



UNIVERSIDAD DE CHILE

UNIVERSIDAD DE CHILE

EL7012

CONTROL INTELIGENTE DE SISTEMAS, OTOÑO

Ejercicio N°1

Autor:

Elias Obrequé
Gustavo Ceballo
Maibeth Sánchez

13 de mayo de 2020

Índice

1. Introducción	1
2. Problema 1	2
2.1. Generación de Datos	3
2.2. Modelos de predicción	5
2.2.1. Modelo lineal	5
2.2.2. Modelo difuso Takagi-Sugeno Tipo-1	6
2.2.3. Modelo de red neuronal	7
3. Resultados	16
4. Conclusión	17

1. Introducción

Esto es una prueba para el trabajo offline.

2. Problema 1

Considere la siguiente serie no lineal dinámica:

$$\begin{aligned} y(k) = & (0,8 - 0,5\exp\{-y^2(k-1)\})y(k-1) \\ & - (0,3 + 0,9\exp\{-y^2(k-1)\})y(k-2) \\ & + u(k-1) + 0,2u(k-2) + 0,1u(k-1)u(k-2) + e(k) \end{aligned} \quad (1)$$

donde el ruido del sistema

$$e(k) = 0,5\exp\{-y^2(k-1)\}\beta(k) \quad (2)$$

depende del estado previo de la salida del modelo, y $\beta(k)$ es un ruido blanco.

Como usted sabe existen varias técnicas que se pueden emplear para la modelación a partir de estos datos, por lo que debe seleccionar el tipo de modelo más adecuado para este tipo de sistema. Para este trabajo se le pide detallar la metodología utilizada para:

- Generar 600 datos a partir de esta serie. Considere 55 % para entrenamiento, 25 % test y 20 % validación.
- Obtener un modelo de predicción lineal, difuso tipo-1 (T&S) y neuronal para la salida. Evaluar las predicciones a 1, 8 y 16 pasos. Comparar el desempeño de todos los modelos a partir de las métricas más apropiadas tales como RMSE, MAPE, MAE, entre otras. Comente.
- Construir el intervalo de predicción de los modelos obtenidos en b) utilizando el método de la covarianza.
- Evaluar los intervalos de predicción obtenidos en b) realizando predicciones a 1, 8, y 16 pasos. Comparar el desempeño de los modelos a partir de las métricas más apropiadas tales como ancho del intervalo, probabilidad de cobertura, entre otras.
- Construir el intervalo de predicción del modelo difuso encontrado en a) con el método de optimización min-max. Compare este intervalo de predicción con el intervalo obtenido utilizando el método de la covarianza. Comente.
- Construir el intervalo de predicción neural utilizando el método de Joint Supervision. Compare con los métodos anteriores.
- Seleccione el modelo más apropiado y justifique.

2.1. Generación de Datos

En esta etapa es necesario generar datos que representen la dinámica del sistema en la mayor cantidad de rangos de operación posibles, ya que el modelo obtenido tiene un ancho de banda acotado, y por lo tanto las dinámicas definidas por fuera de dicha banda podrían no ser representadas adecuadamente. Para lo cual se debe diseñar una entrada $u(k)$ que excite a la planta en el rango de frecuencias en que se encuentran los fenómenos de interés.

En este trabajo se propone el uso de señales binarias pseudo aleatorias (Pseudo Random Binary Signal, PRBS), ya que es una de las señales más utilizadas en identificación de sistemas. Esta es una señal periódica, determinística y que posee principalmente propiedades similares al ruido blanco (contenido muy rico en frecuencias)

Para general la señal se suponen los siguientes parámetros de interés $f_{min} = 0,2Hz$, $f_{max} = 1Hz$ y tiempo de muestreo $T_s = 0,01$. Con los parámetros anteriores, y utilizando la expresión

$$n = \frac{\log(f_c/f_{min} + 1)}{\log(2)} \quad (3)$$

con $f_c = 2,5 * f_{max} = 2,5Hz$, se genera una PRBS de orden $n = 4$, por lo que el largo máximo corresponde a $N = 2^n - 1 = 15$. A su vez, la cantidad de muestras por bit son $N_s = 40$. Luego, el tiempo de un bit, $\Delta t = N_s * T_s = 0,4s$, por lo que la PRBS dura en total 6s y debe ser replicada 400 veces con diferentes condiciones iniciales para obtener los 6000 datos de interés. Finalmente se genera la APRBS variando la amplitud aleatoriamente de la PRBS generada, Fig.1 y se aplica a la serie no lineal como se muestra en la Fig.2.

