

# Reconocimiento de patrones visuales

## Practica 3

David Morales Sáez

Alberto Manuel Mireles Suárez

## ÍNDICE

Planteamiento y descripción del problema a resolver.....	3
Estudio teórico.....	3
Desarrollo de la práctica.....	4
Conclusiones.....	21
Bibliografía.....	22
Anexo.....	23

- Planteamiento y descripción del problema a resolver

En esta práctica hemos tenido que implementar redes con tecnología Backpropagation con el fin de estudiarla, usando como herramienta para ello el simulador JNNS. Para ello, se han resuelto dos problemas, la resolución del XOR y el reconocimiento de caracteres alfanuméricos.

- Estudio Teórico

La Backpropagation es un tipo de red de aprendizaje supervisado que emplea un ciclo de propagación – adaptación de dos fases. Una vez se aplica un patrón a la entrada de la red como estímulo, este se propaga desde la primera capa a través de las capas superiores de la red, hasta generar una salida. La señal de salida es comparada entonces con la salida deseada y se calcula el error para cada una de las salidas.

Las salidas de error son propagadas hacia atrás, partiendo de la capa salida, hacia todas las neuronas de la capa oculta que contribuyen directamente a la salida. Sin embargo, las neuronas de la capa oculta solo reciben una fracción del error total, en base a la contribución de cada neurona a la salida. Este proceso es repetido, capa por capa, hasta que todas las neuronas de la red hayan recibido un error que indique su contribución al error total. Basándose en el error percibido, se actualizan los pesos de conexión de cada neurona, para hacer que la red converja a un estado que permita clasificar los patrones de entrenamiento.

Este proceso es importante ya que a medida que se entrena la red, las neuronas de las capas intermedias se organizan a sí mismas de tal modo que las distintas neuronas aprendan a reconocer las características del espacio total de la entrada. Una vez se completa el entrenamiento, al presentarle a la red un patrón arbitrario de entrada que contenga ruido, las neuronas de la capa oculta de la red responderán con una salida activa si la nueva entrada contiene un patrón que se asemeje a aquellas características que las neuronas individuales hayan aprendido a reconocer durante su entrenamiento. Y, a su vez, las unidades de las capas ocultas tienen tendencia a inhibir su salida si el patrón de entrada no contiene la característica para reconocer, para la cual han sido entrenadas.

El algoritmo de aprendizaje comienza alimentando los valores de la entrada de acuerdo a las siguientes ecuaciones:

$$net_i = \sum_{j \in A} O_j W_{ji} ; \forall i : i \in B,$$

donde A es el grupo de neuronas en una capa, y B es la otra.  $O_j$  es la activación para la neurona j, y  $W_{ij}$  son los pesos asignados a la conexión entre las neuronas j e i. En la ecuación anterior, se toman los valores de salida y se alimentan a la siguiente capa a través de los pesos. Esta operación se realiza para cada neurona en la siguiente capa, produciendo un valor de red para él. Este valor es la suma de todos los valores de activación en las neuronas de la capa anterior, y cada valor de red es aplicado ahora a la siguiente ecuación, conocida como función de activación, para producir la activación de esa neurona:

$$O_i = f(net_i) = \frac{1}{1 + e^{-net_i}}, \text{ esta es la función Sigmoide.}$$

Después de que todas las neuronas tienen un valor de activación asociado a un patrón de valores de entrada, el algoritmo sigue buscando errores en cada neurona que no es entrada. Los errores encontrados para las neuronas de salida, son propagados hacia atrás, a la capa anterior para que puedan ser asignados a neuronas de las capas ocultas, esto se calcula por:

$$\delta_i = f'(net_i) \sum_{j \in E} \delta_j W_{ij} ; \forall i : i \in D ,$$

donde D es el grupo de neuronas en una capa que no es de entrada y E es el grupo de neuronas de la siguiente capa. Este cálculo se repite para cada capa oculta de la red.

Después de que se ha encontrado la activación y el error asociado a cada grupo de neuronas, los pesos se actualizan, primero encontrando el valor que cada peso tiene que modificarse:

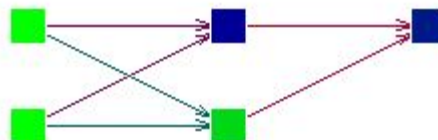
$$\Delta W_{ij} = CO_i \delta_j ; \forall i, j : i \in A, j \in B ,$$

donde C, conocida como el ratio de aprendizaje, es una constante que controla el valor del cambio de los pesos y  $W_{ij}$  es el cambio de los pesos entre la neurona i y j. El peso se modifica evaluando:

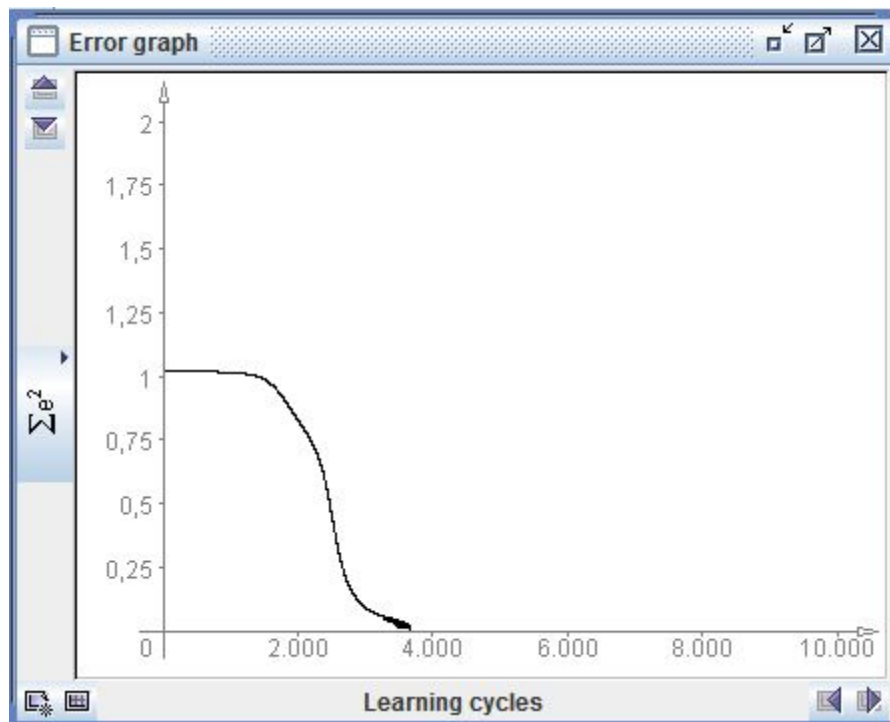
$$W_{ij}^{nuevo} = W_{ij}^{viejo} + \Delta W_{ij}$$

- Desarrollo de la práctica

Tras haber comprendido cuáles son los distintos puntos a realizar en la práctica, hemos resuelto las diferentes redes en el entorno de simulación JNNS. La práctica se divide en dos apartados, primero la realización del XOR empleando Backpropagation. Para ello hemos creado una red con dos unidades en la capa de entrada, dos unidades en la capa oculta y una en la capa de salida:

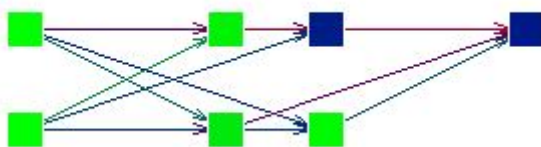


Tras haber creado el archivo de patrones, en el cual se establecen los patrones para el correcto aprendizaje de la red, podemos modificar los parámetros del aprendizaje para lograr una convergencia en un menor número de ciclos. Empleando los parámetros por defecto, obtenemos la siguiente gráfica de error:

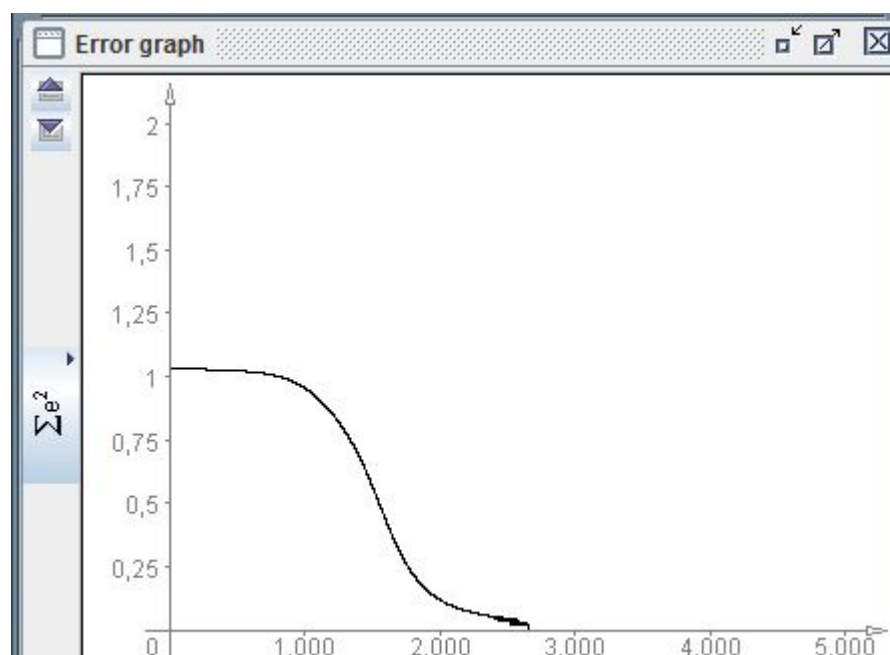


Como observamos, la red requiere aproximadamente 3700 ciclos para aprender.

Si duplicamos el número de elementos de proceso de la capa oculta:

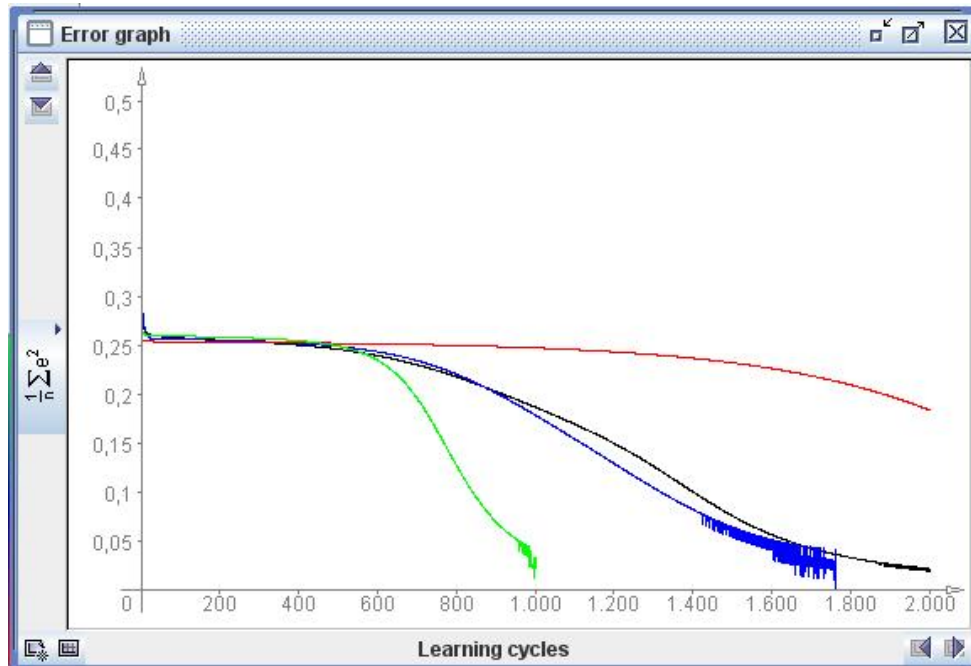


Realizando el aprendizaje con los parámetros por defecto y empleando el mismo archivo de patrones, obtenemos el siguiente error:

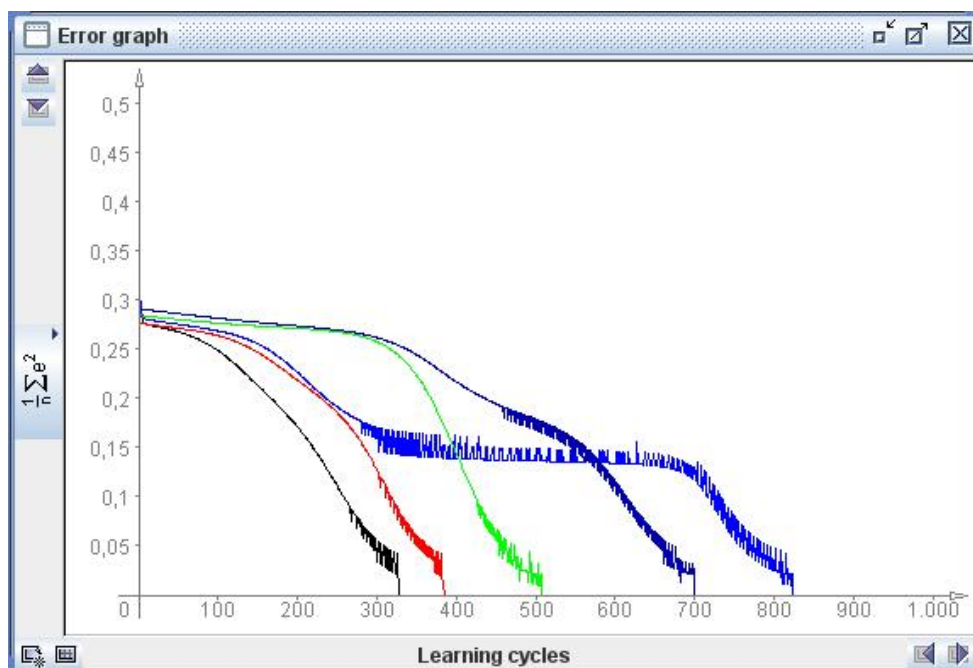


Como se puede observar, al incrementar el número de elementos de proceso, la red converge en un menor número de ciclos que en el ejemplo anterior.

A su vez, podemos modificar la velocidad con la que aprende la red mediante la modificación del ratio de aprendizaje, a continuación se muestra una gráfica de error empleando diferentes ratios (se realizará un estudio más detallado del ratio de aprendizaje en el siguiente apartado de la práctica).



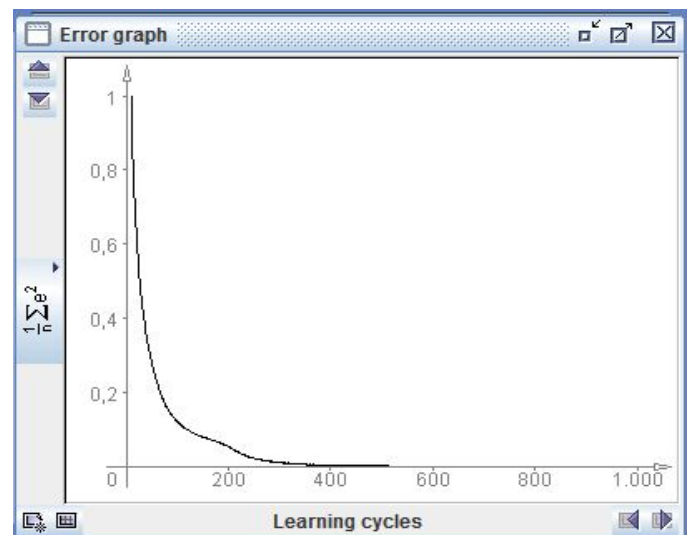
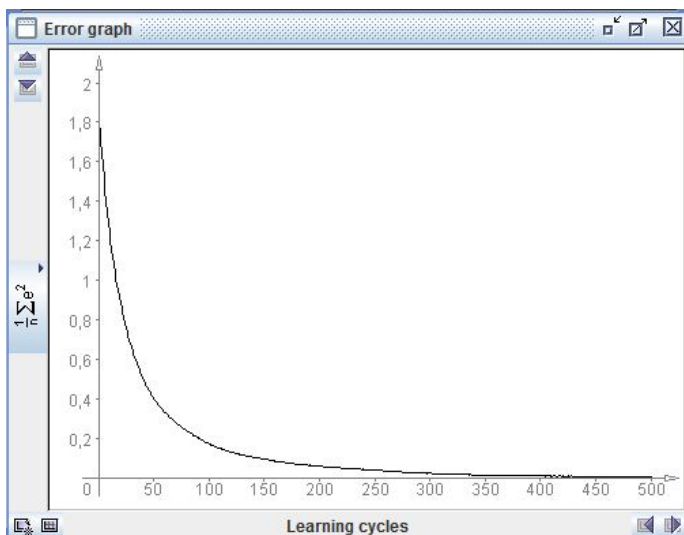
También, al realizar la práctica, hemos podido comprobar como afecta la inicialización de los pesos al aprendizaje de la red. En la siguiente gráfica se representan los errores de la red con los mismos parámetros con diferentes inicializaciones de pesos:



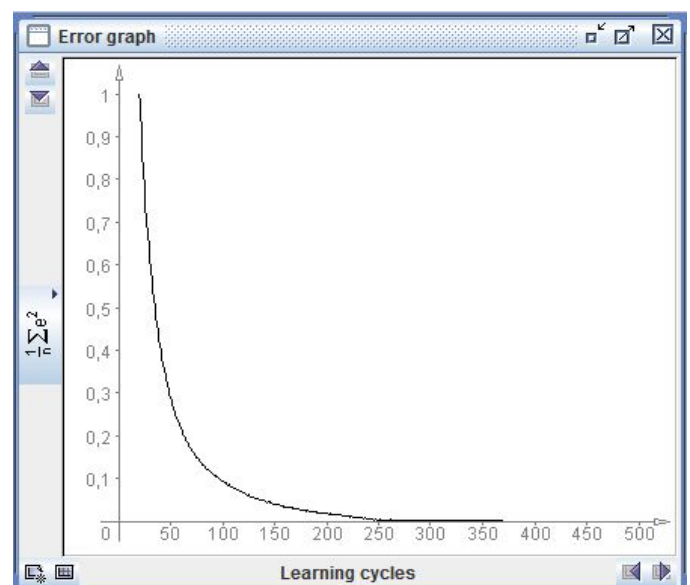
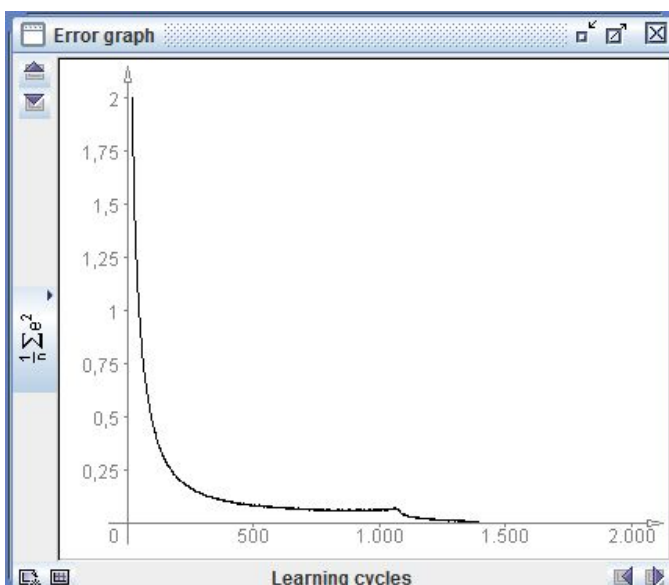
En la segunda parte de la práctica debemos realizar una red que sea capaz de reconocer caracteres alfanuméricos simulados mediante un conjunto de neuronas de entrada. Para la capa de entrada, hemos empleado una representación matricial de 7x5 neuronas. Para la salida, hemos empleado tres representaciones diferentes, una representación igual a la entrada, un conjunto de 8 neuronas que codifican la entrada en código binario y un conjunto de 36 neuronas de manera que cada una representa un carácter.

