye en resultado)

- Número de neuronas

En las capas de entrada y salida lo determina el problema

*Importante **reducir dimensionalidad** (PCA, correlación, sensibilidad de parámetros, ...)*

- Número de capas

*No existe método, **prueba y error** (búsqueda en el espacio de arquitecturas, línea de investigación actual)*

Algoritmo de retropropagación

- Estrictamente se busca el mínimo de la función de error total (E), en la práctica se minimiza $e(n)$ para todo n , w_{ij}^c

$$w(n) = w(n-1) - \alpha \frac{\partial e(n)}{\partial w(n)}$$

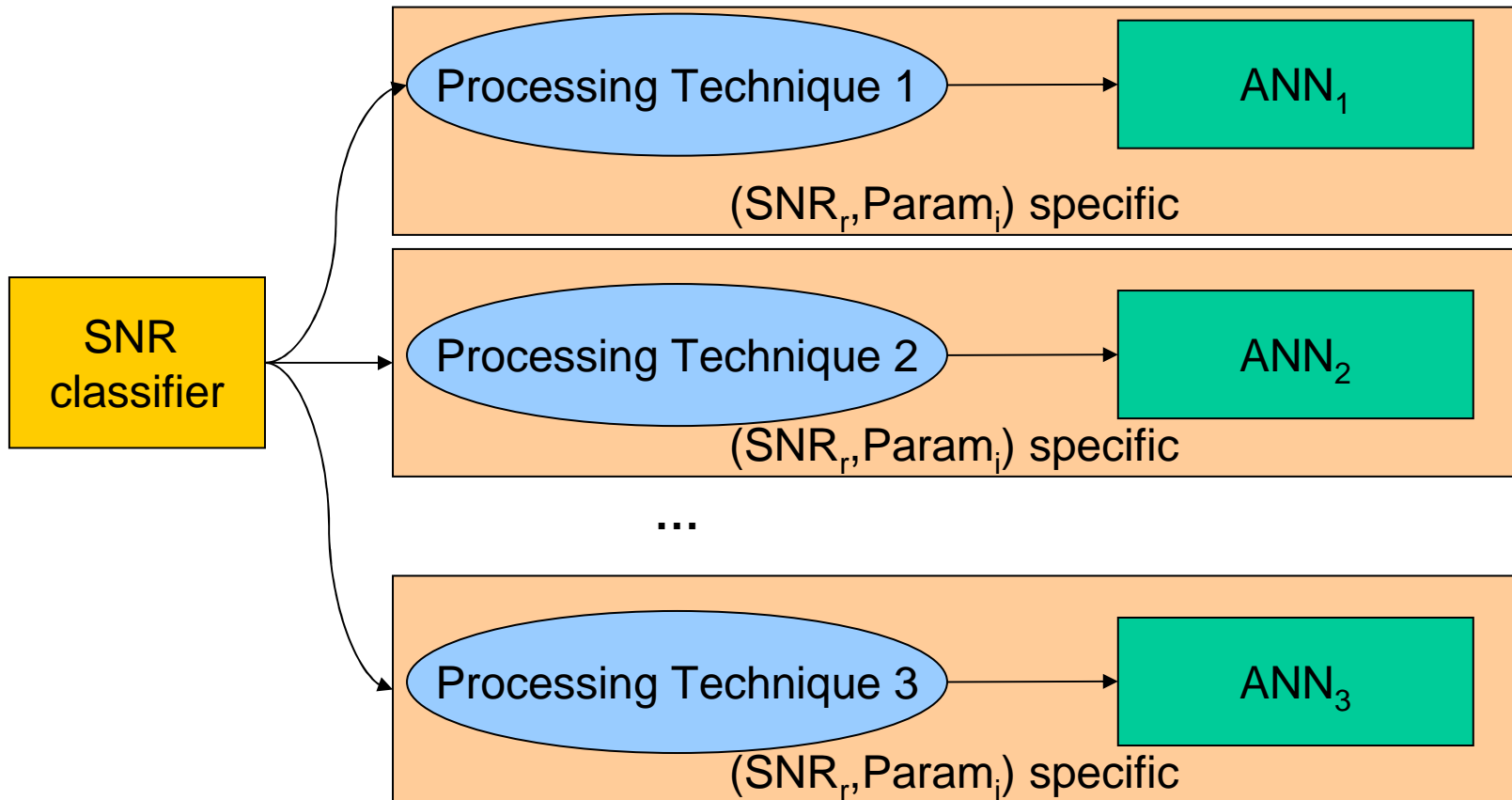
α , tasa de aprendizaje

Los umbrales u_i^k , se pueden ver como entradas de valor -1 y peso u_i^k

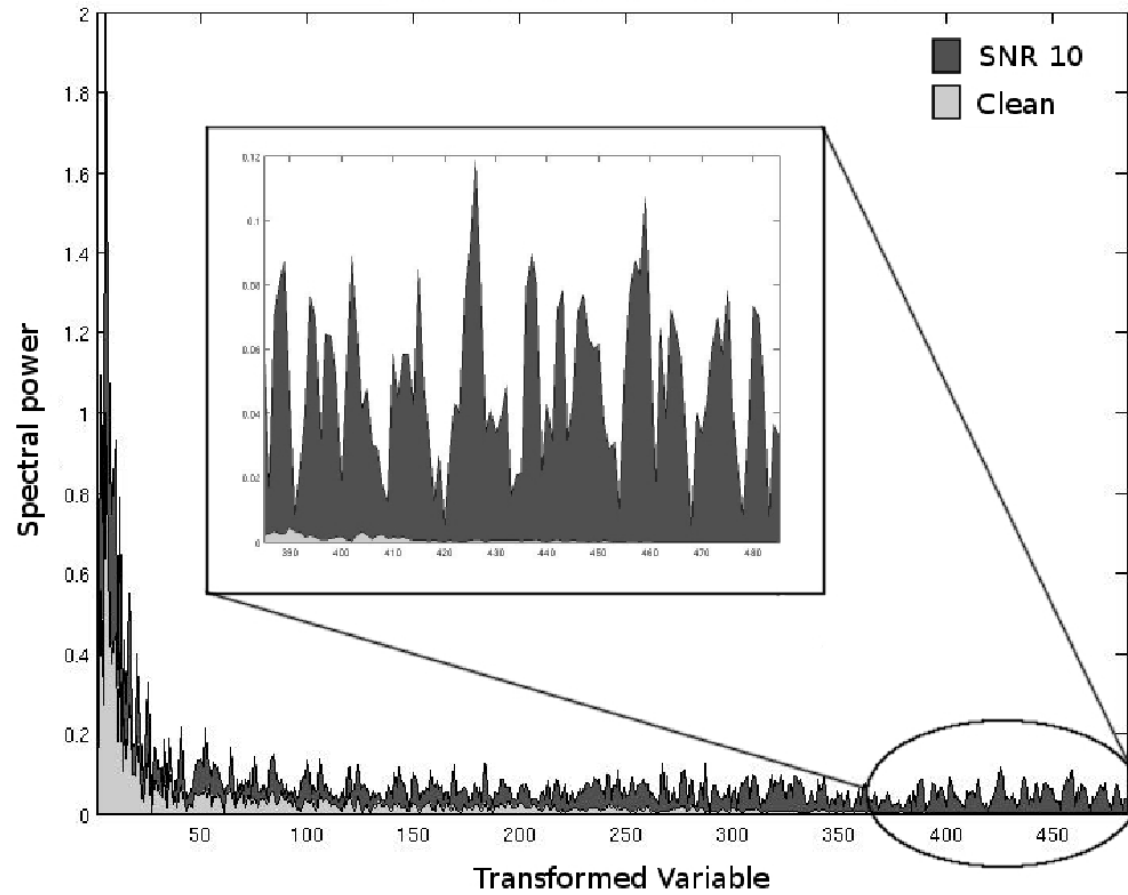
Algunos problemas posible causas

- Sobrentrenamiento
 - Demasiados ciclos de entrenamiento => **repetir entrenamiento**
 - Demasiadas neuronas en capas ocultas => **reducir dimensionalidad** (prueba-error)
- Mínimos locales
 - Necesidad de más grados de libertad => **más neuronas en capa oculta**
