

Redes de Hopfield

Sistemas de Inteligencia Artificial - ITBA

Carlos Sessa

Lucas Pizzagalli

Nicolás Purita

Introducción

Se implementa una *Red de Hopfield* con memoria asociativa direccionada por el contenido, con una aplicación de las reglas de actualización de forma sincrónica. Esto quiere decir que se actualizan todas las neuronas en simultáneo en cada paso. La red memoriza un subconjunto Ψ de los patrones que se pueden observar en la Figura 8.

Luego, se comprueba que la red haya aprendido correctamente, y que los patrones utilizados sean atractores. Finalmente, se analiza como se comporta la red ante la utilización de patrones de entradas distintos tipos de imágenes.

Desarrollo

Como punto de partida para crear la *Red de Hopfield*, se crean funciones en *Matlab* para simular el comportamiento de la red. Se crean redes con distintos conjuntos de patrones con el fin de analizar y comprender el funcionamiento de la misma. El criterio de selección de las imágenes es de acuerdo a sus similitudes, con el fin de disminuir la correlación entre ellas.

Las imágenes poseen un tamaño de 64 x 64 px y se encuentran umbralizadas. Esto quiere decir que los únicos colores que tiene la imagen son 0, negro en RGB, o 255 blanco en RGB, por lo tanto en todos los canales en la posición (i,j) posee los mismos valores. Dada esta característica de las imágenes, se puede tomar un único canal como patrón de entrada de la red. Es necesario hacer una diferenciación entre el negro y el blanco, por lo tanto se realiza una transformación en donde el 0 es mapeado al -1 y el 255 a 1.

Luego de poder cargar las imágenes correctamente, se crean métodos para poder componer la matriz de pesos con las imágenes que se desean aprender. Se debe tener mucho cuidado en la elección del conjunto de patrones que se desean memorizar ya que los mismos no deben solaparse en su mayoría, para poder ser aprendidos correctamente. Las imágenes son cargadas desde un directorio general donde se encuentran todas las imágenes de la Figura 8.

Resultados

Se construyen *Redes de Hopfield* con distintos patrones de entrada para hacer distintos análisis de los resultados.

Conjunto de letras

Para comenzar a estudiar la *Red de Hopfield* se toma el conjunto de imágenes que poseen únicamente letras, como se puede ver en la figura 1.

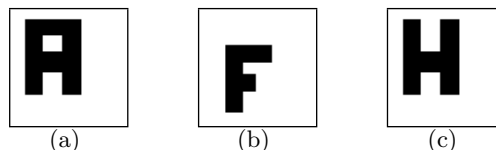


Figura 1: Conjunto de patrones de letras

Se observa que estos patrones **no** son verdaderos **atractores** ya que al presentar el conjunto de patrones a la red una vez entrenada, se obtiene la salida que se muestra en la figura 2. En la misma, se ve como la letra A fue confundida con la H. Para analizar este comportamiento se estudio el *crosstalk* de las figuras, obteniendo como resultado de 1(a) 1.3188, de 1(b) 0.8135 y de 1(c) 1.3774. Al existir valores mayores a 1 se puede concluir que estos efectos indeseados son correctos.

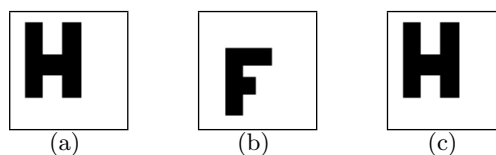


Figura 2: Conjunto de patrones de letras en la salida

Una vez estudiado este comportamiento de la red se decide modificar la imagen de la figura 1(a) para obtener el comportamiento deseado de la red que es que actúe como una memoria direccionable por el contenido. Por lo tanto, se hace una modificación en la figura 1(a) desplazando hacia la derecha, como se muestra en la figura 3. De esta forma disminuye la correlación entre los patrones por lo tanto el *crosstalk* máximo total es de 0.7705, el cual cumple con la condición de ser menor que 1 siendo un punto estable y obteniendo, en la salida, los mismos patrones que en las entradas.



Figura 3: Patrón de la figura 1(a) desplazado a la derecha

Para verificar si existen estados espúreos de *Primera clase* se vuelve a utilizar la red con los patrones de entrenamiento de la figura 1 con la salvedad de que se reemplaza la figura 1(a) por la figura 3. Luego se presentan los patrones opuestos y se observa que llegaron a los estados espúreos deseados, los cuales se muestran en la figura 4.

