3. MEMORIA DEL PROYECTO

En la etapa de documentación de este proyecto se observa que hay numerosos métodos y mecanismos bioinspirados de aprendizaje que tratan de emular (con mayor o menor éxito) el procesamiento biológico existente en el cerebro. Se pueden destacar como sistemas más exitosos aquellos basados en redes neuronales que utilizan convoluciones para el procesamiento de la señal. Dentro de este grupo destacan aquellos que usan el algoritmo de entrenamiento *backpropagation* propuesto por Yann LeCun [1]. Este mecanismo es muy eficiente a la hora de desarrollar tareas de reconocimiento y seguimiento de objetos. Además, este sistema es capaz de auto-corregir la salida experimental con la deseada. Para ello, realiza los ajustes necesarios en los pesos asociados a las sinapsis para conseguir que el resultado empírico se acerque cada vez más al buscado.

Los mecanismos que utilizan el algoritmo *backpropagation* pertenecen al grupo de sistemas cuyo aprendizaje es *supervisado*. Este método de aprendizaje consiste en un entrenamiento en el que se introducen repetidas veces un determinado patrón, de modo que las respuestas que va dando el sistema se van comparando con las respuestas deseadas. En virtud de estas comparaciones, se reajustan los pesos. El error (diferencia entre la respuesta obtenida y la deseada) se propaga hacia atrás con la misión de modificar los pesos asociados a las neuronas previas. Con este reajuste de la fuerza de las conexiones la red se acerca cada vez más a la respuesta correcta. En la Fig. 8 se presenta un esquema que muestra cómo funcionan los sistemas de aprendizaje supervisados.

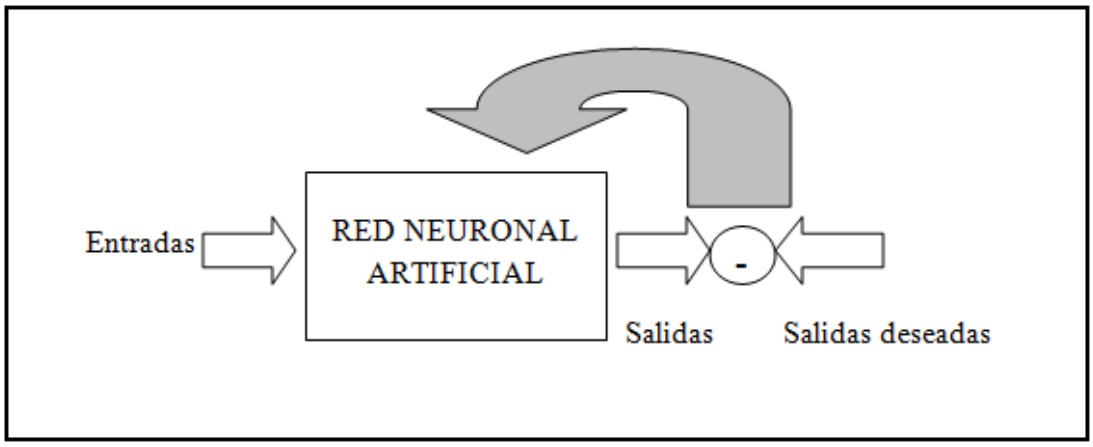

Figura 8. Sistema de aprendizaje supervisado.

Aun existiendo todas las ventajas enumeradas este mecanismo es insuficiente, ya que al ser un sistema de aprendizaje *supervisado* es necesario conocer las entradas y las salidas deseadas en todo momento para poder así minimizar el error. Podemos solucionar este problema utilizando un método de aprendizaje que sea *no supervisado*. Por ello, en nuestro proyecto utilizamos un sistema bioinspirado basado en el mecanismo *no supervisado* STDP (*Spike Timing Dependent Plasticity*) desarrollado por T. Masquelier, R. Guyonneau y S. J. Thorpe [23]. Este mecanismo es capaz de ajustar los pesos sinápticos teniendo solo en cuenta la relación entre los pulsos de entrada y de salida, todo ello sin conocer la salida deseada. Esto significa que no hay información externa que ayude a decidir cómo modificar los pesos de las conexiones. Ha de ser la red la que vaya adecuando los pesos en función de la información interna que vaya recogiendo de las entradas. La modificación de los pesos asociados a las sinapsis modela el proceso de aprendizaje o memorización de la neurona. Cuando la sinapsis se activa, su peso contribuye a la alteración del potencial de la neurona de salida. Por tanto, los sistemas basados en STDP son capaces de implementar el aprendizaje de manera autónoma. Hasta ahora sigue siendo una de las hipótesis con mayor acogida por toda la comunidad científica. En la Fig. 9 se representa un esquema general de los sistemas de aprendizaje *no supervisado*.

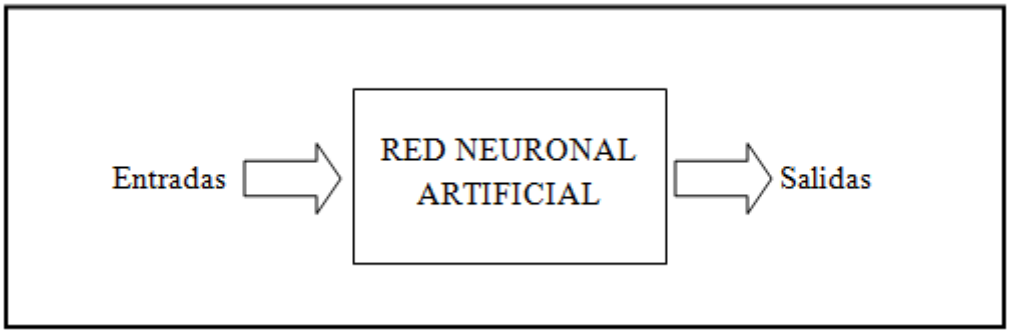

Figura 9. Sistema de aprendizaje *no supervisado*.

En el presente trabajo se utiliza STDP para implementar un mecanismo artificial bioinspirado de aprendizaje. En concreto, nos vamos a basar en el reconocimiento de los diez dígitos numéricos (0 a 9).

En nuestro sistema podemos diferenciar claramente dos etapas:

- La primera etapa se compone de un banco de filtros de Gabor, que está bastante aceptado como una de las primeras etapas del procesamiento en el cerebro. Sus principales tareas son la captación y el acondicionamiento de la señal de entrada.
- La segunda etapa es una red neuronal de una etapa, compuesta por diez neuronas de salida. Cada neurona ha sido entrenada por separado usando STDP.

Aplicaremos los principios de la plasticidad sináptica en nuestro sistema (funcionamiento de las neuronas cuando estas establecen comunicación, modulando la percepción de los estímulos con el medio), ya que la comunidad científica cree que el aprendizaje y almacenamiento de información subyace en la modulación de los circuitos implementados por las sinapsis. Debido a estos circuitos neuronales el sistema nervioso es capaz de transportar el estímulo hasta la neurona. Es posible que el conjunto pueda cambiar de recorrido si el sistema deja de recordar un determinado patrón aprendido con anterioridad, llevando el estímulo a otra neurona diferente. Para realizar este cambio de recorrido se activarían otros circuitos mientras que otros pasarían a un estado de inactividad [23].

3.1. TIPOS DE SISTEMAS DE CAPTURA DE INFORMACIÓN VISUAL

Los sistemas artificiales de visión funcionan de forma diferente a lo que ocurre en el sistema nervioso visual. Por lo general, los sistemas artificiales se dedican a capturar secuencias de fotogramas a las que se les aplican operaciones de convolución, necesarias para alcanzar el reconocimiento deseado de la imagen captada. Este procesamiento mediante convoluciones de fotogramas es lento y costoso computacionalmente, ya que se necesitan realizar muchas convoluciones en secuencia para cada imagen de entrada. En cambio la manera de operar del sistema visual es distinta, los pixeles de la retina envían pulsos a la corteza cada vez que un evento captado alcanza un nivel de actividad superior a un umbral. Todos estos pulsos son transmitidos a medida que están siendo producidos, sin esperar a un tiempo estipulado para la transmisión del fotograma [3].

Diversos estudios representan el sistema nervioso visual estructurado jerárquicamente en capas [5]. Esto simplifica bastante las operaciones computacionales a realizar, ya que cada capa tiene asignada una misión a realizar. Aun así, sigue siendo necesaria la realización de múltiples convoluciones que pueden ser excesivas para los programas de software basados en fotogramas que se ejecutan hoy en día.

Según novedosos estudios la solución más adecuada para resolver el problema del coste computacional vendrá de la mano de un mayor desarrollo del hardware, combinado con un sistema dividido en capas que utilice mecanismos basados en eventos [3] [12]. De esta manera aumentaremos la rapidez en el procesamiento de la señal. Por tanto, el retardo en el tratamiento del evento de entrada dependerá principalmente del número de capas, y no de la complejidad de los objetos y formas para ser reconocido.

3.1.1. SISTEMAS DE PROCESAMIENTO BASADOS EN EVENTOS FRENTE A LOS BASADOS EN FOTOGRAMAS

Para mostrar la velocidad de procesamiento de los sistemas basado en eventos (*Event-Based*), en comparación con los sistemas basado en fotogramas (*Frame-Based*) consideremos la Fig. 10. En ella se ilustra de manera conceptual la diferencia entre un sistema de captura y procesado basado en fotogramas y un sistema basado en eventos (bioinspirado), a la hora de la detección y procesamiento de la entrada. Cada uno utiliza una cámara diferente para captar la realidad.

En la imagen superior, una cámara basada en la captura de fotogramas percibe una secuencia de imágenes, cada una de las cuales se transmite al sistema de procesado. Cada fotograma es tratado por sofisticados algoritmos de procesamiento de imagen para lograr algún tipo de reconocimiento. El sistema informático tiene que tener todos los valores de los píxeles de un fotograma antes de comenzar cualquier cálculo.

En la imagen inferior, un sensor de visión basado en eventos opera sin tener en cuenta el fotograma completo. Cada píxel envía un evento, por lo general codificando una coordenada *(x, y)*, cuando detecta algo. Normalmente el pixel realiza la acción de envío cuando detecta un cambio de intensidad con respecto a los píxeles vecinos [24] [25]. Los eventos se envían al sistema de procesado, sin esperar a un tiempo marcado por el fotograma detectado. El sistema informático actualiza su estado después de cada evento [3].

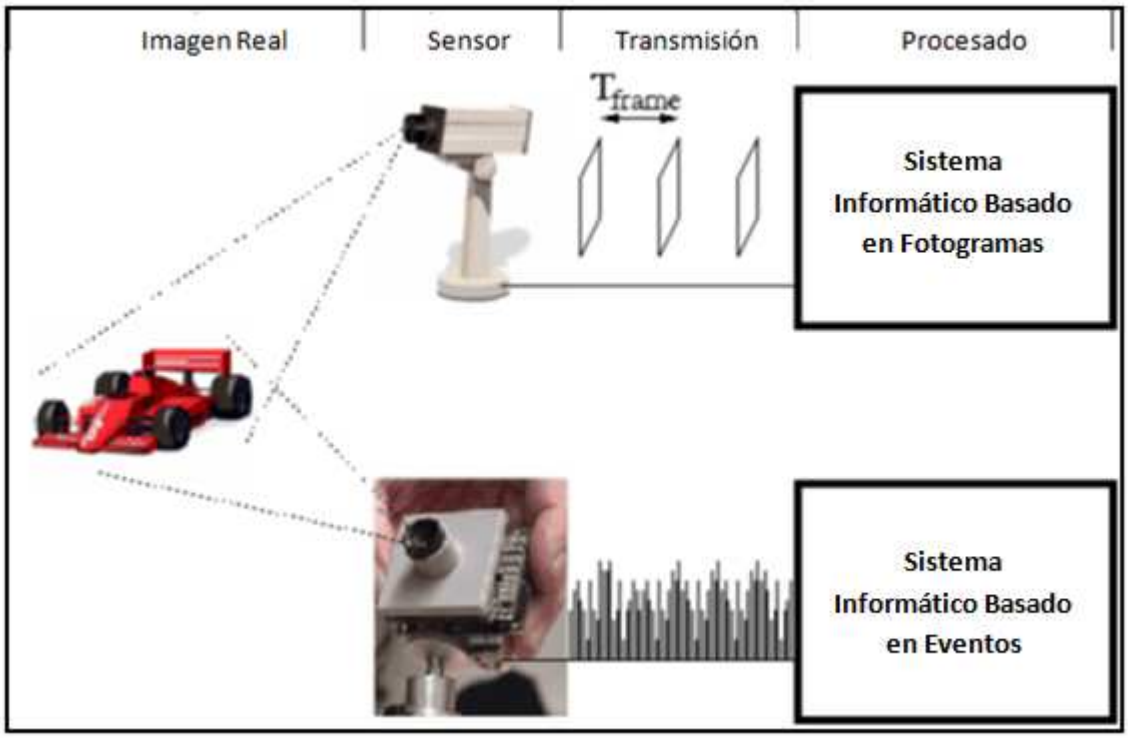

Figura 10. Comparación de los mecanismos de captación y procesado en sistemas basados en fotogramas y en eventos.

