

REDES NEURONALES CONSTRUCTIVAS QUE APRENDEN MIENTRAS CRECEN

Juan Manuel TORRES-MORENO

Lania, A.C. Rébsamen 80, 91090 Xalapa, Veracruz, México
torres@xalapa.lania.mx <http://www.xalapa.lania.mx/~torres>

Resumen

En este artículo se presentan una introducción a la teoría del aprendizaje supervisado y a los fundamentos de las redes de neuronas. Se muestran las arquitecturas clásicas utilizadas y se introducen nuevos paradigmas incrementales, de redes que aprenden conforme crecen. Se trata el problema de la clasificación de datos y se muestran algunas aplicaciones al mismo.

INTRODUCCION

¿Pueden las computadoras aprender a resolver problemas a partir de ejemplos? Esta cuestión que bordeaba no hace mucho tiempo en la frontera de la ciencia ficción es actualmente objeto de profundos estudios. Las redes de neuronas formales son máquinas que poseen esta capacidad de aprendizaje. Estas máquinas han sido propuestas como modelos extremadamente simplificados del funcionamiento del cerebro, puesto que no retienen más que algunas características esenciales: i) las neuronas no pueden encontrarse más que en dos estados posibles, activas o en reposo; ii) están interconectadas mediante sinápsis que pueden ser modificadas por aprendizaje y iii) el estado de una neurona a cada instante es determinado por el de otras neuronas, información que es transmitida por las sinápsis. Aunque muy esquemático, este modelo presenta una riqueza sorprendente de estados y de comportamientos y ha sentado las bases de un modelo de memoria y aprendizaje como un fenómeno emergente colectivo: el sistema global presenta propiedades complejas que no pueden predecirse a partir del estudio individual de sus componentes. Así, el todo es mucho más que la suma de sus partes. Por otro lado, las aplicaciones en diversos dominios no tardaron en aparecer. Los estudios teóricos de redes de neuronas reflejan estos dos aspectos: el del modelado de fenómenos cognoscitivos y el del desarrollo de aplicaciones. Aunque las redes de neuronas hayan sido aplicadas a diversos campos, en el presente artículo se hará énfasis en la clasificación de datos: para los humanos una actividad tan trivial que pasa desapercibida, pero que presenta dificultades importantes para una máquina.

APRENDIZAJE Y CLASIFICACION

La clasificación es la atribución de una clase específica a un objeto. Esta atribución necesita un cierto grado de abstracción para poder extraer generalidades a partir de los ejemplos de los cuales se dispone. Para una máquina, la clasificación de rostros, de datos médicos o de formas son tareas bastante difíciles, en tanto que para un humano son cuestiones cotidianas. Por ejemplo, en el caso de reconocimiento de caracteres manuscritos, es difícil enunciar una descripción general que tenga en cuenta todas las variaciones particulares de cada carácter. Una técnica que puede ser utilizada para resolver este problema es el aprendizaje. Así, el criterio para decidir si una imagen corresponde a una letra "A" consiste en comparar si esta imagen es lo suficientemente

similar a otras "A"s vistas anteriormente; con este enfoque, uno no *calcula* la clasificación de letras, sino que se *aprenden* a partir de ejemplos.

El aprendizaje está implícito en el quehacer humano y forma parte imprescindible de las actividades intelectuales. Surge entonces la cuestión inevitable: *¿qué es el aprendizaje?* El aprendizaje -desde un punto de vista pragmático- consiste en la adaptación de los parámetros de un sistema (sea artificial o natural) para obtener una respuesta deseada frente a un estímulo externo. Esta definición amplia del aprendizaje puede ser formalizada con el paradigma del aprendizaje supervisado o del *profesor-alumno*: imagine que existe un profesor que conoce la relación exacta entre los estímulos y las respuestas (entradas-salidas), que un alumno desconoce. Si ambos son expuestos a una entrada cualquiera, el profesor es capaz a cada momento de indicar al alumno la respuesta correcta. Los parámetros del alumno deben ser entonces, ajustados para entregar la misma respuesta que la del profesor. Este ajuste o *aprendizaje* se puede realizar por técnicas iterativas de minimización de un costo (cuantificación de los errores en las respuestas). Por supuesto, en la vida real no existen tales profesores, y de lo único que se dispone son datos en forma de pares de entrada-salida a los que llamaremos *ejemplos*. Si se posee un cierto número de estos ejemplos, entonces se tiene un *Conjunto de Aprendizaje*. En efecto, es únicamente a partir del *Conjunto de Aprendizaje* que son construidos los clasificadores, en el caso que nos ocupa: las redes neuronales.

