Del OCR a la RNA y más allá

1. **Partimos del OCR**
2. **Llegamos a la RNA**
3. **La RNA pasa a la acción.**
4. **RNA convolucionales.**
5. **Superando las imágenes MNIST**
6. **El problema es el contexto.**

Alberto Bañón Serrano, 15 de marzo de 2021

1. Partimos del OCR

En este documento vamos a transitar desde el reconocimiento óptico de caracteres (OCR) hasta las redes neuronales artificiales (RNA) como forma de aprender algo sobre estas últimas.

A partir de una imagen, el OCR, como su propio nombre indica, trata de discernir a que carácter corresponde. Originalmente los caracteres de interés eran las letras del alfabeto y los dígitos del sistema de numeración, pues se trataba de digitalizar documentos impresos en papel.

Las ideas son las mismas para cualquier colección de objetos: perros, gatos, rosas, sillas, mesas, etc., pero la tarea es más difícil cuanto más complejos son los objetos y más aún si hacemos intervenir las tres dimensiones.

Como no estamos interesados en el OCR en sí mismo nos vamos a centrar en los 10 dígitos del sistema de numeración decimal. Las aplicaciones para reconocer estos dígitos en una colección de imágenes es un clásico de la Inteligencia artificial, no hay curso, conferencia o tutorial que no empiece con un ejemplo de esta tarea.

Casi siempre se utiliza una colección de 70.000 imágenes, cada una con uno de los 10 dígitos dibujado a mano, se conoce como el conjunto MNIST y se puede descargar gratuitamente en Internet. Las imágenes son en blanco y negro, tienen el formato de 28x28 pixeles y cada píxel tiene un valor entre 0 y 255 que corresponde al color en ese punto, mejor dicho a la intensidad de gris: del negro (0) al blanco (255). Como vamos a tratar de probabilidades y estamos acostumbrados a que estas sean valores entre 0 y 1, lo primero que se hace es dividir todos los puntos por 255. En la figura 1 se muestran tres ejemplos de imágenes con el dígito 0 y en la 2 de otros dígitos.

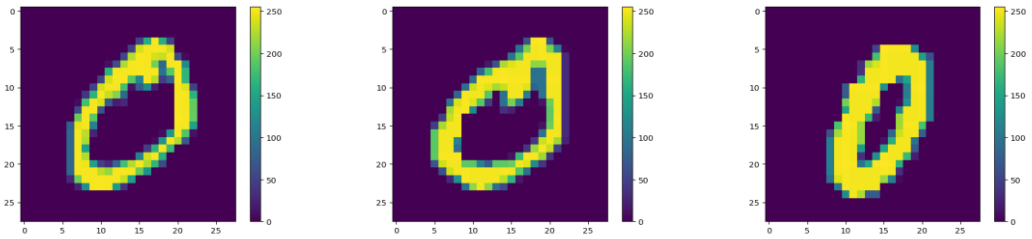

Figura 1. Ejemplos de imágenes con el dígito 0

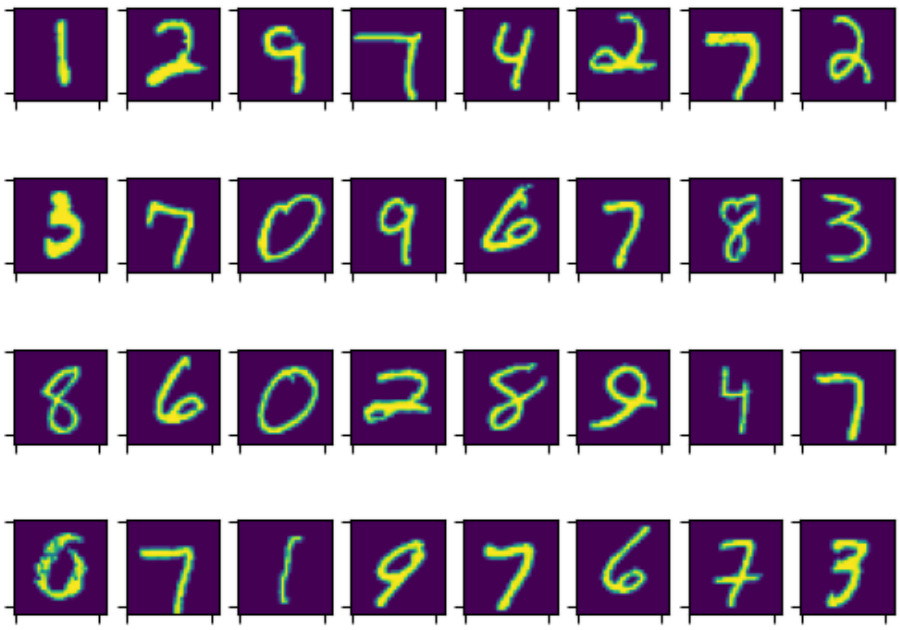

Figura 2. Ejemplos de imágenes del conjunto MNIST

Las 70.000 imágenes están distribuidas en dos grupos, el primero con 60.000 y el segundo con las 10.000 restantes. Al primer grupo nos referiremos como la muestra de entrenamiento y al segundo como la muestra de prueba.

Para nuestro análisis vamos a simplificar más aún el problema y nuestro objetivo se reducirá a distinguir las imágenes del dígito 0 de las restantes. Sólo nos interesa saber si una imagen es un 0 o si no lo es.

Se trata de buscar la esencia del 0, buscar aquello que hace semejantes a los 0 aunque estén dibujados a mano por personas distintas.

Una imagen es una matriz de 28x28 puntos y una forma sencilla, y veremos que eficaz, es ver qué puntos son comunes a la mayoría de las imágenes que representan un 0, o dicho de una forma más precisa, cual es la probabilidad de que cada uno de los 28x28=784 puntos tengan un determinado nivel de gris en una imagen del 0. Esto significaría que buscamos una matriz de 764 (puntos) x 255 (niveles de gris), nada menos que 199.920 valores, pero no necesitamos ser tan exigentes y podemos combinar probabilidad y nivel en un solo valor, si aceptamos que una probabilidad baja de un nivel alto, es equivalente a una probabilidad alta de un nivel bajo, la cosa se simplifica mucho, no sólo en cuanto a tamaño de la matriz si no en la forma de obtenerla, tan sencillo como hacer la media, punto a punto de todas las imágenes que representan un 0.

En la muestra de entrenamiento hay 5.923 imágenes de 0 y la figura 3 muestra la matriz de probabilidad que resulta de hacer lo mencionado en el párrafo anterior:

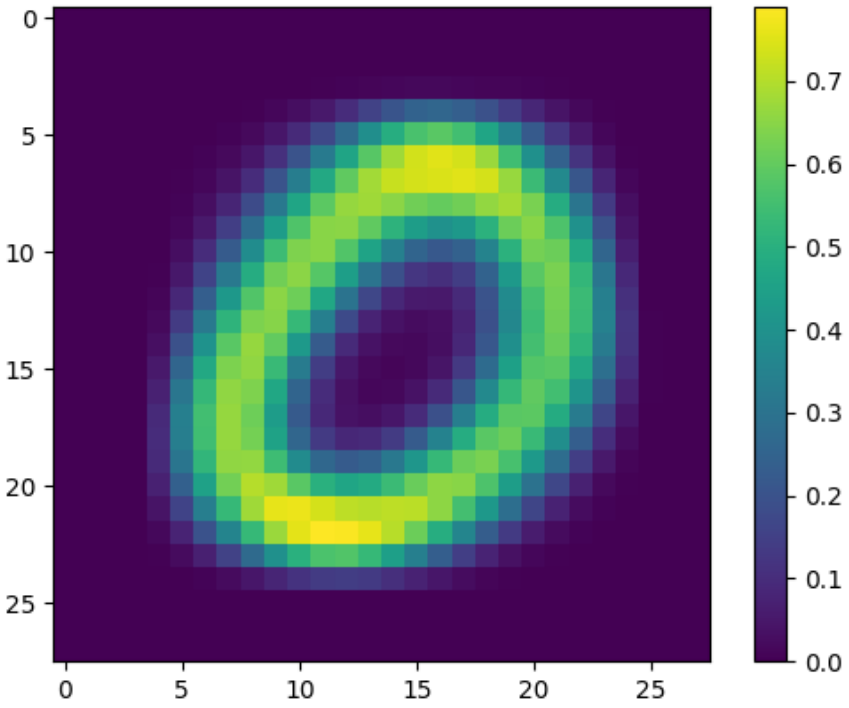

Figura 3. Matriz de probabilidad de las imágenes del dígito 0. Puntos comunes en las imágenes del 0.

El método que vamos a seguir para saber si una imagen representa un 0 será compararla con esta matriz, si los puntos más intensos de la imagen se corresponden con los de mayor probabilidad de la matriz es muy probable que la imagen sea la de un 0. Esta descripción cualitativa hay que precisarla cuantitativamente de forma que para cada imagen podamos obtener una "distancia" respecto de la matriz de probabilidad, cuanto menor sea la distancia más probable es que la imagen represente un 0.

La forma tradicional de calcular esta distancia es el error cuadrático medio (ECM):

Dada una imagen, definimos el error, en un punto, como la diferencia entre la probabilidad de ese punto (valor en la matriz de probabilidad) y el valor del punto en la imagen, elevamos la diferencia al cuadrado, para evitar compensaciones entre positivos y negativos, para finalmente obtener la distancia sumando las diferencias de los 28x28 puntos.

Tenemos la referencia y la forma de calcular distancias, luego ya podemos aplicar el método a todas las imágenes de la muestra de prueba y calcular sus distancias al 0, pero nos falta decidir a partir de que distancia vamos a considerar que la imagen no es un 0.

Si consideramos como punto de corte el valor mínimo de las distancias obtenidas, al aplicar el método a todas las imágenes, tendremos la garantía de que todos los 0 serán etiquetados como 0, una sensibilidad del 100%, pero el grado de acierto general será muy malo porque todas las imágenes restantes también serán etiquetadas como 0 y son mayoría dentro de la muestra. En el otro extremo, poniendo el corte en el valor máximo, la sensibilidad será del 0% (no se detecta

ningún 0) pero el grado de acierto estará próximo al 90% porque el 90% de las imágenes no son 0.