La Fig. 11 ilustra la diferencia de tiempo de procesado y detección entre ambos conceptos. En la parte superior (basada en fotogramas), la secuencia de imágenes es tramitada fotograma a fotograma, con un tiempo de detección igual a T_{frame}. Durante el primer fotograma *T1*, la información producida por este método no llega al sistema computacional hasta que el fotograma completo es capturado y transmitido. Así que hay que sumarle un retraso adicional ($T1+\Delta$). El sistema computacional tiene que procesar la imagen completa, por tanto requiere el manejo de una gran cantidad de datos provocando un alto tiempo de detección. Como se aprecia en la imagen, T_{FC} será el tiempo necesario para procesar la información y que se produzca algún tipo de reconocimiento. En la parte inferior de la Fig. 11, los píxeles de la retina "ven" directamente la escena y generan eventos con pequeños retrasos. Los eventos son procesados a medida que van llegando con un pequeño retraso, debido a la acción de cálculo por parte del sistema. Dicho retraso es representado en la figura como T_{ev} (algunos ns [26]). Para llevar a cabo el reconocimiento no todos los eventos son necesarios. En realidad los acontecimientos más relevantes, por lo general, vienen en primer lugar o con mayor frecuencia. Por consiguiente, el tiempo de reconocimiento T_{rcg} puede ser menor incluso que el tiempo total de los eventos producidos. Nótese que el

reconocimiento es posible antes del tiempo T_1, lo que resulta un T_{FC}' negativo en comparación con el retraso surgido en el reconocimiento de un sistema basado en fotogramas [3].

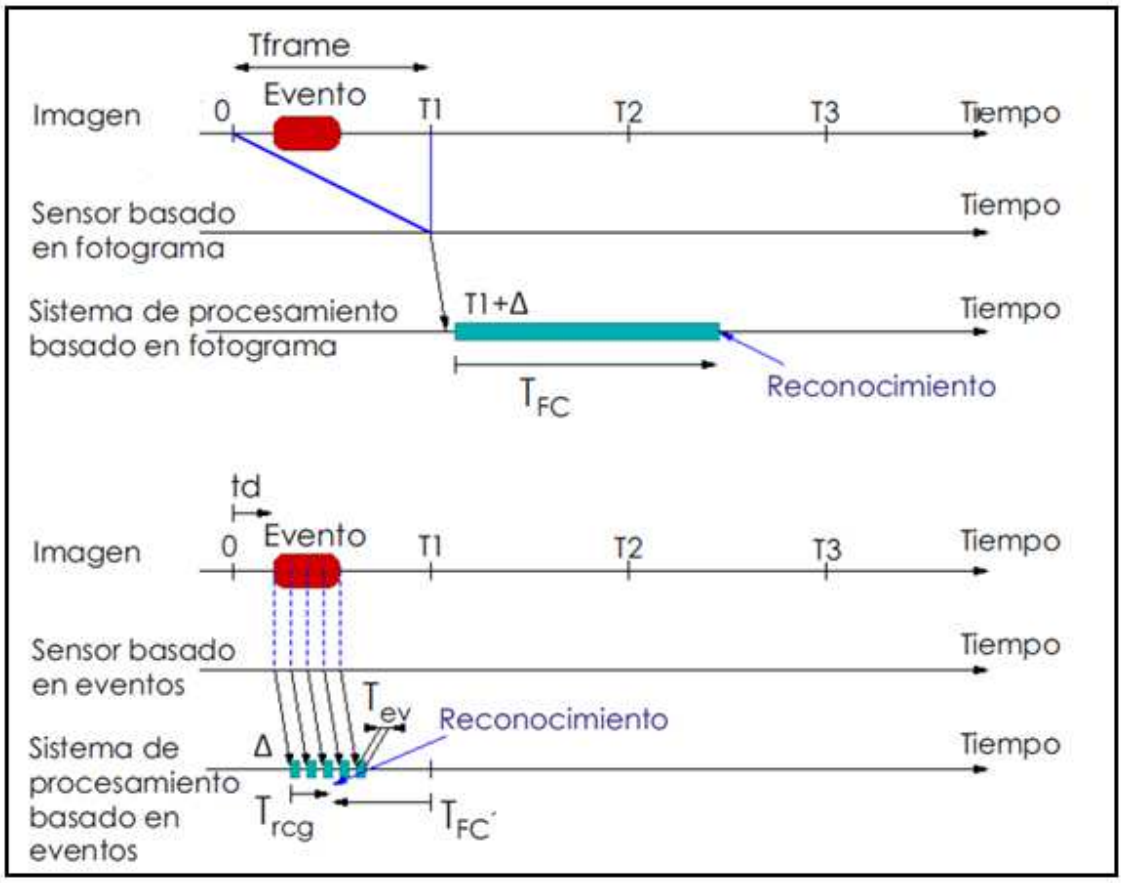

Figura 11. Comparación entre el tiempo de procesado y captación para sistemas basados en fotogramas y en eventos**.**

Por lo desarrollado anteriormente y dado que pretendemos implementar un sistema ligeramente bioinspirado, podemos concluir que el sistema más útil para nuestro proyecto será el basado en eventos, ya que su retraso es bastante menor. A todo esto podemos añadir que el coste computacional es también menor, ya que no es necesario procesar todos los eventos producidos para producir los resultados deseados.

3.1.2. ESQUEMAS DE CODIFICACIÓN PARA SISTEMAS BASADOS EN EVENTOS

La neurociencia, informática y los investigadores de ingeniería de aplicaciones han desarrollado y utilizado varios esquemas de codificación de la información. Vamos a resumir dos tipos de codificaciones que son las que más se han utilizado hasta ahora: la basada en la tasa de los eventos (*Rate Coding*) y la basada en el orden de los eventos (*Rank Order Coding*). Desarrollando estos esquemas explicaremos por qué nos quedaremos con el segundo.

3.1.2.1. CODIFICACION BASADA EN LA TASA DE LOS EVENTOS

La codificación basada en eventos (*Rate Coding*) es un esquema tradicional de codificación, donde la información está contenida en la tasa de eventos que estimulan la neurona. En la mayoría de los sistemas sensoriales la tasa de disparo aumenta de forma no lineal con el incremento de la intensidad del estímulo [27]. Este concepto ha sido aplicado con éxito durante los últimos 80 años y su origen se remonta al trabajo realizado por Adrian [28]. En dicho trabajo mostró que la tasa con la que disparan las neuronas receptoras, que son responsables del estiramiento en los músculos, está relacionada con la fuerza aplicada al músculo. En las décadas siguientes, la medición de las tasas de disparo se convirtió en una herramienta estándar para describir las propiedades de todos los tipos de neuronas sensoriales. Ésto fue así debido a la relativa facilidad de medición de estas tasas de forma experimental. Sin embargo, este enfoque deja de lado toda la información posible relacionada con el momento exacto de llegada de un impulso. Durante los últimos años estudios experimentales han sugerido que el concepto de tasa de disparo (basado en la tasa promedio temporal de acción en la neurona receptora) puede ser demasiado simplista para describir la actividad del cerebro [29]. Esto es así porque la codificación basada en eventos no tiene en cuenta el orden con el que llegan los estímulos externos, que son responsables de que la neurona detectora dispare con una determinada tasa.

3.1.2.2. CODIFICACIÓN BASADA EN EL ORDEN DE LOS EVENTOS

Si bien la idea de tratar una imagen usando una codificación en tasa puede parecer plausible, recientes trabajos experimentales en la retina de un ratón han hecho que se descarte. La cantidad de información contenida en la tasa de disparo de las neuronas puede ser insuficiente para explicar el comportamiento del animal [30]. En los esquemas de codificación basados en el orden de los eventos (*Rank Order Coding*), propuesto originalmente por Thorpe [31], la información está codificada en el orden relativo de llegada de los estímulos a la población de neuronas que se ven envueltas en el experimento (ver también [32]). En respuesta a un estímulo visual células de la retina se cargarán progresivamente hasta alcanzar un umbral. Ésto provocará que la neurona genere un pulso de salida como respuesta al desbordamiento del acumulador. Gracias a este disparo la neurona de salida es capaz de liberar toda la energía retenida, reiniciando así el proceso. Diversas simulaciones han demostrado que mediante el uso del orden en que disparan las neuronas que captan la imagen, es posible reconstruir el objeto identificado [33]. Ésto se ha aplicado en el estudio de la retina de la salamandra [34]. Por tanto, queda demostrado que se puede conseguir la plasticidad sináptica mediante la modulación según el orden de los eventos.

Los sistemas de codificación basados en la tasa de los eventos y el orden de los eventos se comparan en la Fig. 12. En la parte superior de la imagen, se muestran los estímulos recibidos por tres neuronas diferentes (etiquetadas como NA, NB y NC). Como se puede observar en la codificación basada en la tasa de los eventos (en la parte inferior de la ilustración), las actividades de salida de las tres neuronas varían con la intensidad del estímulo. En este esquema no es necesario utilizar señales de referencia temporales. Por el contrario, en la codificación basada en el orden de los eventos (parte central de la imagen), la información se codifica en el orden relativo en que las tres neuronas disparan. Esta vez sí es necesaria una señal de referencia temporal para indicar el inicio de nuevas ventanas de tiempo. Sin embargo, el número de pulsos de entrada requeridos es menor y por lo tanto requiere un ancho de banda menor. Gracias a este método el coste computacional del sistema disminuye en gran medida, ya que solo tiene en cuenta el momento en el que dispara cada neurona y el orden relativo de disparo entre ellas. Aunque se recorta en procesamiento, el sistema tiene que estar bien sincronizado, para que no se solapen disparos de neuronas que se han producido en slots

de tiempos diferentes. Por ser el sistema más parecido al implementado en el sistema visual humano y además ser más eficiente en ancho de banda y cantidad de información, el sistema de codificación utilizado en este proyecto es el basado en el orden de los eventos [3].

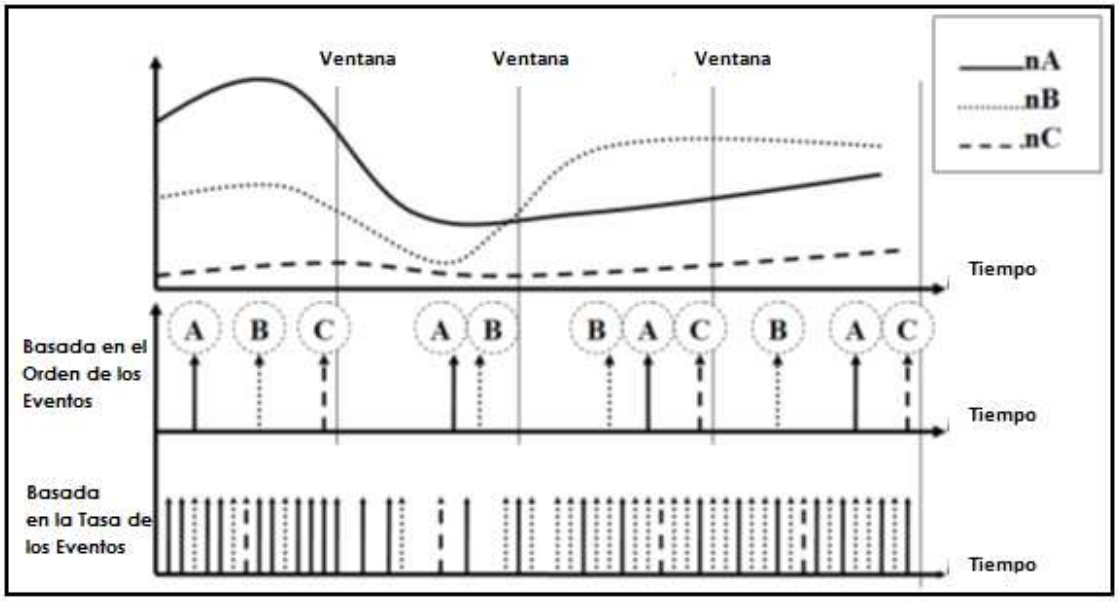

Figura 12. Esquemas de codificación basados en la tasa de los eventos frente a los basados en el orden de los eventos.

3.1.3. APLICACIÓN DEL PROTOCOLO AER PARA SISTEMAS BASADOS EN EVENTOS

Como hemos comentado anteriormente se ha elegido como mecanismo más válido para nuestro proyecto el sistema basado en eventos utilizado con una codificación según el orden en que tienen lugar los eventos, ya que nos proponen una serie de ventajas que se asemejan a la forma de trabajar del sistema nervioso visual. Esta elección no serviría para nada si la tecnología actual no permitiera la implementación de aplicaciones complejas a alta velocidad con una elevada tasa de resultados correctos.