- Saturación (parálisis)
 - se satura la salida de las neuronas => no modifica los pesos => => “parece que ha terminado” => **valores iniciales más pequeños**

Method

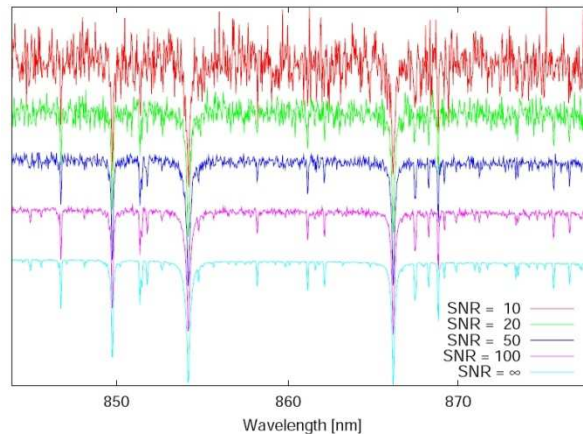


Clean (ligh gray) Vs. SNR 10 (dark gray)



Resultados (I)

SNR	Clean				SNR200			
	Teff	Logg	[Fe/H]	[α /Fe]	Teff	Logg	[Fe/H]	[α /Fe]
FFT	35,833	0,0708	0,059	0,0347	175,38	0,28	0,19	0,09
W	72,2453	0,1475	0,1175	0,0684	80,16	0,16	0,12	0,07
A1	81,1248	0,1605	0,1283	0,0749	106,49	0,17	0,16	0,08
A2	85,3519	0,1849	0,1464	0,0811	90,92	0,19	0,15	0,08
A3	92,5648	0,206	0,17	0,0834	99,35	0,24	0,17	0,09
A4	108,322	0,2191	0,2083	0,0932	111,11	0,22	0,21	0,09
A5	115,155	0,2576	0,2389	0,1092	130,43	0,27	0,25	0,11
D1	166,298	0,2819	0,1868	0,0752	259,67	0,56	0,27	0,12
D2	159,379	0,338	0,1859	0,0732	207,69	0,4	0,2	0,09
D3	138,612	0,2708	0,1716	0,0669	188,39	0,34	0,19	0,09
D4	105,339	0,2278	0,1579	0,0727	147,51	0,29	0,18	0,09
D5	118,439	0,2438	0,2286	0,0857	145,45	0,29	0,24	0,1