Para las tres representaciones, hemos empleado una capa oculta con 10 unidades de proceso. Aunque para observar el proceso de aprendizaje con un número diferente de unidades en la capa oculta, también hemos realizado pruebas empleando el doble. A continuación se muestran las gráficas de error para cada representación para comparar la diferencia del número de elementos en la capa oculta:

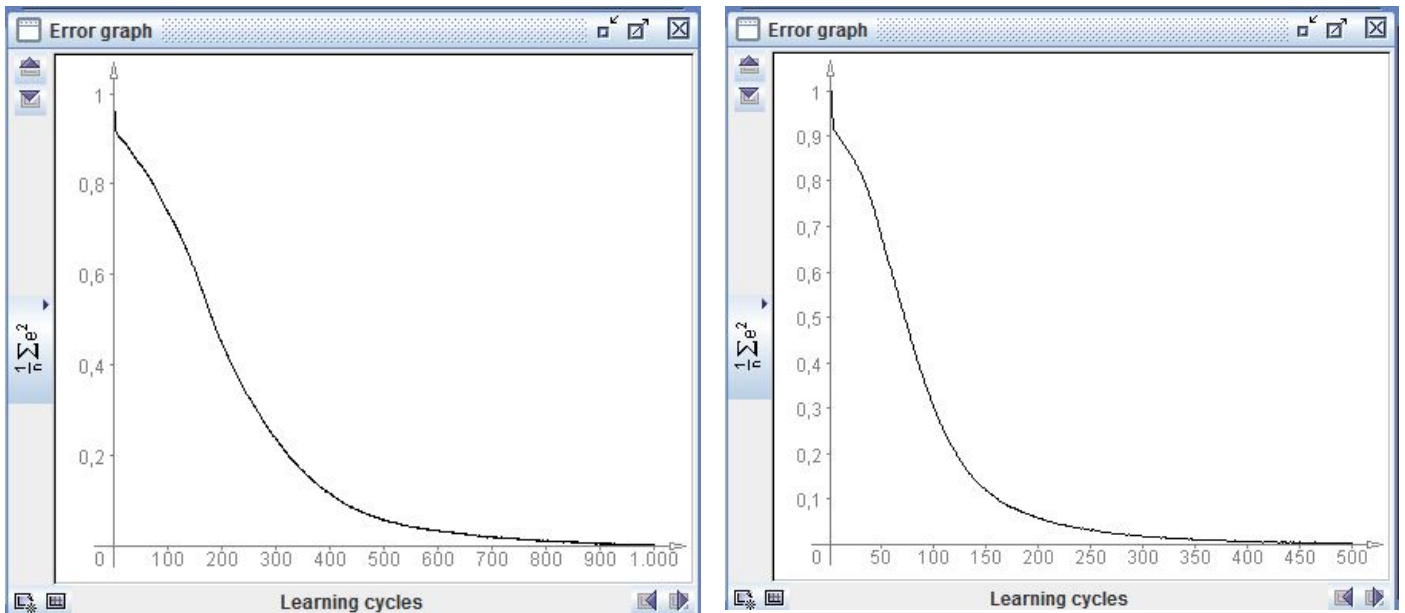
Representación binaria:



Representación matricial:

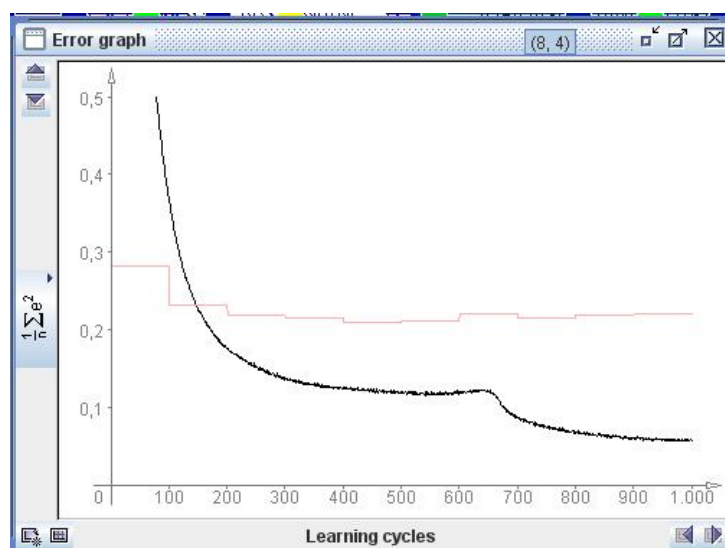


Representación por posición:



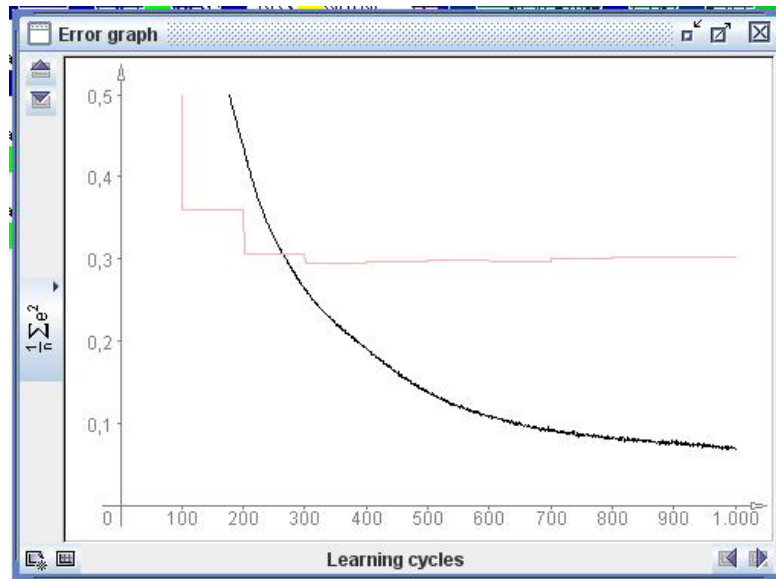
Para que las pruebas de la red sean lo más realistas posibles, hemos añadido un pequeño ruido a los patrones, creando de esta manera los patrones de validación, para comprobar si la red es capaz de generalizar y reconocer caracteres con pequeñas diferencias con respecto al que se ha empleado para el entrenamiento. A su vez, hemos probado con las tres representaciones a modificar los parámetros del aprendizaje y el margen de error, para poder observar las diferencias.

Representación binaria:

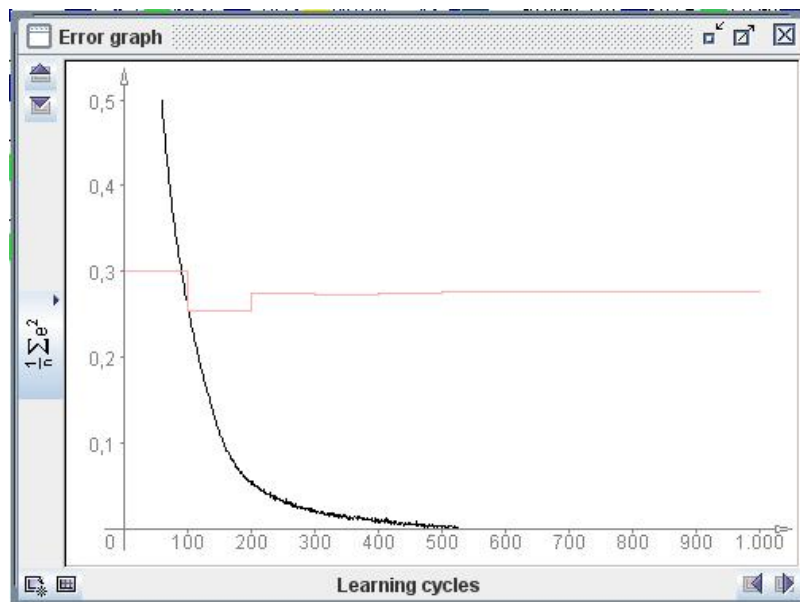


Ratio = 0,2 / dmax= 0,1

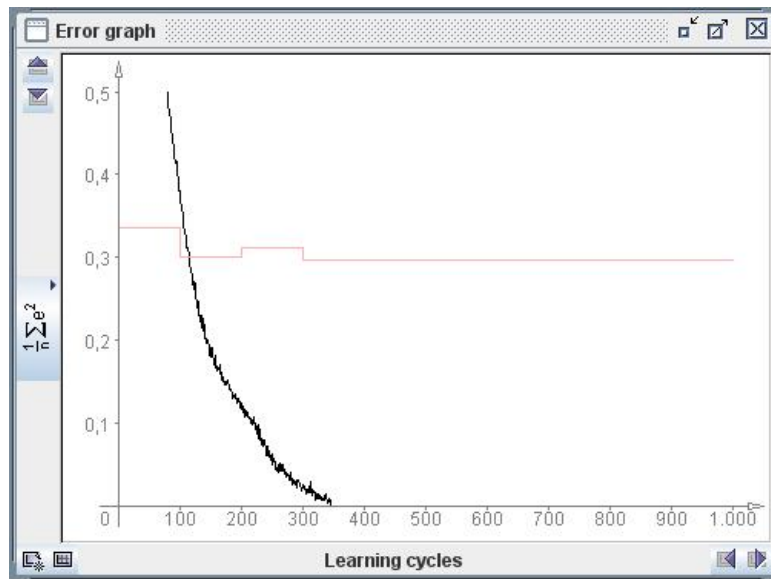




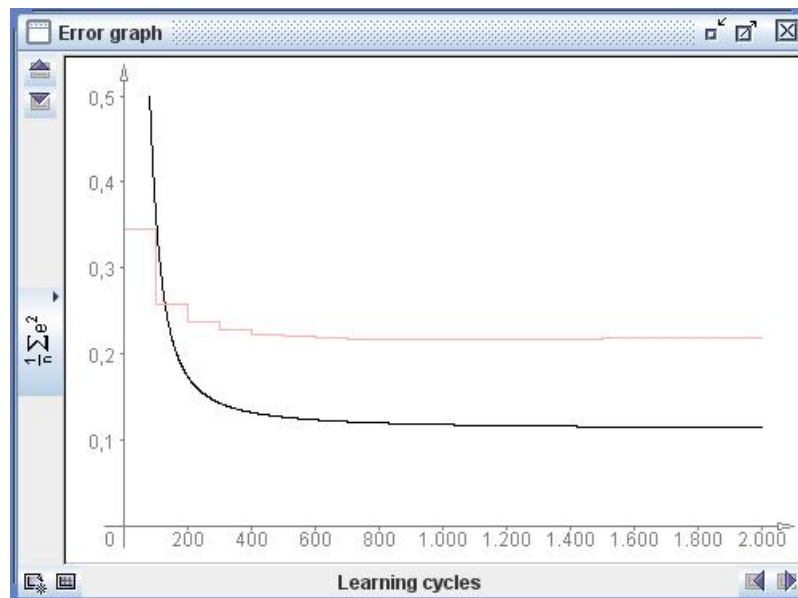
Ratio = 0,1 / dmax = 0,1



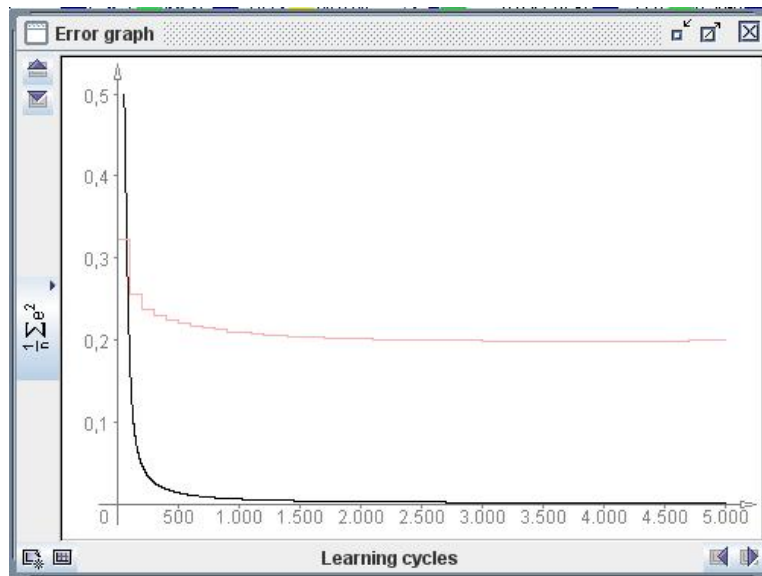
Ratio = 0,3 / dmax = 0,1



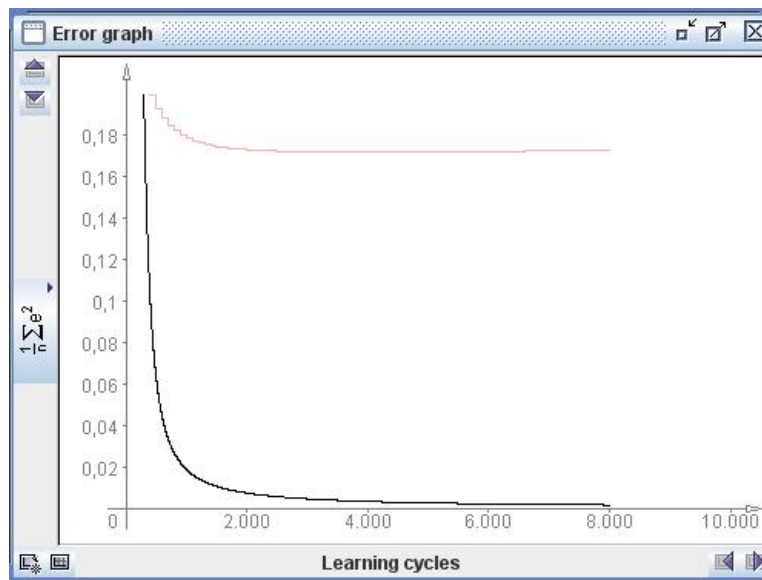
Ratio = 0,2 / dmax= 0,2



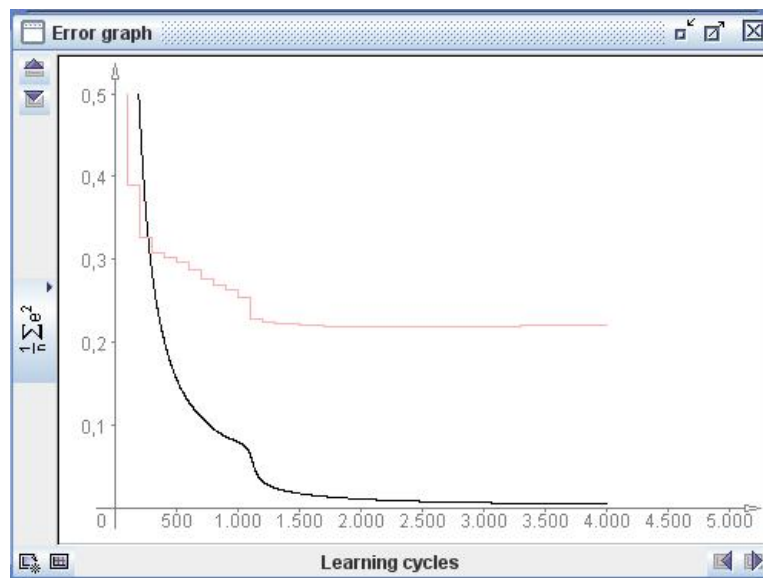
Ratio = 0,2 / dmax= 0



Ratio = 0,3 / dmax= 0

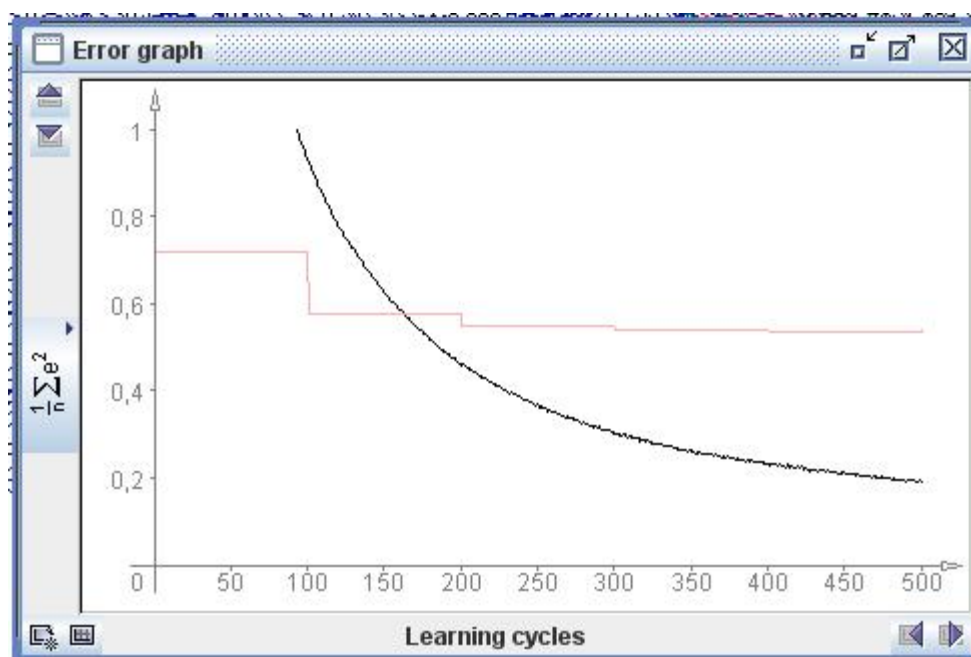


Ratio = 0,1 / dmax= 0

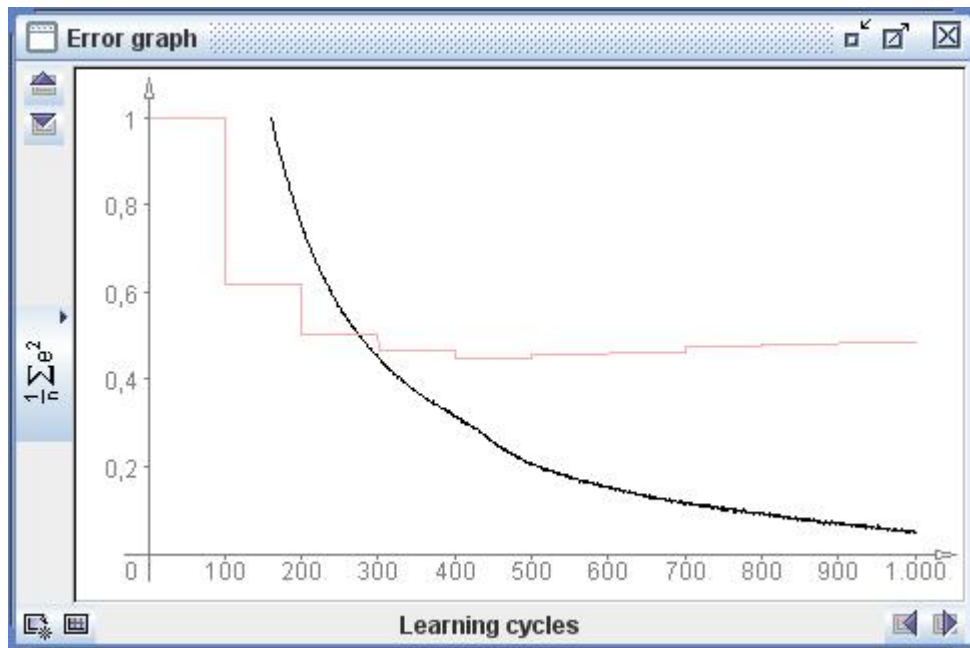


Ratio = 0,1 / dmax = 0,2

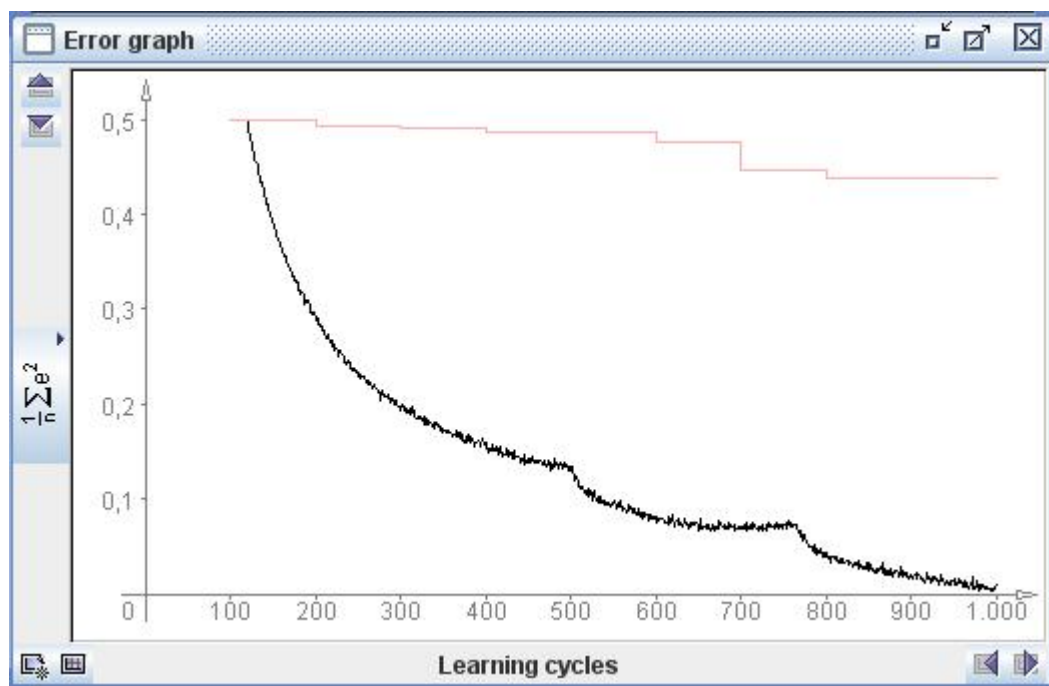
Representación matricial:



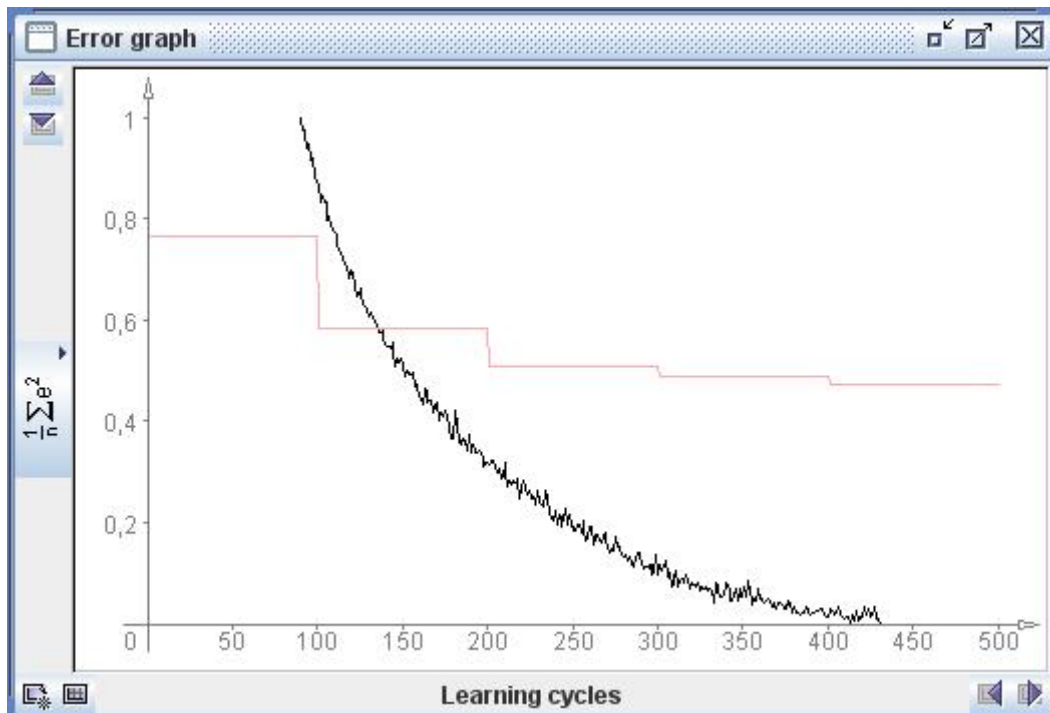
Ratio = 0,2 / dmax = 0,1



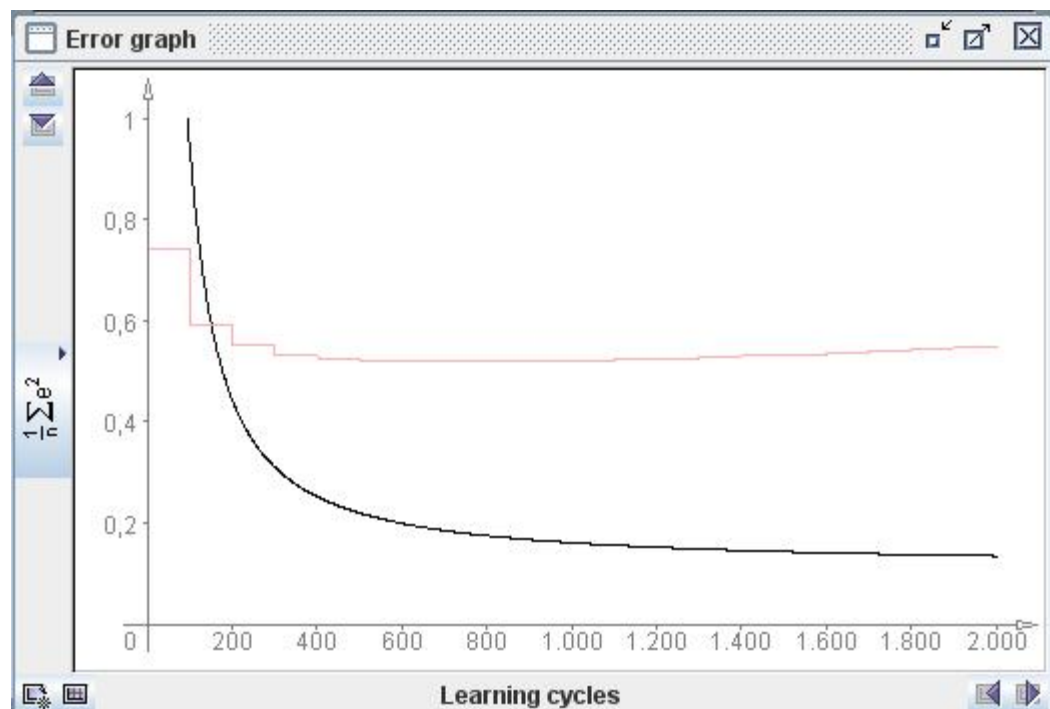
Ratio = 0,1 / dmax= 0,1



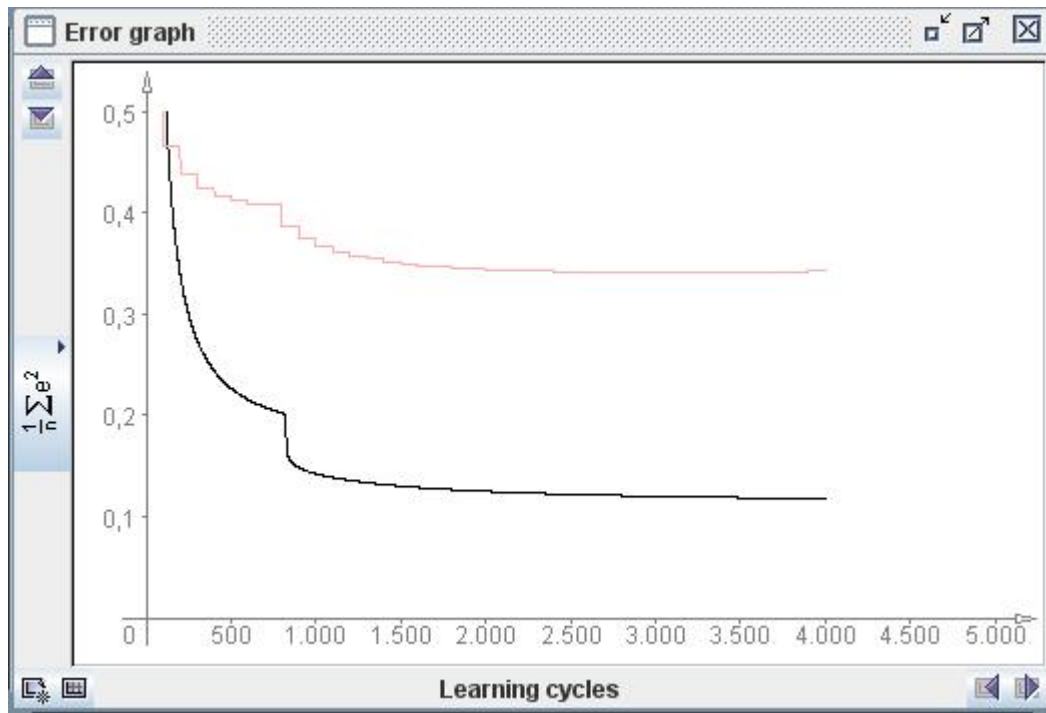
Ratio = 0,3 / dmax= 0,1



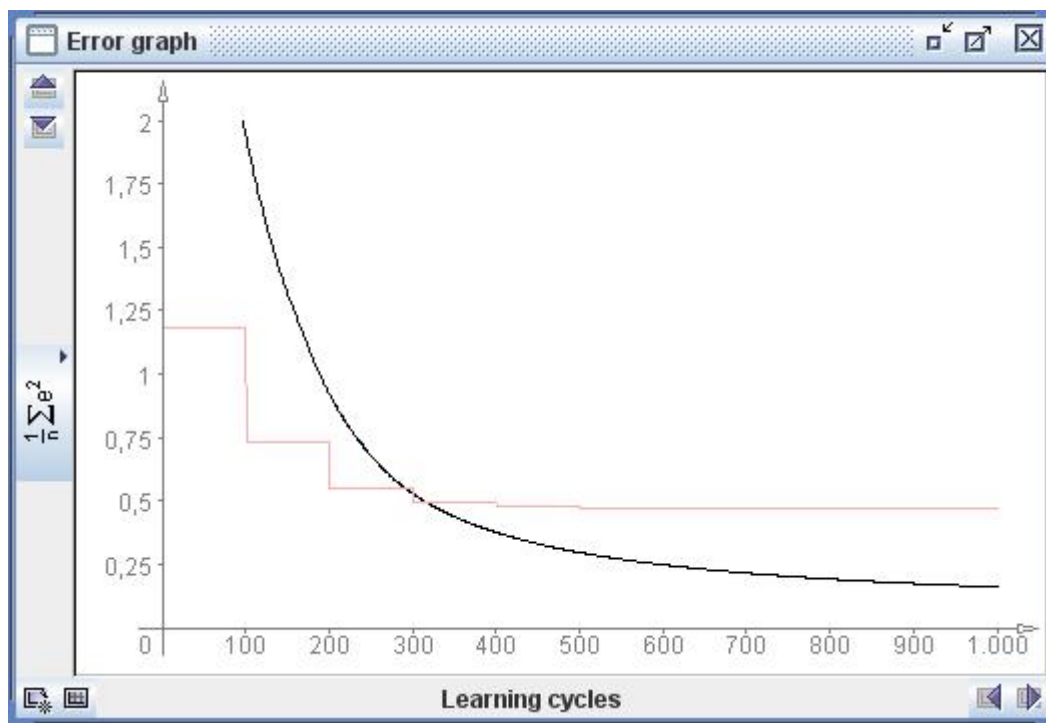
Ratio = 0,2 / dmax= 0,2



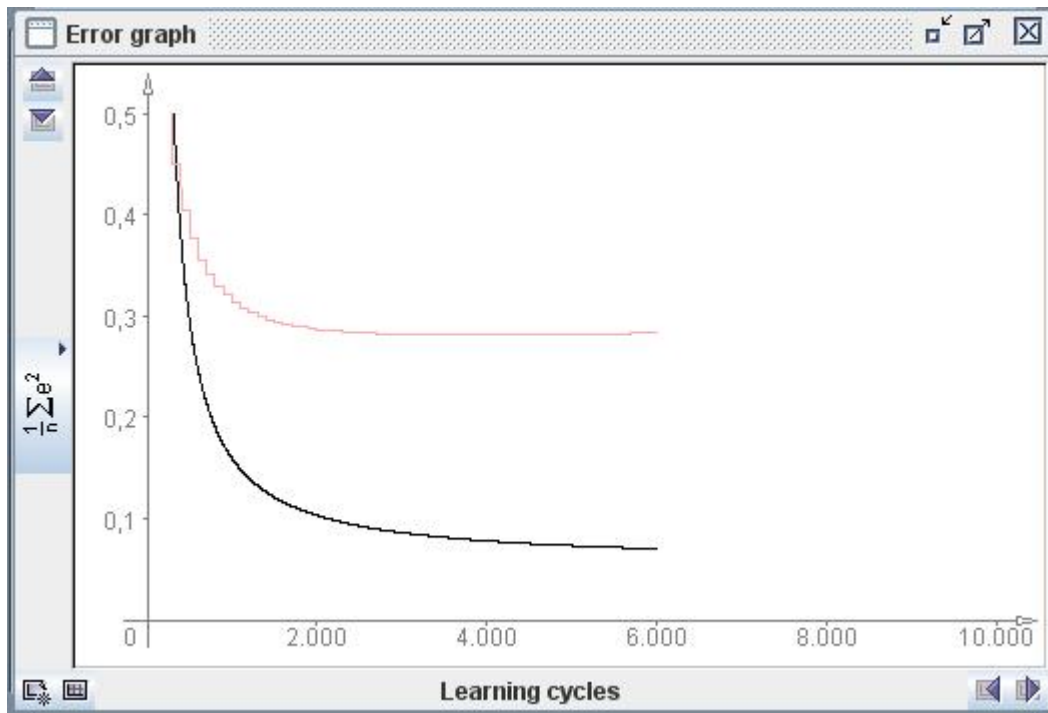
Ratio = 0,2 / dmax= 0



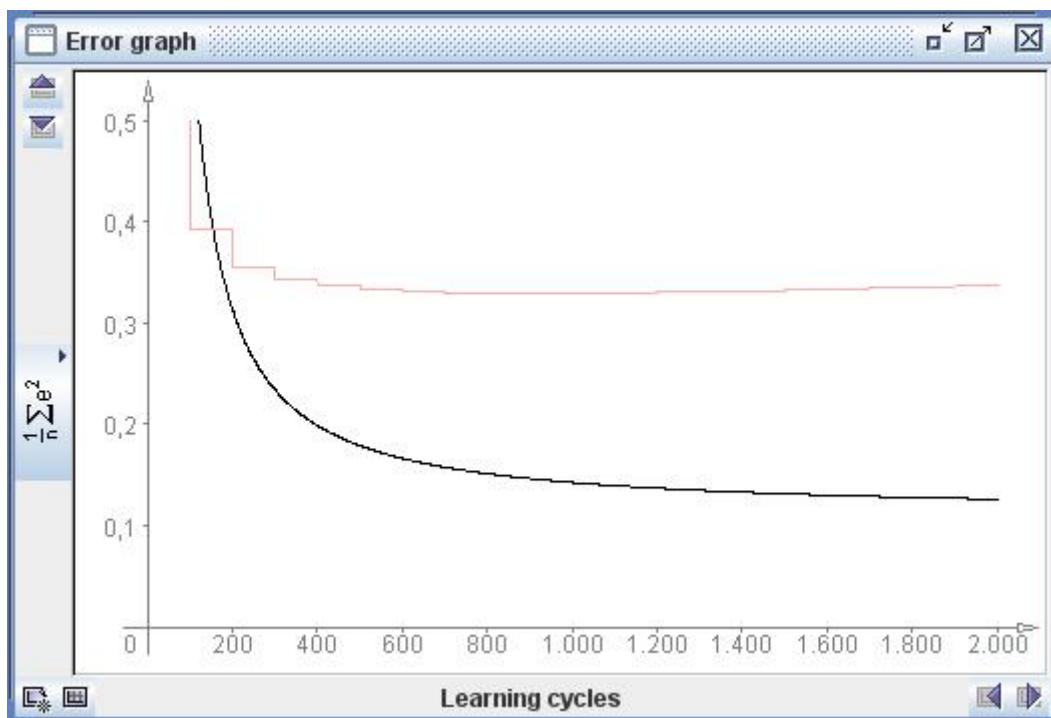
Ratio = 0,3 / dmax= 0



Ratio = 0,1 / dmax= 0



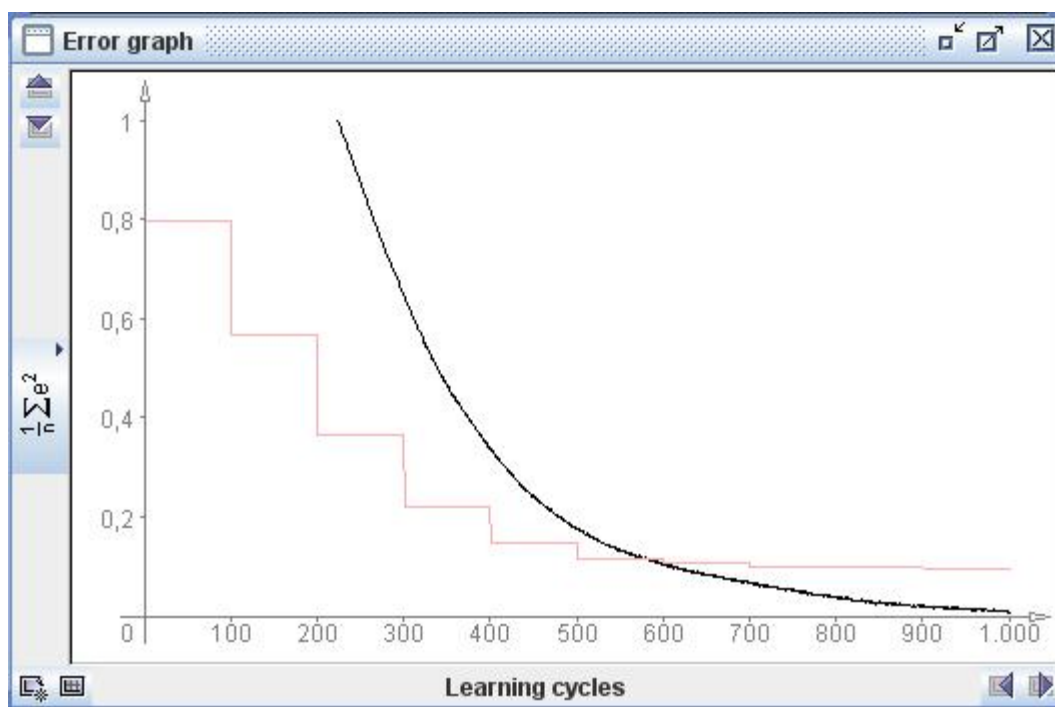
Ratio = 0,1 / dmax= 0,2



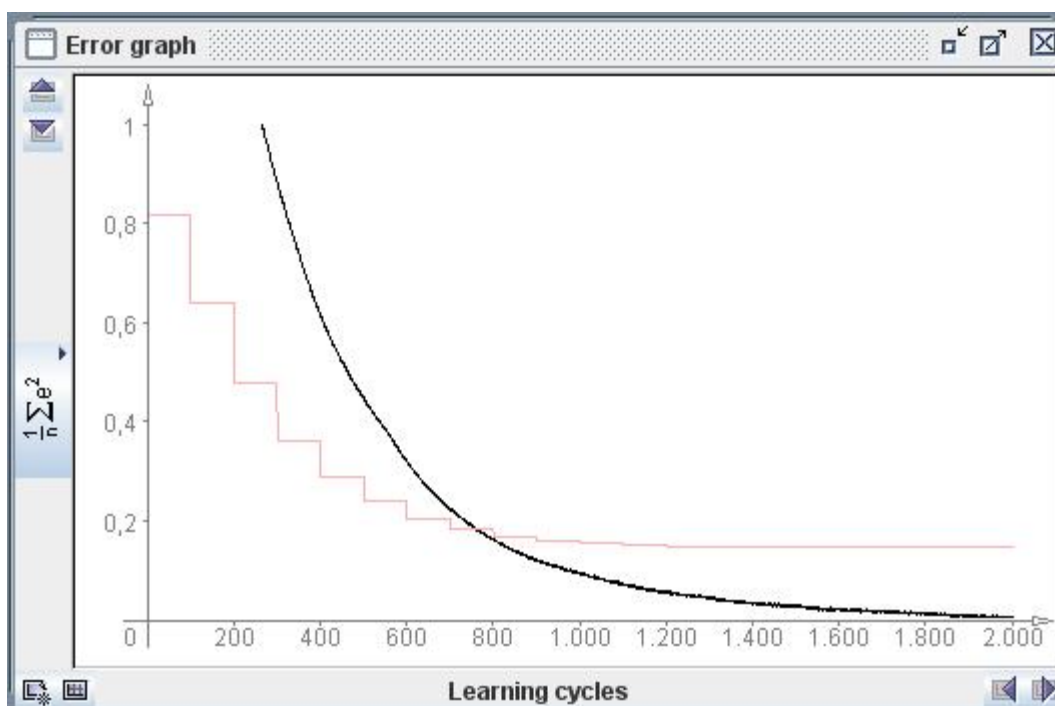
Ratio = 0,3 / dmax= 0,2



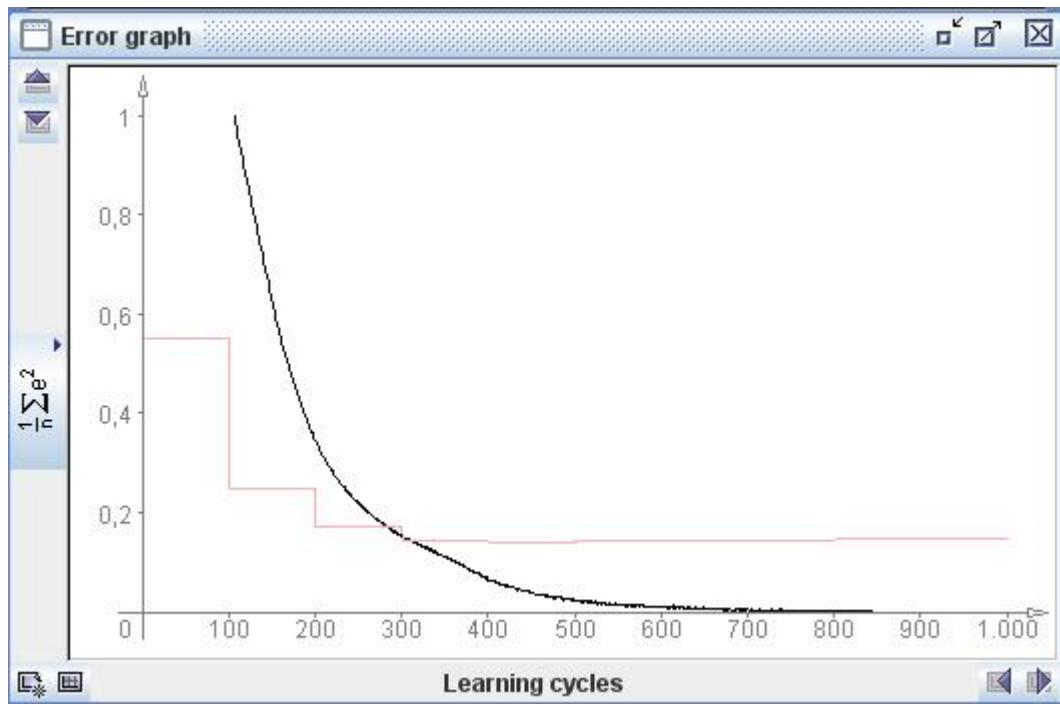
Representación por posición:



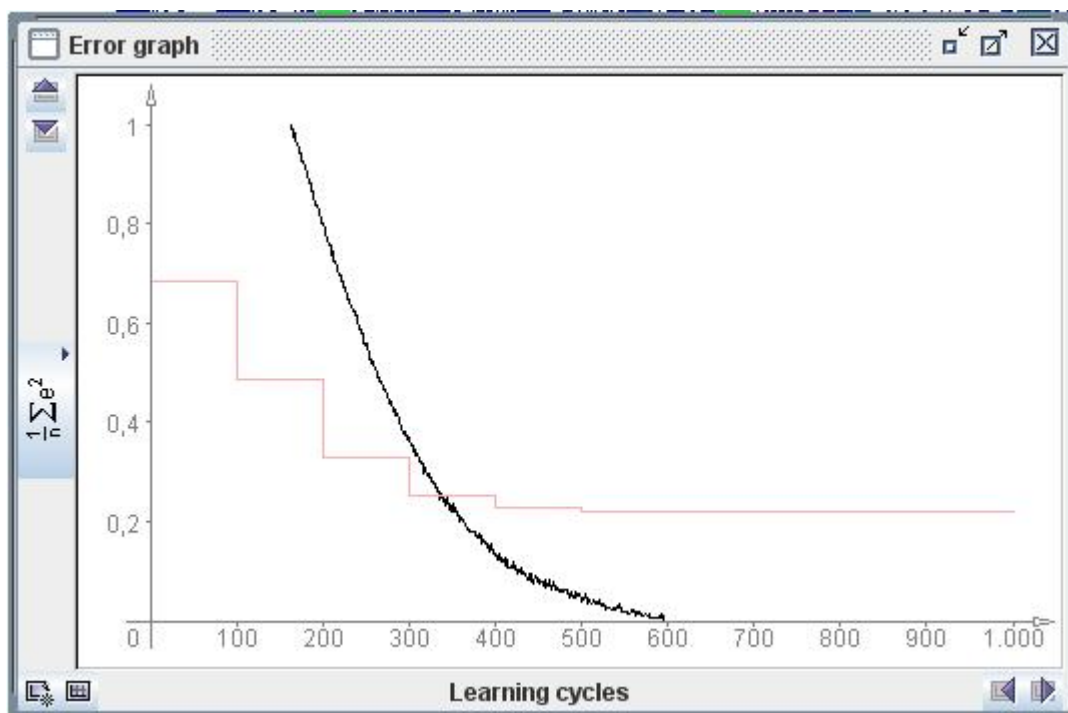
Ratio = 0,2 / dmax= 0,1



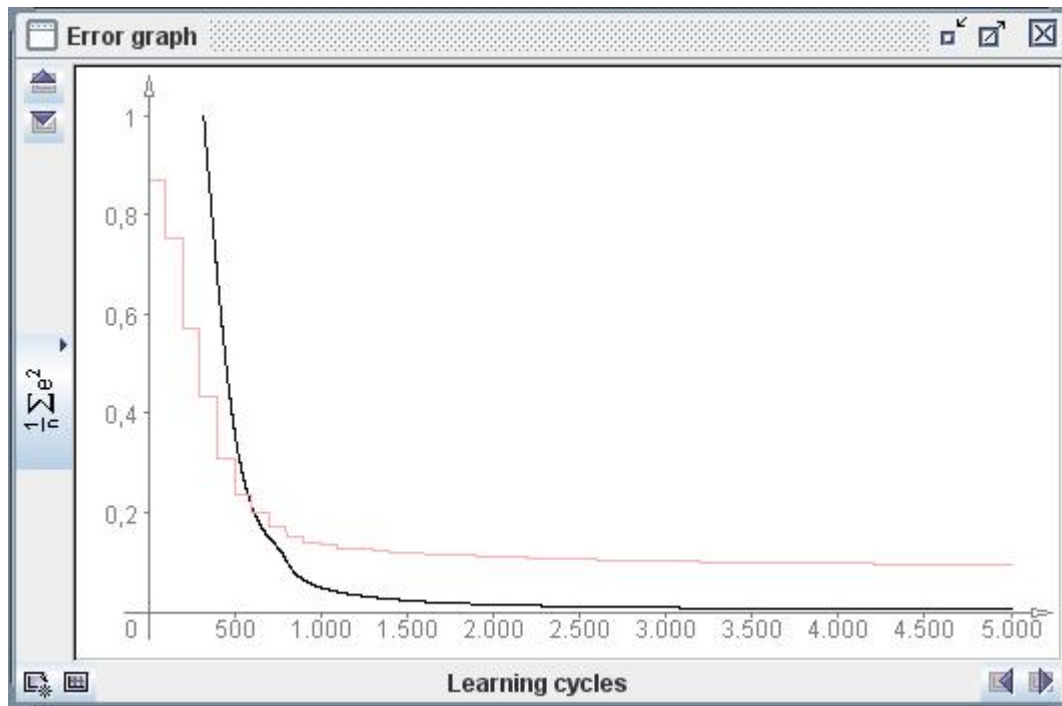
Ratio = 0,1 / dmax= 0,1



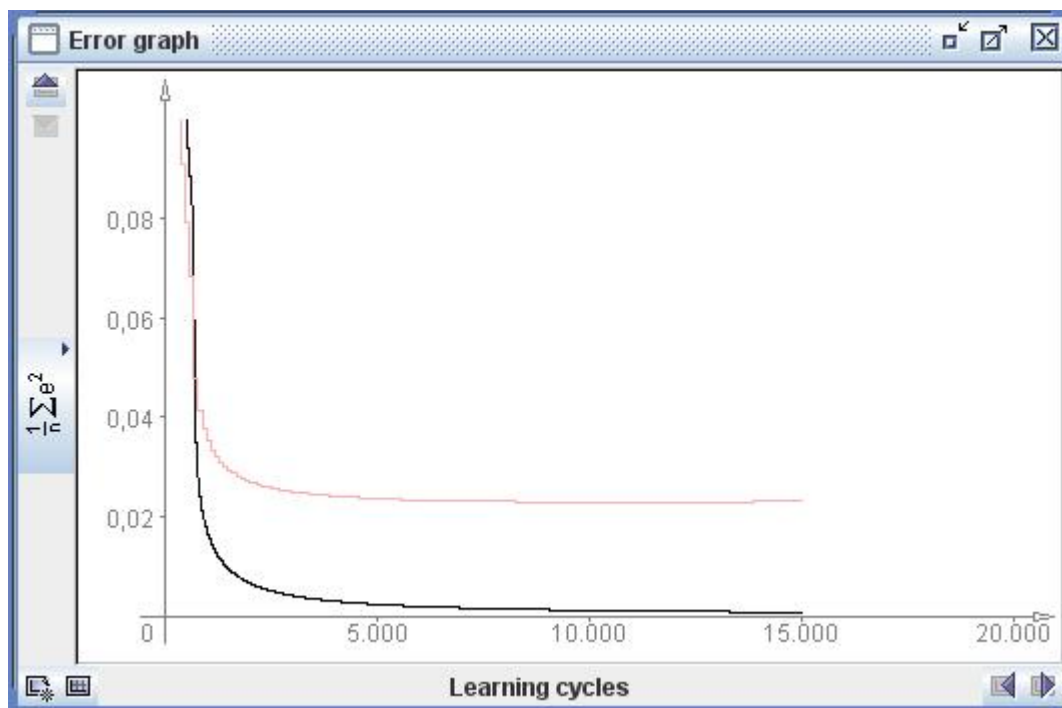
Ratio = 0,3 / dmax= 0,1



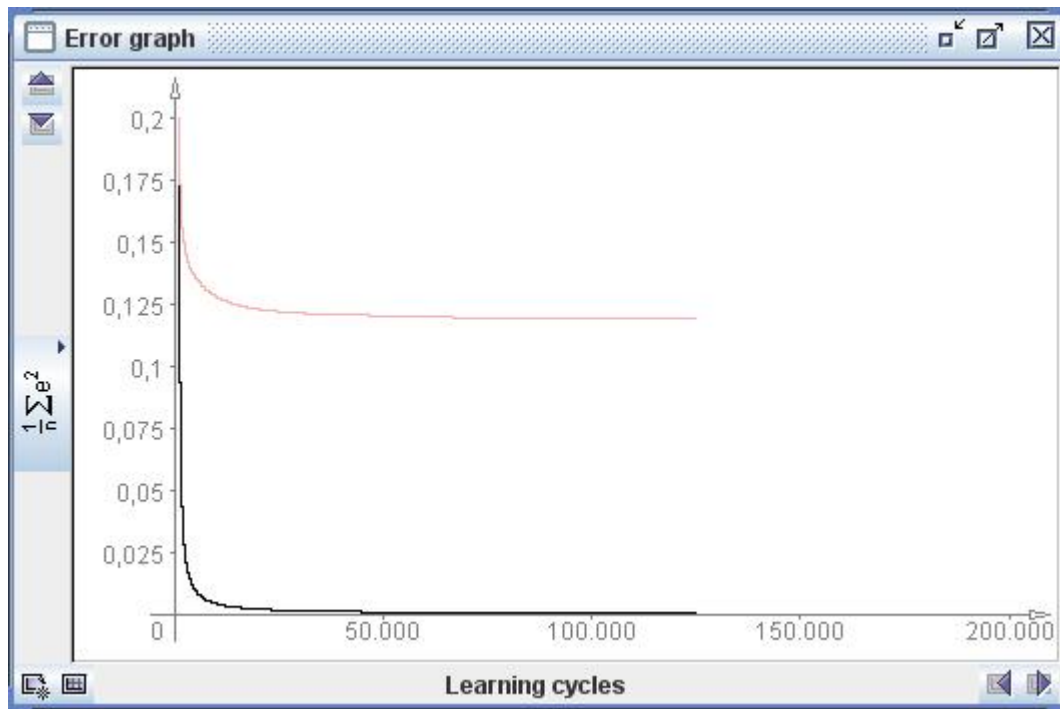
Ratio = 0,2 / dmax= 0,2



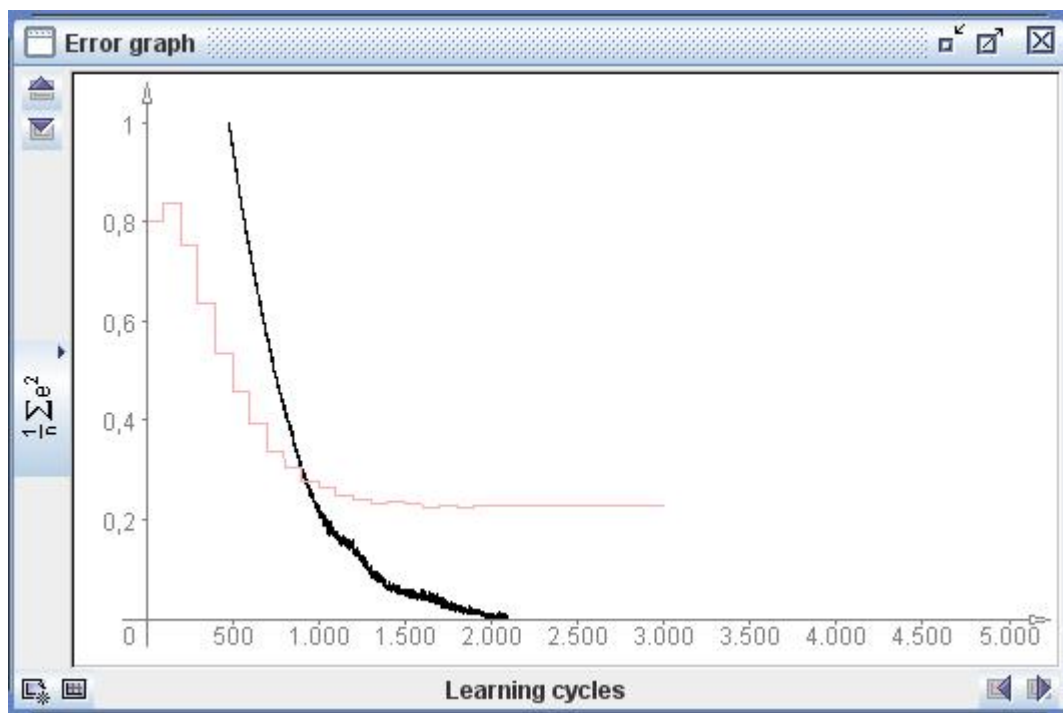
Ratio = 0,2 / dmax= 0



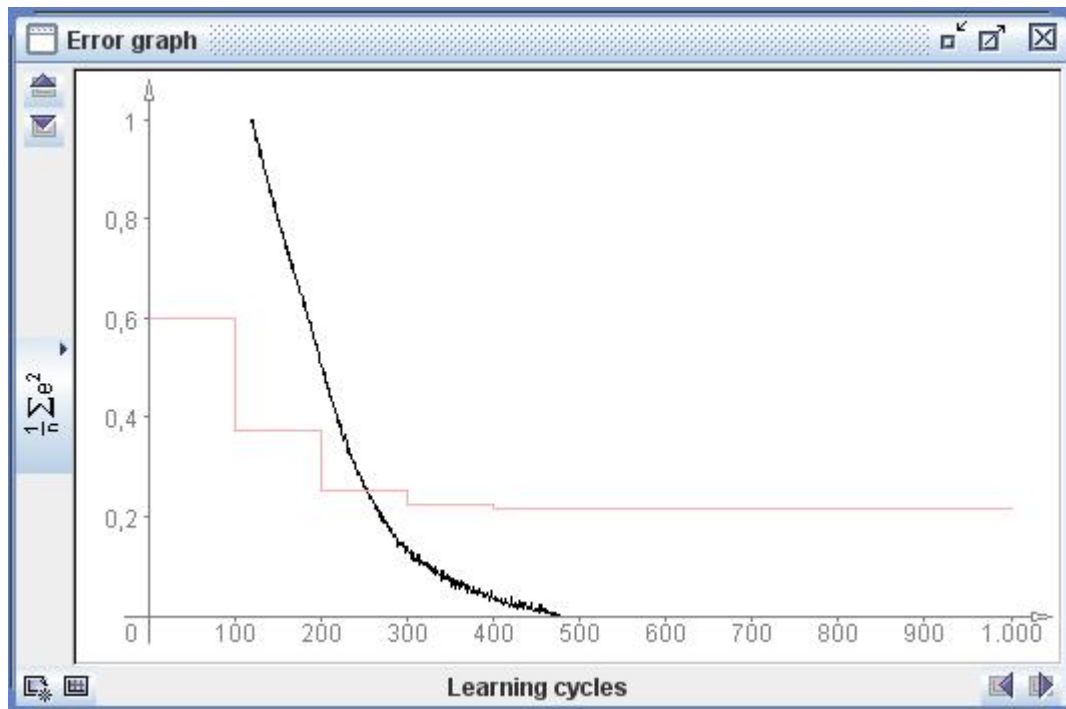
Ratio = 0,3 / dmax= 0



Ratio = 0,1 / dmax= 0



Ratio = 0,1 / dmax= 0,2



Ratio = 0,3 / dmax= 0,2

- Conclusiones

Tras haber utilizado los diferentes modelos de representación, se observa claramente como el número de unidades de proceso de la capa oculta influye directamente en la convergencia de la red, ya que, a mayor número de unidades, la convergencia se realiza de una forma más rápida.

Tal y como podemos ver en la gráfica, la representación binaria parece ser la que converge con mayor rapidez. Aunque en esta última, el número de unidades en la capa oculta no ha afectado tanto como en las otras representaciones.

A su vez, tal y como hemos comprobado, la inicialización de los pesos es algo muy importante ya que influye en los tiempos de convergencia de la red puesto que podemos encontrar un mínimo local, lo que afecta bastante en la convergencia.

Como hemos podido apreciar, el ratio de aprendizaje influye directamente en la velocidad de convergencia, ya que si el valor es mayor, la red converge con mayor rapidez.

Durante el estudio de los caracteres con ruido, hemos apreciado que la red tiene que ser entrenada con un conjunto de ruido “falso” ya que sino, no será capaz de reconocer los caracteres que difieran un poco con los que se han empleado para el entrenamiento, por ello, hemos buscado una serie de ruidos que faciliten la generalización de la red, puesto que hay ciertos tipos de ruido, que afectan de manera importante al funcionamiento de la red, resultando esto en la no generalización de la misma.

Los patrones de validación, se han realizado también de manera que tengamos

un ruido diferente al del entrenamiento, aunque en este caso también, pueden darse casos en el que un ruido en específico imposibilite la generalización de la red.

- Bibliografía

- Neural Networks – Algorithms, Applications and Programming Techniques.  
James A. Freeman / David M. Skapura
- Neural Networks  
Raúl Rojas

- Anexo

### Patrón del XOR

SNNS pattern definition file V3.2  
generated at Fri Apr 22 15:24:52 1994

No. of patterns : 4  
No. of input units : 2  
No. of output units : 1

```
# Input pattern 1:  
0 0  
# Output pattern 1:  
0  
# Input pattern 2:  
0 1  
# Output pattern 2:  
1  
# Input pattern 3:  
1 0  
# Output pattern 3:  
1  
# Input pattern 4:  
1 1  
# Output pattern 4:  
0
```

### Patrón salida binaria

SNNS pattern definition file V3.2  
generated at Fri Apr 22 15:24:52 1994

No. of patterns : 72  
No. of input units : 35  
No. of output units : 8

```
# Input pattern 1:  
1 1 1 1 1  
1 0 0 1 1  
1 0 1 0 1  
1 0 1 0 1  
1 0 1 0 1  
1 1 0 0 1  
1 1 1 1 1  
# Output pattern 1:  
0 0 1 1 0 0 0 0  
# Input pattern 2:  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1
```

```

0 0 0 0 1
0 0 0 0 1
0 0 0 0 1
0 0 0 0 1
# Output pattern 2:
0 0 1 1 0 0 0 1
# Input pattern 3:
1 1 1 1 1
0 0 0 0 1
0 0 0 0 1
1 1 1 1 1
1 0 0 0 0
1 0 0 0 0
1 1 1 1 1
# Output pattern 3:
0 0 1 1 0 0 1 0
# Input pattern 4:
1 1 1 1 1
0 0 0 0 1
0 0 0 0 1
1 1 1 1 1
0 0 0 0 1
0 0 0 0 1
1 1 1 1 1
# Output pattern 4:
0 0 1 1 0 0 1 1
# Input pattern 5:
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
1 1 1 1 1
0 0 0 0 1
0 0 0 0 1
0 0 0 0 1
# Output pattern 5:
0 0 1 1 0 1 0 0
# Input pattern 6:
1 1 1 1 1
1 0 0 0 0
1 0 0 0 0
1 1 1 1 1
0 0 0 0 1
0 0 0 0 1
1 1 1 1 1
# Output pattern 6:
0 0 1 1 0 1 0 1
# Input pattern 7:
1 1 1 1 1
1 0 0 0 0
1 0 0 0 0
1 1 1 1 1

```



```

1 0 0 0 1
1 0 0 0 1
1 1 1 1 1
# Output pattern 7:
0 0 1 1 0 1 1 0
# Input pattern 8:
1 1 1 1 1
0 0 0 0 1
0 0 0 0 1
0 0 0 0 1
0 0 0 0 1
0 0 0 0 1
0 0 0 0 1
# Output pattern 8:
0 0 1 1 0 1 1 1
# Input pattern 9:
1 1 1 1 1
1 0 0 0 1
1 0 0 0 1
1 1 1 1 1
1 0 0 0 1
1 0 0 0 1
1 1 1 1 1
# Output pattern 9:
0 0 1 1 1 0 0 0
# Input pattern 10:
1 1 1 1 1
1 0 0 0 1
1 0 0 0 1
1 1 1 1 1
0 0 0 0 1
0 0 0 0 1
0 0 0 0 1
# Output pattern 10:
0 0 1 1 1 0 0 1
# Input pattern 11:
0 0 1 0 0
0 1 0 1 0
1 0 0 0 1
1 1 1 1 1
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
# Output pattern 11:
0 1 0 0 0 0 1
# Input pattern 12:
1 1 1 1 0
1 0 0 0 1
1 0 0 0 1
1 1 1 1 0
1 0 0 0 1

```

```

1 0 0 0 1
1 1 1 1 0
# Output pattern 12:
0 1 0 0 0 0 1 0
# Input pattern 13:
0 1 1 1 1
1 0 0 0 0
1 0 0 0 0
1 0 0 0 0
1 0 0 0 0
1 0 0 0 0
0 1 1 1 1
# Output pattern 13:
0 1 0 0 0 0 1 1
# Input pattern 14:
1 1 1 0 0
1 0 0 1 0
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
1 0 0 1 0
1 1 1 0 0
# Output pattern 14:
0 1 0 0 0 1 0 0
# Input pattern 15:
1 1 1 1 1
1 0 0 0 0
1 0 0 0 0
1 1 1 1 1
1 0 0 0 0
1 0 0 0 0
1 1 1 1 1
# Output pattern 15:
0 1 0 0 0 1 0 1
# Input pattern 16:
1 1 1 1 1
1 0 0 0 0
1 0 0 0 0
1 1 1 1 1
1 0 0 0 0
1 0 0 0 0
1 0 0 0 0
# Output pattern 16:
0 1 0 0 0 1 1 0
# Input pattern 17:
0 1 1 1 1
1 0 0 0 0
1 0 0 0 0
1 0 1 1 1
1 0 0 0 1
1 0 0 0 1

```

```

0 1 1 1 0
# Output pattern 17:
0 1 0 0 0 1 1 1
# Input pattern 18:
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
1 1 1 1 1
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
# Output pattern 18:
0 1 0 0 1 0 0 0
# Input pattern 19:
0 1 1 1 0
0 0 1 0 0
0 0 1 0 0
0 0 1 0 0
0 0 1 0 0
0 0 1 0 0
0 1 1 1 0
# Output pattern 19:
0 1 0 0 1 0 0 1
# Input pattern 20:
0 0 0 0 1
0 0 0 0 1
0 0 0 0 1
0 0 0 0 1
0 0 0 0 1
1 0 0 0 1
0 1 1 1 0
# Output pattern 20:
0 1 0 0 1 0 1 0
# Input pattern 21:
1 0 0 0 1
1 0 0 1 0
1 0 1 0 0
1 1 0 0 0
1 0 1 0 0
1 0 0 1 0
1 0 0 0 1
# Output pattern 21:
0 1 0 0 1 0 1 1
# Input pattern 22:
1 0 0 0 0
1 0 0 0 0
1 0 0 0 0
1 0 0 0 0
1 0 0 0 0
1 0 0 0 0
1 1 1 1 1

```

```

# Output pattern 22:
0 1 0 0 1 1 0 0
# Input pattern 23:
1 0 0 0 1
1 1 0 1 1
1 0 1 0 1
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
# Output pattern 23:
0 1 0 0 1 1 0 1
# Input pattern 24:
1 0 0 0 1
1 1 0 0 1
1 1 1 0 1
1 0 1 1 1
1 0 0 1 1
1 0 0 0 1
1 0 0 0 1
# Output pattern 24:
0 1 0 0 1 1 1 0
# Input pattern 25:
0 1 1 1 0
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
0 1 1 1 0
# Output pattern 25:
0 1 0 0 1 1 1 1
# Input pattern 26:
1 1 1 1 0
1 0 0 0 1
1 0 0 0 1
1 1 1 1 0
1 0 0 0 0
1 0 0 0 0
1 0 0 0 0
# Output pattern 26:
0 1 0 1 0 0 0 0
# Input pattern 27:
0 1 1 1 0
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
1 0 1 0 1
1 0 0 1 0
0 1 1 0 1
# Output pattern 27:

```

```

0 1 0 1 0 0 0 1
# Input pattern 28:
1 1 1 1 0
1 0 0 0 1
1 0 0 0 1
1 1 1 1 0
1 0 1 0 0
1 0 0 1 0
1 0 0 0 1
# Output pattern 28:
0 1 0 1 0 0 1 0
# Input pattern 29:
0 1 1 1 1
1 0 0 0 0
1 0 0 0 0
0 1 1 1 0
0 0 0 0 1
0 0 0 0 1
1 1 1 1 0
# Output pattern 29:
0 1 0 1 0 0 1 1
# Input pattern 30:
1 1 1 1 1
0 0 1 0 0
0 0 1 0 0
0 0 1 0 0
0 0 1 0 0
0 0 1 0 0
0 0 1 0 0
# Output pattern 30:
0 1 0 1 0 1 0 0
# Input pattern 31:
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
0 1 1 1 0
# Output pattern 31:
0 1 0 1 0 1 0 1
# Input pattern 32:
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
0 1 0 1 0
0 0 1 0 0
# Output pattern 32:
0 1 0 1 0 1 1 0

```

```

# Input pattern 33:
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
1 0 1 0 1
1 1 0 1 1
1 0 0 0 1
# Output pattern 33:
0 1 0 1 0 1 1 1
# Input pattern 34:
1 0 0 0 1
1 0 0 0 1
0 1 0 1 0
0 0 1 0 0
0 1 0 1 0
1 0 0 0 1
1 0 0 0 1
# Output pattern 34:
0 1 0 1 1 0 0 0
# Input pattern 35:
1 0 0 0 1
1 0 0 0 1
0 1 0 1 0
0 0 1 0 0
0 0 1 0 0
0 0 1 0 0
0 0 1 0 0
# Output pattern 35:
0 1 0 1 1 0 0 1
# Input pattern 36:
1 1 1 1 1
0 0 0 0 1
0 0 0 1 0
0 0 1 0 0
0 1 0 0 0
1 0 0 0 0
1 1 1 1 1
# Output pattern 36:
0 1 0 1 1 0 1 0

# Input pattern 1:
1 1 1 1 1
1 0 0 1 1
1 1 1 0 1
1 0 1 0 1
1 0 1 0 1
1 1 0 0 1
1 1 1 1 1
# Output pattern 1:
0 0 1 1 0 0 0 0

```

```

# Input pattern 2:
0 0 0 0 1
0 0 0 0 1
0 0 1 0 1
0 0 0 0 1
0 0 0 0 1
0 0 0 0 1
0 0 0 0 1
0 0 0 0 1
# Output pattern 2:
0 0 1 1 0 0 0 1
# Input pattern 3:
1 1 1 1 1
0 0 1 0 1
0 0 0 0 1
1 1 1 1 1
1 0 0 0 0
1 0 0 0 0
1 1 1 1 1
# Output pattern 3:
0 0 1 1 0 0 1 0
# Input pattern 4:
1 1 1 1 1
0 0 0 0 1
0 0 0 0 1
1 1 1 1 1
0 0 1 0 1
0 0 0 0 1
1 1 1 1 1
# Output pattern 4:
0 0 1 1 0 0 1 1
# Input pattern 5:
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
1 1 1 1 1
0 0 0 0 1
0 0 0 0 1
0 1 0 0 1
# Output pattern 5:
0 0 1 1 0 1 0 0
# Input pattern 6:
1 1 1 1 1
1 0 0 0 0
1 0 0 1 0
1 1 1 1 1
0 0 0 0 1
0 0 0 0 1
1 1 1 1 1
# Output pattern 6:
0 0 1 1 0 1 0 1
# Input pattern 7:

```

```

1 1 1 1 1
1 0 0 0 0
1 0 0 1 0
1 1 1 1 1
1 0 0 0 1
1 0 0 0 1
1 1 1 1 1
# Output pattern 7:
0 0 1 1 0 1 1 0
# Input pattern 8:
1 1 1 1 1
0 0 0 0 1
0 0 0 0 1
0 0 0 0 1
0 0 0 0 1
1 0 0 0 1
0 0 0 0 1
# Output pattern 8:
0 0 1 1 0 1 1 1
# Input pattern 9:
1 1 1 1 1
1 0 0 0 1
1 0 0 0 1
1 1 1 1 1
1 0 1 0 1
1 0 0 0 1
1 1 1 1 1
# Output pattern 9:
0 0 1 1 1 0 0 0
# Input pattern 10:
1 1 1 1 1
1 0 0 0 1
1 0 0 0 1
1 1 1 1 1
0 0 0 0 1
0 1 0 0 1
0 0 0 0 1
# Output pattern 10:
0 0 1 1 1 0 0 1
# Input pattern 11:
0 0 1 0 0
0 1 0 1 0
1 0 0 0 1
1 1 1 1 1
1 0 0 0 1
1 0 0 0 1
1 0 1 0 1
# Output pattern 11:
0 1 0 0 0 0 1
# Input pattern 12:
1 1 1 1 0

```



```

1 0 0 0 1
1 0 0 0 1
1 1 1 1 0
1 0 1 0 1
1 0 0 0 1
1 1 1 1 0
# Output pattern 12:
0 1 0 0 0 0 1 0
# Input pattern 13:
0 1 1 1 1
1 0 0 0 0
1 0 0 1 0
1 0 0 0 0
1 0 0 0 0
1 0 0 0 0
0 1 1 1 1
# Output pattern 13:
0 1 0 0 0 0 1 1
# Input pattern 14:
1 1 1 0 0
1 0 0 1 0
1 0 0 0 1
1 0 1 0 1
1 0 0 0 1
1 0 0 1 0
1 1 1 0 0
# Output pattern 14:
0 1 0 0 0 1 0 0
# Input pattern 15:
1 1 1 1 1
1 0 0 0 0
1 0 1 0 0
1 1 1 1 1
1 0 0 0 0
1 0 0 0 0
1 1 1 1 1
# Output pattern 15:
0 1 0 0 0 1 0 1
# Input pattern 16:
1 1 1 1 1
1 0 0 1 0
1 0 0 0 0
1 1 1 1 1
1 0 0 0 0
1 0 0 0 0
1 0 0 0 0
# Output pattern 16:
0 1 0 0 0 1 1 0
# Input pattern 17:
0 1 1 1 1
1 0 0 0 0