NEURONAS FORMALES

El origen de las Redes de Neuronas (RN) se encuentra en la representación de la neurona biológica por McCulloch y Pitts en 1943. Una red de neuronas formales es esquematizada en la figura 1. La red posee un gran número de unidades en interacción.

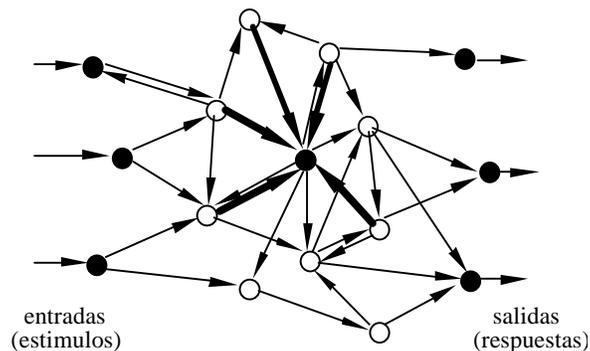


Figura 1. Red de neuronas formales. Las neuronas grises pertenecen a las capas de entrada y de salida. Una neurona particular es indicada en negro. Sus sinápsis de entrada son representadas en línea gruesa.

El modelo de McCulloch y Pitts considera que una neurona puede ser representada por una unidad binaria: a cada instante su estado σ , puede ser activo, $\sigma=1$ o inactivo, $\sigma=-1$. En otros modelos, el estado de la neurona es descrito por una variable continua que toma valores entre $[-1, +1]$. La interacción entre las neuronas se lleva a cabo a través de sinápsis (o *pesos sinápticos*) w . Según el signo de w , la sinápsis es excitadora

($w > 0$) o inhibitoria ($w < 0$). Las convenciones sobre la notación utilizada se resumen en la figura 2.

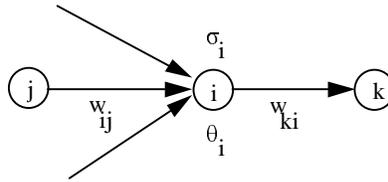


Figura 2. La neurona formal i , con un estado σ_i , con sus sinápsis de entrada y su salida. w_{ij} es el peso sináptico de una neurona j sobre i , y w_{ki} la interacción de i sobre k . El umbral de activación es θ_i

Es generalmente admitido que las sinápsis pueden ser modificadas por la experiencia, es decir, por aprendizaje. Para su análisis, es conveniente considerar una sola neurona con sus sinápsis. Esta unidad es llamada *perceptron*, y constituye la base de las redes de neuronas.

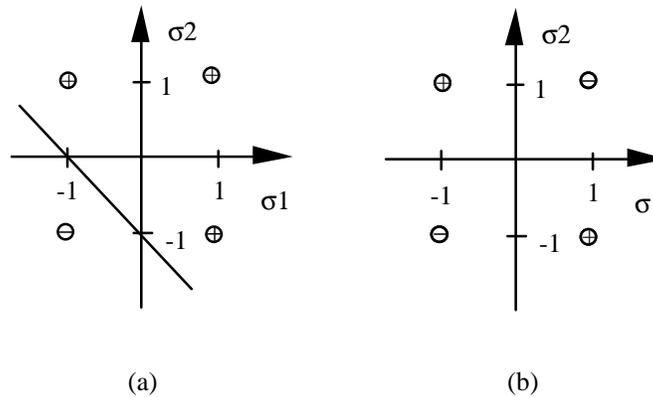


Figura 3. Espacio de entradas de un perceptron con $N=2$. Sobre cada eje se representan los dos estados posibles de las neuronas de entrada. Cada círculo representa una entrada posible, y el signo es la salida correspondiente.
(a) función linealmente separable (AND) (b) función no realizable por un perceptron (XOR)

El perceptron está constituido por las N entradas provenientes de fuentes externas, las N conexiones o pesos $\mathbf{w} = (w_1, w_2, \dots, w_N)$ y la salida σ . En realidad un perceptron es la red neuronal más simple posible: aquella donde no existen capas ocultas. Para cada configuración de los estados de las neuronas de entrada (estímulo) la respuesta del perceptron obedece la siguiente dinámica: sumar los *potenciales sinápticos* $w_{ij}\sigma_j$ (ver figura 2). Esta suma ponderada, también llamada *campo*, se escribe:

$$h_i = \sum_{j=1}^N w_j \sigma_j$$

Si $h_i > \theta_i$, la respuesta de la neurona es $\sigma_i = +1$; si no, es inactiva y $\sigma_i = -1$. Si tanto las entradas como las salidas son binarias; se dice entonces que el perceptron realiza una función booleana de sus entradas. Pero otras posibilidades han sido estudiadas (entradas reales, salidas sigmoideas en función del campo, etc.), en vista de las

