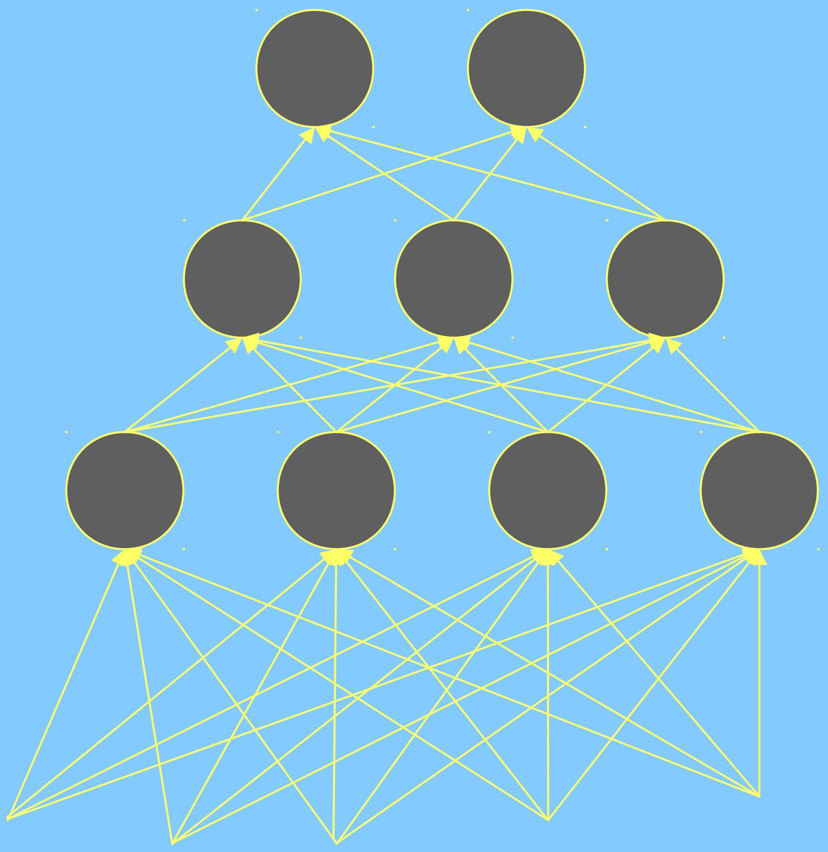
Estrategia general

Entradas, datos representadas como conjunto de valores numéricos. Pasarán a ciertos procesadores

Resultados, salidas también se corresponderán a un conjunto de valores numéricos: activación de determinados procesadores

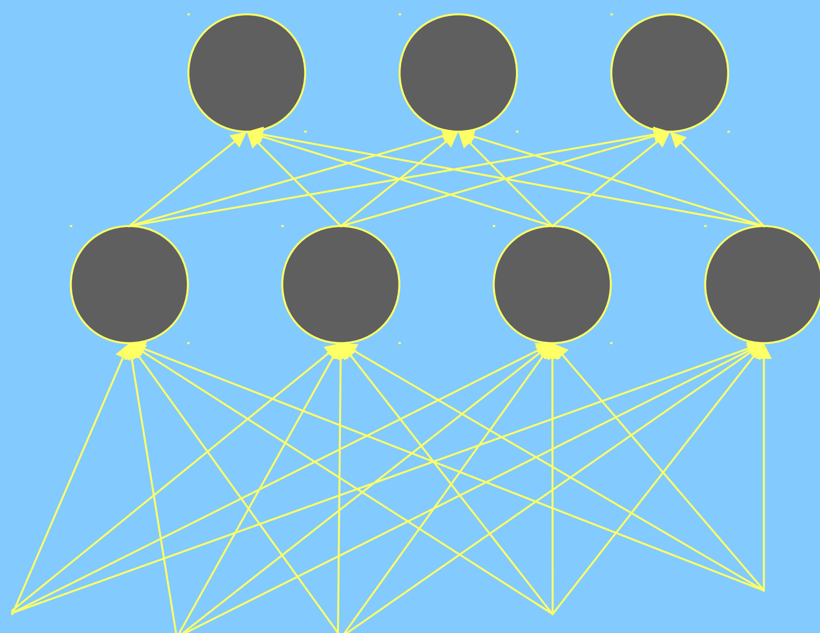
¿Esto vale para cualquier problema?

Perceptrón multicapa - Idea



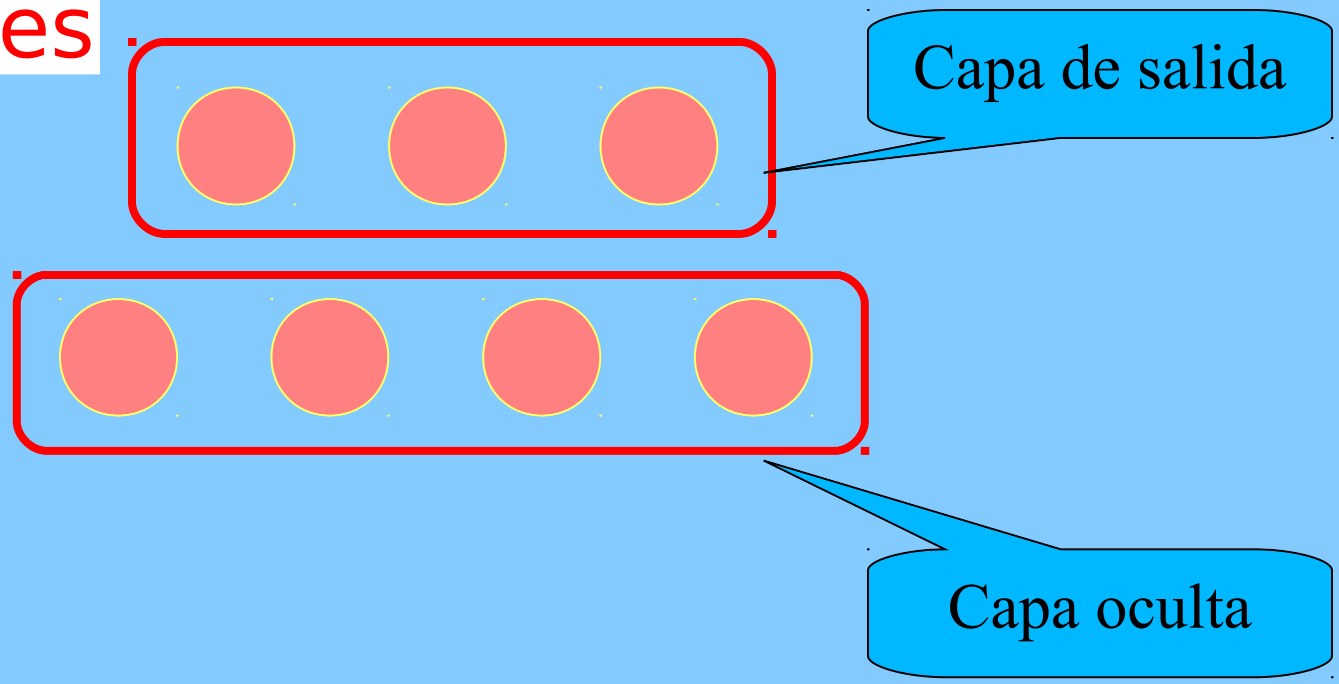
¿Entradas? ¿Salidas?

