



E – Control Inteligente 1

Informe Practica N.º 1

1. Nombre:

Jeferson Gallego Chaverra

2. Semestre:

2023-2

3. Nombre Practica:

Entrenamiento Red Neuronal Sencilla

4. Resumen

En esta práctica se intenta implementar una sencilla red neuronal artificial en el lenguaje de programación Python a partir de la librería Keras la cual permite el desarrollo de redes neuronales trabajando con TensorFlow una biblioteca desarrollada por Google para el aprendizaje automático a través de un rango de tareas.

La red neuronal se entrenará de modo que pueda aprender las combinaciones de dos entradas (-1 o 1), donde la salida será -1 si las dos entradas son iguales o será 1 si las entradas no son iguales.

Es decir que, de cuatro combinaciones posibles, sólo dos tienen salida 1 y las otras dos serán -1, como vemos en la siguiente tabla:

X	Y	Z
-1	-1	-1
-1	1	1
1	-1	1
1	1	-1

Se implementa esta tabla a través de código escrito en Python, contando con los diferentes librerías, funciones y argumentos necesarios para un correcto aprendizaje de esta.

Como argumentos de evaluación de aprendizaje de la red neuronal se contará con:



Time: Tiempo de aprendizaje de la red neuronal.

RMSE: Raíz del error cuadrático medio como función de pérdida.

Learn: La red aprende Sí ☒ o No ☐

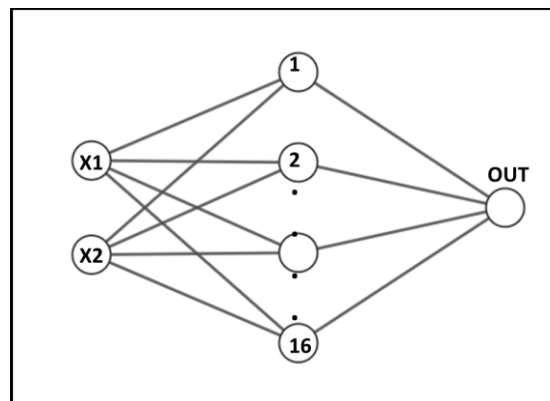
Por último, se tomarán los datos de aprendizaje de la red con el fin de realizar una gráfica 3D a partir de los argumentos X, Y (Entradas), y Z (Salida) y a así poder evaluar de una forma visual si la red consigue aprender a partir los datos.

5. Código

Se envía el archivo .py con sus respectivos comentarios.

6. Resultados Obtenidos:

Para el trabajo de aprendizaje de la red se tomará la siguiente estructura de red neuronal



- Dos entradas X1 y X2
- Capa 1 oculta (2,4,8,16 neuronas)
- Capa de salida (1 neurona)

6.1. Primera Prueba

En las primeras pruebas se intentó hallar que funciones de activación eran óptimas para un correcto aprendizaje de la red neuronal, es necesario para esto que la función de activación pueda entregar datos en el rango de -1 y 1, ya que estos son los resultados posibles según las 4 combinaciones.

Se hicieron 6 pruebas tomando a la función “relu” como función de activación permanente en la capa oculta, se varia la función de activación de la capa de salida para comparar.



Se varia en número de neuronas de la capa de salida (2,4,8,16).

El número de epochs para el aprendizaje es de 1000.

Prueba 1.1

In Funtion	# In Neuron	Out Funtion	# Out Neuron	Time (ms)	RMSE	Learn	Note
Relu	2	Sigmoid	1	107 ms	0.8374	<input type="checkbox"/>	La función Sigmoid no es óptima como función de salida para la solución de problema establecido
	4		1	148 ms	0.5892	<input type="checkbox"/>	
	8		1	100 ms	0.5260	<input type="checkbox"/>	
	16		1	176 ms	0.5106	<input type="checkbox"/>	

Prueba 1.2

In Funtion	# In Neuron	Out Funtion	# Out Neuron	Time (ms)	RMSE	Learn	Note
Relu	2	Tanh	1	100 ms	0.5151	<input type="checkbox"/>	El modelo Aprende a partir de las 4 neuronas en la capa oculta, se mejora el RMSE a partir de que se aumenta las neuronas, pero a la vez aumenta el tiempo de respuesta de la red.
	4		1	89 ms	0.0546	<input checked="" type="checkbox"/>	
	8		1	124 ms	0.0093	<input checked="" type="checkbox"/>	
	16		1	151 ms	0.0066	<input checked="" type="checkbox"/>	

Prueba 1.3

In Funtion	# In Neuron	Out Funtion	# Out Neuron	Time (ms)	RMSE	Learn	Note
Relu	2	Softmax	1	125 ms	2	<input type="checkbox"/>	La función Softmax no es óptima como función de salida para la solución de problema establecido
	4		1	166 ms	2	<input type="checkbox"/>	
	8		1	120 ms	2	<input type="checkbox"/>	
	16		1	93 ms	2	<input type="checkbox"/>	

Prueba 1.4

In Funtion	# In Neuron	Out Funtion	# Out Neuron	Time (ms)	RMSE	Learn	Note
Relu	2	Softplus	1	171 ms	0.8621	<input type="checkbox"/>	La función Softplus no es óptima como función de salida para la solución de problema establecido
	4		1	147 ms	0.5402	<input type="checkbox"/>	
	8		1	85 ms	0.5766	<input type="checkbox"/>	
	16		1	190 ms	0.5111	<input type="checkbox"/>	

Prueba 1.5

In Funtion	# In Neuron	Out Funtion	# Out Neuron	Time (ms)	RMSE	Learn	Note
Relu	2	Softsign	1	156 ms	0.5668	<input type="checkbox"/>	El modelo Aprende a partir de las 8 neuronas en la capa oculta, se mejora el RMSE a partir de que se aumenta las neuronas, pero a la vez aumenta el tiempo de respuesta de la red
	4		1	127 ms	0.2143	<input type="checkbox"/>	
	8		1	108 ms	0.0938	<input checked="" type="checkbox"/>	
	16		1	114 ms	0.0682	<input checked="" type="checkbox"/>	

Prueba 1.6

In Funtion	# In Neuron	Out Funtion	# Out Neuron	Time (ms)	RMSE	Learn	Note
Relu	2	Relu	1	92 ms	1	<input type="checkbox"/>	La función Relu no es óptima como función de salida para la solución de problema establecido
	4		1	240 ms	0.7500	<input type="checkbox"/>	
	8		1	104 ms	0.7500	<input type="checkbox"/>	
	16		1	89 ms	0.7500	<input type="checkbox"/>	



Nota:

Llegado a este punto se llega a la conclusión que las Funciones "**tanh**" y "**softsign**" son las únicas óptimas para ser función de activación para la capa de salida debido a que son las únicas que cuentan con los rangos de salida requeridos (-1,1).

