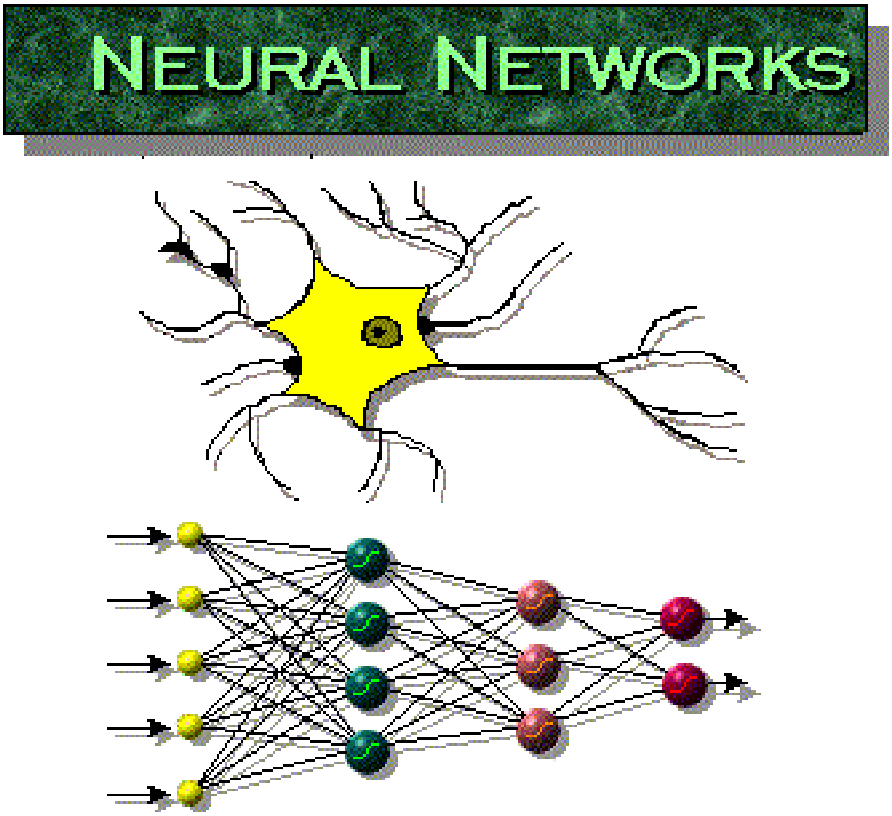
Introducción a las Redes Neuronales



José Manuel Gutiérrez (Universidad de Cantabria)
gutierjm@ccaix3.unican.es
http://ccaix3.unican.es/~gutierjm
http://ccaix3.unican.es/~AIGroup

JAVA:
http://www.aist.go.jp/NIBH/~b0616/Lab/Links.html

Universidad de Cantabria

DEFINICIONES DE INTELIGENCIA ARTIFICIAL

Las "definiciones" de Inteligencia Artificial (IA) involucran las siguientes ideas:

- Métodos y algoritmos que permitan comportarse a las computadoras de modo inteligente.
- Estudio de las facultades mentales a través del uso de modelos computacionales.
- Desarrollo de autómatas (máquinas construidas por el hombre) para simular el proceso deductivo e inductivo humano de adquisición y aplicación de conocimiento.

Dogma central de la IA:

- "los procesos que se llevan a cabo en el cerebro pueden ser analizados, a un nivel de abstación dado, como procesos computacionales de algún tipo".

Universidad de Cantabria

CONTROVERSIA

Durante el desarrollo de la IA siempre ha existido una controversia sobre los límites de esta Ciencia

- **Hubert L. Dreyfus**, *What Computers Can't Do*

  "Great artists have always sensed the truth, stubbornly denied by both philosophers and technologists, that the basis of human intelligence cannot be isolated and explicitly understood."

- **Donald Michie**, Chief Scientist of the *Turing Institute*

  "It is a mistake to take up too much time asking, "Can computers think?" "Can they be really creative?" For all practical purposes they can. The best course for us is to leave the philosophers in their dark room and get on with using the creative computer to the full."

Universidad de Cantabria

EVOLUCION DE LA IA

Primera Generación (hasta 1955)

Esta generación estableció las bases filosóficas y los primeros modelos matemáticos de esta disciplina.

Alan Turing (1950) publicó *Computing Machinery and Intelligence* donde sugiere: "machines may someday compete with men in all purely intellectual pursuits."

Claude Shannon (1950) publicó "A Chess-Playing Machine" en *Scientific American* analizando el problema del juego automático de ajedrez (10^{120} movimientos posibles).

En 1953, publicó "Computers and Automata" con nuevos y sugestivos interrogantes. ¿Podrá construirse una máquina que

1. localice y repare sus propias averías?
2. que se programe a sí misma?
3. que "aprenda"?

Warren McCulloch and Walter Pitts (1943) primer modelo matemático de *red neuronal* en "A Logical Calculus of the Ideas Immanent in Nervous Activity". Este modelo consistía en una red de neuronas binarias y sinapsis. Este modelo es esencialmente equivalente a una máquina de Turing.

Universidad de Cantabria

EVOLUCION DE LA IA

Segunda Generación

- **La Conferencia de Dartmouth**
  Organizada by John McCarthy y Marvin Minsky (1956), fundó las bases modernas de esta disciplina bajo el lema

  > to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it.

- **Logic Theorist** fue uno de los primeros programas eficientes desarrollado Allen Newell, Herbert Simon y J.C. Shaw. Fue capaz de probar 38 de los primeros 52 teoremas del Capítulo 2 del libro *Principia Mathematica* de Whitehead y Russell.

- **Mycin** fue el pionero de los sistemas expertos (desarrollado por Edward Shortliffe). Puede diagnosticar infecciones bacterianas a partir de síntomas.

- En 1957 Allen Newell predijo que en un plazo de 10 años, un ordenador sería el campeón del mundo de ajedrez.

Universidad de Cantabria

EL TEST DE TURING

Alan M. Turing (1912-1954), en el artículo "Computing Machinery and Intelligence", describió un *juego de imitación* para probar la "inteligencia" de las máquinas: *"If conversation with a computer is indistinguishable from that with a human, the computer is displaying intelligence."*

- **Test de Turing: Primera Fase**
  - An interrogator tries to determine which of two communicators is a man and which is a woman by questioning each. The rules of the game require the man to try to fool the interrogator and the woman to try to convince him that she is the woman. Queries are made through a neutral medium such as a remote terminal and each party is isolated in a separate room to eliminate any visual or audible clues.
- **Test de Turing: Segunda Fase**
  - Now the man is replaced by a computer and the game resumes. If the computer can deceive the interrogator as often as the man did, we say the computer is displaying intelligence.

Universidad de Cantabria

CARACTERISTICAS DE LOS MODELOS DE IA

Los modelos y algoritmos "estándard" de la IA tienen las siguientes características:

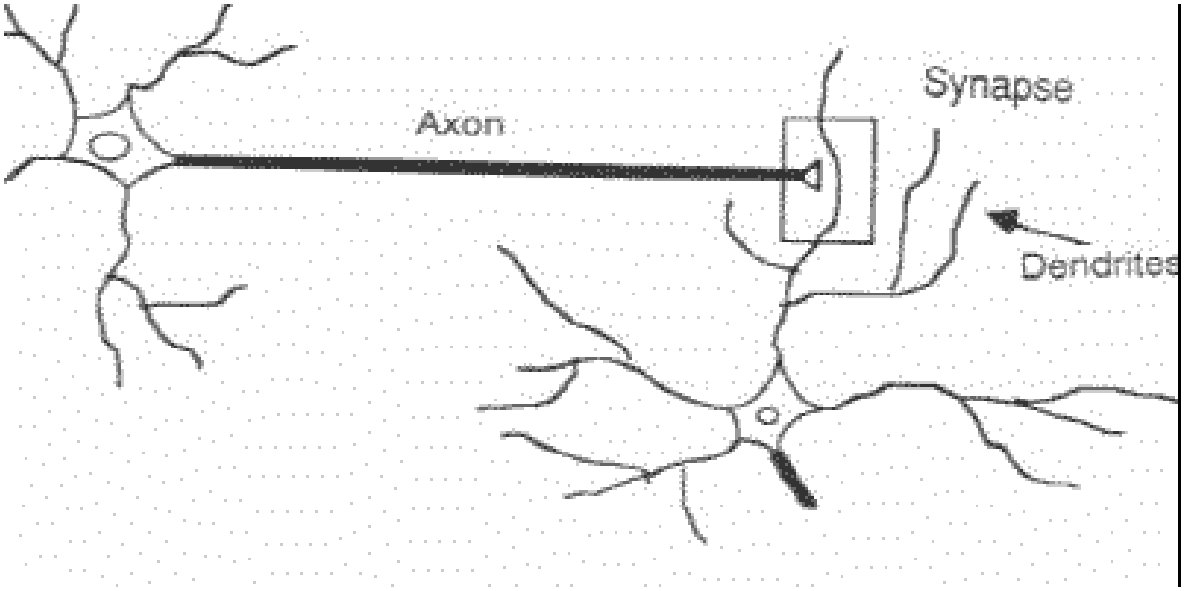
- *El conocimiento se representa explícitamente* usando reglas, redes semánticas, modelos probabilísticos, etc.,
- *Se imita el proceso humano de razonamiento lógico* para resolver los problemas, centrando la atención en las causas que intervienen en el problema y en sus relaciones (encadenamiento de reglas, inferencia probabilística), y
- *Se procesa la información secuencialmente.*

Con el rápido desarrollo de la IA aparecieron un gran número de problemas complejos donde no no se disponía de una representación explícita del conocimiento y de un procedimiento de razonamiento lógico para resolverlo.
Posibles causas de este fallo: Procedimientos algoritmicos y estructura computacional empleados eran incorrectos.
Posible solución: Uso de estructuras computacionales paralelas inspiradas en redes neuronales biológicas.

Universidad de Cantabria

INSPIRACION BIOLOGICA

Las **neuronas** receiven señales (inputs) de otras neuronas via conexiones **sinápticas** que pueden ser excitantes o inhibidoras. En función de las señales recibidas, una neurona envía a su vez una señal a otras neuronas por medio del **axón**.



Una neurona contiene un potencial interno continuo llamado **potencial de membrana**. Cuando éste excede un cierto valor **umbral**, la neurona puede transmitir todo su potencial por medio del axón.

Se estima que el cerebro humano contiene más de cien mil millones (10^{11}) de neuronas y que hay más de 1000 sinápsis a la entrada y a la salida de cada neurona.