aplicaciones. Con una arquitectura tan simple como la del perceptron no se pueden realizar más que una clase de funciones booleanas muy simples, llamadas *linealmente separables*. Son las funciones en las cuales los estados de entrada con salida positiva pueden ser separados de aquellos a salida negativa por un hiperplano. Un *hiperplano* es el conjunto de puntos en el espacio de estados de entrada, que satisfacen una ecuación lineal. En dos dimensiones, es una recta, en tres dimensiones un plano, etc. La figura 3 presenta dos ejemplos de funciones booleanas con 2 entradas ($N=2$). El ejemplo 3b no es realizable por un perceptron. Si se quieren realizar funciones más complejas con redes de neuronas, es necesario intercalar neuronas ocultas entre las entradas y la salida.

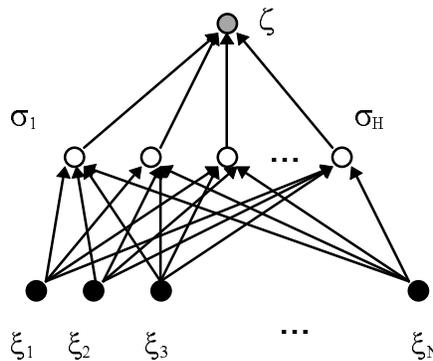


Figura 4. Red neuronal con una capa oculta y H unidades σ

Una red multicapas se define como un conjunto de perceptrones, ligados entre si por sinápsis y dispuestos en capas siguiendo diversas arquitecturas. Una de las arquitecturas más comúnmente usada es llamada *feedforward*: se tienen conexiones de la entrada a las capas ocultas y de aquí a la salida. En la figura 4 se muestra una RN *feedforward* con N entradas $\xi=(\xi_1, \xi_2, \dots, \xi_N)$ y una capa oculta de H unidades ocultas. El perceptron de salida es denotado por ζ .

El funcionamiento de una RN es gobernado por reglas de propagación de actividades y de actualización de los estados. Teóricamente, una RN puede ser vista como un modelo que realiza una función de un espacio de entrada hacia un espacio de salida. El objetivo de esta modelización consiste en que la asociación sea lo más acorde posible con el medio ambiente del fenómeno estudiado. Observe en la figura 4 que el perceptron de salida ζ tiene ahora como entradas las *salidas* de las unidades ocultas σ : es justamente aquí donde se realiza el mapeo del espacio de entrada hacia estados ocultos llamados *representaciones internas*, que son una codificación de la información, y de éstas a la salida.

La Inteligencia Artificial clásica [12,13] (IA) ha generado una cierta decepción al tratar de explicar los procesos cognoscitivos debido a que la representación usando reglas se aleja mucho de cualquier inspiración biológica por una parte y por otra conduce a la creación de sistemas que son demasiado rígidos y al mismo tiempo extremadamente frágiles. El reconocimiento, el aprendizaje y la memoria son mecanismos cognoscitivos que no pueden ser explicados por medio del simbolismo de la IA, sino mas bien en función de unidades simples pero altamente interconectadas de las cuales *emerge* un comportamiento complejo, paralelo y auto-organizado, sin necesidad de tener reglas

explícitas de decisión ni de un procesador maestro o motor de inferencia. La secuenciación y la rigidez de reglas son dos aspectos que han sido duramente atacados en las técnicas de IA. La inmensa ventaja de los métodos conexionistas comparado con los métodos tradicionales de IA es la siguiente: no es necesario conocer ni una expresión ni una construcción de la función a modelar, tan solo se requiere disponer de un conjunto de aprendizaje satisfactorio para que la red pueda aproximar esta función aplicando una *regla de aprendizaje*. Así, muchos fenómenos cognoscitivos han logrado ser modelados a través de sistemas conexionistas. Por otra parte, en lo que concierne a la aplicación tecnológica, las redes de neuronas son actualmente ampliamente utilizadas en aplicaciones tan variadas como la previsión, la clasificación, el diagnóstico automático, el procesamiento de señales, el reconocimiento de formas, la compresión de datos, la optimización combinatoria, la robótica y la búsqueda de documentos, entre otras.