	Min	Max
Effective Temper	4500	7750
Log G	-0,5	5
[Fe/H]	-5	1
[α /H]	-0,2	0,4

Results (II)

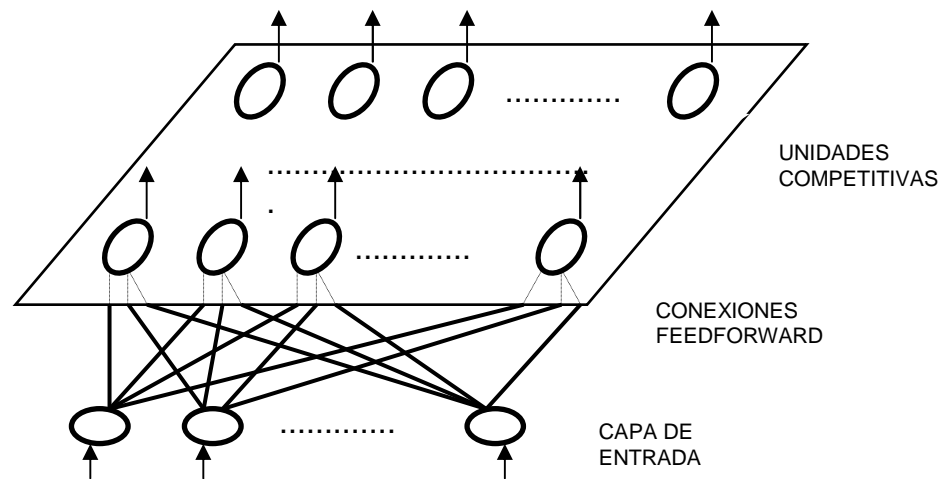
SNR	Teff	Logg	[Fe/H]	[α /Fe]
∞	35,833	0,0708	0,059	0,0347
200	80,16	0,16	0,12	0,07
150	81,39	0,16	0,12	0,08
100	90,06	0,18	0,14	0,09
75	97,42	0,19	0,15	0,09
50	118,50	0,23	0,17	0,10
25	188,53	0,39	0,26	0,14
10	267,77	0,55	0,35	0,16
5	438,17	0,80	0,52	0,18

	Teff	Logg	[Fe/H]	[α /Fe]
[5, 10, 25]	W	A2	A2/A3	A1
[50, 75, 100]	W	W	W	W
[150, 200, 1E3]	W	W	W	W
∞ (Clean)	FFT	FFT	FFT	FFT

Agrupamiento de Outliers

Modelo de red

- Dos capas: entrada y capa competitiva



$\varepsilon = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n)$ vector de entradas

μ_{ij} matriz de pesos sinápticos

$\mu_j = (\mu_{1j}, \mu_{2j}, \dots, \mu_{nj})$ vector de pesos asociados a la célula j de la capa competitiva

Funcionamiento

- En el modelo de Kohonen se utiliza una función distancia para comparar los vectores de entrada $\boldsymbol{\varepsilon}$ y peso sináptico $\boldsymbol{\mu}_j$ de la neurona j (de la capa competitiva) y a partir de ella su salida, si suponemos distancia euclídea

$$\tau_j = \sqrt{\sum_{i=1}^N (\varepsilon_i - \mu_{ij})^2}$$

Se calculan todos los $\tau_j, \forall j$

y se elige aquél que es mínimo, etiquetando esa célula como **neurona ganadora**

Funcionamiento

- Si el método de aprendizaje sigue el modelo *“el que gana se lo lleva todo”* →
→ salida de la neurona ganadora vale 1 y el resto cero
- Si el método de aprendizaje permite definir **regiones** de neuronas que representan **patrones próximos** → **definir vecindarios** → el aprendizaje se produce también en neuronas vecinas (distintas arquitecturas)

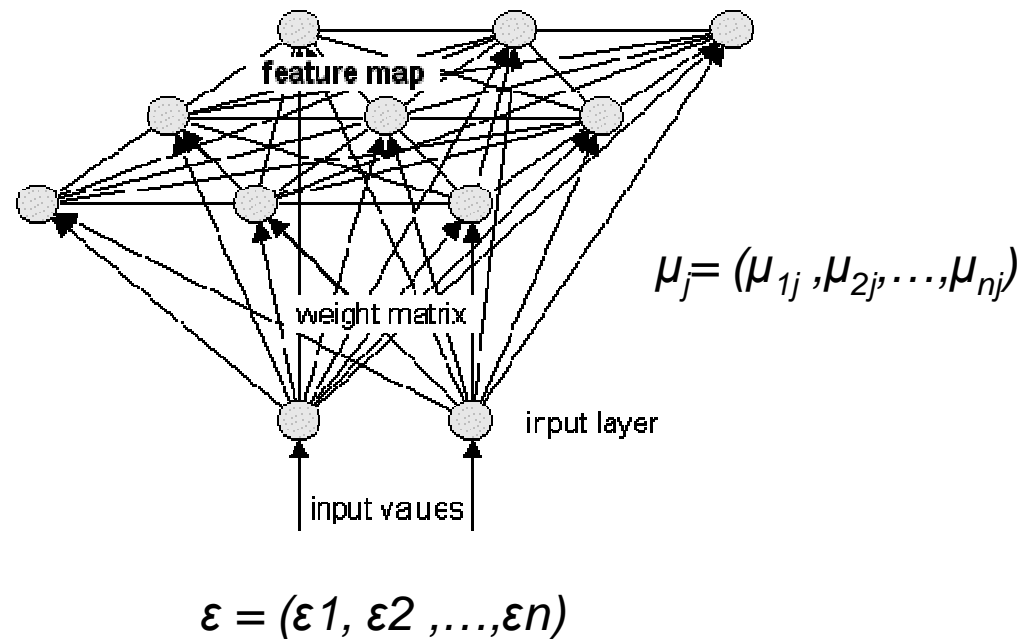
$$\frac{d\mu_{ij}}{dt} = \left\{ \begin{array}{ll} \frac{\alpha(t)}{d(c_i, c_j)} (\varepsilon_i(t) - \mu_{ij}(t)) & \text{si } c_i \text{ ganadora y } d(c_i, c_j) < \theta \\ 0 & \text{en caso contrario} \end{array} \right\}$$

