POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

E – Control Inteligente 1

Informe Practica N.º 2

1. **Nombre:**

   Jeferson Gallego Chaverra

2. **Semestre:**

   2023-2

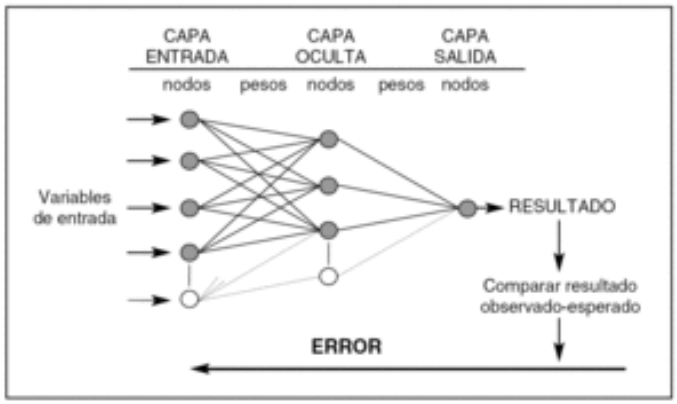
3. **Nombre Practica:**

   Bases de datos dinámicas en redes MLP

4. **Resumen**

En esta práctica se implementa una red MLP (Multi-layer Perceptron) o Perceptrón Multicapa, este tipo de red neuronal se utiliza para el aprendizaje supervisado y automático en sistemas RNA.

La estructura de una red MLP se puede identificar en la siguiente figura:



Donde:

- **Capa de entrada:** Constituida por aquellas neuronas que introducen los patrones de entrada en la red. En estas neuronas no se produce procesamiento.
- **Capas ocultas:** Formada por aquellas neuronas cuyas entradas provienen de capas anteriores y cuyas salidas pasan a neuronas de capas posteriores.

- **Capa de salida:** Neuronas cuyos valores de salida se corresponden con las salidas de toda la red.

Para el entrenamiento de la red MLP se hace uso de uso de las librerías "numpy" la cual permite simular el comportamiento de un sistema a partir de la solución analítica de la ecuación de un sistema de primer orden.

$$\frac{\boldsymbol{Y(s)}}{\boldsymbol{U(s)}} = \frac{\boldsymbol{K}}{\boldsymbol{\tau + 1}};\ \boldsymbol{T = delta}$$

Donde Y(k) se expresa como:

$$\boldsymbol{y(k) = ((((u * k) - y) * delta) / tao) + y}$$

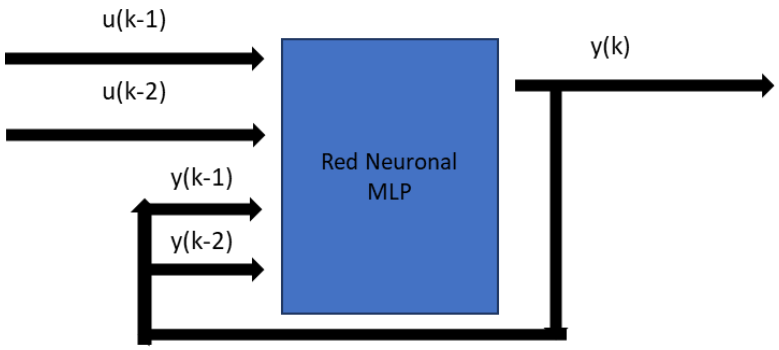
A partir de una función "Time ()" se cambian los valores de u (Aumentar - Disminuir) cada 30 segundos de simulación, se obtiene una base de datos de 3 vectores Tiempo (t), Entrada (u) y Salida (y).

Se toman los vectores **"u"** e **"y"** para poder expresar los 5 vectores necesarios para la red MLP (**yk, yk_1, yk_2, uk_1, uk_2**).

Donde:

- y(k) = Datos de Salida.
- y(k_1) = Datos de Salida con 1 instante anterior.
- u(k_2) = Datos de Salida con 2 instantes anteriores.
- u(k_1) = Datos de Entrada con 1 instante anterior.
- u(k_2) = Datos de Entrada con 2 instantes anteriores.

La Estructura de la Red MLP que se desarrolla durante la práctica es la siguiente:



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

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

Learn: La red aprende Sí ☑ o No ☒

Por último, se toman los datos obtenidos por la simulación y los datos obtenidos por el modelo de la red neuronal y se procede a graficar a ambos en función del tiempo así se tendrá una forma de evaluación visual de si la red consigue aprende o no aprende.