La mayor parte de las reglas de aprendizaje se basan en el ajuste iterativo de pesos de las conexiones, pero otras pueden modificar la arquitectura misma de la red. Un modelo conexionista es definido por el tipo de neuronas, por la arquitectura y por una regla de aprendizaje. Las características deseables y comunes a todos estos modelos son: un fuerte potencial de auto-organización, una cierta robustez frente a perturbaciones externas (gracias a una memoria distribuida, deslocalizada y a la redundancia de la información) y un paralelismo masivo e inherente. Sin embargo, los modelos difieren entre sí por diversos aspectos: sus motivaciones biológicas, su modo de funcionamiento o su campo de aplicaciones.

ARQUITECTURAS CLASICAS vs. ARQUITECTURAS INCREMENTALES

El aprendizaje con redes de neuronas se realiza actualmente siguiendo dos enfoques: algunos algoritmos derivados de la Retropropagación de Gradiente (*Backpropagation*) necesitan introducir *a priori* el número y las conexiones de las unidades ocultas, y determinar los pesos por minimización de un costo. La red así obtenida es eventualmente simplificada eliminando unidades y/o conexiones que parecen inútiles. El principal defecto de estas arquitecturas fijas consiste en la búsqueda de la arquitectura óptima a través de prueba y error.

Por otra parte, con un enfoque constructivo se aprende *al mismo tiempo* el número de unidades y pesos, en el marco de una arquitectura que comienza generalmente con una unidad oculta. La característica de estos algoritmos es que construyen una red de neuronas adaptada a cada problema particular, usando la información contenida en el conjunto de aprendizaje y evitando pruebas y errores en el diseño de la arquitectura. De los primeros algoritmos constructivos desarrollados en Europa, se tiene al algoritmo *Tiling*, y en Estados Unidos a *Cascade Correlation* [5]. A partir de ahí otros algoritmos han sido desarrollados, entre ellos el algoritmo *Upstart*, *Offset*, y *GAL*. *Monoplan* y *NetLines* son dos algoritmos recientemente introducidos por el autor. De aquí en adelante, la discusión será basada únicamente en los métodos constructivos.

Una vez construida la red, ésta debe ser capaz de *predecir* la clase de datos nuevos que no están presentes en el conjunto de aprendizaje. La calidad del algoritmo de aprendizaje se traduce en la capacidad de predicción de la red de neuronas. Esta calidad se mide a través del *error de generalización*, que es la proporción de clasificaciones correctas realizadas por la red sobre nuevos datos. Esta cantidad se mide

empíricamente, sobre una serie de problemas estándar (*benchmarks*) que sirven de prueba.

DOS NUEVOS ALGORITMOS CONSTRUCTIVOS

Los algoritmos constructivos adaptan la cantidad de neuronas ocultas a los datos. Es decir, el tamaño de la red es aprendido al mismo tiempo que los pesos sinápticos, lo que significa una gran ventaja sobre las arquitecturas rígidas. Recientemente se han introducido algoritmos constructivos que usan una sola capa de unidades ocultas binarias [9]. Esta elección no fue fortuita: se requieren redes que sean comprensibles, y compactas: es decir fáciles de interpretar. Ciertos teoremas mostraban que una sola capa oculta, con un número suficiente de unidades es capaz de realizar cualquier función de las entradas. La palabra clave es *suficiente*: ¿cuántas unidades son suficientes? en virtud del carácter constructivo de las redes, la pregunta se traduce en *¿hasta cuando se deben agregar unidades a la capa oculta?* Los teoremas no proporcionan más información, pero intuitivamente se decidió que la respuesta estaba justamente en los datos: el número de unidades dependerá de cada problema en particular. De esta manera, se decidió crear arquitecturas con una sola capa oculta cuyo tamaño fuera calculado automáticamente por el mismo algoritmo de aprendizaje. Rompiendo así con el círculo perverso de prueba y error, característico de la *Backpropagation* que puede llegar a consumir un tiempo asombrosamente grande en el diseño de la red.

