

# MICROREDES ( $\mu$ REDES)

Título: Microredes ( $\mu$ Redes)

Subtítulo: V3.0.0

Autor: Amro Xpike (propietario del canal de Youtube "Xpikuos")

Formato: A5 = 148 x 210 mm.

Páginas: 16.

Licencia:



Este trabajo está licenciado bajo la licencia de **Atribución-NoComercial-CompartirIgual 4.0 Internacional (CC BY-NC-SA 4.0)**

Para ver una copia de esta licencia, visita

<https://creativecommons.org/licenses/by-nc-sa/4.0/deed.es>

Usted es libre de:

- Compartir — copiar y redistribuir el material en cualquier medio o formato.
- Adaptar — remezclar, transformar y crear a partir del material.
- El licenciador no puede revocar estas libertades mientras cumpla con los términos de la licencia. Bajo las condiciones siguientes:
  - Reconocimiento — Debe reconocer adecuadamente la autoría, proporcionar un canal a la licencia e indicar si se han realizado cambios. Puede hacerlo de cualquier manera razonable, pero no de una manera que sugiera que tiene el apoyo del licenciador o lo recibe por el uso que hace.
  - No Comercial — No puede utilizar el material para una finalidad comercial.
  - Compartir Igual — Si remezcla, transforma o crea a partir del material, deberá difundir sus contribuciones bajo la misma licencia que el original.
  - No hay restricciones adicionales — No puede aplicar términos legales o medidas tecnológicas que legalmente restrinjan realizar aquello que la licencia permite. Al reutilizar o distribuir la obra, tiene que dejar bien claro los términos de la licencia de esta obra. Alguna de las condiciones puede no aplicarse si se obtiene el permiso del titular de los derechos de esta obra. Nada en esta licencia menoscaba o restringe los derechos morales del autor

## 1. Definición

Una  $\mu$ Red es una estructura inspirada en la neurociencia, las cadenas de Markov, las redes bayesianas, los árboles de decisión, la lógica borrosa, y en las redes neuronales artificiales, si bien difiere de éstas en los siguientes puntos:

- Los pesos y las neuronas tienen sentido.
- Los valores iniciales de los pesos no son aleatorios, lo cual permite repetir los experimentos y analizar mejor los resultados.
- No existen las fases de aprendizaje y de "recuerdo" como procesos separados. La red aprende y "recuerda" constantemente.
- No usa el algoritmo de Backpropagation (a menos que se requiera realizar una regresión precisa (aunque esto está aún por ver, pero en cualquier caso, en principio, el algoritmo de Backpropagation es, de momento, algo opcional)). En su lugar se basa en el aprendizaje de Hebb y en la señal de "ECO" que se dan en ciertas neuronas biológicas.
- Los hiperparámetros tienen significados claros.
- No hay que indicar el número de capas ocultas ni el número de neuronas por capa. Sólo hay que definir la capa de entrada, y la de salida. Por defecto hay una capa intermedia que generará internamente las unidades de procesamiento (neuronas) necesarias, sus interconexiones y las estructuras jerárquicas convenientes para llevar a cabo el aprendizaje.
- El procesamiento es totalmente paralelizable usando hilos convencionales.
- No requiere de cálculos pesados, por lo que no requiere de GPUs.
- Ella misma lleva a cabo su propio mantenimiento interno eliminando aquellas conexiones, neuronas, y estructuras que vayan quedándose obsoletas con el paso del tiempo.
- Se puede saber con exactitud qué neuronas han intervenido en la generación de una determinada salida y "verbalizar" el "camino" seguido por la información dentro de la estructura para conseguir dicha salida. Es decir, tanto su funcionamiento, como el flujo de información que dentro de ella fluye es "traceable", "debuggeable", y expresable en términos fácilmente comprensibles. Por tanto, podemos saber por qué la  $\mu$ Red ha generado cierta salida.

