

MAPAS AUTORGANIZADOS

(clustering - rna)



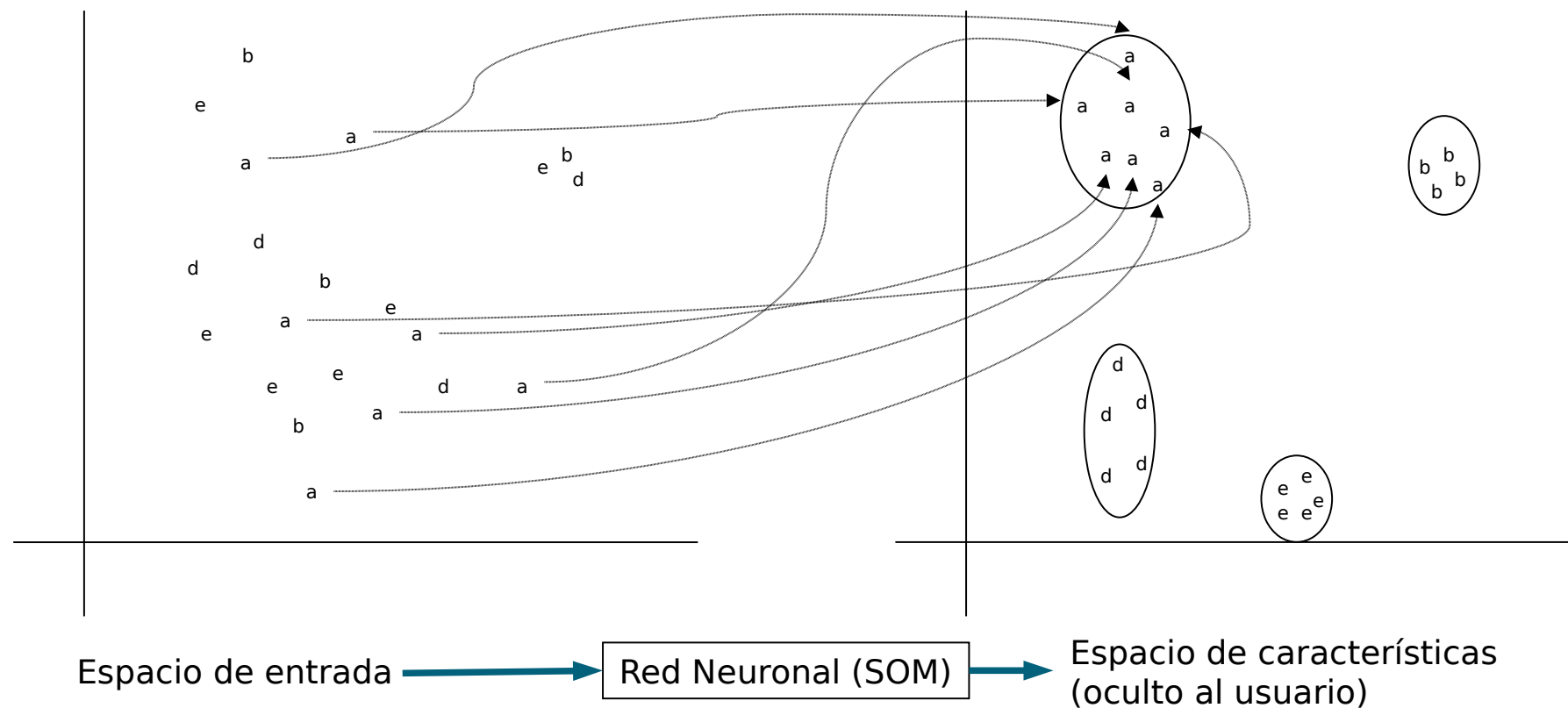
MINERÍA DE DATOS

4º Curso. Grado en Ingeniería Informática
5º Curso. Doble Grado Informática/Estadística (INDAT)