6.2. Segunda Prueba

Se continuará haciendo pruebas de modelos variando las funciones de activación la capa oculta y con las funciones "tanh" y "softsign" como funciones de activación de la capa de salida.

Se varía el número de neuronas de la capa de salida (8,16).

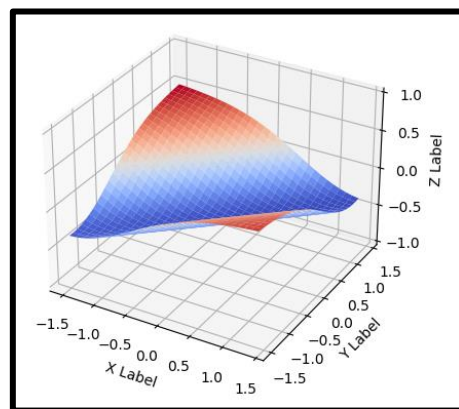
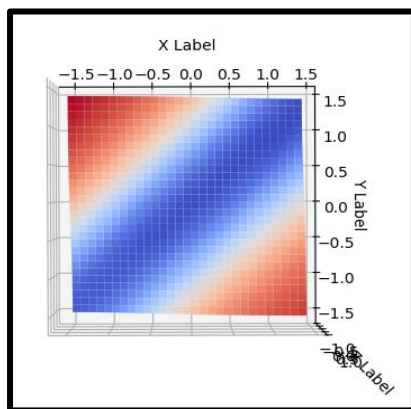
El número de epochs para el aprendizaje es de 1000.

Prueba 2.1.

Test	In Funtion	# In Neuron	Out Funtion	# Out Neuron	Time (ms)	RMSE	Learn
2.1.1	Sigmoid	8	Tanh	1	106	0.4080	✗
2.1.2		16		1	109	0.0592	✓
2.1.3	Sigmoid	8	Softsign	1	109	0.3446	✗
2.1.4		16		1	85	0.4584	✗

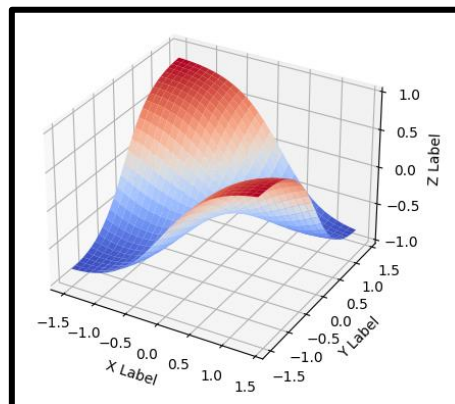
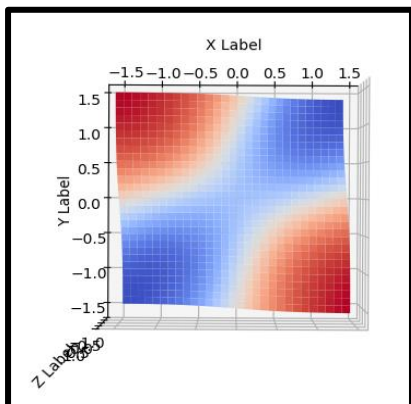
Gráficos

2.1.1.

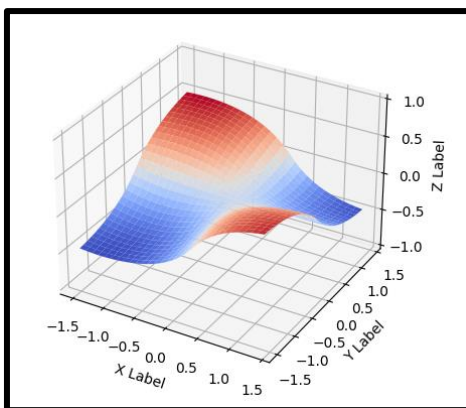
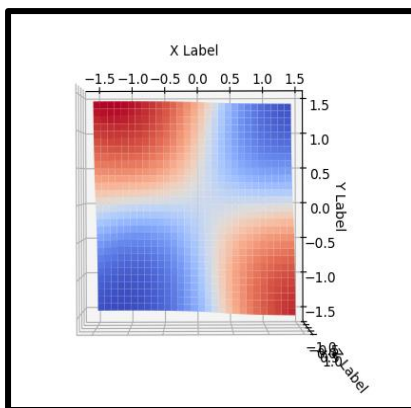




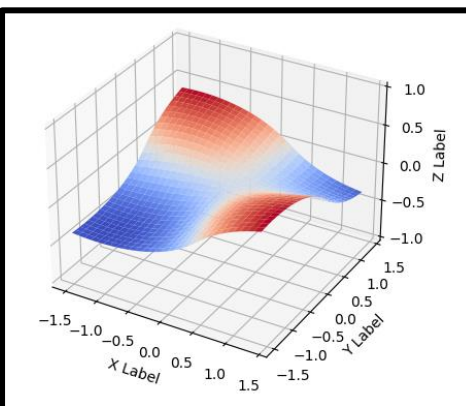
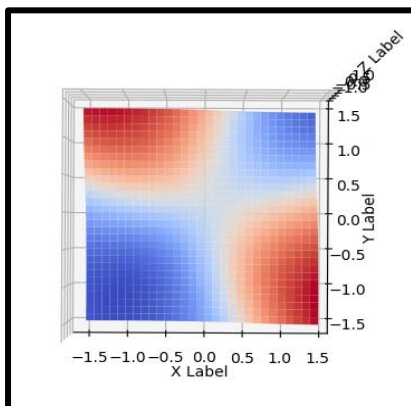
2.1.2.



2.1.3.



2.1.4.