Una vez obtenidos los datos experimentales de entrada-salida, éstos son clasificados en tres conjuntos con distinta información: datos de entrenamiento, datos de validación y datos de prueba; esto con el fin de evaluar adecuadamente los modelos generados. El conjunto de entrenamiento se utiliza para determinar los parámetros del modelo. El conjunto de prueba permite comparar distintas estructuras de los modelos generados. Finalmente, el conjunto de validación permite verificar el sobreajuste del modelo óptimo obtenido, evaluándolo en un nuevo conjunto de datos (distintos a los datos del conjunto de entrenamiento y validación), analizando su capacidad de generalización. En este caso se utiliza una división de 55 % de los datos para entrenamiento, 25 % para prueba y 20 % validación.

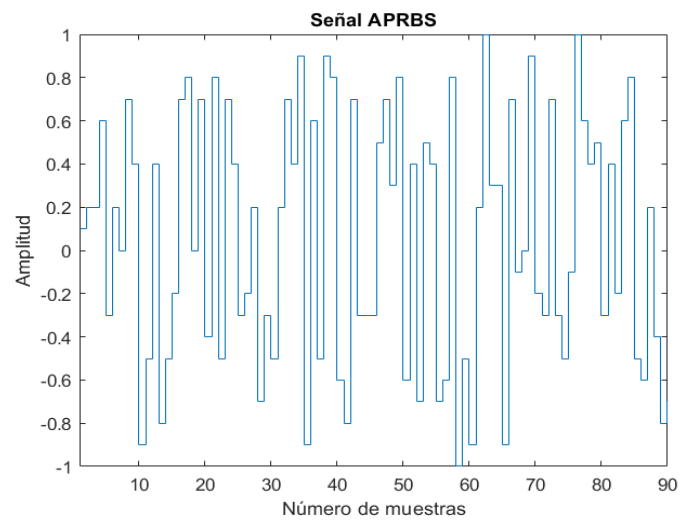


Figura 1: Señal APRBS con Amplitud entre -1 y +1.

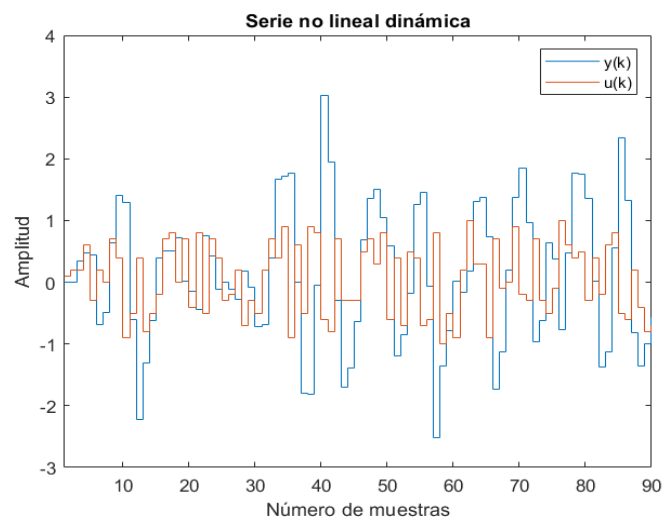


Figura 2: Respuesta de la serie no lineal.

2.2. Modelos de predicción

2.2.1. Modelo lineal

En este caso, supondremos que se ajustará un modelo lineal suponiendo que el sistema real es lineal con ruido blanco gaussiano aditivo, es decir,

$$y(k) = a_1y(k-1) + a_2y(k-2) + b_1u(k-1) + b_2u(k-2) + e(k) \quad (4)$$

Luego, se propone un modelo lineal para llevar a cabo la predicción a 1 paso, de modo tal que:

Predicción a 1 paso:

$$\hat{y}(k) = \hat{a}_1y(k-1) + \hat{a}_2y(k-2) + \hat{b}_1u(k-1) + \hat{b}_2u(k-2) \quad (5)$$

Este modelo no considera un valor constante o bias dado el supuesto que el sistema es lineal con ruido blanco aditivo. En caso que se sospechara que existe un bias o tendencia (trend) en el sistema, se puede agregar otro vector de unos a la matriz de regresores (o matriz de información).