5. Código

Se envían 3 archivos .py con sus respectivos comentarios:

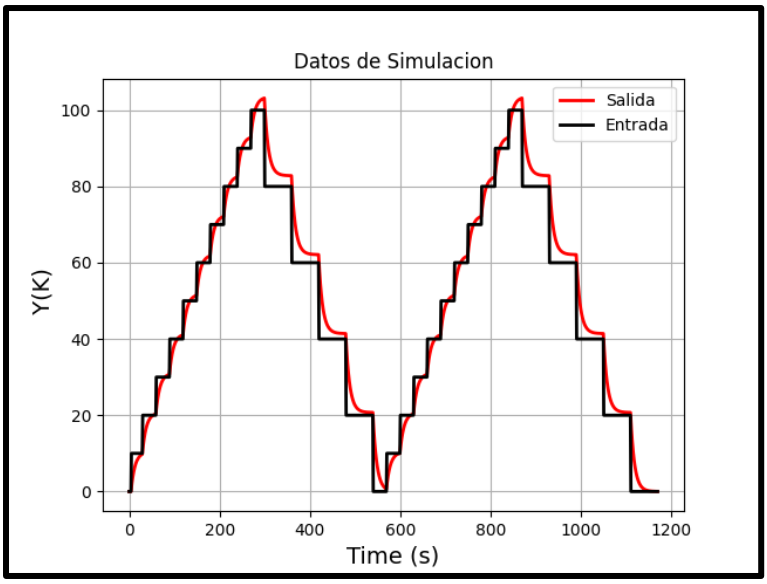
5.1. Datos.py
5.2. Entrenamineto.py
5.3. Verificacion.py

6. Resultados Obtenidos:

6.1. Datos

Se hace la toma de datos a través de la simulación, con un total de 1170 datos, durante la simulación se generan 10 aumentos del escalón a un 10%, posterior a esto se disminuye 5 veces el escalón a un 20%, esta secuencia se repite 2 veces.

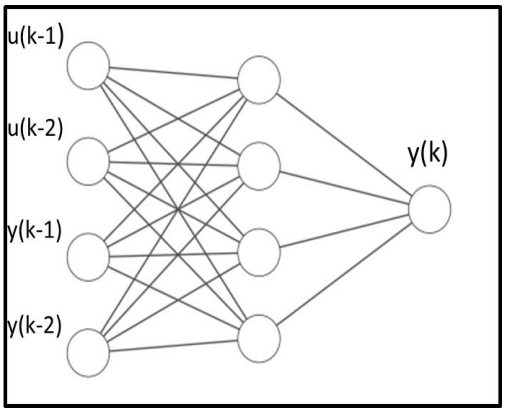
$$\frac{\boldsymbol{Y(s)}}{\boldsymbol{U(s)}} = \frac{\boldsymbol{K}}{\boldsymbol{\tau + 1}} = \frac{\boldsymbol{1.035}}{\boldsymbol{0.918s + 1}} \; \boldsymbol{T = 0.1}$$



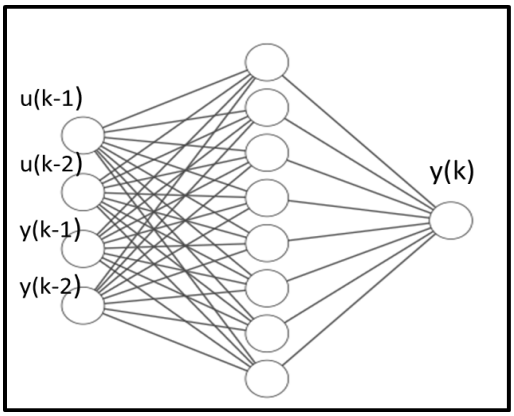
6.2. Primera Prueba

Para el trabajo de aprendizaje de la red se tomarán las siguientes estructuras de redes neuronales.

a.



b.



- 4 entradas yk_1, yk_2, uk_1, uk_2.
- Capa 1 oculta (4,8, neuronas).
- 1 Salida yk.

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 0 y 1.

Se hicieron 10 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 (4,8).

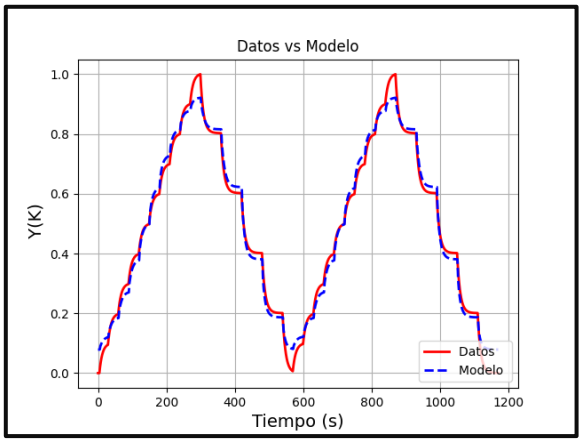
El número de epochs para el aprendizaje es de 1000.