- La información aprendida por una mRed puede ser fusionada con otra mRed que haya aprendido conceptos semejantes aún cuando la estructura de ambas mRedes sea distinta (siempre y cuando, eso sí, se cumplan ciertos requisitos mínimos. En concreto: entradas y salidas con las mismas funcionalidades y significados en ambas mRedes tienen que estar etiquetadas de la misma forma). Del mismo modo, se pueden extraer "trozos" de estructuras y de información de una mRed para crear otras mRedes sin tener que empezar de cero. Por todo esto, el "Transfer learning" aplicado en las redes neuronales es también permitido en las mRedes, sin embargo, la "ablación" está totalmente prohibida aquí ya que en una mRed todo tiene sentido, por lo que la eliminación de una parte de la mRed podría tener efectos devastadores. No obstante, una mRed tiende al uso mínimo de recursos (y por extensión, de conexiones, capas, unidades de procesamiento (i.e. neuronas o XCeldas), etc., y ella misma se encarga de eliminar las estructuras y conexiones que vayan quedando obsoletas mediante un proceso inspirado en la apoptosis neuronal.
- Si el aprendizaje se realiza, en los primeros momentos, con patrones claramente diferenciables del resto, la  $\mu$ Red aprenderá mucho más rápido ya que aprende comparando lo aprendido anteriormente con lo que se tiene que aprender en ese momento.
- Para comprobar el aprendizaje se desactiva el incremento de todos los contadores y la señal de ECO, y se contabilizan los errores cometidos para cada patrón de entrada de testeo. Esto nos dará un % de aciertos.
- Es capaz de interactuar con el usuario durante el proceso de aprendizaje y preguntar si debe aprender un patrón que se sale de lo normal y/o detectar una posible falsa clasificación en los patrones de entrenamiento.
- Se inspira en aspectos del funcionamiento de las redes de neuronas biológicas.

Consta de varias partes:

- XCeldas
- Capas con XCeldas
- Bloques
- Grupos
- Columnas

## **2. Partes de una $\mu$ Red**

### **2.1 XCelda**

NOTA: A continuación se designará:

- con el índice 'h' a todo lo que lleve implícito el concepto de "anterior"
- con el índice 'i' a todo lo que lleve implícito el concepto de "local"
- con el índice 'j' a todo lo que lleve implícito el concepto de "destino"

- con el índice 'k' a todo lo que lleve implícito el concepto de "tiempo" o "iteración"

Una XCelda 'i' es la unidad mínima de procesamiento de una  $\mu$ Red. Si hubiera que poner un símil se podría decir que es equivalente a una neurona de una red neuronal.

- Tiene un número  $L_i$  (*Level*) que indica el nivel al que pertenece dentro de la estructura arbórea que se va formando. Este parámetro sólo tiene utilidad de cara a facilitar la representación gráfica de la estructura de XCeldas que se da dentro de una capa-AND).
- Cada vez que una XCelda<sub>i</sub> da la orden de crear otra nueva XCelda<sub>j</sub>, le asigna a esta otra un nivel  $L_j$  (más adelante se explica cómo asignar este valor).
- Este nivel será usado para ejecutar ordenadamente las operaciones y facilita la implementación del mecanismo de sístole(lectura de los valores de entrada y precesamiento de los mismos)-diástole(salida de los valores internos calculados) para paralelizar operaciones.
- Tiene un identificador 'ID<sub>i</sub>' que contiene cierta información que posteriormente será útil para identificar la XCelda de forma unívoca en toda la  $\mu$ Red.
- Tiene un contador  $N_{ii}$  que contabiliza el número de veces que esta XCelda es activada. Inicialmente tiene el valor  $N_{ii}=1$  evitar las divisiones por cero.
- Tiene canales de entrada y de salida. Cada canal 'ij' (que apunta, o conecta la XCelda 'i' con la XCelda 'j') consta de:
  - Contador de excitación ( $N_{ij}^+$ ): cuenta las veces que el canal 'ij' ha contribuido con un valor positivo durante el aprendizaje. Inicialmente tiene el valor  $N_{ij}^+ = 0$ .
  - Contador de inhibición ( $N_{ij}^-$ ): cuenta las veces que el canal 'ij' ha contribuido con un valor negativo durante el aprendizaje. Inicialmente tiene el valor  $N_{ij}^- = 0$ .
  - Señal de información analógica ( $A_{ij}$ ): contiene la información ponderada de las 'h' entradas de la XCelda 'i' según esta expresión:

$$A_{ij}=f(\sum_h A_{hi} \cdot (P_{hi}^+ - P_{hi}^-) \cdot W_{hi}), \text{ siendo } P_{hi}^+ = N_{hi}^+ / N_{ii}, \quad P_{hi}^- = N_{hi}^- / N_{ii}$$

Según esta expresión y comparándola con la de un perceptrón  $Y_{ij}=f(\sum_h X_{hi} \cdot W_{hi})$ :

- $W_{hi}$  serían los pesos de un perceptrón (usados aquí, sólo para hacer un ajuste fino de cara a realizar una regresión. Sólo para ese caso se usaría el Backpropagation tradicional, donde  $f$  sería una RELU. Inicialmente, todos tienen el valor 1)
- $A_{hi} \cdot (P_{hi}^+ - P_{hi}^-)$  serían equivalentes a las entradas  $X_{hi}$  de dicho perceptrón
- $A_{ij}$  sería equivalente a la salida  $Y_{ij}$  de dicho perceptrón

Por otra parte, si  $W_{hi}=1$  en  $A_{ij}$ , entonces tendríamos:

$$A_{ij} = f(\sum_h A_{hi} \cdot (P_{hi}^+ - P_{hi}^-)), \text{ siendo } P_{hi}^+ = N_{hi}^+ / N_{ii}, \quad P_{hi}^- = N_{hi}^- / N_{ii}$$