El algoritmo resultante, llamado *Monoplan*[1] trabaja de la siguiente manera: suponga que tiene un conjunto de aprendizaje formado de ejemplos de la clase “+” y “-“. Un primer perceptron σ_1 es entrenado para separar lo mejor posible las dos clases. Si no hay errores, el conjunto de ejemplos es linealmente separable; si no lo es, entonces un segundo perceptron σ_2 es agregado para corregir los errores cometidos por el perceptron anterior: los errores son marcados ahora como “-“ y los ejemplos bien aprendidos como “+“. Este proceso iterativo agrega sucesivamente neuronas σ en la capa oculta hasta que sean clasificados correctamente todos los ejemplos del conjunto de aprendizaje. Una vez que esto ocurre, se construye el perceptron de salida ζ , usando como entradas los estados de las neuronas ocultas (representaciones internas) y las clases originales “+” y “-“ a aprender. Numerosas aplicaciones [3,4,10] de *Monoplan* a *benchmarks* mostraron su excelente desempeño, y sugirieron al mismo tiempo una variante del método: ¿qué sucedería si en lugar de tratar de clasificar los ejemplos en la capa oculta se clasificaran a la *salida* de la red? El producto de esta idea es *NetLines* [2], poderoso algoritmo de clasificación que puede calificarse como “método impaciente”, pues trata de construir el perceptron de salida ζ que separe las clases *cada vez que una nueva unidad es añadida*, sin esperar que en la capa oculta sean bien clasificados todos los ejemplos (figura 5). El perceptron de salida es alimentado con las representaciones internas σ disponibles hasta ese momento tratando de separar las dos clases; si no lo logra, entonces otra nueva unidad oculta es agregada, para corregir los errores cometidos por la unida de salida, marcándolos como “-“ y separándolos de los “+” bien aprendidos, en forma iterativa. Este método permite tratar adecuadamente los casos donde las entradas están codificadas con números reales, en tanto que *Monoplan* está mejor adaptado a tratar entradas binarias.

Ambos algoritmos utilizan unidades ocultas de tipo binario (codificadas en particular como ± 1) lo que permite por una parte, una utilización inmediata en problemas de dos clases (comúnmente usados en aplicaciones prácticas) y por otra parte, se centran en el problema *pertinente* de la búsqueda de las fronteras entre clases, a diferencia de la *Backpropagation* donde la simple disminución del costo cuadrático medio no proporciona información de donde se hallan estas fronteras. Eventualmente, las fronteras pueden ser mas fácilmente interpretadas.

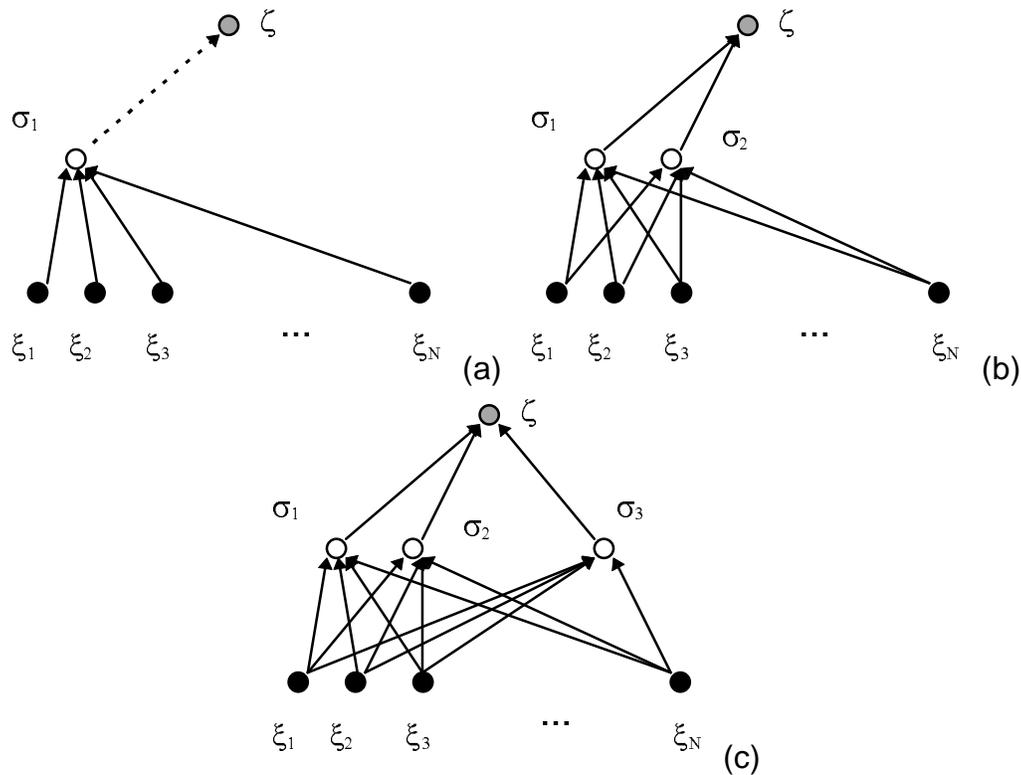


Figura 5. Crecimiento incremental con NetLines con 3 unidades ocultas y N entradas. En (a) la red se reduce a un perceptron simple si el problema es linealmente separable. En (b) y (c) se agregan unidades ocultas que permitan separar correctamente todos los ejemplos de la clase positiva de aquellos con clase negativa. La salida ζ es construida usando como entradas las Representaciones Internas σ correspondientes a los estados de activación de las unidades ocultas en función de las entradas externas ξ . Los pesos son calculados con el algoritmo de aprendizaje *Minimerror*.