Perceptrón multicapa - Idea



Perceptrón multicapa - Idea

Procesadores

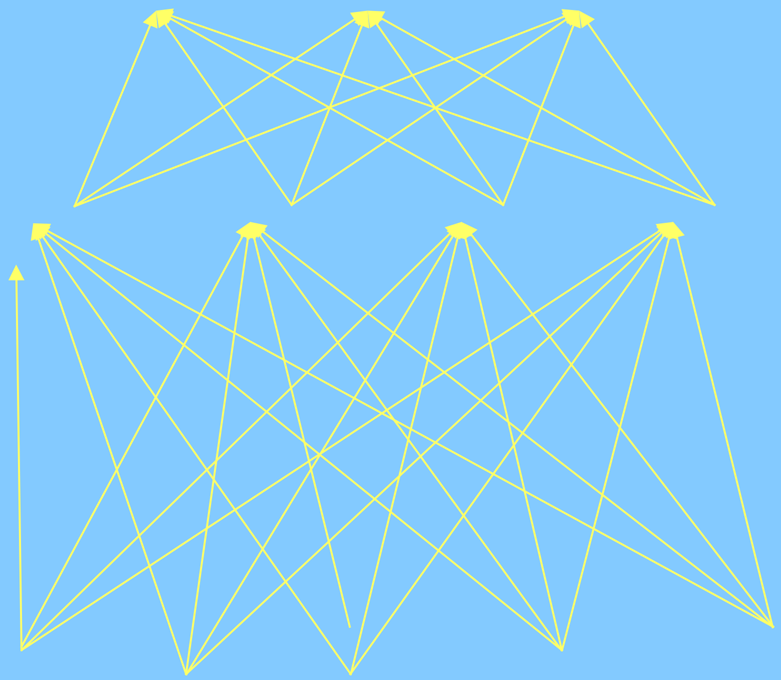


¿Cuántas capas ocultas puede haber?

¿Y de salida?

Perceptrón multicapa - Idea

Conexiones



¿Hay un patrón?

Perceptrón multicapa - Cálculo

Cada procesador recibe una serie de entradas

$$e_i = p_{i0} + \sum_j p_{ij} \, y_j$$

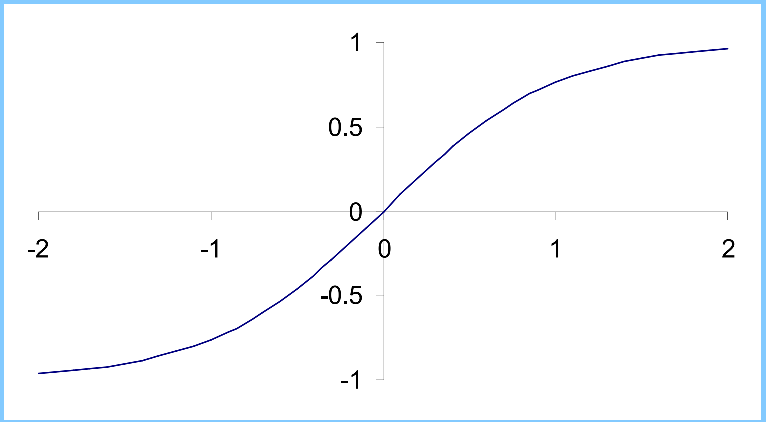
p_{ij}: peso de la conexión de procesador i a procesador j

Puede considerarse asociado a $y_0 \equiv 1$

y calcula su salida

$$y_i = \tanh(e_i)$$

Afín a la logística

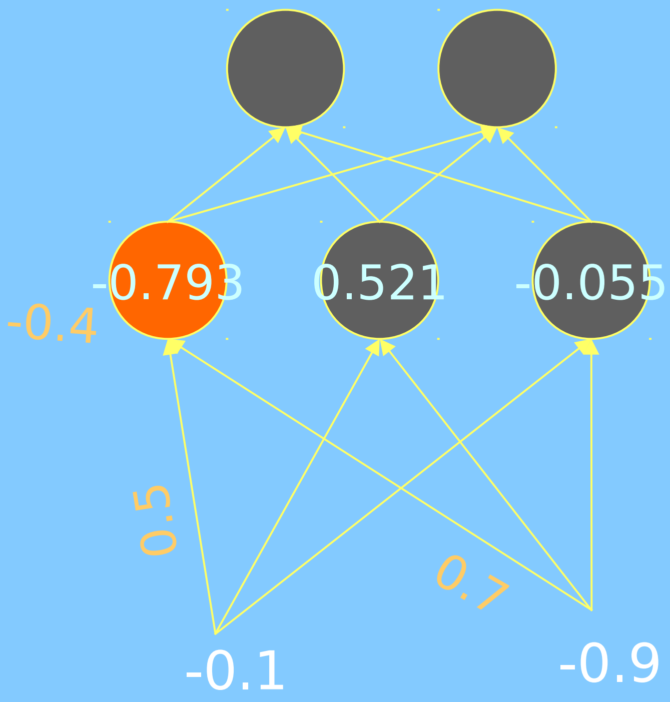


¿Qué características pedimos?

Perceptrón multicapa - Cálculo

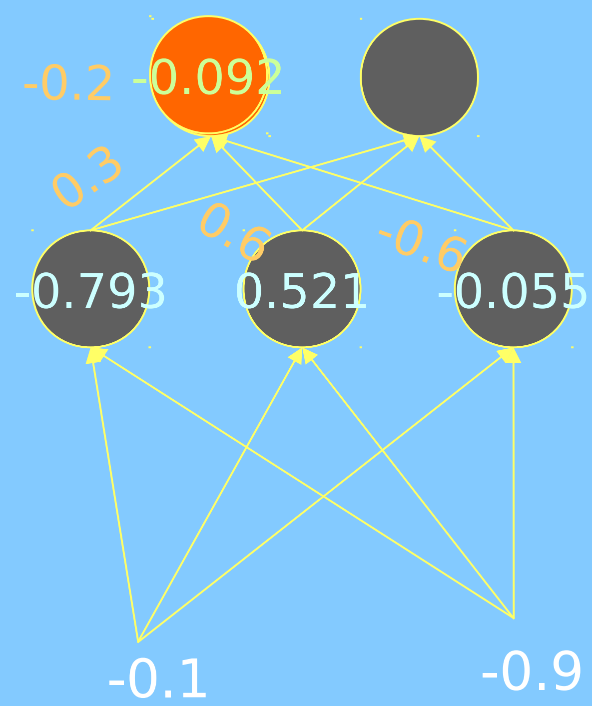
$$e = -0.4 + 0.5 \times (-0.1) + 0.7 \times (-0.9) \approx -1.08$$

$$y = \tanh(-1.08) \approx -0.793$$



Perceptrón multicapa - Cálculo

[redacted]



Perceptrón multicapa - Ideas

Cada capa es una proyección de R^m a ¿?

Cada componente es una transformación cuasilineal de R^m a ¿?

Cada componente en una capa posterior maneja un desarrollo en serie de n términos sobre R^m

El problema conjunto es aproximar cierta función R^i a R

Es regresión no lineal

Teorema Kolmogorov-Arnold: $$f(\mathbf{x}) = f(x_1, \ldots, x_n) = \sum_{q=0}^{2n} \Phi_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right)$$

Construcción de Sprecher: $$f(\mathbf{x}) = \sum_{q=0}^{2n} \Phi \left(\sum_{p=1}^{n} \lambda_p \phi(x_p + \eta q) + q \right)$$

$$\Phi\colon \mathbb{R} \to \mathbb{R} \qquad \phi\colon [0,1] \to [0,1]$$

Perceptrón multicapa – Implementación

1. Cargar vector de entradas o matriz, si queremos operar varios casos simultáneamente. Cada columna es un caso, cada fila es una variable.
2. Los pesos de todos los procesadores de una capa componen una matriz. Cada columna es el peso de una de las variables de entrada, y cada fila es un procesador. Para operar conjuntamente los términos independientes, se pueden añadir a esta matriz e incluir en la matriz entradas una fila de valor fijo uno.

Perceptrón multicapa – Implementación

3. Para obtener las combinaciones de entrada a los procesadores (e_i), ¿? Cada fila corresponde a un procesador y cada columna a un caso de entrada.
4. A continuación se aplica elemento a elemento la función de transferencia de los procesadores. Con esto tenemos ¿?, cada fila un procesador y cada columna un caso.
5. Esa matriz es la entrada de la siguiente capa, así que integramos todo el proceso por capas.

Perceptrón multicapa – Implementación

¿Elemento clave? Se recomiendan implementaciones optimizadas, biblioteca BLAS o superconjuntos. Hay equivalentes en CUDA; MATLAB/Octave y Numpy lo usan, aunque quizá no la versión más optimizada.