El valor correcto del punto de corte será un valor intermedio, que podemos obtener mediante tanteo, hasta obtener una sensibilidad y acierto que nos parezcan aceptables, si es que esto es posible ya que puede resultar que la metodología, en sí misma, no sea suficientemente buena para distinguir las imágenes con 0 del resto.

Para no tratar con muchos resultados intermedios, antes de hacer la aplicación práctica de lo dicho vamos a incorporar una mejora al método.

Hasta ahora hemos hecho énfasis en lo que se parecen los 0 entre sí, pero no hemos considerado aquello que los diferencia del resto de los dígitos, si esto fuese posible se mejorarían los resultados.

Una forma sencilla de resaltar lo diferente es modificar la matriz de probabilidad sustituyendo los valores inferiores a un cierto valor, digamos que a 0,1, por -1. De esta forma cuando una imagen muestre un punto en esa posición, el error se magnifica, ya que en lugar de medirse respecto a 0,1 se mide respecto a -1 dando lugar a una distancia mayor. En la figura 4 se muestra la nueva matriz que llamaremos patrón y no de probabilidad porque tiene valores negativos.

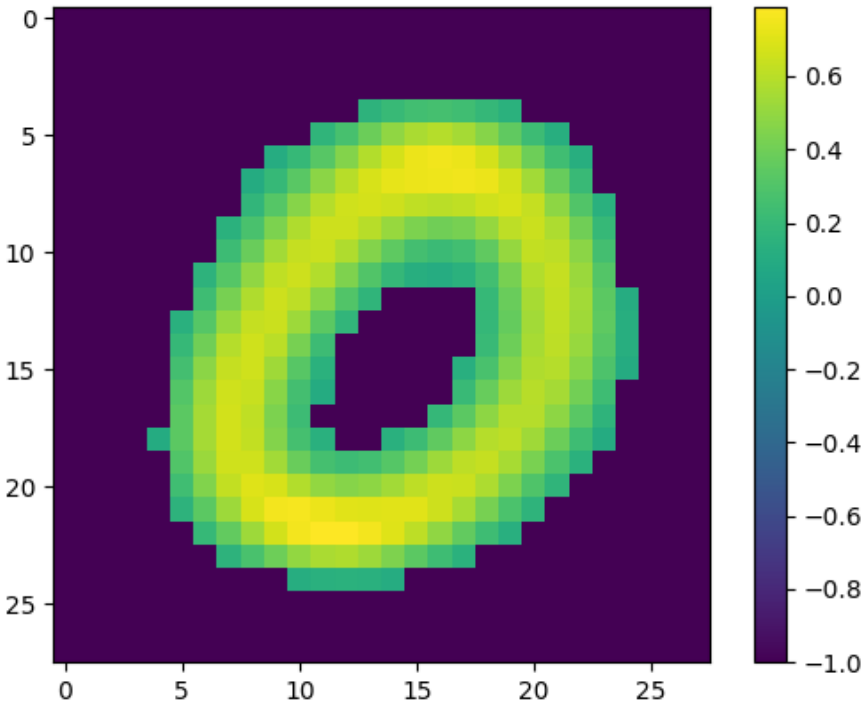

Figura 4. Patrón de las imágenes del dígito 0.

Por fin estamos en disposición de aplicar la metodología de reconocimiento de 0. Para orientar el tanteo que nos debe llevar al punto de corte, en la figura 5 representamos la distribución de distancias (número de veces que se repite cada distancia). Estas distancias son las 60.000 obtenidas con la muestra de entrenamiento, la muestra de prueba la reservamos para comprobar el método, sin que sus imágenes intervengan en el proceso previo.

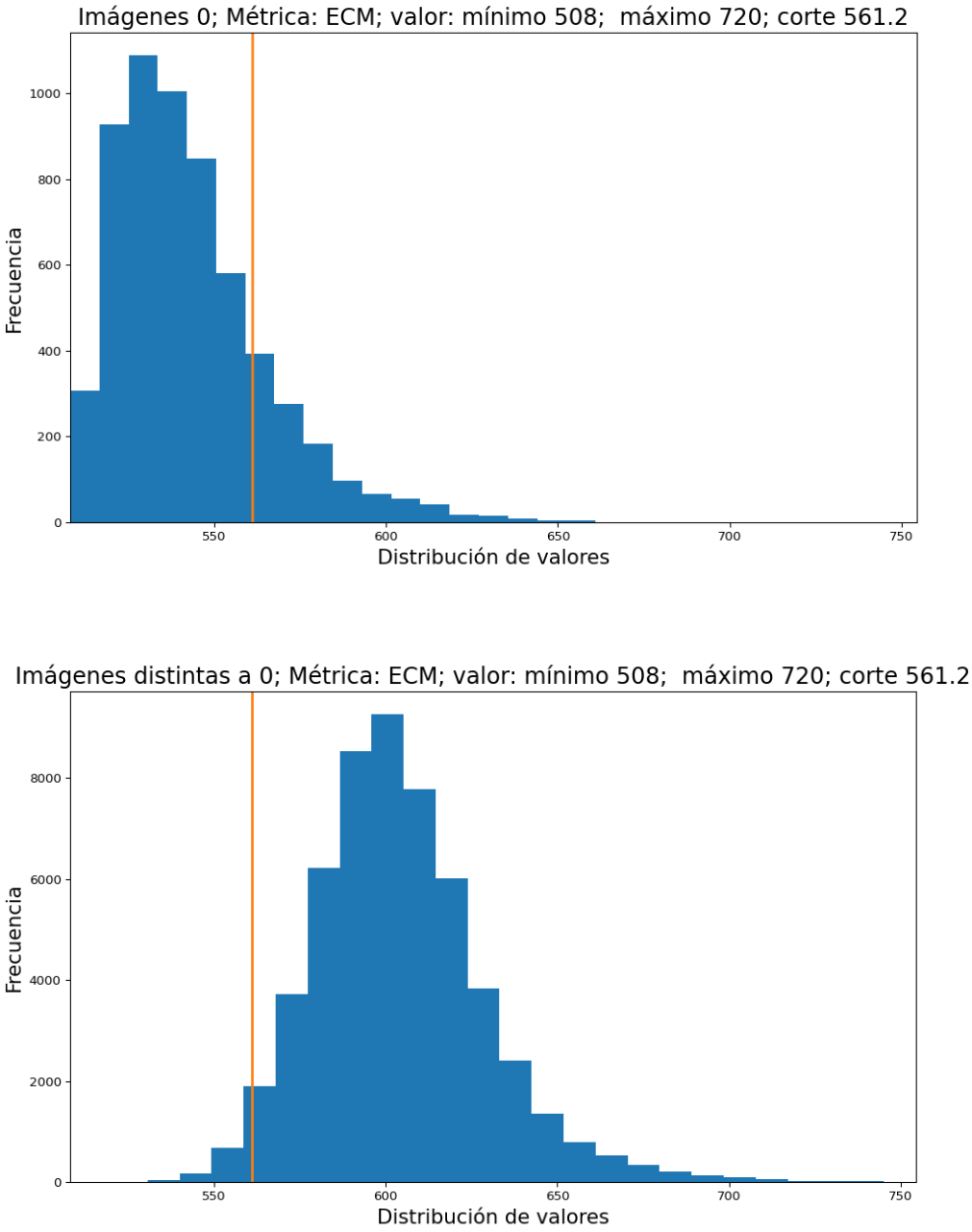

Figura 5. Distribución de distancias para la muestra de entrenamiento con métrica ECM. Arriba para las imágenes 0 y abajo para el resto de las imágenes

En la figura 5 está marcado un punto de corte en la distancia 561,2 que parece razonable. Al mover el punto de corte a la derecha mejora la sensibilidad (ceros acertados) y empeora el acierto. En la otra dirección todo lo contrario. La imagen muestra como el criterio de distancia separa a izquierda y derecha las imágenes 0 del resto, aunque no puede evitar que en las cercanías de punto de corte se solapen.

Los resultados que se obtiene al procesar las 10.000 imágenes de la muestra de prueba son los siguientes (tabla 1):

```
Muestra de prueba. dígitos 0 : 980; dígitos del 1 al 9: 9,020; total  10,000

Distancia mínima       508.2       0.648 por punto

Distancia máxima       720.4       0.919 por punto

Punto de corte     561.209       0.716

Falsos positivos   (dígitos del 1 al 9 confundidos con 0)          :     205  2.3%

Falsos negativos   (dígitos 0 no reconocidos)                      :     150 15.3%

Aciertos positivos (dígitos 0 reconocidos como tal -sensibilidad-) :     830 84.7%

Aciertos negativos (dígitos 1 al 9 reconocidos como no 0)          :   8,815 97.7%

Aciertos totales                                                   :   9,645 96.5%
```

Tabla 1. Resultados del modelo de reconocimiento de 0 con métrica ECM

Un 96,5% de acierto parece muy bueno, pero no hay que engañarse, se puede acertar el 90,2% sin ningún método, es lo que resulta si siempre se dice que no es un 0. Pero si se considera el 84,7% de sensibilidad, ya es otra cosa, porque la predicción trivial tendría un 0% de sensibilidad, pero nuevamente insistir en que no queremos valorar la bondad del método, tan sólo hemos dado nuestro primer paso hacia las redes neuronales artificiales, ahora vamos a dar el segundo.

Para medir la distancia de una imagen al patrón hemos utilizado el error cuadrático medio (ECM), pero hay otras posibilidades, como la siguiente:

Dada una imagen, vamos a definir el criterio de semejanza (distancia) como la suma de los productos punto a punto de la imagen y el patrón. Hay que reseñar que cuanto mayor sea la distancia más se parecerá la imagen a un 0, al contrario de lo que ocurre con el ECM. Esta métrica es una especie de covarianza, su valor es máximo cuando los puntos de más nivel de color coinciden en la imagen y el patrón, por lo que la llamaremos métrica covariante (CC). En la figura 6 se muestra la distribución de las distancias de esta métrica y en la tabla 2 los resultados para la muestra de prueba.