La clave importante de estos nuevos algoritmos consiste en el *correcto* aprendizaje de cada una de las unidades individuales, ya que es este punto el que determinará la complejidad de la red. El algoritmo estándar del perceptron no permite aprender más que conjuntos que son linealmente separables (por tanto, inutilizable en la mayor parte de los casos) y no se detiene jamás si el conjunto no lo es. Algoritmos como *Pocket*, que guarda la mejor solución a cada iteración y que garantiza hallar una buena solución en un tiempo suficientemente largo, dependen de la paciencia del usuario. Por ello, se adaptó un reciente algoritmo de aprendizaje basado en Mecánica Estadística llamado *Minimerror* [11], que utiliza una función de costo involucrando las estabilidades γ (o distancias de los ejemplos al plano separador) como información para situar al plano separador lo más cerca posible de la frontera entre clases: γ es positiva si el ejemplo

está bien clasificado (clase del lado correcto del plano \mathbf{w}) o negativa si está del lado incorrecto (figura 6)

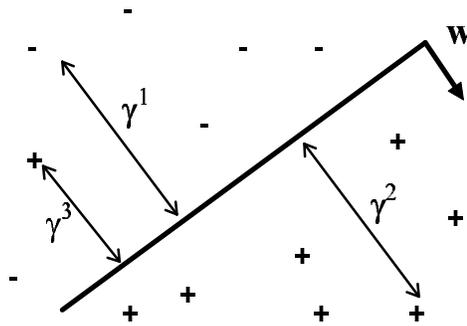


Figura 6. Estabilidades γ que representan distancias al plano separador: γ^1, γ^2 son positivas porque los ejemplos están bien clasificados mientras que γ^3 es negativa: es un ejemplo mal clasificado por el vector \mathbf{w}

La estabilidad de un ejemplo cualquiera μ se define por el producto punto del vector de entradas por el vector que representa el hiperplano separador (normalizado), con su respectiva clase:

$$\gamma^\mu = \frac{\vec{\xi} \cdot \vec{w}}{\|\vec{w}\|} \tau^\mu$$

La función de costo utilizada es:

$$E = -\frac{1}{2} \left(1 - \tanh \left(\frac{\gamma^\mu}{2T} \right) \right)$$

E cuantifica de cierta manera el número de errores de aprendizaje (aquellos ejemplos con γ negativa) sobre todo el conjunto de P ejemplos de aprendizaje, $\mu=1, \dots, P$. Lo que se desea es *minimizar* el número de errores cometidos por el hiperplano separador. La función de costo E esta modulada por la pseudo-temperatura T : a alta temperatura (T cercana a infinito) E es una función de tipo escalón que cuenta exactamente el número de errores de aprendizaje, pero a una temperatura finita, E resulta ser una función continua y derivable, lo que permite una búsqueda de su mínimo con un descenso gradiente. La temperatura T variable permite efectuar un aprendizaje iterativo utilizando la información pertinente de los ejemplos próximos a las fronteras a través de la derivada de la función de costo, que es una especie de campana centrada en el hiperplano separador. Se minimiza la función de costo a una temperatura finita (T decreciente en el tiempo) y se actualizan los pesos de la neurona en el tiempo t a través de un descenso en gradiente simple del tipo:

$$\vec{w}(t+1) \leftarrow \vec{w}(t) + \delta \vec{w}(t)$$

donde:

$$\delta \vec{w}(t) = \varepsilon \frac{\partial E}{\partial w_i} = -\frac{\varepsilon}{4T} \frac{\sum_{\mu} \xi_i^{\mu} \tau^{\mu}}{\cosh^2\left(\frac{\gamma^{\mu}}{2T}\right)}$$

El parámetro ε representa la tasa de aprendizaje. T permite escapar a posibles mínimos locales a través de un proceso tipo recocido determinista que disminuye iterativamente la temperatura. Este método produce un aprendizaje robusto [3] y resultados comparables a aquellos obtenidos con el perceptron de Bayes [6,7]. Este algoritmo permitió descubrir que un *benchmark* (problema del Sonar) conocido en la literatura por su gran dificultad, es en realidad, linealmente separable [8]. Esta prueba mostró la gran calidad de *Minimerror* al enfrentarse a problemas de alta dimensionalidad. El binomio *Minimerror-Monoplan* y *NetLines* parece ser una nueva fórmula dentro de los paradigmas del aprendizaje supervisado que muestra un alto desempeño en aprendizaje y generalización sobre datos académicos y o realistas, como se muestra en la siguiente sección.