Clustering con Redes Neuronales

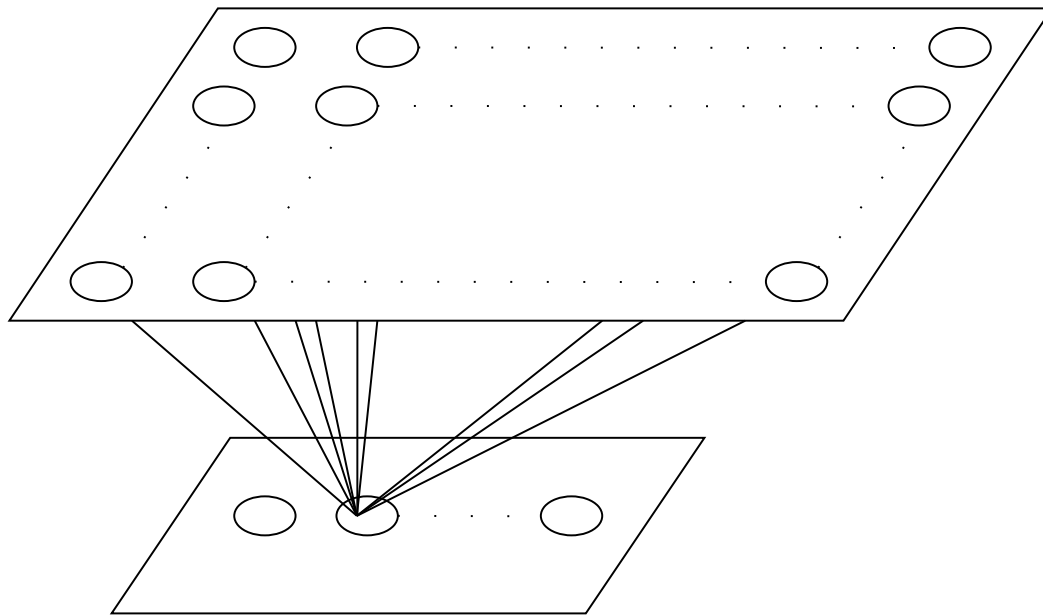
- Se encuadraría en los métodos **no jerárquicos**
- Paradigma de **aprendizaje** no supervisado y **competitivo**
- A nivel teórico realiza un **agrupamiento** en un **espacio de características** totalmente opaco al usuario
- Cada **neurona** vendría a representar un **cluster**
- Dada la cantidad de ellas, se suelen agrupar, a su vez, en **zonas topológicamente cercanas**, esto es, en **vecinas**
- La **interacción** entre neuronas es “vecinal”, **no** por **conexiones** directas ponderadas por un peso

Agrupamiento en un espacio de características



Arquitectura y funcionamiento

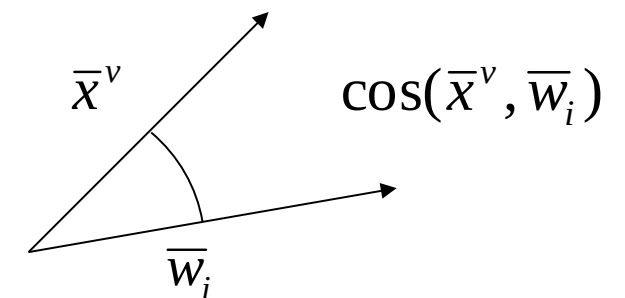
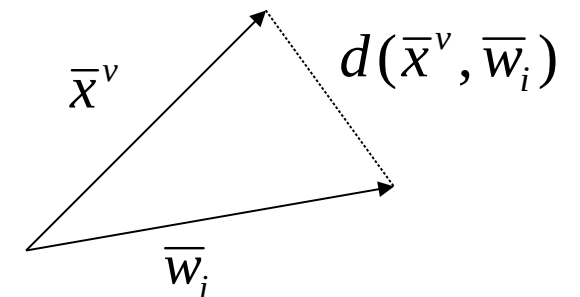
Capa de salida



Capa de entrada

Salida de la neurona i-ésima:

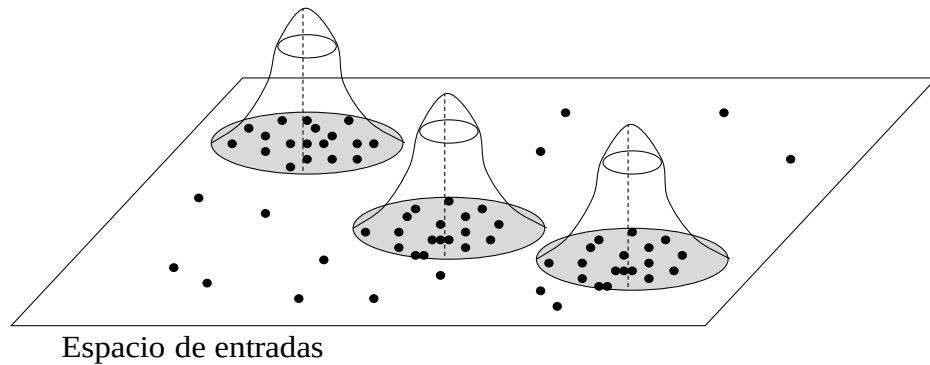
- **Distancia** entre vectores de pesos y entrada



Salida con entradas y pesos normalizados:

- **Ángulo** entre vectores de entrada y pesos
- El coseno sería igual al **producto escalar**
- Producto escalar igual a modelo de **McCulloch-Pitts**

Comparativa con RBF



- Ambas salidas se basan en una función radial:

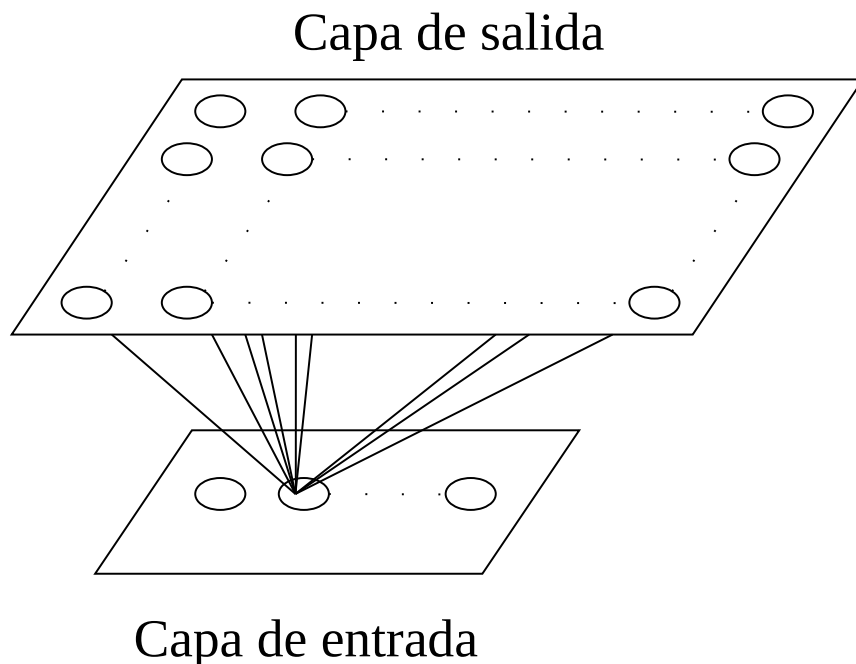
- Gausiana distancia al cuadrado (RBF)
- Distancia (SOM)

- La distancia es respecto:

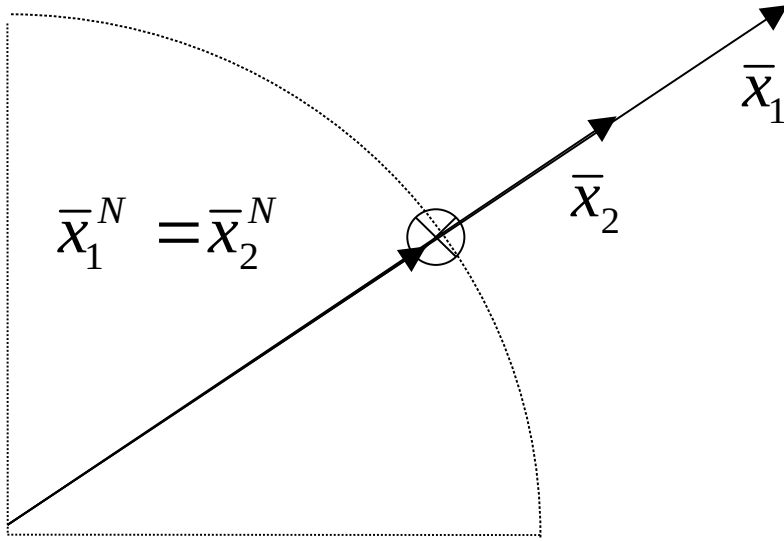
- Baricentro (RBF)
- Vector de pesos (SOM)

- El agrupamiento por distancia en el espacio:

- Entradas (RBF)
- Características (SOM)