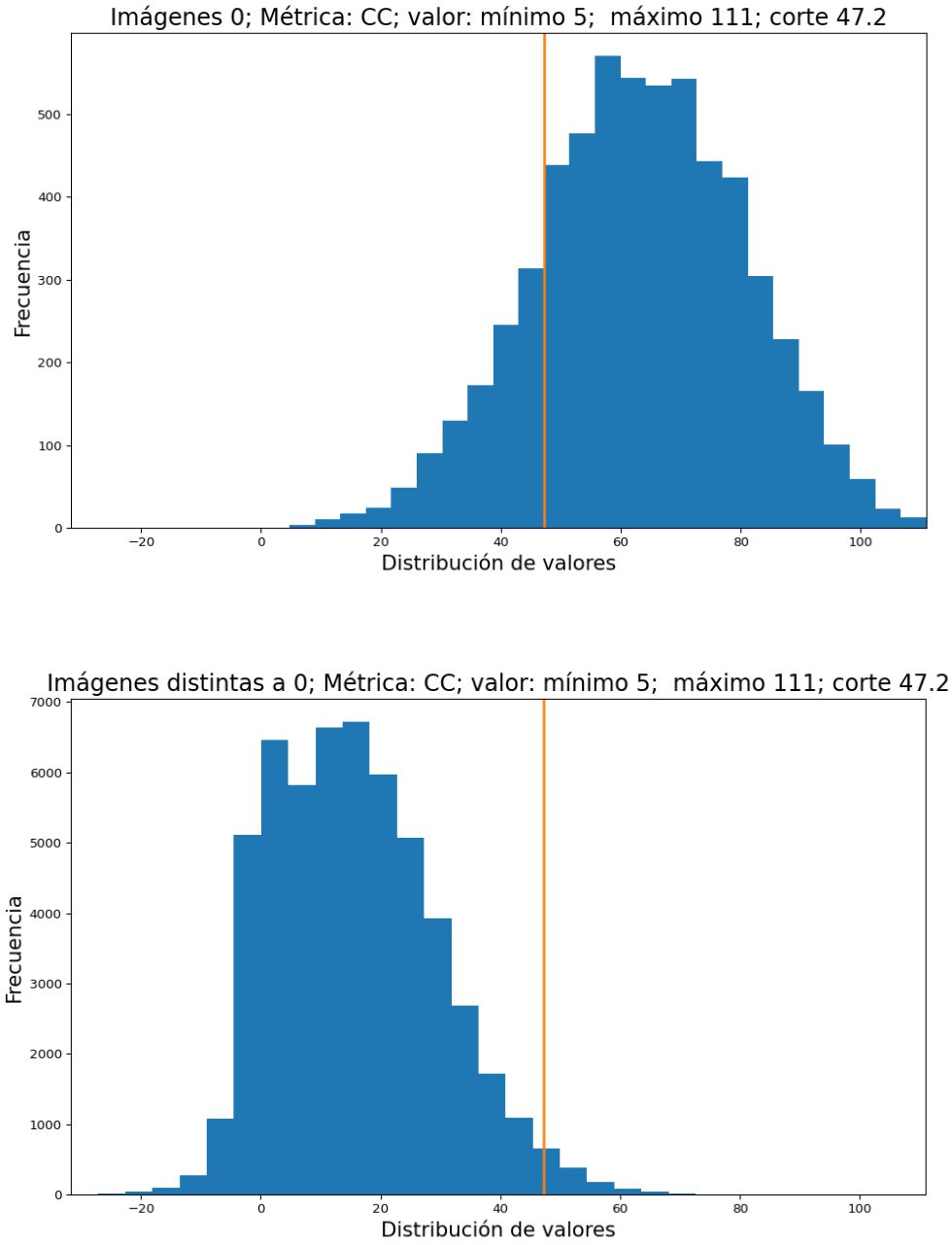

Figura 6. Distribución de distancias para la muestra de entrenamiento con métrica CC.

El punto de corte ahora está a la izquierda, por que a mayor distancia CC, más semejanza.

Distancia mínima		4.7	0.006 por punto		
Distancia máxima		110.9	0.142 por punto		
Punto de corte		47.202	0.060		
Falsos positivos	(dígitos del 1 al 9 confundidos con 0)			:	219 2.4%
Falsos negativos	(dígitos 0 no reconocidos)			:	175 17.9%
Aciertos positivos	(dígitos 0 reconocidos como tal -sensibilidad-)			:	805 **82,1%**
Aciertos negativos	(dígitos 1 al 9 reconocidos como no 0)			:	8,801 97.6%
Aciertos totales				:	9,606 **96.1%**

Tabla 2. Resultados del modelo de reconocimiento de 0 con métrica CC

Los resultados son similares a los obtenidos con la métrica ECM, algo peores. Al margen de usar una métrica ECM o CC, nuestro método tiene una posibilidad de mejora nada despreciable en el cálculo del patrón, sus valores los hemos obtenido mediante reglas bien simples: un valor medio y un valor -1 para todos los que resultan menores que 0,1. Tanto el 0,1 como el -1 son arbitrarios y podemos, mediante tanteo, buscar valores mejores, pero ya puestos, lo que podemos buscar son los 784 valores que conduzcan a los mejores resultados, pero **AQUÍ QUERIAMOS LLEGAR**.

2. Llegamos a la RNA

Nuestro OCR básico con la métrica CC es estrictamente una red neuronal artificial (RNA). En concreto una red densa de topología 784:1, es decir, 784 entradas y una salida (figura 7) y lo que se hace al entrenar una red es buscar los pesos que más acierto producen, los pesos son, en este caso, exactamente lo mismo que los valores de nuestro patrón, luego el entrenamiento hace lo que acabamos de decir que habría que hacer para mejorar los resultados.

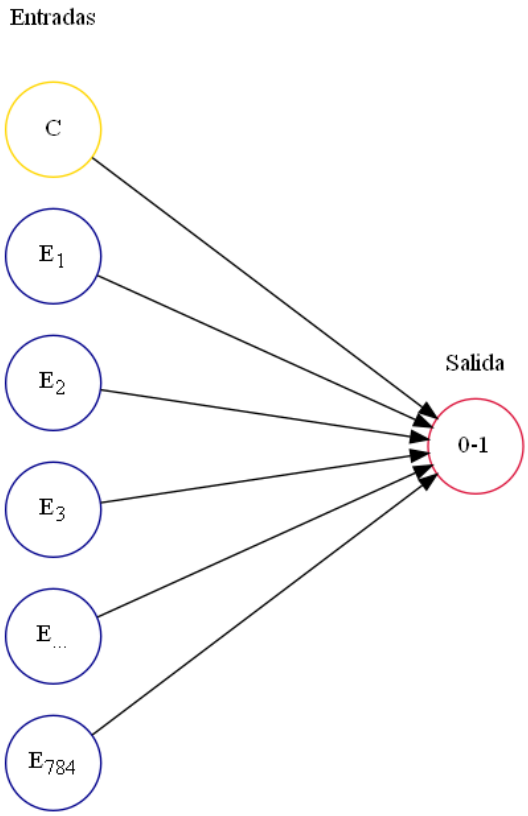

Figura 6. OCR con métrica CC, o Red Neuronal Artificial 784:1, ya que son la misma cosa. La "C" enmarcada en amarillo es una constante independiente de las entradas que se denomina sesgo y que se suma a las salidas de las neuronas de entrada para determinar el valor final de salida de la RNA. Es como el termino independiente de una ecuación.

El entrenamiento de la RNA determina el valor de los pesos mediante la técnica de backtracking, que en esencia consiste en repartir el error que se observa a la salida de la red (nuestra distancia) entre las entradas de forma proporcional al valor de los pesos que unen cada entrada con la salida. En mi opinión, este método es el responsable del apogeo de las RNA y no tanto el incremento exponencial de las capacidades de computación, como normalmente se apunta. El backtracking se compromete y la capacidad de computación colabora.

Las figuras 7 y 8 y la tabla 3 muestran los resultados de la RNA 784:1 que es lo mismo que nuestro OCR con métrica CC.

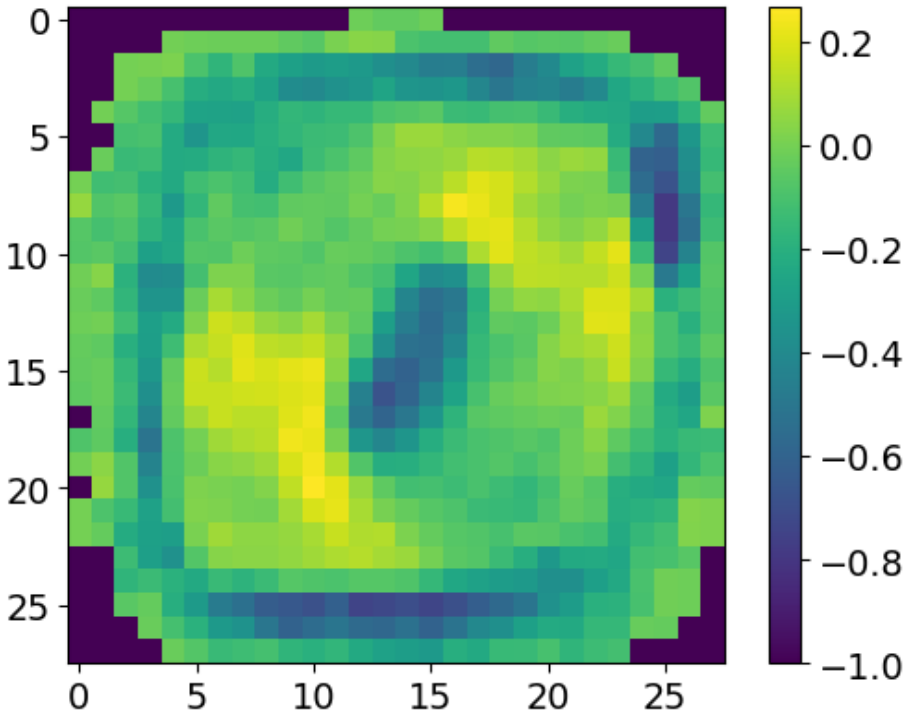

Figura 7. Matriz de pesos de la RNA 784:1

Resultados para la muestra de prueba			
Dígitos 0 :	980 dígitos del 1 al 9:	9,020 total	10,000
Falsos positivos	(dígitos del 1 al 9 confundidos con 0)	:	31 0.3%
Falsos negativos	(dígitos 0 no reconocidos)	:	32 3.3%
Aciertos positivos	(dígitos 0 reconocidos como tal -sensibilidad-)	:	948 **96.7%**
Aciertos negativos	(dígitos 1 al 9 reconocidos como distintos a 0)	:	8,989 99.7%
Aciertos totales		:	9,937 **99.4%**