medida de distancia

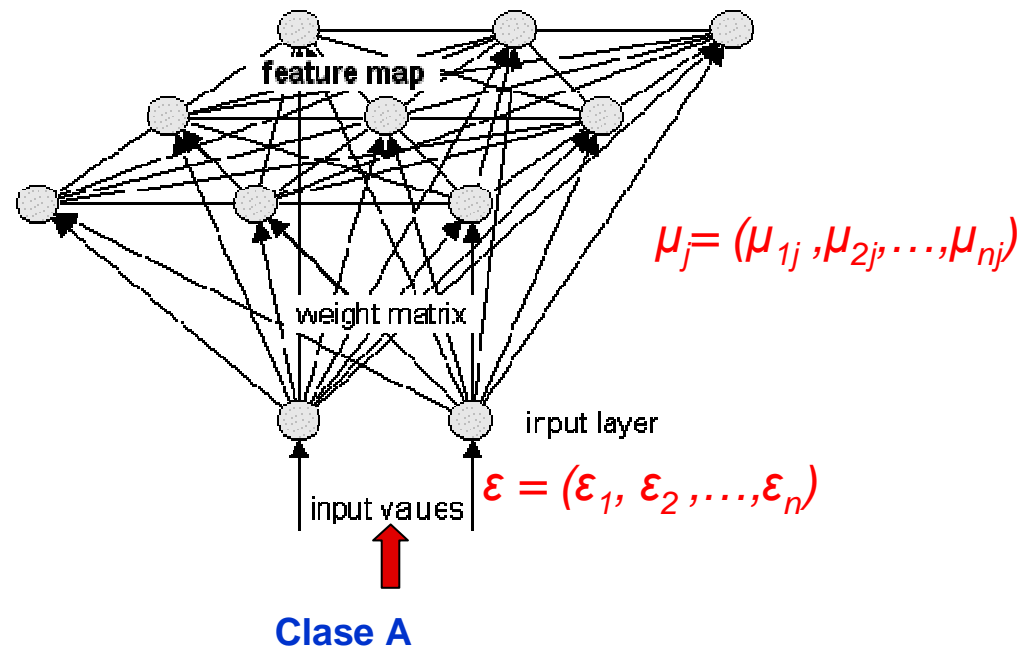
entero que marca los límites del vecindario

Algoritmo de agrupamiento

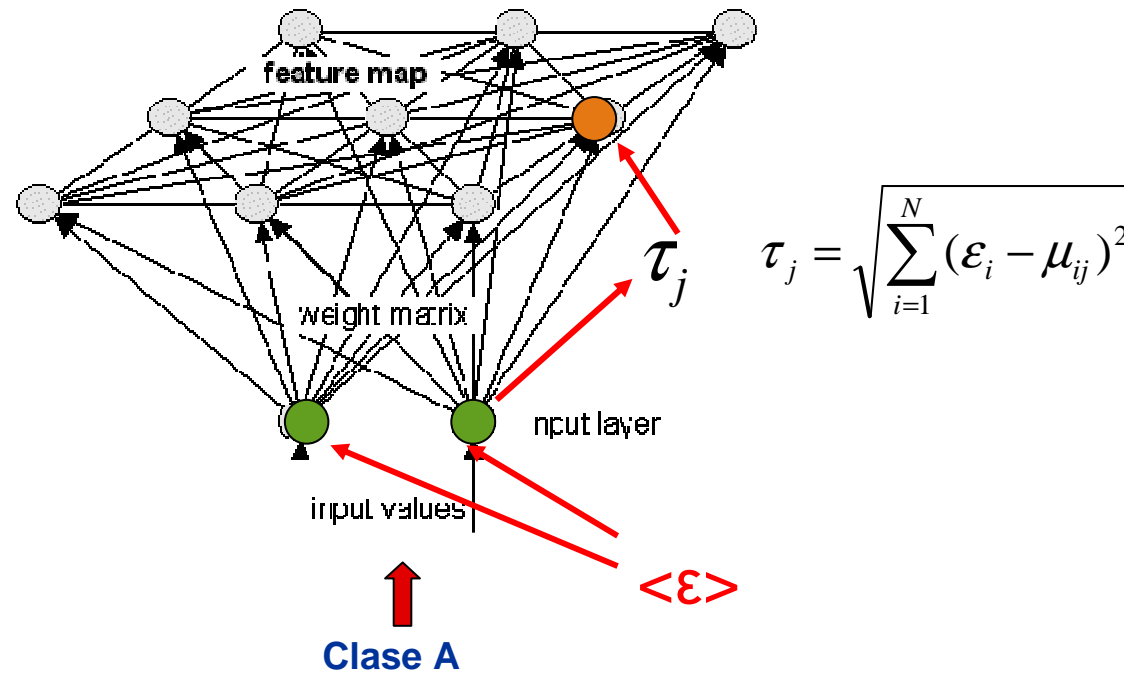
- Self Organization Map + Kohonen algorithm