Los sistemas de captación basados en eventos trabajan de manera similar a la retina, cada píxel envía pulsos a la corteza cerebral cuando su nivel de actividad alcanza cierto umbral. Aquellos píxeles muy activos enviarán más pulsos que los menos activos. Todos estos pulsos son transmitidos a medida que están siendo producidos, y no esperan a un tiempo artificial (*tiempo de frame*) antes de enviarlos a la siguiente etapa de procesamiento [4]. Las características extraídas son propagadas y procesadas etapa por etapa tan pronto como han sido producidas, sin esperar a finalizar la recolección y procesamiento de los datos de fotogramas completos. Un problema importante que encuentran los ingenieros cuando tratan de implementar sistemas de procesamiento de visión bioinspirados es conseguir la masiva cantidad de interconexiones hacia delante y de realimentación que aparece entre las etapas neuronales existentes en el sistema de procesamiento de visión humano. La representación de datos basada en direcciones de eventos AER (*Address Event Representation* [35] [36]) es una posible solución. Esta solución fue propuesta por primera vez en 1991 en el Instituto de tecnología de California (Caltech). Este sistema se ha utilizado desde entonces por una amplia comunidad de ingenieros de hardware neuromórficos. La Fig. 13 ilustra la comunicación en un enlace punto a punto AER tradicional cuando se utiliza codificación basada en tasa de eventos [3].

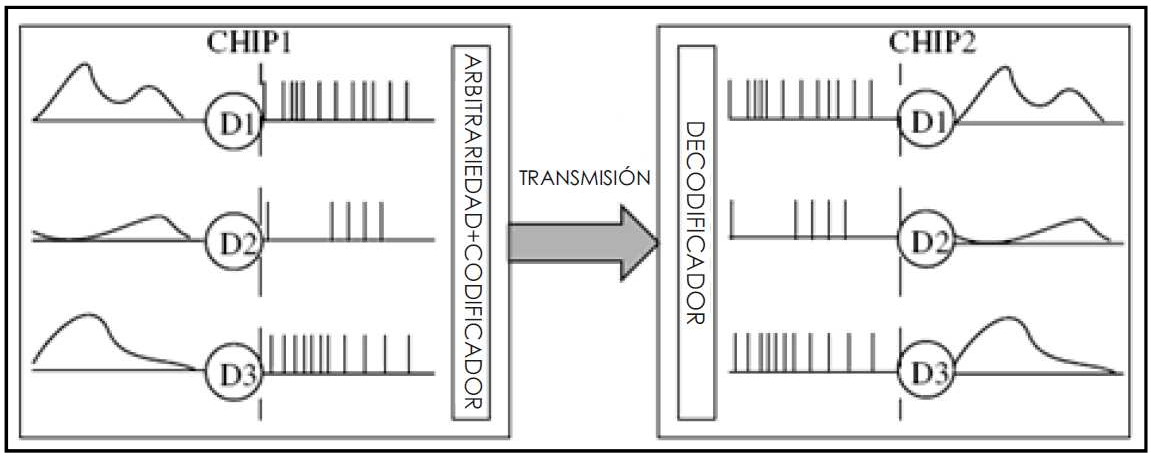

Figura 13. Comunicación interchip punto a punto aplicando protocolo AER.

En el sistema más simple AER y basado en una codificación "rate-based", el estado continuo en el tiempo de las neuronas emisoras en un chip es transformado a una secuencia de pulsos digitales muy rápidos (eventos) de anchura mínima (del orden de ns) pero con intervalos entre pulsos del orden de ms (similar a las neuronas cerebrales). Este alto intervalo entre pulsos permite una potente multiplexación, y los pulsos generados por las neuronas emisoras pueden ser multiplexados en tiempo en un bus de salida común de alta velocidad. Cada vez que una neurona emite un pulso o evento, la dirección de esa neurona aparece en el bus digital junto con sus señales de establecimiento de la comunicación (*request* y *acknowledgede*). Esto se conoce como evento de dirección. El chip receptor lee y decodifica las direcciones de los eventos entrantes y envía pulsos a las neuronas receptoras correspondientes, que integran esos pulsos y son capaces de reproducir el estado de las neuronas emisoras. Esta es la comunicación entre chips basada en AER más simple. Sin embargo, esta comunicación punto a punto puede ser extendida a un esquema multi-emisor o multi-receptor [37], donde rotaciones, traslaciones o procesamientos más complicados como convoluciones pueden ser implementados por chips de procesamiento que reciben estos eventos [38]. Además, la información puede ser trasladada o rotada fácilmente simplemente cambiando las direcciones de los eventos al tiempo que viajan de un chip al siguiente. Existe una creciente comunidad de usuarios del protocolo AER para el diseño de aplicaciones de visión y audición bioinspiradas, robótica, seguimiento y reconocimiento de objetos, etc., como ha sido demostrado por el éxito en los últimos años de los participantes en las 'Neuromorphic Engineering Workshop series' [39]. El éxito de esta

comunidad es diseñar sistemas grandes jerárquicamente estructurados multi-chip multi-etapa capaces de implementar procesamientos complejos de matrices en tiempo real. El éxito de tales sistemas dependerá en gran medida de la disponibilidad de herramientas robustas y eficientes de diseño y depuración de sistemas AER [37] [40].

3.2. IMPLEMENTACIÓN

En este punto se van a explicar todos los elementos intervinientes en este proyecto y además se va a desarrollar la base teórica del mecanismo bioinspirado de aprendizaje STDP. Con todo ello, hemos desarrollado una simulación de un sistema artificial de reconocimiento de dígitos a través de programa del cálculo Matlab.

3.2.1. DESCRIPCIÓN DEL SISTEMA

El sistema implementado en este trabajo está compuesto principalmente por dos etapas, como se puede observar en la Fig. 14. La primera de ellas es un banco de filtros de Gabor (1 escala y 6 orientaciones) y la segunda una red neuronal mono-etapa compuesta por diez neuronas de salida. El nexo de unión entre las dos etapas son las sinapsis, que actúan como auténticos circuitos neuronales que conducen los impulsos detectados. Las imágenes de salida de los filtros de Gabor (salida de la primera etapa) están formadas por píxeles, y cada uno de ellos tiene una conexión (sinapsis) con cada una de las neuronas de salida. Según las uniones que se activen nuestro sistema reconocerá una imagen u otra. Teniendo en cuenta esto podemos diferenciar dos etapas [23]:

- Acción Temporal Presináptica: La acción ocurre en la primera etapa de nuestro sistema. Los píxeles a la salida de los filtros de Gabor envían pulsos cada vez que un evento es captado. A su vez, dichos pixeles realizan una codificación intensidad-tiempo de la señal percibida antes de transmitirla en forma de pulso a través de las sinapsis.
- Acción Temporal Postsináptica: La acción la lleva a cabo una neurona final conectada a los píxeles mediante las sinapsis. Dicha neurona actúa como un auténtico detector de coincidencia, sólo reacciona cuando reconoce una imagen aprendida previamente.

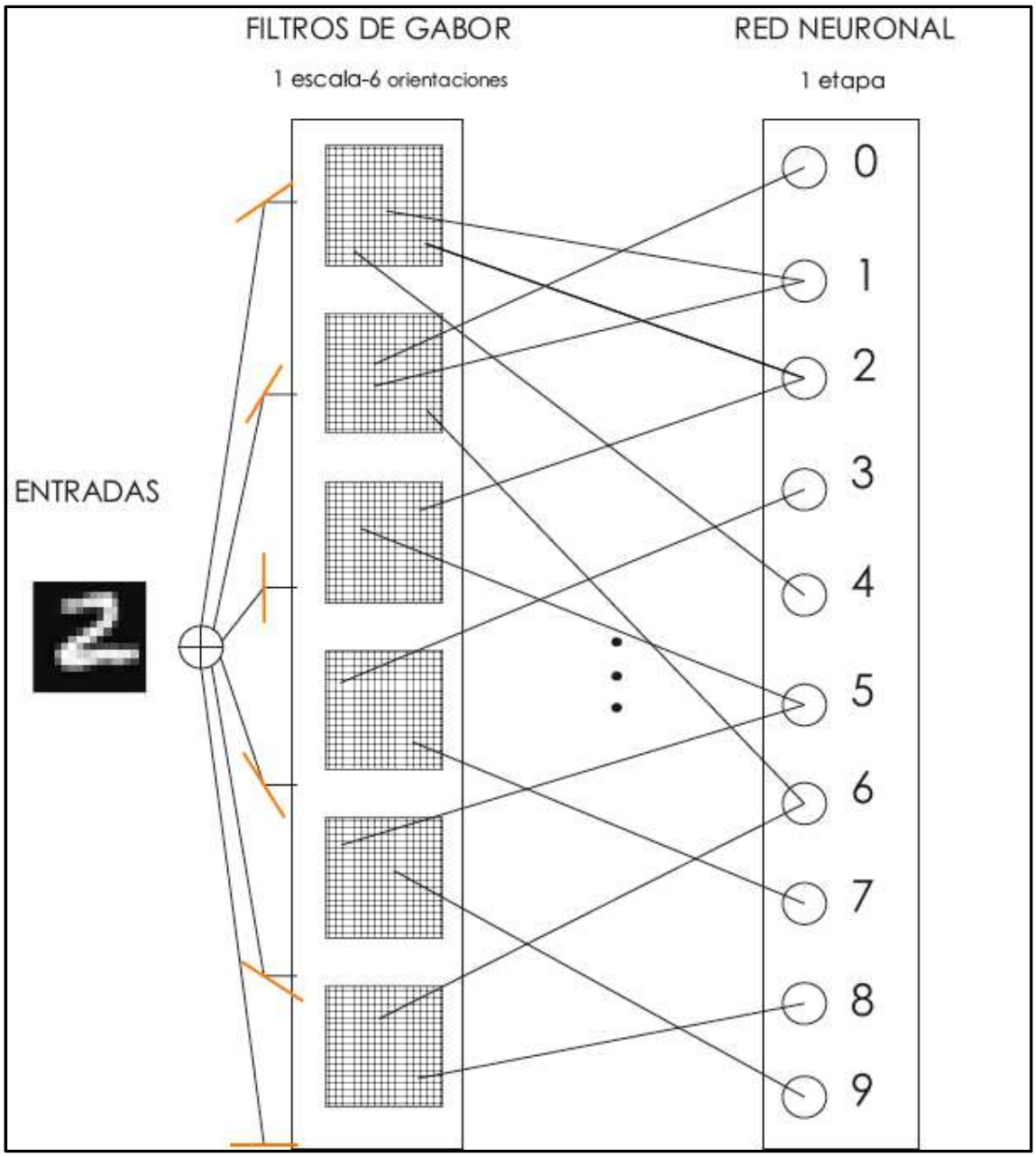

Figura 14. Sistema bioinspirado de aprendizaje de dígitos.

El sistema recibe como entrada dígitos numéricos del rango [0,9], obtenidos de la base de datos MNIST [14]. En la Fig. 15, se muestran ejemplos de los 10 dígitos que representan cada uno de los números naturales, siendo una buena herramienta para el uso en sistemas de reconocimiento y aprendizaje.

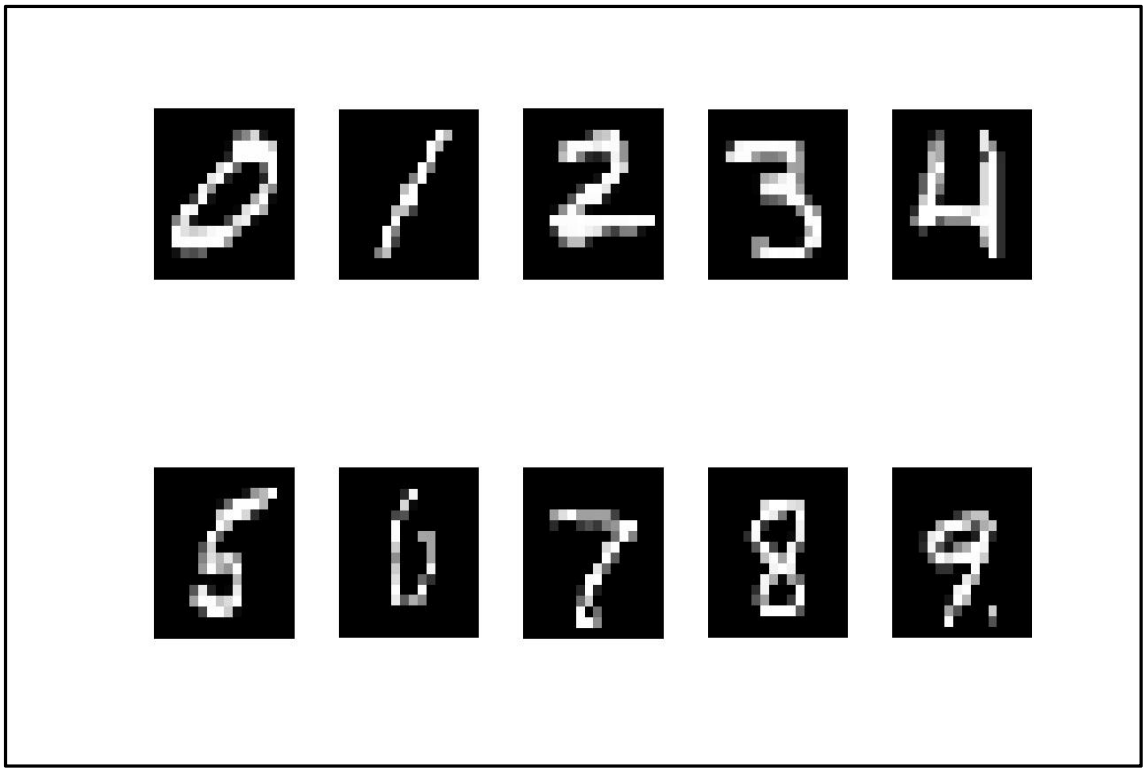

Figura 15. Números naturales de la base de datos MNIST.

