

Relación 6 - Redes neuronales

Cuestiones

Cuestión 1. Verdadero o Falso. Si tomamos la función identidad como función de activación, entonces el perceptrón simple es capaz de aprender la función booleana binaria f_{AND} ($f_{AND}(1, 1) = 1$, $f_{AND}(0, 1) = 0$, $f_{AND}(1, 0) = 0$, $f_{AND}(0, 0) = 0$).

Si la respuesta es **Verdadero** tienes que dar un perceptrón que caracterice a f_{AND} . Si la respuesta es **Falso**, tienes que dar las razones de por qué no puede existir tal perceptrón.

¿Verdadero o falso?:

Cuestión 2. Verdadero o Falso. El perceptrón simple es capaz de representar la fórmula lógica $(A \vee B) \wedge \neg C$ usando la función *umbral* como función de activación.

$$umbral(x) = \begin{cases} 1 & \text{si } x > 0 \\ 0 & \text{si } x \leq 0 \end{cases}$$

Si la respuesta es **Verdadero** tienes que dar los pesos del perceptrón. Si la respuesta es **Falso**, tienes que dar las razones de por qué no puede existir tal perceptrón.

¿Verdadero o falso?:

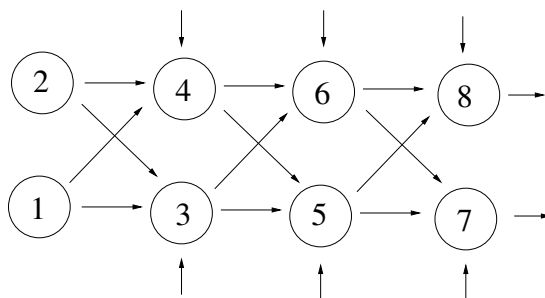
Problemas

Ejercicio 1. Consideremos un perceptrón con pesos $w_0 = w_1 = w_2 = 0,5$, función sigmoide como función de activación y factor de aprendizaje $\eta = 0,8$. Sea D el conjunto de entrenamiento $D = \{\langle(1, 0), 1\rangle, \langle(0, 1), 0,5\rangle\}$. Se pide calcular la variación que se produce en el error cuadrático cometido sobre el conjunto de entrenamiento D tras un paso del algoritmo de descenso por gradiente.

Ejercicio 2. Considera un perceptrón con función de activación sigmoide y pesos $\vec{w}' = (0,1, 0,2, 0,3)$ y un conjunto de entrenamiento $D = \{E_1, E_2\}$ con $E_1 = \langle(0,5, 0,5), 0,3\rangle$ y $E_2 = \langle(0,4, 0,7), 0,2\rangle$. [Nota: Para hacer los cálculos tomaremos los cuatro primeros decimales.]

- Calcula el error cuadrático del perceptrón con pesos \vec{w}' sobre el conjunto de entrenamiento D .
- Calcula el valor del gradiente de la función anterior $\nabla E(\vec{w})$ para el vector de pesos \vec{w}' .

Ejercicio 3. Consideremos la siguiente red neuronal

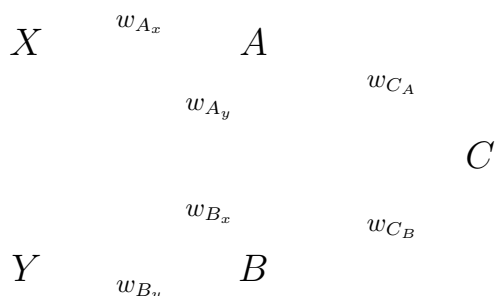


Representaremos por w_{ij} el peso asociado a la sinapsis desde la neurona i a la neurona j y por w_{0j} el peso asociado a la entrada virtual $a_0 = -1$ de la neurona j . Consideremos un factor de aprendizaje $\eta = 0.1$ y la función sigmoide

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

como función de activación. Consideraremos todos los pesos iniciales de la red neuronal valen 1.0. ¿Cuánto vale el peso w_{06} tras dar un paso del **algoritmo de retropropagación** asociado al ejemplo $\langle (1, 1), (1, 1) \rangle$? Detallar todos los pasos del algoritmo necesarios para obtener el valor pedido. Nótese que en el ejemplo el valor suministrado a cada neurona de entrada es 1 y cada una de las neuronas de salida tienen como salida 1. [**Nota:** Para hacer los cálculos tomaremos los cuatro primeros decimales.]

