Sesión 6
Programacion paralela
Docente: Mg. Huarote Zegarra Raúl

Objetivos:

- Conoce el fundamento, características, tipos de la red neuronal.
- Conoce el proceso de aprendizaje del perceptron simple.

TEMAS A TRATAR:

- Fundamento de la red neuronal.
- Tipos de red neuronal
- Proceso de aprendizaje de la red neuronal.
- Aprendizaje de la compuerta lógica OR en Python.

Prerrequisito

 Identificación de características comunes de un determinado objeto.

Ejemplo:

- -Silla: tiene espaldar, tiene patas, etc.
- -Huella dactilar: líneas rugosas (crestas), valles, terminaciones, bifurcaciones, core, delta, etc.

Preámbulo



• Un niño en clase le decimos que vea varias veces las diferentes sillas y le decimos que eso es una silla, como ya identificó sus características del objeto entonces cuando llegue a su casa y vea algo similar dirá: ¡ese es una silla;.

ENTONCES APRENDIÓ



Inteligencia artificial:

"Es evidente que las computadoras que posean una inteligencia a nivel humano o superior, tendrán repercusiones importantes en la vida diaria, así como en el devenir de la civilización" P. P. Cruz (2010)

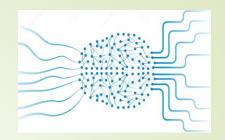
"El medio por el cual las computadoras, los robots y otros dispositivos realizan tareas que normalmente requieren de la inteligencia humana" G. H. Renato (2013)

"La manera de emular en lo posible cada una de las capacidades propias del humano, agenciándose para su implementación las tecnologías de información, hardware y software" Huarote R.(2018)

Objetivos de la red neuronal artificial

• Conseguir que las máquinas den respuestas similares a las que es capaz de dar el cerebro que se caracterizan por su generalización y su robustez, en el proceso de aprendizaje.

Red neuronal artificial:



"Se basan en las redes neuronales biológicas capaces de resolver funciones no lineales que corresponden a sistemas cuyo comportamiento puede ser complejo y frecuentemente impredecibles, generalmente difíciles (o imposibles) de modelar" Rivas, & Mazon(2018)

"Cerebro como un modelo de aparato computacional que opera en paralelo, desarrollada con modelos matemáticos simplificados de sistemas cerebrales" Baron R. (1987)

"Son redes entrenadas a través de las entradas obtenidas a partir de escenarios externos o internos en el sistema y estas entradas se multiplican por pesos asignados al azar" Oliveira Colabone, Ferrari, Da Silva Vecchia, & Bruno Tech(2015).

Se basan en las redes neuronales biológicas del cerebro humano. Constituidas por elementos que se comportan de manera similar a la neurona biológica en sus funciones mas comunes. Organizados de la manera similar a la que posee el humano.

El funcionamiento de la neurona biológica del humano es a través de impulsos eléctricos generándose de acuerdo a procesos químicos y a su vez por potencial eléctrico.

Las concentraciones de Na₊y de K₊hacen que el potencial de la membrana vuelva a su valor de reposo.

Se considera que en el cerebro hay unos 100.000 millones de neuronas (10¹¹) y diez veces más conexiones (10¹²). Es materialmente imposible crear un sistema electrónico con tal cantidad de procesadores y de conexiones.

Características de las redes neuronales artificiales

Tiene una serie de características:

- Aprender. Adquiere el conocimiento de una cosa por medio del estudio, ejercicio o experiencia. Muestra un conjunto de entradas y ellas mismas se ajustan para producir unas salidas consistentes.
- Generalizar. Extender o ampliar una cosa. Ya que su propia estructura y naturaleza lo permite. Presentan respuestas correctas a entradas que presentan pequeñas variaciones debido a los efectos del ruido o distorsión.
- Abstraer. Considera por separado las cualidades de un objeto.

Aplicaciones de las redes neuronales artificiales

Presentan resultados razonables en aplicaciones donde las entradas presentan ruidos o las entradas están incompletas, algunas de las aplicaciones son:

- Procesamiento neural del lenguaje: Permiten al sistema generar buenos pronósticos para nuevos verbos o verbos desconocidos.
- Compresión de imágenes: Es la transformación de los datos de una imagen a una representación diferente que requiera menos memoria o que se pueda reconstruir una imagen imperceptible.
- Reconocimiento de caracteres: Es el proceso de interpretación visual y reconocimiento de símbolos.

