RED NEURONAL CONVOLUCIONAL RÁPIDA SIN FOTOGRAMAS PARA RECONOCIMIENTO DE DÍGITOS

J. A. Pérez-Carrasco [(1)], C. Serrano [(1)], B. Acha [(1)], T. Serrano-Gotarredona [(2)], and B. Linares-Barranco [(2).]

jcarrasco@imse-cnm.csic.es, cserrano@us.es, bacha@us.es, terese@imse-cnm.csic.es, bernabe@imse-cnm.csic.es

[(1)] Dpto. Teoría de la Señal, ETSIT, Universidad de Sevilla. Avda de los Descubrimientos, s/n, CP41092

[(2)] Instituto de Microelectrónica de Sevilla (IMSE-CNM-CSIC) Avda. Reina Mercedes, s/n, Sevilla. CP41012.

***Abstract*- In this paper a bio-inspired six-layer convolutional network (ConvNet) non-frame based for digit recognition is shown. The system has been trained with the backpropagation algorithm using 32x32 images from the MNIST database. The system can be implemented with already physically available spike-based electronic devices. 10000 images have been coded into events separated 50ns to test the non-frame based ConvNet system. The simulation results have been obtained using actual performance figures for existing AER (Address Event Representation) hardware components. We provide simulation results of the system showing recognition delays of a few microseconds from stimulus onset with a recognition rate of 93%. The complete system consists of 30 convolution modules.**

I. INTRODUCCIÓN

Aún a día de hoy, incluso los ordenadores más rápidos no son capaces de implementar tareas que podrían ser consideradas para una persona realmente simples, tales como el reconocimiento de objetos, toma de decisiones, etc. Estas operaciones, sencillas para los cerebros humanos, requieren múltiples operaciones secuenciales en los ordenadores. El poder de procesamiento en el cerebro humano se debe principalmente al elevado número de neuronas y a las masivas interconexiones presentes entre ellas (sinapsis). Esta interconexión masiva permite un procesamiento en paralelo de toda la información de entrada, proporcionando resultados en tiempo real. En las últimas décadas, los sistemas más exitosos que tratan de emular el comportamiento del cerebro han sido aquellos basados en redes convolucionales (*ConvNets*) [1][2] entrenadas usando algoritmos de aprendizaje basados en gradiente [1]. La gran desventaja de las redes convolucionales desarrolladas hasta ahora es que son sistemas basados en fotogramas y que carecen del procesamiento basado en eventos (y por tanto en tiempo real) implementado en el cerebro. Otro de los problemas de estas redes basadas en fotogramas es que, a pesar de utilizar la técnica de compartir pesos de conexiones (*weight-sharing*), el número de interconexiones presentes es muy grande. Esto conlleva la casi imposibilidad de implementar una versión hardware de un sistema neuronal convolucional de cierta complejidad basado en fotogramas.

Una posible solución para emular el procesamiento basado en eventos y el enorme número de interconexiones presente en los sistemas bioinspirados es el protocolo AER (representación de eventos mediante direcciones, *address-event representation*) [3]. Mediante AER, el estado continuo de neuronas de una capa se codifica en secuencias rápidas de pulsos digitales (también llamados eventos) de ancho temporal mínimo (en el orden de *ns*) y que están multiplexados con eventos de otras neuronas en un bus digital de alta velocidad común entre los chips. Cada vez que una neurona genera un evento, su dirección *(x, y)* se escribe en el bus.

AER ha sido utilizado por una amplia comunidad de investigadores en sistemas neuromórficos en múltiples aplicaciones [4]. Sin embargo, uno de los hitos más importantes en el desarrollo de sistemas neuronales convolucionales bioinspirados fue la implementación física de chips de convolución totalmente programables basados en AER y que funcionan en tiempo real [5][6]. El desarrollo de tales chips permitió el diseño de ConvNets complejas de gran escala para la implementación de aplicaciones de reconocimiento de patrones y objetos. En los módulos de convolución basados en AER, cada vez que un evento llega a un módulo de convolución, un campo de proyección (el núcleo de convolución) es añadido a una matriz de píxeles alrededor de la dirección indicada por el evento de entrada [6]. Este procesamiento basado en campos de proyección se produce también en las primeras etapas de la corteza visual, tal como informaron Hubbel y Wiesel en 1959.