Normalización

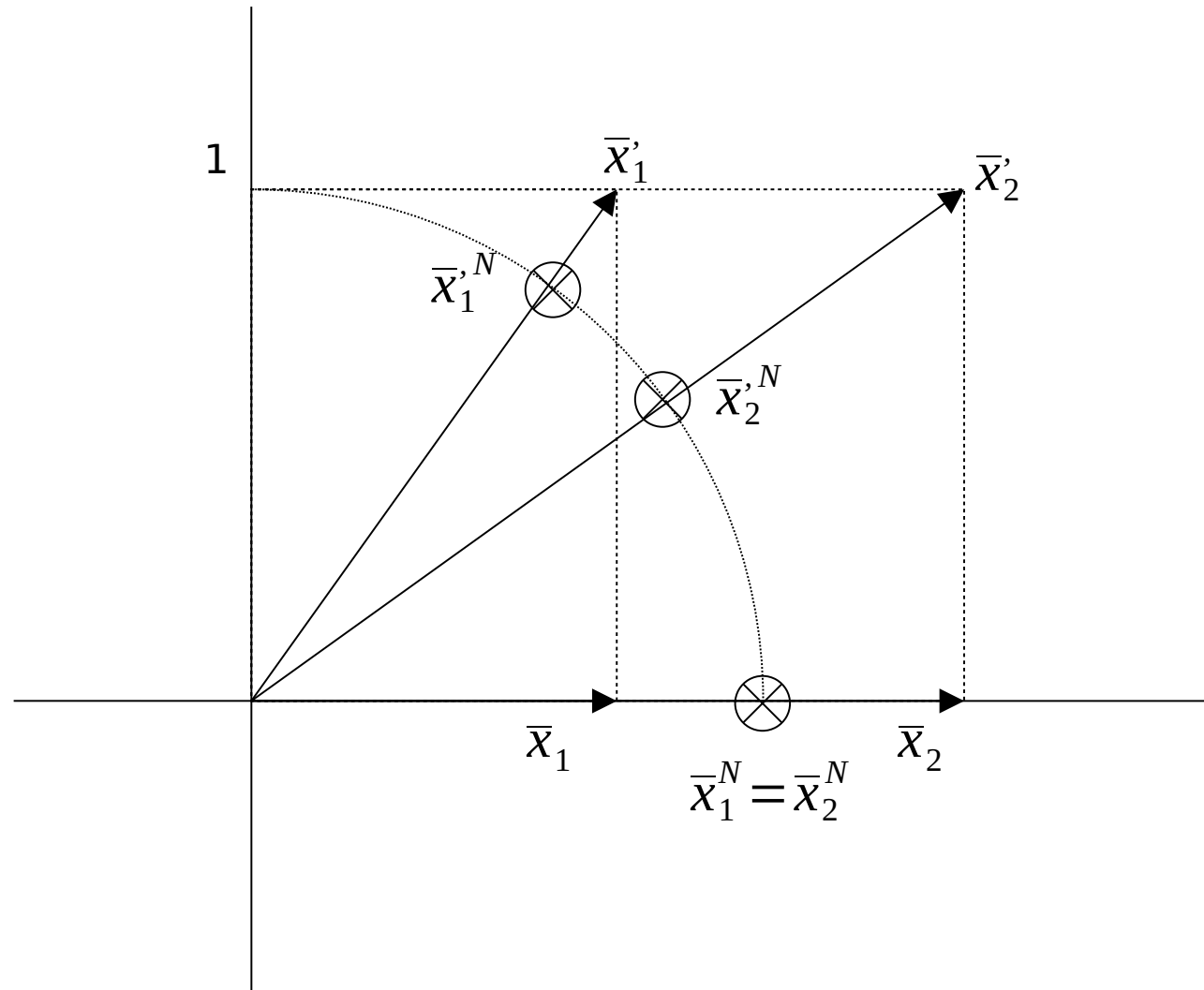


- Consiste en transformar todos los vectores de entrada a norma uno.
- Geométricamente es proyectar sobre la esfera de radio unidad.
- Porque podría suceder que:

$$\text{Si } \bar{x}_1 = \alpha \bar{x}_2 \Rightarrow \left\{ \begin{array}{l} \bar{x}_1^N = \frac{\bar{x}_1}{\|\bar{x}_1\|} = \frac{\alpha \bar{x}_2}{\alpha \|\bar{x}_2\|} \\ \bar{x}_2^N = \frac{\bar{x}_2}{\|\bar{x}_2\|} \end{array} \right\} \Rightarrow \bar{x}_1^N = \bar{x}_2^N$$

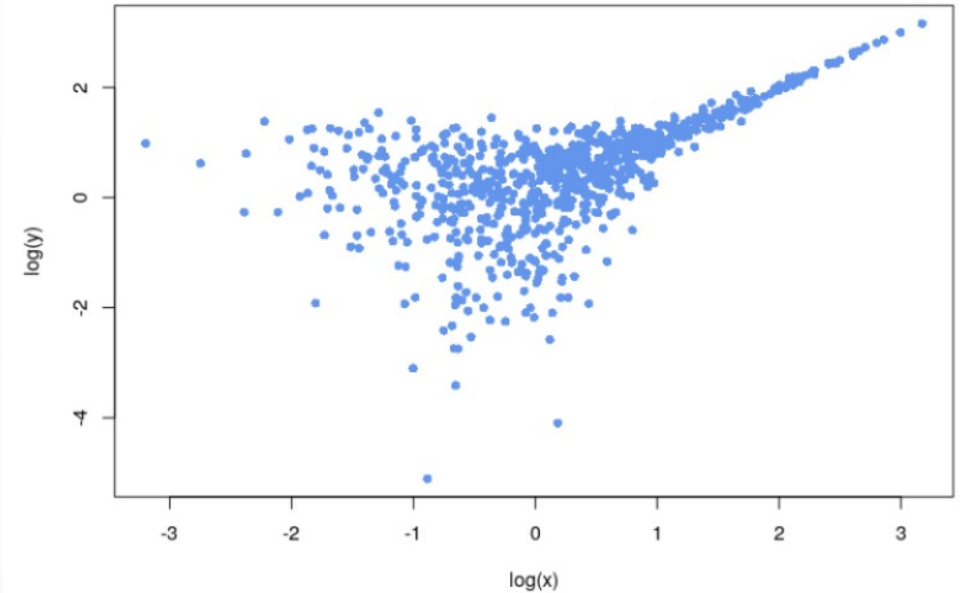
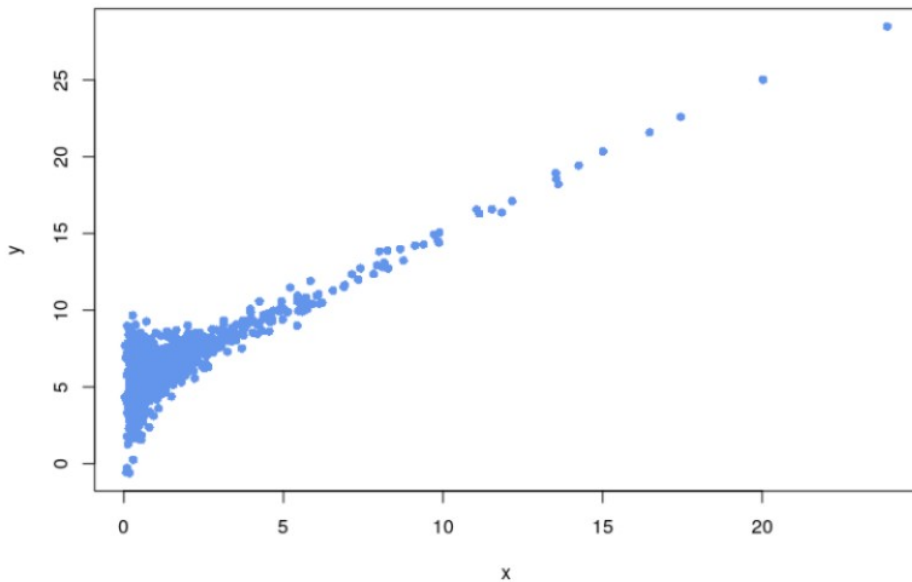
Normalización extendida