Nota:

En esta prueba el único modelo que logro tener una eficacia de aprendizaje fue el modelo **2.1.2**, tiene un RMSE de 0.0592 y un tiempo de respuesta de 109 ms, en la gráfica (X, Y) se observa una que hay una distribución correcta



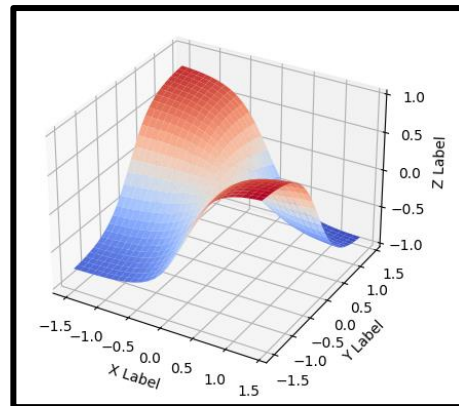
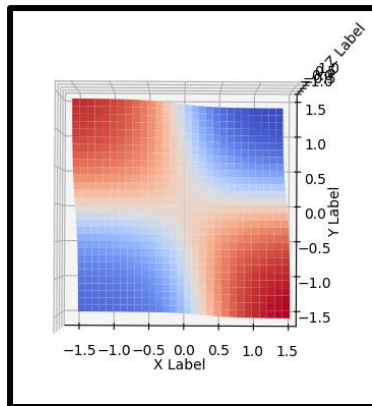
de los datos y se pueden ver rasgos de la figura que se busca (silla de montar) en el grafico 3D.

Prueba 2.2.

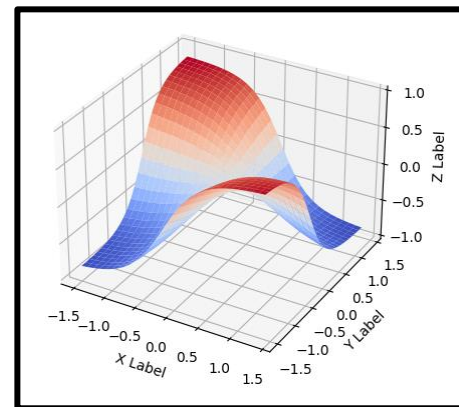
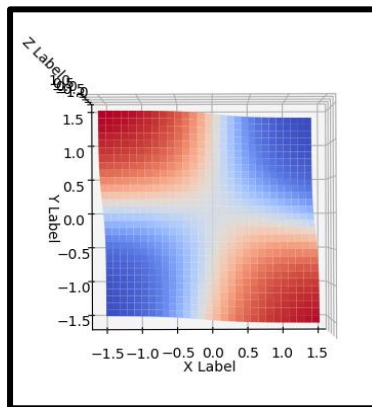
Test	In Funtion	# In Neuron	Out Funtion	# Out Neuron	Time (ms)	RMSE	Learn
2.2.1	Softmax	8	Tanh	1	166	0.0755	✓
2.2.2		16		1	154	0.0513	✓
2.2.3	Softmax	8	Softsign	1	129	0.2654	✓
2.2.4		16		1	155	0.2282	✓

Gráficos

2.2.1.

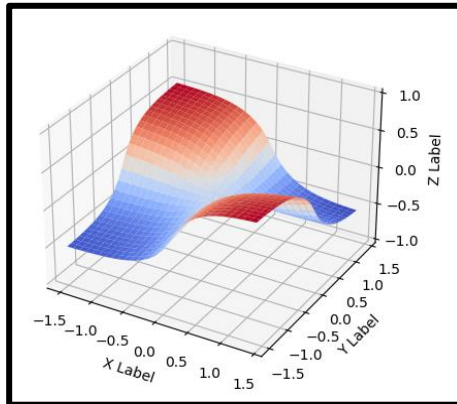
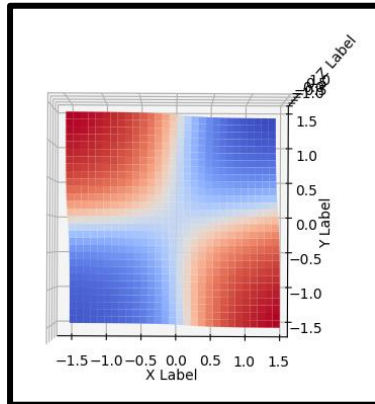


2.2.2.

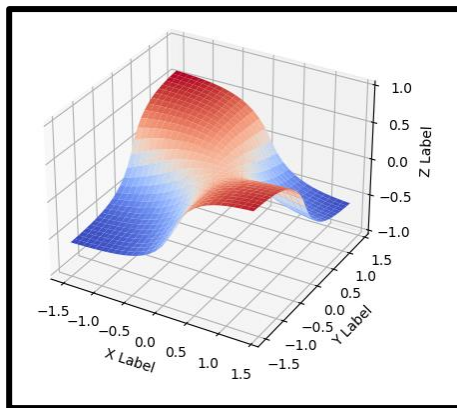
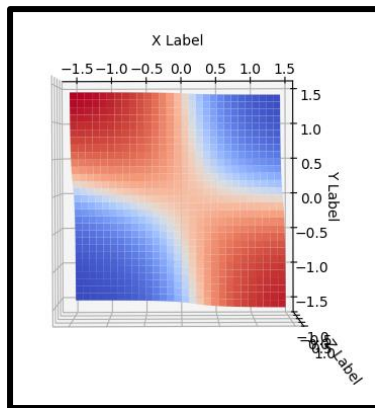




2.2.3.



2.2.4.



Nota:

En esta prueba todos los modelos obtienen un resultado de aprendizaje favorable, los modelos **2.2.1 y 2.2.2** obtienen una mejor RMSE, pero a su vez los modelos **2.2.3 y 2.2.4** obtiene un mejor tiempo de respuesta de la prueba 2.2.

Los modelos 2.2.2 obtiene una mejor distribución en el grafico (X,Y).

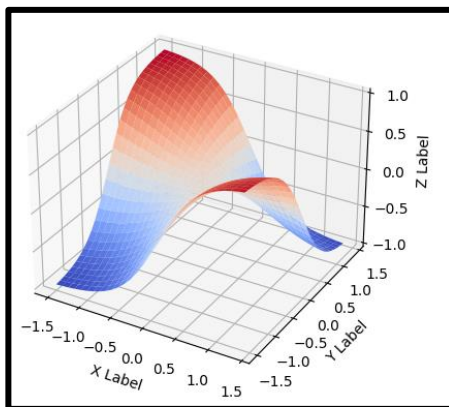
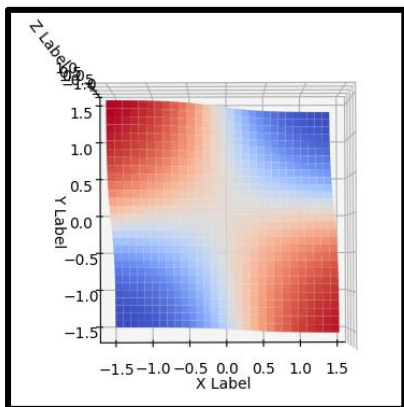
Prueba 2.3.