Algoritmo de agrupamiento

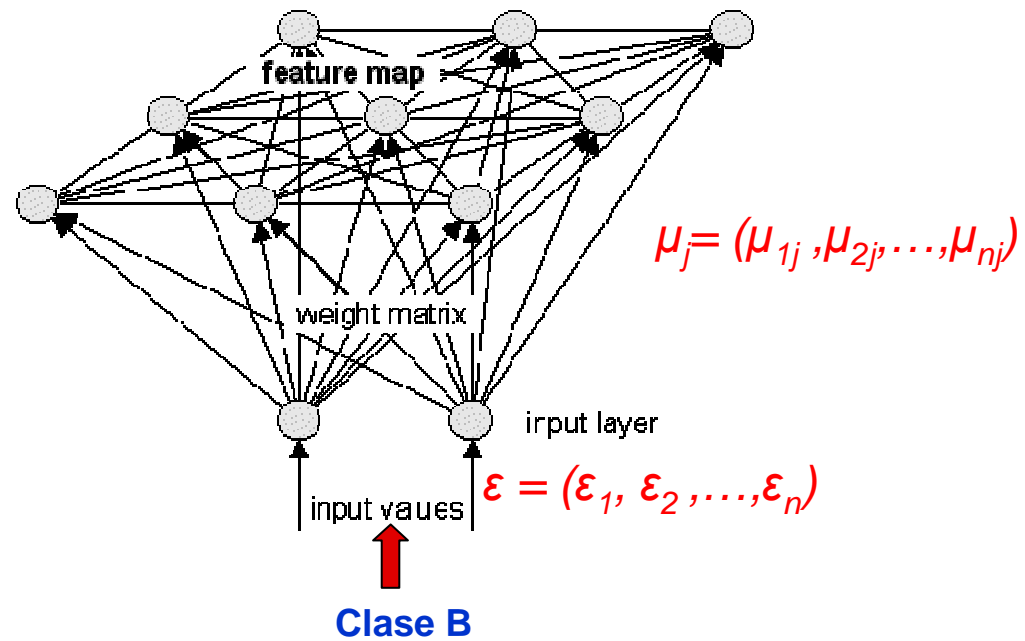


Algoritmo de agrupamiento



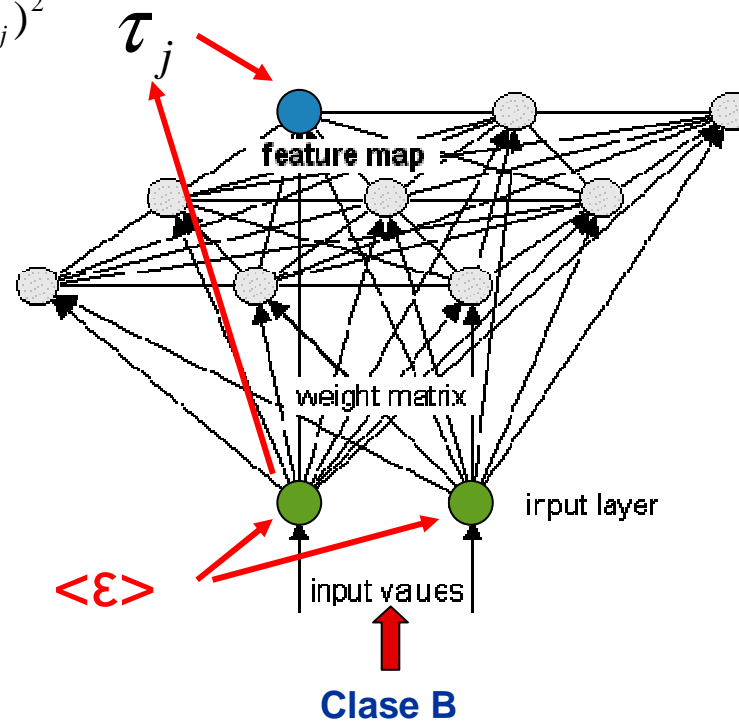
$$\frac{d\mu_{ij}}{dt} = \begin{cases} \frac{\alpha(t)}{d(c_i, c_j)} (\epsilon_i(t) - \mu_{ij}(t)) & \text{si } c_i \text{ ganadora y } d(c_i, c_j) < \theta \\ 0 & \text{en caso contrario} \end{cases}$$