Universidad de Cantabria

REDES NEURONALES ARTIFICIALES (NEURAL NETWORKS)

Neural Network Study (1988, AFCEA International Press, p. 60):

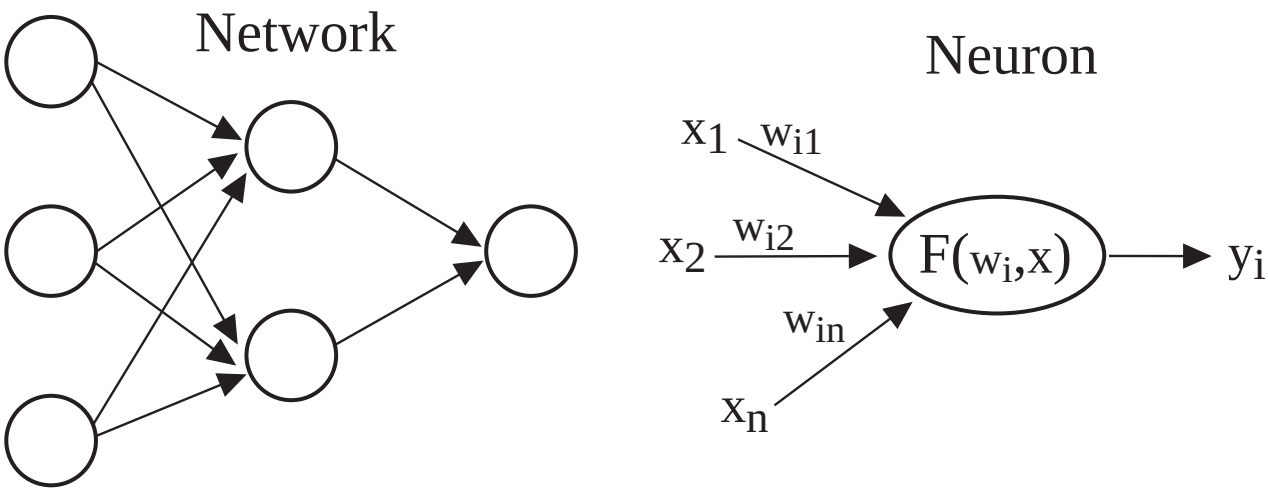
> *... a neural network is a system composed of many simple processing elements operating in parallel whose function is determined by network structure, connection strengths, and the processing performed at computing elements or nodes.*

Haykin, S. (1994), Neural Networks: A Comprehensive Foundation, NY: Macmillan, p. 2:

> *A neural network is a massively parallel distributed processor that has a natural propensity for storing experiential knowledge and making it available for use. It resembles the brain in two respects:*
>
> 1. *Knowledge is acquired by the network through a learning process.*
> 2. *Interneuron connection strengths known as synaptic weights are used to store the knowledge.*

Universidad de Cantabria

REDES NEURONALES ESTRUCTURA



- Las redes neuronales artificiales están copuestas de gran cantidad de procesadores conectados entre sí y actuando en paralelo.

  Los modelos neuronales biológicos son mucho más complejos que los modelos computacionales actuales.

- El comportamiento de la red está determinado por su topología, los pesos de las conexiones y la función característica de las neuronas.

Universidad de Cantabria

REDES NEURONALES CARACTERISTICAS

- **Aprendizaje Adaptativo**: Las RNA aprenden a realizar tareas a partir de un conjunto de datos dados en el proceso de aprendizaje.
- **Auto-organización**: Pueden crear su propia organización o representación de la información recibida.
- **Operación en tiempo real**: Las operaciones realizadas pueden ser llevadas a cabo por computadores paralelos, o dispositivos de hardware especiales que aprovechan esta capacidad.
- **Tolerancia a fallos parciales**: La destrucción parcial de una red daña parcialmente el funcionamiento de la misma, pero no la destruye completamente. Esto es debido a la redundancia de la información contenida.

Universidad de Cantabria

REDES NEURONALES DEFINICIONES

Definition 1 (Neurona o Unidad Procesadora) *Una neurona, o unidad procesadora, sobre un conjunto de nodos N, es una tripleta (X, f, Y), donde X es un subconjunto de N, Y es un único nodo de N y $f :\longrightarrow$ es una función neuronal (también llamada función activación) que calcula un valor de salida para Y basado en una combinación lineal de los valores de las componentes de X, es decir,*

$$Y = f(\sum_{x_i \in X} w_i \, x_i).$$

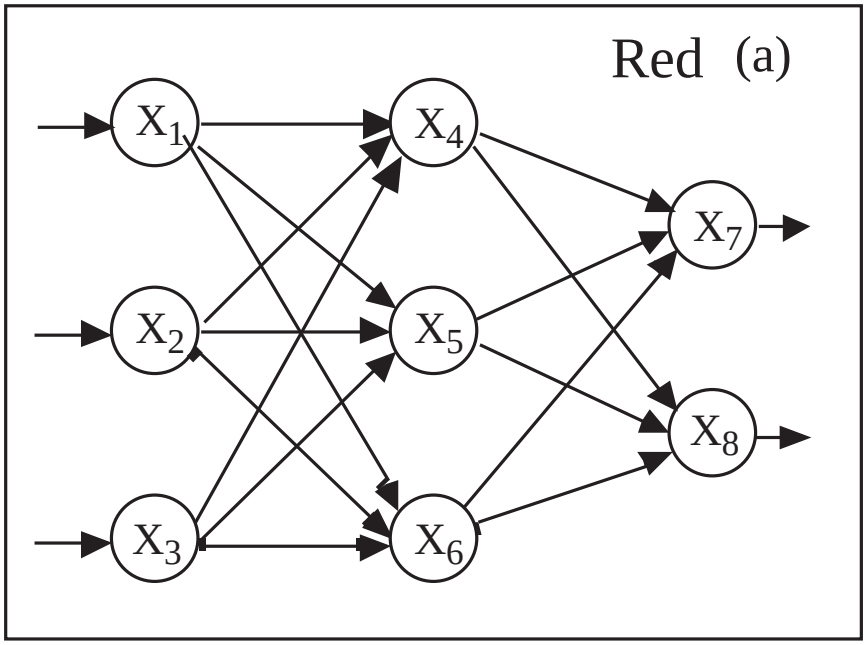
Los elementos X, Y y f se denominan conjunto de nodos de entrada, conjunto de nodos de salida, y función neuronal de la unidad neuronal, respectivamente.

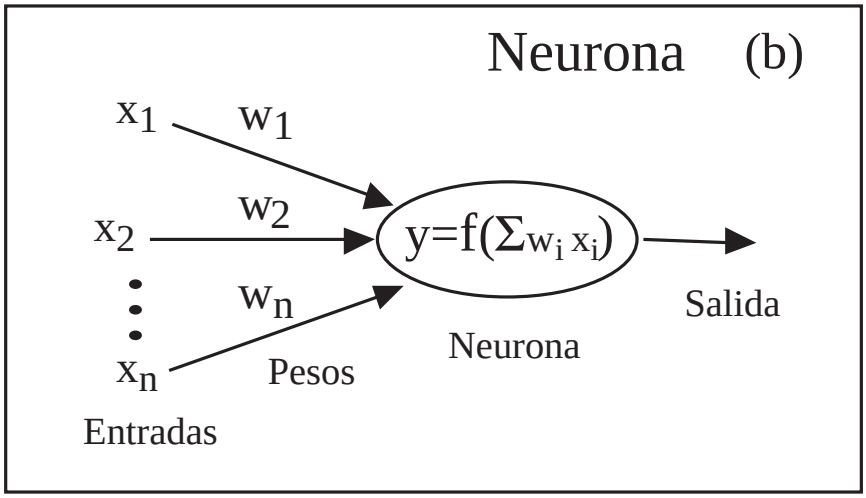
Definition 2 (Red Neuronal Artificial) *Una red neuronal artificial (RNA) es un par (N, U), donde N es un conjunto de nodos y U es un conjunto de unidades procesadoras sobre N que satisface la siguiente condición: Cada nodo $X_i \in N$ tiene que ser un nodo de entrada o de salida de al menos una unidad procesadora de U.*

Universidad de Cantabria

REDES NEURONALES
EJEMPLO

$$U_1 = (\{x_1, x_2, x_3\}, f_1, \{x_4\}),$$
$$U_2 = (\{x_1, x_2, x_3\}, f_2, \{x_5\}),$$
$$U_3 = (\{x_1, x_2, x_3\}, f_3, \{x_6\}),$$
$$U_4 = (\{x_4, x_5, x_6\}, f_4, \{x_7\}), y$$
$$U_5 = (\{x_4, x_5, x_6\}, f_5, \{x_8\}).$$





Universidad de Cantabria

EL APRENDIZAJE

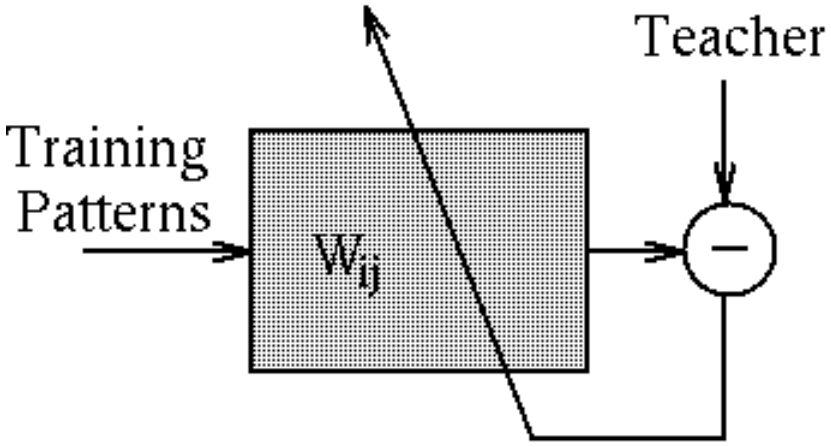
Existen dos fases en toda aplicación de las redes neuronales: la fase de aprendizaje o entrenamiento y la fase de prueba.

- **Fase de Aprendizaje**: una característica de las redes neuronales es su capacidad de aprender. Aprenden por la actualización o cambio de los pesos sinápticos que caracterizan a las conexiones. Los pesos son adaptados de acuerdo a la información extraída de los patrones de entrenamiento nuevos que se van presentando. Normalmente, los pesos óptimos se obtienen optimizando (minimizando o maximizando) alguna "función de energía". Por ejemplo, un criterio popular en el entrenamiento supervisado es minimizar el least-square-error (error cuadratico medio) entre el valor deseado y el valor de salida de la red.
- **Fase de Prueba**: Una vez calculados los pesos de la red, las neuronas de la última capa se comparan con la salida deseada para determinar la validez del diseño.

Universidad de Cantabria

EL APRENDIZAJE METODOS

Supervisado: Los datos están consistituidos por varios patrones de entrada y de salida. El hecho de conocer la salida implica que el entrenamiento se beneficia la supervisión de un maestro.



No Supervisado: Para los modelos de entrenamiento No Supervisado, el conjunto de datos de entrenamiento consiste sólo en los patrones de entrada. Por lo tanto, la red es entrenada sin el beneficio de un maestro. La red aprende a adaptarse basada en las experiencias recogidas de los patrones de entrenamiento anteriores.

Supervisado	No Supervisado
Perceptrón / multicapa	Mapa de características
Modelos temporales	Redes competitivas

Universidad de Cantabria

APRENDIZAJE DE HEBB

Hebb describe una forma de ajustar el peso de una conexión acorde a la correlación exitente entre los valores de las dos unidades de proceso que conecta. En su libro, "The Organization of Behavior (1949)":

> *When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes a part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency as one of the cells firing B is increased.*

Computacionalmente, esto se traduce en:

- **No Supervisado**: el valor del peso w_{ij} es la correlación del valor de las unidades x_i y x_j:

$$\Delta w_{ij} = x_i x_j$$

- **Supervisado:** Se ajusta el valor del peso entre las dos unidades en proporción a la diferencia entre los valores deseado y calculado en cada una de las unidades de la capa de salida.

$$\Delta w_{ij} = \alpha x_i [y_j - \hat{y}_j]$$

α es la constante de aprendizaje $(0 < \alpha \ll 1)$.