Test	In Funtion	# In Neuron	Out Funtion	# Out Neuron	Time (ms)	RMSE	Learn
2.3.1	Softplus	8	Tanh	1	180	0.0358	✓
2.3.2		16		1	138	0.0127	✓
2.3.3	Softplus	8	Softsign	1	150	0.1187	✓
2.3.4		16		1	170	0.1202	✓

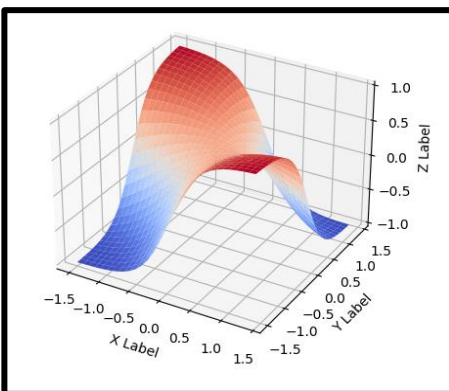
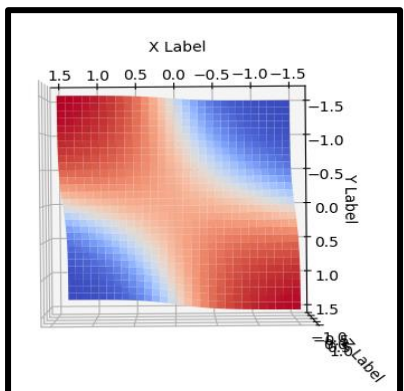


Gráficos

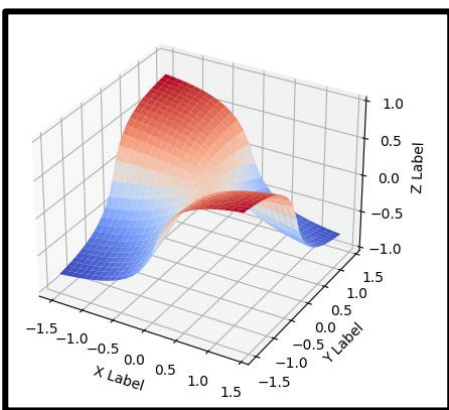
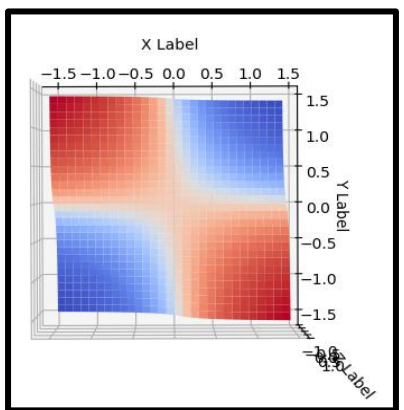
2.3.1.



2.3.2.

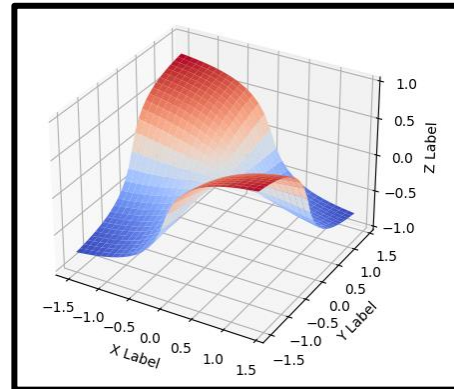
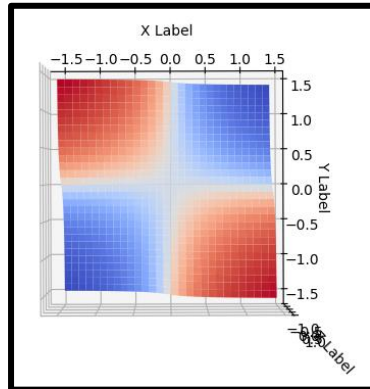


2.3.3.





2.3.4.



Nota:

En esta prueba todos los modelos obtienen un resultado de aprendizaje favorable, el modelo **2.3.2** obtienen el mejor RMSE (0.0127) y a su vez también obtiene el mejor tiempo de respuesta (138 ms) de la prueba 2.3, pero su gráfica no muestran una buena distribución de los datos en el gráfico (X, Y).

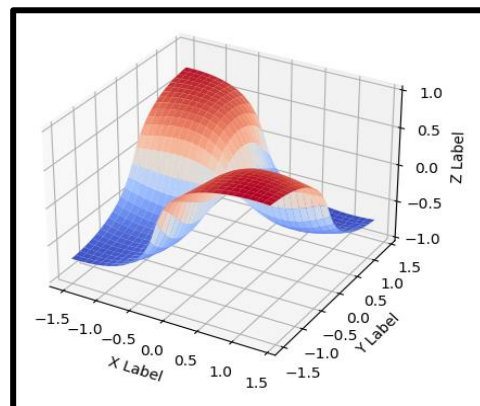
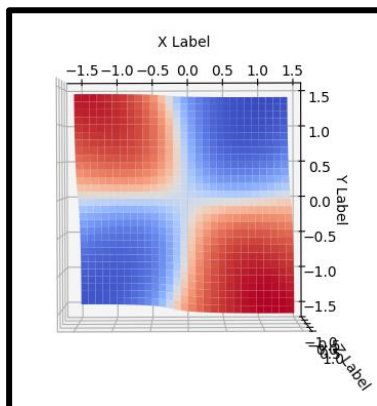
El modelo **2.3.4** tiene una muy buena distribución y diferenciación de los datos en la gráfica (X, Y).

Prueba 2.4.