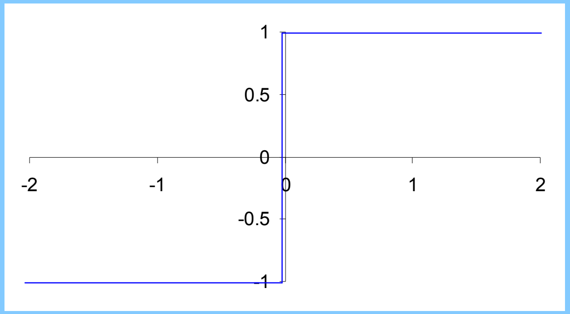
Perceptrón multicapa – Resumen

- ¿Determinista o estocástico?
- Grafo ¿dirigido o no? ¿con o sin ciclos?
- ¿Cuántas capas?
- ¿Ejecución paralela o síncrona?
- En conexiones ¿orden? ¿operación?
- ¿Función activación?

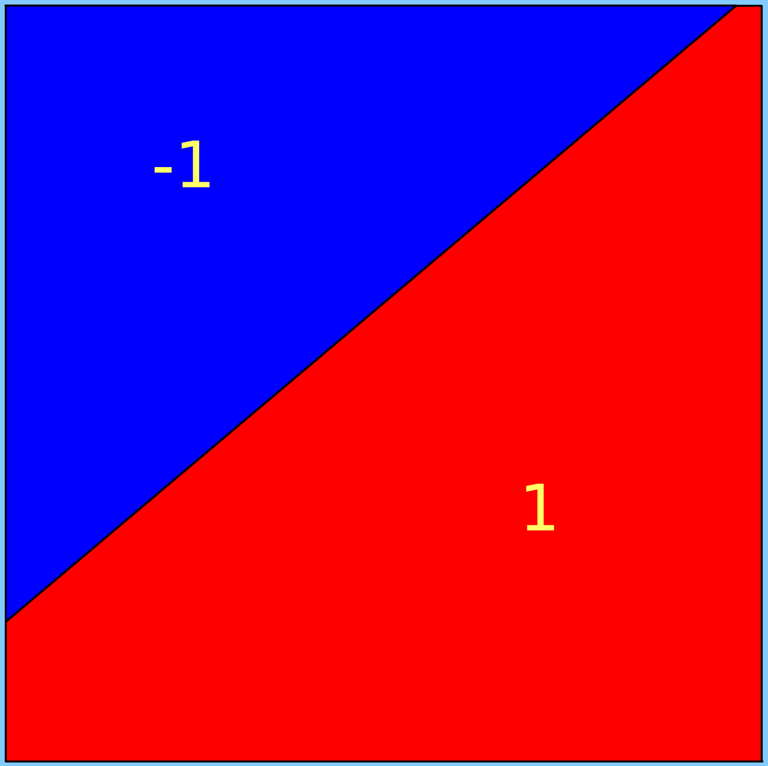
Perceptrón multicapa - Posibilidades

Un perceptrón con una capa oculta es un aproximador universal. **Ampliando suficientemente la capa oculta.**

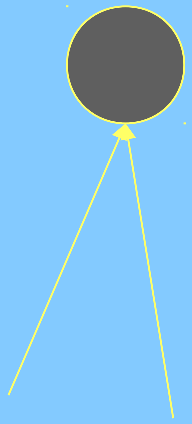
Perceptrón multicapa - Respuesta



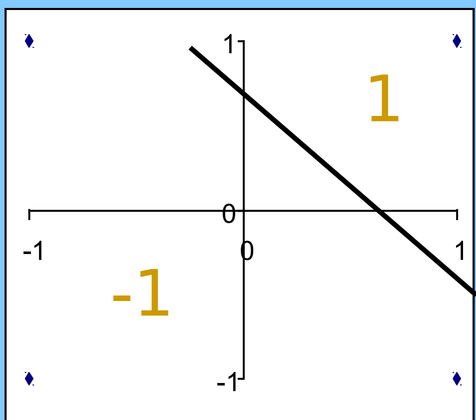
$$e = p_0 + p_1 x_1 + p_2 x_2$$

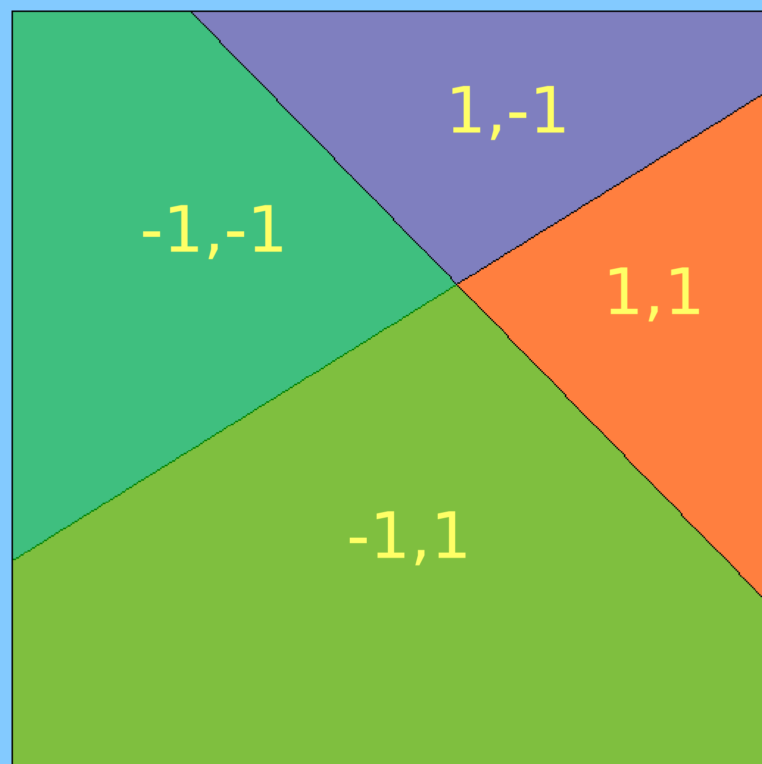


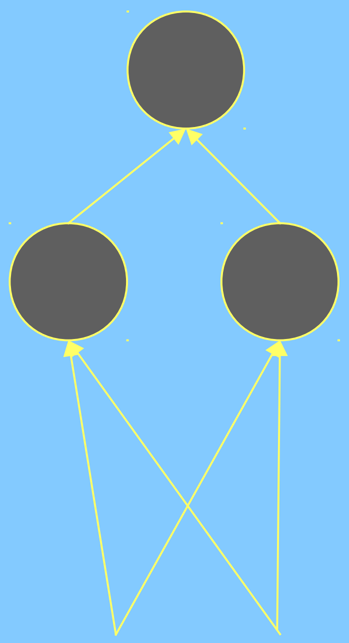
¿Qué es esto?



