

DESARROLLO Y CONSTRUCCIÓN DEL CONOCIMIENTO: UN ENFOQUE CONEXIONISTA

Constantino Malagón Luque

25 de septiembre de 2002

Resumen

Este trabajo es una introducción a las redes neuronales, presentando todos sus modos de aprendizaje, su base como emulador del modelo biológico y su campo de aplicación. Además se estudian modelos como el perceptrón simple, el perceptrón multicapa, la red ADALINE y modelos de redes no supervisadas como los mapas autoorganizados. Por último se describen dos modelos basados en redes neuronales que simulan algunos resultados de experimentos con pacientes esquizofrénicos y una simulación de un problema enmarcado en la psicología del desarrollo, destacable por su arquitectura y la modelización que realiza.

1 Introducción

Las redes neuronales son básicamente modelos matemáticos de procesamiento de información. Están inspiradas en el procesamiento de la información que tiene lugar en el cerebro como alternativa a un modelo que podríamos llamar clásico de tratamiento de la información basado en el manejo de símbolos (enfoque simbólico) y en una arquitectura de tipo Von Newman, y donde la arquitectura del hardware (en este caso las neuronas) cumplen un papel fundamental.

En su génesis constituyeron un intento de reproducir o imitar las capacidades de procesamiento de información de un sistema biológico, como puede ser el sistema nervioso central humano, o de cualquier organismo biológico en general. Además lo que se pretende es simular los procesos cognitivos básicos dentro del marco de la Ciencia Cognitiva, y responder preguntas del tipo: ¿Qué hace que una persona sea más inteligente que un gato? ¿Por qué una máquina no puede ser tan inteligente como una persona? Preguntas claves de la IA que se respondían como que era un problema de software, es decir, si contásemos con los algoritmos y programas adecuados podríamos llegar a alcanzar ese objetivo. El conexionismo, o la IA conexionista, basándose en estas redes neuronales van a poner el acento en el hardware como un aspecto fundamental en este problema.

Vamos a recordar la división de los niveles de conocimiento para tener bien presente en donde nos movemos.

1.1 Representación del conocimiento

La adquisición del conocimiento se puede en principio realizar de dos maneras:

1. Extrayendo los conocimientos de un experto humano para organizarlos y modelizarlos conceptualmente, en un proceso llamado Educación de conocimientos. Esto constituye el núcleo del desarrollo de los sistemas expertos, en lo que se ha dado en llamar IA clásica o simbólica (porque supone que un sistema cognitivo cualquiera va a tratar la información como símbolos que puede manejar) o también llamado **soft computing** porque sólo se hace referencia al software del sistema y no al hardware para representar ese conocimiento.
2. Por medio de un proceso de generalización y aprendizaje a partir de unos patrones de entrada iniciales. Esto forma parte de la adquisición de conocimientos en las redes neuronales, o IA conexionista, o más recientemente la neurocomputación. Se le introducen datos a un sistema de cálculo matemático llamado red neuronal artificial, y se hace que ésta aprenda para construir ella misma su base de conocimientos mediante un proceso de aprendizaje. Esta visión en la que se intenta emular el comportamiento de las neuronas biológicas se conoce como **conexionismo** o también como **hard computing** porque sólo piensa que no tanto el software del sistema como el hardware es necesario e influyente para la representación de ese conocimiento.

La representación del conocimiento, es decir, representar el conocimiento adquirido de forma que sea procesable por una máquina. En el caso de sistemas expertos esto se realiza mediante lenguajes formales, como el LISP o el PROLOG, u otros medios que comentaré más adelante.

La generación de inferencias o el proceso de razonamiento a partir de esos datos adquiridos.

2 El nivel de conocimiento de Allen Newell

Newell introdujo en 1981 una clasificación de los distintos niveles de computación muy explicativa del proceso del conocimiento. El objetivo de Newell era primero razonar acerca de la naturaleza del conocimiento, y proponer luego la existencia de un nivel específico del conocimiento cuyas entidades básicas son las creencias, objetivos, planes e intenciones y el principio de racionalidad que conecta causalmente las intenciones con las acciones.

En el primer punto Newell distingue entre el conocimiento y su representación, dotándoles de entidades diferenciadas. Y así distingue tres niveles, que empezando por el superior son:

1. **Nivel de conocimiento.** El **medio** o los objetos que maneja es el conocimiento, y éste es el nuevo nivel introducido por Newell. Los **componentes** o procesos primitivos básicos a partir de los cuales se pueden construir todos los demás son las metas, creencias, acciones, intenciones, procedimientos de inferencia y racionalidad, y los **operadores** son los que usa el lenguaje natural.

El nivel de conocimiento posee un carácter abstracto, genérico e independiente de los lenguajes usados para representar el conocimiento (lógica, reglas, marcos o redes semánticas) y para usarlo (inducción y deducción).

2. **Nivel simbólico,** cuyo medio son los símbolos con los que representaremos el conocimiento (y ésta es la hipótesis fundamental de IA clásica, todavía no demostrada en los casos reales como pueden ser los humanos y animales). Ésto lo hará un programador (**sistema** por medio de un lenguaje de alto nivel, como puede ser PROLOG.
3. **Nivel de implementación o físico,** cuyo medio son los 0 y 1, los componentes son los registros, ALUs y los distintos componentes del sistema físico, y el sistema será el procesador, memoria,...

Las transiciones entre los niveles de conocimiento y simbólico se pueden describir así:

- El paso del nivel de conocimiento al nivel simbólico lo llamaremos **reducción** o representación.
- El paso del nivel simbólico al nivel de conocimiento lo llamaremos **recuperación** o interpretación.

Las transiciones entre todos los niveles dependen del enfoque (simbólico o conexionista) que consideremos, pues en el enfoque conexionista la transición entre los niveles de conocimiento y de implementación se realiza directamente, sin pasar por el nivel simbólico, pues la representación se encuentra distribuida entre cada uno de los nodos de la red neuronal, y se realiza mediante el entrenamiento de la red. La transición entre los niveles simbólico y de implementación o físico en el enfoque simbólico lo realiza el compilador.

3 Fundamentos biológicos de las redes neuronales

El cerebro es un órgano de menos de un Kg y medio de peso y con más de cien mil millones de células, cada una de las cuales establece como media un millar de conexiones con sus vecinas, a través de múltiples prolongaciones, de un diámetro inferior a una diezmilésima de milímetro y que pueden llegar a medir más de un metro de longitud.

Y todo este complejo entramado da como resultado la capacidad de un ser biológico como puede ser un humano de razonar, sentir, aprender, escapar de una

casa en llamas o llegado el caso entrar en ella para salvar a otra persona, emocionarse, comunicarse con sus semejantes o quizá uno de los rasgos, junto con el del lenguaje, más distintivos del ser humano: la capacidad de ser conscientes de uno mismo y de su propia muerte.

Así, en una neurona podemos distinguir un cuerpo celular o soma, que actúa como centro metabólico de la célula, y dos tipos de prolongaciones, unas generalmente más largas y finas llamadas axón y otras más cortas y numerosas, en forma de ramificaciones llamadas dendritas.

Además de las neuronas, y mucho más numerosas que éstas, existen otro tipo de células nerviosas llamadas células de glía. Su función no es la de transmitir impulsos nerviosos, sino la de envolver a las neuronas proporcionándoles aislamiento eléctrico, servir como soporte físico que permita el crecimiento de éstas, la captación de neurotransmisores y actuar como barrera para protegerlas de los elementos tóxicos de la sangre.