Test	In Funtion	# In Neuron	Out Funtion	# Out Neuron	Time (ms)	RMSE	Learn
2.4.1	Tanh	8	Softsign	1	151	0.0907	✓
2.4.2		16		1	95	0.0881	✓
2.4.3	Tanh	8	Tanh	1	153	0.0112	✓
2.4.4		16		1	125	0.0063	✓

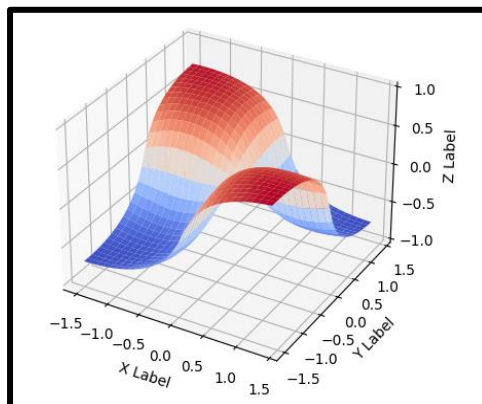
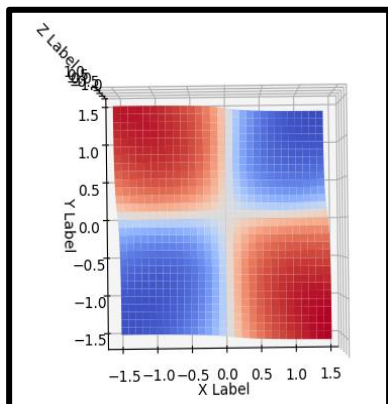
Gráficos

2.4.1.

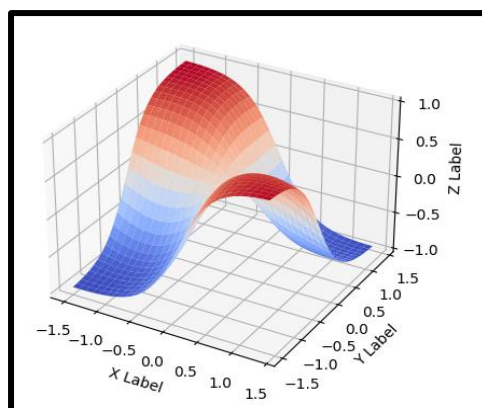
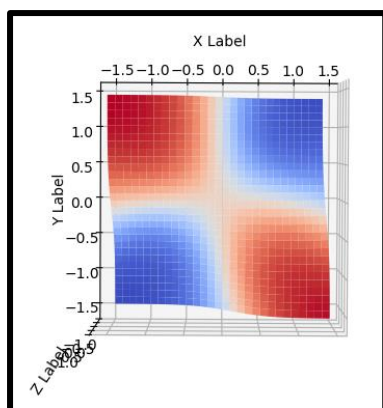




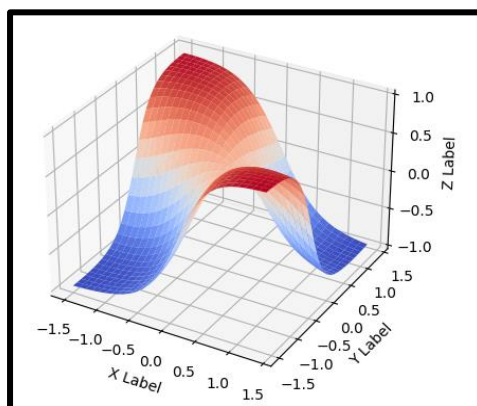
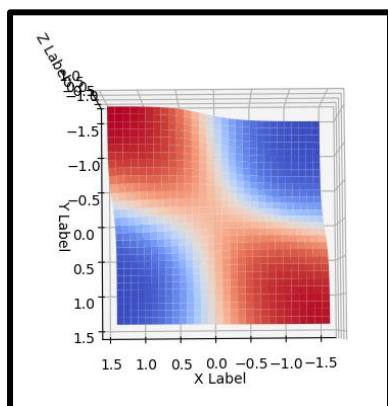
2.4.2.



2.4.3.



2.4.4.



Nota:

En esta prueba todos los modelos obtienen un resultado de aprendizaje favorable, el modelo **2.4.4** obtiene el mejor RMSE (0.0063) y el modelo **2.4.2** obtiene el mejor tiempo de respuesta (95 ms), estos últimos son los mejores



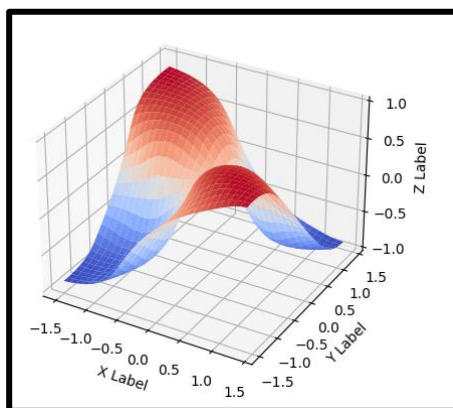
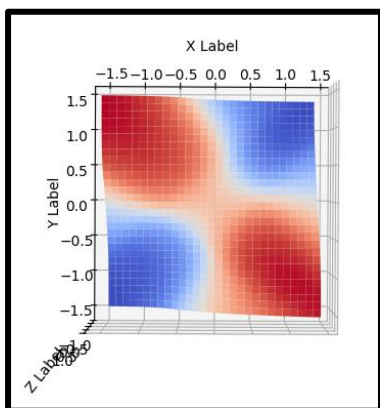
resultados en los argumentos de evaluación en **todas** las pruebas realizadas en la **Prueba 2**.

El modelo **2.4.2** tiene una muy buena distribución y diferenciación de los datos en la gráfica (X, Y)

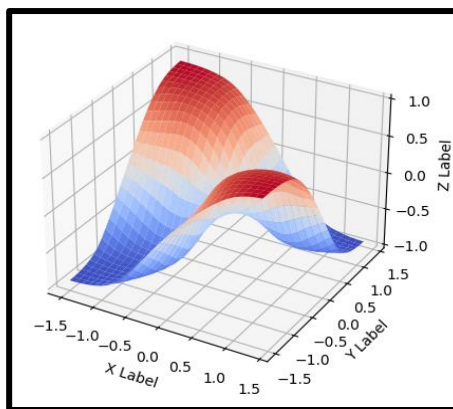
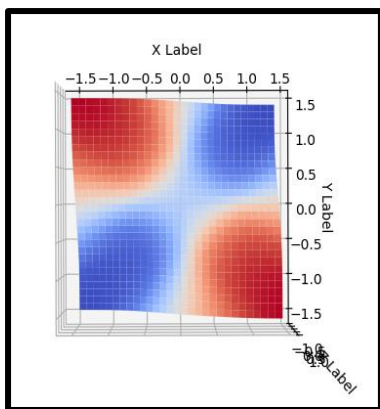
Prueba 2.5.

Test	In Funtion	# In Neuron	Out Funtion	# Out Neuron	Time (ms)	RMSE	Learn
2.5.1	Softsign	8	Tanh	1	103	0.0276	✓
2.5.2		16		1	145	0.0126	✓
2.5.3	Softsign	8	Softsign	1	159	0.1780	✓
2.5.4		16		1	165	0.1074	✓

2.5.1.