Para llevar a cabo la estimación de los parámetros del modelo se utilizó la técnica de mínimos cuadrados, es decir:

$$\hat{\theta} = (Xent^T * Xent)^{-1} * Xent^T * \hat{y}(k) \quad (6)$$

En que $\hat{\theta} = [\hat{a}_1 \quad \hat{a}_2 \quad \hat{b}_1 \quad \hat{b}_2]^T$ es el vector de parámetros y $Xent$ es la matriz de regresores con los valores de las n muestras ordenados por filas.

Los valores que se obtuvieron de los parámetros fueron los siguientes:

$$\hat{\theta} = \begin{pmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \hat{b}_1 \\ \hat{b}_2 \end{pmatrix} = \begin{pmatrix} 0,8601 \\ -0,6930 \\ 0,9724 \\ 0,3486 \end{pmatrix} \quad (7)$$

A continuación, en la Tabla 1, se presentan las métricas de bondad del ajuste o errores en los diversos conjuntos de datos, a saber, conjunto de datos de entrenamiento, prueba o test y validación.

Tabla 1: Errores o Métricas de bondad de ajuste a 1 paso

Métricas	Conjunto En-trenamiento	Conjunto de Prueba	Conjunto de Valida-ción
RMSE	0.0115	0.019	0.0241
MAPE	123.1323	101.9383	169.4761
MAE	0.3313	0.3402	0.3524

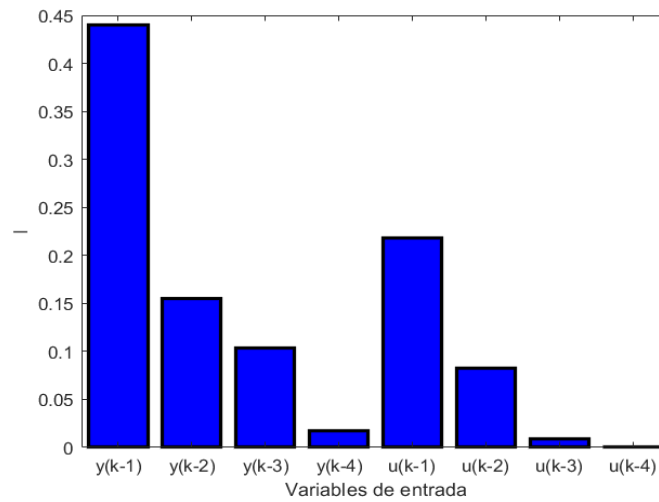


Figura 3: Índice de Sensibilidades.

2.2.2. Modelo difuso Takagi-Sugeno Tipo-1

Para seleccionar las variables que actúan como entrada al sistema difuso, se realiza un análisis de sensibilidad. Suponiendo una estructura del modelo inicial difuso con 8 variables de entrada $y(k-1), \dots, y(k-4), u(k-1), \dots, u(k-4)$. En la Fig. 3 se muestran los índices de las sensibilidades del modelo inicial para las 8 variables de entrada, comprobándose que las variables $y(k-3)$, $y(k-4)$, $u(k-1)$ y $u(k-4)$ presentan menores índices de las sensibilidades, por lo cual no son incluidos en el modelo difuso.

La Tabla 2 indica el valor del Error Cuadrático Medio (MSE) para los modelos encontrados.

$$MSE = \frac{1}{N} \sum_{k=1}^N (y(k) - y_{fuzzy}(k))^2 \quad (8)$$

donde N es la cantidad total de datos, $y(k)$ es la salida de la planta real en el instante k , e $y_{fuzzy}(k)$ es la predicción realizada por el modelo difuso en el instante k .

Tabla 2: Valores de los Índices de Error Utilizando Análisis de Sensibilidades.

Modelo	Variables de entrada	MSE
1	$y(k-1), y(k-2), y(k-3), y(k-4), u(k-1), u(k-2), u(k-3), u(k-4)$	0.1517
2	$y(k-1), y(k-2), u(k-1), u(k-2)$	0.0902

La optimización de la estructura del modelo difuso consiste principalmente en determinar el número óptimo de reglas del modelo difuso. En este caso se definió un número máximo de 11 clusters y se entrenó el modelo para cada una de las posibles valores de clusters.