Tabla 3. Resultados del modelo de reconocimiento del dígito 0 mediante la RNA

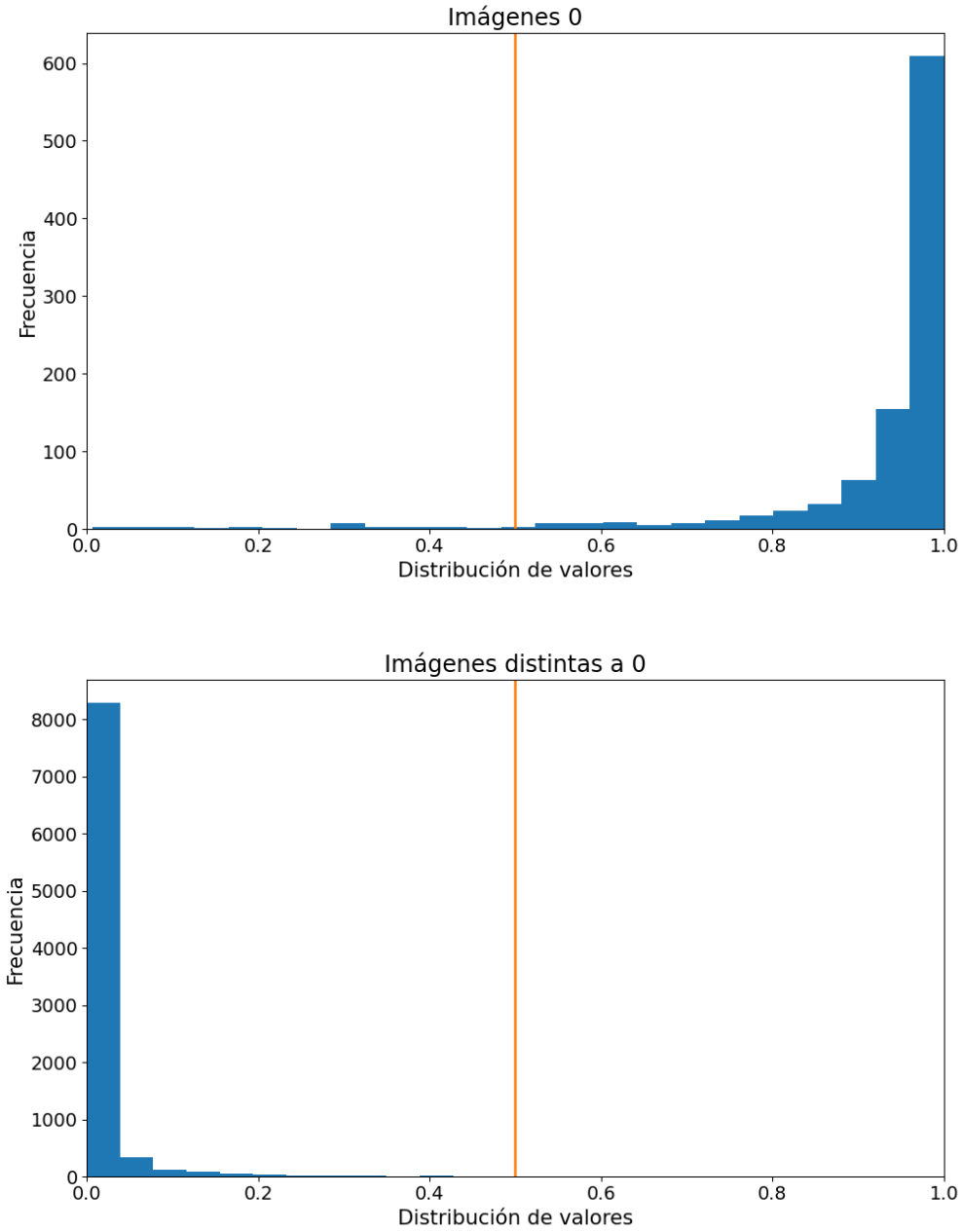

Figura 8. Distribución de las predicciones (distancias) de la RNA para la muestra de prueba.

Como era de esperar, al optimizar los pesos, uno a uno, los resultados mejoran tanto la sensibilidad como el acierto de forma considerable y la mejor prueba es la figura 7 donde se puede ver que las predicciones para las imágenes del 0 están separadas de las predicciones para el resto de las imágenes de forma casi perfecta, aunque una vez más hay que recordar que esto no es lo importante aquí. Lo importante es constatar que La RNA utiliza la maraña de pesos para expresar las semejanzas entre las observaciones de una misma clase y las diferencias con las otras, de una forma finalista sin atender a criterios de lógica, teoría o conocimiento del problema, a diferencia de cómo hemos hecho nosotros con el OCR. La ventaja de la RNA es que no se necesitan conocimientos previos y el precio es que la red de pesos no suele decirnos nada, incluso en el presente caso, donde sabemos de antemano de lo que se trata, cuesta reconocer

lo que la figura 6 representa. Según hemos ido avanzando hacia la red se ha ido difuminando el patrón del 0 (figura 8)

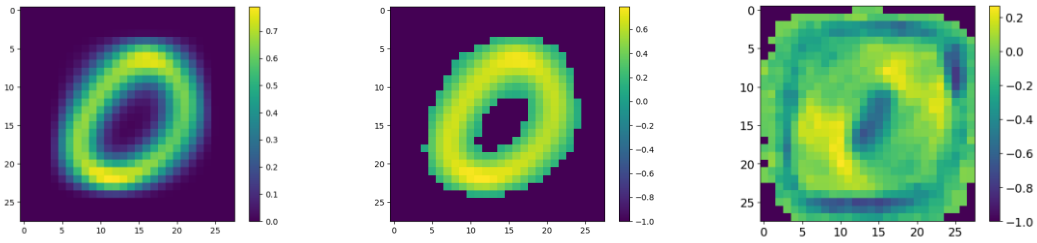

Figura 8. Evolución del patrón de las imágenes del dígito 0

Si esto ocurre en un caso extremadamente simple, podemos imaginar lo que ocurrirá cuando se trate con una red mínimamente complicada. Si no se tiene el conocimiento previo, la RNA no lo da (ni siquiera lo presta). No hay más, la RNA codifica, en la red de pesos, semejanzas y diferencias entre las clases de objetos que se quieren clasificar, sean los objetos lo que sean y las semejanzas y diferencias se deban a lo que se deban. Esto es tremendamente eficaz pero nada instructivo.

A pesar de esto, desde el principio se ha intentado descodificar la maraña de pesos en busca de respuestas sobre lo que la RNA "aprende", para encontrar el porqué de las cosas según los estándares de comprensión humana. En mi opinión esto tiene poco sentido y produce reflexiones absurdas. No hace mucho que la RNA AlphaZero, desarrollada por Google, ha demostrado que es capaz de ganar al ajedrez a cualquier persona o máquina de las que hoy existen y lo hace de forma contundente. En todas partes se repite que ha aprendido a jugar al ajedrez ella sola sin más información que las reglas básicas del ajedrez y en unas pocas horas, lo que en esencia es falso, AlphaZero no sabe jugar al ajedrez en lo que respeta a su RNA. AlphaZero es una aplicación Monte Carlo que simula partidas realizando movimientos de forma aleatoria, de acuerdo con una función de probabilidad que es una RNA entrenada jugando contra sí misma. Monte Carlo explora el futuro que se deriva de una jugada moviendo aleatoriamente y viendo en que acaba la partida, la RNA le dice a Monte Carlo que juagadas debe probar con más probabilidad de entre todas las posibles.

Una vez que hemos llegado a la RNA, desde el OCR, vamos a seguir con ella para entenderla mejor. La RNA que hemos utilizado (figura 1) es la más sencilla que se puede considerar, las entradas viertan directamente a la salida, pero esto es excepcional, lo normal es que existan capas de neuronas intermedias entre la entrada y la salida, como en la figura 9.

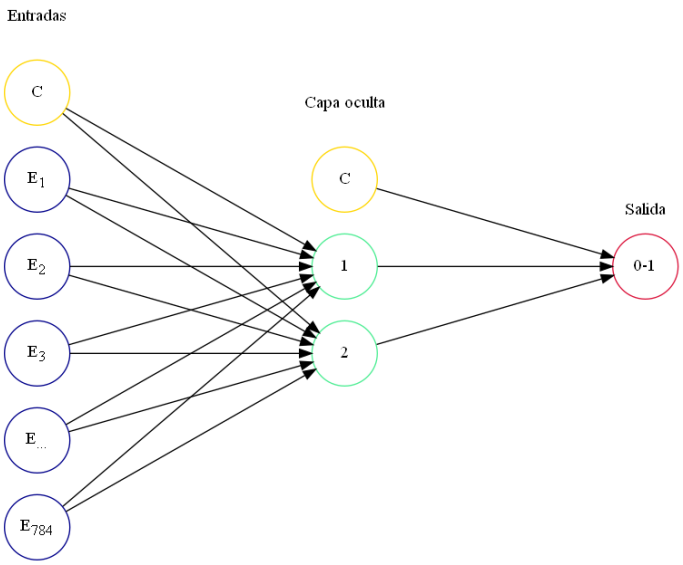

Figura 9. RNA con una capa intermedia, normalmente llamada oculta, de dos neuronas

Esta RNA con una capa intermedia la podemos interpretar como la combinación de dos RNA sencillas: cada una formada por las entradas y una de las neuronas intermedias que es la salida de la subred (figura 10), luego, los resultados de estas subredes (la salida de las neuronas de la capa intermedia) se combinan con los pesos que van de la capa intermedia a la de salida, en lo que sería una tercera RNA sencilla.