El tamaño de las imágenes de entrada es de 32x32, aunque para nuestra aplicación estas imágenes han sido submuestradas para tener un tamaño de 16x16. Dichas imágenes son convolucionadas con un conjunto de 6 filtros de Gabor, que implementan una escala y 6 orientaciones. El tamaño de los filtros de Gabor es 10x10. Cada uno de estos filtros se obtiene mediante las siguientes expresiones [3][41]:

$$g\ (x,y) = \frac{1}{2\pi \cdot \sigma_x \sigma_y} \cdot exp\left(-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right) + \ j2\pi \cdot Wx\right)$$

Realizando unas determinadas dilataciones y rotaciones a la función de transferencia *g(x, y)* obtenemos cada filtro. Cada uno codifica una orientación:

$$g_{s,k}(x,y) = \ a^{-s} \cdot g(x', y')$$

$$x' = \ a^{-s}(x \cdot \cos\theta + y \cdot \sin\theta)$$

$$y' = \ a^{-s}(-x \cdot \sin\theta + y \cdot \cos\theta)$$

En estas expresiones θ representa la orientación y S la escala. Los parámetros $\{a, \theta, W, \sigma_x, \sigma_y\}$ son descritos en el método implementado por Manjunath [42]. En la Fig. 16 podemos observar gráficamente las orientaciones de los filtros de Gabor obtenidos a través de las dilataciones y rotaciones mencionadas anteriormente.

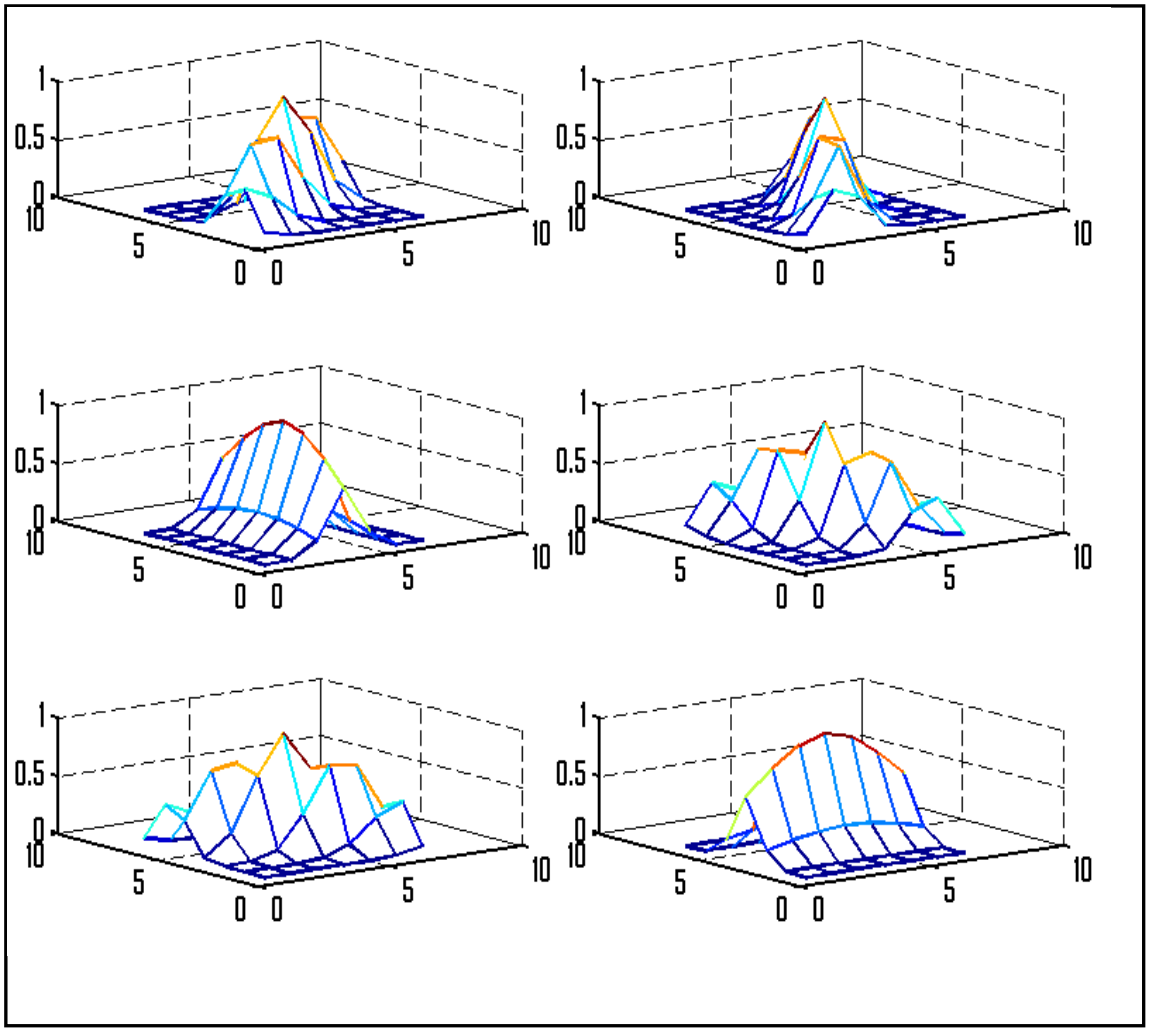

Figura 16. Representación de las 6 orientaciones de los filtros de Gabor.

Para eliminar información redundante y reducir parte del cálculo numérico, se han realizado una serie de simplificaciones en nuestra subrutina codificada en el programa Matlab. Buscando eliminar cualquier elemento superfluo, tras cada convolución de la imagen de entrada con cada filtro de Gabor hemos reducido su tamaño a una matriz 10x10. Se ha comprobado experimentalmente que este conjunto de

píxeles contiene la mayor parte de la información, ya que en la periferia casi todos los valores son iguales a cero (dichos valores no aportan información).

Con la convolución se busca encontrar cuales son las componentes direccionales más dominantes en cada número. Por ejemplo, para el dígito “1” podemos intuir con facilidad que su componente más potente es la vertical. Por tanto, la convolución entre este dígito y la orientación vertical del filtro de Gabor da como resultado el mayor valor posible. El resultado de la convolución es tan alto debido a la similitud existente. Este tipo de rasgos nos ayudan a diferenciar un dígito de otro distinto. Así que podemos utilizar las diferencias existentes en las componentes espaciales y de orientación de cada dígito para lograr que cada neurona de salida aprenda un determinado patrón [41].

En la Fig. 17 muestra un ejemplo de la operación de convolución entre una imagen de entrada (en este caso el dígito “0”) y las 6 componentes de los filtros de Gabor. El resultado muestra la relación de similitud entre el digito “0” y las 6 orientaciones de los filtros de Gabor. Para este caso se aprecia que el dígito “0” no tiene ninguna componente direccional que destaque sobre las demás, ya que aparecen componentes verticales, horizontales e inclinadas de manera equitativa. Esta características es útil a la hora de clasificar el aprendizaje del dígito “0”, ya que su patrón está compuesto por todas las componentes existentes.

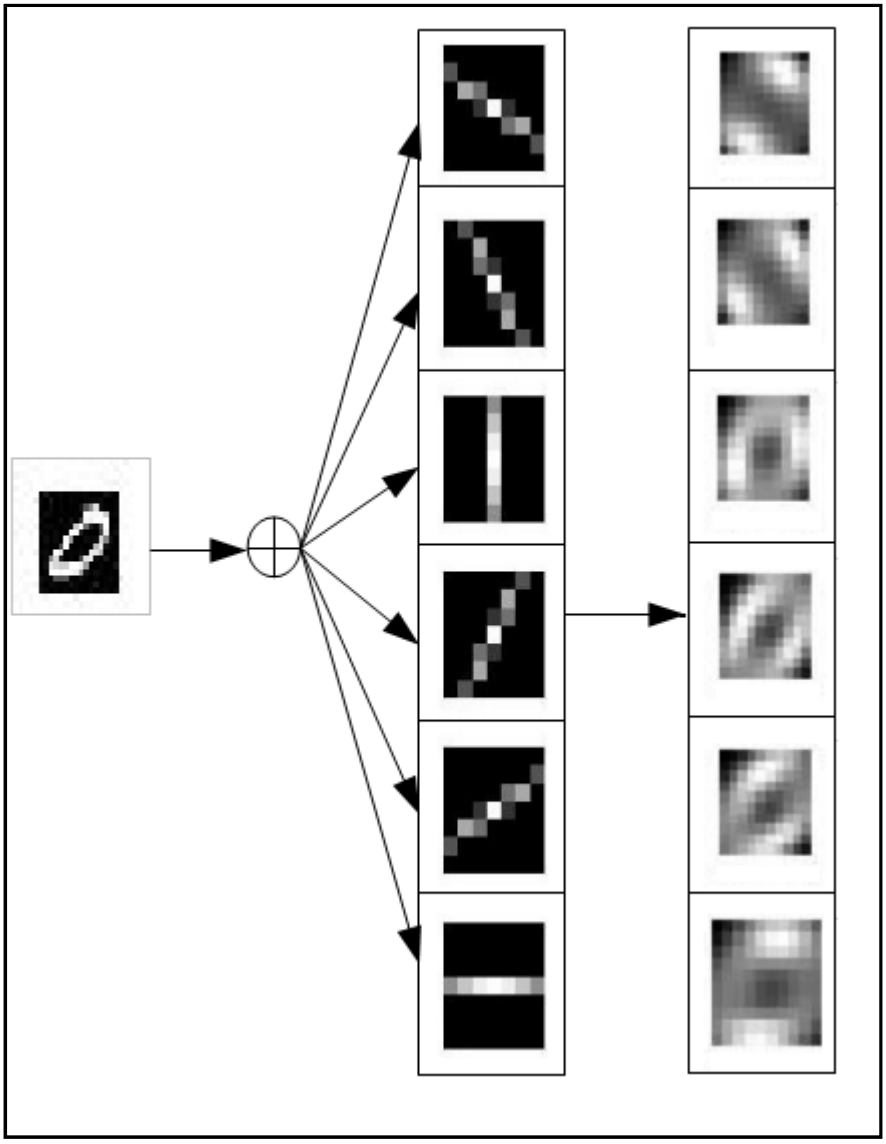

Figura 17. Implementación del sistema bioinspirado.

Como para cada dígito no intervienen todas las orientaciones espaciales (y si intervinieran la cantidad captada para alguna de las orientaciones será ínfima) se pueden descartar diferentes niveles de intensidad sin pérdida de información. En nuestro experimento hemos optado por despreciar aquellos valores a la salida de los Gabor que no superan el 75% del valor más alto para una imagen de entrada. Esta reducción no solo viene motivada por la continua búsqueda de simplificar el sistema para ahorrar en coste computacional, sino también para evitar posibles errores en el aprendizaje de la neurona de salida. Hay que tener cuidado con las contribuciones indeseadas de algunas componentes espaciales que no son características del dígito entrenado, ya que pueden

llevar a respuestas incorrectas en la neurona de salida. Ésto puede desembocar en respuestas erróneas como las falsas alarmas (disparos en instantes erróneos que no se corresponden con la correcta detección del patrón entrenado).

Otra operación realizada consiste en comparar los píxeles de salida de cada filtro de Gabor que coinciden espacialmente. Al usar 6 filtros de Gabor tenemos 6 píxeles por posición (que detectan diferentes orientaciones). Sólo vamos a conservar la información aportada por aquel píxel que contenga un mayor nivel de intensidad. Se desecha la información aportada por el resto de píxeles, ya que podemos considerarla redundante. De este modo se implementa una operación MAX entre las distintas orientaciones y conseguimos un ahorro computacional considerable, sin que afecte al resultado final de todo el proceso constructivo del sistema.

En la segunda etapa de nuestra implementación es donde interviene el sistema de aprendizaje STDP. Este mecanismo biológico de aprendizaje se basa en los tiempos relativos entre los pulsos producidos por una neurona y los que recibe a su entrada y que pueden ser los que le han hecho activarse y disparar. Es por ello que en nuestra implementación necesitamos codificar el valor de cada píxel a la salida de cada convolución en valores temporales. En nuestro sistema hemos realizado una codificación intensidad-tiempo según el orden de llegada de los eventos (*Rank Order Coding* [31]). Cada píxel activo a la salida de los filtros de Gabor (perteneciente todavía a la primera etapa) codifica un pulso en el dominio del tiempo, cuyo valor será inversamente proporcional al valor de intensidad del píxel. Así, un píxel con valor de intensidad cercano a 0 producirá un pulso “tardío” y un píxel con valor de intensidad cercano a 255 producirá un pulso casi instantáneo. Estos pulsos se enviarán a las neuronas de salida a través de las sinapsis.