Perceptrón multicapa - Respuesta

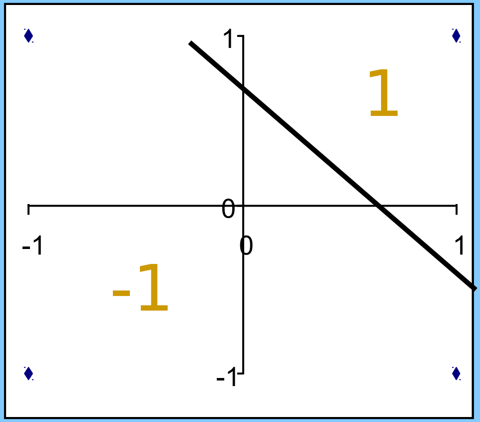




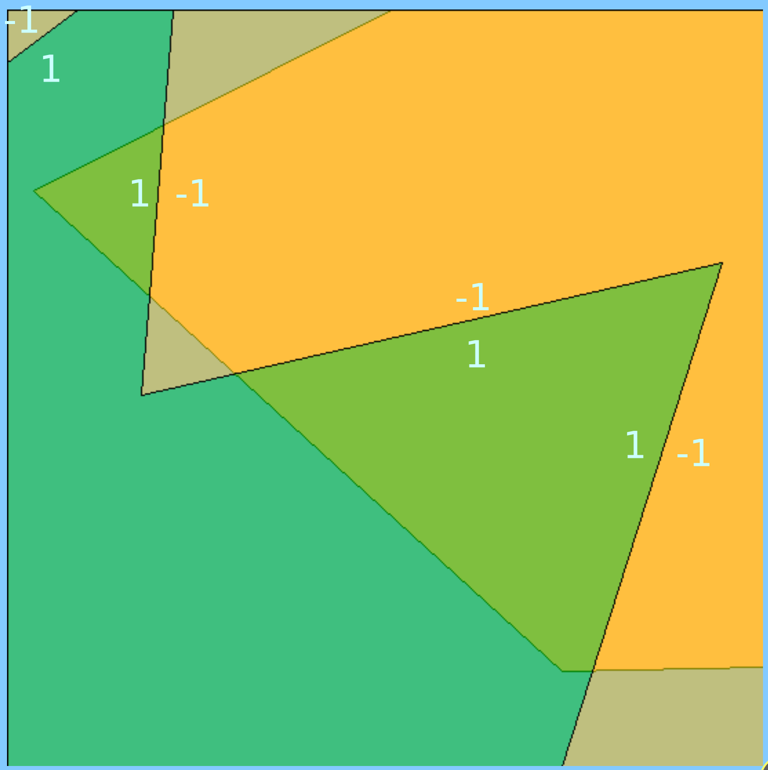


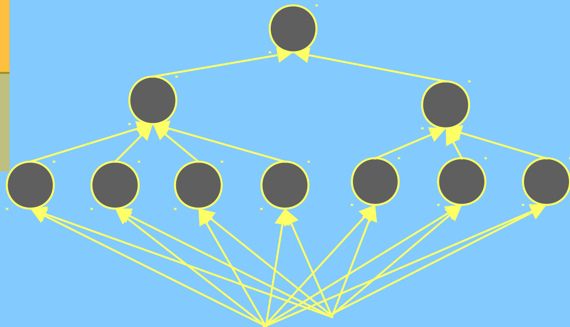
¿?

Perceptrón multicapa - Respuesta

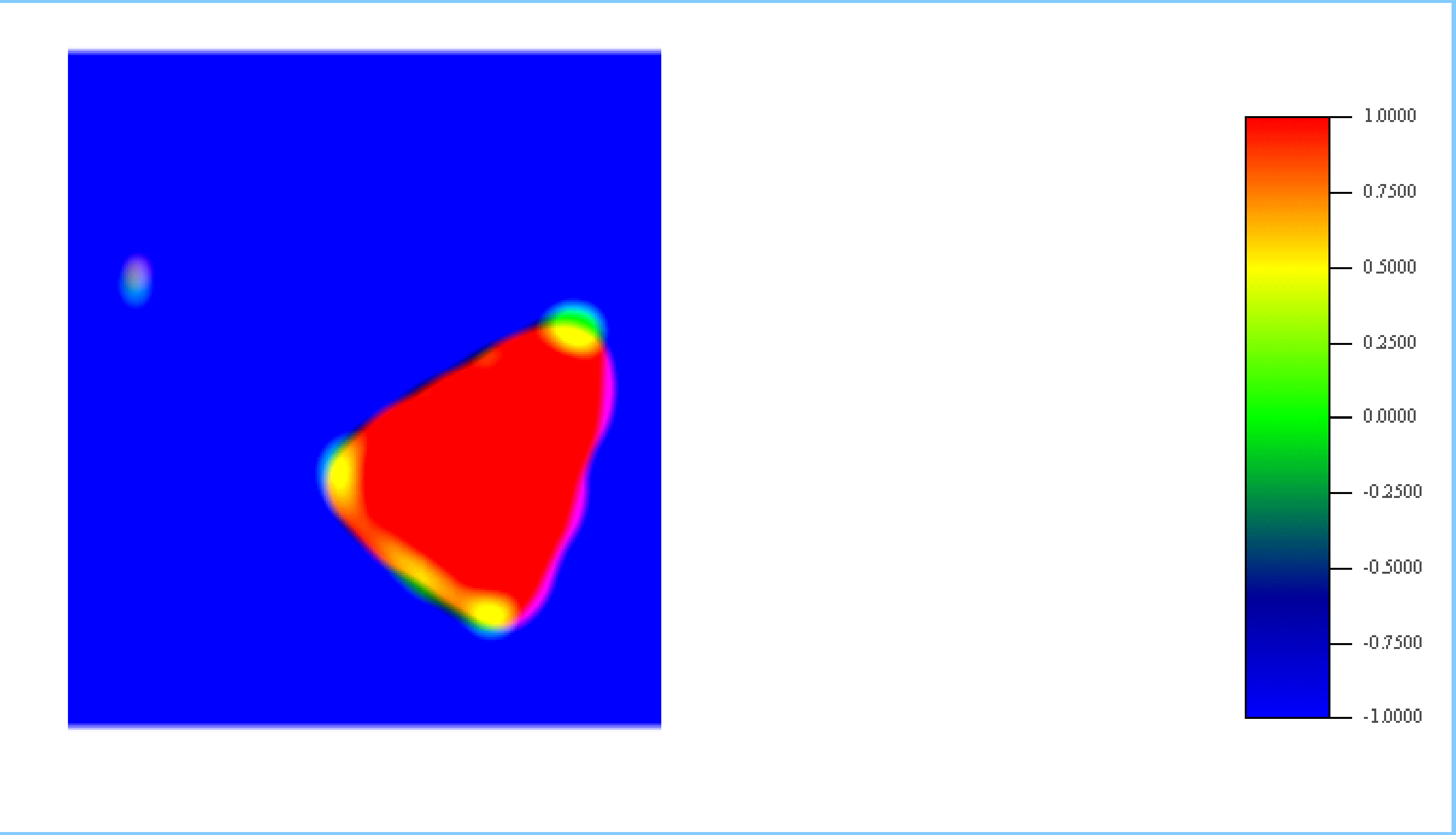


¿¿??





Perceptrón multicapa - Respuesta



¿¿¿???

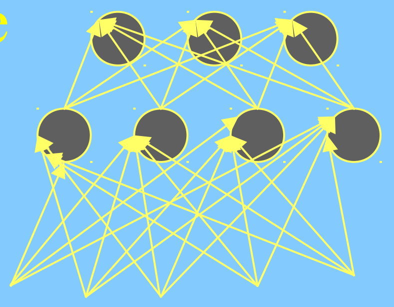
Perceptrón multicapa - Diseño

Diseño:

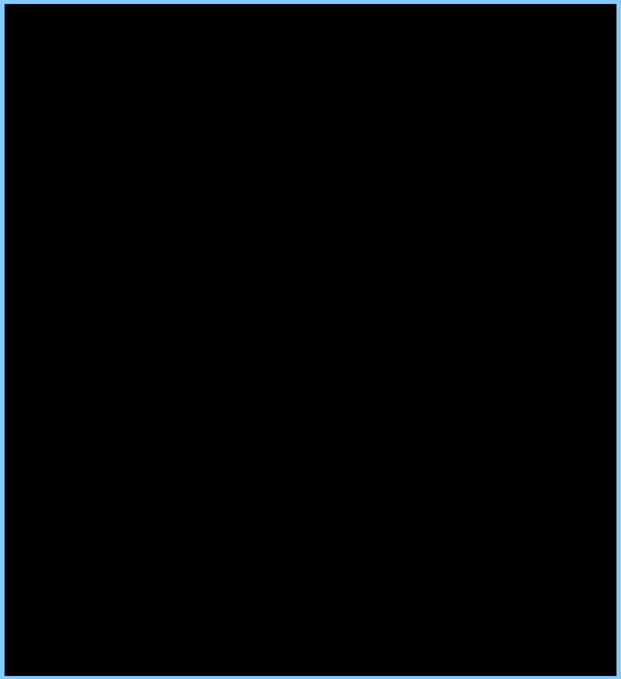
- Genérico: ¿nº de grados de libertad?, búsqueda y selección
- Constructivo: procesador a procesador, añadir y comparar
- Destructivo: grande y eliminar
- Ad-hoc: estructura de función
- Métodos especiales, p.ej., "*cascade correlation*"

Perceptrón multicapa - Aprendizaje

Ajuste de pesos: optimización basada en gradiente



$$\frac{\partial E}{\partial p_{ij}} = \left(\sum_k \frac{\partial E}{\partial y_{s_k}} \frac{\partial y_{s_k}}{\partial e_{s_k}} \frac{\partial e_{s_k}}{\partial y_i} \right) \frac{\partial y_i}{\partial e_i} \frac{\partial e_i}{\partial p_{ij}}$$



Con error cuadrático y activación tangente hiperbólica

Métodos:

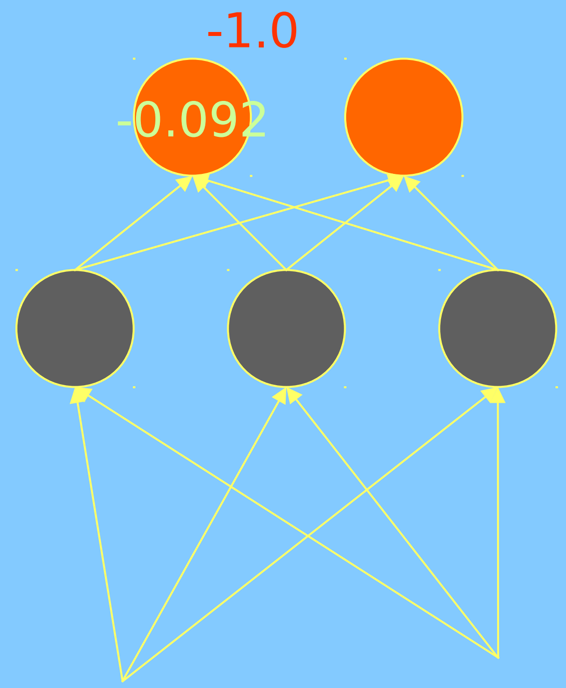
- Descenso gradiente
- Gradiente conjugado
- Levenberg-Marquardt
- Métrica variable
- ...etc...¿cuáles sabes?

Perceptrón multicapa - Aprendizaje

$$\left.\begin{aligned}\frac{\partial E}{\partial y} &= 2\left(y-c\right) = ¿? \\ \frac{\partial y}{\partial e} &= 1-y^2 = ¿?\end{aligned}\right\} \frac{\partial E}{\partial e} \approx ¿?$$

$$\frac{\partial e}{\partial y_i} = ¿?$$

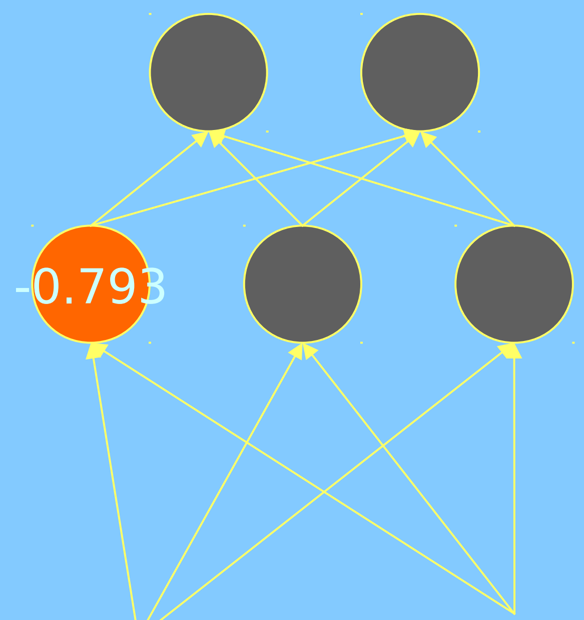
$$\frac{\partial e}{\partial p_i} = y_i \Rightarrow \frac{\partial E}{\partial p_i} = ¿?$$



Perceptrón multicapa - Aprendizaje

$$\left.\begin{array}{l} \dfrac{\partial E}{\partial y} = \displaystyle\sum_{s_k} \dfrac{\partial E}{\partial y} = 1.422 \\[2ex] \dfrac{\partial y}{\partial e} = \text{¿?} \end{array}\right\} \frac{\partial E}{\partial e} \approx \text{¿?}$$

$$\frac{\partial e}{\partial p_i} = \text{¿?} \Rightarrow \frac{\partial E}{\partial p_i} = \text{¿?}$$



Perceptrón multicapa – Implementación

Inicializar los pesos a valores aleatorios. Posibles optimizaciones escogiendo la distribución. ¿Qué distribución?

Ciclo superior:

1)Ejecutar la red sobre la muestra de casos

2)Ciclo de obtención de derivadas

3)Aplicar algoritmo de optimización para obtener nuevos valores de pesos

Iterar ¿hasta?

Perceptrón multicapa – Implementación

Ciclo de obtención de derivadas :

Aplicar la definición de la función objetivo para obtener una matriz de derivadas de la función objetivo respecto a la salida de la red. Por ejemplo: 2*(salidasred-salidasreales)

1)Obtener las derivadas respecto a las entradas combinadas de la red ¿cómo?

2)La matriz de derivadas de los pesos se obtiene ¿cómo?

Perceptrón multicapa – Implementación

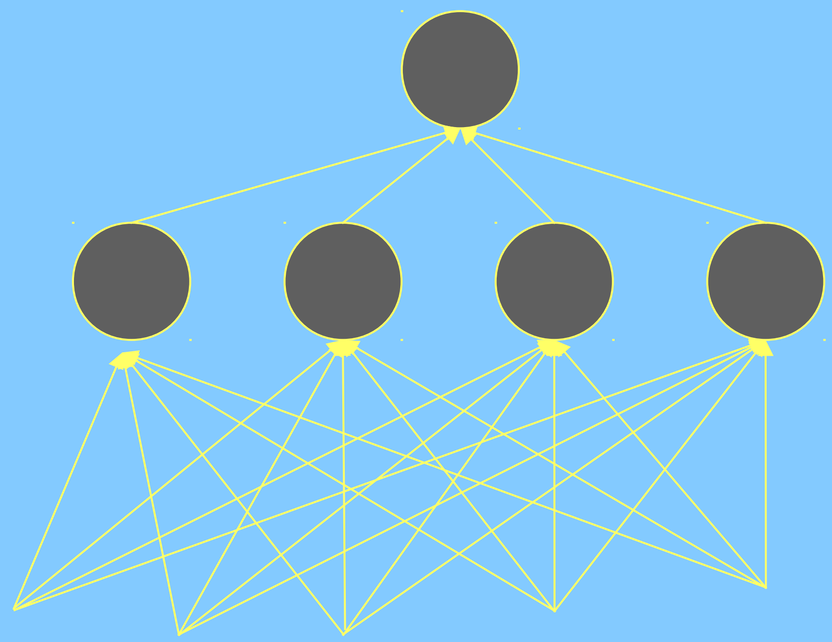
3)La matriz de derivadas respecto a las entradas de la capa (traspuesta) se obtiene multiplicando la matriz de pesos (traspuesta) por la matriz del paso 1

Esta matriz es la de derivadas respecto a la salida de la capa inferior, con lo cual iteramos el proceso hacia las capas inferiores una a una, hasta llegar a la más baja.

Tipos de redes neuronales

- Perceptrón multicapa
- Redes de respuesta radial
- Competitivas

Base radial - Idea



¿Se parece al perceptrón?

Base radial - Cálculo

Cada procesador oculto recibe una serie de entradas

$$e_i = \sum_j \left(p_{ij} - x_j \right)^2$$

y calcula su salida

$$y_i = e^{-e_i}$$

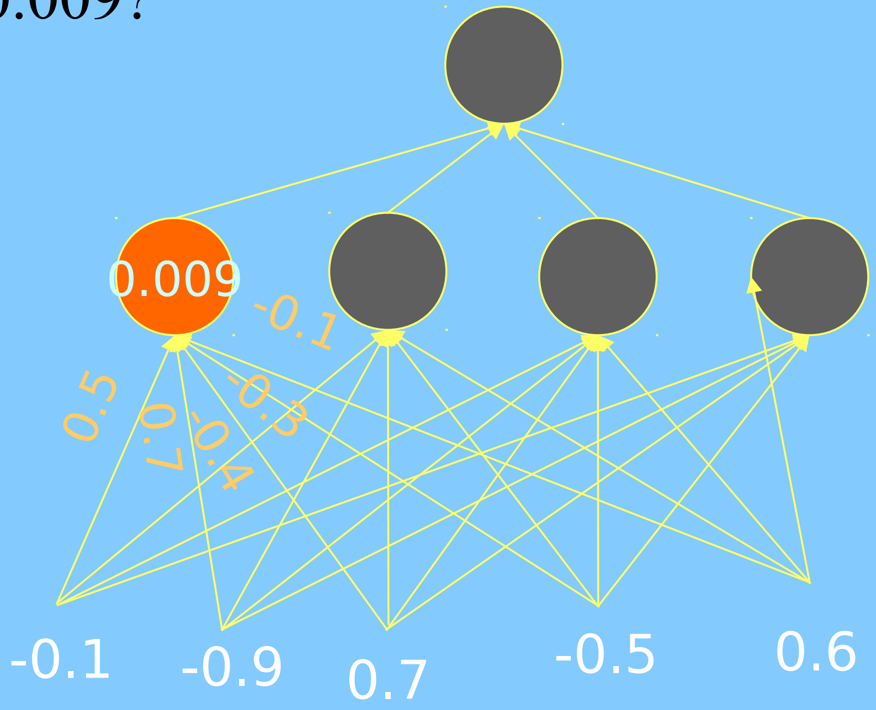
Es una expresión de una gaussiana con varianza unitaria. Podría haber una matriz de varianzas diagonal (**ancho de respuesta**) o incluso una matriz completa

El procesador de salida da una combinación lineal de los ocultos

¿Se parece al perceptrón?