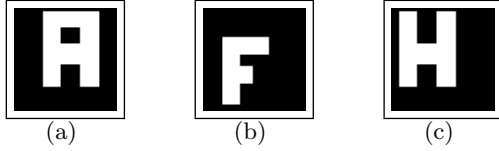


Figura 4: Resultado de presentar los patrones inversos a los utilizados para crearla

Comparación de patrones

Con el objetivo de poder realizar mejores comparaciones, se decide tomar tres conjuntos de patrones, Ψ_1 compuesto por las imágenes de la figura 5, Ψ_2 compuesto por los patrones de la figura 6 y Ψ_3 como se ve en la figura 7. Luego de analizar los tres conjuntos, se obtiene que Ψ_1 posee un *crosstalk* máximo de 0.9893, Ψ_2 de 1.0552 y Ψ_3 de 1.729. Por lo que se asume que el primer conjunto es el único que tiene todos sus atractores estables.

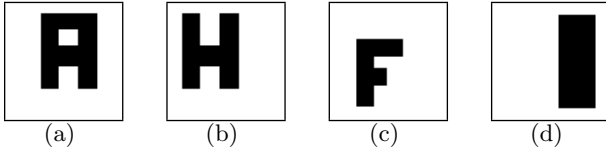


Figura 5: Patrones de entrenamiento Ψ_1

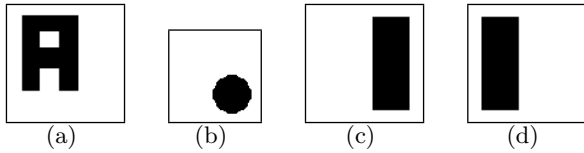


Figura 6: Patrones de entrenamiento Ψ_2

Presentando en cada red, sus respectivos patrones memorizados, se puede observar en la figura 9 que Ψ_1 logra la salida deseada mientras que Ψ_2 logra una salida parecida a los patrones de entrada como se puede ver en la figura 10, aunque como ya se sabe Ψ_2 posee un *crosstalk* mayor a 1, por lo que se sabe que no es un buen atractor. Por otro lado, en la figura 11, se puede ver como Ψ_3 muestra los patrones desdibujados, y no puede considerarse un atractor estable.

Versiones inversas de los patrones aprendidos

Para verificar si existen estados espúreos de *primera clase* se presentan los patrones opuestos y se observa que la red Ψ_1

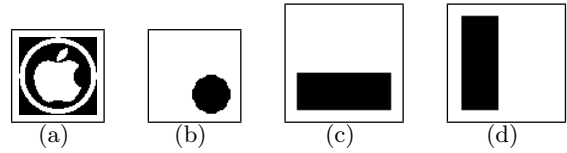


Figura 7: Patrones de entrenamiento Ψ_3

llegará a los estados espúreos deseados y nuevamente, Ψ_2 se encuentra muy cerca, los cuales se muestran en las figuras 12 y 13.

Nuevamente, la red entrenada con Ψ_3 no logra pasar la prueba, como se puede observar en la figura 14.

Versiones ruidosas

Para estudiar el desempeño de las redes, y cómo funcionan sus atractores, se decide presentar a cada una de las redes, una entrada con los patrones aprendidos distorsionados. Esto se logra al presentar a cada red ya entrenada, los mismos patrones con los que fueron entrenadas, pero con una cantidad de píxeles invertidos distribuidos en forma aleatoria por la imagen. Las imágenes generadas tienen un ruido que va desde el 10% hasta el 90%. Algunos ejemplos de las mismas se pueden observar en la primera columna de las figuras 15, 16 y 17, donde se alteraron el 10% de los píxeles, obteniendo igualmente una imagen legible al ojo humano. Mientras que en la segunda columna de las mismas se pueden observar las salidas ante dichas entradas.

A simple vista, se puede observar como la red entrenada con Ψ_1 puede reconocer todas las imágenes, hasta con un 40% de ruido (figura 18), mientras que si el ruido sube a más de 50%, la red tiende a catalogar la imagen de entrada, como si fuese su inversa, como se puede apreciar claramente en las figuras 22, 23 y 24, para los diferentes Ψ . Esto se debe, a que al existir los estados espúreos de primera clase (los cuales son inversos a los patrones originales, y poseen la misma energía), la entrada es atraída por los mismos (recordar que la inversa de una imagen, se logra al intercambiar todos los píxeles por su opuesto. En este caso, al haber intercambiado más de 50%, la imagen de entrada se parece más al inverso que a la original). Por otro lado, justo en el punto donde se modifica el 50% de las entradas, como en la figura 21 la neurona no es capaz de asociarla con ninguno de los atractores, ni siquiera utilizando el conjunto Ψ_1 . Esto se debe a que se encuentra equidistante entre el atractor original, y su estado espúreo de primera clase.