APLICACIONES A LA CLASIFICACIÓN AUTOMÁTICA

Desde un punto de vista metodológico, el uso de redes neuronales multicapas no es trivial, incluso cuando el algoritmo sea fácil de implementar. La dificultad viene del hecho que el tamaño de la red depende fuertemente de la naturaleza de la aplicación a tratar, y que los raros resultados teóricos actualmente conocidos son solo fronteras inaplicables directamente a los casos prácticos. La correcta calibración de los parámetros de aprendizaje y de la arquitectura de la red son cruciales. Sin embargo, cuando las redes son correctamente adaptadas a un problema, permiten alcanzar a menudo tasas superiores de aprendizaje y generalización que métodos de IA o estadísticos.

En el caso de *Monoplan* y *NetLines*, la mayor parte de los parámetros se calibrados por si solos, de tal manera que permiten una cierta independencia del usuario. Este último no tiene que preocuparse del tedioso problema de definir ninguna arquitectura: la red aprende mientras crece en un proceso dinámico y autónomo. Las pruebas de ambos algoritmos han mostrado sin lugar a dudas su extraordinaria calidad, tanto desde el punto de vista de la generalización como del tamaño (pequeño) de las redes generadas. He dividido las aplicaciones en temas académicos y reales o realistas. Entre los primeros se hallan los problemas clásicos del XOR llevado a su generalización llamada *Paridad* de N entradas (la clase es positiva si el número de bits en el estado "1" es par o negativa si es impar). En este problema tanto *Monoplan* como *NetLines* encontraron la solución "humana" que consiste en una red con una capa oculta y un número de unidades ocultas igual al número de entradas. Los problemas de *Monk's*, de formas de ondas de Breiman y el problema sintético de dominios magnéticos fueron igualmente resueltos con éxito [4,5,9,10].

Los problemas reales o realistas plantean dificultades interesantes: como provienen de mediciones del mundo real, generalmente los datos poseen ruido o están incompletos, es decir: una o mas de sus atributos no existen; además, las bases son generalmente pequeñas y, por lo tanto, hay pocos ejemplos para construir el clasificador. La dimensionalidad del espacio de entradas puede ser alta: en el problema de ecos de

sonar, por ejemplo se trabaja con espectros muestreados en un espacio de 60 dimensiones. Aun así, los clasificadores deben ser prudentemente probados con estas aplicaciones. En ese problema, por ejemplo, la *Backpropagation* [14] reporta resultados muy inferiores a *Minimerror*. Un área interesante de aplicación es la ayuda al diagnóstico médico: aquí *Monoplan* y *NetLines* fueron confrontados a los problemas de detección de cáncer de seno y de diabetes, usando datos de bases de dominio público en Internet. Los resultados [2,8,9] son realmente satisfactorios, pues muestran que las redes generadas por estos dos algoritmos tienen la mejor relación en generalización y complejidad de la red, comparados contra redes entrenadas con *Backpropagation*, redes constructivas como *Cascade Correlation*, métodos de IA y métodos estadísticos. El diagnóstico de comas toxicológicas en el hospital universitario de Grenoble y el control de calidad de tubos de acero son dos aplicaciones en curso, de las cuales se tienen resultados preliminares prometedores [9].

CONCLUSIONES

Este panorama (de las redes neuronales) ha permitido mostrar que partiendo de motivaciones más o menos biológicas, numerosas clases de redes han sido propuestas estos últimos años. Los modelos conexionistas difieren por sus arquitecturas y modos de funcionamiento, sin embargo, todos ellos son basados en el aprendizaje a partir de ejemplos, lo que les permite modificar sus parámetros. Los métodos constructivos tienen por ventaja importante sobre las arquitecturas fijas basadas en *Backpropagation* el hecho de no tener que definir la arquitectura de la red, pues adaptan el tamaño de ésta a los datos de los que se dispone. Por otra parte, las unidades ocultas binarias permiten efectuar correctamente tareas de clasificación, ya que las fronteras entre clases son codificadas de manera natural a través de las representaciones internas.