El tiempo de creación de un pulso vendrá relacionado (inversamente) con el nivel de píxel, según la expresión:

$$t_{pulso} = K \cdot (1 - pixel/255)$$

En nuestra implementación t_{pulso} tiene un valor máximo de 2,5 ms. En la Fig. 18, se representa la curva que modela la codificación en tiempo de los niveles de intensidad de los píxeles activos:

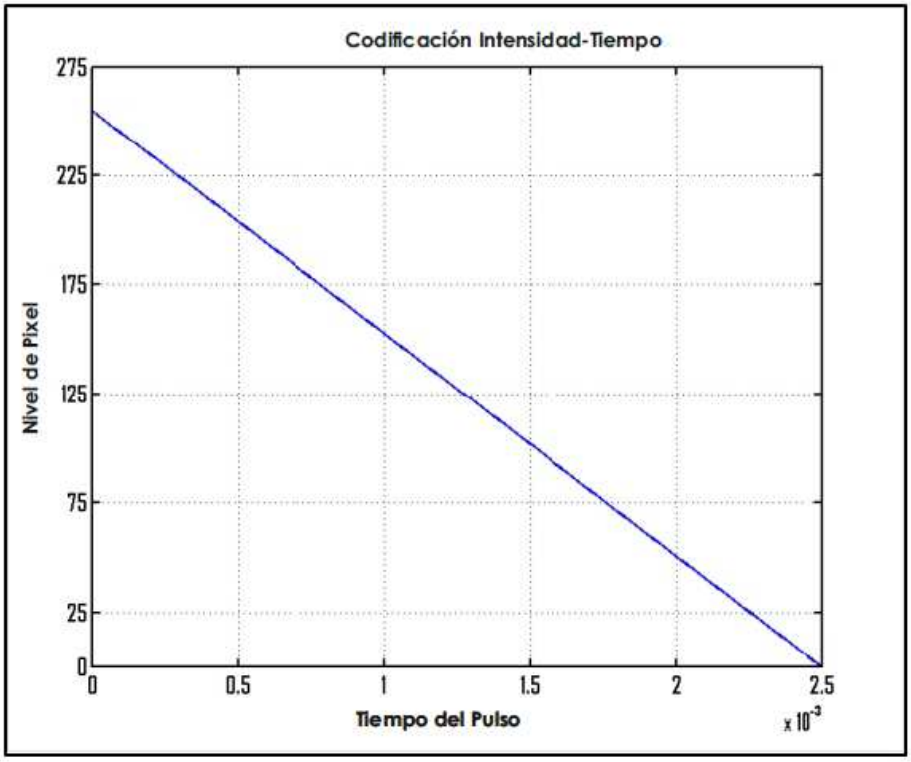

Figura 18. Codificador Intensidad-Tiempo.

En la Fig. 19 se muestra un ejemplo de la codificación del nivel de intensidad de los píxeles en tiempo, como podemos apreciar la relación existente entre ellos es inversamente proporcional. Por tanto, aquellos píxeles que sufren un nivel de intensidad alto toman un valor menor cuando se convierten al dominio del tiempo. Por el contrario, si el valor del nivel del pixel es bajo su transformación al dominio del tiempo adquiere un valor muy alto. Este ejemplo es meramente ilustrativo, ya que realmente los valores no se corresponden con los codificados en este proyecto.

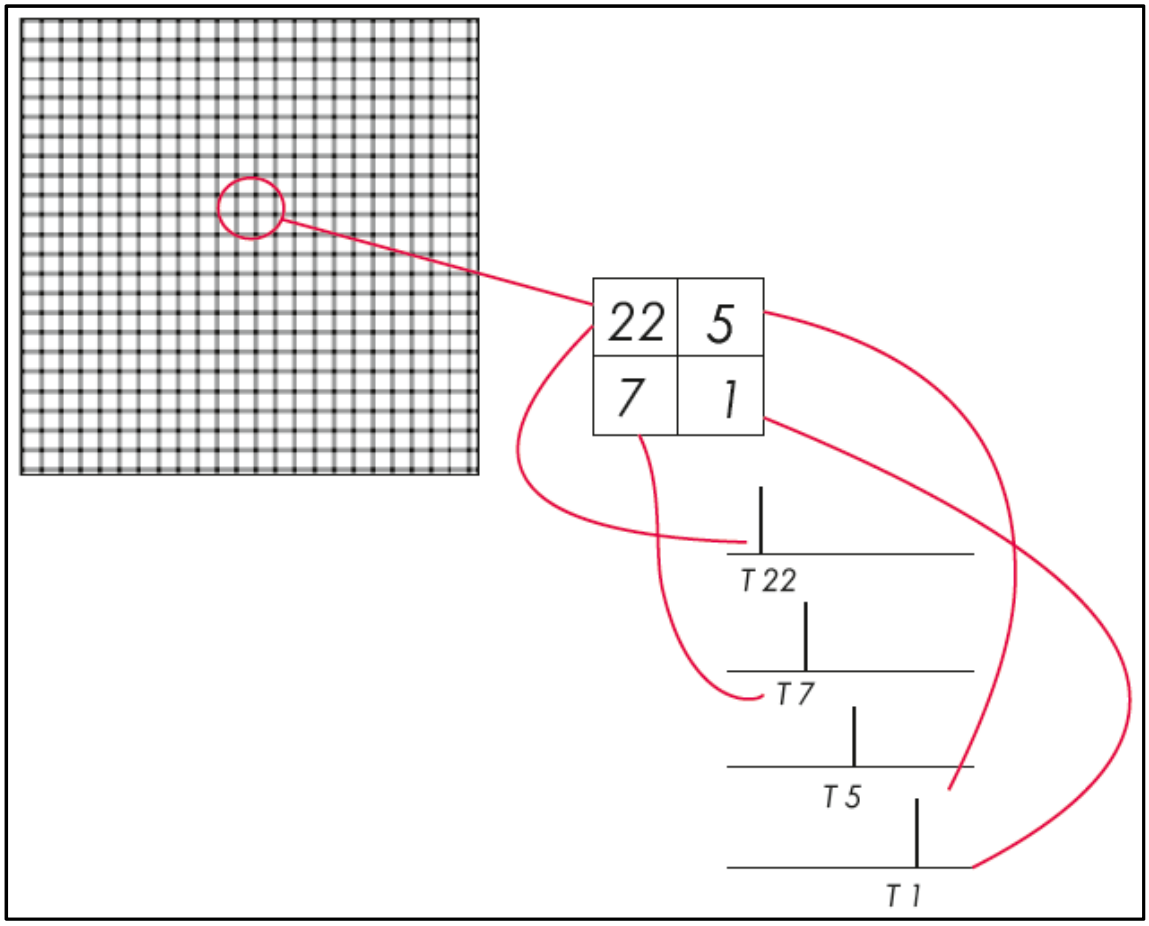

Figura 19. Conversión de nivel de píxel en tiempo.

En nuestro sistema, los píxeles de salida de la primera etapa están conectados con cada una de las 10 neuronas de salida de la segunda etapa. Por lo cual, tenemos 6x100x10 (6 filtros de Gabor, 100 píxeles de salida de cada filtro y 10 neuronas de salida) que hace un total de 6000 conexiones (sinapsis) en nuestro sistema bioinspirado neuronal. Tras la codificación temporal los píxeles envían los pulsos a cada neurona a través de cada sinapsis, empezando así el periodo de aprendizaje y competición entre ellas.

3.2.2. MECANISMO DE APRENDIZAJE STDP

En nuestra implementación cada neurona de salida ha sido entrenada por separado. En una implementación futura y más eficiente se permitirá que todas las neuronas aprendan a la vez. Si además permitiésemos *inhibición* entre ellas conseguiríamos que cada neurona aprendiera un dígito diferente [13]. Por tanto, para la implementación aquí presentada, cada neurona de salida *i*, correspondiente al dígito de entrada *i* (*i*=0,..,9), ha sido entrenada por separado utilizando un conjunto de 5000 imágenes correspondientes al mismo dígito. Una vez entrenadas, las neuronas se hacen selectivas al estimulo de entrada, es decir, la respuesta a un dígito que no sea el entrenado es mucho más lenta que si fuera el patrón aprendido. Esta lentitud en la resolución nos lleva a concluir que nuestra neurona no ha memorizado ese dígito.

Pero nuestra pregunta sigue sin ser respondida, ¿cómo se implementa el aprendizaje basado en mecanismos bioinspirados? Esta pregunta la podemos responder gracias al postulado de Hebb (vínculo de causalidad) [43] [44]:

"Se refuerzan aquellas conexiones en la que las neuronas presinápticas disparan un poco antes que las neuronas postsinápticas."

La primera etapa de nuestro sistema, mostrada en la Fig. 14, implementa las funciones que realizan las neuronas presinápticas. En los sistemas biológicos esta etapa está constituida por las neuronas aferentes (sensorial) cuyo cometido es captar la imagen. Por otro lado, las neuronas postsináptica se corresponden con las neuronas finales de nuestro sistema, equivalente a la segunda etapa de la Fig. 14. Esta etapa tiene la misión de memorizar el digito, actuando las neuronas de salida como auténticos detectores de coincidencia.

Así que gracias a la cita anterior podemos afirmar que el mecanismo de aprendizaje STDP (*Spike-Timing-Depedent-Plasticity*) se basa en el principio en el cual la fuerza de las conexiones con las neuronas de salida ("pesos") se van reforzando o debilitando con la repetición de unos pulsos de entrada determinados [13] [23]. Aquellas conexiones cuyos pulsos transportados provoquen un disparo en la neurona de salida se verán reforzadas. En cambio, aquellas conexiones por las que viajan pulsos

que llegan a la neurona de salida posteriormente al disparo de la misma se verán debilitadas. STDP es un mecanismo *no supervisado*, es decir, las neuronas aprenden de manera autónoma ante la repetición de los patrones de entrada. Gracias a este mecanismo, cuando la neurona haya aprendido un patrón de entrada le bastaran muy pocos pulsos para producir un disparo.

El mecanismo STDP se resume del siguiente modo: Inicialmente el peso de todas las conexiones es el mismo (o puede estar inicializado aleatoriamente), pero cada vez que la neurona de salida produce un pulso los pesos de las conexiones se actualizan de forma positiva o negativa (según en qué instante haya recibido el pulso de entrada la neurona de salida, antes o después de su propio disparo). Los pesos contribuyen de manera acumulativa al potencial de la neurona de salida. Si este potencial alcanza un umbral determinado se produce una reacción en forma de disparo. Una vez que nuestro detector de coincidencia (neurona de salida) responde al estímulo de entrada con un disparo, el acumulador de la neurona entrenada se vacía. Todo este proceso de actualización de los pesos de las sinapsis se rige según las expresiones siguientes [13] [23]:

$$w_o = \Delta w_{inst} + w_{ant}$$

$$\Delta w_{inst} = \begin{cases} a^+ \cdot exp\left(\frac{t_j - t_i}{\tau^+}\right); & t_j \leq t_i \\ -a^- \cdot \exp\left(-\frac{t_j - t_i}{\tau^-}\right); & t_j > t_i \end{cases}$$

En estas ecuaciones w_o es el peso actualizado de la sinapsis una vez que la neurona de salida dispara. Consiste en la suma de w_{ant} (es el peso anterior al disparo del detector de coincidencia asociado a una de las sinapsis) y Δw_{inst} (es el incremento de peso debido al disparo de la neurona final). Los parámetros a^+, a^-, τ^+ y τ^- son constates predefinidas [3], que toman los valores siguientes en nuestra práctica:

$$a^+ = 0.03125;$$

$$a^- = 0.25 \cdot a^+;$$

$$\tau^+ = 16.8e - 3;$$

$$\tau^- = 90e - 3;$$

Por último, el valor t_j - t_i representa la diferencia de tiempo entre el pulso producido por los píxeles activos a la salida de los filtros de Gabor en el instante t_j (cuya conexión con la neurona de salida se está actualizando) y el disparo producido por la neurona final en el instante t_i. La Fig. 20 representa la curva de actualización de pesos. En ella se puede ver que aquellas conexiones cuyos pulsos sean anteriores al disparo de la neurona de salida (y por tanto causantes del mismo) se verán reforzadas. Esto se corresponde con la curva de la parte izquierda, conocida como etapa de potenciación o aprendizaje de la neurona LTP (*Long Term Potentiation*). En cambio, los pesos de aquellas conexiones con pulsos posteriores al disparo de la neurona de salida se verán atenuados. Esto se corresponde con la curva de la derecha, siendo conocida como etapa de depresión u olvido de lo aprendido, LTD (*Long Term Depression*) [23].