Ramón y Cajal formuló dos hipótesis fundamentales para el desarrollo de lo que luego se conoció con Neurociencia. La primera era que existían lo que él llamaba “ondas nerviosas” que viajaban desde las dendritas al cuerpo neuronal, y de éste al axón, y la otra era que las neuronas se comunicaban entre sí por contigüidad y no por continuidad a través de unas conexiones especializadas llamadas sinapsis.

Fue a comienzos del siglo XX cuando se pudo registrar la entrada de señales eléctricas que viajaban a través del axón a gran velocidad, y se supo que éstos impulsos eléctricos que se llamaron potenciales de acción o impulsos nerviosos los usaba el sistema nervioso para la transmisión de información. Estos potenciales pueden repetirse dentro de una misma célula, llegando a frecuencias de 1000/segundo, y ésta información es función de la intensidad de la señal eléctrica y de la frecuencia de disparo (por ejemplo se comprobó que a mayor intensidad de un estímulo se correspondía una mayor frecuencia de impulsos nerviosos dentro de la neurona registrada).

4 Mecanismo intracelular para la transmisión de la señal

Las células nerviosas están recubiertas por una membrana que tiene unas proteínas llamadas canales iónicos. Estas proteínas actúan a modo de poros para ciertos iones en función de su tamaño. Estos iones pueden ser iones de Sodio (Na^+), Calcio (Ca^{2+}), Potasio (K^+) o Cloro (Cl^-). En condiciones de reposo la célula no se encuentra en un estado eléctricamente neutro, debido a la distribución asimétrica de éstos iones, existiendo por tanto una diferencia de potencial entre el interior y el exterior de la célula (es decir, a cada lado de la membrana) de unos 60-90 mV (siendo el interior electronegativo con respecto al exterior). Como consecuencia de un estímulo que llegue a la célula pueden abrirse éstos canales iónicos dejando entrar o salir aquellos iones a los que sean permeables dichos canales, lo que lleva a la desaparición de la diferencia de potencial por el

equilibrio de cargas en un instante dado, es decir, se produce la despolarización de la célula. Este cambio de potencial es lo que llamamos potencial de acción o impluso nervioso.

5 Mecanismo de transmisión de la señal entre diferentes neuronas

Desde finales del siglo XX se suponía que la comunicación entre neuronas se realizaba a través de contactos especializados en las diferentes terminaciones de las neuronas llamadas sinapsis. Pero no se sabía si la naturaleza de la señal transmitida era eléctrica o química. La hipótesis de que la despolarización de la terminación presináptica liberaba un neurotransmisor que se difundía hasta la membrana de la célula postsináptica, produciendo a su vez en ésta una despolarización y como consecuencia un potencial de acción fue formulada años más tarde por Bernard Katz.

Las terminaciones de las células nerviosas contienen pequeñas vesículas rellenas de neurotransmisor, denominadas vesículas sinápticas. El proceso de liberación de neurotransmisor depende de los iones de calcio presente en el interior de la terminación. Así, cuando se produce la llegada a ésta del potencial de acción desencadenado en el interior de la célula se abren en la membrana de la célula presináptica canales iónicos permeables al Ca^{2+} , dejando pasar a éstos iones al interior de la célula y produciendo la liberación del neurotransmisor que se produce a consecuencia de la presencia de éstos iones (ésto no es exactamente cierto puesto que en estado de reposo se liberan pequeñas cantidades de neurotransmisor debido a la presencia ya de iones de calcio, y lo que se hace después al aumentar la concentración de calcio es acelerar esa liberación). Cada vesícula contiene un número parecido de moléculas de neurotransmisor, pero la cantidad de éste liberada varía en función de muchos factores.

Al llegar a la membrana de la célula postsináptica se abren a su vez unos canales por unión con el neurotransmisor, lo que hace posible que pasen al interior de la célula iones, fundamentalmente de sodio, que despolarizan a la célula. En otros casos el efecto es inhibitorio, a través de la activación de canales permeables al Cl^- que producen generalmente una hiperpolarización de la neurona postsináptica.

6 Procesamiento distribuido paralelo

Partimos del hecho de considerar el cerebro como un dispositivo de computación que trabaja en paralelo. En principio hay varias evidencias de que ésto puede ser así. Por una parte las neuronas trabajan a una velocidad mucho más pequeña que los ordenadores actuales, pero en cambio son capaces de realizar tareas de una gran complejidad computacional como por ejemplo comprender una oración. Además se ha demostrado la poca influencia que tienen las neuronas individuales y sí un conjunto de ellas de forma distribuida, por ejemplo

desde el punto de vista de las lesiones que afectan a áreas del cerebro. Otra evidencia es el aparente carácter paralelo de funciones cognitivas como la influencia de la sintáxis y la semántica en la comprensión de textos: si por ejemplo alguien me dice "el otro día fui a Madrid a ver un partido de fútbol" soy capaz de comprenderlo porque conozco las reglas sintácticas del castellano. Pero a veces se producen ambigüedades como por ejemplo: "Los Pérez de Llanes vieron los Picos de Europa mientras iban volando a Inglaterra", en donde no sé, ateniéndome sólo a la sintáxis si los que vuelan son los Pérez de Llanes o los Picos de Europa, o por ejemplo "vi las ovejas pastando en el campo", donde no sé si las que pastan son las ovejas o el que me ha dicho la frase. Pero el caso es que todos entendemos sin ambigüedades esa frase, y parece claro que es por una clara influencia de la semántica sobre la sintáxis, que actúa antes o a la vez que ésta.

7 Conceptos básicos de redes neuronales artificiales

Se denomina unidad de procesamiento o neurona a un dispositivo simple de cálculo que a partir de un vector de entrada x_j procedente del exterior o de otras neuronas proporciona una única respuesta o salida y_i . Cada neurona se encuentra conectada con otras por medio de conexiones (sinapsis) a las que se les asocia un número real llamado peso de la conexión, formando lo que se llama una red neuronal artificial (en adelante ANN - *Artificial Neural Network*).

Un modelo típico de ANN consiste en varias capas de unidades de procesamiento que simulan neuronas. Cada unidad (neurona) añade información de unidades de la capa anterior, ejecuta una simple operación como por ejemplo decidir si está por encima de un valor umbral, y pasa su resultado a la capa siguiente. El patrón de actividad en la primera capa representa el estímulo asociado al modelo. Este patrón es gradualmente transformado para producir el patrón en la capa final, la respuesta del modelo. La influencia de una unidad en una capa sobre otra unidad en la otra capa depende de la fuerza de la conexión entre ellas (o peso de la conexión). El aprendizaje para producir la correcta salida en respuesta a un estímulo se consigue variando esta fuerza entre unidades.

Así pues, una red neuronal puede concebirse como un dispositivo computacional que dada una entrada calcula el valor de una función a través de un proceso de aprendizaje y generalización a los patrones previos que le hayan sido presentados. Esa función $\mathfrak{R}_n \rightarrow \mathfrak{R}_m$ representa una correspondencia entre un patrón de entrada n-dimensional real (x_1, x_2, \dots, x_n) y un patrón de salida m-dimensional real (y_1, y_2, \dots, y_m) .