Ejercicio 4. Consideremos la red neuronal de la siguiente figura



Se pide usar un razonamiento geométrico para encontrar los nueve pesos w_{0A} , w_{XA}, w_{YA}, w_{0B} , w_{XB}, w_{YB}, w_{0C} , w_{AB}, w_{AC} de manera que la red se comporte como la función XOR_2 usando la función umbral como función de activación. **Nota:** Usar un razonamiento geométrico con dos rectas paralelas para separar regiones del plano.

Ejercicio 5. Consideremos un perceptrón con pesos $\vec{w} = (w_0, w_1, w_2)$, $w_0 = w_1 = w_2 = 0.8$, la función sigmoide como función de activación y un conjunto de entrenamiento $D = \{E_1, E_2\}$ con $E_1 = \langle (0, 1), 1 \rangle$ y $E_2 = \langle (1, 0), 0 \rangle$.

- Calcular el error cuadrático $E(\vec{w})$ que se comete al considerar este perceptrón sobre ese conjunto de entrenamiento.
- Devolver el valor del peso w_2 tras una actualización usando el algoritmo de entrenamiento del perceptrón por DESCENSO POR EL GRADIENTE, con un factor de aprendizaje $\eta = 0.2$. Sólo hay que dar los cálculos necesarios para esa actualización.

Ejercicio 6. Explicar cómo se usaría una red neuronal para obtener un reconocedor de letras escritas a mano. Describir con precisión qué estructura de red usarías, qué representarían tanto la entrada como la salida de la red, en qué consistiría un conjunto de entrenamiento y cómo se podría obtener, y cómo se buscarían unos pesos adecuados para la red ¿Qué entendemos en este contexto por “pesos adecuados”?

Ejercicio 7. Diseñar un perceptrón simple con n valores de entrada y función umbral de activación que sirva para calcular la función MAYORIA-SIMPLE; esta función recibe n entradas (cada una puede ser un 0 o un 1) y devuelve como salida un 1 si hay estrictamente más 1s que 0s, o 0 en caso contrario.

Ejercicio 8. Demostrar geoméricamente que un perceptrón simple no puede calcular la función XOR. Construir una red neuronal (con función umbral como función de activación) que sí la calcule. *Indicación:* Tener en cuenta que XOR se puede obtener mediante AND y OR, y que estas dos funciones si pueden ser calculadas por perceptrones.

Ejercicio 9. La función PARIDAD es aquella que recibiendo n bits, devuelve 1 si hay un número par de ellos igual a 1, y 0 en caso contrario ¿Se puede calcular la función PARIDAD mediante un perceptrón simple? ¿Y mediante una red neuronal con una capa oculta y función umbral como función de activación? *Indicación:* incluir n neuronas en la capa intermedia, donde cada neurona i de la capa intermedia se debe activar con un 1 si hay más de i entradas iguales a 1.

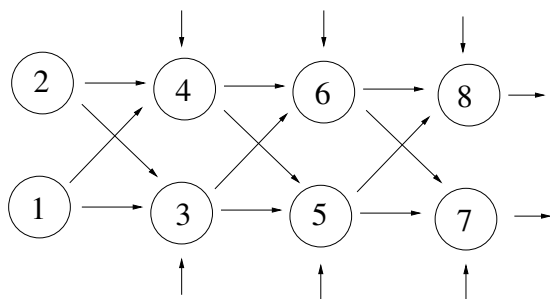
Ejercicio 10. Sea f una función de $R \times R$ en $\{-1, 1\}$ tal que $f(-1, 1) = f(0, 0) = 1$ y $f(-1, 0) = f(0, 1) = -1$. Supongamos que con ese conjunto de ejemplos aplicamos el algoritmo de entrenamiento del perceptrón simple bipolar ¿Será el algoritmo capaz de encontrar los pesos adecuados para que la unidad bipolar correspondiente calcule correctamente los cuatro ejemplos anteriores? En el caso de usar el algoritmo de entrenamiento de la regla delta para encontrar un perceptrón simple con función activación diferenciable, ¿hacia qué converge el vector de pesos que va construyendo el algoritmo?