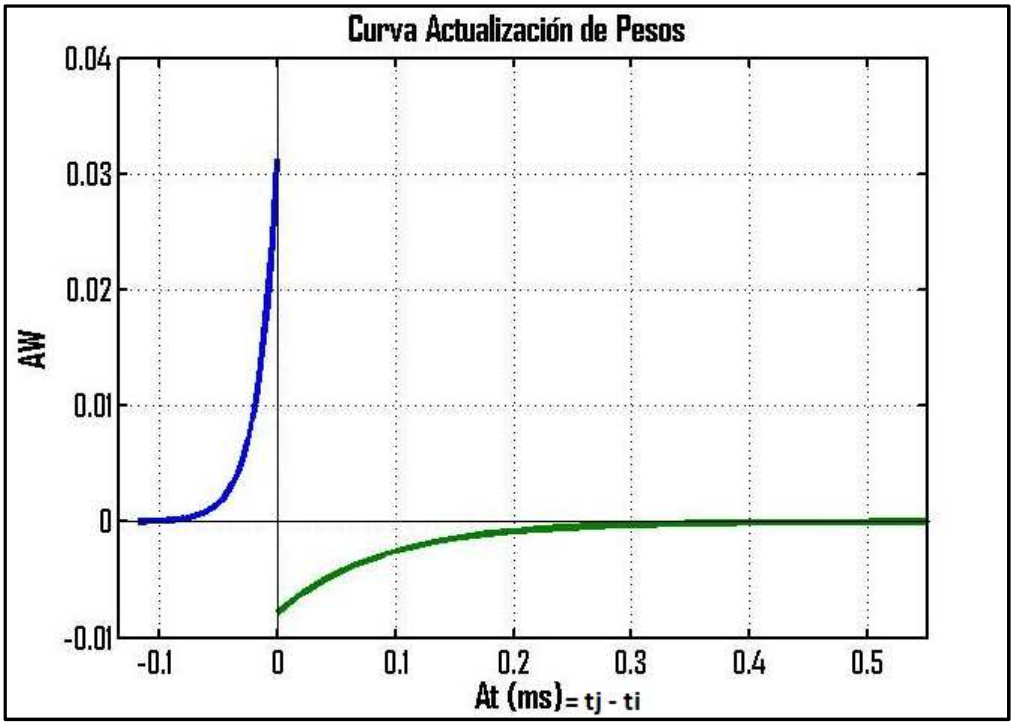

Figura 20. Curva aprendizaje STDP.

Esta actualización de pesos tiene su sentido, ya que si una neurona de salida aprende a reconocer un determinado dígito habrá un grupo de sinapsis que tengan asociadas un mayor peso. Estas sinapsis tendrán una mayor probabilidad de activarse cuando llegue una entrada que sea el dígito entrenado. Podemos utilizar ésto como un factor diferenciador, ya que nos ayuda a que nuestro sistema consiga memorizar un patrón determinado de manera autónoma y rápida (a mayor peso asociado a una conexión mayor probabilidad que la neurona de salida dispare de manera instantánea obviando el resto de enlaces).

En la Fig. 21 podemos observar un ejemplo sobre la actualización de los pesos asociados a las sinapsis, para ello utilizamos la curva de aprendizaje STDP. Los puntos rojos significan aquellos píxeles activos en la etapa 1 (salida de los Gabor). En nuestro modelo, el punto rojo que aparece circulado se corresponde con el píxel activo que envía un pulso (codificado en el dominio del tiempo) a través de una sinapsis y que causa el disparo de la neurona de salida. Podemos decir que la sinapsis que transporta este pulso es casi la principal culpable de que nuestro detector de coincidencia dispare, por eso el incremento de peso será máximo. Los eventos que llegan antes que nuestro sistema responda, provocan una actualización positiva pero en menor medida. Esto se corresponde con la etapa de potenciación de las conexiones involucradas. En cambio los estímulos posteriores a la respuesta final del sistema se ven claramente envueltos en una etapa de depresión, disminuyendo el valor de los pesos asociados a las sinapsis que transportan dichos pulsos.

Si nos detenemos un poco más en la curva de actualización de los pesos, podemos apreciar que hemos premiado con mayor intensidad y pendiente la etapa de potenciación o aprendizaje (parte de la izquierda de la curva) con respecto al periodo de depresión dónde se empieza a olvidar la regla memorizada (parte de la derecha de la curva). Esto es así, ya que tras un cierto periodo de aprendizaje la respuesta del sistema es automática e instantánea. Cuando la neurona de salida haya sido entrenada durante un tiempo, la neurona disparará tras la llegada de unos pocos pulsos tempranos, debido al alto valor de los pesos de aquellas sinapsis que los transportan. Las sinapsis que apenas transporten pulsos o que vengan con mucho retardo irán disminuyendo el valor de sus pesos poco a poco. Estas sinapsis son mayoritarias con respecto a las que intervienen activamente en el disparo de la neurona de salida. Es posible que alguna vez, sinapsis

que suelen mandar pulsos tempranos los envíen posteriormente a aquéllas que producen la activación a la salida, actualizándose sus pesos negativamente. Así que para que no haya muchos errores hemos decidido que la intensidad y pendiente de la curva de depresión sea menor que la de potenciación. Por tanto, la actualización de los pesos en la parte de la curva de depresión es mucho más suave.

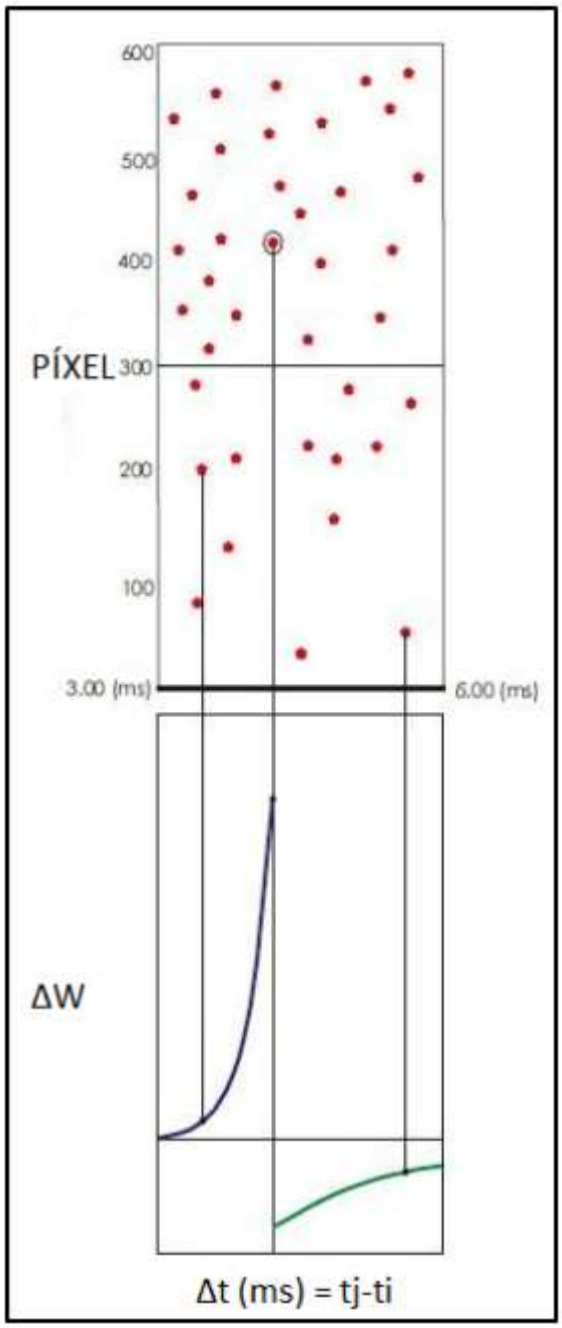

Figura 21. Ejemplo aplicado curva aprendizaje STDP.

3.3. RESULTADOS EXPERIMENTALES

Podemos diferenciar dos fases en nuestro experimento. La primera fase se corresponde con el entrenamiento de cada neurona de salida por separado. La segunda fase se produce cuando la neurona de salida ha interiorizado el entrenamiento. Podemos decir que cada neurona ha memorizado el dígito con el que ha sido entrenada.

En la Fig. 22, hemos representado la fase de entrenamiento correspondiente a la neurona “1”. El eje *y* de la gráfica superior muestran los píxeles activos que disparan un pulso codificado en el dominio del tiempo (se corresponden con la salida de la imagen convolucionada con el banco de filtros de Gabor). Como recordamos, había 600 píxeles posibles unidos a cada neurona de salida. El pulso producido por cada uno de estos píxeles (sólo se han representado aquellos con un valor superior a un nivel predefinido) se ha representado como un punto rojo. A su vez, aquellos que aparecen circulados son los desencadenantes de que el acumulador de la neurona de salida se desborde y provoque la respuesta de nuestro sistema. El eje *x* codifica el tiempo en segundos. Cada imagen captada produce una serie de pulsos en una ventana de duración 3 ms (recordemos que el tiempo utilizado en la codificación intensidad-tiempo es de 2,5 ms pero dejamos un margen de 0,5 ms para que no haya solapamiento entre las imágenes detectadas). La imagen inferior, representa el potencial acumulado por la neurona de salida que se está entrenando. Se puede observar en esta etapa que la neurona necesita cargar durante un tiempo largo (del orden de 22ms, ~8 imágenes) la información correspondiente a los patrones para producir un pulso de salida y descargar su acumulador de potencial. En esta etapa inicial no tenemos la certeza de que cuando se produce la respuesta del sistema sea realmente porque se ha detectado un patrón, ya que es posible que haya ocurrido una falsa alarma debida a la acumulación de pesos (recordamos que todas las conexiones al inicio tienen el mismo peso asociado). Esto se va corrigiendo con la repetición sucesiva de entradas. En el momento del disparo entra en juego la actualización de los pesos de cada conexión, contribuyendo así a que en un futuro la neurona de salida detecte rápidamente el patrón de entrada.

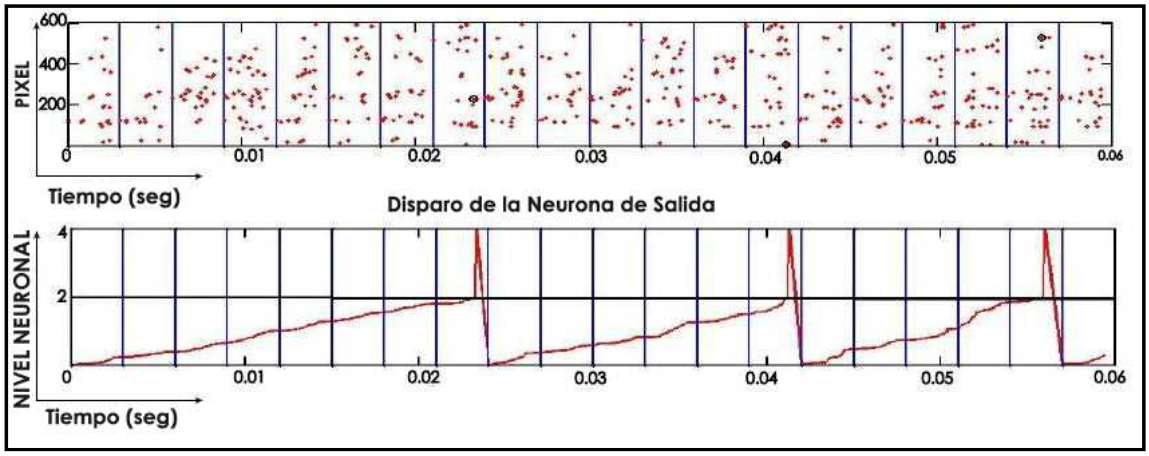

Figura 22. Etapa de entrenamiento de la neurona '1'.

Poco a poco, conforme van llegando las imágenes que representan el dígito "1" el sistema va evolucionando, cada vez habrá menos slots donde no se produzca disparo. Debido al aprendizaje nuestro detector se hace poco a poco más selectivo y le cuesta menos reconocer el patrón entrenado evitando así la reproducción de falsas alarmas.

En la Fig. 23 se muestra la etapa final del entrenamiento. Se puede observar que la neurona sólo necesita unos poquitos pulsos para activarse y producir un disparo. Ésto es prueba de que ya se ha entrenado y ha reconocido el patrón a la entrada (del orden de décimas de ms). La neurona ya ha memorizado él dígito. Además la fiabilidad aumenta, como veremos en puntos posteriores de esta memoria. Podemos apreciar que cada vez que llega una imagen que contiene el patrón entrenado la salida dispara (en cada slot de la Fig. 23 se produce una respuesta del sistema). En esta etapa disminuye considerablemente la ocurrencia de falsas alarmas, ya que al detectar tan rápidamente el patrón memorizado no hay posibilidad de que algún elemento extraño al patrón produzca confusión.

