

Respuestas de Simulación de una Neurona Artificial

Diego Iván Perea Montealegre (2185751) diego.perea@uao.edu.co

Facultad de Ingeniería, Universidad Autónoma de Occidente

Cali, Valle del Cauca

Pregunta 1.

La función de conducción de los coches autónomos de Tesla, Autopilot, cada vez se desarrolla más y adquiere mayor relevancia en la carrera hacia la conducción totalmente autónoma. Recientemente, se ha dado a conocer, a través de Andrej Karpathy, Director de Inteligencia Artificial y Visión por Ordenador de Tesla, información sobre el proceso para entrenar las redes neuronales del Tesla Autopilot. cuenta con doble CPU redundante, es capaz de ofrecer más de 144 Tera Operaciones Por Segundo (TOPS) de rendimiento de redes neuronales. El FSD puede procesar toda la información que recibe, con una capacidad de hasta 2,5 gigapíxeles por segundo y 36,8 TOPS. Con respecto a la memoria RAM, ésta opera a 4266 Gbps y cuenta con un ancho de banda máximo de 68 Gbps. Y el ordenador FSD tiene integrados sus propios procesadores de señal de imagen de 24 bits, que son compatibles con mapeo de tonos avanzado y reducción de ruido, capacitados para realizar operaciones de hasta un gigapíxel por segundo. Otro punto interesante para el desarrollo del Autopilot de Tesla está en 'Operación Vacación'. Gracias a esta, prácticamente es posible seguir entrenando las redes neuronales mientras el coche sigue circulando. De esta forma, todos los coches Tesla aportan para crear una gran base de datos que alimenta y entrena estas redes neuronales.

Y los números de datos recopilados son realmente impresionantes. Se trata de más de 1000 millones de millas de operación del Tesla Autopilot, sólo en Norteamérica. Se incluyen más de 200.000 cambios de carril automáticos, también la operación en más de 50 países, y más de medio millón de sesiones de Smart Summon, la nueva función para invocar de forma autónoma a nuestro Tesla.

Pregunta 2. ,

Realicé la siguiente ecuación $5x+1=6$, en donde la ecuación está dada en lineal $m*x + b = y$, en donde en el perceptrón es x , es la entrada , m es el peso (w) y b es el factor de activación y como $z \geq 0$ la salida es igual a 1 . Por lo que sería lo correspondiente:

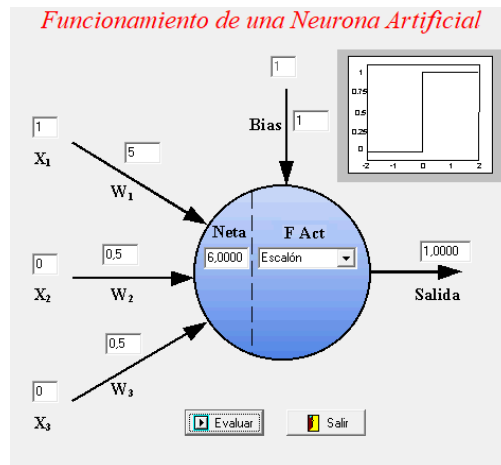


Figura 1. Funcionamiento del perceptrón con ecuación de $5 \cdot x + 1 = 6$

Pregunta 3. Con los valores de $w=2.5$ y $b=-1$

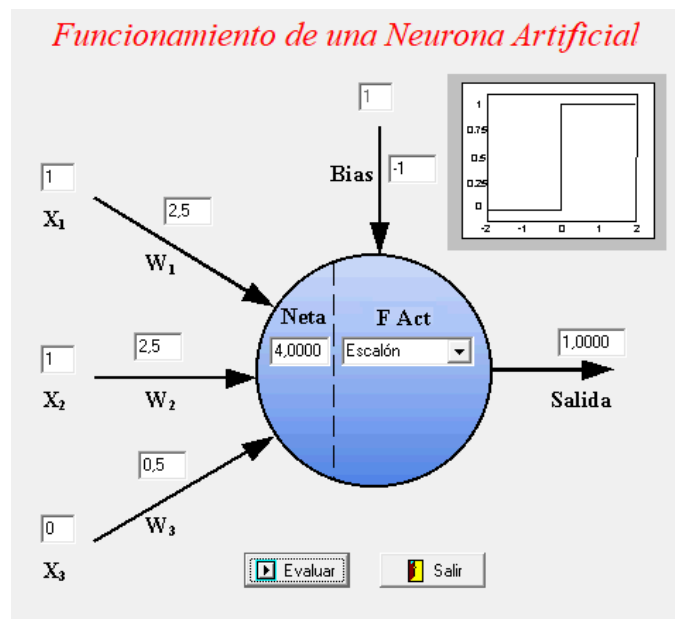


Figura 2. Funcionamiento del perceptrón con OR

Pregunta 4.