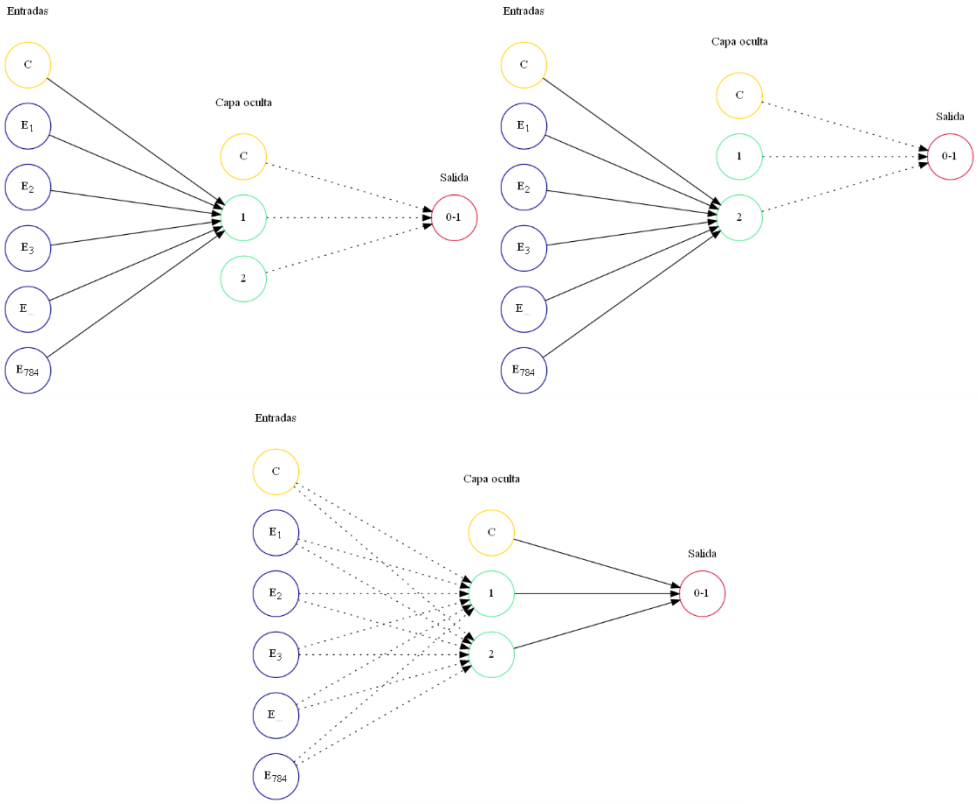

Figura 10. Interpretación de la RNA con una capa oculta de dos neuronas como tres RNA del tipo sencillo donde las entradas vierten directamente a la salida.

En las RNA de tipo convolucional, a las que pronto llegaremos, se dice que las dos primeras RNA codifican dos “características” de los datos de entrada. Cuando se trata de imágenes, las “características” se interpretan como rasgos del tipo de imagen, si son rostros, pueden ser la forma de la nariz, orejas, mentón etc. (esta interpretación me genera dudas). Aunque lo esencial de Las RNA convolucionales es que añaden la desubicación de esos rasgos, como ya veremos.

Los resultados obtenidos con la RNA sencilla (784:1) son excelentes pero aun así podemos tratar de mejorarlos con una RNA más compleja como la de la figura 9, se logré o no, será un paso hacia algo que está por encima del grado de acierto.

La tabla 4 muestra los resultados de aplicar a la muestra de prueba la nueva RNA entrenada y la figura 11 la matriz de pesos resultantes, que ahora son dos: una para cada una de las dos primeras subredes (la matriz de pesos de la tercera subred sólo tiene dos valores)

Falsos positivos	(dígitos del 1 al 9 confundidos con 0)	:	27	0.3%
Falsos negativos	(dígitos 0 no reconocidos)	:	39	4.0%
Aciertos positivos	(dígitos 0 reconocidos como tal -sensibilidad-)	:	941	**96.0%**
Aciertos negativos	(dígitos 1 al 9 reconocidos como distintos a 0)	:	8,993	99.7%
Aciertos totales		:	9,934	**99.3%**

Tabla 4. Resultados del modelo de reconocimiento del dígito 0 mediante la RNA 784:2:1

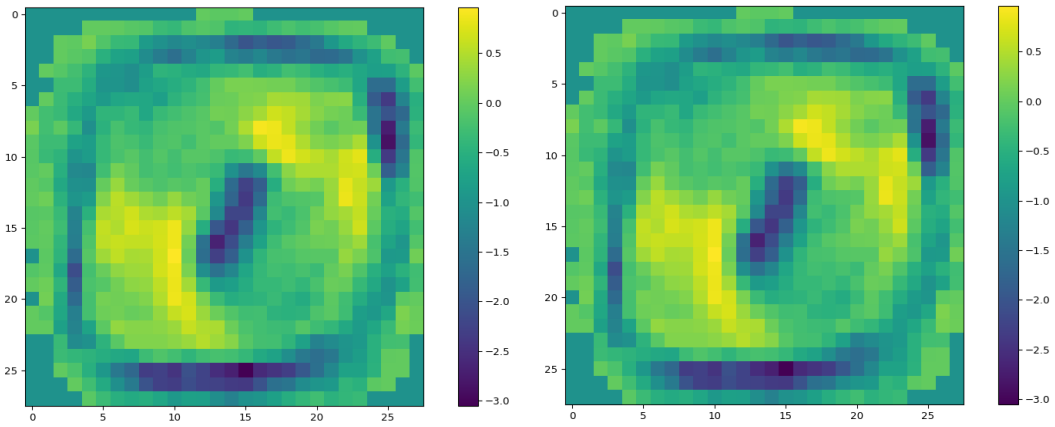

Figura 11. Matrices de pesos desde las entradas hasta las dos neuronas de la capa intermedia.

Al comparar las tablas 3 y 4 se ve que la nueva RNA no mejora los resultados de la primera, que es más sencilla. Además, las dos matrices de pesos (figura 11) son idénticas entre sí y con la de la RNA 784:1 (figura 7). Para encontrar alguna diferencia hay que recurrir a la distribución de distancias, la figura 12 muestra esta distribución para la nueva RNA.

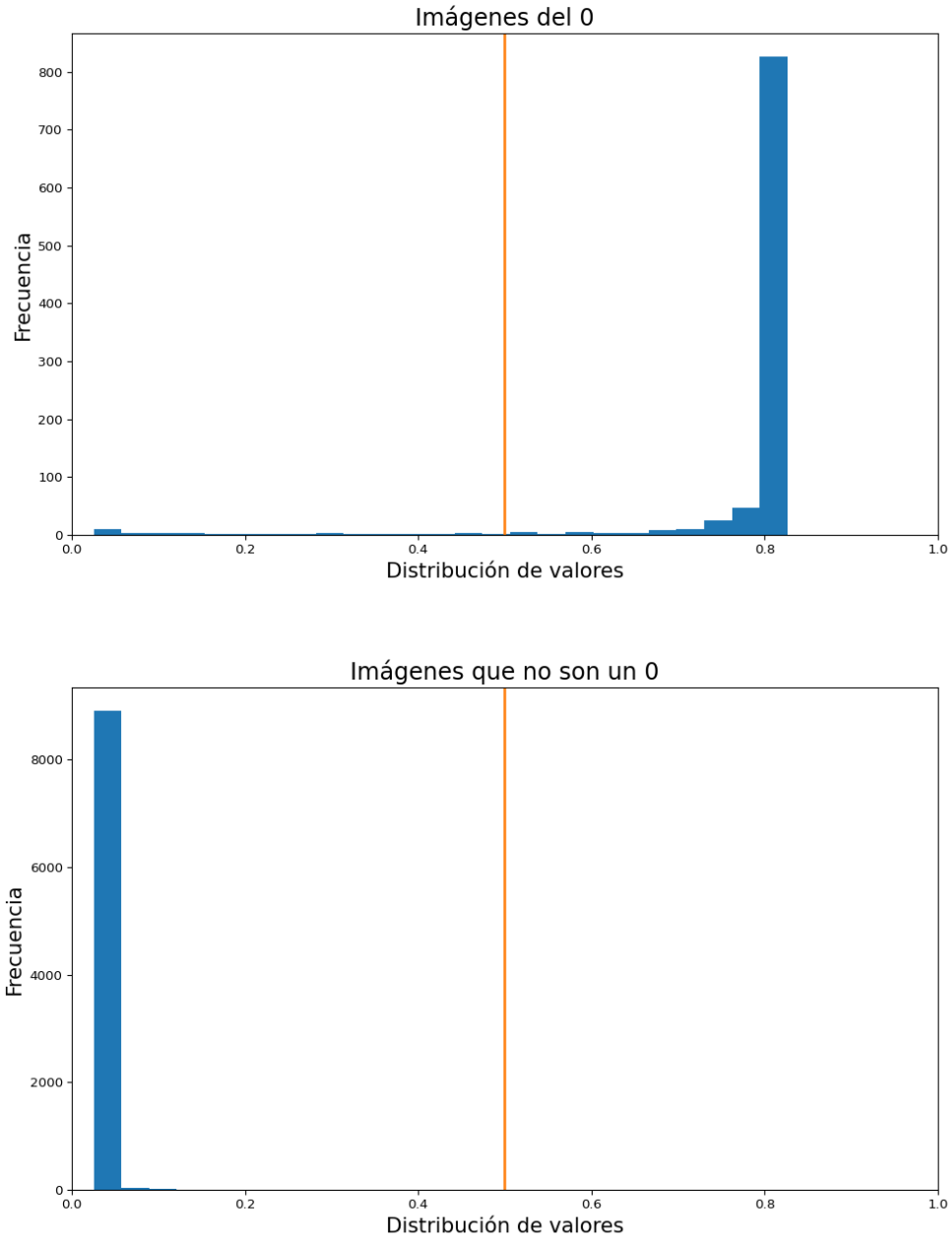

Figura 12. Distribución de las predicciones de la RNA 784:2:1 para la muestra de prueba.

La separación entre imágenes del 0 y resto de imágenes es igual de nítida en la figura 8 como en la 12, pero en la figura 12 se ve que el valor máximo que la RNA otorga a las imágenes del 0 es del 0,8 y no del 1,0 como ocurre en la figura 8. Hay que recordar que este valor es la confianza de la RNA en que la imagen corresponde a un 0. La RNA sencilla se siente confiada al 100% pero la nueva sólo al 80%. Si esto tiene algún significado o es un mero artefacto matemático no lo sé, pero antes de continuar vamos a introducir un nuevo elemento en el análisis.