Prueba 1.1 y 1.2

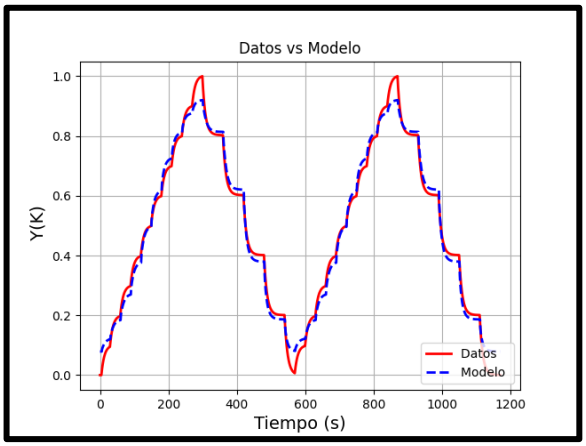
Test	In Funtion	# In Neuron	Out Funtion	# Out Neuron	RMSE	Learn	Note
1.1	Relu	4	Sigmoid	1	0.15858	☒	La función Sigmoid no es óptima como función de salida para la solución de problema establecido
1.2		8		1	0.15895	☒	

POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

1.1.



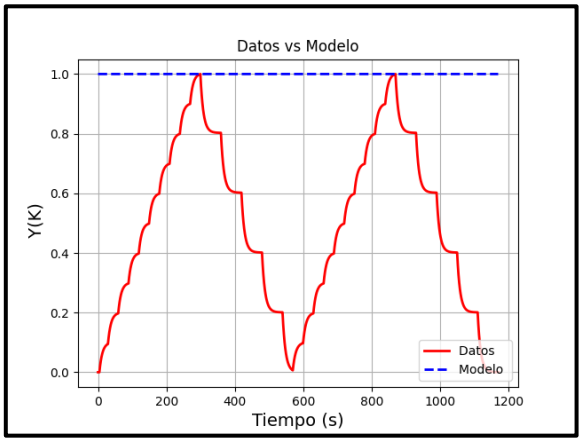
1.2.



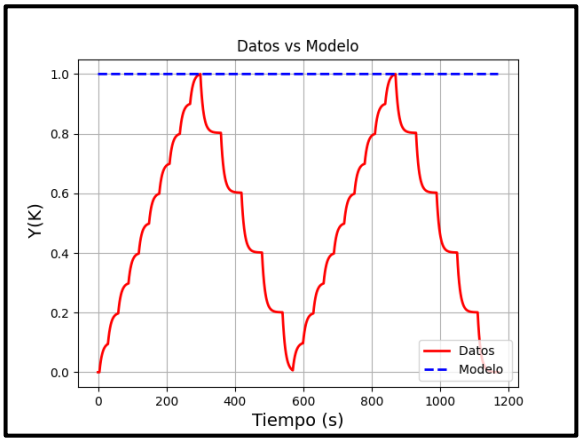
Prueba 1.3 y 1.4

Test	In Funtion	# In Neuron	Out Funtion	# Out Neuron	RMSE	Learn	Note
1.3	Relu	4	Softmax	1	-	☒	La función Softmax no es óptima como función de salida para la solución de problema establecido
1.4		8		1	-	☒	

1.3.



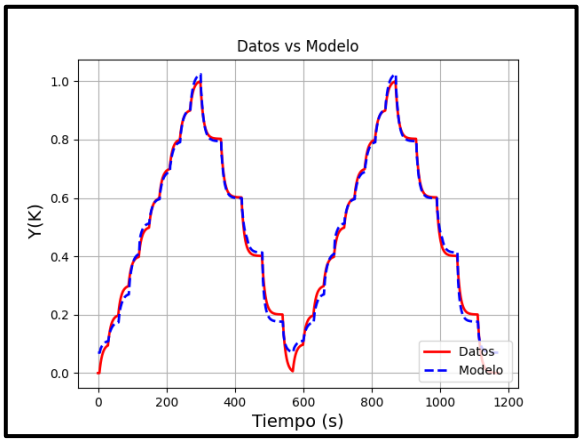
1.4.



