Sistemas de Inteligencia artificial
Perceptron

Rodrigo Ramele, Juliana Gambini,
Juan Santos, Paula Oseroff,
Eugenia Piñeiro, Santiago Reyes

2022

Redes neuronales artificiales

Introducción

Podríamos introducir el tema tratando de dar respuestas a algunas de estas preguntas:

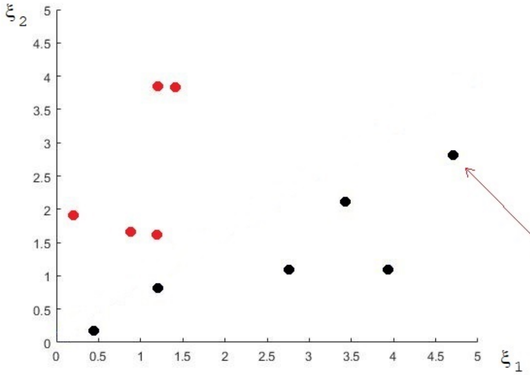
- ¿Cómo es una red neuronal biológica?
- ¿Cómo es una neurona biológica?
- ¿Cómo se formularon los primeros modelos de neurona?
- ...

Redes neuronales artificiales

Introducción

Pero vamos primero a presentar un problema que tiene la particularidad de ser fácil de entender.

Redes neuronales artificiales



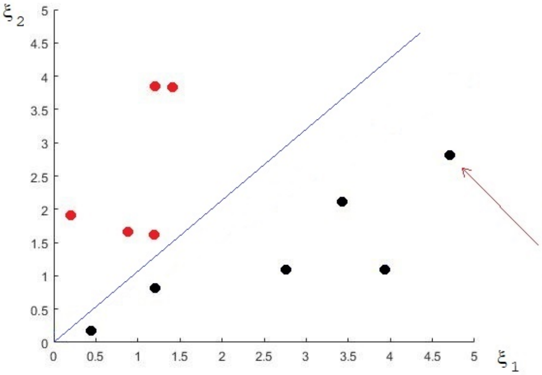
ξ_1	ξ_2	ζ
1.1946	3.8427	1
0.8788	1.6595	1
1.1907	1.6117	1
1.4180	3.8272	1
0.2032	1.9208	1
2.7571	1.0931	-1
4.7125	2.8166	-1
3.9392	1.1032	-1
1.2072	0.8132	-1
3.4799	1.9982	-1
0.4763	0.1020	-1

Redes neuronales artificiales

Introducción

- Podemos observar que podemos trazar una recta que separa los puntos marcados con rojo de los puntos marcados en negro.
- Es decir, separa los datos en dos clases.
- Si los puntos pertenecieran a R^n, en vez de una recta tendríamos un hiperplano.
- En el caso particular que n fuera 3, entonces en vez de una recta tendríamos un plano.

Redes neuronales artificiales



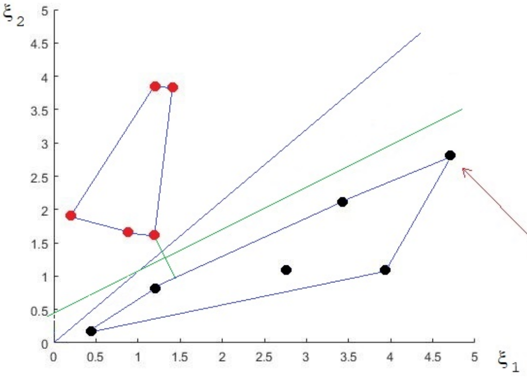
ξ_1	ξ_2	ζ
1.1946	3.8427	1
0.8788	1.6595	1
1.1907	1.6117	1
1.4180	3.8272	1
0.2032	1.9208	1
2.7571	1.0931	-1
4.7125	2.8166	-1
3.9392	1.1032	-1
1.2072	0.8132	-1
3.4799	1.9982	-1
0.4763	0.1020	-1

Redes neuronales artificiales

Introducción

- ¿Cómo encontramos la recta (en general, el hiperplano)?
- Es decir, ¿cómo podría ser el método que nos permita encontrar la recta (en general, el hiperplano)?
- Una posibilidad es obteniendo, para cada clase, el convexo cerrado para todos los puntos de la clase. Luego trazamos la recta más corta que une ambos convexos y trazamos una perpendicular a dicha recta que pase por el medio de la misma.