8 Componentes de una red neuronal básica

Vamos a ver en detalle los componentes básicos de una red neuronal. Se suelen encontrar los siguientes elementos:

1. Un conjunto de **unidades de procesamiento** o neuronas artificiales. Representaremos a cada una de ellas como x_j .
2. Un **estado de activación** a_j que caracteriza a cada una de las unidades de procesamiento x_j en un tiempo t_1 .
3. Los **pesos sinápticos** w_{ij} , que representa que representa la fuerza de la interacción entre la neurona presináptica j y la postsináptica i (notar que los índices están cambiados en el orden que sería natural por razones históricas. Si w_{ij} es pequeña la unidad j tendrá poca influencia sobre la unidad i . Este peso puede ser positivo o negativo.
4. La **regla de propagación** $\sigma(w_{ij}, x_j(t))$, a veces denotada como $netinput_i$ para propagar patrones de actividad a través de las unidades y que proporciona el valor del potencial postsináptico $h_i(t) = \sigma(w_{ij}, x_j(t))$ de la neurona i en función de sus pesos y entradas. Una función típica que se suele tomar en muchos modelos es la suma ponderada de las entradas con los pesos sinápticos $h_i(t) = \sum_j w_{ij}x_j$ o en notación vectorial como el producto escalar $h_i(t) = \mathbf{w}_j^T \cdot \mathbf{x}$. También pueden considerarse como en el caso de los mapas de Kohonen como función de propagación la distancia euclídea $h_i^2 = \sum_j (x_j - w_{ij})^2$.
5. Una **regla de aprendizaje** por la cual modificamos los pesos de las distintas unidades.
6. La **función de activación** o **actividad** $a_i(t)$, que proporciona el estado de la neurona i como función de su estado anterior $a_{i-1}(t)$ y de su potencial postsináptico actual. Si a_j es positivo aumentará las entradas de todas las unidades a las que esté conectado con un peso positivo, y disminuirá si la conexión tiene un peso negativo. Como función de activación típica se suele tener la función escalón $y = \text{signo}(x)$, que vale +1 para $x \in \{0, 1\}$ y -1 para $x \in \{-1, 0\}$, aunque también se puede considerar en el dominio $(-\infty, 0] \cup [0, \infty)$ o la función de Heaviside $y = H(X)$ definida en el dominio $\{0, 1\}$ para el caso de neuronas con entradas digitales binarias. Otras funciones pueden ser la función identidad $y = x$, la función sigmoideal $y = \frac{1}{1+e^{-x}}$ o $y = \text{tgh}(x)$, o una función gaussiana del tipo $y = A \cdot e^{-Bx^2}$. Según la función escogida se tendrá un mayor o menor determinismo en el comportamiento del modelo.
7. La **función de salida** $F_i(a_i(t))$, que proporciona la salida actual $y_i(t) = F_i(a_i(t))$ de la neurona i en función de su estado de activación. Luego se tiene que:

$$y_i(t) = F_i(f_i[a_i(t-1), \sigma_i(w_{ij}, x_{ij}(t))])$$

La función de salida puede ser la identidad $y = f(x)$, en cuyo caso $y_i(t) = F_i(a_i(t)) = a_i(t)$, o de tipo escalón lo que supone que la neurona no se dispara hasta que la activación supere un cierto valor umbral

Se denomina unidad de procesamiento o neurona a un dispositivo simple de cálculo que a partir de un vector de entrada x_j procedente del exterior o de otras neuronas proporciona una única respuesta o salida y_i . Estas unidades pueden ser de entrada (si reciben las entradas de fuentes externas al sistema que se estudia, a la manera en que lo hacen los nervios eferentes), de salida (que envían señales hacia fuera del sistema, a la manera de los nervios aferentes) u ocultas (las entradas y salidas están dentro del sistema. Cada neurona se encuentra conectada con otras por medio de conexiones (sinapsis) a las que se les asocia un número real llamado peso de la conexión, formando lo que se llama una red neuronal artificial (en adelante ANN - *Artificial Neural Network*). **Estos pesos representan el nivel de conocimiento de la red**, estando por tanto distribuido entre cada uno de los nodos de la red. Un modelo típico de ANN consiste en varias capas de unidades de procesamiento que simulan neuronas. Cada unidad (neurona) añade información de unidades de la capa anterior, ejecuta una simple operación como por ejemplo decidir si está por encima de un valor umbral, y pasa su resultado a la capa siguiente. El patrón de actividad en la primera capa representa el estímulo asociado al modelo. Este patrón es gradualmente transformado para producir el patrón en la capa final, la respuesta del modelo. La influencia de una unidad en una capa sobre otra unidad en la otra capa depende de la fuerza de la conexión entre ellas (o peso de la conexión). El aprendizaje para producir la correcta salida en respuesta a un estímulo se consigue variando esta fuerza entre unidades.

Así pues, una red neuronal puede concebirse como un dispositivo computacional que dada una entrada calcula el valor de una función para esa entrada a través de un proceso de aprendizaje y generalización de los patrones previos que le hayan sido presentados. Esa función $\mathfrak{R}_n \rightarrow \mathfrak{R}_m$ representa una correspondencia entre un patrón de entrada n -dimensional real (x_1, x_2, \dots, x_n) y un patrón de salida m -dimensional real (y_1, y_2, \dots, y_m) .

En resumen, tenemos unidades de cómputo individuales que se agrupan en estratos o capas, estando mutuamente inhibidos los elementos de una misma capa, y cada uno de ellos se encuentra conectado a todos y cada uno de los elementos de la capa siguiente, excitando o inhibiendo en función del peso que tenga. Es decir, a una unidad x_j le llegarán entradas de todas las unidades anteriores x_i , en función de una regla de propagación *netinput_j*, que viene dada por la suma ponderada de los valores de las entradas y los pesos sinápticos asociados a cada una de las conexiones de las neuronas x_i con la neurona x_j . Dicha neurona aglutinará y procesará todas esas entradas pasándola por una función de activación, típicamente una función sigmoideal de la forma $y = \frac{1}{1+e^{-x}}$. Eso modificará el estado de activación a_j de la neurona x_j , y ésta dará una salida $y_i(t) = F_i(a_i(t))$, donde la $F_i(a_i(t))$ puede ser la identidad o una función de tipo escalón que devuelva un valor sólo cuando se supera un cierto umbral de disparo. Es decir, si la acción de todas esas entradas supera un cierto valor límite la neurona se activará y excitará o inhibirá a cada una de las de la siguiente capa.

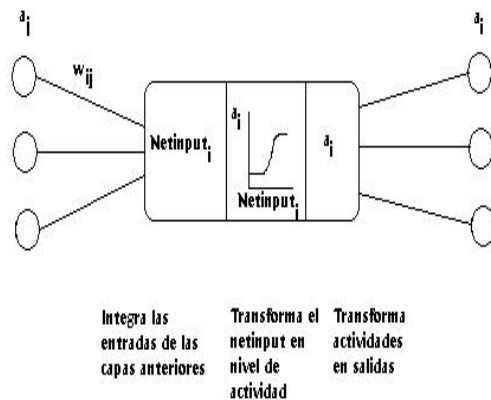


Figura 1: Las redes neuronales son básicamente procesadores matemáticos

9 El asociador lineal de patrones.

Vamos a ver la arquitectura de un tipo especial de red, un asociador de patrones, y el funcionamiento de una regla de aprendizaje particular, en la cual se basan muchos de los modelos de redes neuronales artificiales: la regla de Hebb. Lo que vamos a hacer es presentar a la red durante el entrenamiento un par de patrones. Si el aprendizaje se realiza con éxito entonces la red presentará el patrón de salida cuando se le presente el otro de entrada. Así, podrá responder después ante patrones nuevos, generalizando a partir de su experiencia con patrones similares. Al fin y al cabo de lo que se trata es de agrupar (o correlacionar) objetos que recibimos por los sentidos. Por ejemplo, un estímulo podría ser el sabor del chocolate y otro su apariencia. Inicialmente no hay conexión entre ellos para por ejemplo un niño. Una vez ha asociado los patrones, la apariencia del chocolate para un niño le evocará el sabor de éste. Y si el sabor le produce salivación, también se lo producirá la apariencia o la vista de un trozo de chocolate (como explicó Pavlov). Otros ejemplos pueden ser la lectura (nosotros aprendemos a leer memorizando que la l con la a se lee la), que María tiene el pelo rubio,...