Para el presente artículo, se ha desarrollado una red neuronal convolucional (ConvNet) bioinspirada compuesta por seis etapas basada en eventos y no en fotogramas. La red tiene como objetivo el reconocimiento de dígitos numéricos escritos a mano en tiempo real. El sistema implementado se basa en la ConvNet Lenet-5 implementada por Y. LeCun [1]. Sin embargo, para la posible implementación hardware que mantenga una alta tasa de reconocimiento se llevaron a cabo varias simplificaciones y modificaciones a la estructura de Y. LeCun.

II. RED NEURONAL CONVOLUCIONAL BASADA EN FOTOGRAMAS

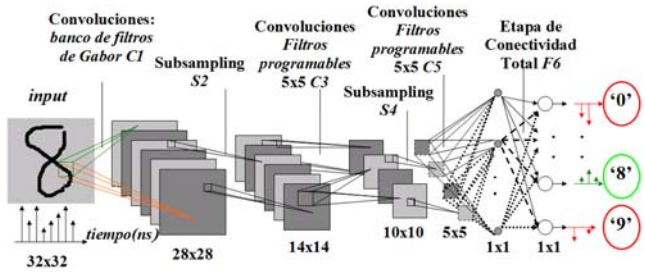

Fig. 1. ConvNet basada en fotogramas para el reconocimiento de dígitos

La versión basada en fotogramas de la ConvNet implementada para el reconocimiento de dígitos se muestra en la Fig. 1. En el sistema, la salida de cada una de las seis capas consiste en un conjunto de imágenes de salida o planos llamados "mapas de características" (*feature maps*), que se componen de conjuntos de neuronas. Las neuronas que pertenecen a un mapa están conectadas únicamente con neuronas en mapas de la capa siguiente a través de campos de proyección (máscaras de convolución). Una unidad (píxel o neurona), ubicada en la posición *(i, j)* dentro de un mapa *q* (de tamaño *KxL*, *q = 1,…, Q*) perteneciente a la capa *l* tendrá un valor $y_l^q(i,j)$ calculado como:

$$x_l^q(i,j) = \sum_{p\in P}\sum_{m\in M}\sum_{n\in N}(y_{l-1}^p(m,n) * W_l^{p,q}(i-m, j-n)) + b_l^q$$
$$y_l^q(i,j) = A\tanh(S * x_l^q(i,j)) \qquad (1)$$

donde *P* es el número total de mapas de características en la etapa precedente *l-1*, *Q* es el número total de mapas en la etapa actual *l*, $x_q^l(m,n)$ es el estado del píxel *(i, j)* en el mapa *q*, $W_l^{p,q}$ es la máscara de convolución que conecta el mapa de características y_{l-1}^p con el mapa de características y_l^q. b_l^q es un offset y *A* y *S* son contantes [1].

La primera etapa *C1* en la estructura Lenet-5 es una etapa convolucional con 6 mapas de características. Cada unidad en uno de los mapas está conectada a una vecindad de 5x5 unidades en la imagen de entrada. Los pesos de esas conexiones son entrenables y compartidos por todas las unidades. El tamaño de los mapas es de 28x28. En nuestro sistema hemos utilizado un banco de filtros de Gabor de tamaño 10x10 con dos escalas y tres orientaciones, evitando el entrenamiento de los 156 parámetros entrenables en el Lenet-5 original. Un banco de filtros de Gabor es a menudo la primera etapa de procesamiento en muchos sistemas y también en el cerebro humano [1]. La segunda capa (S2) es una capa de submuestreo (*subsampling*) con 6 mapas de características de tamaño 14x14. Cada unidad de cada mapa está conectada a una vecindad de 2x2 unidades en el mapa correspondiente de la etapa anterior C1. En nuestro sistema, hemos implementado las capas de submuestreo como capas de promediado, donde cada píxel calcula la suma de sus cuatro pixeles en el mapa correspondiente de la capa anterior. La capa S2 tiene 4704 conexiones y no tiene ni coeficientes entrenables ni funciones sigmoides no lineales.