Según esta expresión y comparándola con la de un perceptrón  $Y_{ij} = f(\sum_h X_{hi} \cdot W_{hi})$ :

- $(P_{hi}^+ - P_{hi}^-)$  serían equivalentes a los pesos  $W_{hi}$  de un perceptrón.
- $A_{hi}$  serían equivalentes a las entradas  $X_{hi}$  de dicho perceptrón
- $A_{ij}$  sería equivalente a la salida  $Y_{ij}$  de dicho perceptrón

NOTA: Si no se va a usar el algoritmo de la Backpropagation, se supondrá que

$$A_{ij} = \sum_h A_{hi} \cdot (P_{hi}^+ - P_{hi}^-)$$

ya que el “enrutamiento” de la información, la ordenación de la misma mediante XCeldas-AND y XCeldas-OR y el uso de las capas de difusión, ya introducen comportamientos no lineales, por lo que no es necesario la introducción de la función no-lineal “ $f$ ”.

- Estado de actividad de un canal: indica sólo si el canal ‘ $ij$ ’ se ha activado. Un canal se activa cuando hay un patrón de entrada que contiene su ID y  $P_{ij}^+ + P_{ij}^- > \pi$ . Con este valor  $\pi$  se controla el nivel de certeza que se quiere obtener en el resultado de la  $\mu$ Red. Es, por tanto un parámetro global para toda la  $\mu$ Red. Durante el aprendizaje puede tomar el valor 0.
- Señal de ECO: Toma los valores discretos 1,0,-1. Normalmente se correspondería con el signo de la señal de error entre la salida estimada generada por la  $\mu$ Red y el valor deseado para la misma, pero se le puede dar otros usos para modificar el comportamiento de la  $\mu$ Red durante el aprendizaje.

NOTA: Por motivos de implementabilidad, el valor analógico que entra al canal así como el ID del canal se agrupan bajo lo que llamaremos PIN. Cuando un canal se ha creado pero aún no se ha conectado a ninguna XCelda, se habla de “BOTÓN” (que contendrá un ID y un valor).

NOTA: Los contadores  $N_{ij}^+$  y  $N_{ij}^-$  sólo se incrementarán cuando el canal 'ij' esté activo y las XCeldas 'i' y 'j' también lo estén. Igualmente  $N_{ii}$  se incrementará sólo cuando la XCelda 'i' esté activa.

NOTA: Una XCelda que tenga sólo tenga BOTONES de salida será eliminada cuando la  $\mu$ Red esté "durmiendo" o cuando tenga, además de sólo BOTONES de salida, un valor de  $N_{ii} > 2$  (No hay que olvidar que  $N_{ii}$  inicialmente vale 1 y que si recibiese alguna señal de ECO por sus canales de salida,  $N_{ii}$  se incrementaría, por lo que tomaría el valor 2 y esto indicaría que alguno de los canales de la XCelda ha sido usado (de lo contrario, sería imposible haber recibido una señal de ECO).

- Una XCelda se activa cuando  $A_{ij} \geq \alpha_i$ , siendo  $\alpha_i$  un umbral que se calcula de forma dinámica. Para ello:
  - se construye el histograma de la suma de los valores de entrada de la XCelda<sub>i</sub> mapeados sobre un rango de números enteros positivos
  - obtenemos un centro de masa definido en base a los dos valores máximos del histograma ( $V_1$  y  $V_2$ ) y sus ocurrencias correspondientes ( $C_1$  y  $C_2$ ) de la siguiente forma:

$$\alpha_i = (C_1 V_1 + C_2 V_2) / (C_1 + C_2)$$

(inicialmente  $\alpha_i = 0$  para favorecer las difusiones)

- $\alpha_i$  puede disminuir su valor si la XCelda<sub>i</sub> se activó en la iteración anterior. Esto haría que la XCelda fuese más fácilmente excitable aún cuando los canales de entrada tuvieran valores más bajos en la siguiente iteración. Esto estaría justificado biológicamente en el hecho de que los patrones no desaparecen instantáneamente, sino que siguen siendo observables por el sistema durante un cierto tiempo (si acaso alterados con pequeñas variaciones de escala y/o posición). Además, de esta forma, teóricamente, se conseguiría que la  $\mu$ Red fuese más insensible frente a traslaciones y se implementaría una cierta capacidad de aprendizaje de secuencias temporales condicionadas. (De todos modos, esto habría que simularlo)
- Conviene pararse aquí y fijarse que la información  $A_{ij}$  "viaja" a través del canal hacia la siguiente XCelda donde volverá a ser procesada hasta llegar a la salida, en donde se calculará el error en base al valor recibido de los  $A_{ij}$  que se han ido ponderando y propagando a través de la  $\mu$ Red.
- Una XCelda implementa el "olvido dinámico" cuando haya casos de desbordamiento en los contadores, cuando así se le ordene explícitamente a la  $\mu$ Red, o cuando la  $\mu$ Red esté "durmiendo". (NOTA:

Habr  que ver si durante dicho proceso se permite o no que las XCeldas difundan, si se permite la propagaci n de la se al de ECO, o , en el caso de que se permita, si se permite la actualizaci n de los contadores cuando se produzca dicha se al de ECO. Lo que s  est  claro, es que durante este proceso pueden aparecer comportamientos inesperados, por lo que si hubiera actuadores conectados a las salidas de la  $\mu$ Red, habr a que desactivar dichas salidas para evitar acciones inesperadas de los actuadores). Tras un proceso de olvido din mico (o varios de ellos) algunas XCeldas podr n morir por inactividad dando lugar a lo que llamaremos "apoptosis".

Veamos c mo se implementar a el olvido din mico:

- Supongamos que un contador  $N_{ij}$  de una XCelda<sub>i</sub> llega a su valor m ximo permitido como registro de tipo "entero sin signo" (el tama o del registro no debe ser muy grande ya que una  $\mu$ Red aprende muy r pido).
  - Entonces, se miran los valores de todos los contadores (de excitaci n o de inhibici n) de los canales de salida (ej.:  $N_{ij}$ ,  $N_{ix}$ , ...) y se toma el menor valor  $> 1$  de ellos (llam mosle 'V').
  - Si  $V=0$  para los contadores de excitaci n y de inhibici n contador (por ejemplo,  $N_{ix}^+$  y  $N_{ix}^-$ ) entonces se elimina el canal 'ix' de las entradas de la XCelda<sub>x</sub> y de las salidas de la XCelda<sub>i</sub>.
  - A continuaci n dividimos a todos los contadores antes mencionados (excepto a los que se hayan eliminado) (i.e.:  $N_{ii}$ ,  $N_{ij}$ , ...) entre el valor V tomando como nuevo valor la parte entera del resultado de la divisi n. (As  se asegura que los valores peque os se vayan desvaneciendo).
  - Si tras esta operaci n, una XCelda queda sin canales de entrada o sin canales de salida, se elimina, as  como los canales que pudieran quedar sueltos.

## 2.2 Capas

Una capa est  formada por un conjunto de XCeldas. La cantidad de XCeldas dentro de una capa se establece de forma din mica, as  como la interconexi n entre ellas y con otras XCeldas de otras capas.

Todas las XCeldas de una capa empiezan el procesamiento de la informaci n a la vez cuando todas las XCeldas de la capa anterior han terminado de procesar su informaci n.

Se definen varios tipos de capas.

### 2.2.1 Capa de entrada

Puede ser de varios tipos:

- IMG (para tratamiento de im genes)(a n hay que mejorarla):

Lleva a cabo un preprocesamiento de las imágenes. La imagen inicial se divide en 4<sup>n</sup> regiones cuadradas. Cada región es asignada a la entrada de una XCelda-IMG especializada en calcular histogramas, detectar bordes y generar un patrón con él: un subcontorno.

Cada XCelda-IMG se ejecuta en un hilo a parte de forma tal que su ejecución se solapa con la fase de aprendizaje de la parte de la  $\mu$ Red formada por la capa-AND y la capa-OR.

Cada XCelda-IMG está conectada con sus 4 XCeldas-IMG colindantes (UP, DOWN, RIGHT, LEFT).

Las XCeldas-IMG intercambian ese subcontorno con aquellas XCeldas-IMG colindantes que tengan subcontornos con histogramas semejantes. Así una XCelda-IMG que tenga un subcontorno con un histograma asociado menos intenso que el de una XCelda-IMG contigua, se lo enviará a dicha XCelda-IMG. Al final todos los subcontornos que pertenezcan a un único contorno se concentrarán en una única XCelda-IMG. Así, cada XCelda-IMG que tenga un contorno lo volcará finalmente sobre el POOL DE CONTORNOS, el cual se encarga de:

- Convertir dichos contornos en otros invariantes a escalas, rotaciones y traslaciones: tendrán el mismo número de puntos y éstos estarán ordenados lexicográficamente. A estos contornos los llamaremos "contornos invariantes".
  - Cada coordenada de cada punto del contorno invariante se pasa a una XCelda-FUZZY (que generaría coordenadas invariantes tolerantes a variaciones gracias a que serían valores borrosos, que llamaremos "coordenadas invariantes borrosas/fuzzy" o CIFs)
  - Cada par de CIFs correspondientes a un punto, se asignarían a la entrada de una XCelda-AND (cuyo ID será las coordenadas invariantes borrosas de dicho punto), cuya salida (llamémosla SPCIF (salida asociada al punto con CIF) se pasaría a una capa-AND.
  - La capa-AND creará así, automáticamente, una XCelda-AND cuyas entradas serán las SPCIFs del paso anterior, y tendrá un ID dado por el conjunto CIFs correspondientes a las XCeldas-AND que generaron dichas SPCIFs. Así, gracias al funcionamiento de la capa-AND, un contorno podrá ser descompuesto posteriormente en contornos más pequeños que se podrán usar para construir contornos más complejos más adelante.
- Calcular los centros de dichos contornos.
- Establecer relaciones espaciales entre los centros de todos los contornos. Dichas relaciones espaciales son realmente vectores con el origen en el centro del contorno mayor y que "apuntan" a los centros del resto de contornos.



- Cada coordenada de cada vector se pasa a una XCelda-FUZZY (que generaría coordenadas invariantes tolerantes a variaciones gracias a que serían valores borrosos, que llamaremos "coordenadas invariantes borrosas/fuzzy" o CIF)
- Cada par de CIFs correspondientes a un punto, se asignarían a la entrada de una XCelda-AND (cuyo ID será las coordenadas invariantes borrosas de dicho punto), cuya salida (llamémosla SPCIF (salida asociada al punto con CIF) se pasaría a una capa-AND.
- La capa-AND creará así, automáticamente, una XCelda-AND cuyas entradas serán las SPCIFs del paso anterior, y tendrá un ID dado por el conjunto CIFs correspondientes a las XCeldas-AND que generaron dichas SPCIFs. Así, el conjunto de vectores que definen una imagen en base a la posición relativa entre sus contornos, quedarán representadas bajo una XCelda-AND.

Como se aprecia, el funcionamiento de esta capa está inspirada en las CAPSULE NEURAL NETWORKs.

#### ■ FUZZY (para tratamiento de señales analógicas, o valores reales):

Convierte el conjunto de entradas en otras más genéricas de tipo borroso. Para ello, hay una XCelda-FUZZYMaster<sub>i</sub> por cada entrada que analiza el valor de dicha entrada 'i' y lo descompone en regiones cuyo número y tamaño irá cambiando durante el proceso de aprendizaje.

Para ello se aplicará el siguiente algoritmo:

- Para una entrada dada 'i':
  1. Mapear los valores de entrada en un rango  $\rho$  dado discreto de números enteros (típicamente menor. Ej.: 0, 1, 2, ..., R)
  2. Contar las veces 'Nm<sub>i</sub>' que un valor de entrada mapeado 'm<sub>i</sub>' aparece.
  3. Obtener el máximo M de todos los 'Nm<sub>i</sub>'
  4. Estos valores se usarán posteriormente para decidir la forma más correcta de "trocear" y definir las nuevas funciones de pertenencia de las XCeldas-FUZZY <sub>$\mu_{mi}$</sub>

Así pues una XCelda-FUZZYMaster<sub>i</sub> es una XCelda que analiza una entrada 'i' y genera un conjunto de XCeldas-FUZZY <sub>$\mu_{mi}$</sub>  conectadas a dicha entrada 'i'. En concreto:

- Creará una XCelda-FUZZY <sub>$\mu_{mi}$</sub>  con una función de pertenencia triangular  $\mu_{mi}$  (que tendrá un valor máximo para el valor de la entrada mapeado 'm<sub>i</sub>' y una base con una anchura igual a 'R') con un ID dado por ID<sub>i</sub>=m<sub>i</sub>, donde ID<sub>i</sub> es el ID de la entrada 'i'.

Esto se hace con cada entrada 'i' y este ID<sub>i</sub> se pasa al difusor de patrones DP-I a partir del cual se inicia todo el proceso de construcción de las estructuras AND y OR necesarias para codificar el patrón.

- En la capa-FUZZY se calcula un coeficiente ' $\sigma_{InP}$ ' que nos indica el grado de similitud de un patrón de entrada InP. Para ello, se toma el valor ' $v_i$ ' de cada entrada 'i' y se obtiene la salida  $\mu_{mi}(v_i)$  de su XCelda-FUZZY $\mu_{mi}$  correspondiente, se suman todos estos valores para todas las 'N' entradas 'i' y se calcula su media:

$$\sigma_{InP} = (1/N) \sum_{i=1 \dots N} \mu_{mi}(v_i)$$

- Definimos  $\chi$  como un umbral que indica el nivel de precisión para las clasificaciones (clustering)

Si  $\sigma_{InP} < \chi$  se considera que el patrón InP no encaja bien con los patrones previos, por lo que la capa-FUZZY envía una señal "NUEVO PATRÓN" a la capa-OR de su grupo para que se genere otro ID para otra nueva clase con su correspondiente vector one-hot que será usado en la capa-OR como target de salida. Esto desencadenará todo el proceso de construcción de relaciones AND y OR dentro de la  $\mu Red$ .