Redes neuronales artificiales



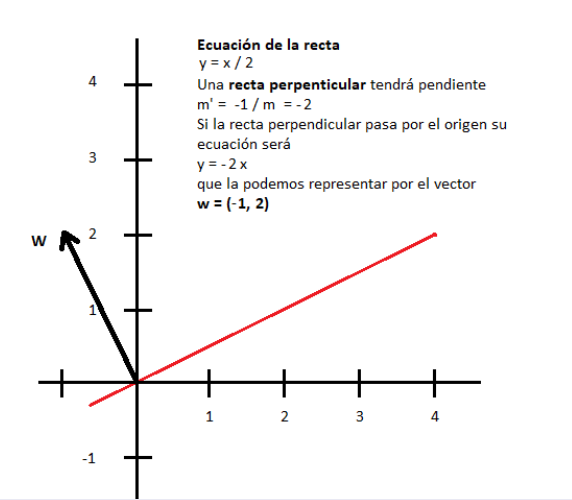
ξ_1	ξ_2	ζ
1.1946	3.8427	1
0.8788	1.6595	1
1.1907	1.6117	1
1.4180	3.8272	1
0.2032	1.9208	1
2.7571	1.0931	-1
4.7125	2.8166	-1
3.9392	1.1032	-1
1.2072	0.8132	-1
3.4799	1.9982	-1
0.4763	0.1020	-1

Redes neuronales artificiales

Introducción

- Existen otros métodos para obtener un hiperplano de separación pero, por ahora, nos focalizaremos en caracterizar cómo representar dicho hiperplano.
- Consideremos al hiperplano representado por su normal que llamaremos w.
- En R^2 dicha normal es un vector (w_1, w_2).
- Grafiquemos a w en la gráfica que teníamos.

Redes neuronales artificiales

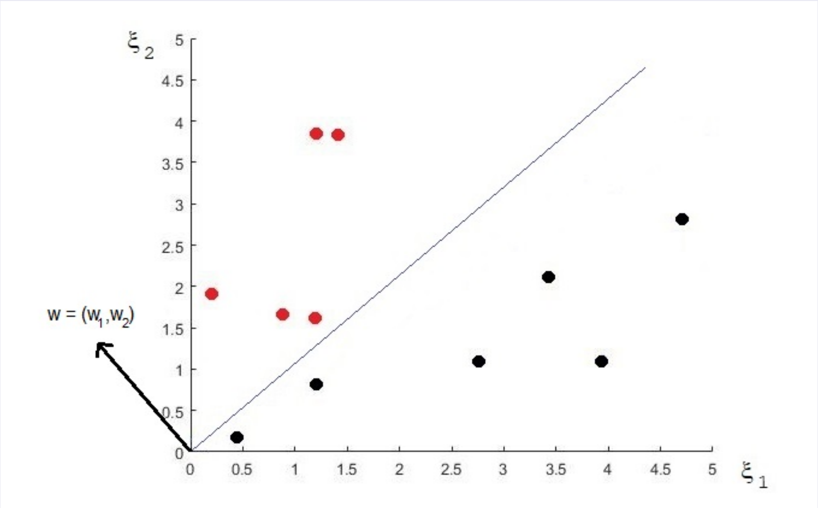


Redes neuronales artificiales

Introducción

- Grafiquemos a w en la gráfica que teníamos.

Redes neuronales artificiales

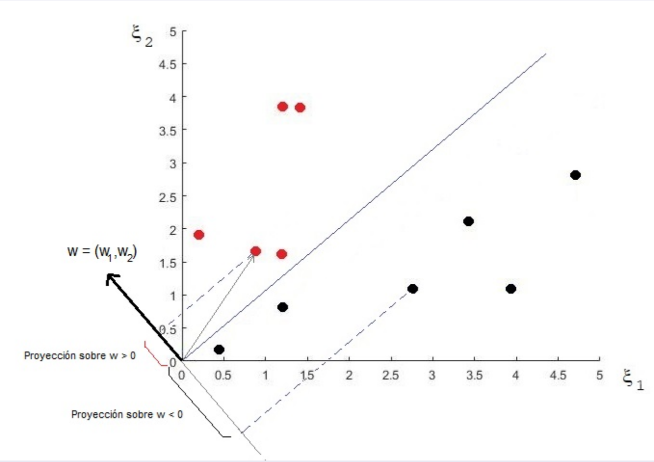


Redes neuronales artificiales

Introducción

- Ahora, tomemos un punto de la clase roja y consideremos el vector que comparte el origen de w y llega al punto en cuestión.
- Y luego otro de la clase negra.