Aprendizaje supervisado: se basa en su aprendizaje en un juego de datos de entrenamiento previamente etiquetados. Por etiquetado entendemos que para cada ocurrencia del juego de datos de entrenamiento conocemos el valor de su atributo objetivo. Esto le permitirá al algoritmo poder “aprender” una función capaz de predecir el atributo objetivo para un juego de datos nuevo

Aprendizaje no supervisado: Los métodos no supervisados (unsupervised methods) son algoritmos que basan su proceso de entrenamiento en un juego de datos sin etiquetas o clases previamente definidas. Es decir, a priori no se conoce ningún valor objetivo o de clase, ya sea categórico o numérico. El aprendizaje no supervisado está dedicado a las tareas de agrupamiento, también

llamadas clustering o segmentación, donde su objetivo es encontrar grupos similares en el conjunto de datos.

El aprendizaje por refuerzos o Reinforcement Learning es un área del aprendizaje automático (machine learning). Su principal particularidad es que es capaz de funcionar sin grandes cantidades de datos de entrenamiento. Tan “sólo” necesita una serie de indicaciones para ir aprendiendo a través de prueba y error. A diferencia del aprendizaje supervisado basado en un conjunto de datos que le indica a la máquina qué debe hacer, aquí se utilizan recompensas para reforzar el comportamiento deseado.

El perceptrón es de tipo de aprendizaje supervisado debido a que dependiendo de la entrada con los pesos en su variación los clasifica.

Pregunta 5.

Redes Monocapa: Primer Nivel asociado al vector de entrada. , No hay procesamiento de estos datos, por esta razón no se considera formalmente como una capa. , Este primer nivel tiene la misma dimensión del vector de entrada, la información entra al mismo nivel y los datos son transferidos al siguiente nivel, modificados por los pesos sinápticos. , Como las neuronas de este nivel reciben el mismo tipo de información lo denominamos capa y como a su vez corresponde a la salida de la red, la llamaremos Capa de Salida.

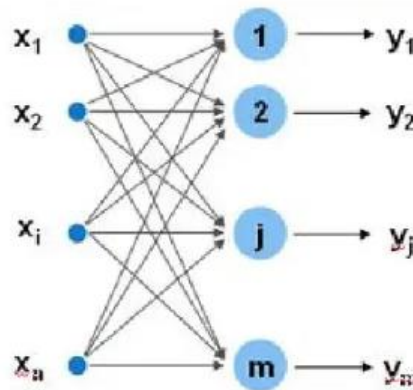


Figura 3. Redes monocapa

Redes multicapa: La red tiene un nivel de entrada con n neuronas y una capa de salida de neuronas. La red incluye una capa intermedia, denominada Capa Oculta, que está conformada por h neuronas. ¿Por qué el nombre de oculta? Como en el caso anterior, la información fluye en única dirección. En este ejemplo, presentamos una única capa oculta, pero una red puede tener más de una capa intermedia.

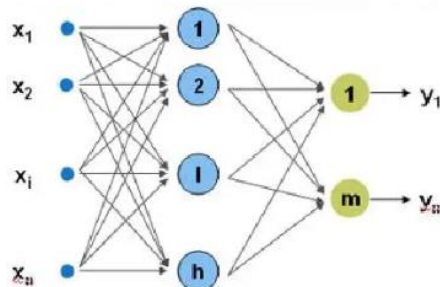


Figura 4. Redes multicapa

Redes feed forward :Este tipo de redes son una extensión del perceptrón. Constan de varias capas de neuronas, "fully connected" (es decir, una celda está conectada con todas las celdas de la siguiente capa), hay al menos una capa oculta, la información circula de izquierda a derecha ("feed forward") y el entrenamiento de la red se suele realizar mediante back-propagation (que revisaremos un poco más adelante).

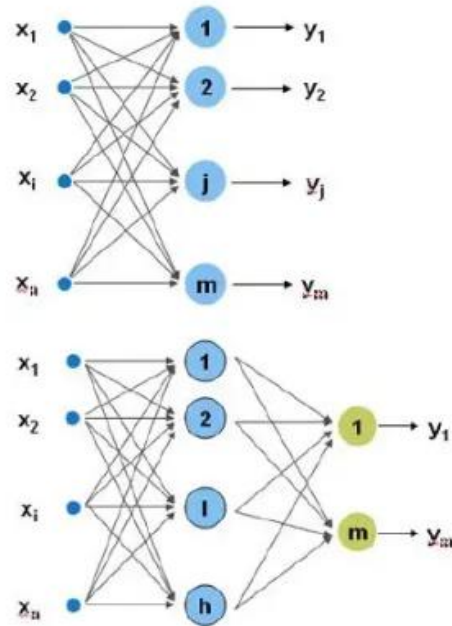


Figura 5. Redes feed forward

Redes recurrentes: La información no siempre fluye en un sentido, puesto que puede realimentarse hacia capas anteriores a través de conexiones sinápticas