La etapa C3 es de nuevo una capa convolucional, con cuatro mapas de características de tamaño 10x10. Cada píxel de la tercera capa tiene conexiones de entrada de cada uno de los seis mapas de la capa anterior. Por tanto, hay 24 filtros

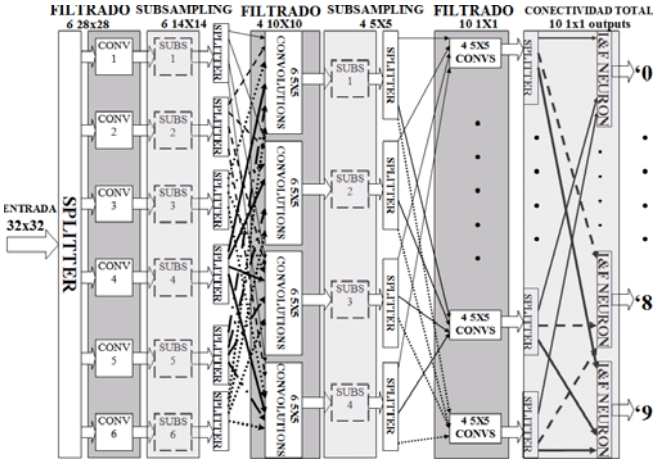

Fig. 2. ConvNet basada en AER para el reconocimiento de dígitos.

5x5 entrenables diferentes. Esta capa tiene 600 coeficientes entrenables y el número de conexiones es 60000.

La capa S4 es de nuevo un bloque de submuestreo con cuatro mapas de características 5x5. El número de parámetros entrenables es cero otra vez y el número de conexiones es de 400. La etapa C5 es de nuevo una capa convolucional con diez mapas de tamaño 1x1. Cada unidad (píxel) está conectada a todos los píxeles de todos los mapas de características en S4 a través de campos de proyección. En la etapa C5 hay 40 filtros entrenables de tamaño 5x5. Por lo tanto, la capa C5 tiene 1000 conexiones entrenables.

Finalmente, la última capa de F6 contiene 10 unidades de salida (10 dígitos diferentes) y está totalmente conectada a C5. Esta etapa tiene 100 conexiones entrenables. El sistema descrito basado en fotogramas tiene 506850 conexiones y sólo 1700 parámetros entrenables. Todos los parámetros entrenables se han calculado utilizando el algoritmo backpropagation adaptado para redes convolucionales (pesos compartidos) [1].

III. RED NEURONAL CONVOLUCIONAL BASADA EN EVENTOS

En la Fig. 2 se muestra la estructura de la red neuronal libre de fotogramas y basada en AER. Un conjunto de imágenes de dígitos de la base de datos MNIST [1] fueron codificadas a eventos separados entre sí 50*ns*, creando de este modo estímulos de duración 0.23*ms* en promedio.

Los eventos correspondientes a cada dígito fueron utilizados como entrada en el sistema. Como se muestra en la Fig. 2, cada evento de entrada es repetido en 6 canales diferentes haciendo uso de un módulo splitter 1-6 [7][4]. Estas seis réplicas están conectadas a la capa C1, que consta de seis módulos de convolución AER [6]. La etapa C1 implementa un banco de filtros de Gabor con 2 escalas y 3 orientaciones. Cada módulo de convolución dispone internamente de una matriz de píxeles de tamaño 28x28 y un filtro de gabor de tamaño 10x10 utilizado como máscara de convolución. Los eventos obtenidos a la salida de la etapa C1 de convolución son enviados a la etapa S2 de submuestreo, que implementa una reducción del espacio de direcciones de los eventos. Estos módulos simplemente modifican la dirección de entrada codificada por cada evento. De este modo, cada evento de entrada *(xin, yin)* codificando una dirección *(i,j)*, con *i,j=1,…,28*, se modifica para obtener una nueva dirección *(k,l)* con *k,l=1,…,14*. Los eventos de salida

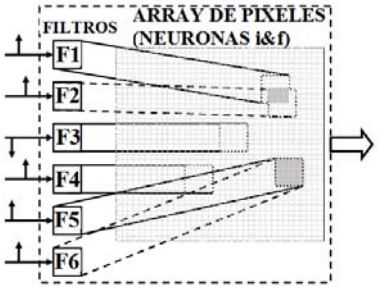

Fig. 3. Estructura de convolución utilizada en las etapas C3 y C5. Cada evento de entrada hace que una máscara de convolución (filtro) se añada en el array de píxeles.

de la etapa de submuestreo son replicados en cuatro canales utilizando módulos splitter 1-4. Cada evento correspondiente a uno de los cuatro canales de salida de los módulos splitter se envía a una de las estructuras de convolución de la etapa C3. Cada estructura de convolución tiene seis puertos de entrada disponibles. Una descripción detallada de una de estas estructuras se muestra en la Fig. 3.