3. La RNA pasa a la acción.

Hasta ahora nos hemos limitado a “enseñar” imágenes a las RNA, primero para entrenarlas y luego para que las clasifiquen (prueba), ha llegado el momento de pedirles que sean ellas las

que nos enseñen imágenes a nosotros, es decir, que dibujen imágenes del 0, esto tiene una importancia capital como veremos más adelante.

La RNA por sí misma no puede dibujar imágenes pero podemos hacer algo parecido a AlphaZero, La RNA puede guiar a otra aplicación que será quien dibuje. La idea es bastante simple, la aplicación para dibujar asignará un valor a cada uno de los puntos de la imagen (pixeles), en nuestro caso estos valores serán cero u uno. Partimos de una matriz 28 x 28 con los 784 valores a cero, esta imagen se la enseñamos a la RNA para que nos diga cual es la probabilidad de que sea la imagen de un 0 (suponemos que pequeña), a continuación ponemos a uno el píxel de arriba a la izquierda y preguntamos a la RNA, si la probabilidad es mayor, dejamos el píxel a uno, en caso contrario lo devolvemos a cero. Repetimos el proceso, píxel a píxel para los 783 pixeles restante y tendremos una imagen que esperamos que alcance una probabilidad alta de corresponderse al dígito 0. Luego vamos a introducir dos refinamientos que lo cierto es que han sido casi inútiles. Si la probabilidad de la imagen sintetizada no alcanza un valor que consideremos aceptable, pasamos a genera puntos al azar y conmutamos su valor, si ese píxel está a cero ponemos un uno y viceversa, tras cada cambio preguntamos a la RNA si el cambio produce mejora y si es así lo aceptamos o revertimos en caso contrario. Finalmente, limpiamos la imagen, para ello hacemos algo parecido al principio, recorremos los pixeles uno a uno, interesándonos sólo por los que están a uno, si es así lo pasamos a cero y preguntamos a la RNA, aceptando, o rechazando el cambio, si mejora la probabilidad de que la imagen corresponda a un 0.

Esto se puede hacer igual con el OCR, aunque podamos predecir que va a dibujarnos el “patrón”, porque no hay imagen más parecida al patrón que la de el mismo, al menos nos servirá para comprobar que nuestra aplicación de dibujo trabaja correctamente. La figura 13 muestra el resultado del OCR con la métrica ECM y CC

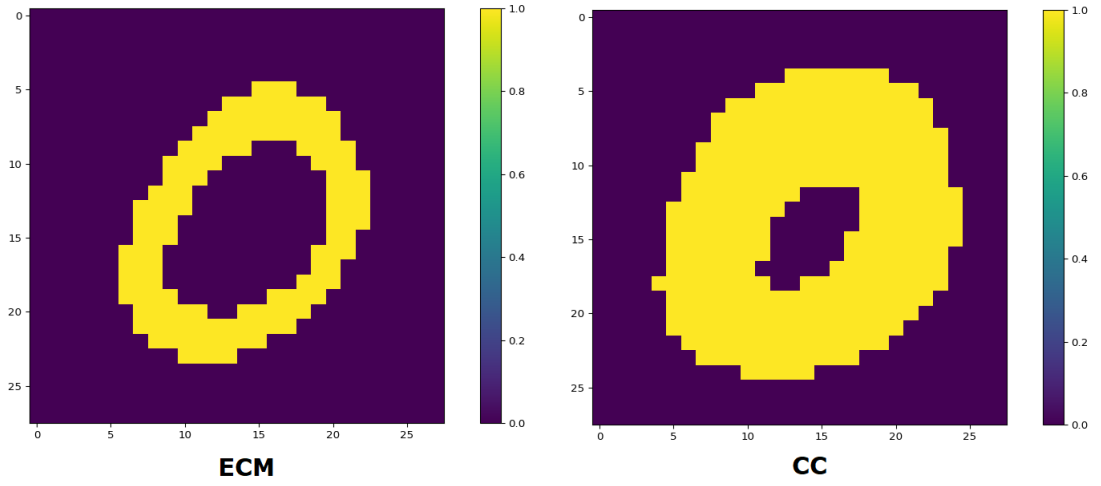

Figura 13. Dibujos del dígito 0 creados mediante el OCR

Para los dibujos creados por las RNA podemos esperar que serán algo parecido a las matrices de pesos (figuras 7 y 11), el equivalente al patrón OCR. La figura 14 muestra el resultado.

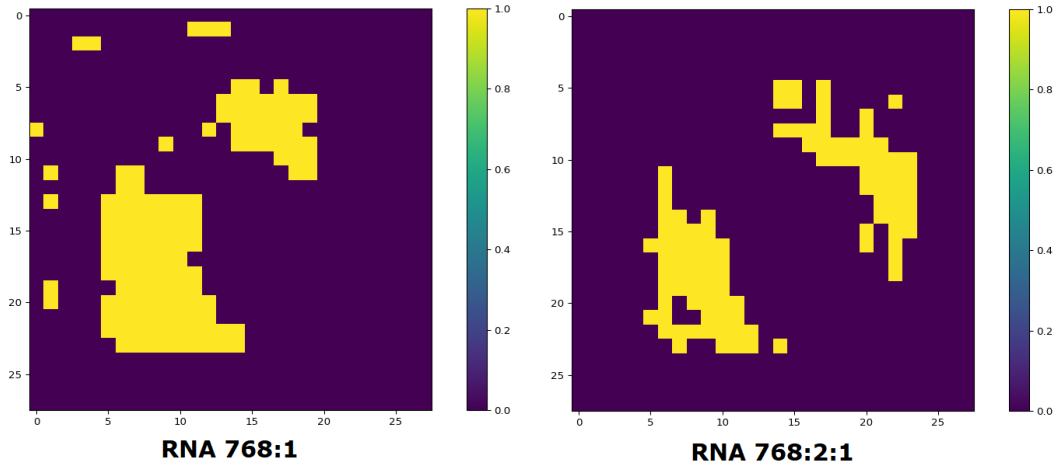

Figura 14. Dibujos del dígito 0 creados con las RNA.

Creo que un observador, no informado previamente, no reconocería el 0 en las imágenes de la figura 14, pero tienen su importancia al mostrar con más claridad que la matriz de pesos lo que la RNA considera esencial.

Primero, aclaremos que se pueden crear muchas imágenes de 0 distintas, pero sólo si eliminamos la primera fase de nuestro procedimiento para dibujar: el recorrido sistemático de la imagen vacía para ver que píxeles ponemos a uno. Este procedimiento, por sí sólo, genera imágenes que ambas RNA valoran con una alta probabilidad de ser la imagen de un 0, el Monte Carlo siguiente no aporta nada y la limpieza final es sólo una cuestión estética.

Durante un tiempo estuve intrigado porque la RNA 784:2:1 no era capaz de dibujar imágenes con una probabilidad mayor del 80% de corresponder al dígito 0, hasta que descubrí, mirando la figura 12, que este es el valor máximo que la RNA otorgaba a cualquier imagen. Como he mencionado, no sé por qué ocurre esto, pero a la vista (figura 14) de lo que la RNA considera, que es una imagen del 0 (con su máxima probabilidad), parece un ejercicio de modestia no asignarle una confianza del 100% como hace la RNA sencilla.

Aunque ya tenemos elementos para avanzar alguna conclusión, a mi parecer importante, vamos a introducirnos antes en las redes convolucionales con las que tendremos evidencias más concluyentes.

4. RNA convolucionales.

Con las RNA que hemos visto, hemos conseguido separar las imágenes del dígito 0, del resto de imágenes, con aciertos cercanos al 100%, pero hay que ser consciente de que las imágenes con las que trabajamos lo ponen fácil. Aunque los dígitos están dibujados a mano y por manos muy diferentes (figura 2), todas tienen el mismo tamaño y el dígito ocupa la parte central de las mismas con los bordes de la imagen vacíos y tamaños de dígitos similares, sin estas circunstancias es fácil imaginar que los resultados se degradarían considerablemente, quizás hasta el absurdo. Esta es la razón por la que se investigó como construir una RNA que pudiesen detectar objetos con independencia de su lugar en la imagen o su tamaño. El éxito actual de la IA debe mucho al éxito con las RNA convolucionales, que son la respuesta al reto planteado. Las RNA secuenciales, que escapan de lejos a este trabajo, son el otro pilar de avance de la IA, son las responsables de los logros en el habla, la escritura y la traducción. Su elemento diferenciador

es que hacen intervenir al tiempo, sus salidas se convierten en las entradas de la red en el instante siguiente.

La forma de funcionar de las RNA convolucionales es conceptualmente sencilla, es lo que hacemos nosotros cuando exploramos una imagen con una lupa: vamos desplazando la lupa por la imagen hasta encontrar lo que buscamos.

Comencemos por construir un OCR convolucional, que llamaremos “filtro”. Tenemos un conjunto de imágenes en blanco y queremos saber cuáles tienen una mancha en alguna parte (un cuadrado de 2x2), por ejemplo la de la figura 15.

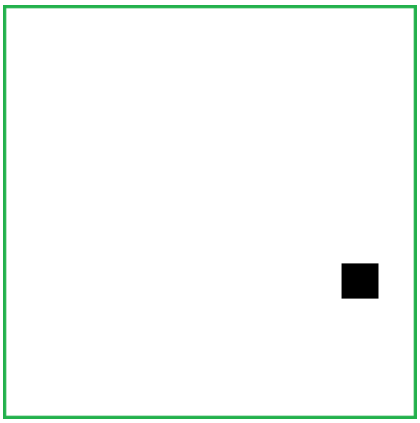

Figura 15. Imagen con un cuadrado 2x2

Nuestra lupa deja ver un cuadrado interior de 4x4 píxeles (que llamaremos marco), y sabe contar píxeles negros. Si al superponer la lupa sobre la imagen resulta que el número de pixeles negros que ve es de exactamente 4, registra un valor 1 en una lista, en caso contrario registra un 0. A la lista la llamaremos lista de salidas.