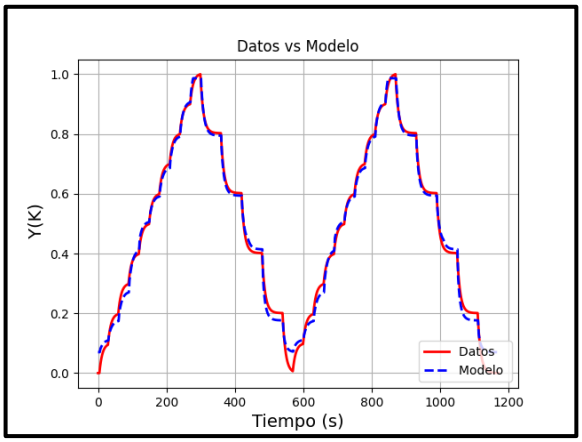
Prueba 1.5 y 1.6

Test	In Funtion	# In Neuron	Out Funtion	# Out Neuron	RMSE	Learn	Note
1.5	Relu	4	Softplus	1	0.16056	☑	El modelo Aprende a partir de las 4 neuronas en la capa oculta
1.6		8		1	0.16164	☑	

1.5.



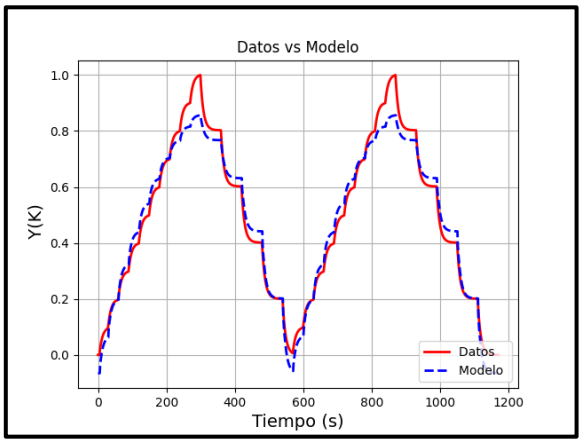
1.6.



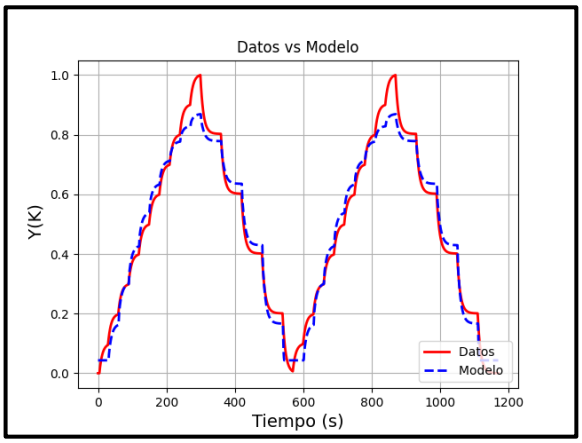
Prueba 1.7 y 1.8

Test	In Funtion	# In Neuron	Out Funtion	# Out Neuron	RMSE	Learn	Note
1.7	Relu	2	Tanh	1	0.15769	☒	La función Sigmoid no es óptima como función de salida para la solución de problema establecido
1.8		4		1	0.15763	☒	

1.7.



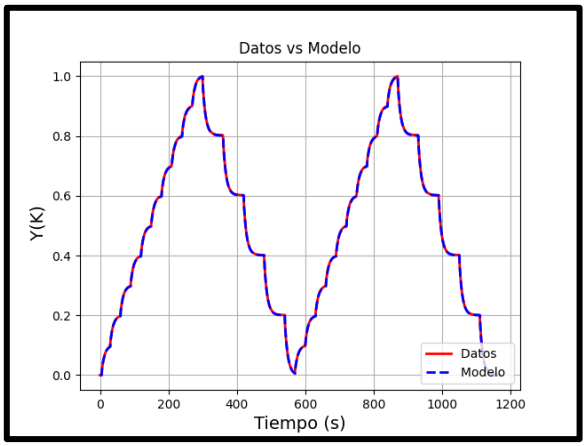
1.8.



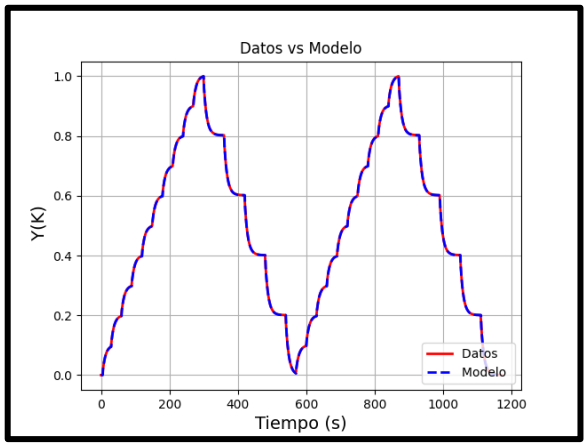
Prueba 1.9 y 1.10

Test	In Funtion	# In Neuron	Out Funtion	# Out Neuron	RMSE	Learn	Note
1.9	Relu	2	Relu	1	0.16358	☑	El modelo Aprende a partir de las 4 neuronas en la capa oculta
1.10		4		1	0.16344	☑	

1.9.