Se le presentan durante el aprendizaje dos patrones simultáneamente, uno como

de entrada y otro como la salida deseada. Por ejemplo, el patrón 1 representa el sabor del chocolate, y proviene quizás de otra red de neuronas. El patrón 2 representaría la apariencia del chocolate. El aprendizaje consiste en asociar esos patrones modificando los pesos sinápticos.

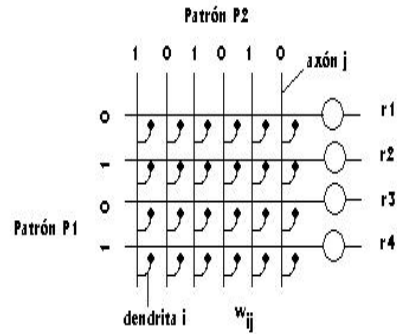


Figura 2: Asociador lineal

9.1 La regla de Hebb

Donald Hebb, The organization of the behavior, 1949

Se formula de la siguiente manera: Si la neurona de entrada x_j está activa cuando la neurona de salida x_i lo está también, entonces el peso sináptico w_{ij} aumentará. Matemáticamente se formularía así:

$$\Delta w_{ij} = \epsilon a_i a_j$$

Es decir, la variación en los pesos es directamente proporcional al producto de las actividades, siendo el factor de proporcionalidad ϵ , que especifica la magnitud del cambio en las sinapsis debidas a a_i y a a_j . Vamos a ver un ejemplo.

Consideremos neuronas binarias y $\epsilon = 1$. Sea P_2 el patrón que representa la apariencia del chocolate, y P_1 el patrón que representa el sabor del chocolate. Lo que vamos a hacer es enseñar a la red a asociar los dos patrones, de forma que a la presentación de una onza de chocolate (la apariencia) le evoque el sabor del chocolate (a la manera en que lo aprenden los niños). Para ello calculo la matriz de pesos

$$\delta w_{ij} = \epsilon a_i a_j$$

siendo $P_1 = (1, 1, 0, 0)$ y $P_2 = (1, 0, 1, 0, 1, 0)$ y esa será la matriz de asociación de P_2 y P_1 . Después, si le presento P_2 , y calculo la entrada total como $netinput_j = \sigma_i(w_{ij}, a_j)$, obtengo P_1 sin más que considerar una función de activación de tipo escalón con un umbral $\theta = 2$.

Si hacemos lo mismo pero ahora con el patrón de la apariencia del melocotón y el patrón del sabor, ¿interferirán en los pesos anteriores (es decir, olvidará lo aprendido del chocolate)? Es fácil ver que no ocurre eso, calculando la matriz de pesos para el melocotón, sumándola a la del chocolate y pasándole después el patrón de la apariencia del melocotón.

9.2 Características del asociador de patrones

El asociador de patrones presenta dos características fundamentales:

- La **tolerancia al error**, es decir, si una o varias neuronas "mueren", el resultado sigue siendo el mismo.
- La **generalización**, es decir, durante el proceso de recuperación o recuerdo, los asociador de patrones generalizan, de forma que si presento un patrón similar al que usé en el aprendizaje me dará una respuesta similar a la aprendida. En definitiva esto es una regla adaptativa (cuando yo aprendo algo debo saber reaccionar ante situaciones similares).

En nuestro ejemplo, si presento un patrón P_3 que represente por ejemplo la apariencia del albaricque, la red reaccionará de acuerdo a lo que sabe clasificándolo como un elemento de la clase de los melocotones. En realidad lo trata como una versión con ruido de la que conoce (otra de las características de los asociadores lineales).

10 Autoasociador de patrones

Es un tipo de red que reproduce a la salida el mismo patrón que se le presenta a las unidades de entrada. Las aplicaciones prácticas más inmediatas es como filtro de ruido en aparatos de telecomunicaciones o en aparatos para realizar ecocardiogramas, por ejemplo.

El diseño de la red se muestra en la figura [3].

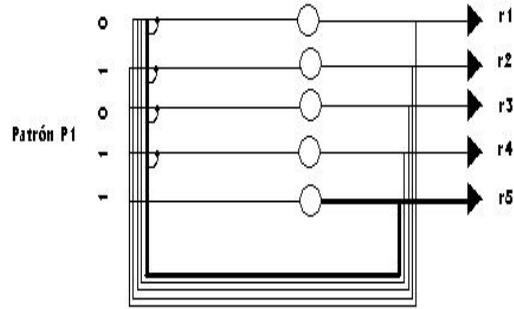


Figura 3: Autoasociador lineal. Se ha destacado la última unidad por claridad de exposición. Como se ve realimenta a todas excepto a ella misma.

En este caso el $Netinput_i$ consiste en el patrón de entrada externo, que denotaremos $Extinput_i$, procedente de la codificación de la señal de entrada o posiblemente de otras capas anteriores. La entrada interna, que llamaremos $Intinput_i$ generada por retroalimentación (*feedback*) desde otras unidades del autoasociador menos de sí misma. Por lo tanto tenemos que:

$$Netinput_i = Extinput_i + Intinput_i = P_1 + \sum_j a_j w_{ij}, \forall i \neq j$$

En estas redes hay un posible factor de riesgo, y es que debido a la realimentación las actividades crezcan mucho. Para evitar esto se usa una función de activación no lineal, como puede ser la función sigmoide.

Para la regla de aprendizaje usaremos la llamada **regla delta**, que se basa en calcular la diferencia entre las entradas externa e interna (es decir, entre la salida deseada y la salida producida), y se cambiarán los pesos de forma que esa diferencia se haga mínima. Definimos

$$\delta_i = Extinput_i - Intinput_i$$

, que representa el error en la unidad i . Por lo tanto el cambio en los pesos será

$$\Delta w_{ij} = \delta_i a_j$$

Lo que ocurrirá en los pasos siguientes es que las unidades tenderán a acercarse o alejarse del valor de su actividad al valor inicial de entrada en las siguientes iteraciones. O dicho de otra forma, el patrón interno se irá acercando al externo, y el aprendizaje cesará cuando sean iguales (o cuando su diferencia sea menor que un número ϵ suficientemente pequeño, que representará el error máximo admitido).