La señal "NUEVO PATRÓN" puede usarse para interactuar con el usuario y preguntarle si realmente el patrón no encaja bien con el valor esperado y realmente constituye un nuevo patrón o es que el target de salida deseado de entrenamiento es erróneo.

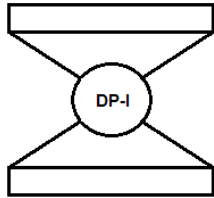
Gracias a la señal "NUEVO PATRÓN" se implementa el aprendizaje no supervisado de la  $\mu Red$ , y gracias a esto mismo, se podrían ir añadiendo grupos de capas mínimos sucesivos de forma automática (en base, por ejemplo, a un criterio de número de salidas OR máximo)

Si al propagarse el patrón InP hasta la capa OR de salida el patrón generado por la  $\mu Red$  no coincide con el target de salida deseado para la capa OR y  $\mu_{mi}(v_i) < \chi$ , entonces se envía la señal de ECO hacia atrás correspondiente.

Cuando la señal de ECO llega a la XCelda-FUZZYMaster<sub>i</sub>:

- se crea una nueva XCelda-Fuzzy $\mu_{mi}'$  y se actualiza la anchura de la XCelda-Fuzzy $\mu_{mi}$  antigua
- se lanza un nuevo ciclo de aprendizaje para reforzar el aprendizaje de este nuevo patrón.

### 2.2.2 Difusor de patrones de tipo I (DP-I)



Recoge todos los  $ID_i$  procedentes de las Xceldas-FUZZY activas, los agrupa ordenándolos lexicográficamente (formando lo que llamaremos un patrón de entrada "InP" (*Input Pattern*)), y los reparte entre los canales de salida. Sabiendo que:

- Hay un canal de salida por cada  $ID_i$
- Si un canal de salida no existe, se crea y se le asigna un  $ID = ID_i$  (aquí, realmente, se estaría creando un BOTÓN),
- Si el patrón InP no contiene ningún  $ID_i$  que coincida con el ID del canal de salida, entonces por dicho canal no se transmite nada.
- Si el patrón InP contiene el ID de un canal de salida, entonces se transmite dicho InP por dicho canal.
- Al llegar a una  $XCelda_x$  todos los InP a través de sus canales, se eliminan allí, en todos los InP recibidos, todos los IDs que aparezcan también en el ID de la  $XCelda$  (esto dará lugar a lo que llamaremos los "InP reducidos" o InPR). Después de esto pueden darse varias situaciones:

- Han llegado patrones InP por todos los canales: De nuevo caben dos posibilidades:
  - No queda ningún ID: Entonces la  $XCelda_x$  se activa si los  $P_{ix}$  lo permiten.
  - Sobran uno o más IDs: Se propaga un nuevo patrón con esos IDs sobrantes por aquellos canales de salida de la  $XCelda$  que contengan alguno de esos IDs sobrantes. Si ningún canal de salida contiene alguno de esos IDs sobrantes entonces:
    - se crea un canal de salida en el difusor de patrones con un  $ID = IDs$  sobrantes
    - la  $XCelda_x$ 
      - crea otra nueva  $XCelda_j$  con:
        - $L_j = L_i + 1$
        - $ID_j = (ID \text{ de la } XCelda_x \mid IDs \text{ sobrantes})$ .
        - Un canal de entrada conectado al canal de salida de la  $XCelda_x$ , con un  $ID = ID$  de la  $XCelda_x$
        - Un canal de entrada conectado al nuevo canal de salida creado en el difusor de patrones con un  $ID = IDs$  sobrantes
      - crea un nuevo canal de salida con  $ID = ID$  de la  $XCelda_x$  conectado en su extremo con la  $XCelda_j$
- Han llegado patrones InP sólo por algunos canales:
  - La  $XCelda$  no se activa y no se propaga absolutamente nada por su salida.
  - Se comprueba si se puede llevar a cabo la atomización de la  $XCelda_x$  (necesaria para reutilizar funciones, estructuras, patrones aprendidos, etc.). Para ello:

- se miran los InPR de aquellos canales por los que han llegado patrones InP. Se eligen aquellos InPR que sean iguales y de menor longitud y anotamos los IDs de los canales (llamémosles  $C_n$ ) por los que se han recibido. La concatenación de los IDs de dichos canales formará otro ID que corresponderá al  $ID_n$  de la nueva  $XCelda-AND_n$  que se creará.
- La nueva  $XCelda-AND_n$  tendrá:
  - $L_n = L_x - 1$
  - como canales de entrada, los canales  $C_n$  de la  $XCelda_x$  original,
  - un canal de salida con  $ID = ID_n$  que se conecta a la entrada de la  $XCelda_x$ ,
  - otro canal de salida hacia las salidas de la capa-AND

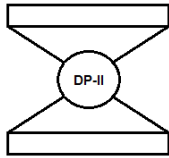
### 2.2.3 Capa AND

Tiene un conjunto de entradas (a las que se conectan, típicamente, las salidas de las  $XCeldas-FUZZY$  de la capa anterior, pero además, podrían tenerse más entradas procedentes de otras  $XCeldas$  de otras estructuras de otras  $\mu$ Redes).