Universidad de Cantabria

VALIDACION DEL MODELO

Una vez que ha terminado el proceso de aprendizaje y los pesos de la red neuronal han sido calculados, es importante comprobar la calidad del modelo resultante. Algunas medidas estándar del error son:

1. La suma de los cuadrados de los errores (Sum Square Errors, SSE), definida como

$$\sum_{p=1}^{r} \| b_p - \hat{b}_p \|^2 . \qquad (1)$$

2. La raíz cuadrada del error cuadrático medio (Root Mean Square Error, RMSE) definida como

$$\sqrt{\sum_{p=1}^{r} \| b_p - \hat{b}_p \|^2 / r}. \qquad (2)$$

3. El error máximo,

$$max\{\| b_p - \hat{b}_p \|,\ p = 1, \ldots, r\}, \qquad (3)$$

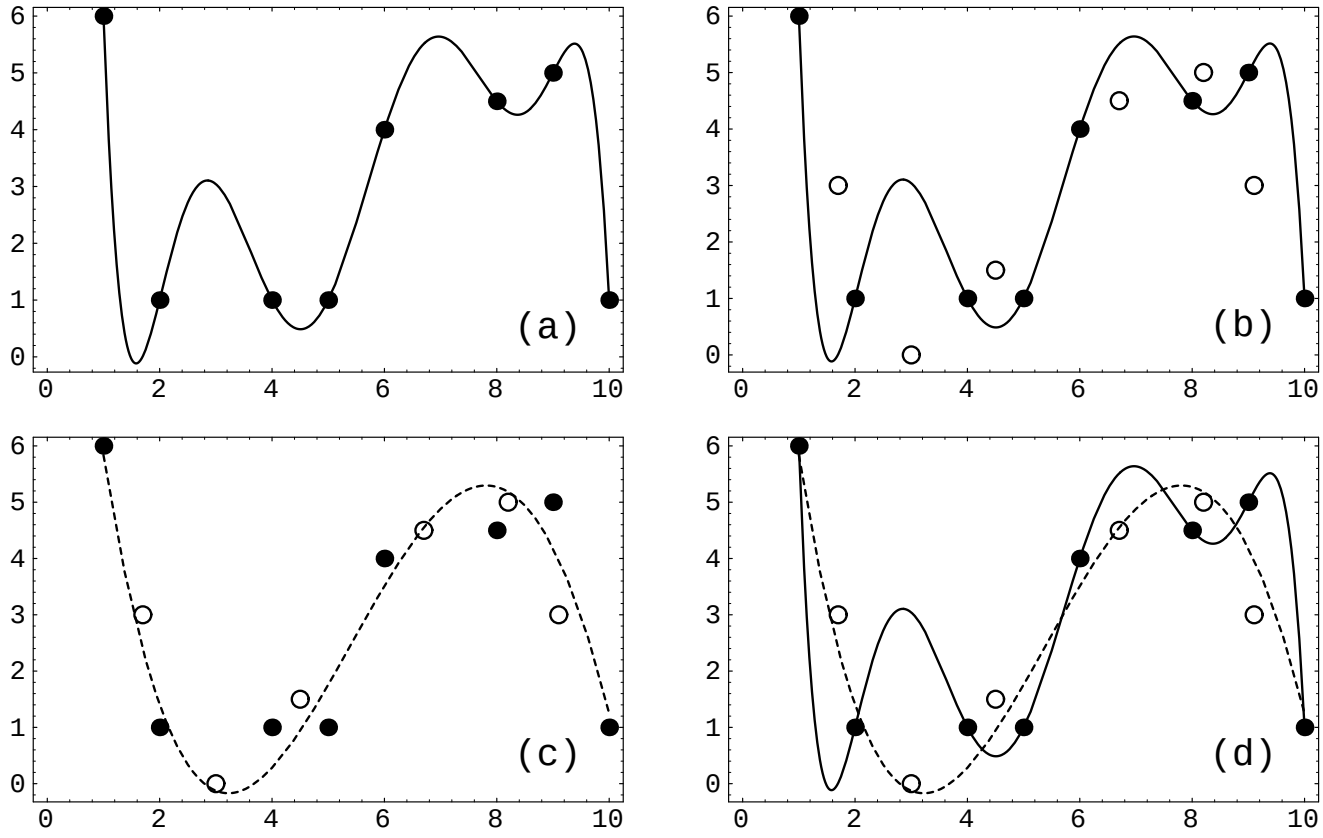
donde $\hat{b}_p$ es la salida de la red para el vector de entrada a_p. Nótese que en el caso de una única salida, la función norma $\| \, . \, \|$ se reduce a la función valor absoluto $| \, . \, |$ usual.

SOBREAJUSTE (OVERFITTING)

Universidad de Cantabria

En estadística es bien conocido que cuando se utiliza un modelo con muchos parámetros para ajustar un conjunto de datos procedente de proceso con pocos grados de libertad, el modelo obtenido puede no descubrir las tendencias reales del proceso original, aunque pueda presentar un error pequeño.

La curva (a) pasa exactamente por los puntos de entrenamiento. La curva (b) muestra el comportamiento sobre un conjunto alternativo. La curva (c) es un polinomio de tercer grado. La diferencia entre ambos modelos puede verse en (d).



Universidad de Cantabria

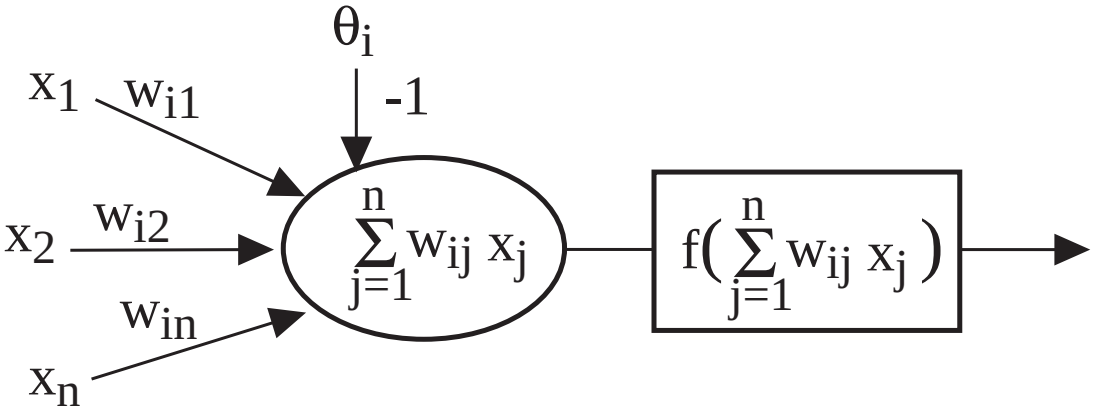
NEURONAS
UNIDADES DE PROCESO

- **Neuronas:** $X = (x_1, \ldots, x_i, \ldots, x_n)$, donde x_i es el nivel de actividad de la i-ésima neurona.
- **Pesos:** los pesos de la neurona i-ésima forman un vector $W_i = (w_{i1}, \ldots, w_{ij}, \ldots, w_{i,n})$, donde w_{ij} es el peso de la conexión de x_j a x_i.

La actividad lineal de x_i está dada por la función,

$$F(x_i, W_i) = \sum_{j=1}^{n} w_{ij} x_j,$$

que depende de los pesos W_i.



Para incluir un valor umbral Θ_i para la neurona x_i, se considera una neurona auxiliar de valor $x_0 = -1$ y se conecta a x_i con un peso Θ_i.

$$u(w, x_i) = \sum_{j=1}^{n} w_{ij} x_j - w_{i0}\Theta_i \quad or \quad u(w, x_i) = W_i \cdot X$$

FUNCIONES DE ACTIVACION

- **Funciones lineales:** $f(x) = x$.
- **Funciones paso:**Dan una salida binaria dependiente de si el valor de entrada está por encima o por debajo del valor umbral.

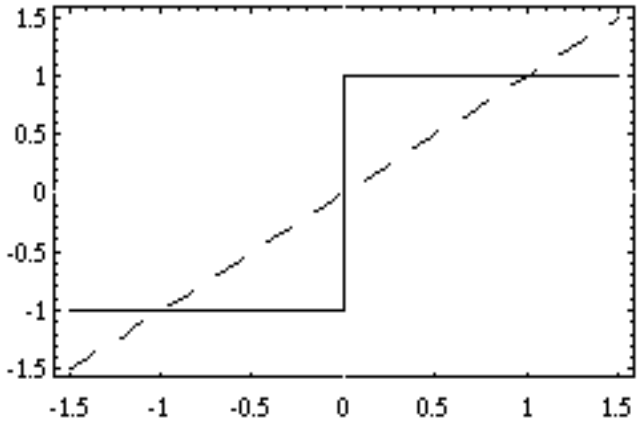
$$sgn(x) = \begin{cases} -1, \text{ si } x < 0, \\ 1, \text{ sino,} \end{cases} \quad , \quad \Theta(x) = \begin{cases} 0, \text{ si } x < 0, \\ 1, \text{ sino.} \end{cases}$$

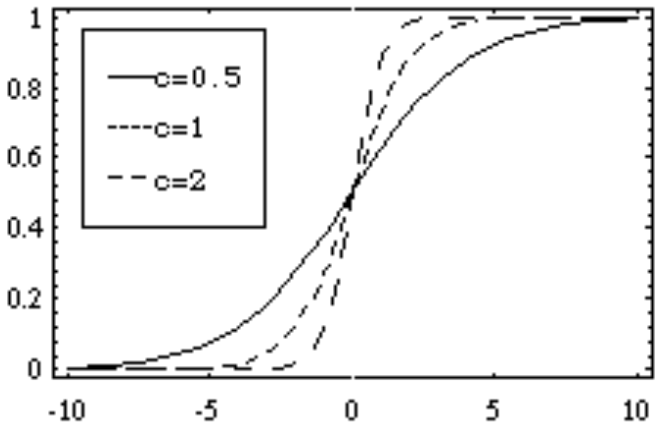
- **Funciones sigmoidales:** Funciones monótonas acotadas que dan una salida gradual no lineal.

1. La función logística de 0 a 1:

$$f_c(x) = \frac{1}{1 + e^{-c\,x}}.$$

2. La función tangente hiperbólica de -1 a 1 $f_c(x) = tanh(c\,x)$.

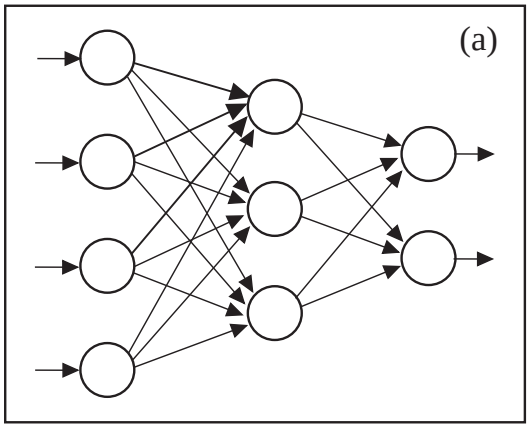


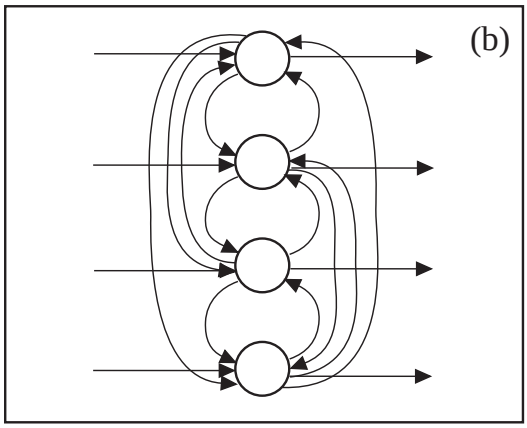


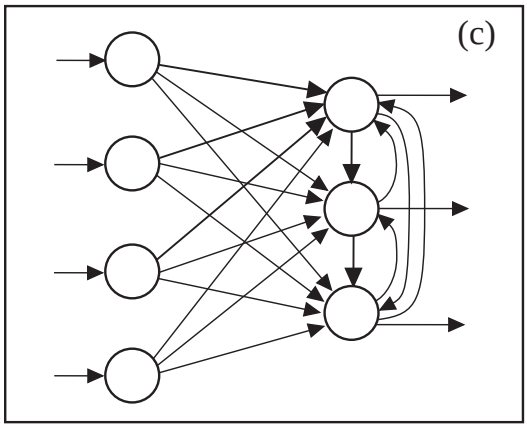
Universidad de Cantabria

ARQUITECTURAS DE RED

1. *Conexiones hacia delante.*
2. *Conexiones laterales.*
3. *Conexiones hacia atrás (o recurrentes).*







LA RED DE HOPFIELD

La red neuronal de Hopfield es una arquitectura formada por una sóla capa principalmente que se utiliza principalmente como memoria autoasociativa, para almacenar y recuperar información.

La información se almacena usando un método simple de aprendizaje no supervisado que obtiene la matriz de pesos que hace que dado cada uno de los patrones de entrenamiento (almacenamiento) la red devuelva el mismo patrón (recuperación).