Para saber si en la imagen hay un cuadrado 2x2 iremos moviendo la lupa de la siguiente forma:

1. La situamos en la parte superior izquierda de la imagen. Miramos y la lupa apuntará un 0 o un 1 en la lista de salidas.
2. Desplazamos la lupa un píxel a la derecha, lo que hace que vea un nuevo conjunto de 16 píxeles, casi todos serán los mismos que en el paso anterior, se perderá una columna de 4 píxeles por la izquierda y entra una columna nueva por la derecha, pero esto no es relevante. La lupa apunta un 0 o un 1 en la lista de salidas, contando cuantos pixeles negros ve en esta posición.
3. Nos seguimos desplazando a la derecha de píxel en píxel y anotando en la lista de salidas hasta que el borde derecho de la lupa coincida con el borde derecho de la imagen. Si la imagen es de 64x64 pixeles esto ocurrirá después de desplazarnos 64 – 4 = 60 veces y en la lista tendremos apuntados 61 valores (el inicial y los 60 desplazamientos).
4. Volvemos a poner la lupa en la parte izquierda de la imagen, pero esta vez un píxel por debajo de donde la habíamos puesto al empezar y repetimos el proceso de desplazarnos a la derecha. Al terminar tendremos otros 61 valores más en la lista de salidas.
5. Repetimos los recorridos a la derecha, hasta que al terminar uno y volver a poner la lupa en la izquierda ya no encaja, vemos por debajo una línea que no es de la imagen. En este momento tendremos 61 x 61 valores en la lista de salidas.
6. Si la lista de salida son todos ceros menos un valor que es 1, nuestro filtro dictaminará que en la imagen hay un cuadrado 2x2, en caso contrario que no lo hay.

Somos conscientes de que si en las imágenes hay algo más que la mancha cuadrada de 2x2 (por ejemplo una mancha de 3x2) nuestro filtro se equivocará, pero es perfecto para situaciones en las que una de dos: no hay nada o hay una mancha de 2x2. El contexto es importante.

Transformar el filtro en una capa convolucional es muy simple, en lugar de contar píxeles negros, dotamos a la lupa de una RNA 16:1 y en la lista de salidas apuntamos el valor de la salida de esta RNA en lugar del número de pixeles negros. Las entradas de la RNA 16:1 son los 16 píxeles que la lupa ve en cada en posición, cambian al cambiar de posición, pero los pesos de la RNA son siempre los mismos.

En realidad una capa convolucional tiene más de un filtro, frecuentemente 32 o 64 pero pueden ser cualquier número. Todos los filtros funcionan igual pero cada uno tiene su propia RNA con una matriz de pesos distinta y el número de salidas de la capa convolucional se multiplica por el número de filtros.

Si el marco es igual a la imagen, sólo se puede superponer una vez y la RNA produce un solo valor, podemos considerar que nuestra RNA 678:1 es una RNA convolucional con un marco del mismo tamaño que la imagen y un solo filtro, con dos filtros sería la RNA 678:2:1. Esto no tiene más importancia que mostrar como las capas convolucionales son una generalización de las capas de una RNA densa.

En una capa convolucional el número de pesos a optimizar viene multiplicado por el número de filtros y si se apilan capas convolucionales el crecimiento es exponencial. Para limitar ese crecimiento se utilizan las llamadas capas de concentración (pooling). La capa de concentración reduce las salidas de la capa convolucional agrupando varias de ellas (por ejemplo cada 16) en una sola, una forma habitual es quedarse con el mayor valor de cada grupo.

Cada filtro extrae una “característica” de la imagen. A mi entender los filtros crean “piezas” que son los elementos que se combinan para obtener el resultado de la red, el proceso de optimización crea las piezas que mejor modelan el problema.

5. Superando las imágenes MNIST

Necesitamos un conjunto de imágenes que no estén tan bien construidas como las del conjunto MNIST para ver a las RNA convolucionales en acción. Aunque en internet no faltan colecciones tales como conjuntos de perros y gatos, esta vez las construiremos nosotros mismos.

He preparado una aplicación que dibuja tres figuras geométricas sencillas: círculos, cuadrados y triángulos, en cada imagen sólo hay una de estas figuras pero puede ser de cualquier tamaño (que quepa) y giradas cualquier ángulo (cosa que no afecta al círculo).

Para mayor variedad de ubicaciones he elegido imágenes más grandes que las MIIST, en concreto de 64x64 píxeles que sólo pueden ser blancos o negros. La figura 15 muestra algunos ejemplos de la colección creada, que llamaré conjunto CIRCUATRI. Esta colección contiene 36.000 imágenes: 12.000 círculos y otros tantos cuadrados y triángulos. La vamos a dividir en 6.000 para entrenar y 30.000 para probar, esto es totalmente atípico ya que para probar nunca se reserva más del 40% del total de observaciones disponibles, pero el principal objetivo es que la RNA no pueda memorizar y se esfuerce en encontrar un patrón. Con el espacio reducido de una superficie de 64x64, la posibilidad de que las figuras se repitan, si no exactamente, si en lugares y tamaños próximos no es despreciable. Por la misma razón, la muestra de prueba es

cinco veces mayor que la de entrenamiento para que si se produce sobre ajuste (la RNA memorice) los resultados con la muestra de prueba difieran mucho de los del entrenamiento.

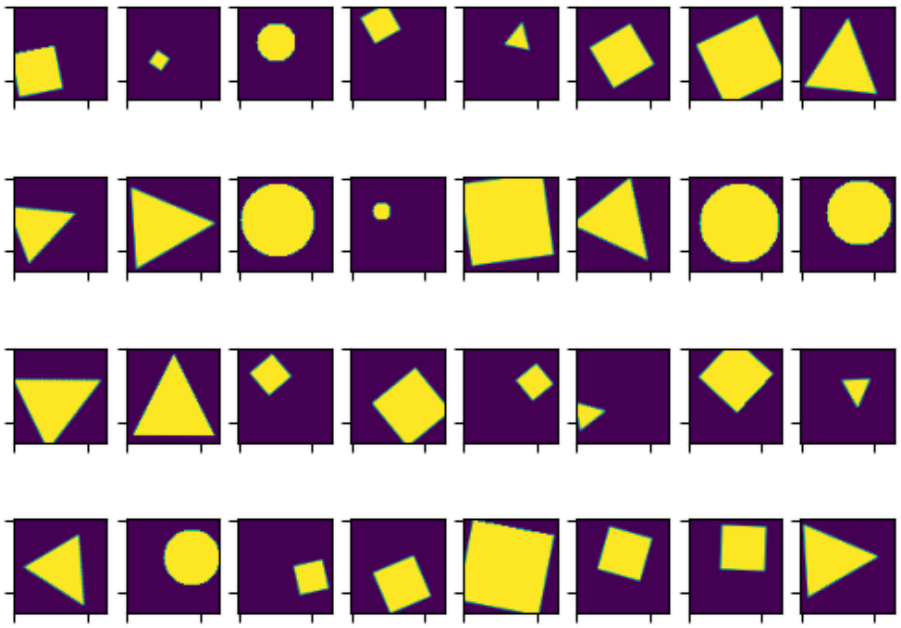

Figura 15. ·2 imágenes 64x64 elegidas al azar de la colección CIRCUATRI que contiene 36.000 imágenes.

Como ya hicimos con los dígitos, no estamos interesados en que la red nos diga que figura es la que contiene la imagen, sólo si esa figura es un círculo, o no lo es, para facilitar el análisis de los resultados.

Inicialmente vamos a probar una red densa de las sencillas (sin capas intermedias), como la utilizada con los dígitos con la única diferencia del número de entradas, que en lugar de 784 (28x28), será de 4.096 (64x64). La tabla 5 muestra los resultados de esta RNA 4096:1

Círculos	: 10,000	resto de formas : 20,000	total 30,000
Falsos positivos	(formas confundidas con Círculos sin serlo)	: 1,675	8.4%
Falsos negativos	(Círculos no reconocidos)	: 5,316	53.2%
Aciertos positivos	(Círculos reconocidos como tal -sensibilidad-)	: 4,684	**46.8%**
Aciertos negativos	("no Círculos " reconocido que no lo son)	: 18,325	91.6%
Aciertos totales		: 23,009	**76.7%**

Tabla 5. Resultados del modelo de reconocimiento de círculos mediante la RNA 4096:1

El grado de acierto (77%) y sobre todo la sensibilidad a los círculos (47%) son bajos, aun así me parece espectacular la capacidad de las RNA teniendo en cuenta que las formas tienen tamaños diversos y en ubicaciones variadas.

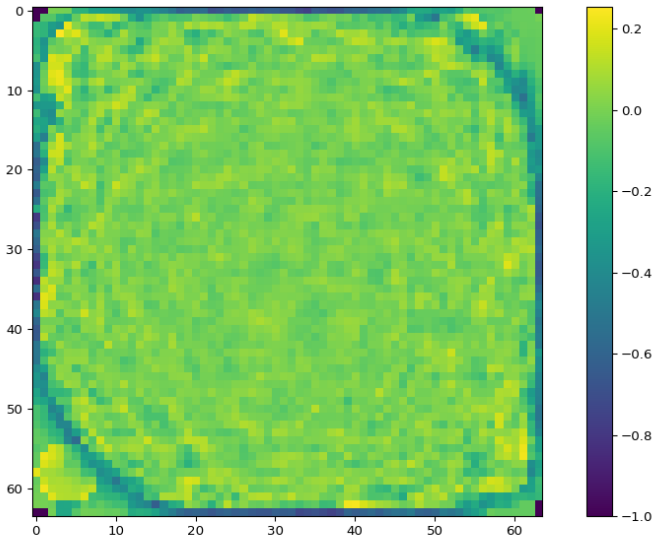

Figura 16. Matriz de pesos desde las entradas de las RNA 4096:1

La matriz de pesos revela que algo tiene que ver con círculos pero poco sobre lo que la RNA ha aprendido, lo contrario me habría sorprendido enormemente, incluso soy incapaz de imaginar la forma que debería tener para que pudiésemos entender porque es un patrón capaz de diferenciar a los círculos, estén donde estén.