- En esta capa se hace una “foto” de las entradas y aquellas que estén activas ‘A’ se agrupan conectándose a las entradas de una  $XCelda-AND$  cuyo ID se construirá concatenando con orden lexicográfico los IDs de las  $XCeldas$  que activaron dichas entradas ‘A’.
- En esta capa se lleva a cabo el proceso de atomización de las  $XCeldas-AND$  en otras más pequeñas para poder descomponer los “conceptos” en otros más pequeños reutilizables para la construcción de otros nuevos en el futuro. Esto dará lugar a una estructura en árbol de  $XCeldas-AND$ .
- En el proceso de atomización se define el parámetro “W” (anchura de atomización permitida) que indica el número de entradas mínimo permitido que pueden tener las nuevas  $XCeldas-AND$  creadas después de una atomización. Por defecto  $W > 1$  para impedir que tras una atomización se creen  $XCeldas-AND$  para cada entrada de una  $XCelda-AND$  original. Eso ayuda también a hacer al sistema más robusto y evitar el overfitting.
- Con el tiempo, gracias al “olvido dinámico” aquellas  $XCeldas-AND$  que no se usen irán muriendo, lo cual irá dejando también inactivas a las otras  $XCeldas-AND$  a las que estén conectadas, por lo que éstas también terminarán muriendo.
- El proceso de atomización se realiza durante la FASE DE DIFUSIÓN (propagación hacia delante), por lo que no requiere la señal de ECO para su implementación.

- Calcula el valor de su señal de información analógica según la operación borrosa AND:  $A_{ij} = \min(A_{hi})$

#### 2.2.4 Difusor de patrones de tipo II (DP-II)



Recoge los IDs de los canales de salida activos de las Xceldas-AND activas de la capa-AND de tal forma que evita que se produzcan falsas relaciones OR cuando se activan canales semejantes procedentes de XCeldas-AND.

- Tiene el mismo número de canales de entrada que de salida
- Los IDs de los canales de entrada y salida son los mismos
- Los canales de entrada están conectados a los canales de salida de las XCeldas-AND que no están conectados a ninguna otra XCelda-AND
- Cada vez que una XCelda-AND transmite su ID por un canal de entrada del DP-II, se recoge en el DP-II dicho ID y si el ID:
  - no está contenido en la lista, se añade,
  - sí está contenido en alguno de los IDs de la lista (ej.: A está contenido en A, o en AB, o en ABC...), no se añade a la lista,
  - es lexicográficamente mayor que alguno de los IDs de la lista y contiene a alguno de ellos, éstos son eliminados de la lista y el nuevo valor es añadido. Ej: ID nuevo= ABC, IDs en la lista = A, AB, C, DE, entonces la lista quedaría: IDs en la lista = ABC, DE
- Cuando el DP-II ha recibido todos los IDs de la Capa-AND, activa sólo los canales de salida cuyo ID aparezca en la lista.

#### 2.2.6 Capa de salida (OR):

- Si lo que se busca es hacer un clasificador supervisado, esta capa implementará una XCelda-OR por cada salida activa que tenga la capa durante el aprendizaje.
- Si lo que se busca es hacer una regresión, esta capa implementará una XCelda-SUMA por cada salida del tipo regresión que tenga la capa.
- Se pueden conectar varias capas-OR a la salida de una misma capa-AND o de una capa-NAND, y cada capa-OR podría realizar tareas diferentes. Por ejemplo, una podría encargarse de hacer una regresión y la otra una clasificación partiendo de los mismos datos aprendidos en la capa-AND.
- En esta capa se calcula el error según la operación borrosa OR:

$$E_{ij} = \text{OUT}_j - \max(C_{hi}), \text{ donde } 0 \leq \text{OUT}_j \leq 1$$

- Una señal de ECO se define como el signo del error (considerando el 0 como un valor positivo)