1.10.



Nota:

Se llega a la conclusión que la funciones **"relu"** y "**softplus"** son las únicas óptimas para ser función de activación para la capa de salida.

6.3. Segunda Prueba

Se realizan varias pruebas de aprendizaje, para así hallar que modelo de red neuronal es más eficaz.

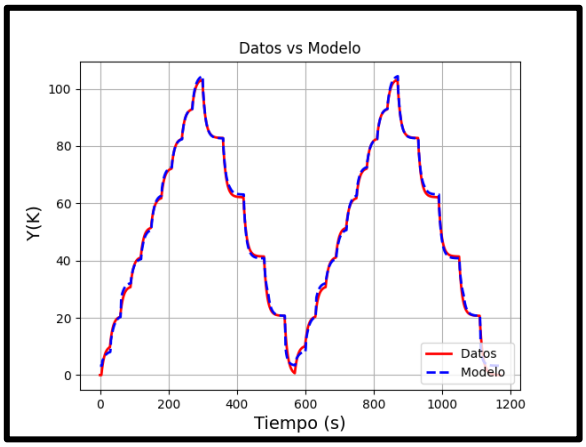
Se trabaja la combinación del modelo **1.9 (relu - relu**), ya que es quien presento mejores resultados en la prueba anterior, se aumenta el número de epochs a 2000 y se agrega una capa de salida que trabaja con las funciones **"relu" y "softplus"** como funciones de activación.

Prueba 2:

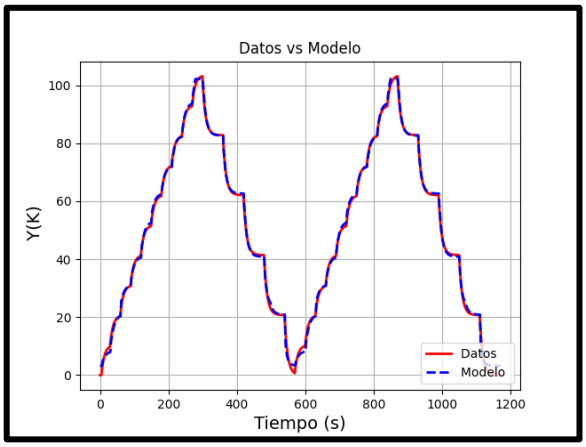
Test	L1 Funtion	# L1 Neuron	L2 Funtion	# L2 Neuron	Out Funtion	# Out Neuron	Time (ms)	Loss	RMSE	Learn
2.1	Relu	4	Relu	2	Softplus	1	103	0.0001012	0.16358	☑
2.2				4		1	102	0.0001315	0.16358	☑
2.3				6		1	112	0.0000847	0.16201	☑
2.4				8		1	102	0.0005417	0.16000	☑
2.5	Relu	4	Relu	2	Relu	1	99	0.0000070	0.16329	☑
2.6				4		1	120	0.0000085	0.16354	☑
2.7				6		1	146	0.0000071	0.16382	☑
2.8				8		1	116	0.0000101	0.16334	☑

Gráficos:

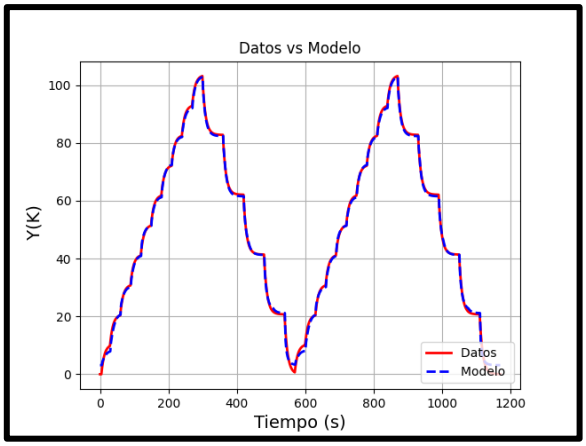
2.1.