Redes neuronales artificiales



Redes neuronales artificiales

Introducción

- Vemos que la proyección del vector del punto rojo sobre w es positiva, mientras que la proyección del vector del punto negro es negativa.
- Y lo mismo que ocurre para el punto rojo elegido, ocurre para el resto de los puntos rojos.
- Lo mismo para los puntos negros.

Redes neuronales artificiales

Introducción

- Sean dos vectores a,b en R^n,
  la proyección de a sobre b es $||a||cos\alpha$
- Por otro lado, el **producto interno** entre vectores a y b es
  $$a.b = \sum_{i=1}^{n} a_i b_i,$$
  donde n es la dimensión de los vectores a y b.
- Pero también (y aquí viene lo bueno...) el **producto interno** se puede definir según
  $$a.b = ||a||||b||cos\alpha$$

Redes neuronales artificiales

Introducción

- De este modo, la proyección de *a* sobre *b* la podemos definir en función del producto interno

$$||a||cos\alpha = \frac{a.b}{||b||}$$

- Es decir, que podemos representar la proyección de *a* sobre *b* como

$$||a||cos\alpha = \frac{\sum\limits_{i=1}^{n} a_i b_i}{||b||}$$

Redes neuronales artificiales

Introducción

- Si notamos los puntos de nuestro conjunto de datos como
  $$\xi_i^\mu$$
  donde μ es el número de ejemplo, $1 \leq \mu \leq 9$ e $1 \leq i \leq 2$, tenemos que la proyección de ξ_i^μ sobre w es
  $$\sum_{i=1}^{2} \xi_i^\mu w_i$$
  asumiendo que w tiene norma 1.

Redes neuronales artificiales

Introducción

En el ejemplo que vimos, la recta (hiperplano) pasa por el origen.

¿Qué ocurre si la recta (hiperplano) no pasa por el origen?

Consideremos la ecuación clásica de una recta en R^2 que no pasa por el origen ($b \neq 0$)

$$y = mx + b$$

que la podemos re-escribir como

$$mx - y + b = 0.$$

Redes neuronales artificiales

Introducción

En R^3 será un plano dado por

$$Ax + By + Cz + D = 0$$

o si escribimos A, B, C y D como w_1, w_2, w_3 y w_0 queda expresada según:

$$w_1 x + w_2 y + w_3 z + w_0 = 0.$$

Redes neuronales artificiales

Introducción

En general, podemos escribir la ecuación del hiperplano en R^n como

$$\sum_{i=1}^{n} w_i x_i - w_0 = 0$$

Si notamos las x_i como ξ_i obtenemos

$$\sum_{i=1}^{n} w_i \xi_i - w_0 = 0$$

Redes neuronales artificiales

Conclusión de la introducción

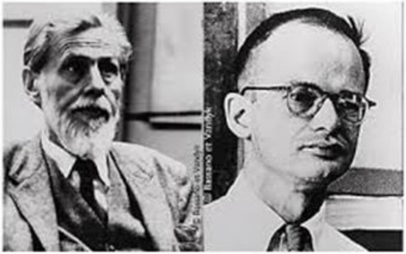
Si en un espacio R^n tenemos puntos distribuidos correspondientes a dos clases diferentes y linealmente separables, entonces
el hiperplano de separación va a estar dado por

$$\sum_{i=1}^{n} w_i \xi_i - w_0 = 0$$

Redes neuronales artificiales

Modelo de McCulloch y Pitts

Ahora vayamos un poco atrás en el tiempo, digamos 1943.



Warren McCulloch (izda.) y Walter Pitts (dcha.)

(foto obtenida de https://www.caminosmadrid.es/9938-2)

Redes neuronales artificiales

Modelo de McCulloch y Pitts

Estos dos investigadores proponen un modelo de neurona basados en

- el **conocimiento** disponible acerca de su morfología y funcionamiento,
- su **interpretación** de aspectos funcionales escenciales,
- **disponibilidad** formal y de **cálculo** para interpretarlo e implementarlo.

Redes neuronales artificiales

Modelo de McCulloch y Pitts

- Warren Sturgis McCulloch (1898-1969) obtuvo su título de grado en psicología y su doctorado en Medicina. Fue miembro fundador de la Asociación americana de cibernética.
- Walter Pitts (1923-1969) fue un precoz estudiante autodidacta en matemáticas y lógica aunque sus estudios futuros se extenderán a la física y la cibernética.