Falta utilizar una RNA convolucional que, en definitiva, es para lo que hemos creado el conjunto CIRCUATRI. Hemos construido una RNA convolucional sencilla:

- Una capa convolucional con 32 filtros y un marco de 5x5 para recorrer las imágenes 64x64. Esta capa produce 115.200 salidas.
- Una capa de concentración (pooling) de 2x2. Cada cuatro salidas se concentran en una, eligiendo el mayor valor de las cuatro, aun así quedan 28.800 salidas.
- Las 28.800 salidas de la capa de concentración se vierten sobre una capa final con dos neuronas, esta fase final es una RNA 28800:2. La primera neurona nos da la probabilidad de que la imagen sea de un círculo y la segunda de que no lo sea (la suma de ambos valores siempre es 1)

La tabla 6 muestra los resultados de esta RNA, a la que nos referiremos como CV 32:5:2:2 (32 filtros, un marco de 5x5, concentrador de 2x2 y 2 neuronas de salida)

Círculos : 10,000 resto de formas : 20,000 total 30,000			
Falsos positivos (formas confundidas con Círculos sin serlo)	:	118	0.6%
Falsos negativos (Círculos no reconocidos)	:	209	2.1%
Aciertos positivos (Círculos reconocidos como tal -sensibilidad-)	:	9,791	**97.9%**
Aciertos negativos ("no Círculos " reconocido que no lo son)	:	19,882	99.4%
Aciertos totales	:	29,673	**98.9%**

Tabla 6. Resultados del modelo de reconocimiento de triángulos mediante la CV 32:5:2:2

Un grado de acierto del 99% y una sensibilidad del 98% es espectacular, sin embargo hay que profundizar en lo que realmente es capaz de hacer esta RNA.

Como hemos hecho anteriormente, le hemos pedio a la RNA que nos dibuje un círculo, la respuesta está la figura 18, para ambas imágenes afirma, con un 100% de confianza, que son círculos.

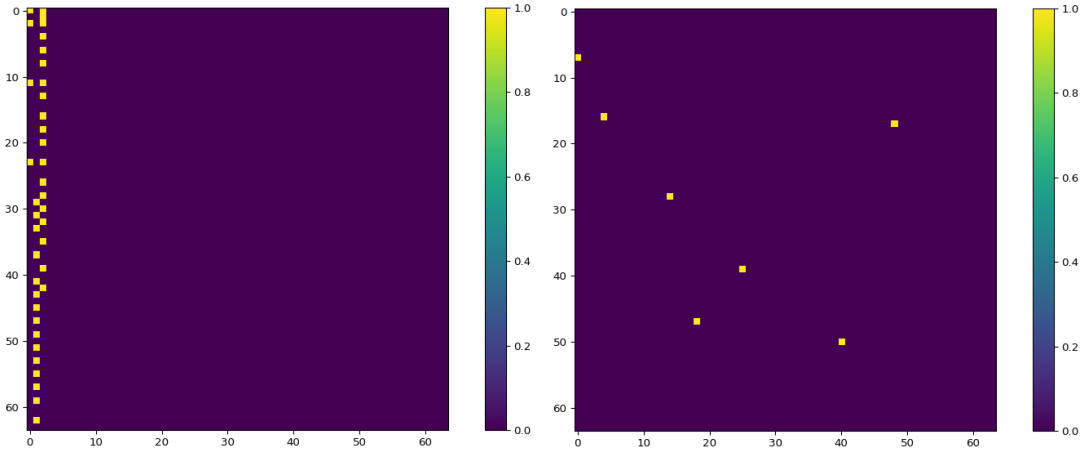

Figura 18. Un círculo dibujado por la CV 32:5:2:2

Le mostramos 30.000 imágenes a La CV 32:5:2:2 y acierta el 99% de las veces, si las imágenes son de un círculo, o si no lo son. Le pedimos que dibuje un ejemplo de círculo y nos responde con la figura 18, es más, si le presentamos a la CV 32:5:2:2 la imagen de la figura 18, nos dirá que corresponde a un círculo con una probabilidad del 100%.

Para despejar dudas, he preparado una muestra de 30.000 imágenes 64x64 que contienen 1.024 puntos distribuidos de forma aleatoria, se ven algunos ejemplos en la figura 19.

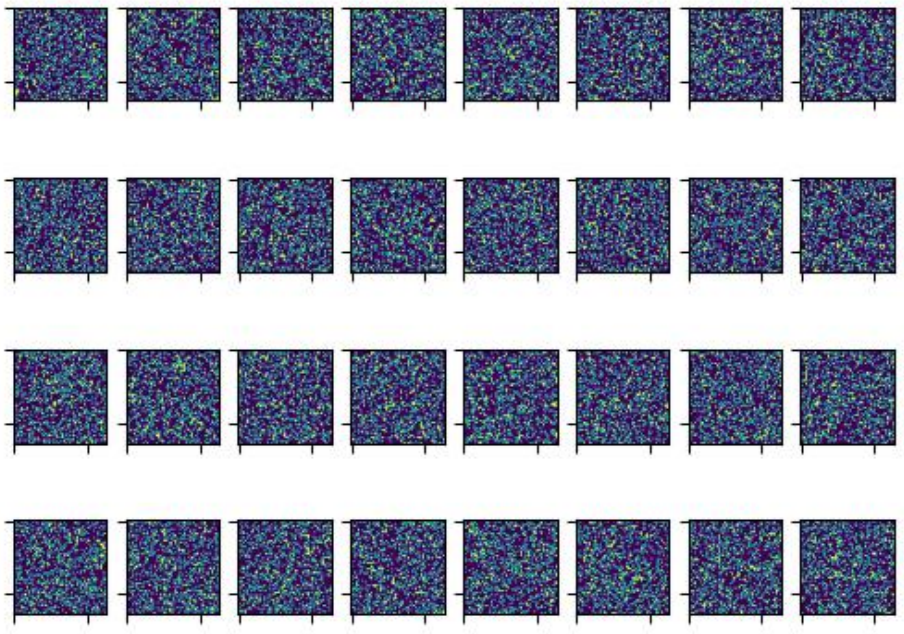

Figura 19. Ejemplo de imágenes aleatorias (ruido)

La CV 32:5:2:2 dice que 29.998 de las 30.000 imágenes de puntos aleatorios son círculos. Por el contrario, las RNA 4096:1 dice que ninguna de las 30.000 imágenes son círculos, además algunos de los círculos que dibuja esta RNA son algo más expresivos, aunque para los dos de la figura 20 les otorga una confianza del 100%.

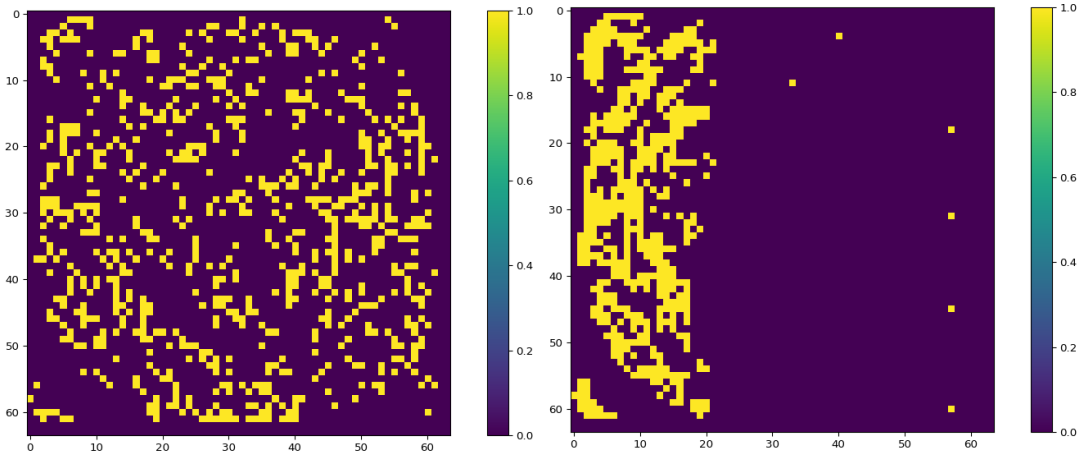

Figura 20. Un círculo dibujado por la RNA 4096:1

El valor de los dibujos es que nos dan pistas sobre lo que la RNA considera importante de las imágenes, no sabremos porque, pero al menos el que.

La CV 32:5:2:2 podría ser útil para separar las piezas defectuosas (círculos imperfectos) de una máquina de troquelar círculos, pero no serviría lo más mínimo para detectar si una imagen contiene un círculo.

6. El problema es el contexto.

El contexto de la RNA es aquello que es común a todas las imágenes del problema, tanto a las imágenes con las que se entrena como a las que servirán para probarla. Que la RNA obtenga un grado de acierto con las imágenes de prueba (que no ha visto durante el entrenamiento) significa que ese contexto existe y que es capaz de modelarlo, en gran parte gracias a que tiene total libertad para ignorar lo que pasa fuera del contexto. Cuando le pedimos que nos pinte un círculo la ponemos en evidencia porque los círculos son la tercera parte de las imágenes de CIRCUATRI y las infinitas imágenes de otros tipos, como las de las figuras 18 y 19. El contexto de la RNA 4096:1 es más amplio, por eso acierta menos, pero a cambio, no confunde las imágenes aleatorias con círculos y los dibuja mejor.

Hace unos años, fue noticia un incidente con un coche de Tesla conducido por una IA. En el coche viajaba un ingeniero de Tesla que se llevó un buen susto. Después de miles de kilómetros conducidos satisfactoriamente por la IA, al pasar por un puente el coche empezó a derivar hacia un costado, y sólo en el último momento el ingeniero pudo hacerse son los mandos, justo cuando estaba a punto de despeñarse. Tras estudiar el problema, la conclusión fue que la IA siempre había conducido por carreteras sobre tierra firma, con márgenes de tierra, hierba o construcciones, pero el vacío que rodea al puente le hizo responder de forma totalmente inesperada, no formaba parte de su contexto (esto último lo digo yo).

Una red neuronal antigua pero muy distinguida (Isaac Newton) encontró la fórmula de la gravitación universal, pero también algo mucho más disruptivo para la época (hoy ni se menciona, se da por hecho): el universo se puede describir con las matemáticas, las matemáticas son el contexto de nuestras redes neuronales (las que llevamos puestas). A partir de entonces, todo lo fue confirmando, alcanzando su apogeo cuando las redes neuronales Einstein y Minkowski presentaron y formalizaron la teoría general de la relatividad, es posible, que sólo unos años después, la mecánica cuántica sacara de contexto a nuestras redes neuronales.