Posteriormente, cuando se tenga una configuración arbitraria de las neuronas como entradas, la red devolverá aquel patrón almacenado que esté más cerca de la configuración de partida en términos de la distancia de Hamming

Universidad de Cantabria

LA RED DE HOPFIELD MEMORIZANDO DATOS

Se considera una red neuronal de una sóla capa sobre un conjunto de neuronas binarias $\{x_1, \ldots, x_n\}$ (con valores en $\{-1, 1\}$) donde cada neurona x_i posee un vector de pesos $w_i = (w_{i1}, \ldots, w_{in})$, con $w_{ii} = 0$ indicando que no existe conexió consigo misma.
Se considera también la siguiente definición binaria de la neurona:

$$x_i = sgn(\sum_{j=1}^{n} w_{ij} x_j). \tag{4}$$

Ahora, supóngase que se quieren obtener los pesos apropiados para "memorizar" un patrón $a = (a_1, \ldots, a_n)$. Entonces, los pesos tienen que satisfacer las siguientes condiciones de estabilidad:

$$a_i = sgn(\sum_{j=1}^{n} w_{ij} a_j),\ i = 1 \ldots, n, \tag{5}$$

p or tanto la red devuelve el mismo patrón dado como entrada. Como se están usando los valores neuronales $\{-1, 1\}$, entonces $a_j^2 = 1$ y las condiciones anteriores de estabilidad se pueden alcanzar considerando los pesos

$$w_{ij} = \frac{1}{n} a_i a_j. \tag{6}$$

LA RED DE HOPFIELD MEMORIZANDO DATOS

El mismo algoritmo puede extenderse a varios patrones, $\{(a_{p1}, \ldots, a_{pn}),\ p = 1, \ldots, r\}$:

$$w_{ij} = \frac{1}{n} \sum_{p=1}^{r} a_{pi} a_{pj}. \tag{7}$$

En este caso, cuando se da como entrada un patrón a_p se obtiene

$$\begin{aligned} x_i &= sgn(\frac{1}{n} \sum_j \sum_{k=1}^{r} a_{ki} a_{kj}\, a_{pj}) \\ &= sgn(\frac{1}{n} \sum_j a_{pi} a_{pj}\, a_{pj} + \sum_j \sum_{k \neq p} a_{ki} a_{kj}\, a_{pj}) \\ &= sgn(a_{pi} + \frac{1}{n} \sum_j \sum_{k \neq p} a_{ki} a_{kj}\, a_{pj}). \end{aligned} \tag{8}$$

El problema de estabilidad se determina por los términos de correlación cruzada en el segundo término de la suma en (8). Si es más pequeño que n, entonces se puede concluir que el patrón es estable. Un análisis estadístico de estos términos cruzados demostró que cuando el número de patrones a ser almacenado, p, y el número de neuronas, n, satisface la relación: $p < 0.138\, n$, entonces menos del 1% de los bits es inestable

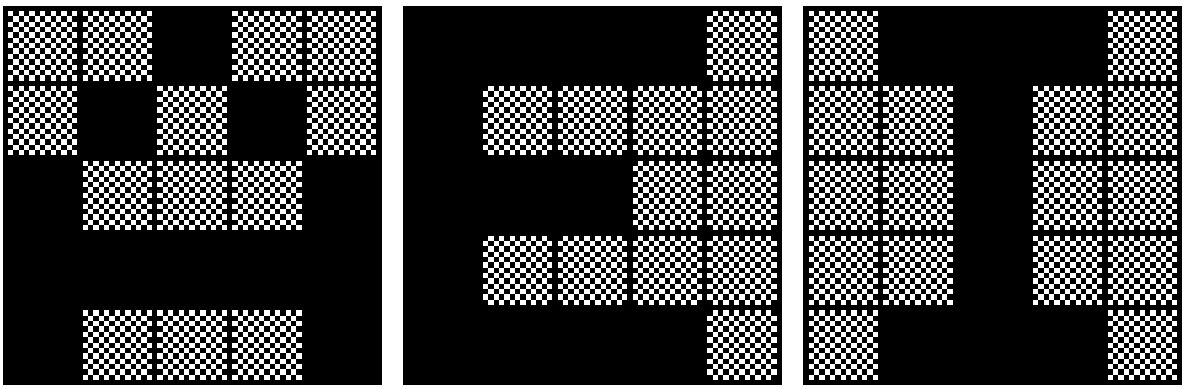
Universidad de Cantabria

EJEMPLO: RECONOCIMIENTO DE CARACTERES

Para este ejemplo se utiliza una red de Hopfield con 25 neuronas consistente en una cuadrícula 5×5 que recibe una imagen de puntos de una letra. Se consideran

únicamente las tres representaciones habituales de las vocales 'A', 'E', e 'I'.

Los dígitos 'A', 'E', e 'I' se representan como '-1-11-1-1…', '1111-1…', y '-1111-1…', respectivamente donde los valores negativos se representan en gris y los positivos en negro.



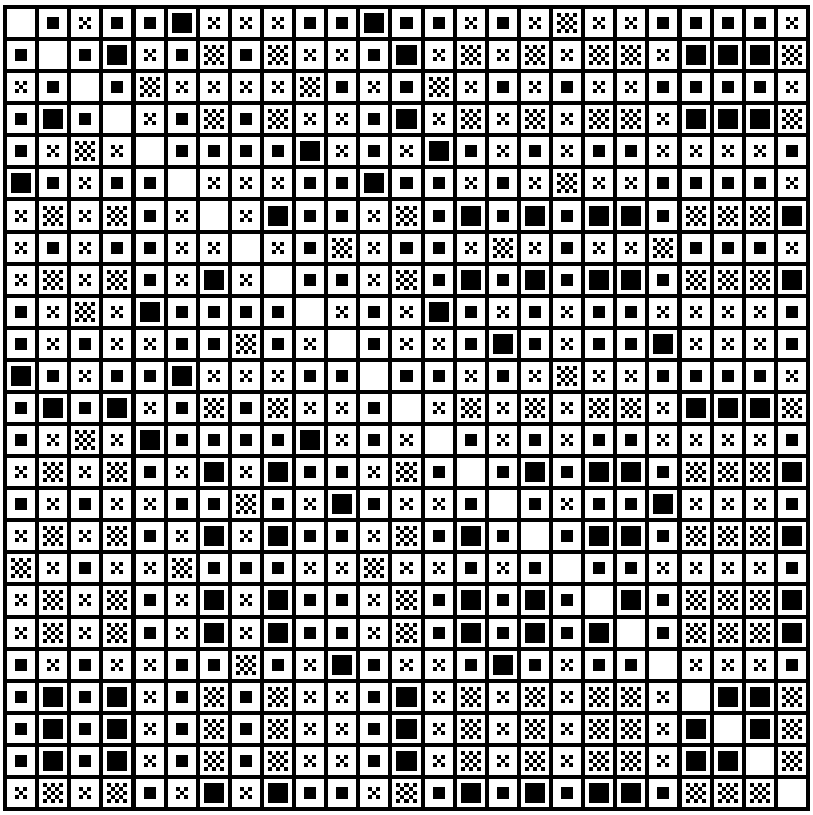
Universidad de Cantabria

EJEMPLO: RECONOCIMIENTO DE CARACTERES

La matriz de pesos 25×25 se obtiene fácilmente. Por ejemplo,

$$w_{12} = \frac{1}{25} \sum_{k=1}^{3} a_{1k} a_{2k} = \frac{1}{25}(-1 \times -1 + 1 \times 1 - 1 \times 1) = \frac{1}{25} \times 1.$$

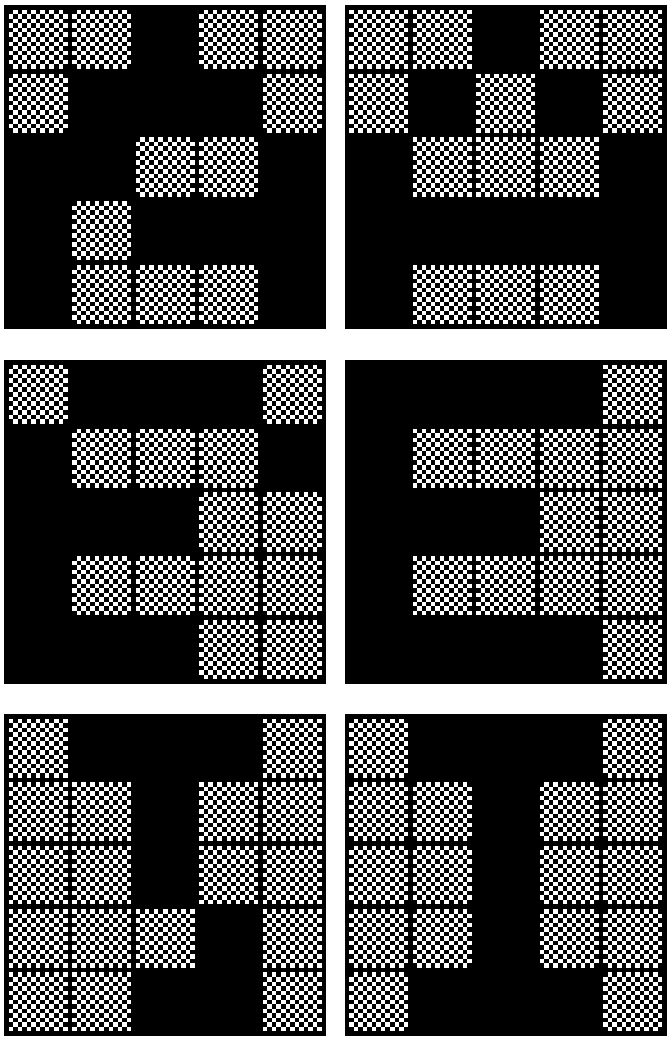
El valor resultante, ignorando la constante de normalización, se representa por un cuadrado negro de pequeño tamaño en la esquina superior izquierda (peso w_{12}). En esta figura, los colores negro y gris están asociados a los pesos positivos y negativos, respectivamente, donde el tamaño de la caja representa la magnitud del valor.



Universidad de Cantabria

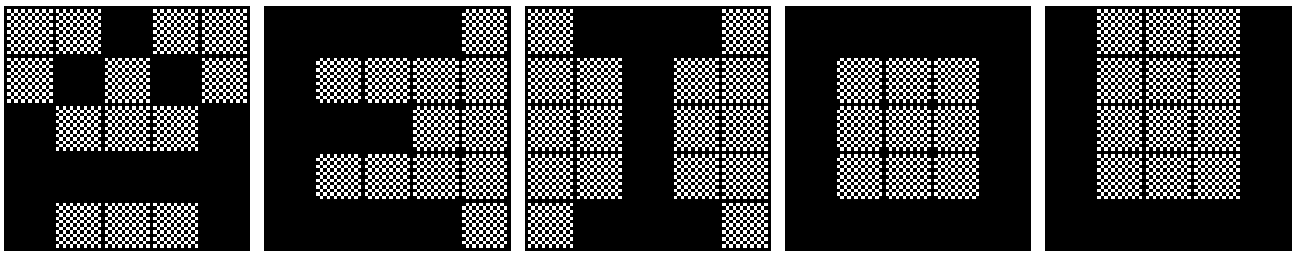
INFORMACION CORRUPTA

Una vez construída la red de Hopfield, ésta reconoce las tres vocales aún cuando se utiliza como entrada cualquier otra cuadrícula (digitalización) 5×5 correspondiente a una versión particular no estándar de la vocal.