En estas estructuras, cada vez que se recibe un evento, la máscara de convolución correspondiente al puerto de entrada y almacenada en la estructura se añade alrededor de la dirección codificada por el evento en la matriz de píxeles. Este tipo de estructuras se pueden implementar con los recientemente desarrollados chips AER de convolución multikernel [5] (con hasta 32 máscaras de convolución). Cada vez que una neurona (unidad o píxel) en la matriz de píxeles alcanza un umbral determinado y además el tiempo transcurrido desde el último evento generado de salida es mayor que un tiempo refractario establecido (*Trefractory*), un nuevo evento de salida se envía a las siguientes capas de procesamiento. La neurona que emite el evento es reseteada a continuación. El uso de tiempos refractarios para limitar la tasa de disparos máxima de las neuronas, emula la rectificación implementada por las funciones no lineales utilizadas en los modelos habituales de neuronas en redes neuronales basadas en fotogramas. La etapa S4 implementa submuestreo del mismo modo que la etapa S2. Los eventos obtenidos a la salida de la etapa S4 se envían de nuevo a estructuras de convolución en C5 como las usadas en la etapa C3. Los eventos producidos en la etapa C5 se replican en diez canales diferentes utilizando módulos splitter 1-10. Todos los eventos procedentes de la etapa C5 se envían a la etapa F6, compuesta por diez neuronas *i&f* (neuronas de integración y disparo). Estas neuronas disparan eventos positivos o negativos indicando si el número a la entrada al sistema corresponde a la categoría codificada por la neurona de salida. En el sistema basado en fotogramas, se consideraron funciones sigmoides no lineales en las etapas C3, C5 y F6. Sin embargo, en la versión AER sólo se han considerado no linealidades (implementadas mediante tiempos refractarios) en las etapas C3 y C5. El no utilizar no linealidades en la etapa F6 no afecta al reconocimiento de los dígitos de entrada, ya que la salida no saturará pero será positiva (y alta) sólo en la neurona de salida correspondiente al dígito de entrada.

IV. RESULTADOS

En esta sección ofrecemos una evaluación realista del funcionamiento que tendría la implementación hardware del sistema descrito no basado en fotogramas. Para ello hemos utilizado un simulador de sistemas AER desarrollado y validado en C++ [8] que permite probar y testear sistemas modulares AER multi-chip antes de su implementación física. Los parámetros y características de funcionamiento de los módulos AER utilizados corresponden a valores de módulos AER físicos existentes y disponibles [4][6][7].

En el sistema basado en AER hay principalmente tres tipos de parámetros que se deben configurar antes de utilizar el sistema. Estos parámetros son los pesos de las máscaras de convolución, los umbrales de disparo en los módulos de convolución y los tiempos refractarios que modelan las no-linealidades. Los pesos de las conexiones se obtuvieron entrenando la red neuronal basada en fotogramas descrita en el punto 2 utilizando el algoritmo *backpropagation*. Para ello se utilizaron 50000 imágenes de dígitos de la base MNIST. Todos los pesos obtenidos en esta primera etapa se almacenaron para ser usados en el sistema AER en la implementación basada en eventos y libre de fotogramas.

Los valores para los umbrales en el sistema AER se establecieron con valores entre 1.5 y 2 veces el valor del peso máximo existente en cada módulo de convolución. De este modo se evita un ruido de cuantización alto y se mantiene una alta tasa de eventos a la salida.

Los tiempos refractarios en las capas C3 y C5 se calcularon estableciendo relaciones matemáticas entre los valores de saturación de las funciones sigmoideas en la versión basada en fotogramas y la máxima tasa de eventos en la versión basada en AER. De este modo se obtuvieron los valores 0,36μs y 0,65μs para los tiempos refractarios en las capas C3 y C5, respectivamente.