Aplicaciones de las redes neuronales artificiales

- Procesamiento de señales: Donde una de ellas es la predicción, modelado de sistema y filtrado de ruido.
- Modelos económicos y financieros: Una de las aplicaciones mas importantes en el modelado y pronostico es la creación de pronósticos económico como por ejemplo los precios de existencia.
- Sistemas expertos: Por su capacidad de aprender, tanto el conocimiento de antecedentes y poder discernir en un nuevo resultado aceptable aun cuando no ha sido parte de su conocimiento.

Aplicaciones de las redes neuronales artificiales

- Reconocimiento de nuestra huella dactilar (para marcar asistencia), viene interiormente una huella dactilar.
- Reconocimiento de rostros en una escena.
- Reconocimiento de cheques, si es que es verdadero y falso tu firma.
- Identificación de tipos de enfermedades.
- Aprendizaje de casos de enfermedades relacionado con un determinado síntoma(s).
- Reconocimiento de una anomalía.
- Reconocimiento de Huellas dactilares,
- Reconocimiento de Retina.
- Reconocimiento de Rostro

Entrenamiento de las redes neuronales artificiales

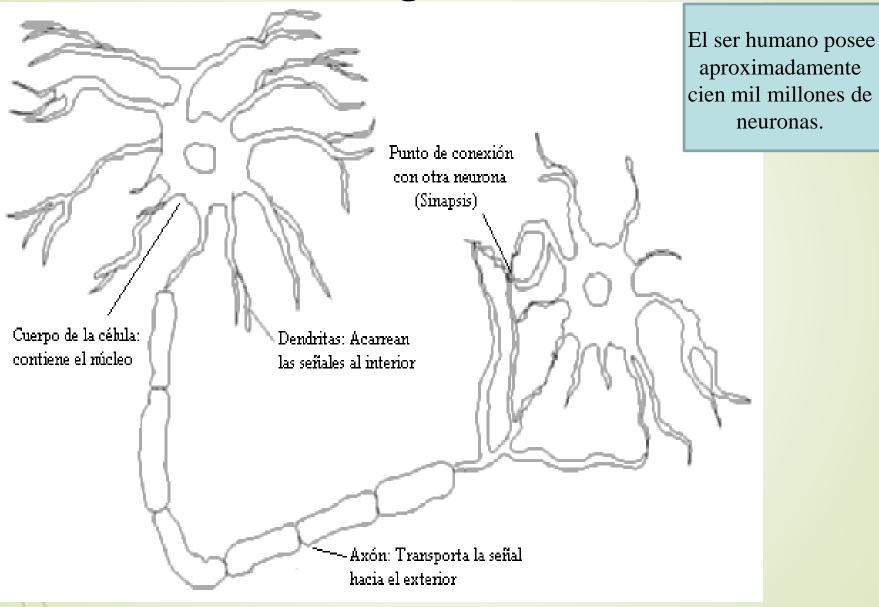
- El entrenamiento de las redes neuronales artificiales muestra algunos paralelismos con el desarrollo intelectual de los seres humanos.
- El objetivo es conseguir una aplicación determinada para un conjunto que produzca el conjunto de salidas deseadas o minimamente consistentes.
- Durante el proceso de entrenamiento los pesos convergen gradualmente hacia los valores que hacen que cada entrada produzca el vector de salida deseada.

Tipos de entrenamiento de las redes neuronales artificiales

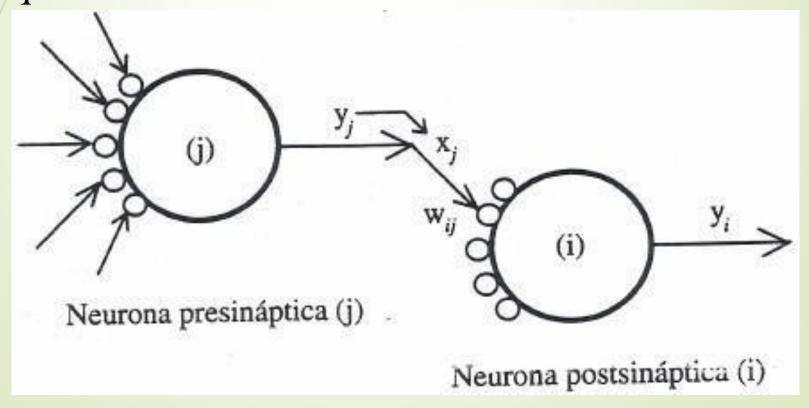
Se pueden clasificar en 2 grupos.