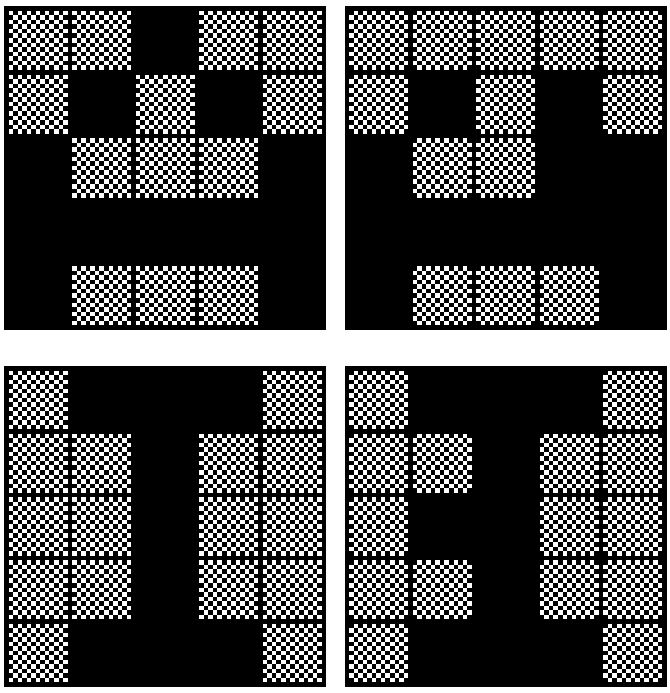


ESTADOS ESPUREOS

Si se intenta utilizar la misma arquitectura de red para reconocer las cinco vocales



como en este caso el número de patrones es mayor de $0.138 \times n = 0.138 \times 25 = 3.45$, pueden aparecer algunos estados de falsa estabilidad en el modelo.



REDES MULTI-CAPA

Entre las arquitecturas de red más populares destacan las llamadas redes multi-capa o de retro-propagación.

Definition 3 (Capa de Entrada de una Red Neuronal) *Una unidad se dice que está en la capa de entrada de una red neuronal* (X, U)*, si es la entrada de al menos un unidad funcional de* U *y no es la salida de ninguna unidad procesadora de* U*.*

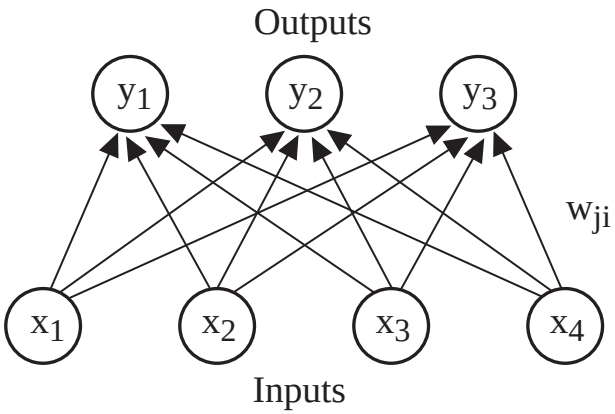
Definition 4 (Capa de Salida de una Red Neuronal) *Una unidad se dice que está en la capa de salida de una red funcional* (X, U)*, si es la salida de al menos una unidad funcional de* U *y no es la entrada de ninguna unidad procesadora de* U*.*

Definition 5 (Capas Intermedias u Ocultas de una Red Neuronal) *Una unidad se dice que está en la capa intermedia de una red neuronal* (X, U)*, si es la entrada de al menos una unidad funcional de* U *y, al mismo tiempo, es la salida de al menos una unidad procesadora de* U*.*

Universidad de Cantabria

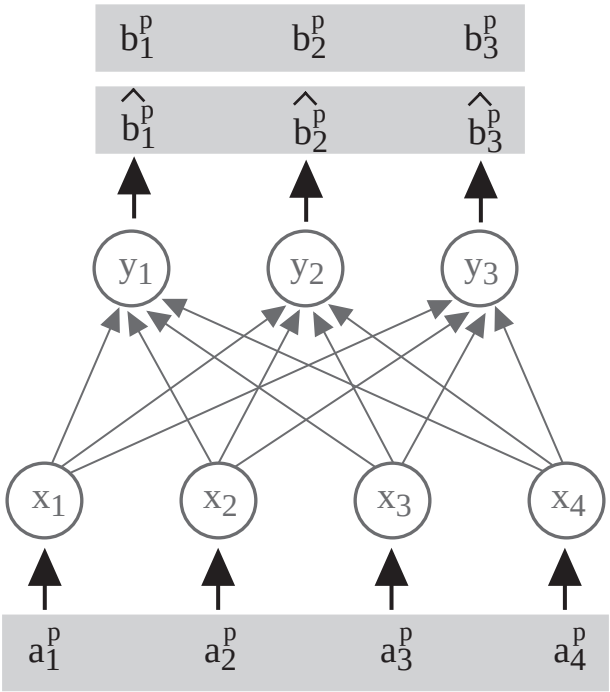
EL PERCEPTRON

El perceptrón es un red neuronal de dos capas (una de entrada y una de salida) con conexiones hacia delante.



$$y_i = f(Y_i) = f(\sum_j w_{ij} x_j)$$

¿Cómo se pueden obtener los pesos adecuados para "aprender" un conjunto de patrones?.



Universidad de Cantabria

EL PERCEPTRON (APRENDIZAJE)

- **Aprendizaje Hebbiano:** Inicialmente se eligen valores aleatorios para los pesos. La idea del aprendizaje Hebiano era modificar los pesos acorde a la correlación entre las unidades. Se eligen los patrones de uno en uno; por ejemplo (a^p, b^p). Si $b_i^p \neq \hat{b_i^p}$, entonces se modifica el peso:

$$\Delta w_{ij} = \eta(b_i^p - \hat{b_i^p})a_j^p$$

- **Descenso de Gradiente:** Inicialmente se eligen valores aleatorios para los pesos. La idea de este método es utilizar un proceso iterativa que minimice la función de error

$$E(w) = \frac{1}{2}\sum_{i,p}(b_i^p - \hat{b_i^p})^2.$$

En el caso lineal $(f(x) = x)$ se tiene

$$\Delta w_{ij} = -\eta\frac{\partial E}{\partial w_{ij}} = \eta\sum_p(b_i^p - \hat{b_i^p})a_j^p.$$

En general, se tiene

$$\Delta w_{ij} = -\eta\frac{\partial E}{\partial w_{ij}} = \eta\sum_p(b_i^p - \hat{b_i^p})f'(B_i^p)a_j^p.$$

El parámetro η se llama tasa de aprendizaje.

EL PERCEPTRON (APRENDIZAJE)

En el caso de funciones sigmoidales, las fórmulas anteriores no involucran derivadas simbólicas, pues

$$f(x) = \frac{1}{1+e^{-c\,x}} \quad \Rightarrow \quad f'(x) = c\,f(x)\,(1-f(x))$$
$$f(x) = tanh(c\,x) \quad \Rightarrow \quad f'(x) = c\,(1-f(x)^2)$$

Se han propuesto distintas mejoras de este método para incrementar su eficiencia.

- Se puede incluir un parámetro de inercia α para acelerar la convergencia al mínimo:

$$\Delta w_{ij}(t+1) = -\eta \frac{\partial E}{\partial w_{ij}} + \alpha \Delta w_{ij}(t-1)$$

- Otros métodos incluyen términos la función de error que penalizan grandes pesos:

$$E(w) = \sum_{p=1}^{r} (y_p - \hat{y}_p)^2 + \lambda \sum_{i,j} w_{ij}^2, \qquad (9)$$

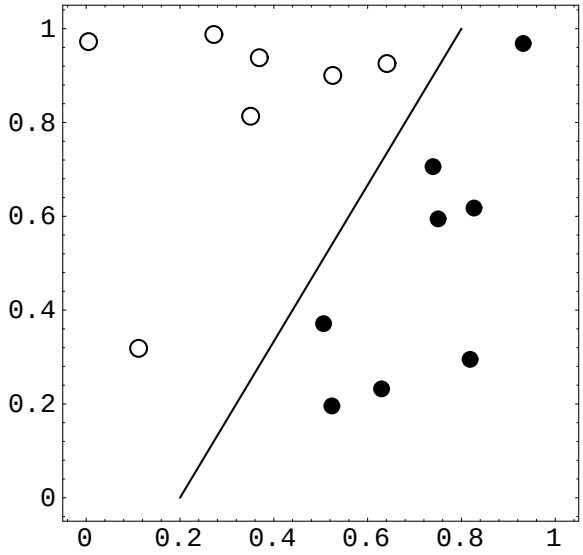
donde λ es un parámetro de regularización, que controla el equilibrio entre el modelo ajustado y la penalización. El efecto de esta regularización de los pesos es suavizar la función de error, ya que los pesos grandes están usualmente asociados a valores de salida altos.

Universidad de Cantabria

CLASIFICACION LINEAL PLANTEAMIENTO

Los problemas de clasificación consisten en asignar patrones dados a sus respectivas clases, o categorías, basándose en patrones representativos de cada clase.

x	y	c	x	y	c	x	y	c
0.272	0.987	0	0.524	0.196	1	0.629	0.232	1
0.506	0.371	1	0.750	0.594	1	0.818	0.295	1
0.526	0.900	0	0.005	0.972	0	0.112	0.318	0
0.932	0.968	1	0.641	0.926	0	0.351	0.813	0
0.369	0.938	0	0.827	0.617	1	0.739	0.706	1



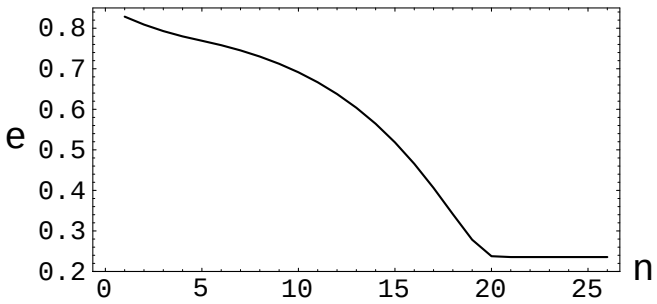
Se puede construir un perceptrón con estos puntos para obtener automáticamente el criterio de clasificación. Por ejemplo, si se considera un perceptrón con dos entradas, x_i y y_i, y una salida c_i con función de activación lineal

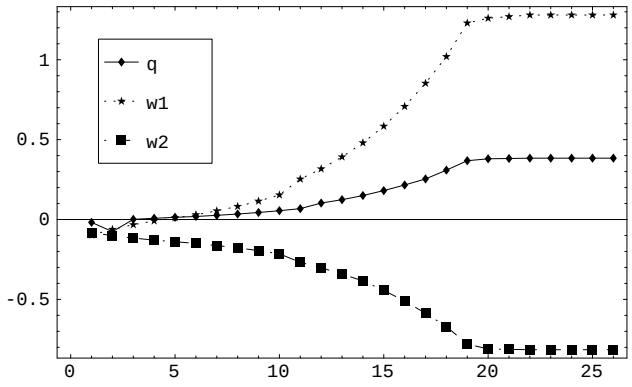
$$c_i = w_1 x_i + w_2 y_i + q, \tag{10}$$

Universidad de Cantabria

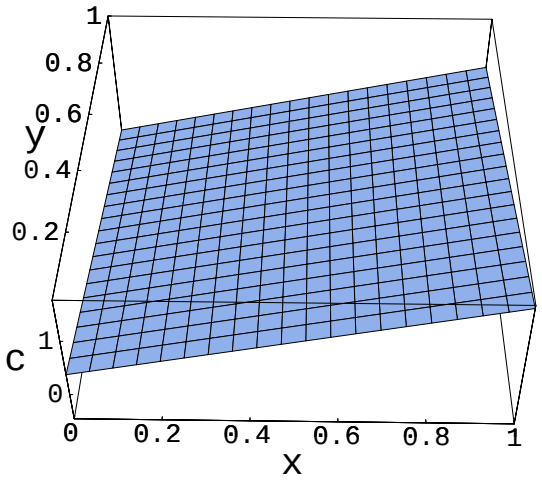
CLASIFICACION LINEAL APRENDIZAJE

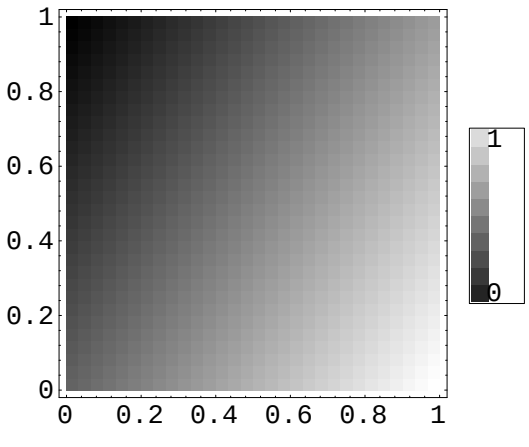
Descenso de gradiente con $\eta = 0.2$.