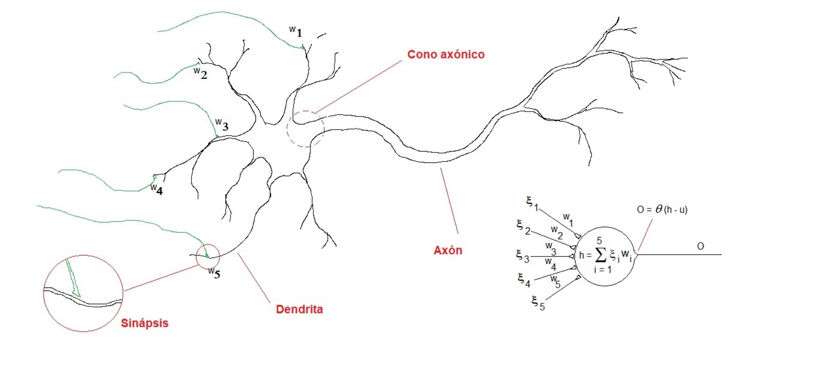
Redes neuronales artificiales

Modelo de McCulloch y Pitts

- Ambos investigadores escriben el artículo 'Un cálculo lógico de ideas inherentes en actividad nerviosa' en 1943 donde proponían un modelo de neurona que les permitía especular acerca del poder de cálculo de un conjunto de ellas interconectadas.

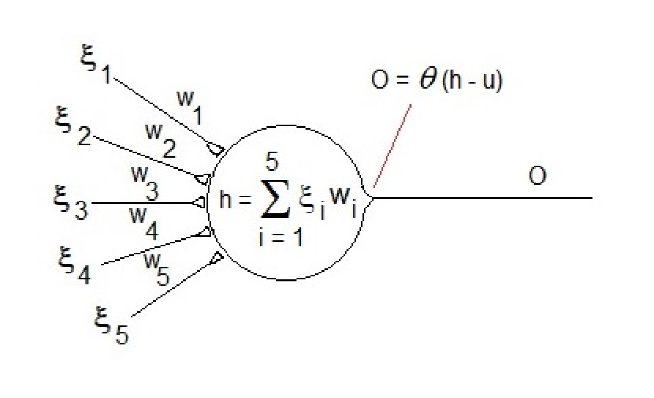
Redes neuronales artificiales

El esquema de neurona en que se basaron fue:



Redes neuronales artificiales

El modelo de neurona que propusieron fue:



Redes neuronales artificiales

Modelo de McCulloch y Pitts

Formalmente, el modelo de neurona de McCulloch y Pitts es

$$O = \theta(\sum_{i=1}^{n} w_i \xi_i - umbral)$$

donde $\theta()$ es la función definida según

$$\theta(x) = \begin{cases} 1 & x \geq 0 \\ 0 & \text{en otro caso} \end{cases}$$

Redes neuronales artificiales

Modelo de McCulloch y Pitts

Si notamos al *umbral* como w_0,
y tomamos $\theta()$ como

$$\theta(x) = \begin{cases} 1 & x \geq 0 \\ -1 & \text{en otro caso} \end{cases}$$

podemos decir que una neurona de McCulloch y Pitts resuelve nuestro problema de separabilidad lineal.

Redes neuronales artificiales

Modelo de McCulloch y Pitts

Por supuesto que, es prematuro decir que lo resuelve ya que aún no sabemos cómo obtener los valores de $w_i, 0 \leq i \leq n$. Pero si aceptamos que:

- nuestras neuronas son parte estructural de nuestro sistema nervioso y,
- que nuestro cerebro aprende,

entonces debería existir algún **proceso** por el cual este se **adapta** para alcanzar la solución del problema.

Redes neuronales artificiales

Donald
Hebb nació en 1904 en Canadá. Su formación y actividades se desarrollaron en distintos ámbitos académicos y laborales pero finalmente su vida se enfocó en la psicología del comportamiento.

Redes neuronales artificiales

Conjetura de Hebb

Su frase

"Neurons that fire together, wire together."