Ejercicio 11. Aplica el ALGORITMO DE ENTRENAMIENTO DE DESCENSO POR EL GRADIENTE hasta obtener la primera actualización del peso w_2 al problema de aprendizaje del perceptrón con $\eta = 0,1$, la función sigmoide como función de activación y el conjunto de aprendizaje

	x_1	x_2	x_3	y
e_1	0,7	0,2	0,1	0,3
e_2	0,3	0,5	0,2	0,8
e_3	0,1	0,1	0,8	0,6

Tomar como pesos iniciales $w_0 = 0,1$, $w_1 = 0,1$, $w_2 = 0,1$ y $w_3 = 0,1$.

Ejercicio 12. Consideremos la siguiente red neuronal, un factor de aprendizaje $\eta = 0.1$ y la función sigmoide como función de activación. Representaremos por



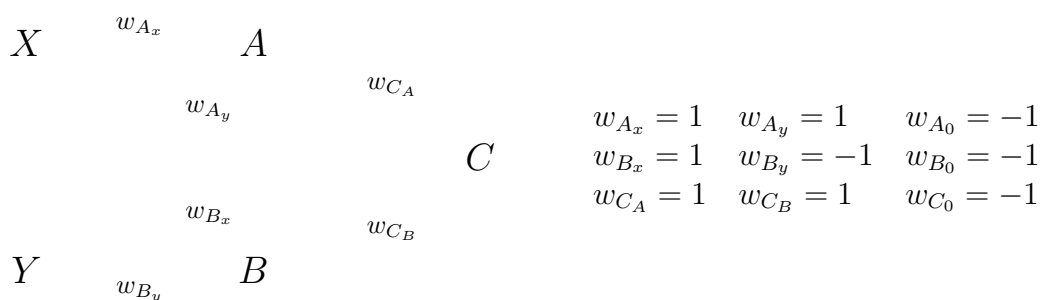
w_{ij} el peso asociado a la sinapsis desde la neurona i a la neurona j y por w_{0j} el peso asociado a la entrada virtual de la neurona j . Consideraremos todos los pesos iniciales de la red neuronal valen 1.0. ¿Cuánto vale el peso w_{36} tras dar un paso del **algoritmo de retropropagación** asociado al ejemplo $\langle (0,5, 0,5), (0,8, 0,8) \rangle$? Detallar todos los pasos del algoritmo. [Nota: Para hacer los cálculos tomaremos 4 decimales.]

Ejercicio 13.

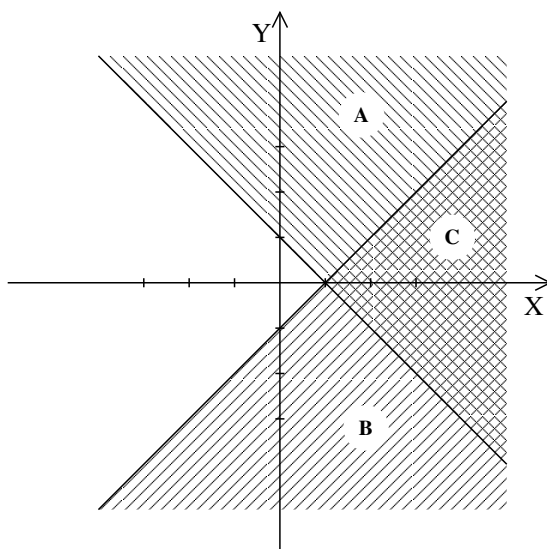
Supongamos que entrenamos un perceptrón cuya función de activación es la identidad, usando la regla Delta como algoritmo de entrenamiento. Si el conjunto de entrenamiento es linealmente separable ¿Está garantizado que en algún momento encontraremos unos pesos que hagan que el perceptrón clasifique correctamente a cada uno de los ejemplos del conjunto de entrenamiento?

Ejercicio 14. Consideremos la red neuronal de la siguiente figura formada por unidades

bipolares (considerando $x_0 = -1$ en todos los casos):



En esta red neuronal las unidades de la capa oculta son, consideradas de forma aislada, perceptrones simples con dos entradas; cada uno de los cuales clasifica una región del plano XY (ver figura). La unidad C implementa una conjunción de las señales de las unidades A y B que, gráficamente, se corresponde con la intersección de las regiones clasificadas por las unidades A y B (ver figura).