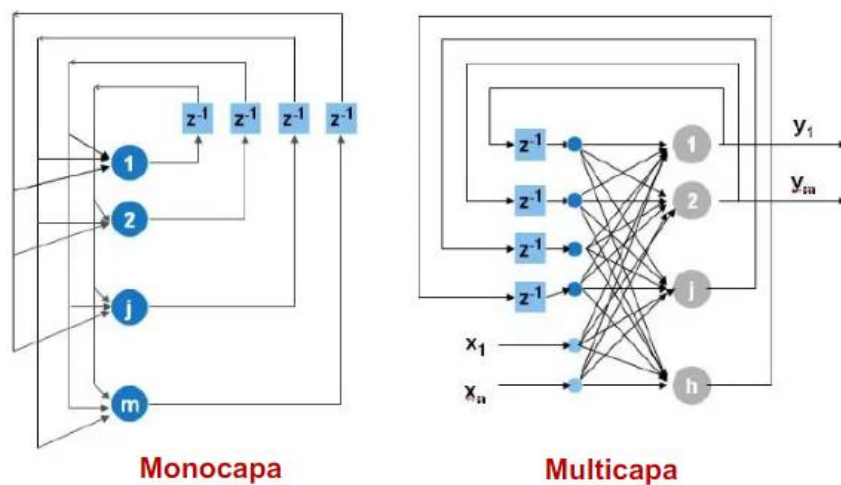
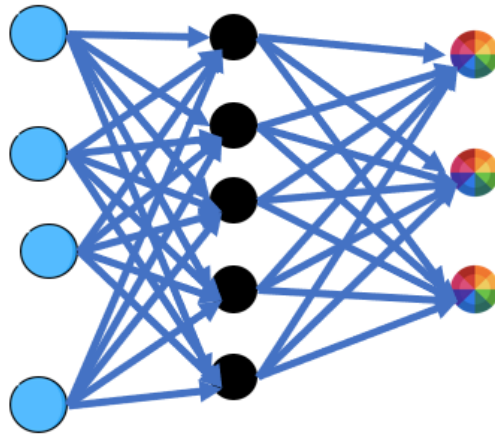


Figura 6. Redes recurrentes

Pregunta 6. Una de las arquitecturas más usadas en redes artificiales son las redes multicapa de una capa oculta. Investigue como se puede expresar el procesamiento de una red neuronal de este tipo de manera matricial. De acuerdo a lo anterior, considera una red de 4 entradas, 5 neuronas en la capa oculta con función de activación tangente sigmoidal y tres salidas con función de activación lineal. Suponga valores para las matrices de pesos y calcula la salida cuando la entrada es $X=[1.0;-0.5;2.5;0.5;]$



Las salidas correspondientes son: `array([[0.02747277], [0.9941399], [0.10586319]])`

Código Python:

```
import pandas as pd
import numpy as np
W1 = np.array([2, 1,4,1,2,1, -3, 4, -2, 5,2, 1,4,1,2,1, -3, 4, -2,
5]).reshape(5, 4) #reshape darle forma a la matriz
W1
b1 = np.array([-1, 0, 3,4,5]).reshape(5, 1)
b1
x1 = np.array([1.0,-0.5,2.5,0.5]).reshape(4, 1)#entradas
x1
from scipy.special import expit

y = expit(W1.dot(x1) + b1) #expit es sigmoid
y
#salidas 3
x3=y
x3
W3 = np.array([-3, 2, -1, 1,5,1,2,3,1,2,-1,-2,-3,-1,-2]).reshape(3, 5)
W3
b3 = np.array([-1, 0, 3]).reshape(3, 1)
b3
salidas_sig = expit(W3.dot(x3) + b3)#forma sigmodal
```

```
salidas_sig  
array([[0.02747277], [0.9941399 ], [0.10586319]])  
#con salida lineal  
salidas_sig = (w3.dot(x3) + b3)#forma lineal  
array([[ -3.56670263], [ 5.13371119], [-2.13371119]])
```

Bibliografía

[1]"» Así es como entrenan las redes neuronales del Tesla Autopilot", Nueva Movilidad, 2022. [Online]. Available: <https://www.nuevamoivilidad.com/movilidad-compartida/asi-es-como-entrenan-las-redes-neuronales-del-tesla-autopilot/>. [Accessed: 05- Aug- 2022]

[2]2022. [Online]. Available: <https://interactivechaos.com/es/manual/tutorial-de-deep-learning/salida-de-la-segunda-capa-oculta>. [Accessed: 06- Aug- 2022]

[3]M. Khan, "La función sigmoidea en Python", *Delft Stack*, 2022. [Online]. Available: <https://www.delftstack.com/es/howto/python/sigmoid-function-python/>. [Accessed: 06- Aug- 2022]