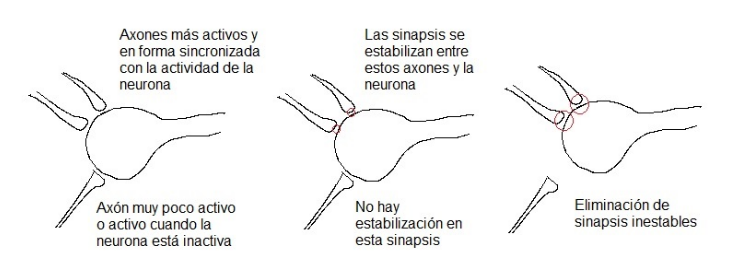
sintetiza uno de sus aportes por el cual fue reconocido en las neurociencias.

¿Qué significa esta frase?

Redes neuronales artificiales

Conjetura de Hebb

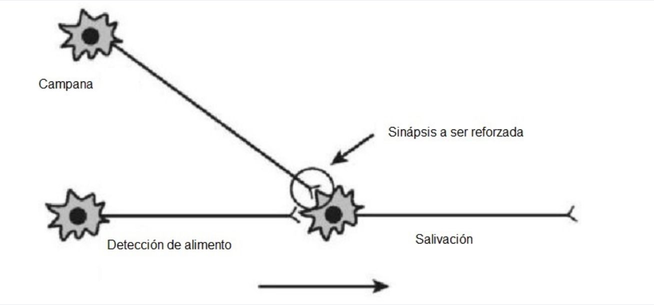
Las conexiones (sinapsis) donde intervienen neuronas cuyo estado de actividad están correlacionados a lo largo del tiempo son reforzadas y este refuerzo depende de la intensidad y período de tiempo en que esto ha ocurrido.



Redes neuronales artificiales

Conjetura de Hebb

Esta conjetura permitió dar explicación a diferentes fenómenos observados en la psicología conductual.
Por ejemplo, al aprendizaje condicionado de Pavlov.



Redes neuronales artificiales

Todo esto es muy bonito, pero ¿cómo formalizamos esto para que nuestra neurona aprenda? Acá podemos incluir a un nuevo protagonista en esta historia. Frank Rosenblatt (1928-1971) que estudió linguistica luego se dedicó a la psicología cognitiva y en 1960 llevó a cabo la construcción del Mark I, una computadora que podía aprender del ensayo y error.

Redes neuronales artificiales

Aprendizaje en el perceptron

Asumiendo la conjetura de Hebb, consideraremos que cada vez que el perceptron recibe un nuevo estímulo (entrada ξ^{μ}) las conexiones sinápticas tendrían la posibilidad de actualizarse:

$$w_i^{nuevo} = w_i^{viejo} + \Delta w_i$$

Aprendizaje en el perceptron

El valor de la actualización Δw_i debería depender de la entrada ξ_i^μ y la salida O^μ, según nos dice la conjetura de Hebb. Como nosotros queremos que O^μ sea igual a ζ^μ (es decir, el estado de activación que esperamos tenga la neurona), entonces podemos definir:

$$\Delta w_i = 2\eta \xi_i^\mu \zeta^\mu$$

donde η es una constante de proporcionalidad llamada tasa de aprendizaje.

Redes neuronales artificiales

Aprendizaje en el perceptron

Entonces, si cuando se presenta la entrada ξ^{μ} el estado de activación O^{μ} de la neurona coincide con su salida deseada ζ^{μ}, no es necesario actualizar w_i mientras que, si no coincide, deberá actualizarse w_i.
Esto es:

$$\Delta w_i = \begin{cases} 2\eta\xi_i^{\mu}\zeta^{\mu} & si O^{\mu} \neq \zeta^{\mu} \\ 0 & \text{en otro caso} \end{cases}$$

Redes neuronales artificiales

Aprendizaje en el perceptron

Si O^μ y ζ^μ adoptan valor 1 o -1, podemos reescribir

$$\Delta w_i = \begin{cases} 2\eta\xi_i^\mu\zeta^\mu & si O^\mu \neq \zeta^\mu \\ 0 & \text{en otro caso} \end{cases}$$