Teniendo en cuenta esta interpretación geométrica de la red neuronal, se pide:

- Determinar los valores de los pesos de la red neuronal para que clasifique como 1 la siguiente región geométrica: $\{(x, y) : y \leq 1 + x \text{ o } y \leq 1 - x\}$
- Determinar los valores de los pesos de la red neuronal para que clasifique como 1 la siguiente región geométrica: $\{(x, y) : -1 \leq x - y \leq 1\}$
- Calcular la función XOR con una red neuronal como la de la figura, utilizando un razonamiento de tipo geométrico (intersección o unión de regiones del plano).

Ejercicio 15.

Sea f una función de $R \times R$ en $\{-1, 1\}$. Consideremos el problema de aprender f mediante un perceptrón simple bipolar, para ello se tiene el siguiente conjunto de entrenamiento:

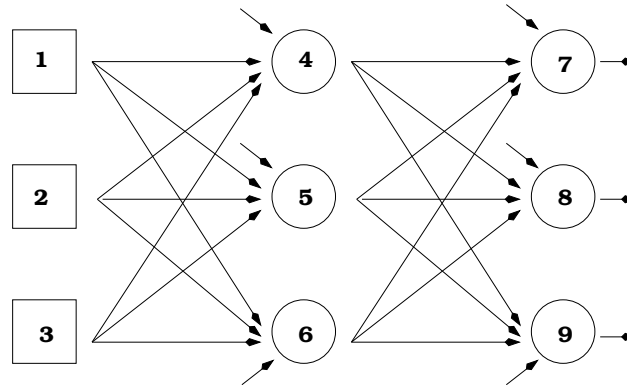
	Entradas	Salida
E_1	(2, 0)	1
E_2	(0, 0)	-1
E_3	(2, 2)	1
E_4	(0, 1)	-1
E_5	(1, 1)	1
E_6	(1, 2)	-1

Aplicar el algoritmo de entrenamiento del perceptrón simple bipolar con el conjunto de entrenamiento anterior, considerando los ejemplos en el mismo orden en que aparecen, hasta que se clasifiquen correctamente todos los ejemplos. Tomar 0 como valor inicial para los pesos y 0,1 como factor de aprendizaje.

Con los pesos aprendidos, ¿qué salida se obtiene para las siguientes entradas: (0, 2), (1, 0) y (2, 1)?

Ejercicio 16.

Consideremos la siguiente red neuronal, que utiliza la función sigmoide σ como función de activación:



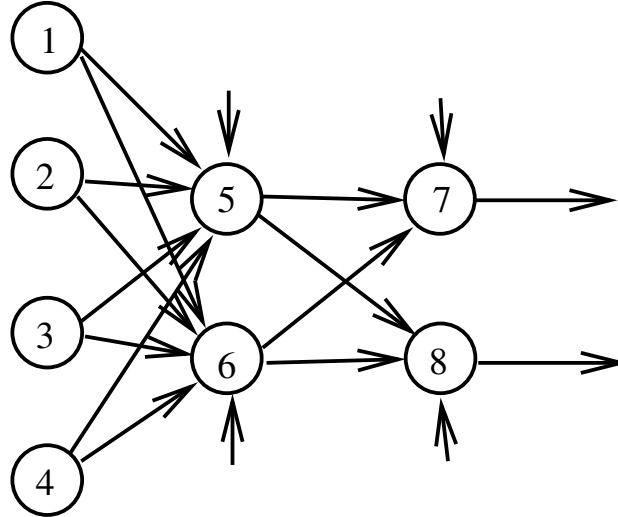
Se pide:

1. Calcular, detallando las fórmulas utilizadas¹, la salida de la red para la entrada $\vec{x} = \langle 1, 0, 0 \rangle$, suponiendo que todos los pesos de la red son 0,5.
2. Consideremos ahora otro ejemplo $\vec{x} = \langle x_1, x_2, x_3 \rangle$ con salida esperada $\vec{y} = \langle 1, 0, 0 \rangle$. Supongamos que en todas las neuronas de la capa oculta la salida obtenida con ese ejemplo es $a_i = 0,2$, y que para las neuronas de la capa de salida se tiene $a_i = 0,5$. Realizar los cálculos necesarios siguiendo el algoritmo de retropropagación hasta hallar el valor del error Δ_4 . ¿Cuáles son las fórmulas para actualizar los pesos $w_{0,4}$, $w_{4,7}$, $w_{4,8}$ y $w_{4,9}$?
3. Supongamos que esta red se ha diseñado para clasificar unos datos que se pueden representar mediante vectores $\vec{x} \in \mathbb{R}^3$. Supongamos que ya hemos implementado algún algoritmo de aprendizaje usando un conjunto de entrenamiento.
 - Explicar el proceso que hay que seguir para clasificar un ejemplo nuevo $\vec{x} = \langle x_1, x_2, x_3 \rangle$.

¹Nota: $\sigma(0) = 0,5$, $\sigma(0,25) = 0,44$

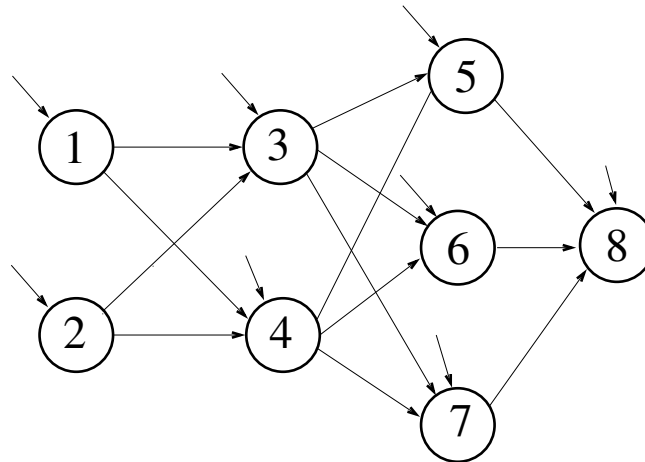
- Si nos interesase clasificar los ejemplos en sólo dos categorías ¿Realizarías modificaciones sobre la red? ¿Cuáles?

Ejercicio 17. Sea una red neuronal con la siguiente estructura en la que se usa el sigmoide como función de activación:



Supongamos dado un ejemplo (x_1, x_2, x_3, x_4) con salida esperada (y_7, y_8) . Supongamos también que ya hemos calculado la salida a_i en cada unidad $i = 1, \dots, 8$. Según el algoritmo de retropropagación: ¿cuáles son las fórmulas para calcular los errores Δ_8 , Δ_7 y Δ_6 , respectivamente? ¿y las fórmulas para actualizar los pesos $w_{6,7}$ y $w_{6,8}$, respectivamente?

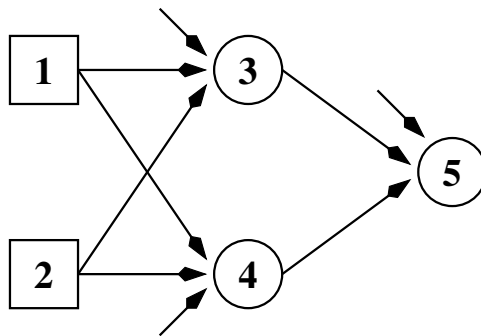
Ejercicio 18. Considérese la red neuronal con la siguiente estructura y con la función sigmoide como función de activación:



Supongamos dado un ejemplo del conjunto de entrenamiento, con entrada (x_1, x_2) y salida esperada y . Se pide mostrar las fórmulas que se usan en una iteración del algoritmo de retropropagación correspondiente a este ejemplo, tanto las de la fase de propagación hacia adelante como las de la fase de propagación hacia atrás, *dejando claro en el orden en el que se producen los cálculos*. ¿Cuál es el objetivo general del algoritmo de retropropagación?

Ejercicio 19.

Consideremos la siguiente red neuronal con función de activación lineal $g(x) = x$ en todas sus unidades:



Sean los ejemplos de entrenamiento $(\vec{x}_1, y_1) = (\langle 0, 0 \rangle, 1)$ y $(\vec{x}_2, y_2) = (\langle 1, 1 \rangle, 0)$. Aplicar una iteración del algoritmo de retropropagación para cada uno de estos ejemplos, tomando 0,1 como factor de aprendizaje y todos los pesos iniciales iguales a 0,5.

Ejercicio 20.