$$c_i = 1.28x_i - 0.815y_i + 0.384. \qquad (11)$$

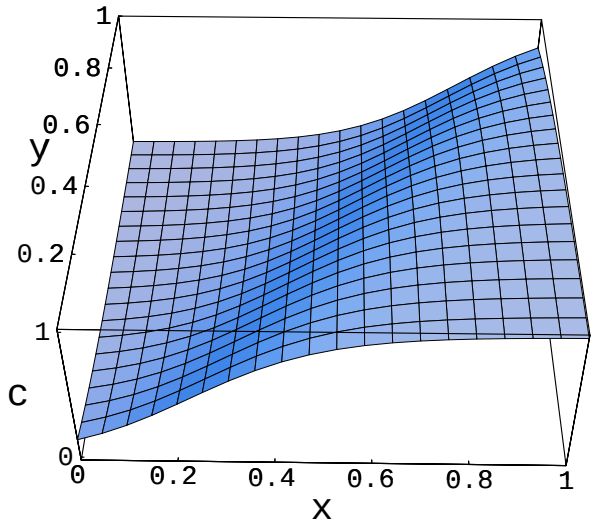


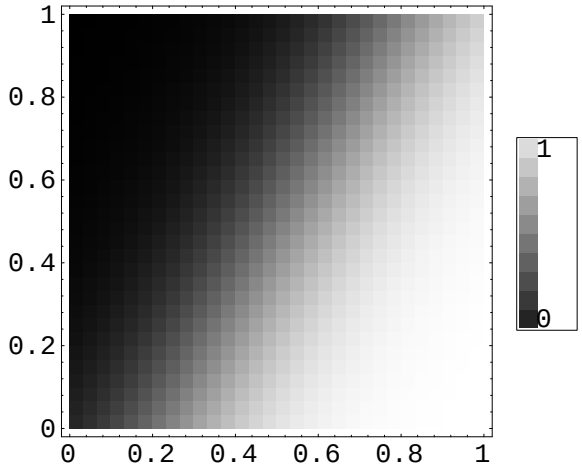


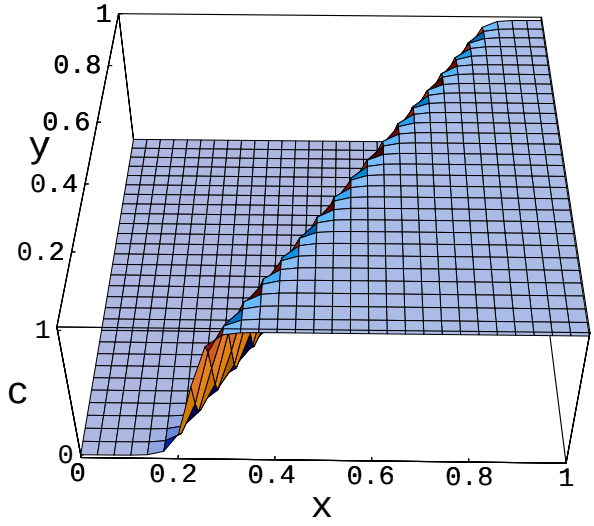
Universidad de Cantabria

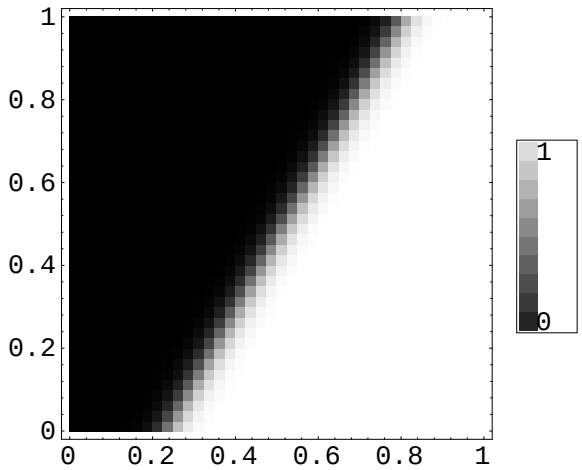
CLASIFICACION LINEAL APRENDIZAJE

Si se considera una función de activación sigmoidal $f(x) = (1 + e^{-x})^{-1}$ o de paso $\Theta(x)$ en (10).



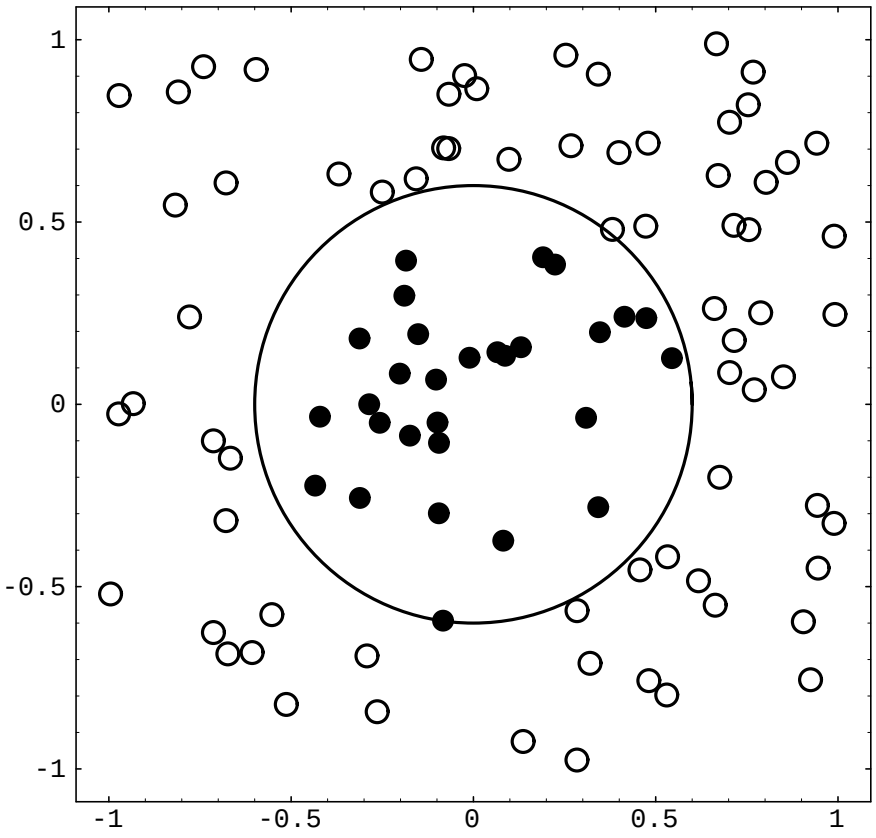






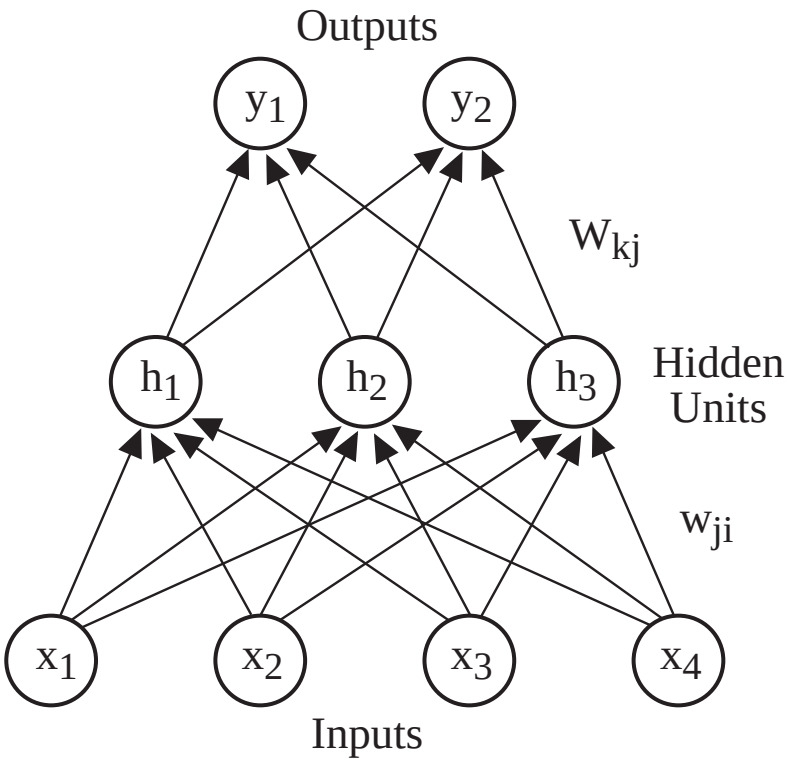
CLASIFICACION NO LINEAL

Supóngase que se tiene un conjunto de 100 puntos aleatorios en el intervalo $(-1, 1)$ clasificados en dos categorías: los que están dentro del círculo (puntos rellenos), y los que están fuera (puntos huecos).



Como estas dos categorías no son linealmente separables, entonces un perceptrón no puede obtener un criterio de clasificación apropiado.

EL PERCEPTRON MULTI-CAPA



En este caso los patrones de aprendizaje son un conjunto de inputs $\{a_i^p, i = 1, \ldots, n\}$ y sus correspondientes outputs $\{b_k^p, k = 1, \ldots, m\}$.

El método más popular de aprendizaje se denomina retro-propagación y está basado en minimizar la función de error mediante un método de descenso de gradiente.

Inicialmente se eligen valores aleatorios para los pesos.

Universidad de Cantabria

RETRO-PROPAGACION LA CAPA DE SALIDA

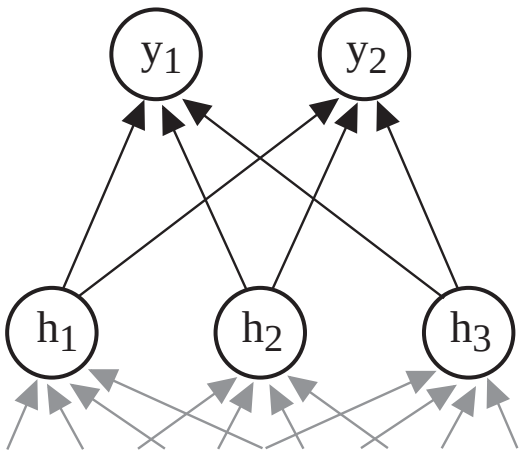
Los pesos de correspondientes a las neuronas de la capa de salida $y_i = f(Y_i)$ son modificados considerando los valores de las neuronas ocultas $h_i = f(H_i)$. En este caso, la función de error es

$$\begin{aligned} E(w) &= \frac{1}{2}\sum_{p,k}(b_k^p - \hat{b_k^p})^2 \\ &= \frac{1}{2}\sum_{p,k}(b_k^p - f(\sum_j W_{kj}\, f(\sum_i w_{ji}\, x_i^p)))^2. \end{aligned}$$

Entonces

$$\Delta W_{kj} = -\eta\frac{\partial E}{\partial W_{kj}} = \frac{\partial E}{\partial \hat{b_k^p}}\frac{\partial \hat{b_k^p}}{\partial \hat{B_k^p}}\frac{\partial \hat{B_k^p}}{\partial W_{kj}}$$

$$\Delta W_{kj} = \eta\, h_j^p\, \delta_k^p, \quad \text{where} \quad \delta_k^p = (b_k^p - \hat{b_k^p}) f'(B_k^p)$$