- Caso ilustrativo de que las entradas sean de una dimensión

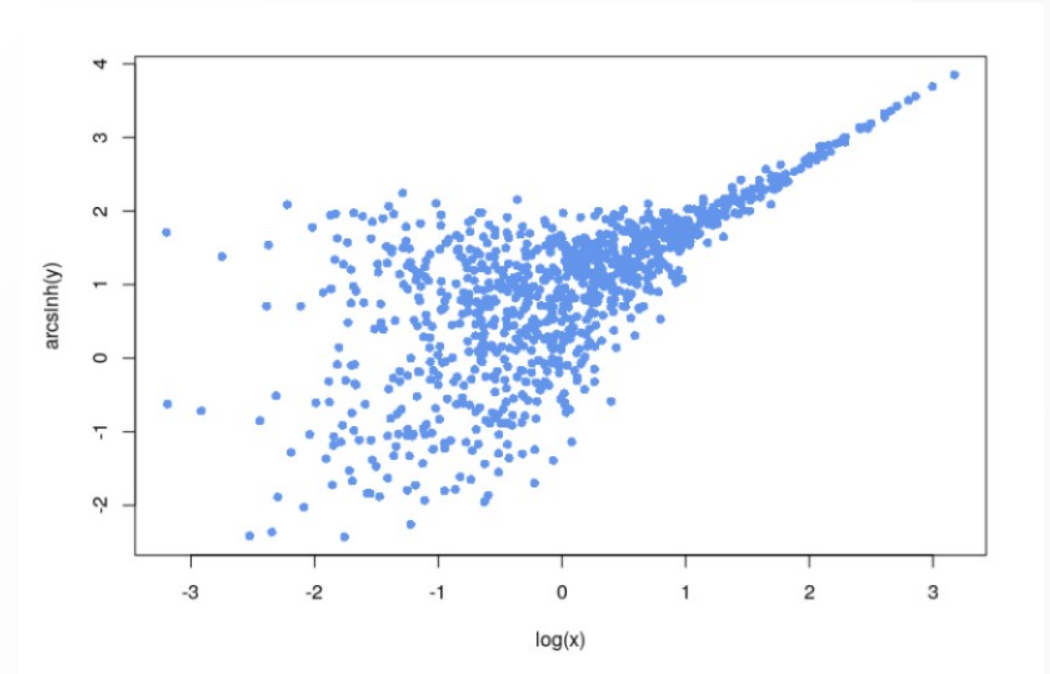
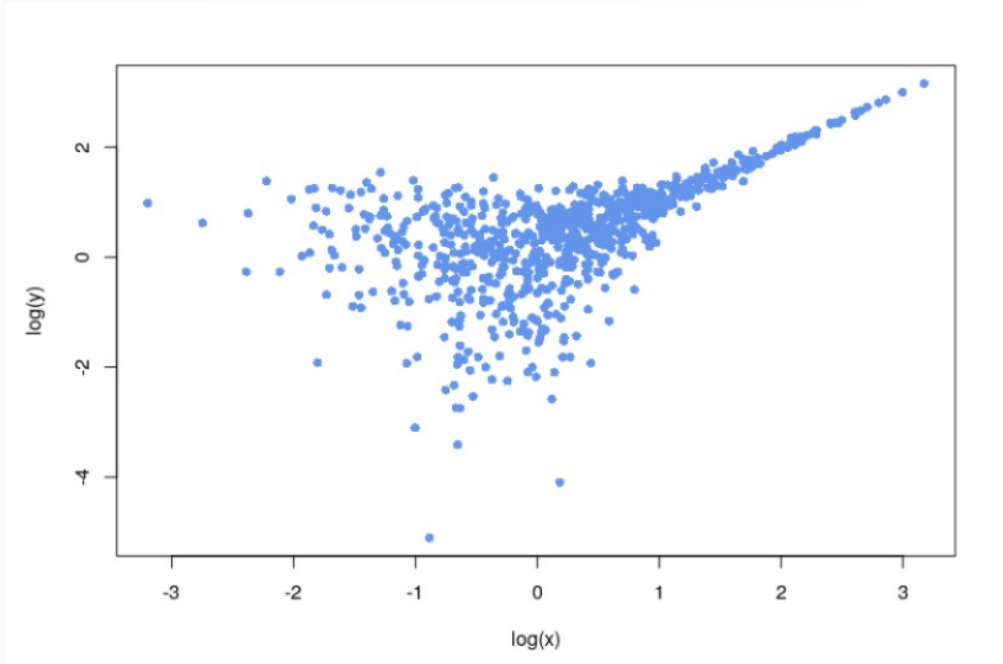