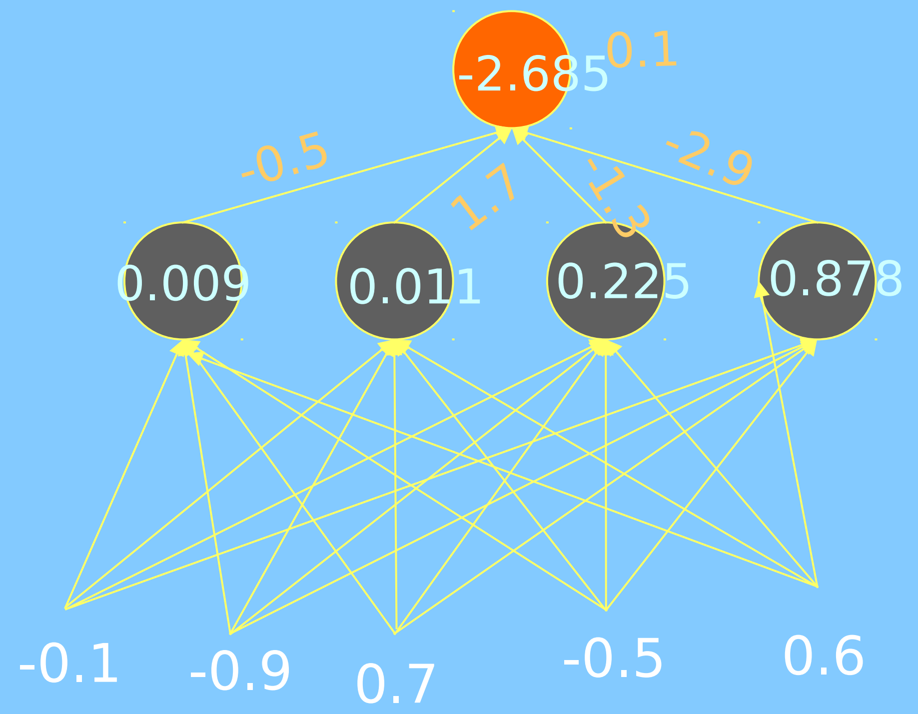
Base radial - Cálculo

¿Por qué vale 0.009?



Base radial - Cálculo

¿Por qué -2.685?



Base radial - Diseño

Diseño:

- Constructivo: procesador a procesador cuando la respuesta baja de un umbral

- ¿Otras posibilidades?

Base radial - Aprendizaje

Ajuste de pesos: una posibilidad

1. Entrada-oculta
2. Oculta-salida

Los pesos de entrada a capa oculta se ajustan buscando agrupamiento de entradas, sin tener en cuenta la salida

1. Valores iniciales: puntos de la muestra
2. Nuevo valor de pesos: media de las entradas para las que este procesador ha sido el de respuesta mayor

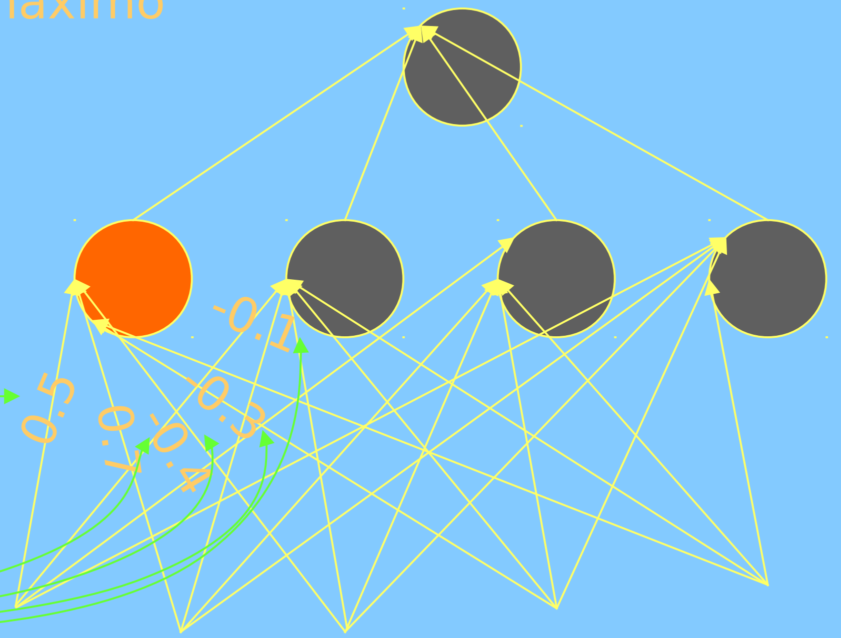
Los pesos de salida son lineales: regresión

¿O si no? ¿Qué otras posibilidades?

Base radial - Aprendizaje

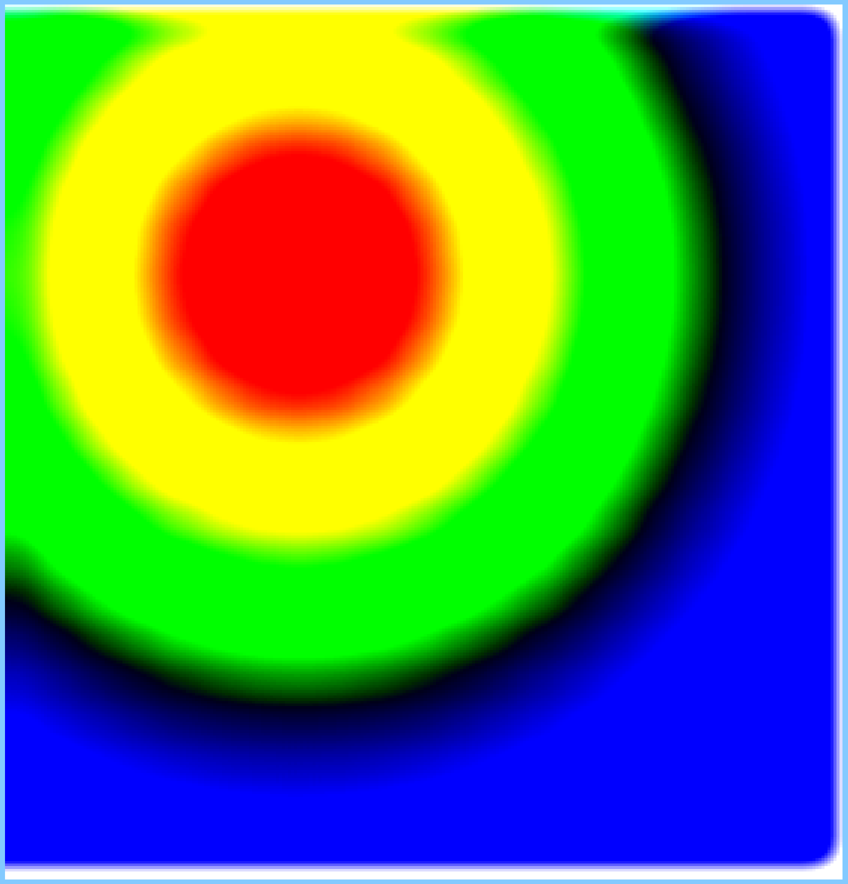
Puntos para los que ha sido el máximo

	0.55	0.6	-0.25	-0.25	0
	0.4	0.65	-0.35	-0.2	0.05
	0.5	0.55	-0.1	-0.3	-0.05
	0.3	0.65	-0.1	-0.27	-0.05
	0.6	0.55	-0.4	-0.25	-0.1
	0.5	0.7	-0.4	-0.3	-0.1
	0.6	0.75	-0.3	-0.25	-0.05
	0.5	0.65	-0.5	-0.35	0
	0.55	0.75	-0.3	-0.5	-0.2
	0.5	0.8	-0.45	-0.25	-0.15
Nuevos	**0.5**	**0.665**	**-0.315**	**-0.292**	**-0.065**