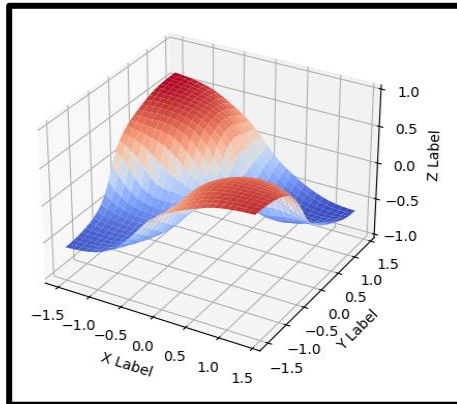
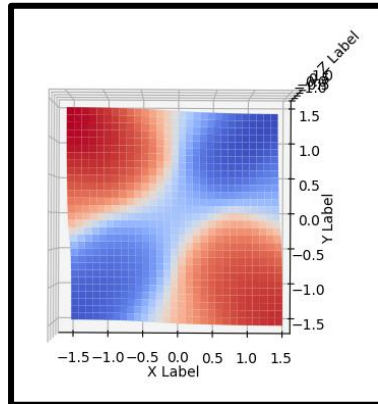


2.5.2.

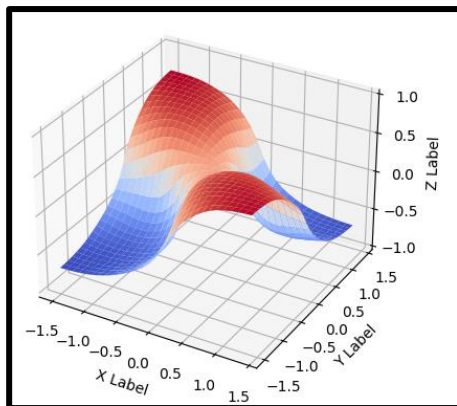
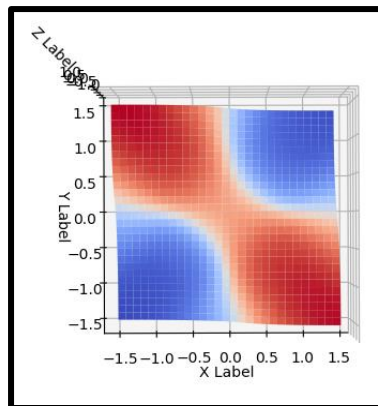




2.5.3.



2.5.4.



Nota:

En esta prueba todos los modelos obtienen un resultado de aprendizaje favorable, el modelo **2.5.2** obtiene el mejor RMSE (0.0126) y el modelo **2.5.1** obtiene el mejor tiempo de respuesta (103 ms) de la prueba 2.5.

En ningún modelo se denota una buena distribución y diferenciación de los datos en la gráfica (X,Y)

Aunque las gráficas 3D se observan rasgos de la figura que se busca (silla de montar) en el gráfico.

6.3. Tercera Prueba

En las pruebas anteriores se obtienen buenos resultados de aprendizaje en los modelos de red neuronal desarrollados, pero por fines de experimentales se agrega otra capa oculta a la estructura de la red neuronal.

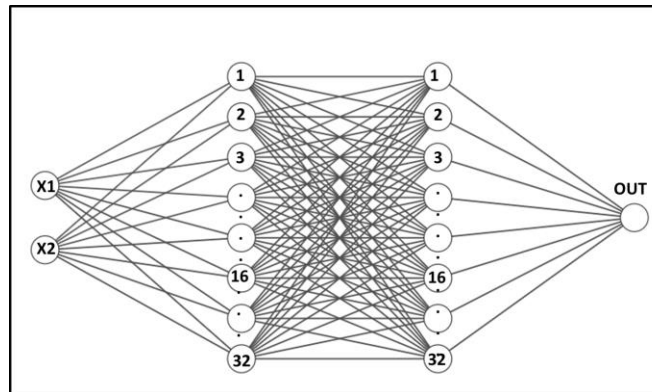
Se trabaja la combinación del modelo **2.4.2 (tanh - sofsign)**, ya que es quien presenta mejores resultados en la prueba anterior, se



aumenta el número de epochs a 2000 y se trabaja con las funciones "**tanh**" y "**softsign**" como funciones de activación en la capa de salida.

Se le dan combinaciones de 16 y 32 neuronas a las dos capas ocultas.

La nueva estructura de la red neuronal es la siguiente:



- Dos entradas X1 y X2
- Capa 1 oculta (16,32 neuronas)
- Capa 2 oculta (16,32 neuronas)
- Capa de salida (1 neurona)

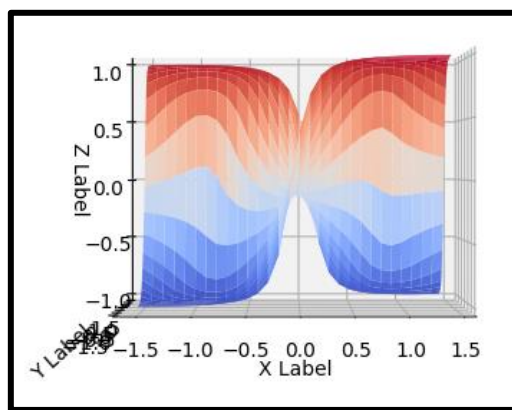
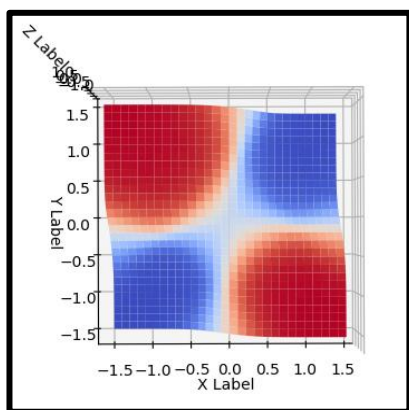
Prueba 3.