Por otro lado, se puede observar como las redes entrenadas con los otros dos conjuntos, y en especial la entrenada con Ψ_3 no logra identificar correctamente las imágenes. Esto se debe nuevamente a que no poseen patrones que sean buenos atractores.

Patrones que no pertenecen

Con el objetivo de analizar como reaccionan las redes al presentar patrones con los cuales no fueron entrenadas, se les presenta a las redes el resto de los patrones. Para mostrar el comportamiento obteniendo, se comparten los patrones

de la figura 25, en la que se puede observar en la primera fila, las imágenes que no fueron previamente aprendidas y en la segunda, los resultados obtenidos al ser estos patrones presentados a la red entrenada con Ψ_1 , la cual debe ser la de mejores resultados. En todos los casos se puede observar como las imágenes o bien son asociadas incorrectamente a imágenes ya aprendidas, como en el caso de las dos primeras imágenes, o no son reconocidas correctamente, sino que, son atraídas por estados espúreos de segunda clase, los cuales son estados mezcla como se puede observar en las cuatro imágenes de la derecha.

Máxima cantidad de patrones posibles a almacenar

Usando un método de actualización aleatorio y asincrónico, se sabe que la capacidad de una red de Hopfield es $0.15 * N$. Sin embargo, esto es verdad sólo cuando todos los patrones son ortogonales entre ellos. Se dice que un patrón binario es ortogonal con otro cuando el 50% o menos de los píxeles son iguales.

En nuestro caso particular sólo logramos agregar seis imágenes manteniendo la red estable. Se podría lograr agregar más imágenes modificándolas un poco. De la misma manera que se movió la letra A para que se aprenda bien, se podría intentar distribuir mejor las imágenes en los 64×64 píxeles para que cada los patrones sean ortogonales y así lograr agregar más patrones.

Conclusiones

Dado un conjunto pequeño de imágenes para entrenar la red, las cuales generan estados estables, la capacidad de reconocimiento de un patrón es bastante buena. La red puede usarse para reconocer patrones “contaminados” con ruido puntual. Sin embargo, notamos que la red no sirve para reconocer patrones afectados por otros tipos de ruido en los que la imagen se deforma topológicamente. Por ejemplo, si se corre 1 píxel a la derecha cada píxel de una imagen de testeo, la capacidad de reconocimiento de la red se vuelve muy mala. Esto se debe a que la red se basa en que dos patrones son similares si la distancia bit a bit entre ambos es pequeña. Por eso, un patrón exactamente igual a otro pero corrido 1 píxel a la derecha es muy diferente al original y puede no ser reconocido por la red.