```

```

1 0 0 0 1
1 0 1 1 1
1 0 0 0 1
1 0 0 0 1
0 1 1 1 0
# Output pattern 17:
0 1 0 0 0 1 1 1
# Input pattern 18:
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
1 1 1 1 1
1 0 0 0 1
1 0 1 0 1
1 0 0 0 1
# Output pattern 18:
0 1 0 0 1 0 0 0
# Input pattern 19:
0 1 1 1 0
0 0 1 0 0
0 0 1 0 0
0 0 1 0 0
0 1 1 0 0
0 0 1 0 0
0 1 1 1 0
# Output pattern 19:
0 1 0 0 1 0 0 1
# Input pattern 20:
0 0 0 0 1
0 0 0 0 1
0 1 0 0 1
0 0 0 0 1
0 0 0 0 1
1 0 0 0 1
0 1 1 1 0
# Output pattern 20:
0 1 0 0 1 0 1 0
# Input pattern 21:
1 0 0 0 1
1 0 0 1 0
1 0 1 0 0
1 1 0 0 0
1 0 1 0 0
1 0 0 1 0
1 1 0 0 1
# Output pattern 21:
0 1 0 0 1 0 1 1
# Input pattern 22:
1 0 0 0 0
1 0 0 0 0
1 0 0 0 0

```

```

1 0 0 0 1
1 0 0 0 0
1 0 0 0 0
1 1 1 1 1
# Output pattern 22:
0 1 0 0 1 1 0 0
# Input pattern 23:
1 0 0 0 1
1 1 0 1 1
1 0 1 0 1
1 0 0 0 1
1 0 0 0 1
1 0 1 0 1
1 0 0 0 1
# Output pattern 23:
0 1 0 0 1 1 0 1
# Input pattern 24:
1 0 0 1 1
1 1 0 0 1
1 1 1 0 1
1 0 1 1 1
1 0 0 1 1
1 0 0 0 1
1 0 0 0 1
# Output pattern 24:
0 1 0 0 1 1 1 0
# Input pattern 25:
0 1 1 1 0
1 0 0 0 1
1 0 0 1 1
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
0 1 1 1 0
# Output pattern 25:
0 1 0 0 1 1 1 1
# Input pattern 26:
1 1 1 1 0
1 0 0 0 1
1 0 0 0 1
1 1 1 1 0
1 0 0 0 0
1 0 0 0 0
1 0 0 1 0
# Output pattern 26:
0 1 0 1 0 0 0 0
# Input pattern 27:
0 1 1 1 0
1 0 0 0 1
1 0 0 1 1
1 0 0 0 1

```

```

1 0 1 0 1
1 0 0 1 0
0 1 1 0 1
# Output pattern 27:
0 1 0 1 0 0 0 1
# Input pattern 28:
1 1 1 1 0
1 0 0 0 1
1 0 1 0 1
1 1 1 1 0
1 0 1 0 0
1 0 0 1 0
1 0 0 0 1
# Output pattern 28:
0 1 0 1 0 0 1 0
# Input pattern 29:
0 1 1 1 1
1 0 0 1 0
1 0 0 0 0
0 1 1 1 0
0 0 0 0 1
0 0 0 0 1
1 1 1 1 0
# Output pattern 29:
0 1 0 1 0 0 1 1
# Input pattern 30:
1 1 1 1 1
0 0 1 0 0
0 0 1 0 0
0 0 1 0 0
0 1 1 0 0
0 0 1 0 0
0 0 1 0 0
# Output pattern 30:
0 1 0 1 0 1 0 0
# Input pattern 31:
1 0 0 0 1
1 0 0 0 1
1 0 0 1 1
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
0 1 1 1 0
# Output pattern 31:
0 1 0 1 0 1 0 1
# Input pattern 32:
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
1 0 0 1 1
1 0 0 0 1

```

```

0 1 0 1 0
0 0 1 0 0
# Output pattern 32:
0 1 0 1 0 1 1 0
# Input pattern 33:
1 0 0 0 1
1 0 0 0 1
1 0 0 1 1
1 0 0 0 1
1 0 1 0 1
1 1 0 1 1
1 0 0 0 1
# Output pattern 33:
0 1 0 1 0 1 1 1
# Input pattern 34:
1 0 0 0 1
1 0 0 0 1
0 1 0 1 0
0 0 1 0 0
0 1 0 1 0
1 0 0 0 1
1 0 0 1 1
# Output pattern 34:
0 1 0 1 1 0 0 0
# Input pattern 35:
1 1 0 0 1
1 0 0 0 1
0 1 0 1 0
0 0 1 0 0
0 0 1 0 0
0 0 1 0 0
0 0 1 0 0
# Output pattern 35:
0 1 0 1 1 0 0 1
# Input pattern 36:
1 1 1 1 1
0 0 0 0 1
0 0 0 1 0
0 0 1 0 0
0 1 0 1 0
1 0 0 0 0
1 1 1 1 1
# Output pattern 36:
0 1 0 1 1 0 1 0

```

Patrón salida matricial

No. of patterns : 72  
No. of input units : 35  
No. of output units : 35

# Input pattern 1:

1 1 1 1 1  
1 0 0 1 1  
1 0 1 0 1  
1 0 1 0 1  
1 0 1 0 1  
1 1 0 0 1  
1 1 1 1 1

# Output pattern 1:

1 1 1 1 1  
1 0 0 1 1  
1 0 1 0 1  
1 0 1 0 1  
1 0 1 0 1  
1 1 0 0 1  
1 1 1 1 1

# Input pattern 2:

0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1

# Output pattern 2:

0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1

# Input pattern 3:

1 1 1 1 1  
0 0 0 0 1  
0 0 0 0 1  
1 1 1 1 1  
1 0 0 0 0  
1 0 0 0 0  
1 1 1 1 1

# Output pattern 3:

1 1 1 1 1  
0 0 0 0 1  
0 0 0 0 1  
1 1 1 1 1

1 0 0 0 0  
1 0 0 0 0  
1 1 1 1 1

# Input pattern 4:

1 1 1 1 1  
0 0 0 0 1  
0 0 0 0 1  
1 1 1 1 1  
0 0 0 0 1  
0 0 0 0 1  
1 1 1 1 1

# Output pattern 4:

1 1 1 1 1  
0 0 0 0 1  
0 0 0 0 1  
1 1 1 1 1  
0 0 0 0 1  
0 0 0 0 1  
1 1 1 1 1

# Input pattern 5:

1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 1 1 1 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1

# Output pattern 5:

1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 1 1 1 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1

# Input pattern 6:

1 1 1 1 1  
1 0 0 0 0  
1 0 0 0 0  
1 1 1 1 1  
0 0 0 0 1  
0 0 0 0 1  
1 1 1 1 1

# Output pattern 6:

1 1 1 1 1  
1 0 0 0 0  
1 0 0 0 0  
1 1 1 1 1

0 0 0 0 1  
0 0 0 0 1  
1 1 1 1 1

# Input pattern 7:

1 1 1 1 1  
1 0 0 0 0  
1 0 0 0 0  
1 1 1 1 1  
1 0 0 0 1  
1 0 0 0 1  
1 1 1 1 1

# Output pattern 7:

1 1 1 1 1  
1 0 0 0 0  
1 0 0 0 0  
1 1 1 1 1  
1 0 0 0 1  
1 0 0 0 1  
1 1 1 1 1

# Input pattern 8:

1 1 1 1 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1

# Output pattern 8:

1 1 1 1 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1

# Input pattern 9:

1 1 1 1 1  
1 0 0 0 1  
1 0 0 0 1  
1 1 1 1 1  
1 0 0 0 1  
1 0 0 0 1  
1 1 1 1 1

# Output pattern 9:

1 1 1 1 1  
1 0 0 0 1  
1 0 0 0 1  
1 1 1 1 1



1 0 0 0 1  
1 0 0 0 1  
1 1 1 1 1

# Input pattern 10:

1 1 1 1 1  
1 0 0 0 1  
1 0 0 0 1  
1 1 1 1 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1

# Output pattern 10:

1 1 1 1 1  
1 0 0 0 1  
1 0 0 0 1  
1 1 1 1 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1

# Input pattern 11:

0 0 1 0 0  
0 1 0 1 0  
1 0 0 0 1  
1 1 1 1 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1

# Output pattern 11:

0 0 1 0 0  
0 1 0 1 0  
1 0 0 0 1  
1 1 1 1 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1

#Input pattern 12:

1 1 1 1 0  
1 0 0 0 1  
1 0 0 0 1  
1 1 1 1 0  
1 0 0 0 1  
1 0 0 0 1  
1 1 1 1 0

# Output pattern 12:

1 1 1 1 0  
1 0 0 0 1  
1 0 0 0 1  
1 1 1 1 0

1 0 0 0 1  
1 0 0 0 1  
1 1 1 1 0

#Input pattern 13:

0 1 1 1 1  
1 0 0 0 0  
1 0 0 0 0  
1 0 0 0 0  
1 0 0 0 0  
1 0 0 0 0  
0 1 1 1 1

# Output pattern 13:

0 1 1 1 1  
1 0 0 0 0  
1 0 0 0 0  
1 0 0 0 0  
1 0 0 0 0  
1 0 0 0 0  
0 1 1 1 1

#Input pattern 14:

1 1 1 0 0  
1 0 0 1 0  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 1 0  
1 1 1 0 0

# Output pattern 14:

1 1 1 0 0  
1 0 0 1 0  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 1 0  
1 1 1 0 0

#Input pattern 15:

1 1 1 1 1  
1 0 0 0 0  
1 0 0 0 0  
1 1 1 1 1  
1 0 0 0 0  
1 0 0 0 0  
1 1 1 1 1

# Output pattern 15:

1 1 1 1 1  
1 0 0 0 0  
1 0 0 0 0  
1 1 1 1 1

1 0 0 0 0  
1 0 0 0 0  
1 1 1 1 1

#Input pattern 16:

1 1 1 1 1  
1 0 0 0 0  
1 0 0 0 0  
1 1 1 1 1  
1 0 0 0 0  
1 0 0 0 0  
1 0 0 0 0

# Output pattern 16:

1 1 1 1 1  
1 0 0 0 0  
1 0 0 0 0  
1 1 1 1 1  
1 0 0 0 0  
1 0 0 0 0  
1 0 0 0 0

#Input pattern 17:

0 1 1 1 1  
1 0 0 0 0  
1 0 0 0 0  
1 0 1 1 1  
1 0 0 0 1  
1 0 0 0 1  
0 1 1 1 0

# Output pattern 17:

0 1 1 1 1  
1 0 0 0 0  
1 0 0 0 0  
1 0 1 1 1  
1 0 0 0 1  
1 0 0 0 1  
0 1 1 1 0

#Input pattern 18:

1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 1 1 1 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1

# Output pattern 18:

1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 1 1 1 1

1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1

#Input pattern 19:

0 1 1 1 0  
0 0 1 0 0  
0 0 1 0 0  
0 0 1 0 0  
0 0 1 0 0  
0 0 1 0 0  
0 0 1 0 0  
0 1 1 1 0

# Output pattern 19:

0 1 1 1 0  
0 0 1 0 0  
0 0 1 0 0  
0 0 1 0 0  
0 0 1 0 0  
0 0 1 0 0  
0 0 1 0 0  
0 1 1 1 0

#Input pattern 20:

0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1  
1 0 0 0 1  
0 1 1 1 0

# Output pattern 20:

0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1  
1 0 0 0 1  
0 1 1 1 0

#Input pattern 21:

1 0 0 0 1  
1 0 0 1 0  
1 0 1 0 0  
1 1 0 0 0  
1 0 1 0 0  
1 0 0 1 0  
1 0 0 0 1

# Output pattern 21:

1 0 0 0 1  
1 0 0 1 0  
1 0 1 0 0  
1 1 0 0 0

1 0 1 0 0  
1 0 0 1 0  
1 0 0 0 1

#Input pattern 22:

1 0 0 0 0  
1 0 0 0 0  
1 0 0 0 0  
1 0 0 0 0  
1 0 0 0 0  
1 0 0 0 0  
1 1 1 1 1

# Output pattern 22:

1 0 0 0 0  
1 0 0 0 0  
1 0 0 0 0  
1 0 0 0 0  
1 0 0 0 0  
1 0 0 0 0  
1 1 1 1 1

#Input pattern 23:

1 0 0 0 1  
1 1 0 1 1  
1 0 1 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1

# Output pattern 23:

1 0 0 0 1  
1 1 0 1 1  
1 0 1 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1

#Input pattern 24:

1 0 0 0 1  
1 1 0 0 1  
1 1 1 0 1  
1 0 1 1 1  
1 0 0 1 1  
1 0 0 0 1  
1 0 0 0 1

# Output pattern 24:

1 0 0 0 1  
1 1 0 0 1  
1 1 1 0 1  
1 0 1 1 1

1 0 0 1 1  
1 0 0 0 1  
1 0 0 0 1

#Input pattern 25:

0 1 1 1 0  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
0 1 1 1 0

# Output pattern 25:

0 1 1 1 0  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
0 1 1 1 0

#Input pattern 26:

1 1 1 1 0  
1 0 0 0 1  
1 0 0 0 1  
1 1 1 1 0  
1 0 0 0 0  
1 0 0 0 0  
1 0 0 0 0

# Output pattern 26:

1 1 1 1 0  
1 0 0 0 1  
1 0 0 0 1  
1 1 1 1 0  
1 0 0 0 0  
1 0 0 0 0  
1 0 0 0 0

#Input pattern 27:

0 1 1 1 0  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 1 0 1  
1 0 0 1 0  
0 1 1 0 1

# Output pattern 27:

0 1 1 1 0  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1

1 0 1 0 1  
1 0 0 1 0  
0 1 1 0 1

#Input pattern 28:

1 1 1 1 0  
1 0 0 0 1  
1 0 0 0 1  
1 1 1 1 0  
1 0 1 0 0  
1 0 0 1 0  
1 0 0 0 1

# Output pattern 28:

1 1 1 1 0  
1 0 0 0 1  
1 0 0 0 1  
1 1 1 1 0  
1 0 1 0 0  
1 0 0 1 0  
1 0 0 0 1

#Input pattern 29:

0 1 1 1 1  
1 0 0 0 0  
1 0 0 0 0  
0 1 1 1 0  
0 0 0 0 1  
0 0 0 0 1  
1 1 1 1 0

# Output pattern 29:

0 1 1 1 1  
1 0 0 0 0  
1 0 0 0 0  
0 1 1 1 0  
0 0 0 0 1  
0 0 0 0 1  
1 1 1 1 0

#Input pattern 30:

1 1 1 1 1  
0 0 1 0 0  
0 0 1 0 0  
0 0 1 0 0  
0 0 1 0 0  
0 0 1 0 0  
0 0 1 0 0

# Output pattern 30:

1 1 1 1 1  
0 0 1 0 0  
0 0 1 0 0  
0 0 1 0 0

0 0 1 0 0  
0 0 1 0 0  
0 0 1 0 0

#Input pattern 31:

1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
0 1 1 1 0

# Output pattern 31:

1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
0 1 1 1 0

#Input pattern 32:

1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
0 1 0 1 0  
0 0 1 0 0

# Output pattern 32:

1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
0 1 0 1 0  
0 0 1 0 0

#Input pattern 33:

1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 1 0 1  
1 1 0 1 1  
1 0 0 0 1

# Output pattern 33:

1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1



1 0 1 0 1  
1 1 0 1 1  
1 0 0 0 1

#Input pattern 34:

1 0 0 0 1  
1 0 0 0 1  
0 1 0 1 0  
0 0 1 0 0  
0 1 0 1 0  
1 0 0 0 1  
1 0 0 0 1

# Output pattern 34:

1 0 0 0 1  
1 0 0 0 1  
0 1 0 1 0  
0 0 1 0 0  
0 1 0 1 0  
1 0 0 0 1  
1 0 0 0 1

#Input pattern 35:

1 0 0 0 1  
1 0 0 0 1  
0 1 0 1 0  
0 0 1 0 0  
0 0 1 0 0  
0 0 1 0 0  
0 0 1 0 0  
0 0 1 0 0

# Output pattern 35:

1 0 0 0 1  
1 0 0 0 1  
0 1 0 1 0  
0 0 1 0 0  
0 0 1 0 0  
0 0 1 0 0  
0 0 1 0 0

#Input pattern 36:

1 1 1 1 1  
0 0 0 0 1  
0 0 0 1 0  
0 0 1 0 0  
0 1 0 0 0  
1 0 0 0 0  
1 1 1 1 1

# Output pattern 36:

1 1 1 1 1  
0 0 0 0 1  
0 0 0 1 0  
0 0 1 0 0

0 1 0 0 0  
1 0 0 0 0  
1 1 1 1 1

# Input pattern 1:

1 1 1 1 1  
1 0 0 1 1  
1 1 1 0 1  
1 0 1 0 1  
1 0 1 0 1  
1 1 0 0 1  
1 1 1 1 1

# Output pattern 1:

1 1 1 1 1  
1 0 0 1 1  
1 0 1 0 1  
1 0 1 0 1  
1 0 1 0 1  
1 1 0 0 1  
1 1 1 1 1

# Input pattern 2:

0 0 0 0 1  
0 0 0 0 1  
0 0 1 0 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1

# Output pattern 2:

0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1

# Input pattern 3:

1 1 1 1 1  
0 0 1 0 1  
0 0 0 0 1  
1 1 1 1 1  
1 0 0 0 0  
1 0 0 0 0  
1 1 1 1 1

# Output pattern 3:

1 1 1 1 1  
0 0 0 0 1  
0 0 0 0 1  
1 1 1 1 1

```

1 0 0 0 0
1 0 0 0 0
1 1 1 1 1
# Input pattern 4:
1 1 1 1 1
0 0 0 0 1
0 0 0 0 1
1 1 1 1 1
0 0 1 0 1
0 0 0 0 1
1 1 1 1 1
# Output pattern 4:
1 1 1 1 1
0 0 0 0 1
0 0 0 0 1
1 1 1 1 1
0 0 0 0 1
0 0 0 0 1
1 1 1 1 1
# Input pattern 5:
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
1 1 1 1 1
0 0 0 0 1
0 0 0 0 1
0 1 0 0 1
# Output pattern 5:
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
1 1 1 1 1
0 0 0 0 1
0 0 0 0 1
0 0 0 0 1
# Input pattern 6:
1 1 1 1 1
1 0 0 0 0
1 0 0 1 0
1 1 1 1 1
0 0 0 0 1
0 0 0 0 1
1 1 1 1 1
# Output pattern 6:
1 1 1 1 1
1 0 0 0 0
1 0 0 0 0
1 1 1 1 1
0 0 0 0 1
0 0 0 0 1
1 1 1 1 1