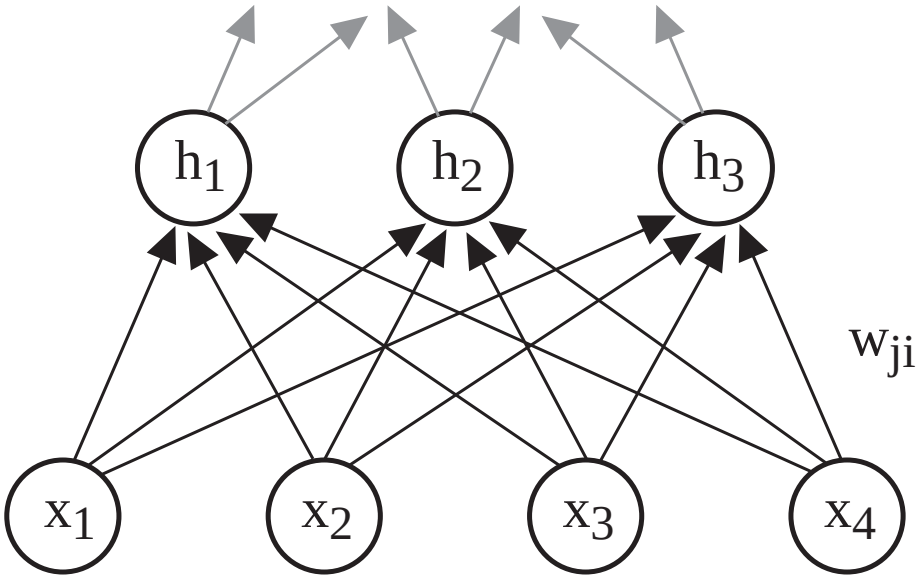
Universidad de Cantabria

RETRO-PROPAGACION
LA CAPA OCULTA

Los pesos de correspondientes a las neuronas de la capa oculta $h_i = f(H_i)$ son modificados considerando los valores de las neuronas de entrada $x_i = f(X_i)$ y los de las neuronas de salida $y_i = f(Y_i)$.

$$\Delta w_{ji} = -\eta \frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial \hat{b}_k^p} \frac{\partial \hat{b}_k^p}{\partial \hat{B}_k^p} \frac{\partial \hat{B}_k^p}{\partial h_j^p} \frac{\partial h_j^p}{\partial H_j^p} \frac{\partial h_j^p}{\partial w_{ji}},$$

$$\Delta w_{kj} = \eta \, a_i^p \, \delta_k^p \, \psi_j^p \quad \text{where} \quad \psi_j^p = \sum_k \delta_k^p \, W_{kj} \, f'(H_j^p).$$



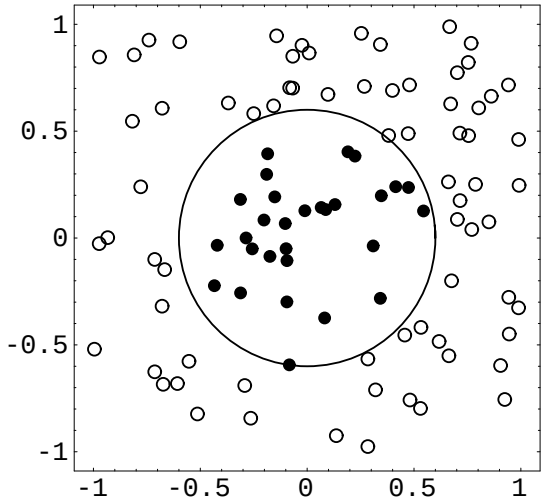
Universidad de Cantabria

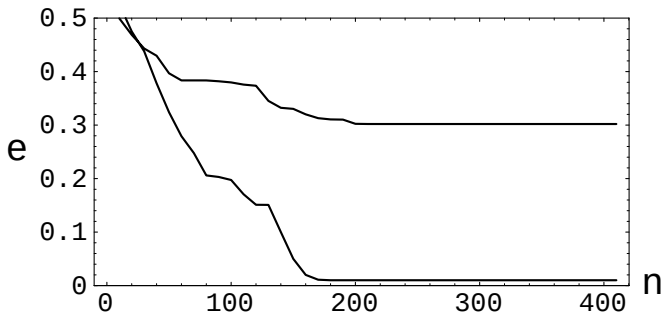
RETRO-PROPAGACION ALGORITMO

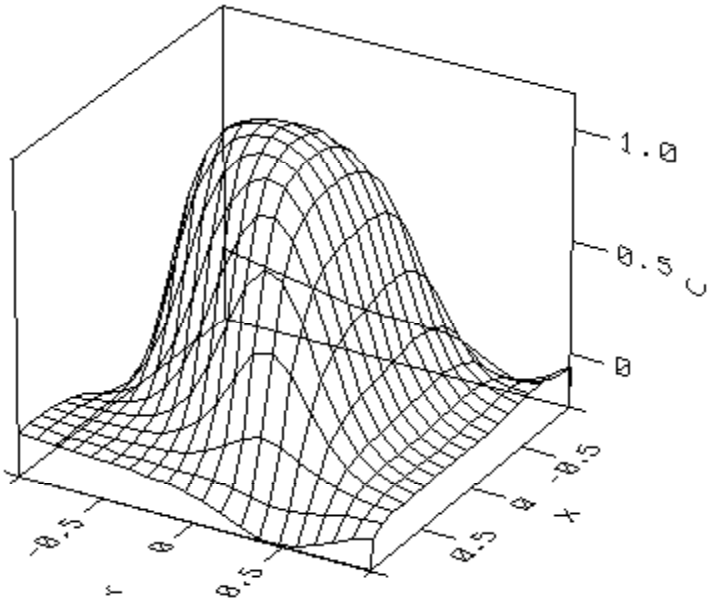
1. Inicializar los pesos con valores arbitrarios.
2. Elegir un patrón p y propagarlo hacia delante. Con ello obtenemos h_j^p y b_k^p (outputs de las capas oculta y de salida).
3. Calcular los errores de salida:
$$\delta_k^p = (b_k^p - \hat{b_k^p}) f'(B_k^p) = (b_k^p - \hat{b_k^p}) \hat{b_k^p} (1 - \hat{b_k^p})$$
4. Calcular los errores de la capa oculta:
$$\psi_j^p = \sum_k \delta_k^p \, W_{kj} \, f'(H_j^p) = \sum_k \delta_k^p \, W_{kj} \, h_j^p (1 - h_j^p)$$
5. Calcular:
$$\Delta W_{kj} = \eta \, h_j^p \, \delta_k^p,$$
y
$$\Delta w_{kj} = \eta \, a_i^p \, \delta_k^p \, \psi_j^p$$
y actualizar los pesos.
6. Repetir los pasos anteriores para cada patrón.

CLASIFICACION NO LINEAL

Perceptrón multicapa 2 : 5 : 1





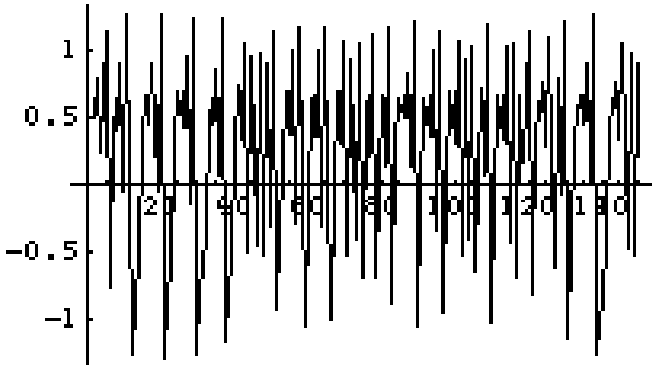


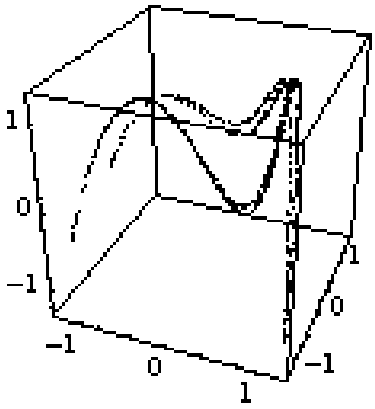
Universidad de Cantabria

MAPAS CAOTICOS: EL MAPA DE HENON

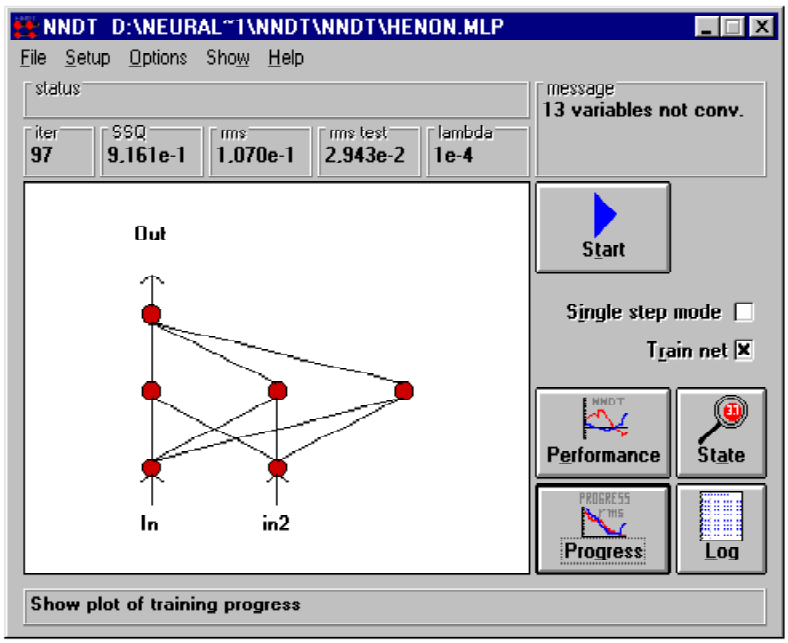
El mapa de Henon es uno de los ejemplos más ilustrativos de sistemas simples con dinámica compleja (caos determinista).

$$x_{n+1} = 1.0 - 1.4\,x_n^2 + 0.3\,x_{n-1}$$



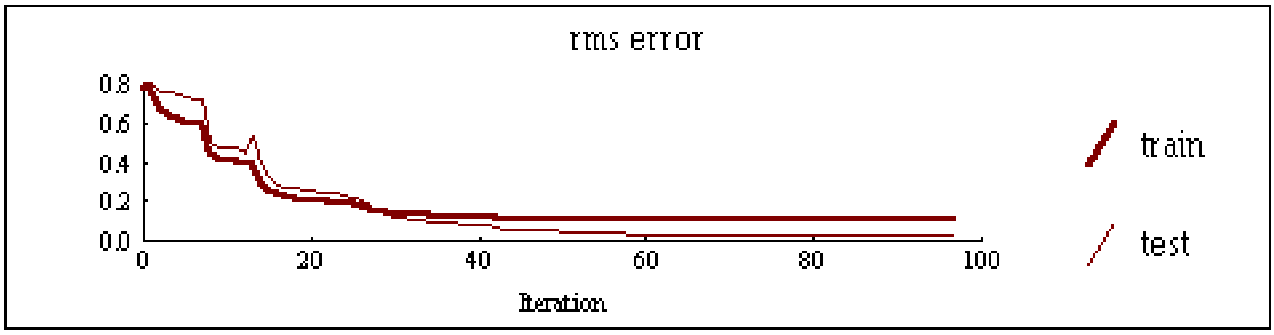


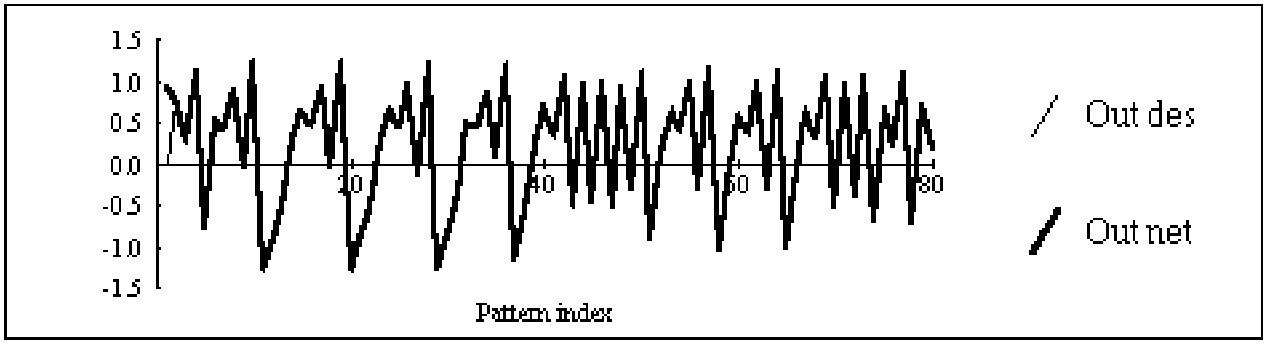
Para aproximar este mapa se utiliza una red neuronal 2:3:1 (la salida es x_n y las entradas x_{n-1} y x_{n-2}).