ESCALADO

- Esta transformación se usa para ESCALAR \neq NORMALIZAR
 - A veces surgen datos que hay que aproximar al resto, para no ser tratados como puntos aislados



Transformación arcsh



$$\text{arcsh}(x) = \ln(x + \sqrt{x^2 + 1})$$

- Permite tratar con valores negativos
- Recupera puntos que quedarían un tanto aislados de usar el logaritmo

Aprendizaje: neurona más próxima a la entrada

- Ante una entrada, calcular la **salida** de cada neurona

$$d_i(\bar{x}^v, \bar{w}_i) = \sqrt{\sum_{j=0}^{n-1} (x_j^v - w_{ij})^2}$$

- Si los pesos y las entradas están normalizadas, se puede hacer uso del coseno de ángulo:

$$\bar{x}^v \bar{w}_i = |\bar{x}^v| |\bar{w}_i| \cos(\bar{x}^v, \bar{w}_i) = \cos(\bar{x}^v, \bar{w}_i) = \sum_{j=0}^{n-1} x_j^v w_{ij}$$

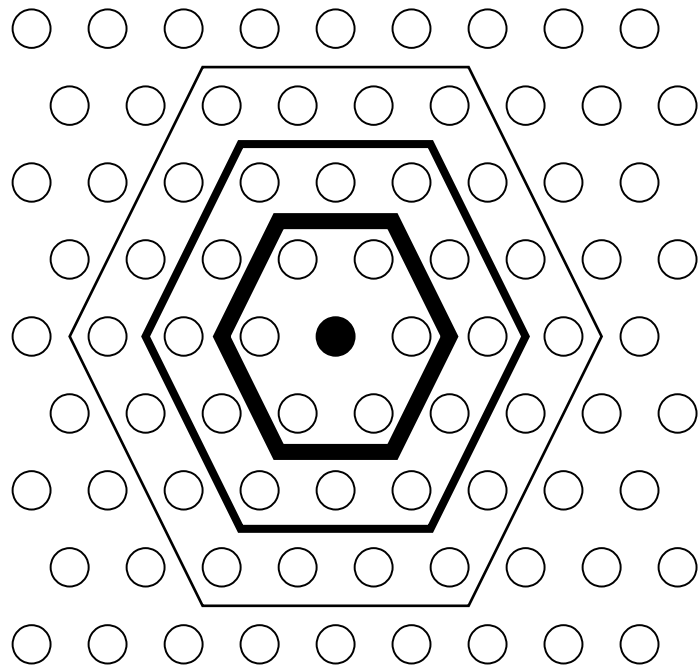
- En este caso coincide con el modelo de **McCulloch-Pitts**
- Y se obtiene la **más próxima** a la entrada:

$$I = \min_{i \in H} d_i(\bar{x}^v, \bar{w}_i)$$

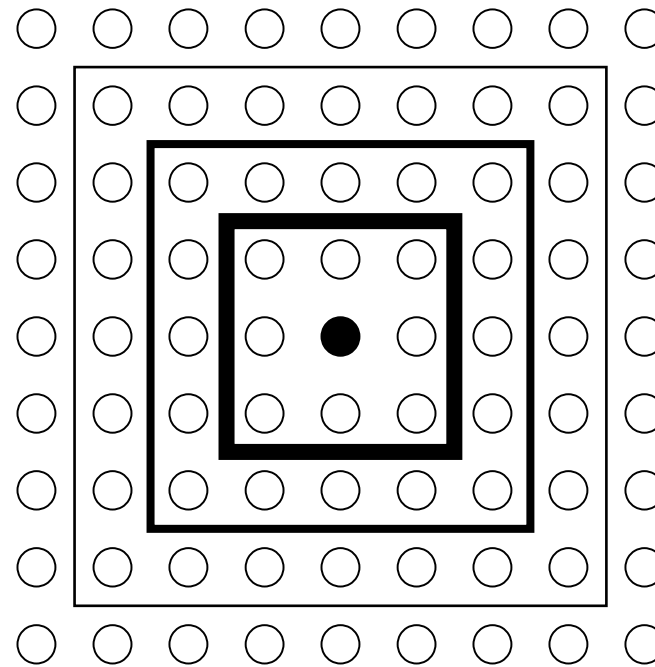
$$I = \max_{i \in H} \cos(\bar{x}^v, \bar{w}_i)$$