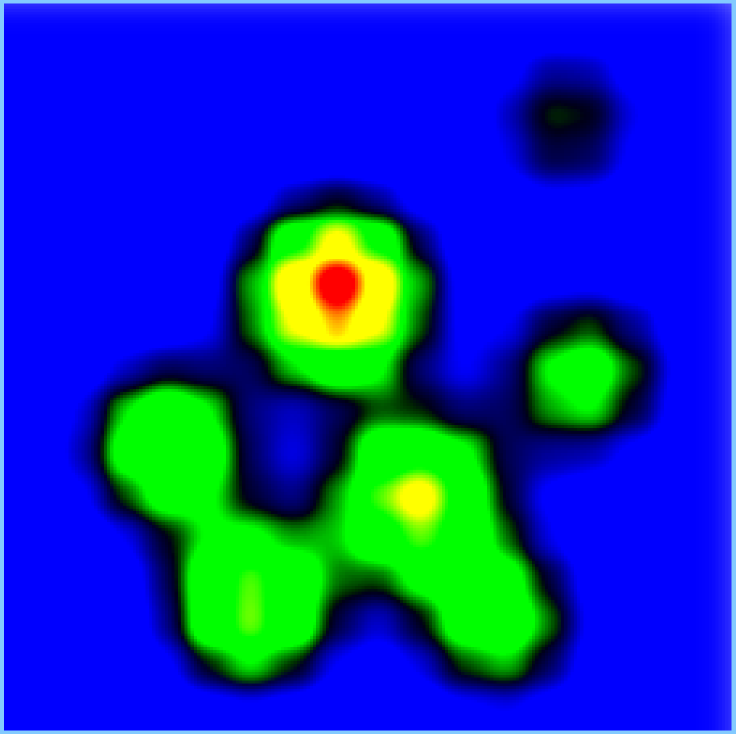
¿Qué son estas columnas?

Base radial - Respuesta



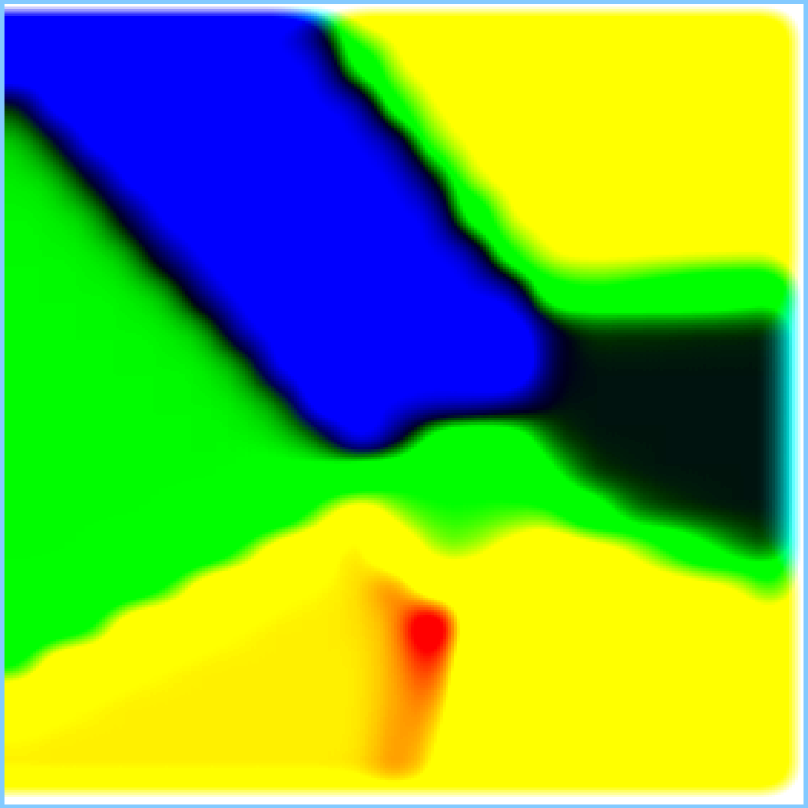
¿?

Base radial - Respuesta



¿¿??

Base radial - Respuesta

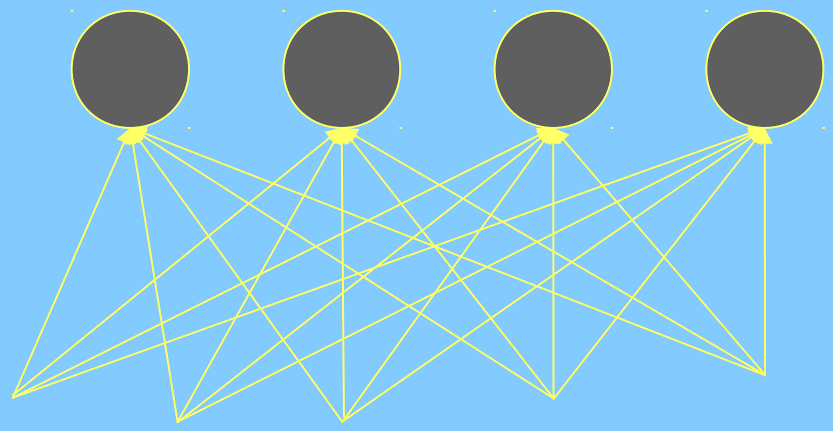


¿¿¿???

Tipos de redes neuronales

- Perceptrón multicapa
- Redes de respuesta radial
- Competitivas

Competitivas - Idea



Los procesadores pueden estar ordenados en retícula (mapa topológico autoorganizativo: MA), con coordenadas $\left(X_i, Y_i\right)$

¿Qué relación hay entre X,Y, entradas y salidas?

Competitivas - Cálculo

Cada procesador recibe una serie de entradas

$$e_i = \sum_j p_{ij} x_j$$

En MA es más usual

$$e_i = -\sum_j \left(x_j - p_{ij} \right)^2$$

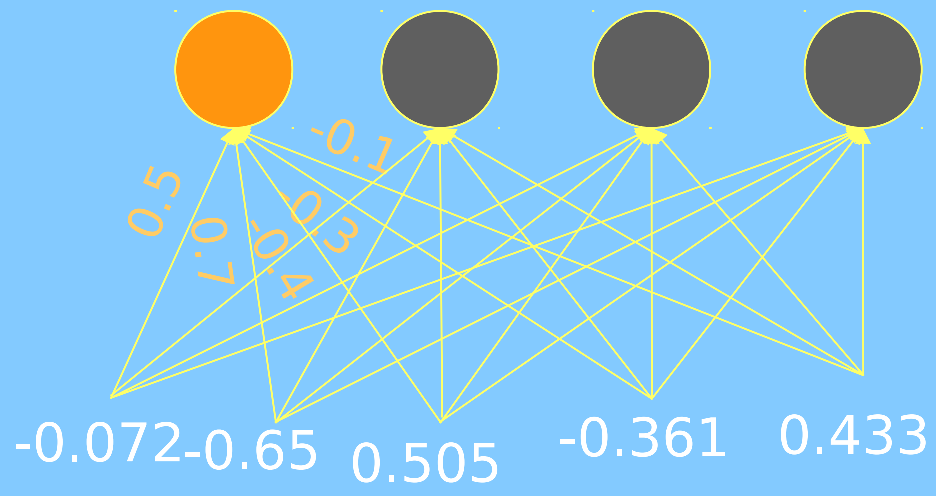
y calcula su salida

$$y_i = \begin{cases} 1 \text{ si } e_i = \max_j \left(e_j \right) \\ 0 \text{ en otro caso} \end{cases}$$