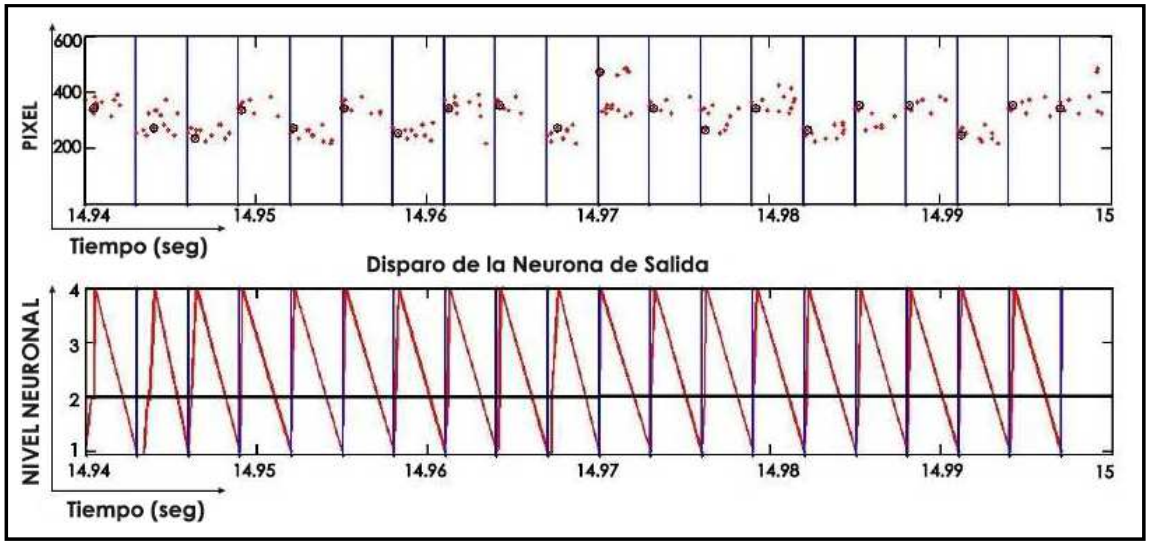

Figura 23. Etapa de entrenamiento de la neurona '1' al final.

Una vez que el sistema asimila el aprendizaje podemos observar como el grupo de píxeles responsables de la activación de la respuesta del sistema suelen ser vecinos, ya que se corresponden con la región donde la componente espacial de Gabor es más fuerte para ese dígito. Por ejemplo en el dígito "1" las componentes más fuertes son las componentes verticales. Por tanto, nuestro mecanismo de aprendizaje se vuelve muy selectivo, sólo responde ante aquellos píxeles que tienen una participación activa en el patrón a memorizar. En este punto sería necesario aplicar el concepto de inhibición si todas las neuronas del sistema compitieran durante la etapa de aprendizaje (esto no se desarrolla en el presente proyecto), ya que como hemos dicho la componente principal del digito "1" es la vertical al igual que la del digito "7". Si la entrada fuera un "7" está claro que la neurona entrenada con el dígito "1" podría ganar frente a la que estuviera aprendiendo a reconocer el dígito "7". Para que no ocurra esto, la neurona entrenada con el dígito "7" debe mandar una señal de inhibición a la neurona "1" cuando detecte la componente horizontal del número "7". Así solucionaríamos el problema de solapamiento de aprendizaje entre neuronas.

En Fig. 24 hemos representado todos los píxeles activos culpables de los disparos de la neurona de salida al introducir 5000 entradas que representan el dígito "1". Esta imagen las hemos relacionado a su vez con los 6 filtros de Gabor utilizados

(parte derecha de la imagen), consiguiendo así localizar las componentes predominantes en el entrenamiento del dígito “1”. Observamos claramente que las componentes vertical e inclinada son las más repetidas, correspondiéndose con los pixeles comprendidos entre [300, 400] y [200, 300]. Esto concuerda con las componentes espaciales y de orientación del dígito “1”.

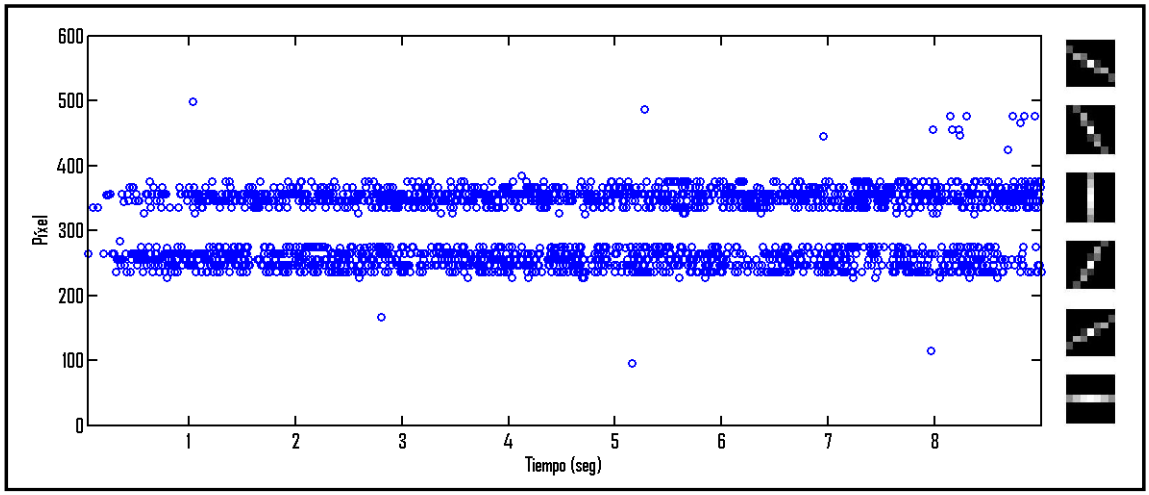

Figura 24. Píxeles activos que provocan el disparo de la neurona de salida cuando se entrena el sistema con el dígito “1”.

En cambio para el caso de aprendizaje del dígito “0” intervienen casi todas las componentes (horizontales, inclinadas y verticales). En la Fig. 25 podemos ver como contribuyen todas las componentes en la respuesta del sistema. Que colaboren todas las orientaciones de los filtros de Gabor no tiene porque ser precisamente malo, ya que podemos utilizar esta característica como un factor diferenciador. Aún así se puede observar que las componentes inclinadas son las más potentes, ya que son las mayores responsables de los disparos de la neurona final.

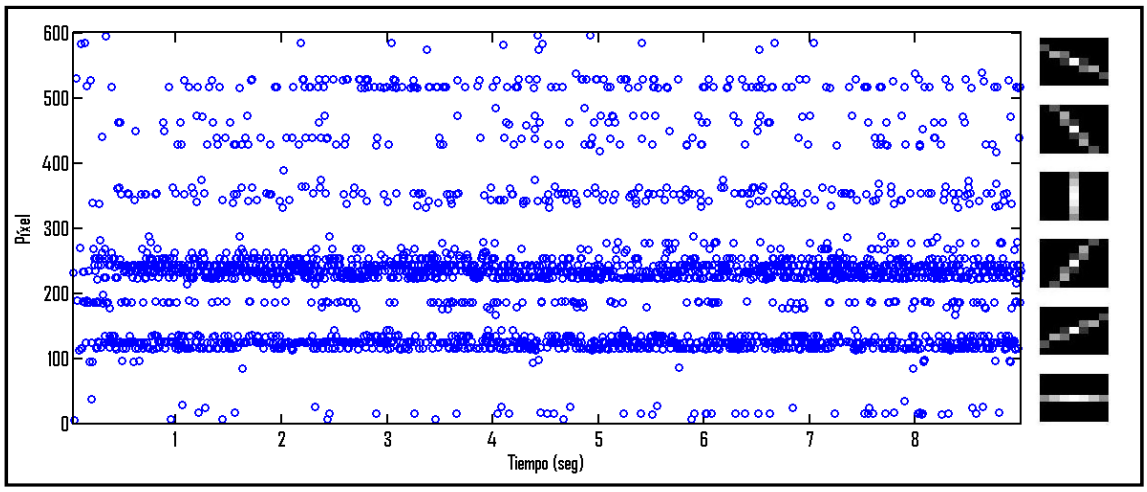

Figura 25. Píxeles activos que provocan el disparo de la neurona de salida cuando se entrena el sistema con el dígito “0”.

Como hemos comentado antes, una de las principales características del mecanismo implementado es la rápida respuesta dada por la neurona de salida cuando ha memorizado una determinada imagen. En la Fig. 22, se aprecia como la neurona de salida tarda una media de 22 ms en disparar ante el continuo bombardeo de imágenes que representan el número natural “1”, siendo un valor muy alto. Debido a la inexperiencia del sistema los pesos asociados a nuestras sinapsis son pequeños e iguales, por lo que los pulsos que recorren las conexiones les cuesta mucho desbordar el umbral del acumulador de la neurona salida y cuando lo consiguen puede que la respuesta no sea correcta (existe la posibilidad que sea una falsa alarma al ser todos los pesos parecidos). En cambio pasa todo lo contrario en la Fig. 23, dónde nuestro modelo ya está muy experimentado y ya existe una gran diferencia entre los pesos de las sinapsis. Habrá sinapsis con pesos elevados que serán responsables del disparo de la neurona de salida. Se reduce drásticamente el tiempo de respuesta a unos pocos milisegundos, esto es, la memorización se ha alcanzado.

La Fig. 26 representa la diferencia de tiempos existentes entre disparos sucesivos de la neurona de salida conforme se va entrenando con el dígito “1”. En el eje *y* se representa la diferencia de tiempo que tiene lugar entre dos disparos sucesivos y en el eje *x* se representa el numero de disparos (es por ello que toma valores discretos). Por ejemplo el valor (*x*=5, y=0’004) significa que la diferencia de tiempo que existe entre los disparos que ocurren en el quinto y cuarto lugar es de 4 ms. Se observa

perfectamente que el tiempo inicial es alto, conforme la neurona de salida va asimilando el aprendizaje la diferencia de tiempo entre disparos va disminuyendo.

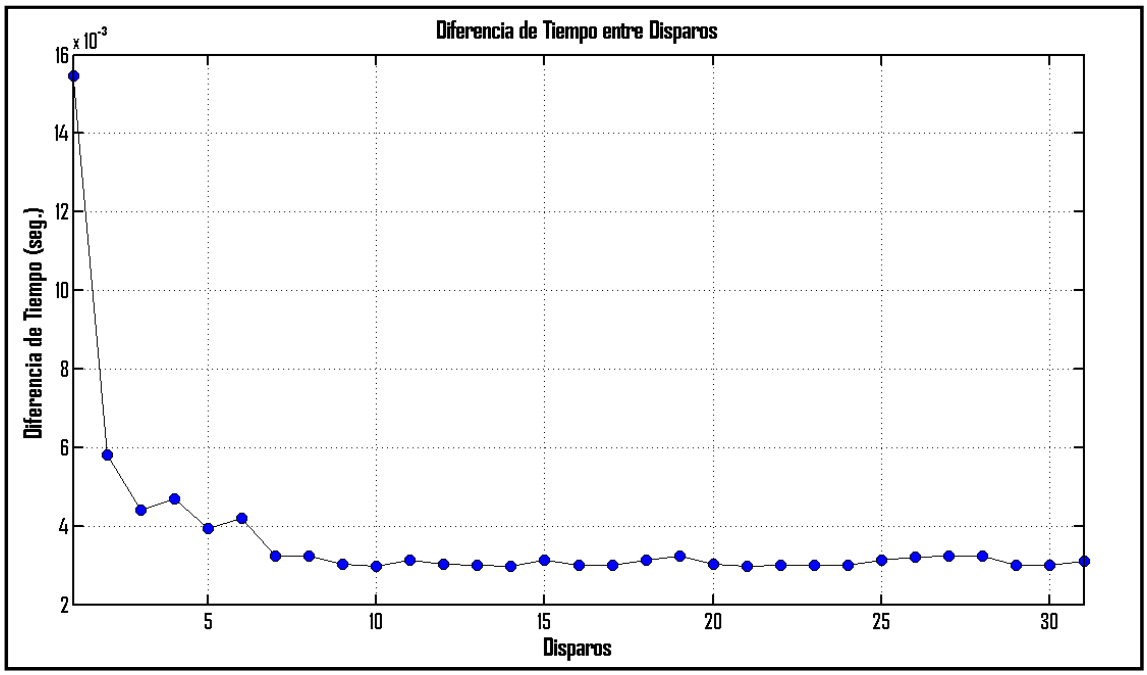

Figura 26. Diferencia de tiempo entre disparos sucesivos de la neurona de salida ($disp_i\ (t) - disp_{i-1}\ (t)$).

En la Fig. 27 observamos la representación de los pesos finales asociados a las sinapsis que han contribuido a que la neurona de salida aprenda a reconocer el dígito "1". Vemos que las únicas componentes importantes de la Fig. 27 son las centrales. Esto quiere decir que los pesos de las componentes verticales e inclinadas son los únicos que se han reforzados positivamente. Queda claro que las orientaciones verticales e inclinadas son las más potentes del dígito entrenado.