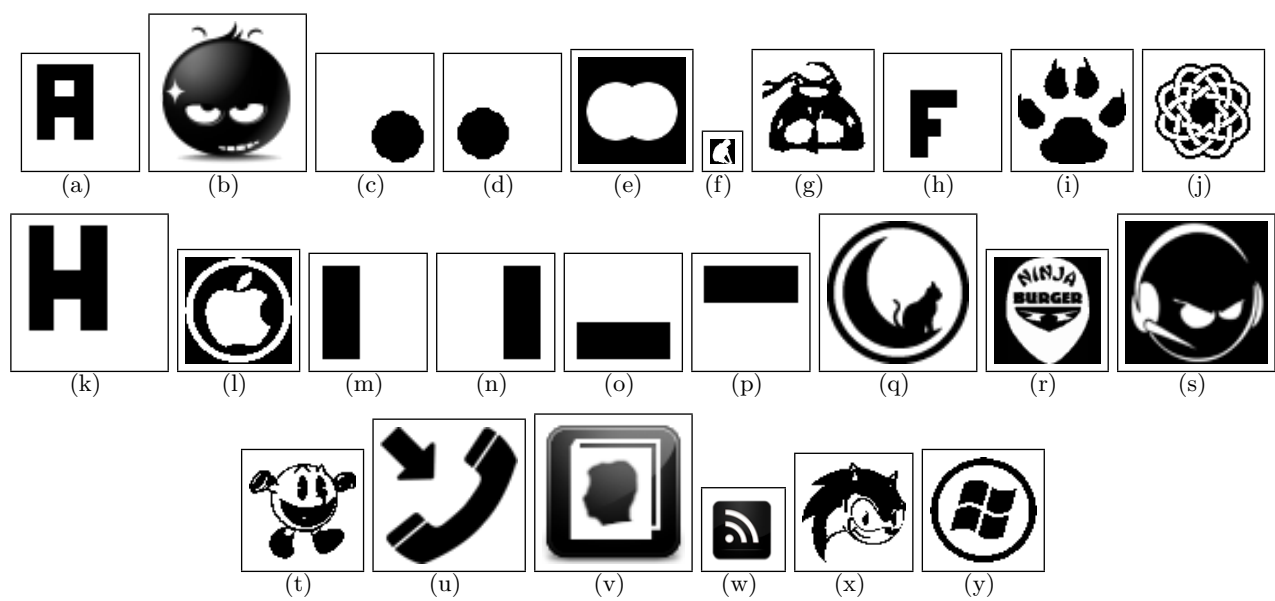


Figura 8: Patrones sobre los que se trabaja

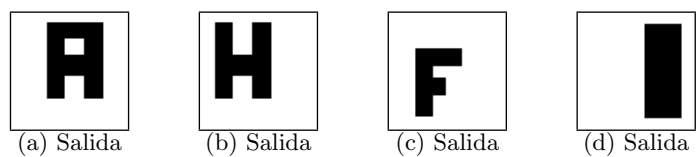


Figura 9: Salida de patrones de entrenamiento Ψ_1

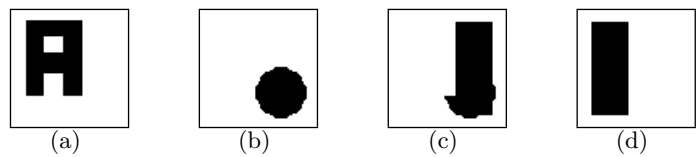


Figura 10: Salida de patrones de entrenamiento Ψ_2

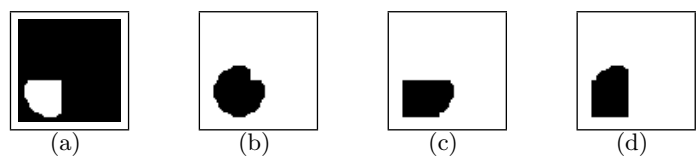


Figura 11: Salida de patrones de entrenamiento Ψ_3

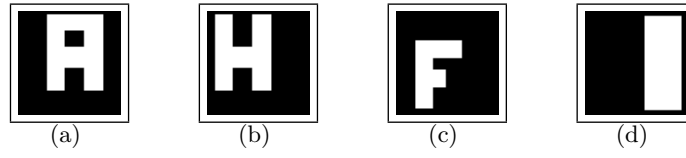


Figura 12: Salida de patrones invertidos Ψ_1

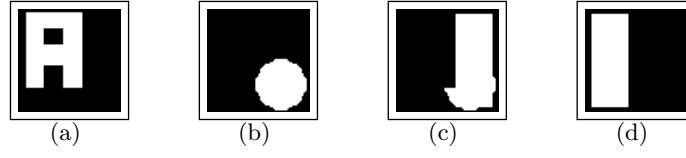


Figura 13: Salida de patrones invertidos Ψ_2

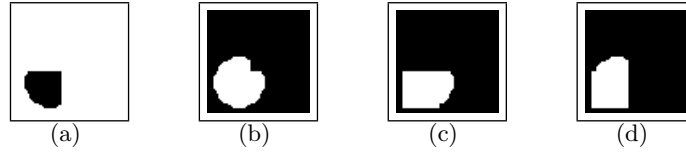


Figura 14: Salida de patrones invertidos Ψ_3