- Supervisado: Requiere el emparejamiento de cada patrón de entrada una respectiva salida deseada. Al aplicar los patrones de entrada se obtiene una salida y esta se compara con la deseada produciendo un error utilizándola para realimentar la red y cambiar los pesos de acuerdo a un algoritmo que tiende minimizar ese error.
- No supervisado: No requiere un vector de salida deseadas, por lo tanto no se comparan las salidas obtenida con las salidas esperadas. Son redes autoorganizativas (comúnmente llamadas SOM)

Modelo biológico de neurona

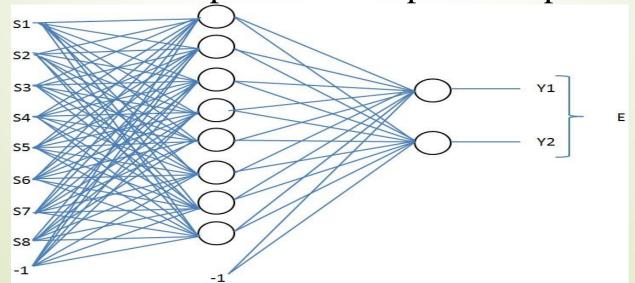


La salida de una neurona (con la función de evaluación) se conecta a la entrada de la siguiente. Con este tipo de conexiones podemos realizar una red neuronal tan compleja como queramos.



Estructura de una redeneuronales

- Debemos de tener en cuanta sus capas:
 - Para los modelos de hoy por la complejidad de problemas que se presenta, requieren necesariamente ser multicapa.
 - Para una capa es el Perceptron simple.



Tipos de redes neuronales artificiales

1	Adaline y Madaline	11 DRS
2	ART	12 FLN
3	Back-Propagation	13 Hamming
4	BAM	14 Hopfield
5	The Boltzman Machine	15 LVQ
6	Brain-State-in a Box	16 Perceptron
7	Cascade-Correlation-Networks	17 PNN
8	Counter-Propagation	18 Recirculation
9	DBD	19 SOM
10.	- DNNA	20 SPR

Comparativa del modelo biológico con el matemático

Núcleo

Proceso de sumatoria

Dendritas (conexión con otras neuronas - Receptoras)

Patrones de entrada (X0,X1...Xn)

Axón

Función de transferencia

Sinapsis (conexión con otras neuronas – Emisora)

Sinapsis (conexión con otras neuronas)

Definición de funciones

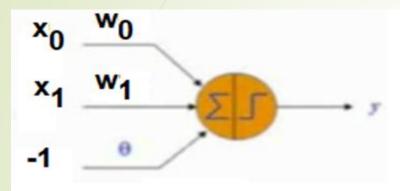
- Función de propagación.
 - Ex citatoria: El valor nuevo a procesar es positivo.
 - Inhibitoria: El valor nuevo a procesar es negativo.
- Función de activación: Modifica el anterior, tenemos: Escalonada, gaussiana, sigmoidea, tangente hiperbólica, etc.
- Función de transferencia: De acuerdo a la función que valor te va a dar, tal como sigmoidea te da [0,1] o la tangente hiperbólica te da [-1,1].

Dentro del proceso de aprendizaje nos damos cuenta que requiere una función de adaptación

tales como:

como.	Función	Rango	Gráfica
Identidad	y = x	[-00, +00]	- 10
Escalón	y = sign(x) $y = H(x)$	{-1, +1} {0, +1}	S(x) x
Lineal a tramos	$y = \begin{cases} -1, & \text{si } x < -l \\ x, & \text{si } +l \le x \le -l \\ +1, & \text{si } x > +l \end{cases}$	[-1, +1]	1 +1 ×
Sigmoidea	$y = \frac{1}{1 + e^{-x}}$ $y = tgh(x)$	[0, +1] [-1, +1])(x)
Gaussiana	$y = Ae^{-Rx^2}$	[0,+1]	NEO X