por

$$\Delta w_i = \eta(1 - O^\mu\zeta^\mu)\zeta^\mu\xi_i^\mu$$

o simplemente

$$\Delta w_i = \eta(\zeta^\mu - O^\mu)\xi_i^\mu.$$

Redes neuronales artificiales

Aprendizaje en el perceptron

Por lo tanto, cada vez que apliquemos

$$\Delta w_i = \eta(\zeta^\mu - O^\mu)\xi_i^\mu$$

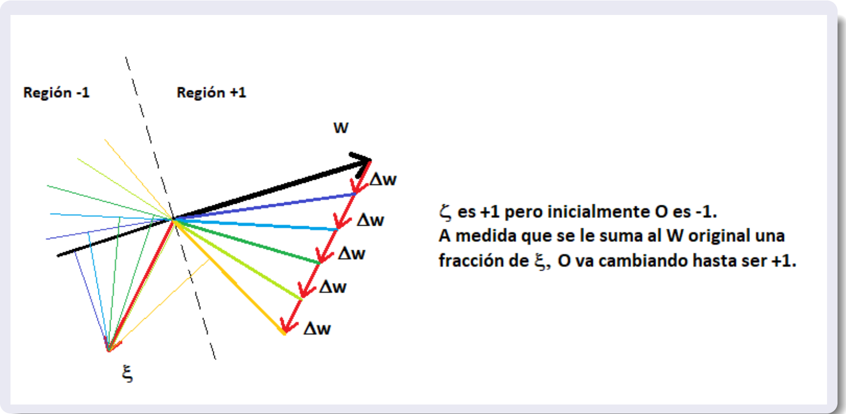
en

$$w_i^{nuevo} = w_i^{viejo} + \Delta w_i$$

si $\zeta^\mu > O^\mu$

lo que estaremos haciendo es mover (una proporción η) el vector w hacia ξ^μ (w se parecerá un poco más a ξ^μ) y esto provocará que la próxima vez que calculemos el producto interno $w\xi^\mu$ será mayor y será más probable que O^μ sea igual a ζ^μ.

Redes neuronales artificiales



Aprendizaje en el perceptron

Por el contrario, cuando apliquemos
$$\Delta w_i = \eta(\zeta^\mu - O^\mu)\xi_i^\mu$$
si $\zeta^\mu < O^\mu$
lo que estaremos haciendo es mover (una proporción η) el vector w alejándolo de ξ^μ (w se parecerá un poco menos a ξ^μ) lo que provocará que la próxima vez que calculemos el producto interno $w\xi^\mu$ será menor y será más probable que O^μ sea igual a ζ^μ.

Algoritmo perceptron simple

- Una época es cuando fueron expuestos al perceptrón todos las entradas del conjunto de entrenamiento.
- p es la cantidad de entradas del conjunto de entrenamiento.
- $x[.]$ es el conjunto de entrenamiento donde cada entrada tiene dimensión $N + 1$ (se le agrega una coordenada en 1 para el umbral),
- $y[.]$ es la salida deseada.
- w es el vector de pesos 'sinápticos' que incluye el umbral.
- signo() es la función de activación.

Algoritmo perceptron simple

i = 0
w = zeros(N+1, 1)
error = 1
error_min = p * 2
while *error* $> 0 \wedge i <$ *COTA*
    Tomar un número i_x al azar entre 1 y p
    Calcular la exitación $h = x[i\_x].w$
    Calcular la activación $O = signo(h)$
    $\Delta w = \eta * (y[i\_x] - O).x[i\_x]$
    $w = w + \Delta w$
    error = CalcularError(x, y, w, p)
    if *error* $<$ *error_min*
        error_min = error
        w_min = w
    end
    i = i + 1
end

Redes neuronales artificiales

Aprendizaje en el perceptron lineal

Si en vez de tener una función de activación escalón, la misma es lineal se denomina al perceptron simple como perceptron simple lineal:

$$O^\mu = \sum_{i=0}^{N} w_i \xi_i^\mu$$

donde O^μ es la activación (o salida) del perceptron para una entrada ξ^μ de dimensión N.
Por otro lado, la salida deseada ζ^μ en vez de pertenecer exlusivamente a $\{-1, 1\}$ o $\{1, 0\}$ como ocurría con las unidades escalón, puede pertenecer a los reales.

Redes neuronales artificiales

Aprendizaje en el perceptron lineal

Un aspecto interesante del perceptron lineal es que se puede formular una función de costo (derivable) que adoptará un mínimo absoluto cuando el valor del vector de pesos w haga que la salida $O^\mu = \zeta^\mu$ para cada entrada ξ^μ:

$$\begin{aligned} E(w) &= \frac{1}{2}\sum_{\mu=1}^{p}(\zeta^\mu - O^\mu)^2 \\ &= \frac{1}{2}\sum_{\mu=1}^{p}(\zeta^\mu - \sum_{i=0}^{N} w_i\xi_i^\mu)^2 \end{aligned}$$