```

# Input pattern 7:

1 1 1 1 1

1 0 0 0 0

1 0 0 1 0

1 1 1 1 1

1 0 0 0 1

1 0 0 0 1

1 1 1 1 1

# Output pattern 7:

1 1 1 1 1

1 0 0 0 0

1 0 0 0 0

1 1 1 1 1

1 0 0 0 1

1 0 0 0 1

1 1 1 1 1

# Input pattern 8:

1 1 1 1 1

0 0 0 0 1

0 0 0 0 1

0 0 0 0 1

0 0 0 0 1

1 0 0 0 1

0 0 0 0 1

1 1 1 1 1

0 0 0 0 1

0 0 0 0 1

0 0 0 0 1

0 0 0 0 1

0 0 0 0 1

0 0 0 0 1

# Input pattern 9:

1 1 1 1 1

1 0 0 0 1

1 0 0 0 1

1 1 1 1 1

1 0 1 0 1

1 0 0 0 1

1 1 1 1 1

# Output pattern 9:

1 1 1 1 1

1 0 0 0 1

1 0 0 0 1

1 1 1 1 1

1 0 0 0 1

1 0 0 0 1

1 1 1 1 1

# Input pattern 10:

1 1 1 1 1

1 0 0 0 1

```

1 0 0 0 1
1 1 1 1 1
0 0 0 0 1
0 1 0 0 1
0 0 0 0 1
# Output pattern 10:
1 1 1 1 1
1 0 0 0 1
1 0 0 0 1
1 1 1 1 1
0 0 0 0 1
0 0 0 0 1
0 0 0 0 1
# Input pattern 11:
0 0 1 0 0
0 1 0 1 0
1 0 0 0 1
1 1 1 1 1
1 0 0 0 1
1 0 0 0 1
1 0 1 0 1
# Output pattern 11:
0 0 1 0 0
0 1 0 1 0
1 0 0 0 1
1 1 1 1 1
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
# Input pattern 12:
1 1 1 1 0
1 0 0 0 1
1 0 0 0 1
1 1 1 1 0
1 0 1 0 1
1 0 0 0 1
1 1 1 1 0
# Output pattern 12:
1 1 1 1 0
1 0 0 0 1
1 0 0 0 1
1 1 1 1 0
1 0 0 0 1
1 0 0 0 1
1 1 1 1 0
# Input pattern 13:
0 1 1 1 1
1 0 0 0 0
1 0 0 1 0
1 0 0 0 0
1 0 0 0 0

```

```

1 0 0 0 0
0 1 1 1 1
# Output pattern 13:
0 1 1 1 1
1 0 0 0 0
1 0 0 0 0
1 0 0 0 0
1 0 0 0 0
1 0 0 0 0
0 1 1 1 1
# Input pattern 14:
1 1 1 0 0
1 0 0 1 0
1 0 0 0 1
1 0 1 0 1
1 0 0 0 1
1 0 0 1 0
1 1 1 0 0
# Output pattern 14:
1 1 1 0 0
1 0 0 1 0
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
1 0 0 1 0
1 1 1 0 0
# Input pattern 15:
1 1 1 1 1
1 0 0 0 0
1 0 1 0 0
1 1 1 1 1
1 0 0 0 0
1 0 0 0 0
1 1 1 1 1
# Output pattern 15:
1 1 1 1 1
1 0 0 0 0
1 0 0 0 0
1 1 1 1 1
1 0 0 0 0
1 0 0 0 0
1 1 1 1 1
# Input pattern 16:
1 1 1 1 1
1 0 0 1 0
1 0 0 0 0
1 1 1 1 1
1 0 0 0 0
1 0 0 0 0
1 0 0 0 0
# Output pattern 16:

```

```

1 1 1 1 1
1 0 0 0 0
1 0 0 0 0
1 1 1 1 1
1 0 0 0 0
1 0 0 0 0
1 0 0 0 0
# Input pattern 17:
0 1 1 1 1
1 0 0 0 0
1 0 0 0 1
1 0 1 1 1
1 0 0 0 1
1 0 0 0 1
0 1 1 1 0
# Output pattern 17:
0 1 1 1 1
1 0 0 0 0
1 0 0 0 0
1 0 1 1 1
1 0 0 0 1
1 0 0 0 1
0 1 1 1 0
# Input pattern 18:
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
1 1 1 1 1
1 0 0 0 1
1 0 1 0 1
1 0 0 0 1
# Output pattern 18:
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
1 1 1 1 1
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
# Input pattern 19:
0 1 1 1 0
0 0 1 0 0
0 0 1 0 0
0 0 1 0 0
0 1 1 0 0
0 0 1 0 0
0 1 1 1 0
# Output pattern 19:
0 1 1 1 0
0 0 1 0 0
0 0 1 0 0

```

```

0 0 1 0 0
0 0 1 0 0
0 0 1 0 0
0 1 1 1 0
# Input pattern 20:
0 0 0 0 1
0 0 0 0 1
0 1 0 0 1
0 0 0 0 1
0 0 0 0 1
1 0 0 0 1
0 1 1 1 0
# Output pattern 20:
0 0 0 0 1
0 0 0 0 1
0 0 0 0 1
0 0 0 0 1
0 0 0 0 1
1 0 0 0 1
0 1 1 1 0
# Input pattern 21:
1 0 0 0 1
1 0 0 1 0
1 0 1 0 0
1 1 0 0 0
1 0 1 0 0
1 0 0 1 0
1 1 0 0 1
# Output pattern 21:
1 0 0 0 1
1 0 0 1 0
1 0 1 0 0
1 1 0 0 0
1 0 1 0 0
1 0 0 1 0
1 0 0 0 1
# Input pattern 22:
1 0 0 0 0
1 0 0 0 0
1 0 0 0 0
1 0 0 0 1
1 0 0 0 0
1 0 0 0 0
1 1 1 1 1
# Output pattern 22:
1 0 0 0 0
1 0 0 0 0
1 0 0 0 0
1 0 0 0 0
1 0 0 0 0
1 0 0 0 0

```



```

1 1 1 1 1
# Input pattern 23:
1 0 0 0 1
1 1 0 1 1
1 0 1 0 1
1 0 0 0 1
1 0 0 0 1
1 0 1 0 1
1 0 0 0 1
# Output pattern 23:
1 0 0 0 1
1 1 0 1 1
1 0 1 0 1
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
# Input pattern 24:
1 0 0 1 1
1 1 0 0 1
1 1 1 0 1
1 0 1 1 1
1 0 0 1 1
1 0 0 0 1
1 0 0 0 1
# Output pattern 24:
1 0 0 0 1
1 1 0 0 1
1 1 1 0 1
1 0 1 1 1
1 0 0 1 1
1 0 0 0 1
1 0 0 0 1
# Input pattern 25:
0 1 1 1 0
1 0 0 0 1
1 0 0 1 1
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
0 1 1 1 0
# Output pattern 25:
0 1 1 1 0
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
0 1 1 1 0
# Input pattern 26:
1 1 1 1 0

```

```

1 0 0 0 1
1 0 0 0 1
1 1 1 1 0
1 0 0 0 0
1 0 0 0 0
1 0 0 1 0
# Output pattern 26:
1 1 1 1 0
1 0 0 0 1
1 0 0 0 1
1 1 1 1 0
1 0 0 0 0
1 0 0 0 0
1 0 0 0 0
# Input pattern 27:
0 1 1 1 0
1 0 0 0 1
1 0 0 1 1
1 0 0 0 1
1 0 1 0 1
1 0 0 1 0
0 1 1 0 1
# Output pattern 27:
0 1 1 1 0
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
1 0 1 0 1
1 0 0 1 0
0 1 1 0 1
# Input pattern 28:
1 1 1 1 0
1 0 0 0 1
1 0 1 0 1
1 1 1 1 0
1 0 1 0 0
1 0 0 1 0
1 0 0 0 1
# Output pattern 28:
1 1 1 1 0
1 0 0 0 1
1 0 0 0 1
1 1 1 1 0
1 0 1 0 0
1 0 0 1 0
1 0 0 0 1
# Input pattern 29:
0 1 1 1 1
1 0 0 1 0
1 0 0 0 0
0 1 1 1 0

```

```

0 0 0 0 1
0 0 0 0 1
1 1 1 1 0
# Output pattern 29:
0 1 1 1 1
1 0 0 0 0
1 0 0 0 0
0 1 1 1 0
0 0 0 0 1
0 0 0 0 1
1 1 1 1 0
# Input pattern 30:
1 1 1 1 1
0 0 1 0 0
0 0 1 0 0
0 0 1 0 0
0 1 1 0 0
0 0 1 0 0
0 0 1 0 0
# Output pattern 30:
1 1 1 1 1
0 0 1 0 0
0 0 1 0 0
0 0 1 0 0
0 0 1 0 0
0 0 1 0 0
0 0 1 0 0
# Input pattern 31:
1 0 0 0 1
1 0 0 0 1
1 0 0 1 1
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
0 1 1 1 0
# Output pattern 31:
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
0 1 1 1 0
# Input pattern 32:
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
1 0 0 1 1
1 0 0 0 1
0 1 0 1 0
0 0 1 0 0

```

# Output pattern 32:

1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
0 1 0 1 0  
0 0 1 0 0

# Input pattern 33:

1 0 0 0 1  
1 0 0 0 1  
1 0 0 1 1  
1 0 0 0 1  
1 0 1 0 1  
1 1 0 1 1  
1 0 0 0 1

# Output pattern 33:

1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 1 0 1  
1 1 0 1 1  
1 0 0 0 1

# Input pattern 34:

1 0 0 0 1  
1 0 0 0 1  
0 1 0 1 0  
0 0 1 0 0  
0 1 0 1 0  
1 0 0 0 1  
1 0 0 1 1

# Output pattern 34:

1 0 0 0 1  
1 0 0 0 1  
0 1 0 1 0  
0 0 1 0 0  
0 1 0 1 0  
1 0 0 0 1  
1 0 0 0 1

# Input pattern 35:

1 1 0 0 1  
1 0 0 0 1  
0 1 0 1 0  
0 0 1 0 0  
0 0 1 0 0  
0 0 1 0 0  
0 0 1 0 0

# Output pattern 35:

1 0 0 0 1  
1 0 0 0 1

0 1 0 1 0  
0 0 1 0 0  
0 0 1 0 0  
0 0 1 0 0  
0 0 1 0 0

# Input pattern 36:

1 1 1 1 1  
0 0 0 0 1  
0 0 0 1 0  
0 0 1 0 0  
0 1 0 1 0  
1 0 0 0 0  
1 1 1 1 1

# Output pattern 36:

1 1 1 1 1  
0 0 0 0 1  
0 0 0 1 0  
0 0 1 0 0  
0 1 0 0 0  
1 0 0 0 0  
1 1 1 1 1

Patrón salida posición

SNNS pattern definition file V3.2

generated at Fri Apr 22 15:24:52 1994

No. of patterns : 72

No. of input units : 35

No. of output units : 36

# Input pattern 1:

1 1 1 1 1  
1 0 0 1 1  
1 0 1 0 1  
1 0 1 0 1  
1 0 1 0 1  
1 0 1 0 1  
1 1 0 0 1  
1 1 1 1 1

# Output pattern 1:

1 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0

# Input pattern 2:

0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1

0 0 0 0 1  
0 0 0 0 1  
# Output pattern 2:  
0 1 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0

# Input pattern 3:  
1 1 1 1 1  
0 0 0 0 1  
0 0 0 0 1  
1 1 1 1 1  
1 0 0 0 0  
1 0 0 0 0  
1 1 1 1 1

# Output pattern 3:  
0 0 1 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0

# Input pattern 4:  
1 1 1 1 1  
0 0 0 0 1  
0 0 0 0 1  
1 1 1 1 1  
0 0 0 0 1  
0 0 0 0 1  
1 1 1 1 1

# Output pattern 4:  
0 0 0 1 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0

# Input pattern 5:  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 1 1 1 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1

# Output pattern 5:

0 0 0 0 1 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0

# Input pattern 6:

1 1 1 1 1  
1 0 0 0 0  
1 0 0 0 0  
1 1 1 1 1  
0 0 0 0 1  
0 0 0 0 1  
1 1 1 1 1

# Output pattern 6:

0 0 0 0 0 1  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0

# Input pattern 7:

1 1 1 1 1  
1 0 0 0 0  
1 0 0 0 0  
1 1 1 1 1  
1 0 0 0 1  
1 0 0 0 1  
1 1 1 1 1

# Output pattern 7:

0 0 0 0 0 0  
1 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0

# Input pattern 8:

1 1 1 1 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1

# Output pattern 8:

0 0 0 0 0 0  
0 1 0 0 0 0  
0 0 0 0 0 0

0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0

# Input pattern 9:

1 1 1 1 1  
1 0 0 0 1  
1 0 0 0 1  
1 1 1 1 1  
1 0 0 0 1  
1 0 0 0 1  
1 1 1 1 1

# Output pattern 9:

0 0 0 0 0 0  
0 0 1 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0

# Input pattern 10:

1 1 1 1 1  
1 0 0 0 1  
1 0 0 0 1  
1 1 1 1 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1

# Output pattern 10:

0 0 0 0 0 0  
0 0 0 1 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0

# Input pattern 11:

0 0 1 0 0  
0 1 0 1 0  
1 0 0 0 1  
1 1 1 1 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1

# Output pattern 11:

0 0 0 0 0 0  
0 0 0 0 1 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0



#Input pattern 12:

1 1 1 1 0  
1 0 0 0 1  
1 0 0 0 1  
1 1 1 1 0  
1 0 0 0 1  
1 0 0 0 1  
1 1 1 1 0

# Output pattern 12:

0 0 0 0 0 0  
0 0 0 0 0 1  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0

#Input pattern 13:

0 1 1 1 1  
1 0 0 0 0  
1 0 0 0 0  
1 0 0 0 0  
1 0 0 0 0  
1 0 0 0 0  
0 1 1 1 1

# Output pattern 13:

0 0 0 0 0 0  
0 0 0 0 0 0  
1 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0

#Input pattern 14:

1 1 1 0 0  
1 0 0 1 0  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 1 0  
1 1 1 0 0

# Output pattern 14:

0 0 0 0 0 0  
0 0 0 0 0 0  
0 1 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0

#Input pattern 15:

1 1 1 1 1

```
1 0 0 0 0
1 0 0 0 0
1 1 1 1 1
1 0 0 0 0
1 0 0 0 0
1 1 1 1 1
# Output pattern 15:
0 0 0 0 0 0
0 0 0 0 0 0
0 0 1 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
```

```
#Input pattern 16:
1 1 1 1 1
1 0 0 0 0
1 0 0 0 0
1 1 1 1 1
1 0 0 0 0
1 0 0 0 0
1 0 0 0 0
# Output pattern 16:
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 1 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
```

```
#Input pattern 17:
0 1 1 1 1
1 0 0 0 0
1 0 0 0 0
1 0 1 1 1
1 0 0 0 1
1 0 0 0 1
0 1 1 1 0
# Output pattern 17:
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 1 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
```

```
#Input pattern 18:
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
1 1 1 1 1
```

1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
# Output pattern 18:  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 1  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0

#Input pattern 19:  
0 1 1 1 0  
0 0 1 0 0  
0 0 1 0 0  
0 0 1 0 0  
0 0 1 0 0  
0 0 1 0 0  
0 0 1 0 0  
0 1 1 1 0

# Output pattern 19:  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
1 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0

#Input pattern 20:  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1  
1 0 0 0 1  
0 1 1 1 0

# Output pattern 20:  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 1 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0

#Input pattern 21:  
1 0 0 0 1  
1 0 0 1 0  
1 0 1 0 0  
1 1 0 0 0  
1 0 1 0 0  
1 0 0 1 0  
1 0 0 0 1

# Output pattern 21:

0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 1 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0

#Input pattern 22:

1 0 0 0 0  
1 0 0 0 0  
1 0 0 0 0  
1 0 0 0 0  
1 0 0 0 0  
1 0 0 0 0  
1 0 0 0 0  
1 1 1 1 1

# Output pattern 22:

0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 1 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0

#Input pattern 23:

1 0 0 0 1  
1 1 0 1 1  
1 0 1 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1

# Output pattern 23:

0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 1 0  
0 0 0 0 0 0  
0 0 0 0 0 0

#Input pattern 24:

1 0 0 0 1  
1 1 0 0 1  
1 1 1 0 1  
1 0 1 1 1  
1 0 0 1 1  
1 0 0 0 1  
1 0 0 0 1

# Output pattern 24:

0 0 0 0 0 0  
0 0 0 0 0 0

0 0 0 0 0 0  
0 0 0 0 0 1  
0 0 0 0 0 0  
0 0 0 0 0 0

#Input pattern 25:

0 1 1 1 0  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
0 1 1 1 0

# Output pattern 25:

0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
1 0 0 0 0 0  
0 0 0 0 0 0

#Input pattern 26:

1 1 1 1 0  
1 0 0 0 1  
1 0 0 0 1  
1 1 1 1 0  
1 0 0 0 0  
1 0 0 0 0  
1 0 0 0 0

# Output pattern 26:

0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 1 0 0 0 0  
0 0 0 0 0 0

#Input pattern 27:

0 1 1 1 0  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 1 0 1  
1 0 0 1 0  
0 1 1 0 1

# Output pattern 27:

0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 1 0 0 0

0 0 0 0 0 0

#Input pattern 28:

1 1 1 1 0

1 0 0 0 1

1 0 0 0 1

1 1 1 1 0

1 0 1 0 0

1 0 0 1 0

1 0 0 0 1

# Output pattern 28:

0 0 0 0 0 0

0 0 0 0 0 0

0 0 0 0 0 0

0 0 0 0 0 0

0 0 0 1 0 0

0 0 0 0 0 0

#Input pattern 29:

0 1 1 1 1

1 0 0 0 0

1 0 0 0 0

0 1 1 1 0

0 0 0 0 1

0 0 0 0 1

1 1 1 1 0

# Output pattern 29:

0 0 0 0 0 0

0 0 0 0 0 0

0 0 0 0 0 0

0 0 0 0 0 0

0 0 0 0 1 0

0 0 0 0 0 0

#Input pattern 30:

1 1 1 1 1

0 0 1 0 0

0 0 1 0 0

0 0 1 0 0

0 0 1 0 0

0 0 1 0 0

0 0 1 0 0

# Output pattern 30:

0 0 0 0 0 0

0 0 0 0 0 0

0 0 0 0 0 0

0 0 0 0 0 0

0 0 0 0 0 1

0 0 0 0 0 0

#Input pattern 31:

1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
0 1 1 1 0  
# Output pattern 31:  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
1 0 0 0 0 0

#Input pattern 32:  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
0 1 0 1 0  
0 0 1 0 0  
# Output pattern 32:  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 1 0 0 0 0

#Input pattern 33:  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 1 0 1  
1 1 0 1 1  
1 0 0 0 1  
# Output pattern 33:  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 1 0 0 0

#Input pattern 34:  
1 0 0 0 1  
1 0 0 0 1  
0 1 0 1 0

0 0 1 0 0  
0 1 0 1 0  
1 0 0 0 1  
1 0 0 0 1  
# Output pattern 34:  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 1 0 0

#Input pattern 35:  
1 0 0 0 1  
1 0 0 0 1  
0 1 0 1 0  
0 0 1 0 0  
0 0 1 0 0  
0 0 1 0 0  
0 0 1 0 0

# Output pattern 35:  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 1 0

#Input pattern 36:  
1 1 1 1 1  
0 0 0 0 1  
0 0 0 1 0  
0 0 1 0 0  
0 1 0 0 0  
1 0 0 0 0  
1 1 1 1 1