11 El perceptrón simple

Este modelo fue desarrollado por Frank Rosenblatt en 1958 [1]. Su principal característica es la capacidad que posee para el reconocimiento de patrones, de forma parecida a como lo hacen los sistemas sensoriales de los animales. Además Rosenblatt introdujo la idea de que la memoria era simplemente un cambio en la relación entre una entrada y una salida, y que podía cambiar con el uso. Lo que hizo realmente fue introducir la idea del aprendizaje y sobre todo la idea crucial de que las conexiones físicas entre las células nerviosas no son inmutables, sino que van variando con el paso del tiempo y da una medida del aprendizaje de la persona.

El Perceptrón fue presentado para responder a las últimas dos de estas tres preguntas fundamentales acerca del cerebro [1]:

1. ¿Cómo se detecta la información proveniente del mundo físico por un sistema biológico?
2. ¿De qué manera esta información es almacenada o recordada?
3. ¿Cómo influye la información almacenada en el reconocimiento de nuevos patrones y en el comportamiento?

El perceptrón básicamente es un modelo compuesto por dos capas de neuronas, una de entrada y una neurona simple de salida. A cada conexión entre la neurona x_j de la capa de entrada con neuronas x_i de la capa de salida se le asocia un peso w_{ij} . Cada una de estas entradas recibe un valor binario positivo del conjunto $\{0,1\}$ y posiblemente algunos valores negativos que harán una función inhibitoria. Si la suma de estas entradas ponderadas supera un cierto valor umbral se producirá una salida de 1. Y se trata de un modelo de aprendizaje supervisado porque primero se le presenta un patrón de entrada y se analiza su salida. Se compara con la deseada y se vuelven a ajustar los pesos, hasta que se produce la salida deseada. Por ello también se le suele llamar **algoritmo por corrección de errores**. En general, si suponemos que la capa de entrada tiene n neuronas y la de salida m , podremos representarlo así:

$$y_i(t) = f\left(\sum_{j=1}^n w_{ij}x_j - \theta_i\right), \forall 1 \leq i \leq m$$

Supondremos entradas discretas $\{0,1\}$ y una función de activación f de tipo escalón. Por lo tanto se tiene:

$$y_i = H\left(\sum_{j=1}^n w_{ij}x_j - \theta_i\right), \forall 1 \leq i \leq m$$

con H la función de Heaviside o escalón.

Vamos a ver cómo el Perceptrón puede ser usado como dispositivo para la clasificación de patrones (luego veremos que deben ser linealmente separables). Supongamos para ello un perceptrón de dos neuronas de entrada, x_1 y x_2 , que producen una salida y , que será, según :

$$y = H(w_1x_1 + w_2x_2 - \theta)$$

o bien

$$y = \begin{cases} 1, & \text{si } w_1x_1 + w_2x_2 > \theta \\ 0, & \text{si } w_1x_1 + w_2x_2 \leq \theta \end{cases}$$

La condición $w_1x_1 + w_2x_2 - \theta = 0 \Rightarrow x_2 = -\frac{w_1}{w_2}x_1 + \frac{\theta}{w_2}$ ecuación de una recta si consideramos un espacio bidimensional con x_1 y x_2 que divide el plano en dos regiones del espacio o semiplanos. Si trabajamos con n dimensiones tendremos un espacio n -dimensional separados en hiperplanos. Por ejemplo veamos cómo separa la función lógica booleana OR, cuya tabla de verdad es:

x_1	x_2	OR
1	1	1
1	0	1
0	1	1
0	0	0

Debo conseguir unos parámetros w_1, w_2 y θ que determinen una recta que separen los pares de valores de entrada (11), (10) y (01) que pertenecen a la clase 1 del par (00) que pertenece a la clase 0 [2], y es fácil ver que por ejemplo con $w_1=0.5$, $w_2 = 1,5$ y $\theta = -0,5$ se consigue. Pero no puedo hacerlo por ejemplo con la función XOR, que no es linealmente separable. Esto fue descubierto y publicado por Minsky y Papert en 1969, [3], y pese a que Rosenblatt ya lo había predicho en su trabajo y había apuntado la necesidad de utilizar varias capas para problemas que no eran linealmente separables, éste trabajo supuso un auténtico jarro de agua fría para los estudiosos en el campo de las redes neuronales y como consecuencia un profundo receso en su investigación. El problema es que aunque se suponía la solución teórica no se disponía de un algoritmo de aprendizaje para un perceptrón multicapa con neuronas ocultas, y no fue hasta mediados de los ochenta cuando el grupo PDP descubriera el algoritmo de retropropagación o *back-propagation*.

11.1 Regla de aprendizaje del perceptrón

Vamos a describir la regla de aprendizaje del perceptrón, llamada por corrección del error. Lo que vamos a hacer es ajustar los pesos de forma que el error

cometido (es decir, la diferencia entre la salida deseada y la obtenida) sea mínimo. La llamaremos técnica del descenso por el gradiente del error. Veremos que este error es cero cuando los patrones de entrada y de salida son linealmente separables en redes monocapa.

Sea un conjunto de p patrones de entrada x^μ , $\mu = 1 \dots p$ con sus salidas deseadas t^μ . Definamos

$$\Delta\theta = -\epsilon\delta_i$$

siendo $\delta_i = t_i - a_i$. Por lo tanto, la forma de expresar la regla del perceptrón es

$$\Delta w_{ij} = \epsilon\delta_i a_j$$

Nótese que si

$$a_i > t_i \rightarrow \delta_i > 0 \rightarrow \Delta\theta < 0 \text{ y } \Delta w_{ij} > 0$$

es decir, disminuyo el valor umbral y aumento los pesos, y que si

$$a_i < t_i \rightarrow \delta_i < 0 \rightarrow \Delta\theta > 0 \text{ y } \Delta w_{ij} < 0$$

es decir, aumento el valor umbral y disminuyo los pesos.

12 ADALINE

ADALINE es el acrónimo de ADaptative LINEar Element. Fue desarrollado por Bernard Widrow y Marcuian Hoff en 1960. Se desarrolló en un ámbito de problemas de computación, y no se hizo por lo tanto ninguna mención al cerebro. Es por ello que no se considera en él ningún umbral de disparo como en el perceptrón, aunque sí algo similar que se denomina *bias*. Así la ecuación de la ADALINE resulta

$$y_i(t) = \sum_{j=1}^n (w_{ij}x_j - \theta_i), 1 \leq i \leq m$$

Como puede comprobarse la salida es igual que en el perceptrón. La diferencia estriba en el algoritmo de aprendizaje, llamado regla de Widrow-hoff o regla LMS (Least Mean Squares, o mínimos cuadrados), en el que la modificación de los pesos es proporcional al error que la neurona comete, reduciendo así el error cuadrático medio que se comete. Podemos escribirla así:

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij} = w_{ij}(t) + (t_i^\mu - y_i^\mu)x_j^\mu$$

12.1 Método general de construcción de reglas de aprendizaje

Vamos a ver una forma general de obtener reglas de aprendizaje para arquitecturas concretas. El método consistirá en proponer una función error o coste que mida el rendimiento de la red, función que dependerá de los pesos sinápticos.

Lo que haremos será calcular el conjunto de pesos que hacen que la función sea mínima. El método de optimización más usado es el de **descenso por el gradiente**.