Redes neuronales artificiales

Aprendizaje en el perceptron lineal

Fijarse que $E()$ depende de los pesos w_i.
Si se logra mover w en la dirección correcta podríamos acercarnos al mínimo de $E()$.
Una posibilidad es usar el método del gradiente descendente el cual sugiere ir cambiando w de acuerdo a una proporción del gradiente:

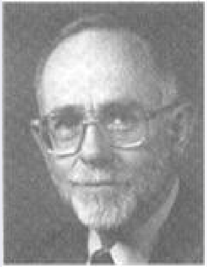
$$\begin{aligned}\Delta w_i &= -\eta \frac{\partial E}{\partial w_i} = \\ &= \eta \sum_{\mu=1}^{p} (\zeta^{\mu} - O^{\mu}) \xi_i^{\mu},\end{aligned}$$

o si se desea cambiar w sólo usando la información de una entrada:

$$\Delta w_i = \eta (\zeta^{\mu} - O^{\mu}) \xi_i^{\mu}.$$

Aprendizaje en el perceptron lineal

Aunque el origen es completamente diferente, esta regla de actualización es la misma que usabamos para el perceptron simple con función de activación escalón. Se la conoce como la regla de Widrow-Hoff (1960) los cuales bautizaron a este perceptron como ADALINE (ADApter LINear Element).



Redes neuronales artificiales

Aprendizaje en el perceptron lineal

Hoff fue estudiante de Widrow
y más adelante, fue el co-inventor
del predecesor del microprocesador
en 1968 (un conjunto de
12 circuitos diferentes en un sólo
chip que cambiaba su funcionalidad
de acuerdo al requerimiento).

Redes neuronales artificiales

Aprendizaje en el perceptron lineal

Fijarse que, mientras el perceptron simple escalón resuelve problemas linealmente separables, el perceptron simple lineal resuelve problemas expresados con pares entradas-salidas linealmente independientes.

En todos los casos encontrará el w que aproxime una solución al sistema:

$$w_0 + w_1\xi_1^1 + w_2\xi_2^2 + \ldots + w_N\xi_N^1 = \zeta_i^1$$

$$\ldots$$

$$w_0 + w_1\xi_1^\mu + w_2\xi_2^\mu + \ldots + w_N\xi_N^\mu = \zeta_i^\mu$$

$$\ldots$$

$$w_0 + w_1\xi_1^p + w_2\xi_2^p + \ldots + w_N\xi_N^p = \zeta_i^p$$

Aprendizaje en el perceptron lineal

La convergencia del método depende fuertemente de η en

$$w_i^{nuevo} = w_i^{viejo} + \Delta w_i$$

donde $\Delta w_i = \eta(\zeta^\mu - O^\mu)\xi_i^\mu$.

Si η no es lo suficientemente pequeña, el método puede llegar a diverger.

Si η es demasiado pequeña puede demorar la convergencia.

Redes neuronales artificiales

Aprendizaje en el perceptron simple no lineal

Es una generalización del perceptron simple lineal.
En este caso estado de activación del perceptron se obtiene siguiendo:
$$O^\mu = g(\sum_{i=0}^{N} w_i \xi_i^\mu)$$
donde $g()$ puede ser una función sigmoidea tal como
$$g(x) = tanh(\beta x)$$
o
$$g(x) = \frac{1}{1+exp^{-2\beta x}}$$

Redes neuronales artificiales

Aprendizaje en el perceptron simple no lineal

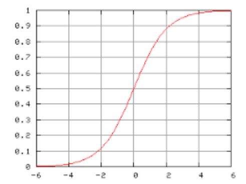


Figura: (Logística)

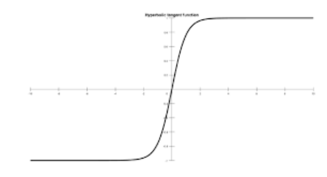


Figura: (Tanh)

Redes neuronales artificiales

Aprendizaje en el perceptron simple no lineal

La función de costo $E(w)$ queda definida igual que con el perceptron simple lineal pero necesitamos explicitar la función $g()$:

$$\begin{aligned} E(w) &= \frac{1}{2}\sum_{\mu=1}^{p}(\zeta^{\mu} - O^{\mu})^2 \\ &= \frac{1}{2}\sum_{\mu=1}^{p}(\zeta^{\mu} - g(\sum_{i=0}^{N} w_i \xi_i^{\mu}))^2 \end{aligned}$$