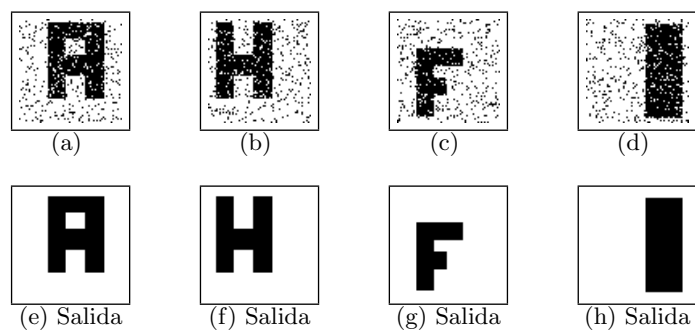


Figura 15: Entradas y sus respectivas salidas con ruido del 10% para Ψ_1

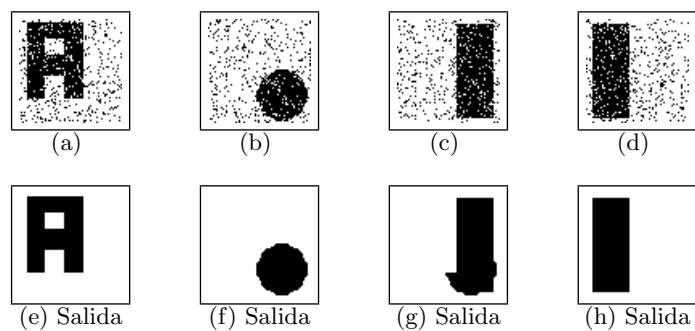


Figura 16: Entradas y sus respectivas salidas con ruido del 10% para Ψ_2

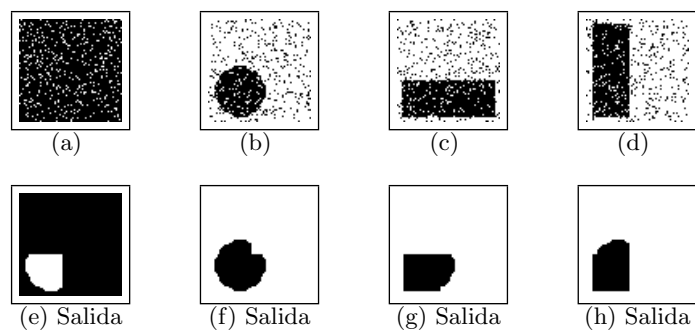


Figura 17: Entradas y sus respectivas salidas con ruido del 10% para Ψ_3

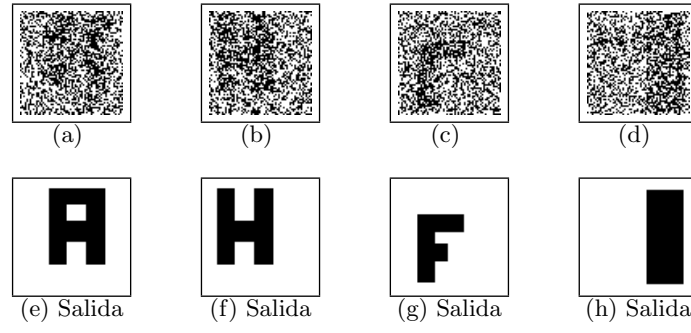


Figura 18: Entradas y sus respectivas salidas con ruido del 40% para Ψ_1

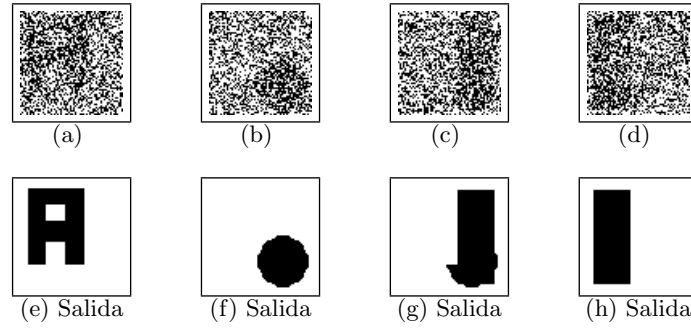


Figura 19: Entradas y sus respectivas salidas con ruido del 40% para Ψ_2

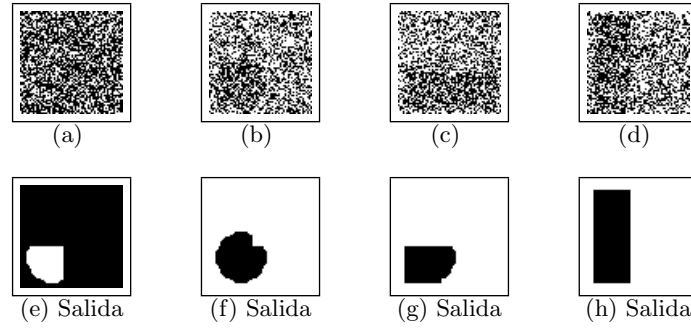