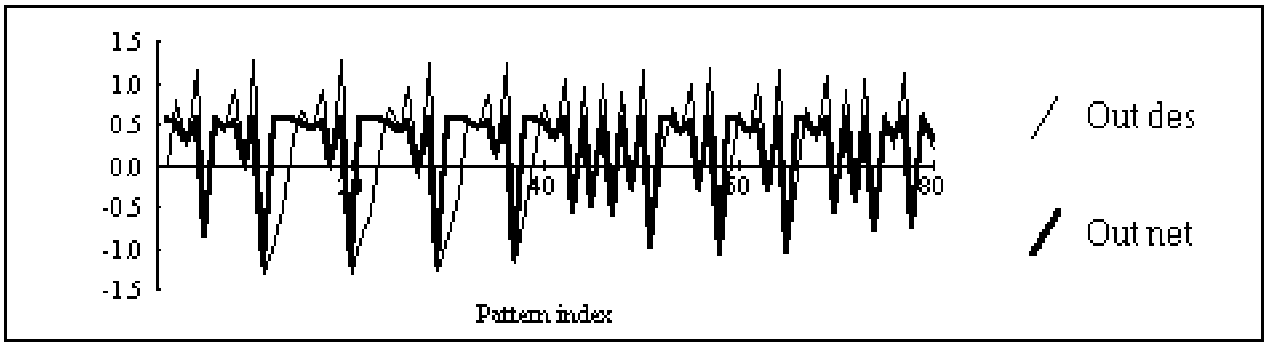
Universidad de Cantabria

EL MAPA DE HENON
RED NEURONAL





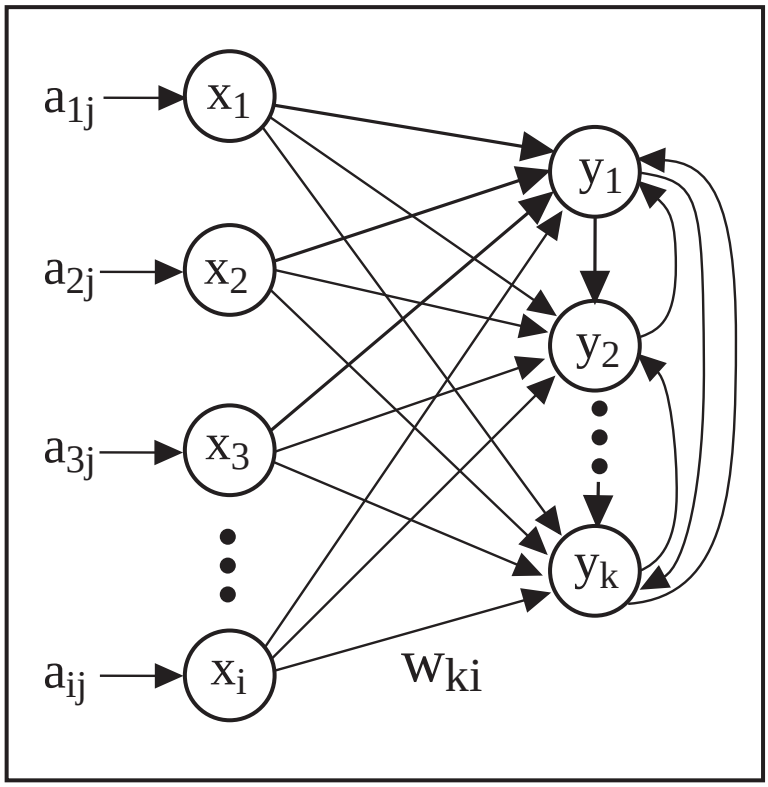
En algunos casos el proceso de optimización no converge al mínimo global, sino a uno local. Esto es debido al gran número de parámetros involucrado en la red y a las características de los modelos de aprendizaje.



Universidad de Cantabria

REDES COMPETITIVAS

Las redes competitivas son muy utilizadas para detectar automáticamente grupos, o categorías, dentro de los datos disponibles. Cada patrón de entrada representa un punto en el espacio de configuración (el espacio de entradas) donde se quieren obtener clases. Para ello, la capa de salida contiene tantas neuronas como clases, o categorías, como se quieran obtener.



REDES COMPETITIVAS APRENDIZAJE

Este tipo de arquitectura se entrena normalmente con un algoritmo consistente en *seleccionar la ganadora* ("winner takes all"), por lo que sólo son actualizados los pesos asociados a la neurona de mayor salida (la ganadora) para un patrón dado.

Considérense los datos de entrenamiento consistentes en un conjunto de patrones de entrada $(a_{1j}, \ldots, a_{nj}), j = 1, \ldots, m$.
Se empieza con pequeños valores aleatorios para los pesos. A continuación, se aplica el patrón $(a_{1j}, \ldots, a_{nj})$, y se selecciona la unidad salida ganadora, sea y_k. Los pesos asociados con esta neurona son modificados de acuerdo con

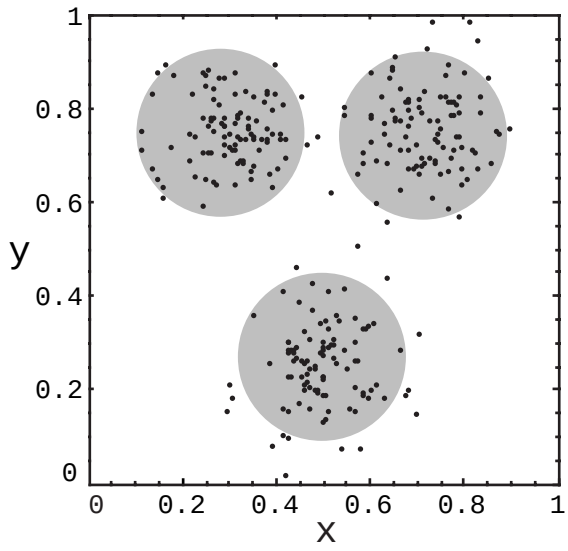
$$\Delta w_{ki} = \eta(a_{ij} - w_{ki}). \qquad (12)$$

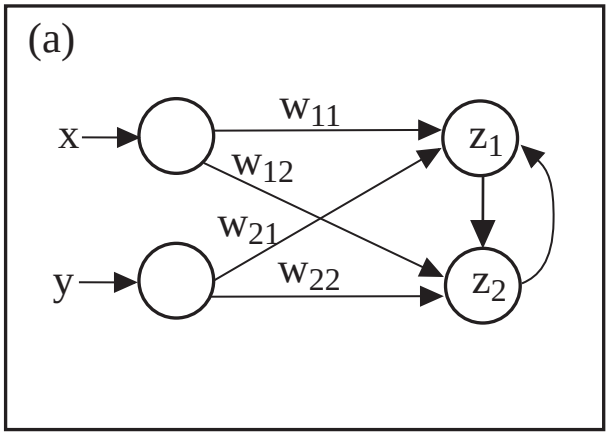
El efecto es mover el vector peso $(w_{k1}, \ldots, w_{kn})$ directamente hacia $(a_{1j}, \ldots, a_{nj})$.

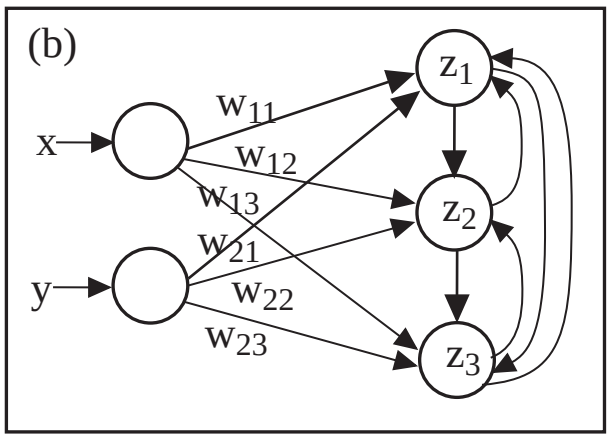
Nótese que en el caso de clasificación supervisada, usando perceptrones multi-capa, el usuario proporciona ejemplos de las diferentes categorías.

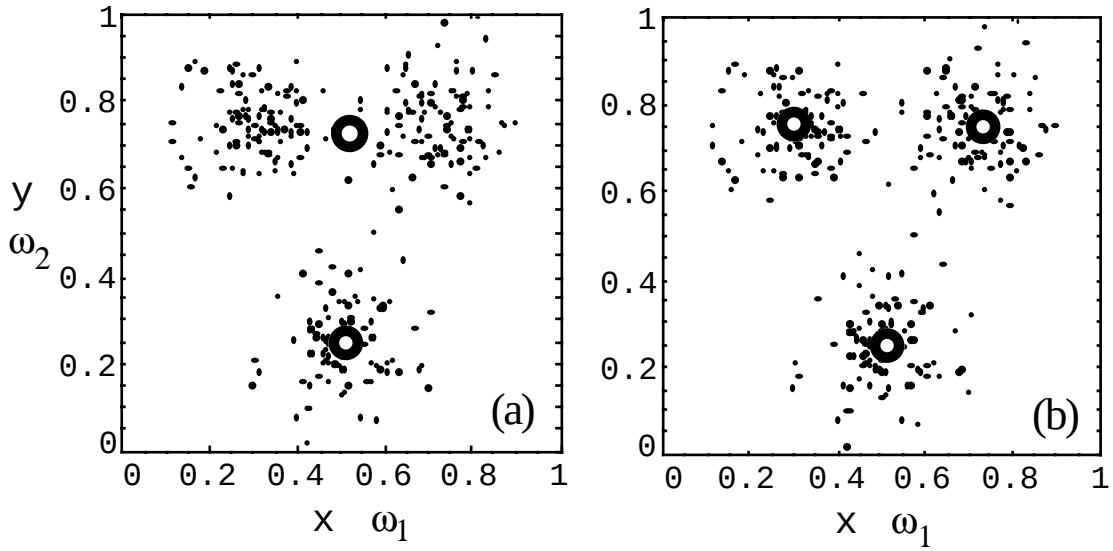
Universidad de Cantabria

REDES COMPETITIVAS
EJEMPLO









Introducción a las Redes Neuronales

Universidad de Cantabria

EJERCICIOS

1. Utilizar un perceptrón con dos unidades de entrada y una de salida para modelizar las funciones lógicas AND y OR. ¿Se puede modelizar también una puerta XOR?

2. Considerar el fichero de datos "sincos.dat" que contiene
   tres colomnas de datos $(x, sin(x), cos(x))$ e intentar aproximarlo con un perceptrón multicapa 2:?:1. Probar varios valores de los parámetros de aprendizaje, η, y momento, α y comparar la convergencia en los distintos casos. ¿Que valores recomiendas para este problema?

3. Considerar la función no lineal

$$y(x) = 20e^{-8.5\,x}(Ln(0.9\,x + 0.2) + 1.5).$$

   Generar un fichero con 50 pares $(x, y(x))$ en el intervalo $(0, 1)$ para entrenar un perceptrón multicapa 1:8:1. Generar también un fichero con otros 50 puntos distintos para comprobar la validez de la aproximación.