Topología rectangular

- Clasificación de SOM:
 - Disposición espacial de nodos
 - Relación de **vecindad**



(a)



(b)

— R=3
— R=2
— R=1
— R=0

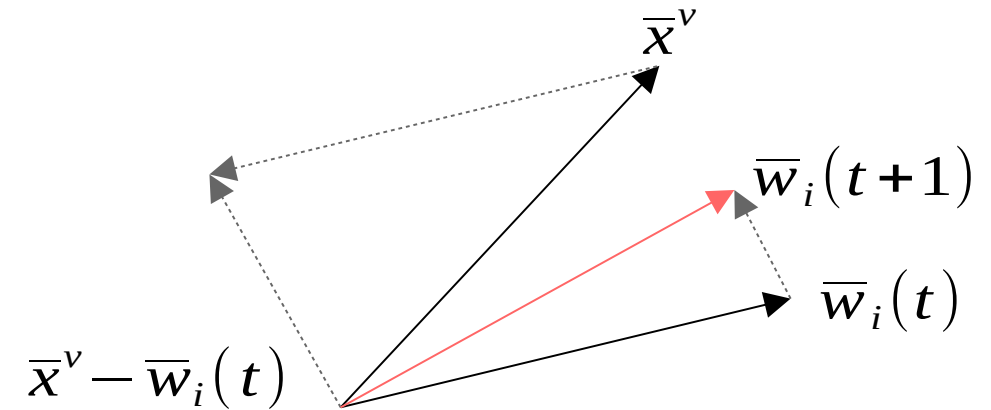
Algoritmo de aprendizaje

- Actualizar el radio de vecindad:
 - Se empieza con uno que abarque a toda una dimensión (la más pequeña) del mapa.
- Para cada muestra en una época
 - Calcular la **salida** de cada neurona
 - Obtener la neurona **ganadora**
 - Generar el conjunto $N(I)$, que depende de:
 - La relación de vecindad o **topología** (ej: rectangular)
 - **Radio** (cuando es cero se queda en la neurona ganadora)
 - Modificar los **pesos**

Final de aprendizaje = n^o de iteraciones prefijadas

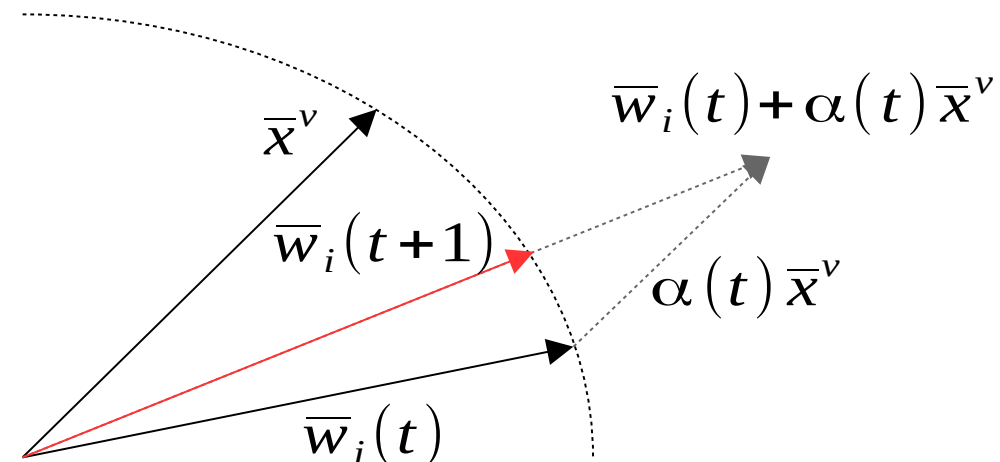
Actualización de los pesos

$$\bar{w}_i(t+1) = \begin{cases} \bar{w}_i(t) + \alpha(t)(\bar{x}^v - \bar{w}_i(t)) & \forall i \in N(I) \\ \bar{w}_i(t) & \forall i \notin N(I) \end{cases}$$



- Con entradas y pesos normalizados:

$$\bar{w}_i(t+1) = \begin{cases} \frac{\bar{w}_i(t) + \alpha(t)\bar{x}^v}{\|\bar{w}_i(t) + \alpha(t)\bar{x}^v\|} & \forall i \in N(I) \\ \bar{w}_i(t) & \forall i \notin N(I) \end{cases}$$