Test	L1 Funtion	# L1 Neuron	L2 Funtion	# L2 Neuron	Out Funtion	# Out Neuron	Time (ms)	RMSE	Learn
3.1	Tanh	16	Softsign	16	Tanh	1	92	0.000449	✓
3.2				32		1	102	0.000189	✓
3.3		32		16		1	112	0.000374	✓
3.4				32		1	102	0.000166	✓
3.5	Tanh	16	Softsign	16	Softsign	1	99	0.0187	✓
3.6				32		1	116	0.0161	✓
3.7		32		16		1	146	0.0239	✓
3.8				32		1	120	0.0112	✓

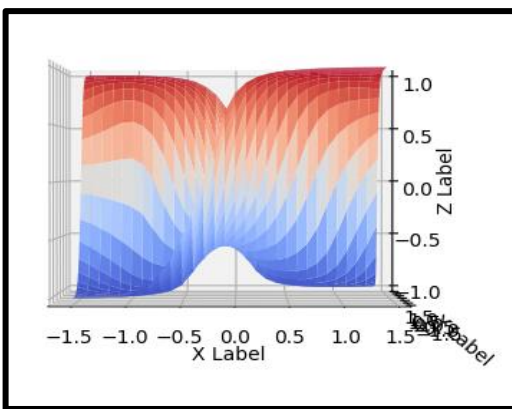
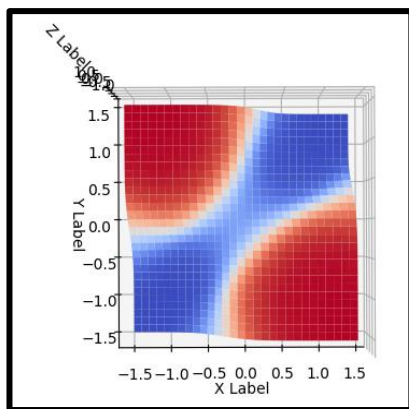


Gráficos:

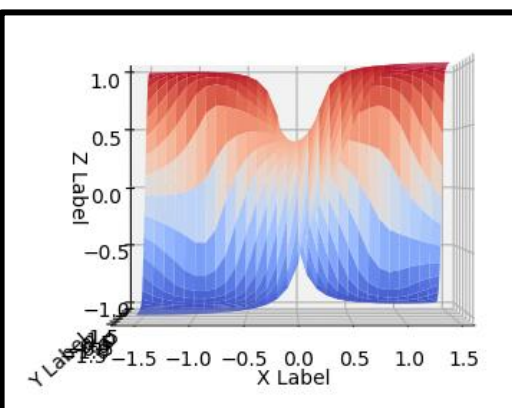
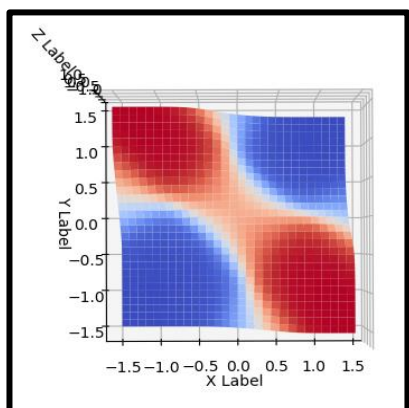
3.1.



3.2.

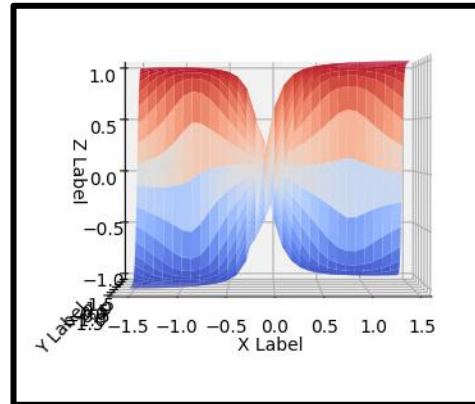
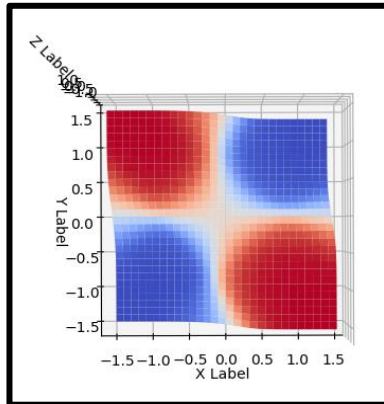


3.3.

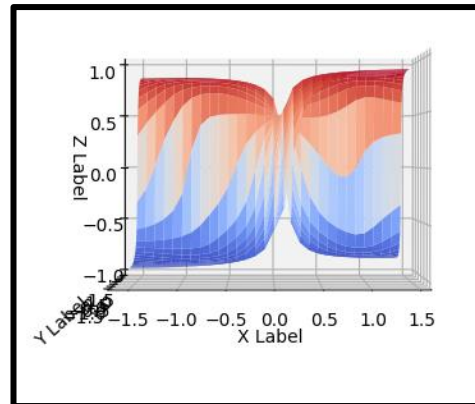
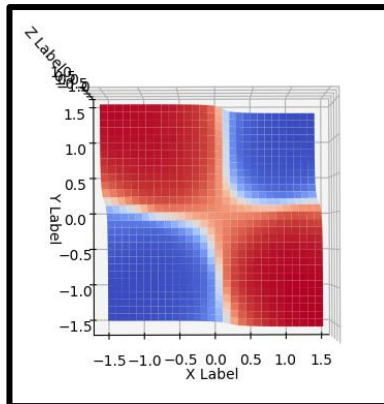




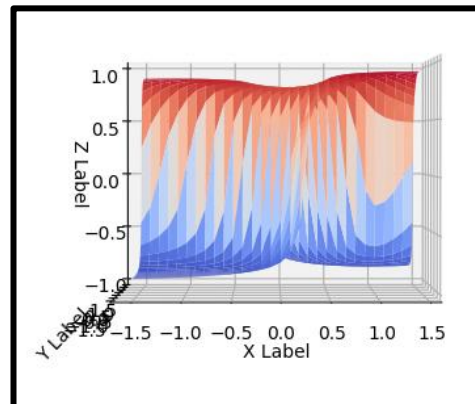
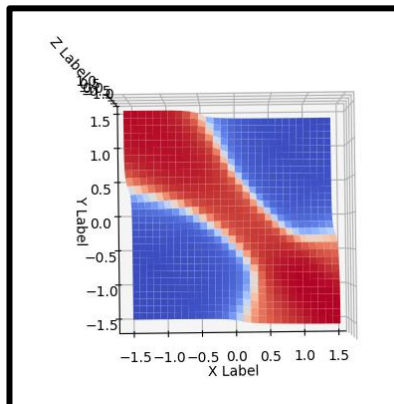
3.4.



3.5.