La Fig. 4 muestra la Raíz del Error Cuadrático Medio (RMSE) para los conjuntos de entrenamiento y prueba. Si no importa la complejidad, el mejor modelo es aquel que tiene menor RMSE. Sin embargo, es posible que un modelo con peor índice de desempeño, pero menos complejo que el modelo óptimo, pueda obtener resultados aceptables bajo un estándar de rendimiento definido preliminarmente. Por lo antes expuesto para este problema se escoge como número de clusters 5, por lo que el modelo difuso contará con 5 reglas.

$$MSE = \sqrt{\frac{1}{N} \sum_{k=1}^N (y(k) - y_{fuzzy}(k))^2} \quad (9)$$

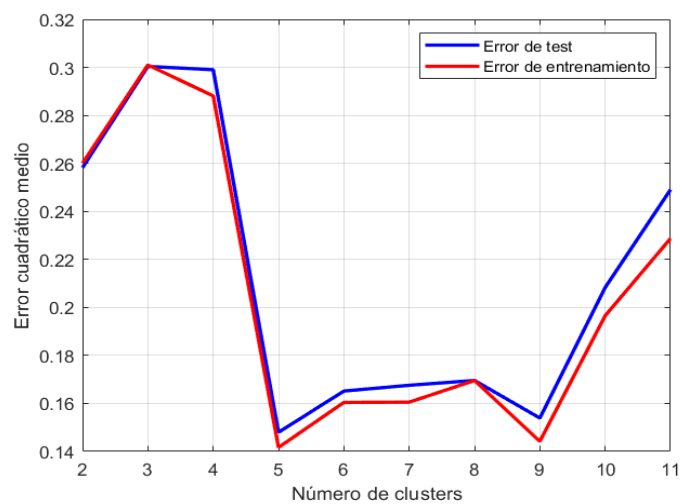


Figura 4: Índice de Sensibilidades.

Tabla 3: Métricas para el Modelo Difuso

Modelos	Estimado	Predicción a 1 pasos	Predicción a 8 pasos	Predicción a 16 pasos
RMSE	0.0044	0.01104	0.0959	0.0105
MAPE	37.61	402.84	428.11	393.38
MAE	0.1038	1.1249	1.2392	1.2769

Figura 5: Clusters.