Para hacer el sistema basado en AER aún más parecido a uno biológico, se incorporaron mecanismos de olvido en las neuronas pertenecientes a las etapas C1, C3, C5 y F6. El mecanismo de olvido es importante porque "*vacía*" el estado almacenado en las neuronas (se olvida), de modo que la información pasada no es relevante para los cálculos. Los valores para los factores de olvido se eligieron empíricamente. El límite superior para estos factores corresponde a obtener una salida nula cuando no se utilizan tiempos refractarios (todas las neuronas vacian por completo sus estados de modo muy rápido).

Una vez que los principales parámetros fueron establecidos, el sistema AER fue testeado con 10.000 imágenes de dígitos de la base de datos MNIST. Estas imágenes fueron codificadas en eventos separados entre sí 50ns, creándose de este modo 10.000 flujos de eventos con una duración promedio de 230μs.

Eventos positivos en una de las neuronas de salida indica que el sistema ha reconocido los eventos de entrada como pertenecientes al dígito codificado por la neurona de salida que produce la actividad positiva. Eventos negativos indican lo contrario. Se considera que los eventos recibidos a la entrada del sistema codifican el dígito codificado por la neurona de salida que produzca la actividad positiva más alta. De este modo, obtuvimos una tasa de reconocimiento del 93%. En las pruebas efectuadas interesa mucho obtener una alta tasa de eventos a la salida, minimizando así el tiempo de respuesta del sistema y un tiempo mímimo para obtener el primer evento positivo de salida desde que se suministra el primer evento de entrada al sistema. En nuestro sistema, el tiempo mínimo conseguido para obtener el primer

evento de salida positivo fue de 4,3μs. La tasa máxima de disparo en cada canal de salida correspondió a tiempos entre eventos mínimos del orden de 3,1μs.

Un segundo experimento consistió en analizar la respuesta del sistema cuando el flujo de entrada se cambia alternativamente entre dígitos diferentes. En la Fig. 4, se muestran los eventos de entrada y de salida obtenidos. Los eventos a la entrada codifican un espacio de direcciones 32x32 (correspondiente al tamaño de la imagen de entrada), sin embargo, a efectos de simplificación, hemos representado tanto los eventos de entrada como los de salida codificando los diferentes dígitos con valores del 1 al 10 (los dígitos van de 0 a 9). Los eventos de entrada se representan con puntos negros. Los eventos de salida están representados con círculos azules. Las neuronas de salida producen actividad positiva o negativa (eventos con signo positivo o negativo). Véase en la figura como el sistema es capaz de detectar el dígito a la entrada en tiempo mínimo produciendo la actividad positiva más alta sólo el canal de salida correspondiente al dígito de entrada. En este segundo experimento, el tiempo promedio obtenido para que el sistema produzca una nueva salida positiva correcta después de una transición del estímulo de entrada fue de 22,4μs. El tiempo mínimo en producir el primer evento positivo a la salida después de una transición a la entrada fue de 4,3μs.

V. CONCLUSIONES

A lo largo de este trabajo hemos presentado una red neuronal convolucional multichip multietapa para la detección de dígitos numéricos usando una configuración basada en eventos y por tanto libre de fotogramas.

La red neuronal convolucional junto con el protocolo AER permite una gran reducción en el número de parámetros entrenables y en el número de conexiones. Téngase en cuenta que el sistema basado en fotogramas descrito requeriría 506850 conexiones y sólo 1700 parámetros entrenables gracias a la técnica *weight-sharing* (uso compartido de los pesos). La multiplexación en el tiempo que implementa el protocolo AER permite que el número de conexiones se reduzca a 195, ya que todas las conexiones entre unidades pertenecientes a mapas de características de etapas consecutivas pueden ser multiplexadas en un único bus digital de alta velocidad (canal).

El sistema ConvNet Lenet-5 original propuesto por Y. LeCun requería 340908 conexiones y 60840 parámetros entrenables. Esta red proporcionaba una tasa de reconocimiento superior al 98%. A pesar de que nuestro sistema ofrece una tasa de reconocimiento ligeramente inferior (93%), sólo requiere 1700 parámetros entrenables, lo que corresponde a un 2,79% del número total de parámetros entrenables necesarios en el original Lenet-5. El sistema completo puede ser implementado con sólo 30 módulos de convolución. El rendimiento podría aumentarse mediante la adición de más módulos en paralelo en cada etapa sin penalizar a los retrasos, aunque al costo de agregar hardware.