Monoplan y *NetLines* son dos ejemplos de las arquitecturas dentro del paradigma de los algoritmos constructivos. Estos nuevos algoritmos aprenden mientras crecen, modificando tanto la arquitectura como las sinápsis a través de la regla de aprendizaje *Minimerror*, mostrando un gran desempeño en tareas de clasificación al utilizar unidades ocultas discretas. Este tipo de algoritmos se perfilan como uno de los instrumentos más poderosos dentro del aprendizaje automático. Quedan sin embargo varias cuestiones abiertas a explorar. Se sabe, por ejemplo que existen problemas de clasificación (donde los datos presentan formas convexas complejas) para los cuales dos capas ocultas son mejores que una; de esta forma, una extensión natural a este trabajo podría ser la incorporación de una segunda capa oculta que permita aumentar la capacidad de generalización de la red, y disminuir al mismo tiempo el número de unidades ocultas. Otra vía la constituyen los perceptrones con separaciones no lineales, como separaciones esféricas. Actualmente se trabaja en la adaptación eficiente de *Minimerror* para la búsqueda de separaciones esféricas, y la incorporación de estos perceptrones en redes híbridas (lineales y esféricas) para resolver una problemas aún más complejos. La medida de la capacidad teórica de aprendizaje y generalización de estas nuevas redes híbridas queda como una cuestión aun abierta. Por supuesto, los resultados obtenidos al trabajar con los métodos conexionistas han permitido profundizar en la parte fundamental de la investigación en RN, en la comprensión de algunos de los mecanismos del intelecto, la cognición, la memoria y su impacto en la creación de artefactos inteligentes, que representa -por si solo- un tema por demás emocionante.

REFERENCIAS

- [1] An evolutive architecture coupled with optimal perceptron learning for classification. J. M. Torres Moreno, P. Peretto, Mirta B. Gordon. Proceedings ESANN'95 - European Symposium on Artificial Neural Networks (Bruxelles, 1995) Michel Verleysen ed., pp. 365-370.
- [2] Efficient adaptive learning for classification tasks with binary units. Juan Manuel Torres Moreno, Mirta B. Gordon. Neural Computation 10(4):1007-1030, 1998.
- [3] Robustness against single event upsets of digital implementations of neural networks. A. Assoum, J. M. Torres Moreno, N.E. Radi, R. Velazco, M.B. Gordon, R. Ecoffet. Proceedings de ICANN'95 - International Conference on Artificial Neural Networks, Maison de la Chimie - Paris, Octobre 9-13, 1995. Session 9. Editors : Fogelman-Soulié and Gallinari EC2.
- [4] Classification par apprentissage : une étude comparative. J. Manuel Torres Moreno, Mirta B. Gordon. Proceedings des Journées Acquisition, Apprentissage. Sète, 8-10 mai 1996, pp.338-341.
- [5] S.E. Fahlman and C. Lebière. The cascade-correlation learning architecture. In D.S. Touretzky, editor, Advances in Neural Information Processing Systems, volume~2, pages 524-532, San Mateo, 1990. (Denver 1989), Morgan Kaufmann.
- [6] Numerical simulations of an optimal algorithm for supervised learning. Arnaud Buhot, J-M. Torres-Moreno, Mirta B. Gordon. Proceedings ESANN'97- European Symposium on Artificial Neural Networks (Bruges, 1997) Michel Verleysen ed. (1997) 151-156
- [7] Finite size scaling of the bayesian perceptron. Arnaud Buhot, Juan Manuel Torres Moreno, Mirta B. Gordon. Physical Review E, 55:7434-7440, 1997.
- [8] Characterization of the Sonar Signals Benchmark. Juan-Manuel Torres Moreno et Mirta B. Gordon. Neural Processing Letters 1:1-4, 1998.
- [9] Apprentissage par des réseaux de neurones: étude de nouveaux algorithmes constructifs. Ph. D. These. INPG Grenoble, 1997. (www.lania.mx/~torres/Publicaciones/these-resume-f.html)
- [10] Comparative study of three connexionist models on a classification problem. R. Baron, H. Paugam-Moisy, M. B. Gordon, J-M. Torres Moreno. Ecole Normale Supérieure de Lyon, Research Report LIP 96-07 (<ftp://ftp.ens-lyon.fr/pub/LIP/Rapports/RR/RR96/RR96-07.ps.Z>)
- [11] M.B. Gordon, P. Peretto, and D. Berchier. Learning algorithms for perceptrons from statistical physics. Journal of Physics I France, 3:377-387, 1993.
- [12] R.S. Michalski, J.G. Carbonell and T.M. Mitchell (Eds.) , Machine Learning: An artificial Intelligende Approach 2 Morgan Kauffmann, Los Altos, CA, 1986.
- [13] Cohen, P.R. and Feigenbaum E..A. (Eds.), The Handbook of Artificial Intelligence 3, (Kaufmann, Los Altos, CA, 1982)
- [14] R.P. Gorman and T.J. Sejnowski. Analysis of hidden units in a layered network trained to classify sonar targets. Neural Networks, 1:75-89, 1988.