# Output pattern 36:  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 1

# Input pattern 1:  
1 1 1 1 1  
1 0 0 1 1  
1 1 1 0 1



1 0 1 0 1  
1 0 1 0 1  
1 1 0 0 1  
1 1 1 1 1  
# Output pattern 1:  
1 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0

# Input pattern 2:  
0 0 0 0 1  
0 0 0 0 1  
0 0 1 0 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1

# Output pattern 2:  
0 1 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0

# Input pattern 3:  
1 1 1 1 1  
0 0 1 0 1  
0 0 0 0 1  
1 1 1 1 1  
1 0 0 0 0  
1 0 0 0 0  
1 1 1 1 1

# Output pattern 3:  
0 0 1 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0

# Input pattern 4:  
1 1 1 1 1  
0 0 0 0 1  
0 0 0 0 1  
1 1 1 1 1  
0 0 1 0 1  
0 0 0 0 1

1 1 1 1 1  
# Output pattern 4:  
0 0 0 1 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0

# Input pattern 5:  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 1 1 1 1  
0 0 0 0 1  
0 0 0 0 1  
0 1 0 0 1

# Output pattern 5:  
0 0 0 0 1 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0

# Input pattern 6:  
1 1 1 1 1  
1 0 0 0 0  
1 0 0 1 0  
1 1 1 1 1  
0 0 0 0 1  
0 0 0 0 1  
1 1 1 1 1

# Output pattern 6:  
0 0 0 0 0 1  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0

# Input pattern 7:  
1 1 1 1 1  
1 0 0 0 0  
1 0 0 1 0  
1 1 1 1 1  
1 0 0 0 1  
1 0 0 0 1  
1 1 1 1 1

# Output pattern 7:  
0 0 0 0 0 0

1 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0

# Input pattern 8:

1 1 1 1 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1  
1 0 0 0 1  
0 0 0 0 1

# Output pattern 8:

0 0 0 0 0 0  
0 1 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0

# Input pattern 9:

1 1 1 1 1  
1 0 0 0 1  
1 0 0 0 1  
1 1 1 1 1  
1 0 1 0 1  
1 0 0 0 1  
1 1 1 1 1

# Output pattern 9:

0 0 0 0 0 0  
0 0 1 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0

# Input pattern 10:

1 1 1 1 1  
1 0 0 0 1  
1 0 0 0 1  
1 1 1 1 1  
0 0 0 0 1  
0 1 0 0 1  
0 0 0 0 1

# Output pattern 10:

0 0 0 0 0 0  
0 0 0 1 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0

0 0 0 0 0 0  
0 0 0 0 0 0

# Input pattern 11:

0 0 1 0 0  
0 1 0 1 0  
1 0 0 0 1  
1 1 1 1 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 1 0 1

# Output pattern 11:

0 0 0 0 0 0  
0 0 0 0 1 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0

#Input pattern 12:

1 1 1 1 0  
1 0 0 0 1  
1 0 0 0 1  
1 1 1 1 0  
1 0 1 0 1  
1 0 0 0 1  
1 1 1 1 0

# Output pattern 12:

0 0 0 0 0 0  
0 0 0 0 0 1  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0

#Input pattern 13:

0 1 1 1 1  
1 0 0 0 0  
1 0 0 1 0  
1 0 0 0 0  
1 0 0 0 0  
1 0 0 0 0  
0 1 1 1 1

# Output pattern 13:

0 0 0 0 0 0  
0 0 0 0 0 0  
1 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0

#Input pattern 14:

1 1 1 0 0

1 0 0 1 0

1 0 0 0 1

1 0 1 0 1

1 0 0 0 1

1 0 0 1 0

1 1 1 0 0

# Output pattern 14:

0 0 0 0 0 0

0 0 0 0 0 0

0 1 0 0 0 0

0 0 0 0 0 0

0 0 0 0 0 0

0 0 0 0 0 0

#Input pattern 15:

1 1 1 1 1

1 0 0 0 0

1 0 1 0 0

1 1 1 1 1

1 0 0 0 0

1 0 0 0 0

1 1 1 1 1

# Output pattern 15:

0 0 0 0 0 0

0 0 0 0 0 0

0 0 1 0 0 0

0 0 0 0 0 0

0 0 0 0 0 0

0 0 0 0 0 0

#Input pattern 16:

1 1 1 1 1

1 0 0 1 0

1 0 0 0 0

1 1 1 1 1

1 0 0 0 0

1 0 0 0 0

1 0 0 0 0

# Output pattern 16:

0 0 0 0 0 0

0 0 0 0 0 0

0 0 0 1 0 0

0 0 0 0 0 0

0 0 0 0 0 0

0 0 0 0 0 0

#Input pattern 17:

0 1 1 1 1

1 0 0 0 0

1 0 0 0 1  
1 0 1 1 1  
1 0 0 0 1  
1 0 0 0 1  
0 1 1 1 0  
# Output pattern 17:  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 1 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0

#Input pattern 18:  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 1 1 1 1  
1 0 0 0 1  
1 0 1 0 1  
1 0 0 0 1

# Output pattern 18:  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 1  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0

#Input pattern 19:  
0 1 1 1 0  
0 0 1 0 0  
0 0 1 0 0  
0 0 1 0 0  
0 1 1 0 0  
0 0 1 0 0  
0 1 1 1 0

# Output pattern 19:  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
1 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0

#Input pattern 20:  
0 0 0 0 1  
0 0 0 0 1  
0 1 0 0 1  
0 0 0 0 1  
0 0 0 0 1

1 0 0 0 1  
0 1 1 1 0  
# Output pattern 20:  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 1 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0

#Input pattern 21:  
1 0 0 0 1  
1 0 0 1 0  
1 0 1 0 0  
1 1 0 0 0  
1 0 1 0 0  
1 0 0 1 0  
1 1 0 0 1

# Output pattern 21:  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 1 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0

#Input pattern 22:  
1 0 0 0 0  
1 0 0 0 0  
1 0 0 0 0  
1 0 0 0 1  
1 0 0 0 0  
1 0 0 0 0  
1 1 1 1 1

# Output pattern 22:  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 1 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0

#Input pattern 23:  
1 0 0 0 1  
1 1 0 1 1  
1 0 1 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 1 0 1  
1 0 0 0 1

# Output pattern 23:

0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 1 0  
0 0 0 0 0 0  
0 0 0 0 0 0

#Input pattern 24:

1 0 0 1 1  
1 1 0 0 1  
1 1 1 0 1  
1 0 1 1 1  
1 0 0 1 1  
1 0 0 0 1  
1 0 0 0 1

# Output pattern 24:

0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 1  
0 0 0 0 0 0  
0 0 0 0 0 0

#Input pattern 25:

0 1 1 1 0  
1 0 0 0 1  
1 0 0 1 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
0 1 1 1 0

# Output pattern 25:

0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
1 0 0 0 0 0  
0 0 0 0 0 0

#Input pattern 26:

1 1 1 1 0  
1 0 0 0 1  
1 0 0 0 1  
1 1 1 1 0  
1 0 0 0 0  
1 0 0 0 0  
1 0 0 1 0

# Output pattern 26:

0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0



0 0 0 0 0 0  
0 1 0 0 0 0  
0 0 0 0 0 0

#Input pattern 27:

0 1 1 1 0  
1 0 0 0 1  
1 0 0 1 1  
1 0 0 0 1  
1 0 1 0 1  
1 0 0 1 0  
0 1 1 0 1

# Output pattern 27:

0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 1 0 0 0  
0 0 0 0 0 0

#Input pattern 28:

1 1 1 1 0  
1 0 0 0 1  
1 0 1 0 1  
1 1 1 1 0  
1 0 1 0 0  
1 0 0 1 0  
1 0 0 0 1

# Output pattern 28:

0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 1 0 0  
0 0 0 0 0 0

#Input pattern 29:

0 1 1 1 1  
1 0 0 1 0  
1 0 0 0 0  
0 1 1 1 0  
0 0 0 0 1  
0 0 0 0 1  
1 1 1 1 0

# Output pattern 29:

0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 1 0  
0 0 0 0 0 0

#Input pattern 30:

1 1 1 1 1  
0 0 1 0 0  
0 0 1 0 0  
0 0 1 0 0  
0 1 1 0 0  
0 0 1 0 0  
0 0 1 0 0

# Output pattern 30:

0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 1  
0 0 0 0 0 0

#Input pattern 31:

1 0 0 0 1  
1 0 0 0 1  
1 0 0 1 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
0 1 1 1 0

# Output pattern 31:

0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
1 0 0 0 0 0

#Input pattern 32:

1 0 0 0 1  
1 0 0 0 1  
1 0 0 0 1  
1 0 0 1 1  
1 0 0 0 1  
0 1 0 1 0  
0 0 1 0 0

# Output pattern 32:

0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 1 0 0 0 0

#Input pattern 33:

1 0 0 0 1

1 0 0 0 1  
1 0 0 1 1  
1 0 0 0 1  
1 0 1 0 1  
1 1 0 1 1  
1 0 0 0 1  
# Output pattern 33:  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 1 0 0 0

#Input pattern 34:  
1 0 0 0 1  
1 0 0 0 1  
0 1 0 1 0  
0 0 1 0 0  
0 1 0 1 0  
1 0 0 0 1  
1 0 0 1 1  
# Output pattern 34:  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 1 0 0

#Input pattern 35:  
1 1 0 0 1  
1 0 0 0 1  
0 1 0 1 0  
0 0 1 0 0  
0 0 1 0 0  
0 0 1 0 0  
0 0 1 0 0  
# Output pattern 35:  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 1 0

#Input pattern 36:  
1 1 1 1 1  
0 0 0 0 1  
0 0 0 1 0  
0 0 1 0 0

0 1 0 1 0  
1 0 0 0 0  
1 1 1 1 1  
# Output pattern 36:  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 1

Patrón de validación  
SNNS pattern definition file V3.2  
generated at Fri Apr 22 15:24:52 1994

No. of patterns : 36  
No. of input units : 35  
No. of output units : 8

# Input pattern 1:  
1 1 1 1 1  
1 0 0 1 1  
1 0 1 0 1  
1 1 1 0 1  
1 0 1 0 1  
1 1 0 0 1  
1 1 1 1 1

# Output pattern 1:  
0 0 1 1 0 0 0 0

# Input pattern 2:  
0 0 0 0 1  
0 0 0 0 1  
0 1 0 0 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1  
0 0 0 0 1

# Output pattern 2:  
0 0 1 1 0 0 0 1

# Input pattern 3:  
1 1 1 1 1  
0 0 0 1 1  
0 0 0 0 1  
1 1 1 1 1  
1 0 0 0 0  
1 0 0 0 0  
1 1 1 1 1

# Output pattern 3:  
0 0 1 1 0 0 1 0

# Input pattern 4:  
1 1 1 1 1

```

0 0 0 0 1
0 0 0 0 1
1 1 1 1 1
0 1 0 0 1
0 0 0 0 1
1 1 1 1 1
# Output pattern 4:
0 0 1 1 0 0 1 1
# Input pattern 5:
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
1 1 1 1 1
0 0 0 0 1
0 0 0 0 1
1 0 0 0 1
# Output pattern 5:
0 0 1 1 0 1 0 0
# Input pattern 6:
1 1 1 1 1
1 0 0 0 0
1 0 1 0 0
1 1 1 1 1
0 0 0 0 1
0 0 0 0 1
1 1 1 1 1
# Output pattern 6:
0 0 1 1 0 1 0 1
# Input pattern 7:
1 1 1 1 1
1 0 0 0 0
1 0 1 0 0
1 1 1 1 1
1 0 0 0 1
1 0 0 0 1
1 1 1 1 1
# Output pattern 7:
0 0 1 1 0 1 1 0
# Input pattern 8:
1 1 1 1 1
0 0 0 0 1
0 0 0 0 1
0 0 0 0 1
0 0 0 0 1
0 1 0 0 1
0 0 0 0 1
# Output pattern 8:
0 0 1 1 0 1 1 1
# Input pattern 9:
1 1 1 1 1
1 0 0 0 1

```

```

1 0 0 0 1
1 1 1 1 1
1 1 0 0 1
1 0 0 0 1
1 1 1 1 1
# Output pattern 9:
0 0 1 1 1 0 0 0
# Input pattern 10:
1 1 1 1 1
1 0 0 0 1
1 0 0 0 1
1 1 1 1 1
0 0 0 0 1
0 0 1 0 1
0 0 0 0 1
# Output pattern 10:
0 0 1 1 1 0 0 1
# Input pattern 11:
0 0 1 0 0
0 1 0 1 0
1 0 0 0 1
1 1 1 1 1
1 0 0 0 1
1 0 1 0 1
1 0 0 0 1
# Output pattern 11:
0 1 0 0 0 0 0 1
# Input pattern 12:
1 1 1 1 0
1 0 0 0 1
1 0 0 0 1
1 1 1 1 0
1 0 1 0 1
1 0 0 0 1
1 1 1 1 0
# Output pattern 12:
0 1 0 0 0 0 1 0
# Input pattern 13:
0 1 1 1 1
1 0 0 0 0
1 0 0 0 1
1 0 0 0 0
1 0 0 0 0
1 0 0 0 0
0 1 1 1 1
# Output pattern 13:
0 1 0 0 0 0 1 1
# Input pattern 14:
1 1 1 0 0
1 0 0 1 0
1 0 0 0 1

```

```

1 0 1 0 1
1 0 0 0 1
1 0 0 1 0
1 1 1 0 0
# Output pattern 14:
0 1 0 0 0 1 0 0
# Input pattern 15:
1 1 1 1 1
1 0 0 0 0
1 0 1 0 0
1 1 1 1 1
1 0 0 0 0
1 0 0 0 0
1 1 1 1 1
# Output pattern 15:
0 1 0 0 0 1 0 1
# Input pattern 16:
1 1 1 1 1
1 0 0 0 1
1 0 0 0 0
1 1 1 1 1
1 0 0 0 0
1 0 0 0 0
1 0 0 0 0
# Output pattern 16:
0 1 0 0 0 1 1 0
# Input pattern 17:
0 1 1 1 1
1 0 0 0 0
1 0 0 1 0
1 0 1 1 1
1 0 0 0 1
1 0 0 0 1
0 1 1 1 0
# Output pattern 17:
0 1 0 0 0 1 1 1
# Input pattern 18:
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
1 1 1 1 1
1 0 0 0 1
1 0 1 0 1
1 0 0 0 1
# Output pattern 18:
0 1 0 0 1 0 0 0
# Input pattern 19:
0 1 1 1 0
0 0 1 0 0
0 0 1 0 0
0 0 1 0 0

```

```

1 0 1 0 0
0 0 1 0 0
0 1 1 1 0
# Output pattern 19:
0 1 0 0 1 0 0 1
# Input pattern 20:
0 0 0 0 1
0 0 0 0 1
0 0 1 0 1
0 0 0 0 1
0 0 0 0 1
1 0 0 0 1
0 1 1 1 0
# Output pattern 20:
0 1 0 0 1 0 1 0
# Input pattern 21:
1 0 0 0 1
1 0 0 1 0
1 0 1 0 0
1 1 0 0 0
1 0 1 0 0
1 0 0 1 0
1 0 1 0 1
# Output pattern 21:
0 1 0 0 1 0 1 1
# Input pattern 22:
1 0 0 0 0
1 0 0 0 0
1 0 0 0 0
1 0 0 1 0
1 0 0 0 0
1 0 0 0 0
1 1 1 1 1
# Output pattern 22:
0 1 0 0 1 1 0 0
# Input pattern 23:
1 0 0 0 1
1 1 0 1 1
1 0 1 0 1
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
1 0 1 0 1
# Output pattern 23:
0 1 0 0 1 1 0 1
# Input pattern 24:
1 0 1 0 1
1 1 0 0 1
1 1 1 0 1
1 0 1 1 1
1 0 0 1 1

```



```

1 0 0 0 1
1 0 0 0 1
# Output pattern 24:
0 1 0 0 1 1 1 0
# Input pattern 25:
0 1 1 1 0
1 0 0 0 1
1 0 1 0 1
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
0 1 1 1 0
# Output pattern 25:
0 1 0 0 1 1 1 1
# Input pattern 26:
1 1 1 1 0
1 0 0 0 1
1 0 0 0 1
1 1 1 1 0
1 0 0 0 0
1 0 0 0 0
1 0 1 0 0
# Output pattern 26:
0 1 0 1 0 0 0 0
# Input pattern 27:
0 1 1 1 0
1 0 0 0 1
1 0 1 0 1
1 0 0 0 1
1 0 1 0 1
1 0 0 1 0
0 1 1 0 1
# Output pattern 27:
0 1 0 1 0 0 0 1
# Input pattern 28:
1 1 1 1 0
1 0 0 0 1
1 0 0 1 1
1 1 1 1 0
1 0 1 0 0
1 0 0 1 0
1 0 0 0 1
# Output pattern 28:
0 1 0 1 0 0 1 0
# Input pattern 29:
0 1 1 1 1
1 0 0 0 1
1 0 0 0 0
0 1 1 1 0
0 0 0 0 1
0 0 0 0 1

```

```

1 1 1 1 0
# Output pattern 29:
0 1 0 1 0 0 1 1
# Input pattern 30:
1 1 1 1 1
0 0 1 0 0
0 0 1 0 0
0 0 1 0 0
1 0 1 0 0
0 0 1 0 0
0 0 1 0 0
# Output pattern 30:
0 1 0 1 0 1 0 0
# Input pattern 31:
1 0 0 0 1
1 0 0 0 1
1 0 1 0 1
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
0 1 1 1 0
# Output pattern 31:
0 1 0 1 0 1 0 1
# Input pattern 32:
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
1 0 1 0 1
1 0 0 0 1
0 1 0 1 0
0 0 1 0 0
# Output pattern 32:
0 1 0 1 0 1 1 0
# Input pattern 33:
1 0 0 0 1
1 0 0 0 1
1 0 1 0 1
1 0 0 0 1
1 0 1 0 1
1 1 0 1 1
1 0 0 0 1
# Output pattern 33:
0 1 0 1 0 1 1 1
# Input pattern 34:
1 0 0 0 1
1 0 0 0 1
0 1 0 1 0
0 0 1 0 0
0 1 0 1 0
1 0 0 0 1
1 0 1 0 1

```

# Output pattern 34:

0 1 0 1 1 0 0 0

# Input pattern 35:

1 0 1 0 1

1 0 0 0 1

0 1 0 1 0

0 0 1 0 0

0 0 1 0 0

0 0 1 0 0

0 0 1 0 0

# Output pattern 35:

0 1 0 1 1 0 0 1

# Input pattern 36:

1 1 1 1 1

0 0 0 0 1

0 0 0 1 0

0 0 1 0 0

0 1 0 0 1

1 0 0 0 0

1 1 1 1 1

# Output pattern 36:

0 1 0 1 1 0 1 0