En el sistema AER propuesto, el tiempo mínimo para conseguir el reconocimiento fue de 4,3μs, que corresponde a sólo el 1.87% de los eventos de entrada que codifican a los diferentes dígitos. Como la separación media entre eventos es de 50ns, 4,3μs corresponden sólo a 86 eventos de entrada aproximadamente.

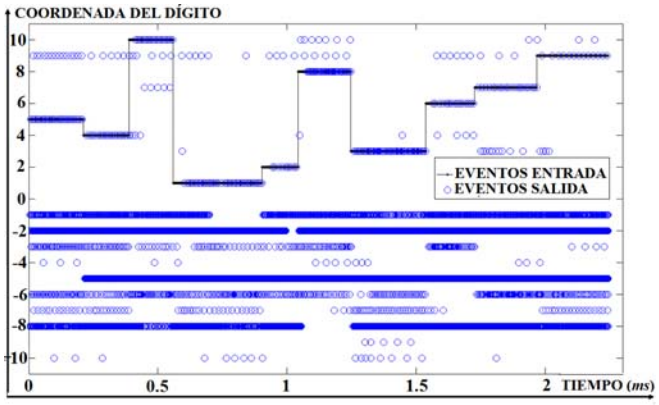

Fig. 4. Eventos de entrada y de salida obtenidos en el sistema.

El diseño de sistemas como el descrito en este trabajo se está convirtiendo en una realidad ya que las ConvNets pueden ser implementadas eficientemente con las tecnologías híbridas CMOS/noCMOS de escala nanométrica [9]. Los sistemas futuros serán capaces de incorporar técnicas de aprendizaje no supervisadas en el diseño, tales como la técnica STDP [10].

AGRADECIMIENTOS

Este trabajo ha sido financiado en parte por el proyecto TEC2009-10639-C04-01 (VULCANO) y el proyecto andaluz P06-TIC-01417 (Brain System). JAPC ha sido financiado por el proyecto andaluz P06-TIC-01417 (Brain System).

REFERENCIAS

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2323, 1998.

[2] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio, “Robust object recognition with cortex-like mechanisms,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 3, pp. 411–426, 2007.

[3] K. A. Boahen, “Point-to-point connectivity between neuromorphic chips using address events,” IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, vol. 47, no. 5, pp. 416–434, 2000.

[4] R. Serrano-Gotarredona, et al., “Caviar: A 45k neuron, 5m synapse, 12g connects/s aer hardware sensory-processing-learning-actuating system for high-speed visual object recognition and tracking,” IEEE Transactions on Neural Networks, vol. 20, no. 9, pp. 1417–1438, 2009.

[5] L. Camuñas-Mesa, et al., “Improved Aer Convolution Chip for Vision ProcessingWith Higher Resolution and New Functionalities,” in Conference on Design of Circuits and Integrated Systems, DCIS, no. 7, pp. 1–6, Barcelona, Spain, 2009.

[6] L. Camuñas-Mesa, et al., “A 32x32 pixel convolution processor chip for address event vision sensors with 155ns event latency and 20 meps throughput,” IEEE Transactions on Circuits and Systems, 2010.

[7] F. Gómez-Rodríguez, et al., “Aer tools for communications and debugging,” IEEE International Symposium on Circuits and Systems, ISCAS, pp. 3253–3256, Kos, Greece, 2006.

[8] J. A. Pérez-Carrasco, et al., “Advanced vision processing systems: Spike-based simulation and processing”, LNCS, vol. 5807, pp. 640–651, 2009.

[9] D. B. Strukov and K. K. Likharev, “Cmol FPGA: A reconfigurable architecture for hybrid digital circuits with two-terminal nanodevices,” Nanotechnology, vol. 16, no. 6, pp. 888–900, 2005.

[10] T. Masquelier, R. Guyonneau, and S. J. Thorpe, “Competitive stdp-based spike pattern learning,” Neural computation, vol. 21, no. 5, pp. 1259–1276, 2009.