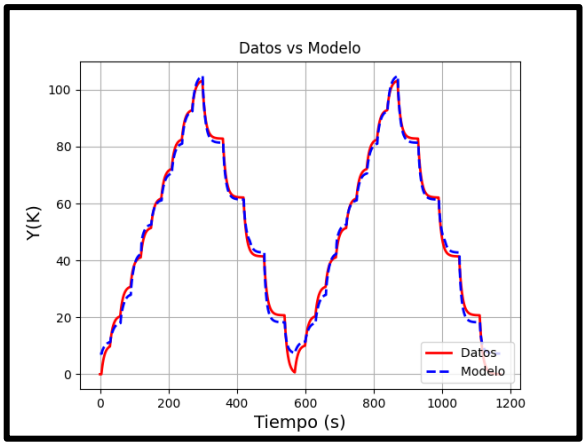
2.2.



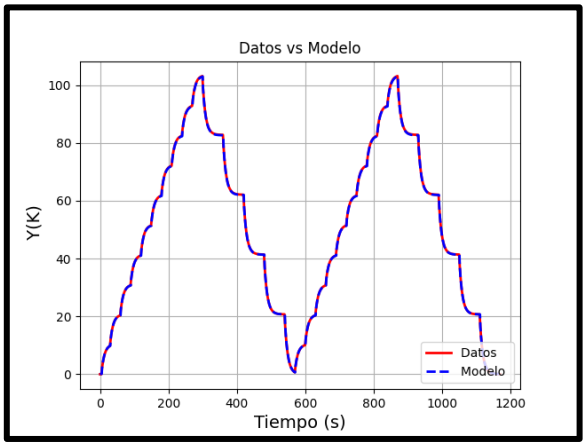
2.3.



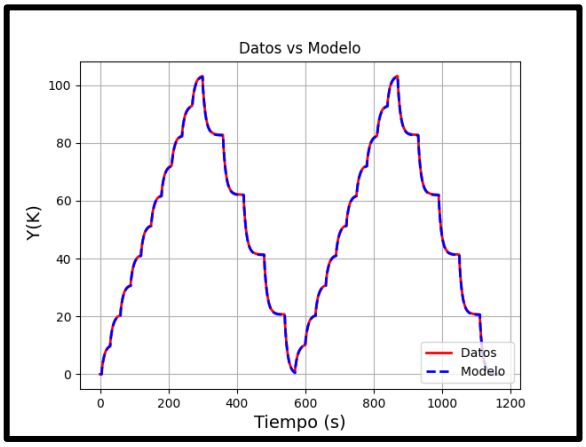
2.4.



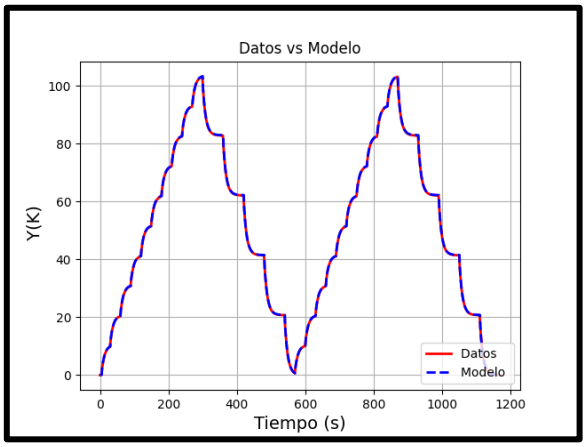
2.5.



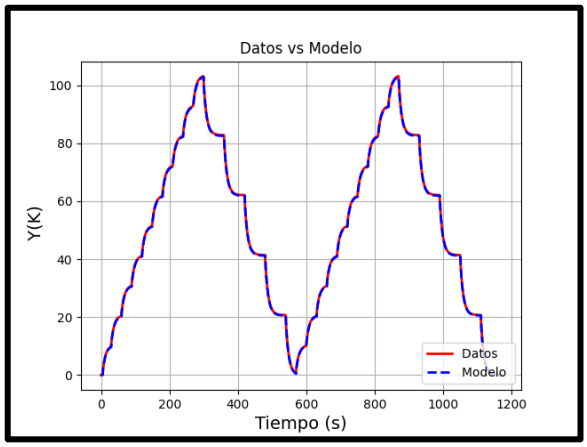
2.6.



2.7.



2.8.



6.4. Evaluación de Modelos:

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

- Modelo **2.4** el cual obtiene el mejor RMSE (0.16000)
- Modelo **2.5** obtiene el mejor Loss (0.0000070)
- Modelo **2.5** obtiene el mejor tiempo de respuesta (99 ms)

.