Algoritmo de agrupamiento



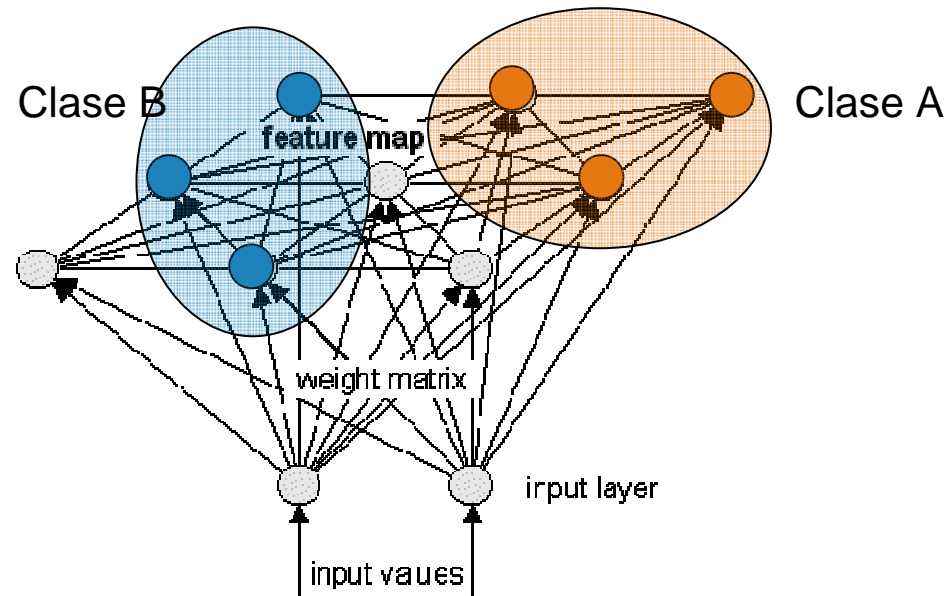
Algoritmo de agrupamiento

$$\tau_j = \sqrt{\sum_{i=1}^N (\varepsilon_i - \mu_{ij})^2}$$

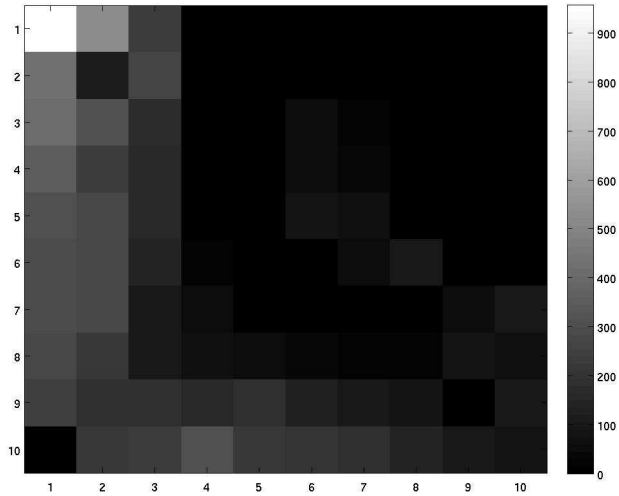


$$\frac{d\mu_{ij}}{dt} = \left\{ \begin{array}{ll} \frac{\alpha(t)}{d(c_i, c_j)} (\varepsilon_i(t) - \mu_{ij}(t)) & \text{si } c_i \text{ ganadora y } d(c_i, c_j) < \theta \\ 0 & \text{en caso contrario} \end{array} \right\}$$