Una *nariz electrónica* analiza mediante sensores los vapores procedentes de determinadas sustancias y las clasifica a partir de la información cuantitativa obtenida. Supongamos que utilizamos 16 sensores para identificar cuatro tipos de vino tinto: Cabernet, Merlot, Syraz y Tempranillo.

Describir en detalle cómo podríamos usar una red neuronal para abordar este problema: en que consistiría un conjunto de entrenamiento y cómo se codificarían los ejemplos, la estructura de la red, el algoritmo de entrenamiento usado y de qué manera se usaría la red una vez entrenada, para identificar un nuevo vino tinto.

Ejercicio 21.

Supongamos que una empresa de televisión por cable quiere diseñar un sistema automatizado para recomendar a sus clientes uno de sus cinco canales temáticos, en función de sus preferencias, que se tratan de adivinar en función de una encuesta con 20 preguntas. ¿Cómo diseñarías el sistema usando una red neuronal? ¿Qué estructura tendría esta red neuronal? ¿Cómo entrenarías la red y cuál sería tu conjunto de entrenamiento? Una vez entrenada, ¿cómo usarías la red obtenida para recomendar un canal temático a un nuevo cliente?

Ejercicio 22.

Supongamos que queremos diseñar un sistema de “anuncios personalizados” para los usuarios de un portal web. En este portal web hay veinte secciones temáticas y la compañía maneja cuatro posibles perfiles publicitarios, cada uno de ellos apropiado a un usuario en función de los temas que más le interesan. Describir cómo se usaría una red neuronal para implementar este sistema: estructura de la misma, conjunto de entrenamiento, aprendizaje de los pesos, uso de la red una vez entrenada...

Ejercicio 23.

Supongamos que queremos diseñar un sistema automatizado para reconocer el estado de ánimo de la gente observando la expresión de su cara. Por simplificar las cosas, supongamos que consideramos cuatro tipos distintos de estados de ánimo: alegre, triste, enfadado y neutro. Suponiendo que nuestro sistema dispone de una cámara que es capaz de obtener imágenes digitalizadas de la cara de una persona ¿cómo diseñarías el sistema usando una red neuronal? ¿en qué consistiría el aprendizaje de esa red?

Ejercicio 24.

Supongamos que tenemos un problema de aprendizaje en el que hay que aprender a clasificar elementos de R^n en dos categorías, para cierta dimensión $n > 0$. No se conoce

la manera exacta de clasificar, pero sí se conoce la clasificación de un conjunto finito de tuplas (es decir, un conjunto de entrenamiento).

- Dar un ejemplo de una aplicación real en el que se pueda plantear una situación como ésta.
- Si el conjunto de entrenamiento no es linealmente separable y el clasificador que se quiere aprender fuera un perceptrón simple, ¿cuántas unidades de entrada y cuántas de salida tendría? ¿qué función de activación emplearía? Una vez aprendido, ¿cómo se usaría para clasificar nuevas instancias?
- ¿Qué algoritmo usarías para aprender los pesos de dicho perceptrón? Justificar la respuesta.
- Para $n = 3$, calcular un paso del algoritmo anterior **correspondiente a un ejemplo positivo** (inventando los pesos de partida, el ejemplo y la tasa de aprendizaje). Por simplificar, considerar que la función de activación es la identidad, aunque no sea ésta la función elegida en el apartado anterior.

Supongamos ahora una situación análoga a la anterior pero con **cuatro categorías de clasificación**.

- Dar un ejemplo de una aplicación real en la que se pueda plantear dicha situación.
- Si el clasificador que se quiere aprender fuera ahora una red multicapa ¿cuántas unidades tendría en la capa de entrada y cuántas en la de salida? ¿qué función de activación usaría? Una vez aprendido, ¿cómo se usaría para clasificar nuevas instancias?
- Considerar en concreto $n = 3$ y una red con una capa oculta con dos unidades. En dicha situación, y usando el algoritmo de retropropagación para aprender sus pesos, describir las fórmulas (simbólicamente) que reflejan cómo se actualizan dichos pesos en **un paso del algoritmo correspondiente a un ejemplo de la tercera categoría**. Se piden tanto las fórmulas de la propagación hacia adelante (en cada unidad) como las de la propagación hacia atrás (en cada peso).