Se comienza definiendo una función error $E(\cdot)$ que proporcione el error actual E que comete la red neuronal, que será una función del conjunto de los pesos sinápticos W , $E = E(W)$, $E : \mathfrak{R}_n \rightarrow \mathfrak{R}_m$. Esta función representa una hipersuperficie con montañas y valles, y donde deberemos buscar aquellos valores de W que se encuentren en un mínimo local o global, es decir, en un valle. Para ello se hace lo siguiente:

Se parte en $t = 0$ de una cierta configuración $W(0)$, y se calcula el sentido de la máxima variación de la función $E(W)$ en $W(0)$, que vendrá dado por su gradiente en $W(0)$; es decir, estamos calculando el máximo global o local de $E(W)$. A continuación se modifican los parámetros W siguiendo el sentido contrario al indicado por el gradiente de la función error. De este modo se lleva a cabo un descenso por la hipersuperficie del error, aproximándose en una cierta cantidad a un mínimo local; el proceso se repite hasta alcanzarlo. Vamos a verlo en forma matemática:

$$W(t + 1) = W(t) - \epsilon \Delta E(W)$$

donde ϵ (que puede ser diferente para cada peso) indica el tamaño del paso tomado en cada iteración. Se puede demostrar [4] que esto es:

$$\delta(E(w_{ij})) = \sum_{ij} \frac{\partial E(w_{ij})}{\partial w_{ij}} \delta w_{ij} = -\epsilon \sum_{ij} \left(\frac{\partial E(w_{ij})}{\partial w_{ij}} \right)^2 \leq 0$$

13 El Perceptrón multicapa (MLP)

Fue presentado en 1986 por el grupo PDP, que lo formaban entre otros Rumelhart, Hinton y McClelland. La gran aportación del PDP fue la introducción del algoritmo de aprendizaje denominado backpropagation (BP) o por retropropagación de errores. La arquitectura del MLP consiste, si consideramos un modelo simplificado de tres capas, en una capa de entrada x_i^μ , una de salida z_k^μ con un valor de umbral de disparo θ'_k y entre medias una capa oculta y_j^μ con un umbral θ_j . Se dispone a su vez de una salida deseada (target) t_k^μ . La salida z_k será:

$$z_k = \sum_j w'_{kj} y_j - \theta'_k = \sum_j w'_{kj} f\left(\sum_i w'_{ji} x_i - \theta'_j\right) - \theta'_k, [3.1]$$

siendo f la función de activación para la capa de neuronas ocultas la función sigmoideal

$$f = \frac{1}{1 + e^{-x}} = \tanh(x)$$

13.1 Algoritmo de aprendizaje por retropropagación de errores

Supongamos un MLP de tres capas como el descrito anteriormente y un patrón de entrada x^μ , ($\mu = 1 \dots p$). Según [3.1] tendremos:

$$z_k^\mu = \sum_j w'_{kj} y_j^\mu - \theta'_k = \sum_j w'_{kj} f\left(\sum_i w'_{ji} x_i^\mu - \theta'_j\right) - \theta'_k, [3.2]$$

La función error que trataremos de minimizar es el error cuadrático medio

$$E(w_{ji}, \theta_j, w'_{kj}, \theta'_k) = \frac{1}{2} \sum_\mu \sum_k \{t_k^\mu - f(\sum_j w'_{kj} y_j^\mu - \theta'_k)\}^2, [3.3]$$

Para calcular el mínimo calcularemos su gradiente (habrá dos, uno respecto de los pesos de la capa de salida y otro respecto de la oculta) y veremos cuando es igual a 0:

$$\delta w'_{kj} = -\epsilon \nabla_{w'_{kj}} E = 0 \text{ y } \delta w_{ji} = -\epsilon \nabla_{w_{ji}} E = 0$$

Así, el procedimiento para entrenar una red mediante BP es como sigue:

- Establecer aleatoriamente los pesos y umbrales iniciales
- Para cada patrón μ del conjunto de aprendizaje
- Llevar a cabo una fase de ejecución para obtener la respuesta de la red ante el patrón μ –ésimo según [3.3]
- Calcular las señales de error asociadas Δ'_k y Δ_j según [3.4] y [3.5]
- Calcular el incremento parcial de los pesos y umbrales debidos a cada patrón μ (elemento de los sumatorios [3.4] y [3.5])
- Calcular el incremento total (para todos los patrones) actual de los pesos y umbrales
- Actualizar pesos y umbrales
- Calcular el error actual y volver al paso 2 si no es satisfactorio.

14 Redes competitivas

Hasta ahora hemos visto redes supervisadas, que se caracterizan porque conocíamos la salida que se debía producir para un conjunto de patrones y guiábamos a la red hasta conseguir la salida deseada reduciendo el error cometido. Por cierto este es el punto más discutible para muchos en la correspondencia entre el modelo psicológico y el modelo computacional conexionista. ¿Es muy real que alguien externo al sistema, haciendo las veces de instructor, lo guíe hasta que de con la solución deseada? ¿Conocemos siempre esa solución?

Ahora en las redes no supervisadas no conocemos la salida que debe producir la red, sino que será la red la que aprenda a clasificar o a categorizar patrones en conjuntos o categorías, mediante un método llamado **aprendizaje no supervisado**.

Vamos a ver un ejemplo de este aprendizaje en las redes competitivas. Supongamos una red con tres neuronas de entrada y dos de salida. Las dos neuronas de la capa de salida compiten entre sí hasta que una permanezca activa. Esta será la neurona ganadora. Este algoritmo de aprendizaje se llama *WTA* (the winner takes it all), es decir, la neurona ganadora (que va a ser aquella que al final presente mayor actividad) será la única que modifique sus pesos, y será la que irá aumentando su actividad. Las neuronas perdedoras la disminuirán. Y lo que van a hacer es básicamente agrupar patrones (*clustering* por características similares).

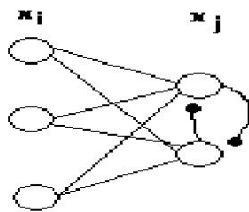


Figura 4: Red competitiva y algoritmo WTA

Vamos a dividir el proceso de aprendizaje en tres fases:

1. **Excitación:** Lo que ocurre en esta fase es la excitación de las neuronas, que como siempre será lo que hemos llamado el $Netinput_j = \sigma_i(w_{ij}, a_j)$.
2. **Competición:** Las actividades de las neuronas de la capa de salida son comparadas para determinar la ganadora, que será aquella que tenga mayor valor de actividad. Esta puede forzar a las demás a volverse inactivas.
3. **Ajuste de pesos:** El ajuste de pesos se realizará únicamente en las conexiones sinápticas de la ganadora. $\Delta w_{ij} = 0$ si la unidad i pierde $\Delta w_{ij} = a_j - w_{ij}$ si la unidad i es la ganadora. Esto sucederá hasta que $a_j = w_{ij}$. Por lo tanto si $w_{ij} \leq a_j$ se aumentan los pesos. Si la misma neurona de salida vence ante la presentación de otro patrón diferente la matriz de pesos se ajustará con otro patrón. Así que patrones similares harán que, después del entrenamiento, la misma neurona gane, y tendrá una matriz de pesos *promedio* para todos los patrones, descubriendo en qué se parecen o cuáles se parecen entre sí. Es decir, es normal que la neurona ganadora vuelva a ganar cuando se le presenten patrones similares.

15 Mapas autoorganizados SOM

Fueron desarrollados en 1982 por Teuvo Kohonen, y son un caso especial de redes competitivas, es decir, de redes no supervisadas. Se han usado para clasificación de patrones, extracción de rasgos, minería de datos y en el reconocimiento del habla.