Los pesos definen un centroide de un grupo de entradas

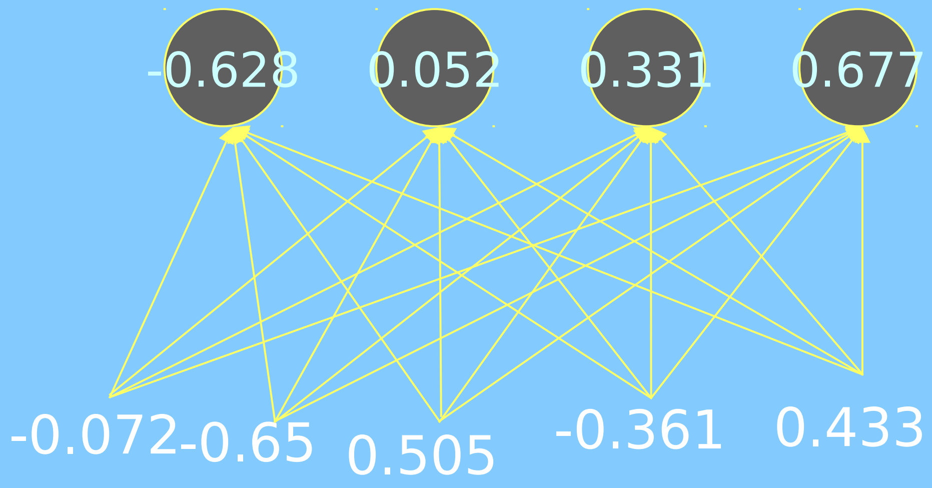
¿Se parece a otras redes?

Competitivas - Cálculo



¿Cuánto se activa el marcado?

Competitivas - Cálculo



¿Y entoces qué pasa?

Competitivas - Diseño

Diseño:

- Constructivo: procesador a procesador cuando la respuesta baja de un umbral
- Destructivo: eliminación de procesadores que no dan respuesta
- ¿O qué otra forma?

Competitivas - Aprendizaje

Ajuste de pesos normalmente no supervisado

$$\Delta p_{ij} \propto T_i \left(x_j - p_{ij} \right)$$

En competitivas a secas:

$$T_i = y_i$$

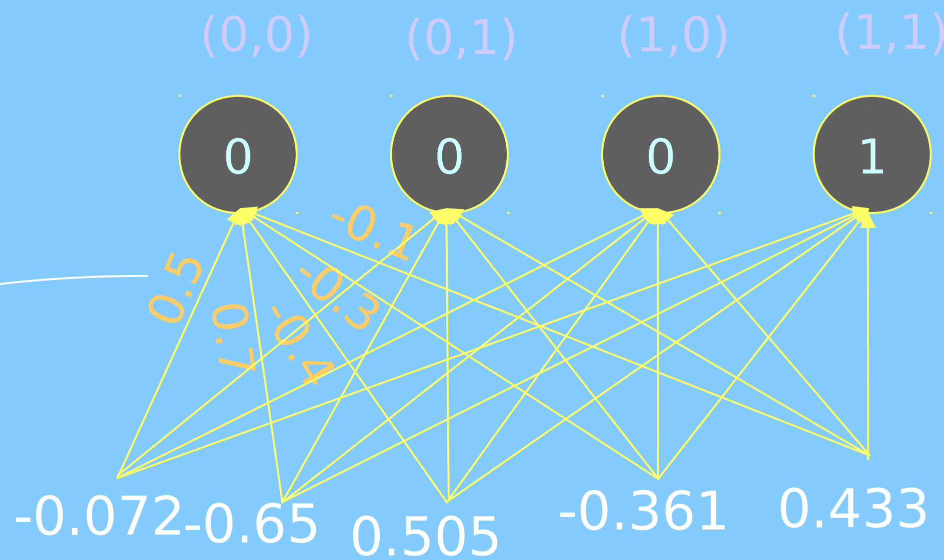
En MA:

¿Pero por qué?

$$T_i = e^{-\frac{d_{ij}^2 n^2}{2}} \text{ en la iteración } n$$

$$d_{ij} = \sqrt{\left(X_i - X_j \right)^2 + \left(Y_i - Y_j \right)^2} \text{ siendo } y_j = 1$$

Competitivas - Aprendizaje

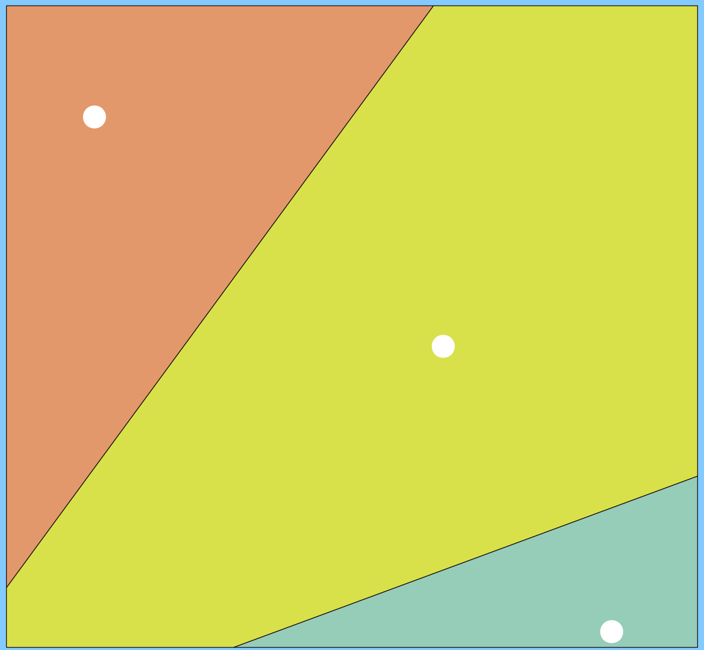


Dist	1.414	1	1	0
T(n=3)	0.0001	0.011	0.011	1

¿De dónde salen estos Dist?

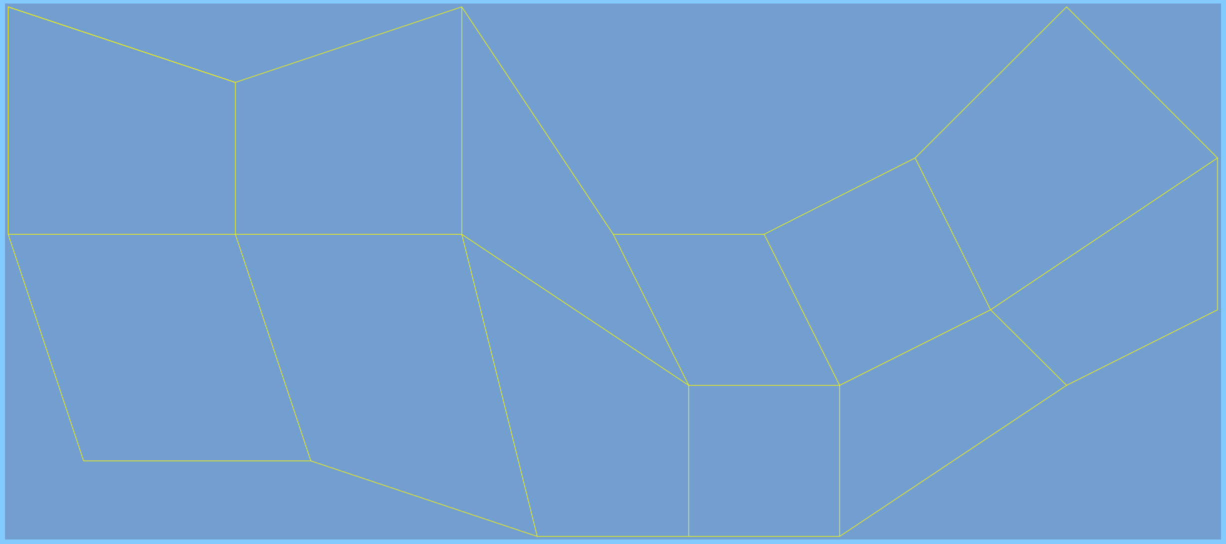
$$\Delta p \propto T\left(x-p\right)=0.0001\left(-0.072-0.5\right)\approx -7\cdot 10^{-5}$$

Competitivas - Respuesta



¿?

Mapas topológicos autoorganizativos - Respuesta



¿¿??