Comparativa del modelo biológico con el modelo artificial



$$S = \sum_{i=1}^{n} w_{ij}x_j - \theta_i$$

Formula_1

$$Y = f(s) = \begin{cases} 1 & \sin x_0 w_0 + x_1 w_1 - \theta \ge 0 \\ -1 & \sin x_0 w_0 + x_1 w_1 - \theta < 0 \end{cases}$$

Formula_2

Comparativa del modelo biológico con el modelo artificial

Formula_3

Reajuste de pesos:

Propagación de una red neuronal

- La manera de propagación de una red neuronal es:
- 1) Recibe los patrones de entrada a la neurona.
- 2) Inicialmente se genera los pesos de manera aleatoria [-1,1] para cada una de las dendritas.
- 3) Se suman de manera Y = Xi*Wi ; i=1.. n dendritas.
- 4) Este valor Y se evalúa con la función de transferencia y respecto a ello se ve si es excitatoria o inhibitoria con respecto al factor de aprendizaje.
- 5) El valor de salida de esta función viene a ser entrada para otra neurona de la red y así sucesivamente.

Propagación de una red neuronal

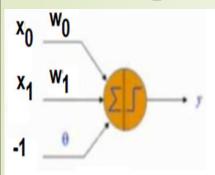
• 5.1.- Si es que el valor obtenido en la función de transferencia no es el deseado, entonces se deben reajustar sus pesos.

Ejemplo: En una tabla lógica OR.

X 0	X1	X0 OR X1
1	1	1
1	-1	1
-1	1	1
-1	-1	-1

X 0	X1	X0 OR X1	Obtenido	Deseada (T)
1	-1	1	1	1
1	-1	1	-1	1reajustar pesos

Implementar en Python la siguiente red neuronal artificial



$$S = \sum_{j=1}^{n} w_{ij} x_j - \theta_i$$
 Formula_1

Y = f(s) =
$$\begin{cases} 1 & \sin x_0 w_0 + x_1 w_1 - \# \ge 0 \\ -1 & \sin x_0 w_0 + x_1 w_1 - \# < 0 \end{cases}$$
 Formula_2

X1	X2	T (salida deseada)
1	1	1
1	-1	1
-1	1	1
-1	-1	-1

Reajuste de pesos:

Formula_3

 Δ wij (t) = wij(t-1)+2 e (Ti) xj; donde:

 Δ wij (t) = peso nuevo

wij (t-1) = peso actual

e = factor de aprendizaje

Ti = salida deseada.

xj = entrada actual.

Preguntas ?

```
import random
    import numpy as np
    import threading
   class Neurona:
 5
        e = 0.5
 6
        w = np.array([random.random(), random.random()])
        u = random.random()
 8
        def init (self):
 9
            print("Creado")
10
11
        def operar(self,X,T):
12
            i = 0
13
            while (i < 4):
14
                 s = sum(X[i][:]*self.w[:]) - self.u
15
                Y = -1.0 \# Y es la variable para la salida obtenida
16
                 if s>=0.0:
17
                     Y = 1.0
18
19
                 if(Y!=T[i]):# evaluar para la formula 3
                     self.w[0] = self.w[0]+2.0*self.e*T[i]*X[i][0]
20
21
                     self.w[1] = self.w[1]+2.0*self.e*T[i]*X[i][1]
2.2.
                     self.u = self.u+2*self.e*T[i]*(-1)
23
                     i = -1
```

```
23
                     i = -1
                 i = i + 1
24
25
             print("Terminado entrenamiento")
26
2.7
        def entrenar(self,X,T):
2.8
             hilo = threading. Thread (target=self.operar, args=(X,T,))
29
             hilo.start()
30
             hilo.join()
31
             # . . . .
             # usando eventos
32
33 🖨
        def mapear(self,Xi):
34
             y = Xi[0]*self.w[0] + Xi[1]*self.w[1] - self.u
35
             if y >=0:
36
                 return 1.0
37 🖨
             else:
38
                 return -1.0
39
40
   PX = np.array([[1.0,1.0],
41
            [1.0, -1.0],
42
           [-1.0, 1.0],
43
            [-1.0, -1.0]
44
45
    T = np.array([[1.0], [-1.0], [-1.0], [-1.0]])
```

```
T = np.array([[1.0],[-1.0],[-1.0],[-1.0]])
n = Neurona()
n.entrenar(X,T)
print("---- mapeo o comprobacion -----")
for i in range(4):
    print(X[i,:],n.mapear(X[i,:]))
```