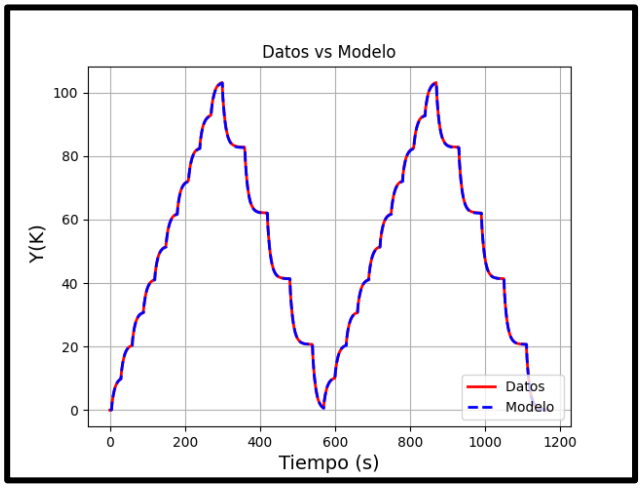
6.5. Modelo a Escoger:

- **Modelo 2.5**

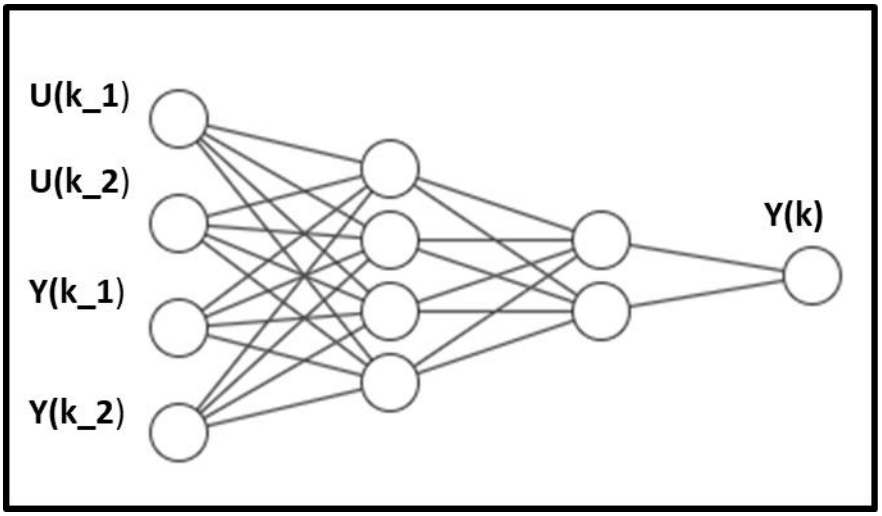
- **Tabla de Resultados**

Test	L1 Funtion	# L1 Neuron	L2 Funtion	# L2 Neuron	Out Funtion	# Out Neuron	Time (ms)	Loss	RMSE	Learn
2.5	Relu	4	Relu	2	Relu	1	99	0.0000070	0.16329	☑

- **Gráfico:**



- **Estructura:**



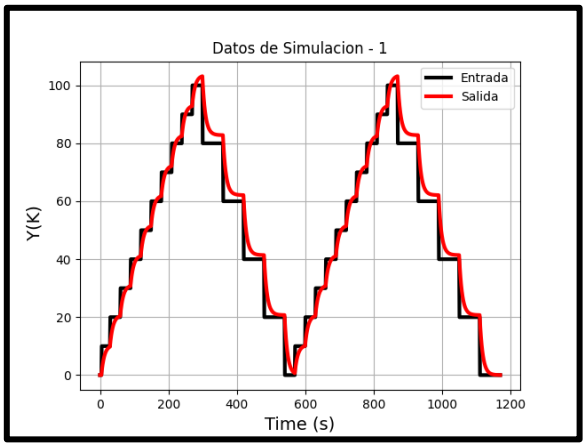
Nota: Se escoge el modelo **2.5** por ser quien cuenta con un tiempo de respuesta de 99 ms que es el más rápido de todas las pruebas realizadas durante la práctica y un muy bajo RMSE 0,16329 y Loss 0.0000070.

Al ser una red neuronal que cuenta con solo 3 capa (2 Oculta, 1 Salida) esto me permitirá un poco de ahorro en el esfuerzo computacional.

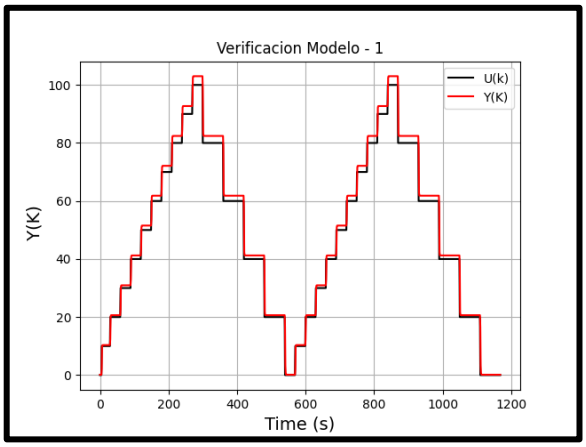
6.6. Verificación de Modelo

Para la verificación del modelo se hace uso 3 bases datos, la original con la que se entrenó a la red neuronal y otras dos donde se varían los escalones de una manera diferente a la primera simulación.