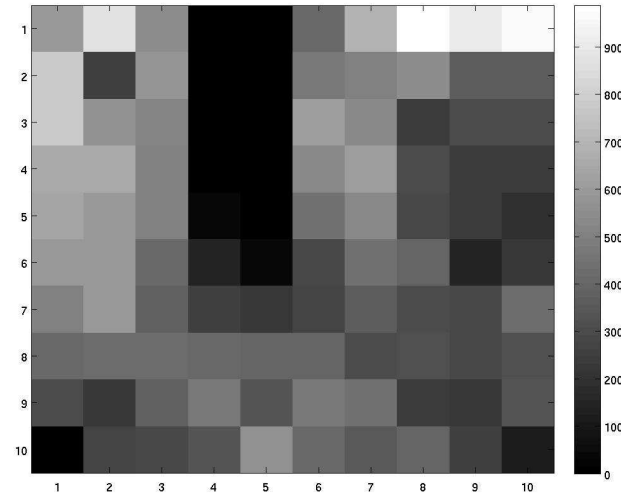
Algoritmo de agrupamiento



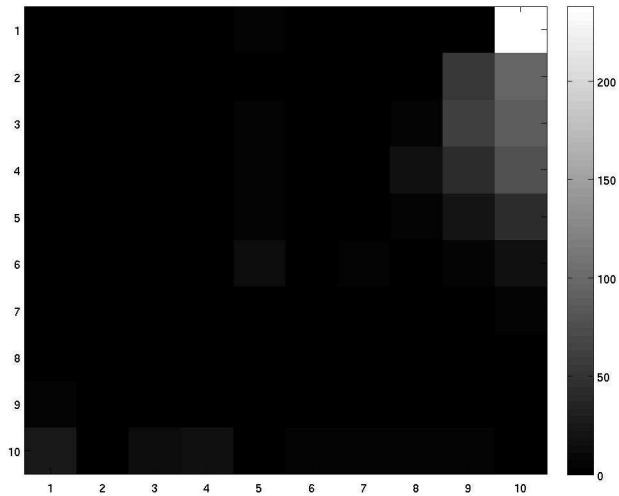
Run_1. A1/BP 10x10



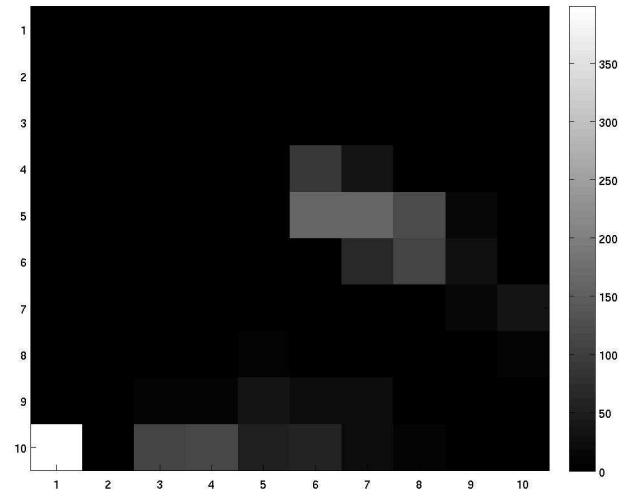
Physbinary



Star

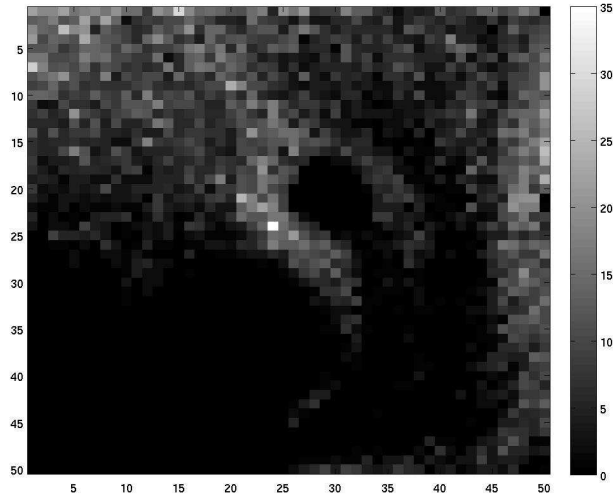


Quasar

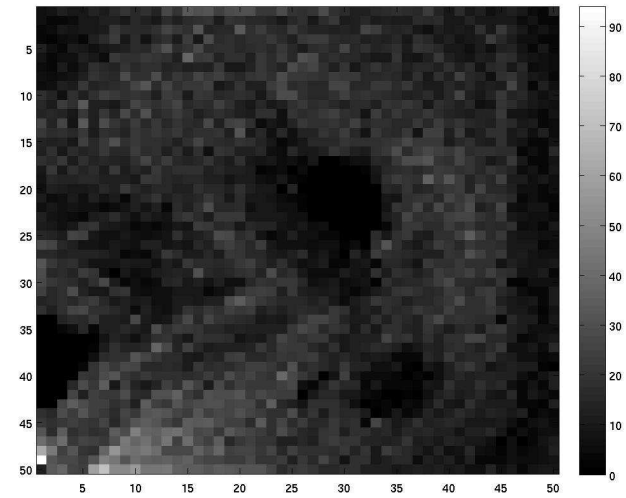


Galaxy

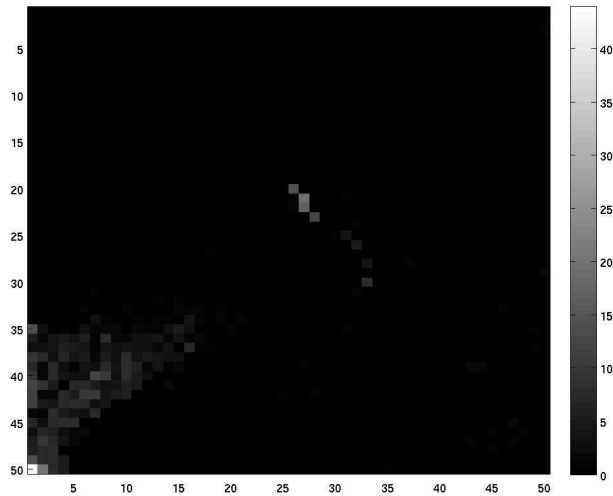
Run_1. A1/BP 50x50



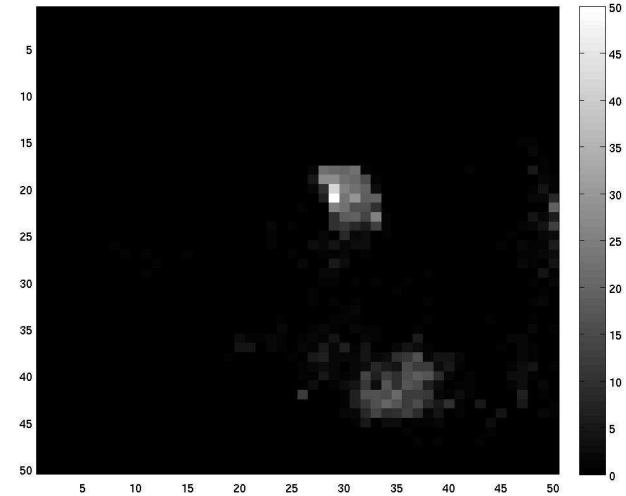
Physbinary



Star

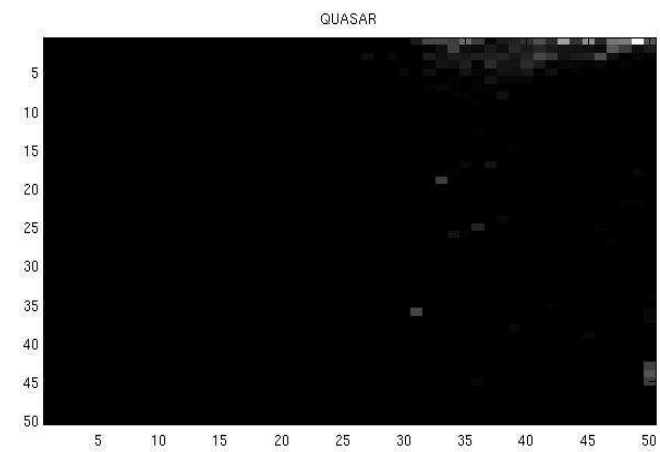
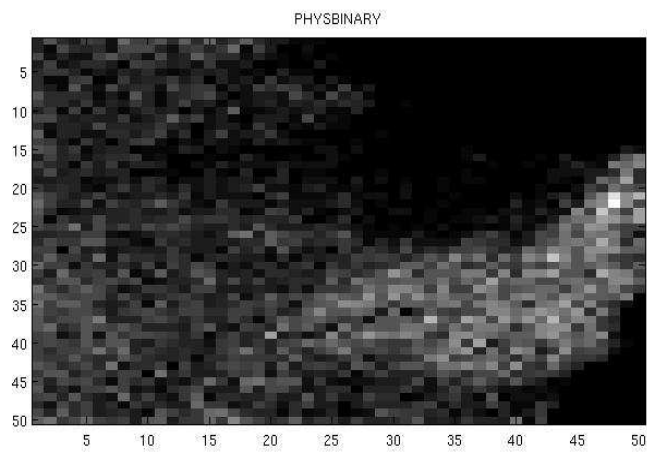
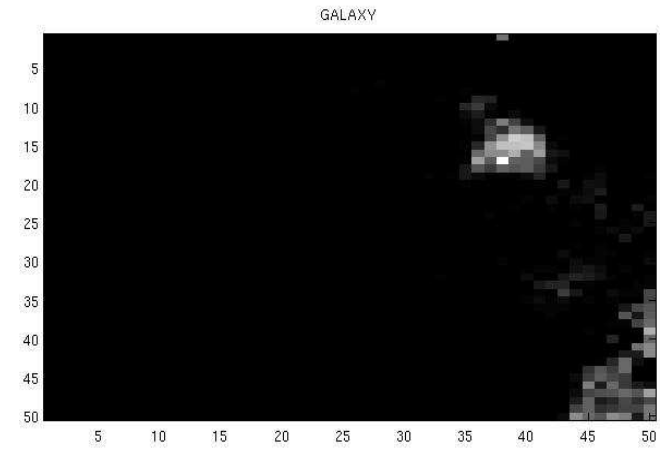
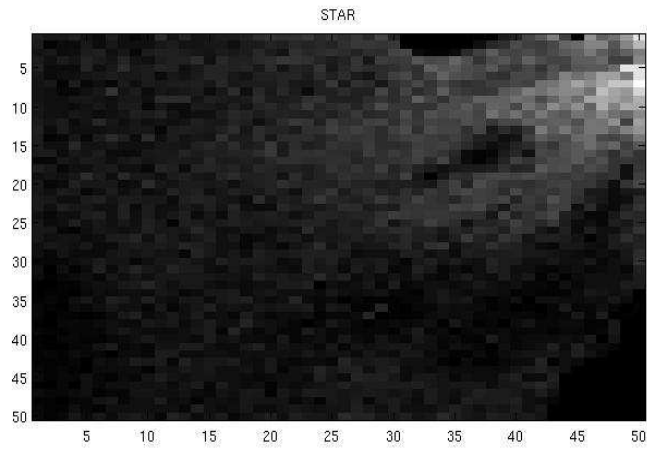


Quasar



Galaxy

RUN2- 50x50



Example of output map and Confusion matrix

10x10
output
map

confusion
matrix %

957	526	548	007	000	410	688	988	907	238
432	256	256	008	000	467	506	554	053	093
418	300	515	002	000	616	539	237	062	087
351	236	494	011	003	090	610	295	042	076
306	282	506	038	005	160	156	123	019	041
287	280	414	147	040	285	435	108	151	231
291	271	376	259	220	276	356	305	285	424
283	219	424	403	387	400	298	309	291	320
241	189	181	475	330	470	440	242	222	328
399	218	226	304	052	058	024	397	255	116

%	PHYSBINARY	STAR	QUASAR	GALAXY	Total
PHYSBINARY	64	28	0	8	100 %
STAR	31	52	8	9	100 %
QUASAR	4	13	78	5	100 %
GALAXY	13	20	1	66	100 %

Gracias