La arquitectura de la red es de dos capas. La primera capa es la capa de entrada o *sensorial*, que consiste en m neuronas, una por cada variable de entrada. El procesamiento se realiza en la segunda capa, que consiste en una estructura bidimensional de $nx * ny$ neuronas. Cada neurona de entrada la denotaremos por un subíndice k ($1 \leq k \leq m$), y las $nx * ny$ neuronas de la segunda capa con un par de índices $i = (i, j)$, con ($1 \leq k \leq nx$) y ($1 \leq k \leq ny$) que determinan su localización espacial (a modo de coordenadas). Cada neurona k está conectada a todas las neuronas i .

Distinguiremos dos fases:

1. **Fase de ejecución.** En esta fase los pesos permanecen fijos. Primero cada neurona (ij) calcula la similitud entre el vector de entrada x y su propio vector de pesos sinápticos w_{ij} , según una medida de distancia $d(w_{ij}, x)$, que puede ser la distancia euclídea o cualquier otra, como la de Minkowsky. A continuación se declara ganadora la neurona $g = (g_1, g_2)$ cuyo vector de pesos w_g es más similar al de entrada, de acuerdo a esta distancia. Es decir:

$$d(w_g, x) = \min_{ij} d(w_{ij}, x)$$

2. **Fase de aprendizaje.** Tras la presentación y procesamiento del vector x , la neurona ganadora modifica sus pesos de manera que se parezcan un

poco más al patrón x . De este modo, ante el mismo patrón de entrada responderá la neurona vencedora con más intensidad. Al final, con varios patrones se producirá un agrupamiento de ellos.

16 Simulación conexionista de procesos cognitivos

Vamos a ver algunos ejemplos de simulaciones de procesos cognitivos mediante redes neuronales. La primera es una simulación de la tarea de Stroop en pacientes que sufren esquizofrenia. La segunda es un famoso problema que propuso Piaget en el marco de la psicología del desarrollo. Lo que se pretende con esta última simulación es presentar una original arquitectura de la red que sirve para modelar el problema. Los datos del experimento y su correspondencia con los datos experimentales reales no se expondrán aquí, pudiéndose consultar las referencias bibliográficas para un estudio más detallado de estos resultados.

16.1 Simulación conexionista de la esquizofrenia y la atención selectiva

Vamos a empezar a estudiar los casos prácticos de simulación mediante redes neuronales con el estudio del comportamiento de los pacientes que sufren esquizofrenia. Estudiaremos cómo se comportan en tareas que requieren de la atención selectiva, factor muy relacionado con la consciencia.

Nuestro sistema sensorial está continuamente recibiendo estímulos desde el exterior, y gran cantidad de ellos con información irrelevante que nuestro sistema atencional debe filtrar. Así mismo debe ser capaz de reconocer un estímulo con información importante, mediante un mecanismo que se conoce como atención selectiva.

Se sabe que los pacientes esquizofrénicos sufren trastornos en esta atención selectiva, y es este trastorno el que intentaremos simular con una red neuronal.

Un ejemplo de experimento para estudiar la atención selectiva se conoce como Stroop; en este experimento se le presentan a los sujetos una lista con palabras escritas en colores, y deben decir el color de la letra intentando ignorar la letra en sí. Lo que se observa es que los sujetos tardaban más en responder Green (verde) cuando la palabra coloreada era RED (rojo) que cuando era BED (cama) [13]. La explicación es la interferencia que se produce al percibir a la vez el color y el nombre del color al que alude la palabra escrita. En el caso de red y rojo interfería (para los angloparalantes) mientras que bed y rojo no. En resumen, podemos decir que poca interferencia en la palabra implica una buena atención selectiva y mucha interferencia una atención selectiva más pobre.

Es de estos experimentos a partir de los cuales se vio el déficit en la atención selectiva que poseen los esquizofrénicos. Los resultados experimentales se muestran en la figura [1]. Los dos primeros apartados de la gráfica muestran el tiempo de reacción cuando se le presentan al sujeto un sólo estímulo. El primer apartado muestra sólo la palabra que designa el color pero en negro, mientras que el

segundo muestra sólo el color, es decir, palabras escritas en un color pero que no nombran ese color. El tercer punto es el punto clave y muestra las palabras con la interferencia explicada anteriormente. Se puede ver que los sujetos normales muestran un retardo de 400 ms respecto al segundo punto, mientras que este retardo es de 700 ms en pacientes esquizofrénicos.

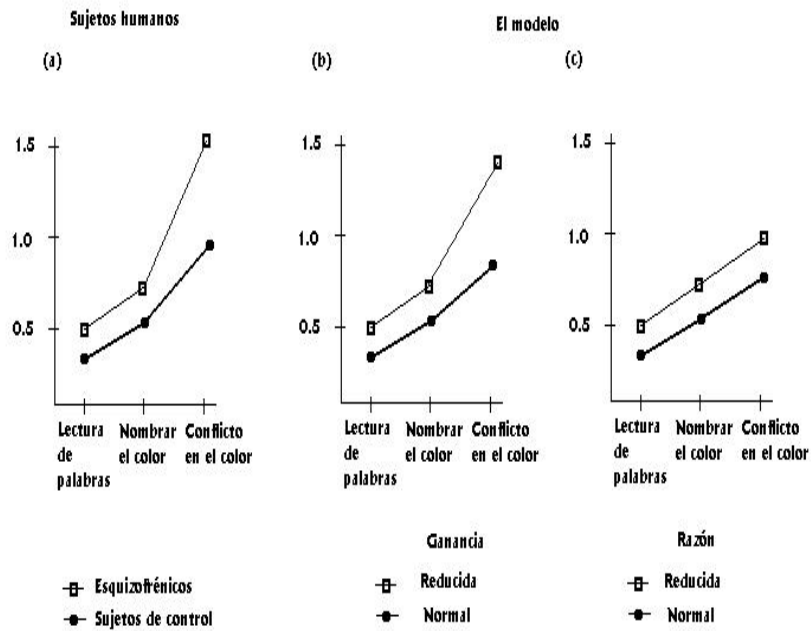


Figura 5: Los tiempos de reacción en tres condiciones del experimento de Stroop. La figura (a) muestra datos de pacientes esquizofrénicos y sujetos de control. Las figuras (b) y (c) muestran los resultados de la simulación. En el (b) los datos de los pacientes esquizofrénicos se simulan lesionando el módulo de atención de la red. En el (c) los datos de los pacientes esquizofrénicos se simulan lesionando todos los módulos de la red, no sólo el de atención.

El problema a la hora de interpretarlo surge porque los pacientes esquizofrénicos también muestran un retardo en los dos primeros puntos de 100 y 200 ms respectivamente. Así parece que el retardo no es achacable a un déficit en la atención selectiva, sino quizá a un déficit en todo el sistema atencional, aunque existe quien defiende la primera hipótesis.

La idea es decidir cuál de las dos hipótesis es más exacta mediante la simulación conexionista del experimento, y lesionando la parte de la red neuronal que haría el papel de sistema atencional [8].

16.1.1 Modelo conexionista de la esquizofrenia

Cohen y Servan-Shreiber usaron un modelo conexionista como el de la figura [1-r]. Se compone de 6 neuronas de entrada (2 para el color de la tinta, rojo o verde, dos para la palabra y otras dos para la tarea pedida, nombrar el color o leer la palabra. Se ha supuesto que el mundo o el espacio de estados de la red son sólo el conjunto de esos cuatro valores. Los patrones son las palabras GREEN y RED, que pueden estar escritas en rojo o en verde. Según le presentemos el color rojo o la palabra RED se activará la neurona correspondiente. El algoritmo de aprendizaje usado es el de retropropagación del error típico del perceptrón multicapa.