Factor de aprendizaje / Vecindad

- Debe ser grande al inicio e ir decreciendo conforme aumenta las iteraciones

$$\alpha(t) = \alpha_0 + (\alpha_f - \alpha_0) \frac{t_\alpha}{t}$$

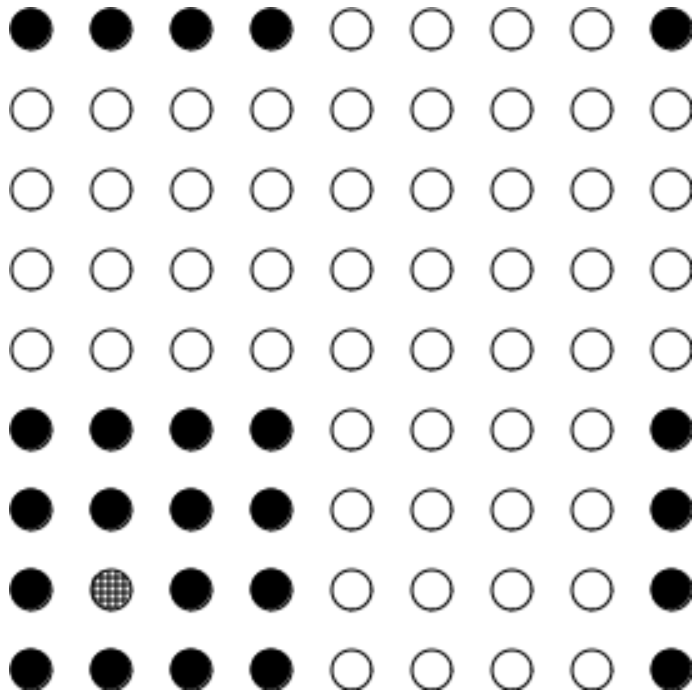
$$\alpha(t) = \alpha_0 \left[\frac{\alpha_f}{\alpha_0} \right]^{t_\alpha / t}$$

$$\alpha(t) = \frac{\alpha_0}{1 + \frac{t}{P}}$$

La **más utilizada**, donde P es el tamaño de una época

- **N(i)** decreciente con las iteraciones
 - **Radio** de vecindad:
 - Va decrementándose por épocas completas
 - Hasta alcanzar $R=0$: la propia neurona

Cliclicidad del SOM



Neurona ganadora



Neuronas vecinas

pseudo-código:

```
vecinas_por_fila = []
for f in f_max-R:f_max+R
    if f < 1
        append!(vecinas_por_fila, FILAS + f)
    elseif f > FILAS
        append!(vecinas_por_fila, f - FILAS)
    else
        append!(vecinas_por_fila, f)
    end
end

vecinas_por_columna = []
for c in c_max-R:c_max+R
    if c < 1
        append!(vecinas_por_columna, COLUMNAS + c)
    elseif c > COLUMNAS
        append!(vecinas_por_columna, c - COLUMNAS)
    else
        append!(vecinas_por_columna, c)
    end
end

for i in vecinas_por_fila
    for j in vecinas_columnas
        neurona[i,j] ← ACTUALIZAR PESOS
    end
end
```

Cuantización Vectorial

- Al final del algoritmo, se ha conseguido:

$$T: R^M \rightarrow R^N$$

- Donde $M > N$
- Por tanto, se tiene una reducción de la dimensionalidad, que se puede interpretar como:
 - Compresión
 - Extracción de características (Minería de Datos),
 - No hay la vuelta atrás. No importa. Lo relevante es cómo serán las diferencias entre muestras en el nuevo espacio.

Clasificación

- Se puede conseguir con el etiquetado por **moda local**:
 - Crear una **lista**, por neurona, con las **etiquetas** de las muestras, que han hecho a dicha neurona ganadora.
 - Acabada la pasada por todas las muestras, etiquetar cada neurona con la **moda** de su lista de etiquetas creada en el punto anterior.
- O por etiquetado por **moda global**:
 - Impone que una neurona sólo se use, si sólo gana con muestras de una **misma clase**
 - Si no, se etiqueta como **error**
 - Igual si no gana **nunca**.

Etiquetado por neuronas

- Se fija **una neurona** y se registran la salida de todas las muestras, a través de una lista de:
 - 1) **Distancias** al vector de pesos
 - 2) **Coseno** del ángulo con dicho vector
- Se etiqueta esta neurona con la correspondiente a la muestra ganadora en el ranking anterior:
 - 1) Mínimo
 - 2) Máximo
- Ventajas:
 - No hay neurona sin etiquetar
 - Es prácticamente imposible que haya conflicto de etiquetas

Práctica (entrega)

- Crear un SOM de 15x9 (rectangular) para clasificar dígitos manuscritos del conjunto de datos FASHION-MNIST:
 - Factor de aprendizaje inicial de 1.
 - Total de épocas = 20
- Con la salida del SOM por cada muestra, construir un conjunto de
 - Aprendizaje: (60000, 135)
 - Test: (10000, 135)
- Alimentar a un MLP (135x60x10)