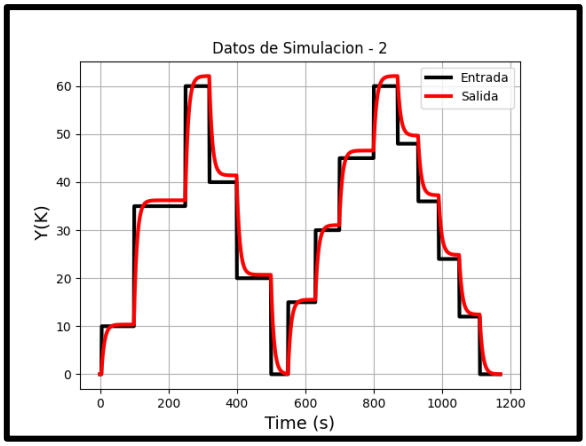
Data 1:



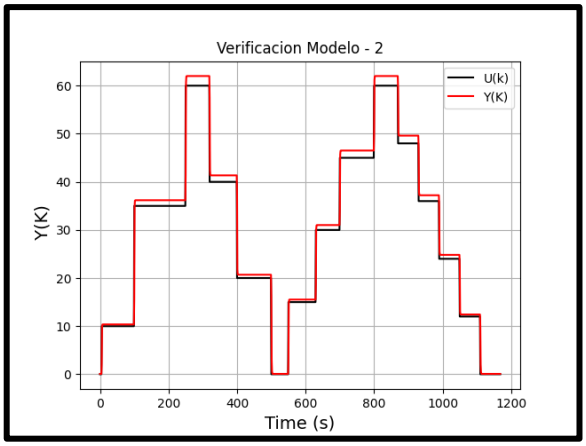
Verificación 1:



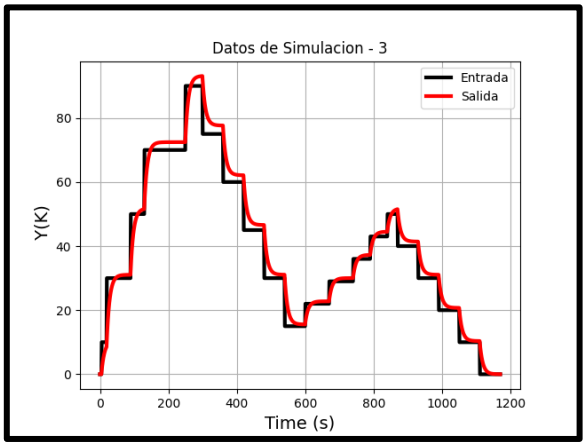
Data 2:



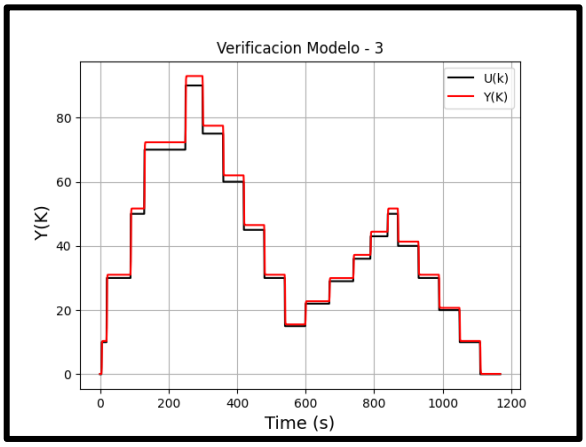
Verificacion 2:



Data 3:



Verificacion 3:



Nota: Se Observa que en las 3 pruebas de verificación yk responde a los cambios de uk, la red neuronal entrega salidas para yk las cuales se mantiene en los valores de los datos simulados, esto demuestra que la red ha sido entrenada correctamente, ya que no solo tiene la capacidad de hacerlos con los datos de entrenamiento, sino también con las 2 bases de datos.

7. Conclusiones

- La mayoría de los modelos obtenidos en esta práctica, obtuvieron buenos resultados en contra del problema que se está desarrollando, son redes neuronales muy simples, las cuales tienen pocas capas y pocas neuronas.
- La capacidades de las redes neuronales son inmensas, según la teoría y la base matemática con la que sea desarrollada, se pueden llegar a soluciones de problemas que impliquen condiciones mucho más adversas.