Redes neuronales artificiales

Aprendizaje en el perceptron simple no lineal

Nuevamente, la actualización de los pesos w se obtendrá usando el método del gradiente descendente:

$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i} =$$

excepto que en este caso, al usar la regla de la cadena, aparece la derivada de $g()$

$$\Delta w_i = \eta \sum_{\mu=1}^{p} (\zeta^\mu - g(h^\mu)) g'(h^\mu) \xi_i^\mu,$$

donde

$$h^\mu = \sum_{i=0}^{N} w_i \xi_i^\mu$$

Aprendizaje en el perceptron simple no lineal

Si se desea cambiar w sólo usando la información de una entrada μ:

$$\Delta w_i = \eta(\zeta^\mu - O^\mu)g'(h^\mu)\xi_i^\mu.$$

Redes neuronales artificiales

Aprendizaje en el perceptron simple no lineal

Un aspecto conveniente del uso de la función logística o la tanh es que sus derivadas se pueden expresar en función de la misma función:

$$\text{Si } g(h) = tanh(\beta h) \text{ entonces } g'(h) = \beta(1 - g^2(h))$$

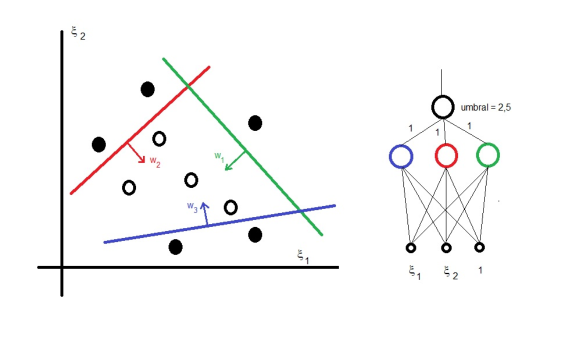
o

$$\text{Si } g(h) = \frac{1}{1+exp^{-2\beta h}} \text{ entonces } g'(h) = 2\beta g(h)(1 - g(h))$$

Redes neuronales artificiales

Perceptron - Consideraciones finales

Los perceptrones se podrían combinar para resolver problemas más complicados.



Redes neuronales artificiales

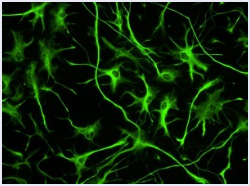
Perceptron - Consideraciones finales

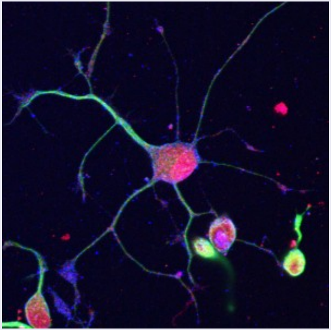
- El modelo de neurona es una simplificación que brinda soluciones para determinado tipos de problemas.
- Pero tantos las neuronas como las redes de neuronas son sistemas mucho más complejos.

Redes neuronales artificiales

Perceptron - Consideraciones finales

https://www.scienceweek.net.au/neural-knitworks/neuron-gallery/ Células cultivadas y diferenciadas a partir de células indiferenciadas

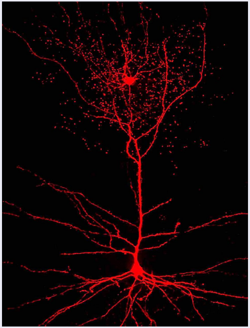




Redes neuronales artificiales

Perceptron - Consideraciones finales

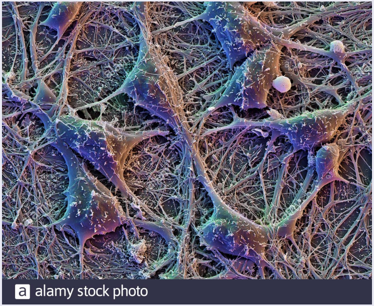
https://www.forbes.com/sites/andreamorris/2018/08/27/scientists-discover-a-new-type-of-brain-cell-in-humans/#1f8dd4b23acd. Neurona inhibitoria corteza humana



Redes neuronales artificiales

Perceptron - Consideraciones finales

https://www.alamy.com/brain-nerve-cells-scanning-electron-micrograph-sem-of-cortical-neurons-nerve-cells-from-the-brain-showing-an-extensive-network-of-interconnectin-image



Redes neuronales artificiales

Perceptron - Consideraciones finales

https://www.canadiannaturephotographer.com/rberdan_scanning_electron_microscopy2017.html