

Enero 2022

Examen - Programación de Redes Neuronales
CAPACITACIÓN LABORAL JUVENIL MÓDULO: PROGRAMACIÓN DE REDES
NEURONALES CON PYTHON
Clave: G135-PYT-SAB-MAT-2

Nombre del alumno: Jorge Luis Rodríguez Guzmán

Fecha de elaboración: 22 de enero del 2022

Instructor(es): **César Guillermo Villanueva Fernández.**

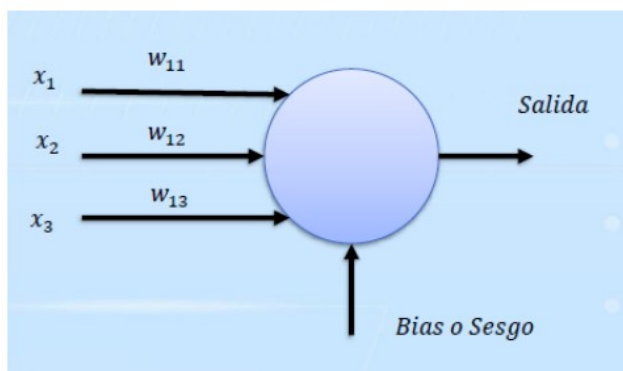
Calificación:

Objetivo: Evaluar las competencias descritas en el programa Construyendo una Red Neuronal con Python

Instrucciones Generales: Lee tu examen cuidadosamente. Pon atención a los detalles.

Parte I. CÁLCULOS TEÓRICOS .

1. Analiza y desarrolla la solución de la neurona de la siguiente figura.



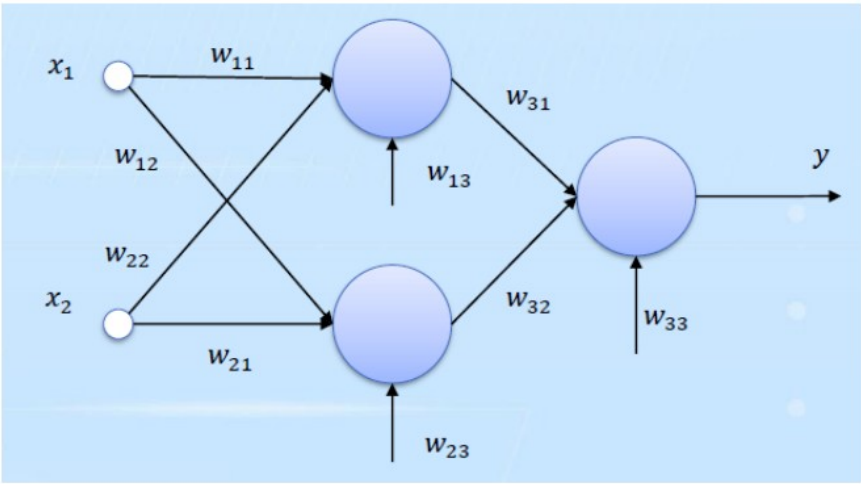
$$\text{Datos: } x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 7 \\ 21 \\ -10 \end{bmatrix} ; w = [w_{11} \ w_{12} \ w_{13} \ w_{1b}] = [-2 \ 4.5 \ 8.2 \ 0.5]$$

Función de activación : **Función Sigmoide**

¿Cuál es el resultado de salida de la neurona?

0.2689

2. Analiza y desarrolla la solución de la Red Neuronal de la siguiente figura.



Datos: $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0.1 \\ 0.5 \end{bmatrix}$; $w = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix} = \begin{bmatrix} -4.8 & 4.6 & -2.6 \\ 5.1 & -5.2 & 3.2 \\ 5.9 & 5.2 & -1.3 \end{bmatrix}$

Función de activación :**Función Sigmoide**

¿Cuál es el resultado de salida de la neurona?

0.174137

Parte II. PROGRAMACIÓN APLICADA A LAS RNA

3. Desarrolla un programa en lenguaje Python 3.8 utilizando programación orientada a objetos y siguiendo un estándar internacional PEP.

Requisitos del Programa:

- Dar solución al problema No.1 y comparar los resultados.
- El programa tiene que ser una solución general y no específica.
- Añade tu archivo .py.

4. Desarrolla un programa en lenguaje Python 3.8 utilizando programación orientada a objetos y siguiendo un estándar internacional PEP.

Requisitos del Programa:

- Dar solución al problema No.2 y comparar los resultados.
- El programa tiene que dar solución a toda la Red y **No** Neurona por Neurona, es decir , utilizar N veces el programa para dar solución individual , te puedes apoyar del programa creado en el ejercicio 3 e interconectar los objetos creados.
- Añade tu archivo .py.

5. Desarrolla un programa en lenguaje Python 3.8 utilizando programación orientada a objetos y siguiendo un estándar internacional PEP.

De acuerdo a los datos siguientes



Los requisitos del programa son:

- Codifica de tal forma que la neurona tenga como entrada los siguientes parámetros al azar y tenga como prueba final un conjunto de entradas , el resultado de la neurona no tiene que ser exacto , pero sin tiene que tener una aproximación considerable.

Prueba				Datos conocidos			
Entradas			Salidas	Entradas			Salidas
X	Y	Z	Q	X	Y	Z	Q
1	0	1	0	0	0	1	1
				1	0	0	0
				0	1	0	0
				1	1	1	1
				0	1	1	1

- El programa puede ser general o personalizado para la solución del problema
 - Al final del programa realiza una validación de la respuesta de la neurona de tal forma que obtengas un resultado exacto de salida. Ejemplo: if **Valor de salida es mayor a 0.666 Salida es igual a 1** , Si no **Salida es igual a 0**.
 - Añade tu archivo .py.
6. Desarrolla un programa en lenguaje Python 3.8 utilizando programación orientada a objetos y siguiendo un estándar internacional PEP.

Requisitos del Programa:

Analiza un problema de un Perceptrón con un mínimo de 2 entradas y con una función de activación Sigmoide. (Omite el valor del Bias)

- Analizar y desarrollar matemáticamente las funciones del error para generar una ecuación de gradiente local utilizando el error cuadrático medio.
- El programa tiene que llegar hasta el calculo de los pesos sinapticos.
- Añade tu archivo .py.

Este ejercicio pude darte puntos extra en tu calificación final si desarrollas el backpropagation con re-alimentación

Requisitos del de puntos extra:

- Puedes utilizar la librería Numpy para utilizar arreglos de vectores.
- El programa debe ser para solución general de N entradas.
- Comparar las iteraciones(Epocas) en una tabla de rendimiento de tu programa.
- Añade tu archivo .py.