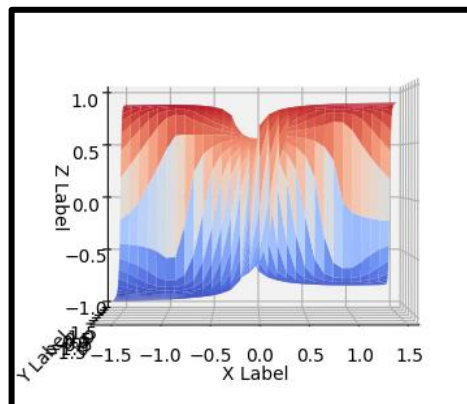
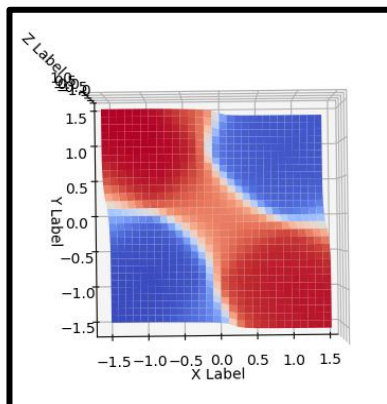


3.6.

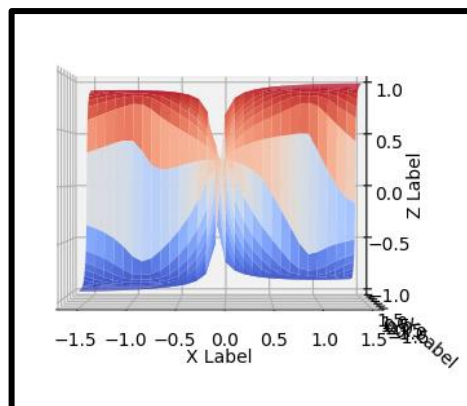
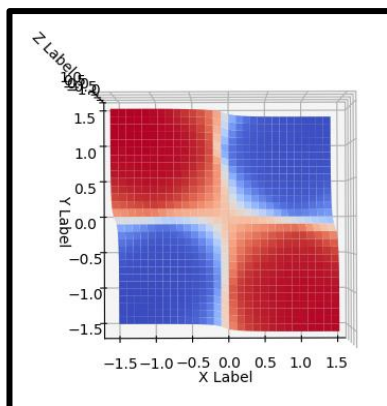




3.7.



3.8.



Nota:

El Modelo **3.1** obtiene el mejor tiempo de respuesta (92 ms), el modelo **3.4** obtiene el mejor RMSE (0.000166) de todas las pruebas realizadas en la prueba 3.

Gráficamente el modelo que menor se comporta en aprendizaje en el **3.4** se denota que hay una correcta distribución y separación de los datos de salida limitada por el punto cero (0) de los ejes X.

6.4. Evaluación de Modelos:

Por Argumentos de evaluación los mejores modelos de red neuronal serian:

- Modelo **3.4** el cual obtiene el mejor RMSE (0.000166)
- Modelo **3.1** obtiene el mejor tiempo de respuesta (92 ms)

Por visualización de gráficos



- Modelo **2.4.2** tiene una muy buena distribución y diferenciación de los datos en la gráfica (X, Y), y se observa los mejores rasgos de formación de la figura a buscar (silla de montar) y se demarca una limitación de los dichos datos por el punto Cero (0) del eje X.

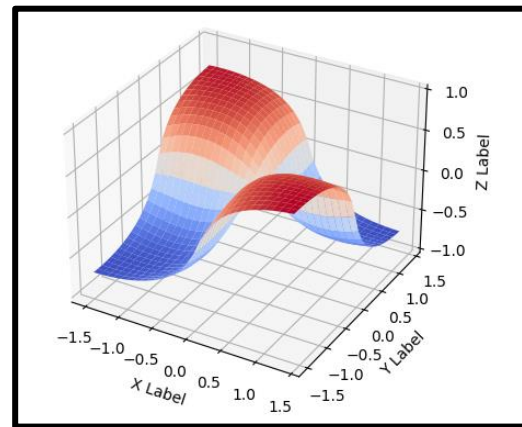
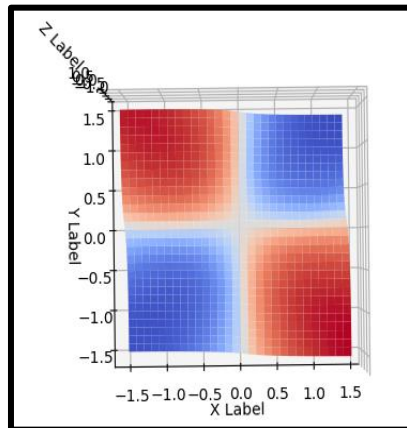
6.5. Modelo a Escoger:

- **Modelo 2.4.2.**

- **Tabla de Resultados**

Test	In Funtion	# In Neuron	Out Funtion	# Out Neuron	Time (ms)	RMSE	Learn
2.4.2	Tanh	16	Softsign	1	95	0.0881	<input checked="" type="checkbox"/>

- **Gráficos:**



Nota:

Se escoge el modelo **2.4.2** por ser quien mejor distribuyo los datos gráficamente se denota que hay un límite en los puntos ceros (0) de los ejes Y y X.

Además, cuenta con un tiempo de respuesta de 95 ms que es el segundo más rápido de todas las pruebas realizadas durante la práctica y un muy bajo RMSE 0.0881.

Al ser una red neuronal que cuenta con solo 2 capa (1 Oculta, 1 Salida) esto me permitirá un poco de ahorro en el esfuerzo computacional.



7. Conclusiones

- Para el correcto aprendizaje de cualquier red neuronal, ya sea sencilla o compleja, lo primordial siempre será conocer muy bien las características y parámetros del problema que se requiere sea aprendido y resuelto por la red neuronal, con esto aseguramos un correcto uso de argumentos que la red neuronal debe recibir y al mismo tiempo entregar.
- El desarrollo de una red neuronal tiene sus bases en la infinita combinación de estructuras que se pueden lograr para esta; a partir de aumentar o disminuir la cantidad de neuronas en las capas me permite obtener un cambio en los resultados obtenidos del aprendizaje de la red neuronal.
- La calidad de los resultados que puede entregar una red neuronal depende de la cantidad y calidad de los datos con lo que es alimentado el modelo.
- La idea principal de conseguir la implementación de redes neuronales artificiales; es que estas permiten el desarrollo de sistemas de control en tiempo real, lo que brinda la posibilidad de llevarlas a la práctica en sistemas de control inteligente.