Además hay una capa oculta con cuatro neuronas y una capa de salida con sólo dos neuronas, correspondientes a rojo y verde, que es lo que el paciente respondería. La red devolverá la palabra o el nombre del color con el que está escrita la palabra. El modelo a su vez devolverá un tiempo de reacción simulado por un umbral de disparo para las neuronas de salida, y se mide por el número de iteraciones necesarias para alcanzar ese umbral de disparo.

Las dos neuronas de la capa de entrada que nos dan la tarea pedida simulan el sistema atencional, y permiten a la red atender a la presentación de uno u otro estímulo, activando la neurona correspondiente. Durante el entrenamiento la red aprende una sola tarea, por ejemplo primero a leer la palabra RED y la palabra GREEN, y luego a nombrar los colores, pero sin competir entre las distintas tareas.

Los resultados se muestran en la figura [1-b], y puede verse que simulan exactamente los resultados mostrados por los sujetos normales: la tarea de nombrar el color es más lenta que la lectura de la palabra.

Para simular el comportamiento de los sujetos con esquizofrenia se lesionó el modelo, inhibiendo las nidades que correspondan. Los resultados se muestran en el mismo gráfico. La correspondencia entre los resultados obtenidos por la simulación y los resultados esperados es muy grande.

16.2 Simulación conexionista del problema del balancín

Esta tarea fue estudiada por Piaget para demostrar el desarrollo por estadios de los procesos cognitivos. Se les muestra a los niños un columpio balancín, con discos de pesos variables en sus extremos y a diferentes distancias del centro. Lo que debían ver los niños era hacia donde se inclinaría el balancín por efecto de los pesos.

Designaremos como problema *Peso* (P) a aquellos cuyo factor determinante es qué peso es mayor a la misma distancia del centro, y problema *Distancia* (D)

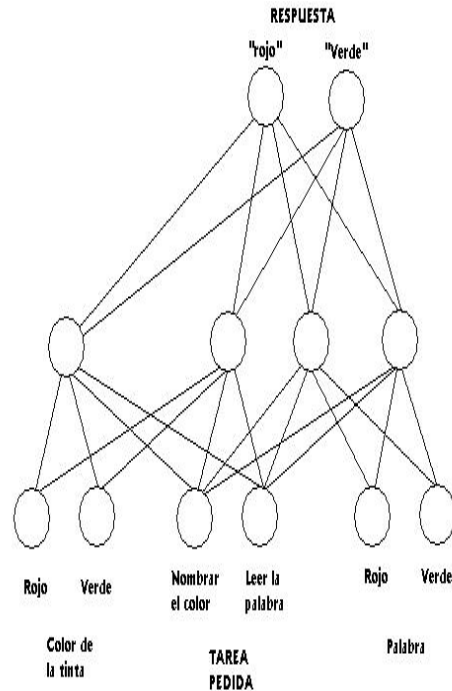
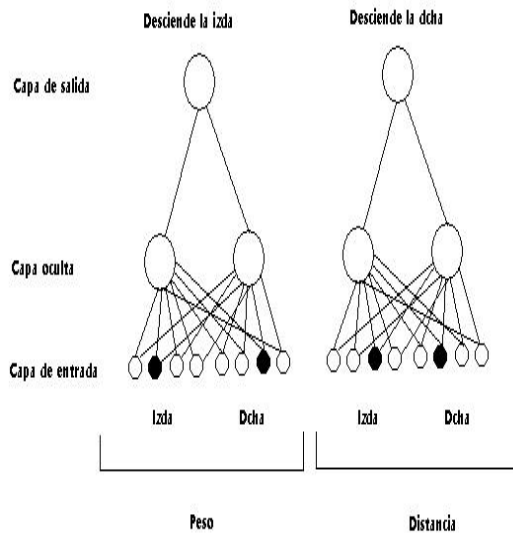


Figura 6: Una red para modelar la tarea de Stroop. Puede leer letras coloreadas o nombrar el color de la tinta con el que está escrito. Las unidades *tarea pedida* simulan el mecanismo de atención selectiva en el modelo.

a aquellos cuyo factor determinante es la distancia del centro, a igualdad de pesos. En los problemas llamados *Conflicto* el mayor peso está a la distancia más pequeña del centro, por lo que el niño debía atender a las dos variables [13]. El objetivo de esta explicación no es mostrar todos los resultados de la tarea en cuestión, no directamente relacionada con el tema del trabajo, sino describir la arquitectura de la red neuronal como posible modelo para procesos que muestren rasgos de lateralización, por lo que vamos a pasar a describirla a continuación.

Lo novedoso de esta arquitectura es que está dividida en dos partes claramente diferenciadas, que simulan las dos variables que intervienen en el problema. Llamaremos a la parte de la izquierda *Canal peso* y a la parte de la derecha *Canal distancia*. A su vez estas cada una de estas partes está dividida en dos, que simulan las dos mitades del balancín. Todo esto sería la capa de entrada. En nuestro caso se compondría de 20 unidades (dos partes de 10, y cada una de



ellas de 5+5) como muestra la figura.

Para el caso en el que por ejemplo quisieramos representar dos discos de una unidad de peso cada uno situados uno a cada lado del balancín activaríamos la parte de la derecha que denota el peso con un patrón $P_w = (1000000001)$. Supongamos que estuviera a tres unidades de distancia del centro en la parte de la izquierda y a dos en la derecha. El patrón con el que activaríamos la red sería $P_d = (0010001000)$. Si juntamos las dos partes el patrón final de entrada quedaría:

$$P = (10000000010010001000)$$

Esto excitaría la capa oculta, que se compone de 4 unidades (2 + 2), que a su vez propagaría al señal a la capa de salida, formada por 2 unidades. La razón de que la capa de salida se componga de dos unidades es muy clara: simula hacia donde va a inclinarse el balancín, si a la derecha o a la izquierda, según se active una neurona u otra.

Referencias

- [1] Rumheltart, J. y McClelland, D. and the PDP Research Group. (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 1.* The MIT Press.
- [2] Rumheltart, J. y McClelland, D. and the PDP Research Group. (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 2.* The MIT Press.
- [3] McLeod, P., Plunkett, K., Rolls, E. T. (1998). *Introduction to Connectionist Modelling of Cognitive Processes.* Oxford University Press.
- [4] McLeod, P., Plunkett, K., Rolls, E. T. (1998). *Rethinking Innateness.* Oxford University Press.
- [5] McLeod, P., Plunkett, K., Rolls, E. T. (1998). *Exercises in Rethinking Innateness.* Oxford University Press.
- [6] Bonifacio Martínez del Brio. (2001) *Redes neuronales y sistemas borrosos.* Ed. Ra Ma.
- [7] James A. Anderson. *Neurocomputing. Foundations of Research.* Ed. The MIT Press.
- [8] Jose R. Hiler. *Redes neuronales artificiales.* Ed. Ra Ma.
- [9] Wicker, Devert (2002). *E-Net: Evolutionary neural network synthesis* Neurocomputing 42, 171-196.
- [10] Browne, Anthony. Sun, Ron (2001) *Connectionist inference models* Neural Networks 14, 1331-1335.