Figura 20: Entradas y sus respectivas salidas con ruido del 40% para Ψ_3

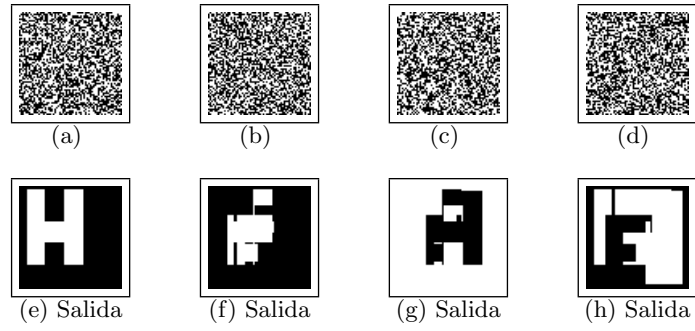


Figura 21: Entradas y sus respectivas salidas con ruido del 50% para Ψ_1

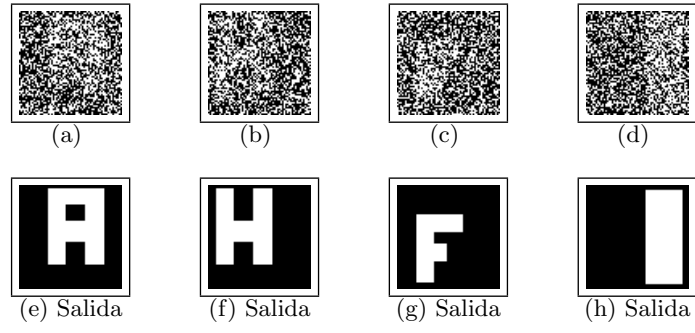


Figura 22: Entradas y sus respectivas salidas con ruido del 60% para Ψ_1

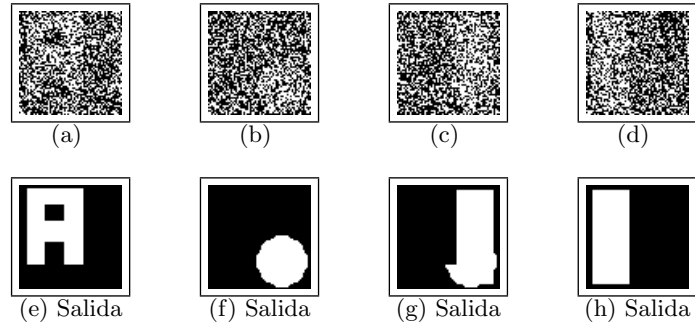


Figura 23: Entradas y sus respectivas salidas con ruido del 60% para Ψ_2

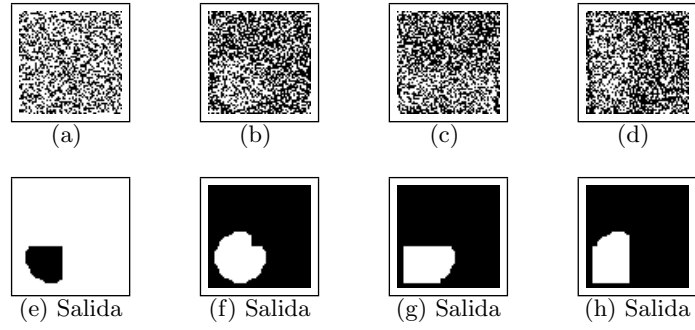


Figura 24: Entradas y sus respectivas salidas con ruido del 60% para Ψ_3

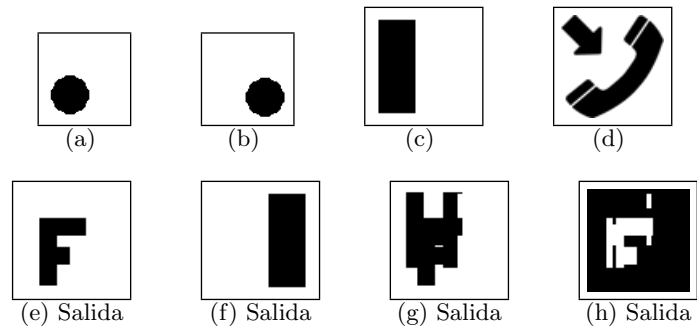


Figura 25: Entradas que no fueron aprendidas y sus respectivas salidas para Ψ_3