- Cada XCelda-OR o XCelda-SUMA genera una señal de ECO hacia atrás que se propaga SÓLO por las XCeldas activas y por los canales que tuvieron el canal activo durante la FASE DE DIFUSION. Esto provoca un incremento del contador  $N_{ij}$  de las XCeldas 'i' que tenían todos sus canales de entrada activos y algún canal de salida activo, y un incremento de aquellos contadores  $N_{ij}$  según el siguiente criterio:
  - si cuando el canal 'ij' estaba activo transmitió una señal  $A_{ij}$  positiva:
    - si el ECO es positivo, se incrementará  $N_{ij}^+$
    - si el ECO es negativo, se incrementará  $N_{ij}^-$
  - si cuando el canal 'ij' estaba activo transmitió una señal  $A_{ij}$  negativa:
    - si el ECO es positivo, se incrementará  $N_{ij}^-$
    - si el ECO es negativo, se incrementará  $N_{ij}^+$

Es decir, el signo del contador  $N_{ij}$  que hay que incrementar viene dado por:  $\text{sign}(A_{ij}(k-1)) \cdot \text{sign}(\text{ECO}(k))$

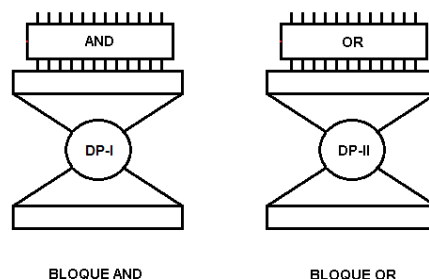
- Con la señal de ECO se ajustan las funciones de pertenencia de las XCeldas-FUZZYMaster y de sus XCeldas-FUZZY correspondientes
- El inicio de la transmisión hacia atrás de las señales de ECO marca el inicio de la FASE DE CORRECCIÓN.
- Cuando la(s) señal(e)s de ECO llegan a la capa de entrada se da por terminada la FASE DE CORRECCIÓN, lo cual provoca un nuevo muestreo de las señales de entrada de la capa de entrada, después de lo cual comenzará una nueva FASE DE DIFUSIÓN.

NOTA: No es realmente necesario implementar la señal de eco

## 2.3 Bloques

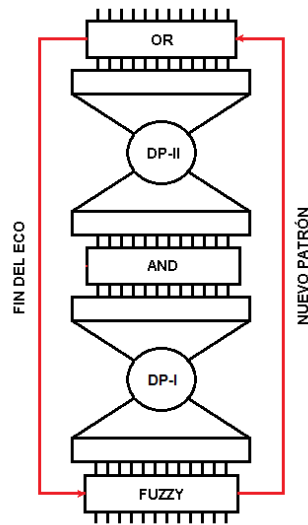
Las capas se suelen agrupar en bloques que se repiten. Así, se pueden clasificar 2 tipos de bloques:

- Bloque AND: compuesto por una capa DP-I y una capa-AND
- Bloque OR: compuesto por una capa DP-II y una capa-OR. Se usa para generar salidas logísticas o de regresión para un aprendizaje supervisado.



## 2.4 Grupos

Un grupo es un conjunto de bloques que comienza con una capa-FUZZY y acaba con un bloque OR. Un grupo típico sería uno constituido por una capa-FUZZY, un bloque AND, y un bloque OR:



## 2.5 Columna

Está formado por una sucesión en cascada de grupos.

## 3. Similitudes biológicas

- Las XCeldas se inspiran en las neuronas, y los canales en los axones y en las dendritas (más concretamente, en las sinapsis).
- Las XCeldas-Fuzzy se inspiran en las neuronas aferentes pseudounipolares.
- El difusor de patrones se inspira en las interneuronas.
- Los IDs se inspiran en el etiquetado de sinapsis.
- La señal de ECO se inspira en las señales de retropropagación en sinapsis tipo *gap*.
- Las columnas se inspiran en las columnas del neocórtex.
- La eliminación de conexiones y XCeldas inútiles se inspira en la apoptosis neuronal.

#### **4. Referencias**

<https://es.wikipedia.org/wiki/Neurona>

<https://en.wikipedia.org/wiki/Apoptosis>

<https://en.wikipedia.org/wiki/Synapse>

[https://en.wikipedia.org/wiki/Neural\\_backpropagation](https://en.wikipedia.org/wiki/Neural_backpropagation)

[https://en.wikipedia.org/wiki/Long-term\\_potentiation](https://en.wikipedia.org/wiki/Long-term_potentiation)

[https://en.wikipedia.org/wiki/Synaptic\\_tagging](https://en.wikipedia.org/wiki/Synaptic_tagging)

[https://en.wikipedia.org/wiki/Gap\\_junction](https://en.wikipedia.org/wiki/Gap_junction)

<https://en.wikipedia.org/wiki/Neocortex>

<https://en.wikipedia.org/wiki/Cerebellum>

[https://en.wikipedia.org/wiki/Cerebellar\\_granule\\_cell](https://en.wikipedia.org/wiki/Cerebellar_granule_cell)