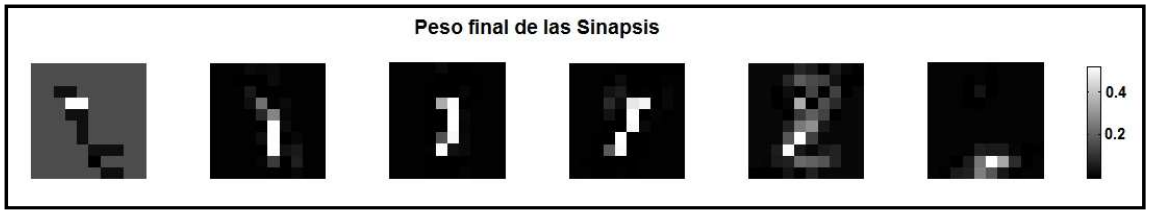

Figura 27. Peso final de las sinapsis cuando el sistema se entrena con 5000 entradas que representan el digito "1".

3.3.1. ESTUDIO DE LA RESPUESTA DE LA NEURONA ANTE DIFERENTES ENTRADAS

Una vez desarrollado nuestro sistema, necesitamos acotar el grado de fiabilidad del sistema. Para ello, vamos a realizar un pequeño estudio donde compararemos el grado de selectividad obtenido por la neurona de salida entrenada con un determinado patrón. Básicamente el experimento consiste en estudiar la respuesta del sistema ante un número memorizado y otro que no lo es.

Para la realización de esta práctica nos hemos surtido de los pesos finales de una neurona entrenada con el patrón que representa el número natural "7" (se ha bombardeado el sistema con 5000 entradas que representan el mismo dígito, asegurándonos así que los pesos asociados a las sinapsis son los resultantes cuando la neurona final ha memorizado el patrón). La respuesta del sistema es instantánea cuando se introduce el patrón entrenado, como se aprecia en la Fig. 28. Vemos que no es necesario que el mecanismo procese toda la señal de entrada, ya que le basta con el procesamiento de tan solo unos pocos pulsos enviados por los primeros píxeles activos para dar una respuesta.

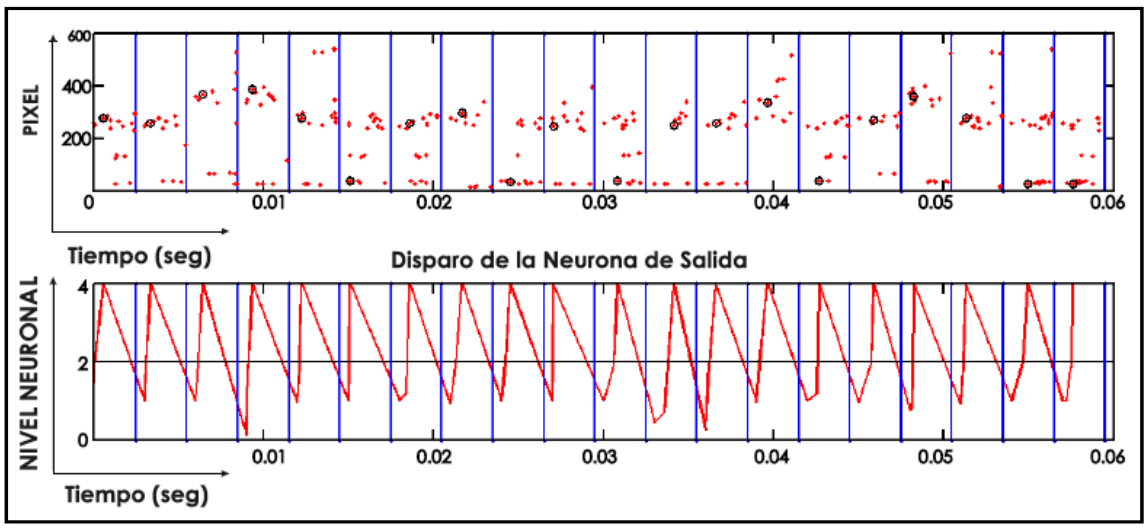

Figura 28. Respuesta ante el dígito aprendido "7".

En la Fig. 28, vemos la representación gráfica de lo expuesto previamente. Hay muy pocos píxeles que detecten señal debido a que la actividad de los mismos no supera la sensibilidad mínima a la entrada, esto es debido a que el digito “7” está compuesto principalmente por dos orientaciones (inclinada y horizontal). Otro elemento reseñable en la imagen es la alta fiabilidad de la estructura, en cada slot de tiempo se produce disparo del sistema. Podemos concluir que siempre que se introduzca el patrón entrenado se produce respuesta indicando que ha habido un reconocimiento del dígito.

En la Fig. 29 introducimos al sistema entradas que representan el número natural “6”. Ahora vemos que la respuesta cambia ya que la neurona de salida no reconoce el patrón memorizado en las entradas. Ahora hay muchas componentes activadas que antes no lo estaban, ya que para este nuevo dígito predominan componentes verticales, inclinadas y horizontales. Ahora la respuesta es mucho más lenta e incluso puede que no haya disparo en alguna ventana. La neurona dispara por la continua excitación a la entrada (por la acumulación de la contribución de los pesos de las conexiones), por lo que entendemos que las respuestas del sistema pueden ser falsas alarmas en su mayoría.

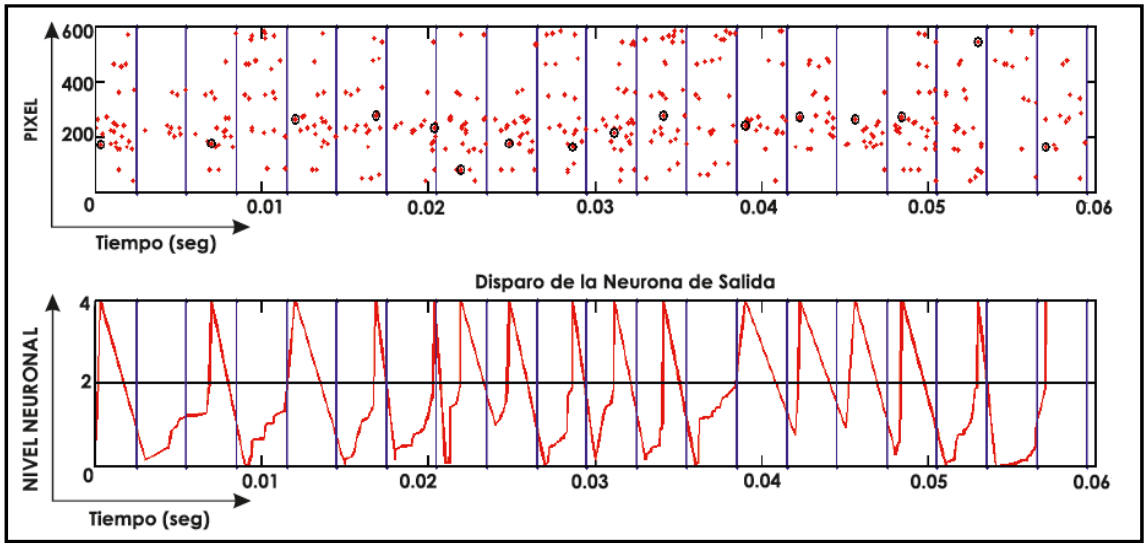

Figura 29. Respuesta ante el dígito no aprendido “6”.

Si la neurona de salida entrenada con el dígito “7” empieza a entrenarse con otra señal (como en este caso el dígito “6”), olvidará lo aprendido hasta hora y comenzará de

nuevo una etapa de aprendizaje. Así queda demostrado que las neuronas y sus sinapsis pueden sufrir un proceso de reciclaje, donde conexiones que antes no intervenían ahora empiezan un proceso de actividad. Por tanto, podemos hablar de las sinapsis como uniones “semipermanentes”. Las conexiones estarán establecidas durante el periodo de actividad en el que llega el patrón entrenado. En cambio, durante el periodo de inactividad o de llegadas de entradas aleatorias (diferentes a la entrada memorizada) estas conexiones se irán diluyendo poco a poco conforme vaya pasando el tiempo.

3.3.2. EVALUACIÓN DEL SISTEMA

En el apartado anterior comprobamos de manera gráfica como la respuesta de la neurona es más rápida cuando se estimula con un patrón de entrada entrenado frente a otro diferente. Para demostrar empíricamente estos datos se desarrolla a lo largo de este apartado el estudio del *rate* o tasa de éxito. Consideramos éxito el que haya disparado la neurona deseada (aquella entrenada con el dígito que recibe a la entrada) antes que las demás. El cálculo viene expresado según la fórmula:

$$rate = \frac{succ}{total} \cdot 100$$

En esta ecuación, *succ* se corresponde con el número de disparos que se producen correctamente, es decir aquellas entradas aplicadas a neuronas de salida que no han sido entrenadas con dicha entrada provocan un tiempo de disparo mayor que en el del detector que si ha sido entrenado con dicho entrada (sería el éxito de nuestro sistema). Por otro lado la variable *total* se corresponde con el número de disparos totales que se producen en el testing.

Para conseguir un valor fiable del *rate* del sistema se ha entrenado cada una de las diez neuronas de salida con 5000 imágenes que codifican uno de los dígitos (estos dígitos han sido obtenidos de la base de datos MNIST [14]). Una vez que cada neurona ha memorizado su dígito correspondiente, se toma cada una de manera individual y se la expone a excitaciones que se corresponden con cada uno de los dígitos involucrados en este experimento. El siguiente paso es tomar los tiempos de disparo de la neurona de salida y aplicar la fórmula dada por el *rate*.

Sólo a modo de ejemplo, en la Tabla 1 se muestra el tiempo que tardan en disparar las diferentes neuronas, entrenadas cada una previamente con los diferentes dígitos intervinientes, cuando a la entrada se introducen diez imágenes que representa el digito “0”. La tabla viene sombreada en verde cuando la neurona entrenada con el dígito “0” reconoce su patrón antes que el resto de neuronas entrenadas anteriormente con otros estímulos. En cambio va sombreado en rojo cuando uno de los detectores entrenado con un digito que no es el “0” dispara antes que la neurona que si lo está.

Neurona	1ºdisp	2ºdisp	3ºdisp	4ºdisp	5ºdisp	6ºdisp	7ºdisp	8ºdisp	9ºdisp	10ºdisp
0	0,78	0,85	1,17	0,65	1,09	0,79	0,92	2,79	0,71	0,77
1	2,22	9,50	1,75	4,89	3,5	5,03	3,50	6,74	4,84	10,86
2	1,28	2,51	1,88	0,99	2,70	0,92	1,68	1,09	1,53	2,38
3	1,20	2,52	3,80	2,71	1,99	0,85	1,61	6,84	2,10	0,83
4	4,65	2,10	5,07	1,59	1,62	1,98	1,83	2,65	1,66	4,59
5	1,17	2,79	2,26	1,97	3,50	2,81	1,31	1,91	1,64	2,31
6	0,95	2,10	1,67	0,92	1,21	1,05	1,54	1,48	1,16	1,63
7	1,36	4,27	2,90	1,95	1,42	1,63	1,91	2,78	4,12	3,57
8	1,04	3,60	1,24	2,72	4,73	2,10	1,98	2,90	3,50	2,89
9	2,11	2,21	6,50	1,74	5,47	4,40	2,82	1,56	1,58	2,81

Tabla 1. Disparos de las neuronas de salida ante el mismo estímulo (*ms*).

Este caso es bastante ilustrativo y da un buen resultado, ya que en los 10 primeros disparos de cada detector de coincidencia en solo una ocasión se produce una detección errónea. Para este caso último caso, la salida reacciona más rápido ante un patrón no memorizado previamente. Haciendo los cálculos:

$$rate_{N_0} = \frac{succ}{total} \cdot 100 = \frac{9}{10} \cdot 100 = 90\%$$

Haciendo un estudio un poco más exhaustivo para 500 imágenes de entrada a cada neurona obtenemos para cada salida entrenada el caculo de *rate* mostrado en la Tabla 2:

Neurona	Rate
0	93.33
1	94.11
2	75.89
3	78.11
4	79.22
5	80.78
6	83.33
7	76.89
8	80.89
9	76.11

Tabla 2. Tasa de éxito para cada neurona después de 500 imágenes de entrada (%).

Podemos observar en la tabla anterior una aceptable tasa de éxito, ya que el nivel más bajo obtenido es de 75,89 % para la neurona encargada de detectar el dígito “2”. Es posible que este dígito falle más que otros porque comparte componentes parecidas a los dígitos “3” y “7”. Podemos observar en la Tabla 2 que las neuronas encargadas de memorizar estos dígitos también tienen un porcentaje de éxito menor al resto (78.11% y 76.89 respectivamente), debido a la similitud de las direcciones y orientaciones de los patrones entrenados. Por eso, a la hora de realizar un estudio del éxito del sistema las neuronas pueden equivocarse en la detección al confundir entradas parecidas, bajando así la tasa de éxito.

Por último hemos hecho un promediado de todos los *rate*, obteniendo un valor que consideramos aceptable:

$$rate_{total} = \frac{\sum_{i=1}^{n=10} Rate_i}{n} = 81.87\ \%$$

Se hace evidente que una gran mejora del sistema será el entrenamiento conjunto de las diez neuronas de salida, donde la inclusión de inhibición entre las diferentes neuronas hará que las características no presentes en un dígito y que aparezcan a la entrada inhiban, haciendo al sistema por tanto más selectivo.