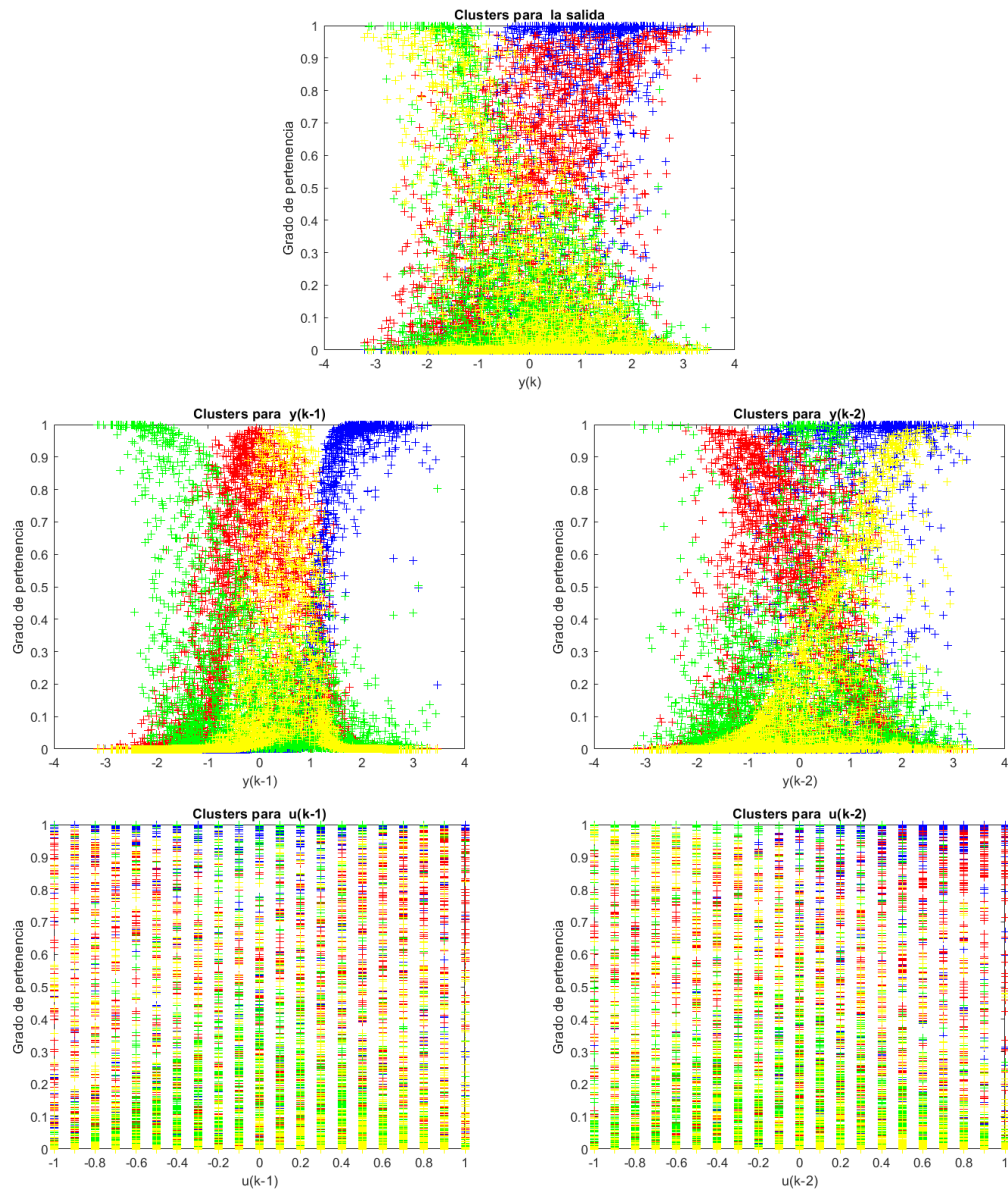


Figura 6: Salida del modelo.

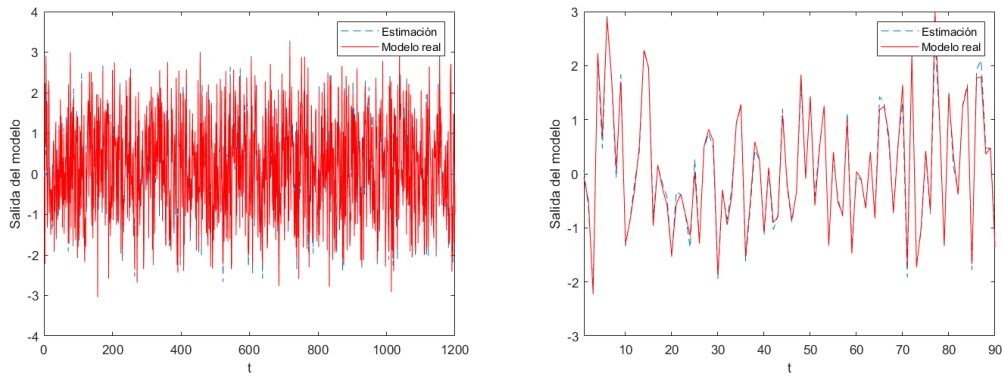


Figura 7: Salida del modelo a 1 paso.

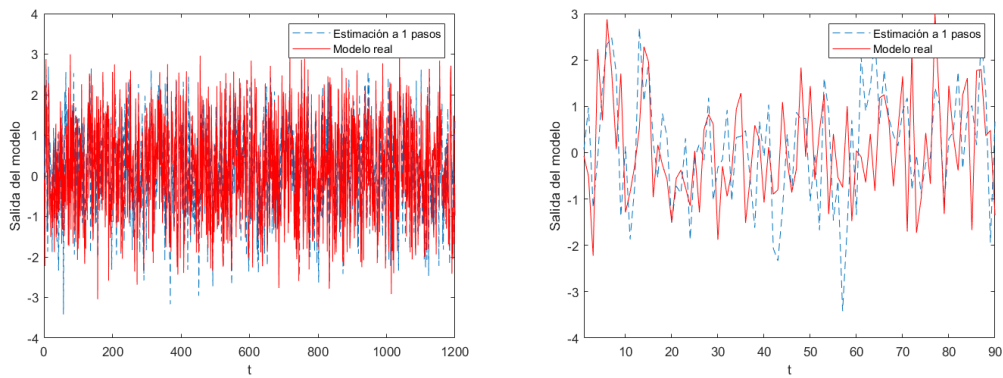


Figura 8: Salida del modelo a 8 paso.

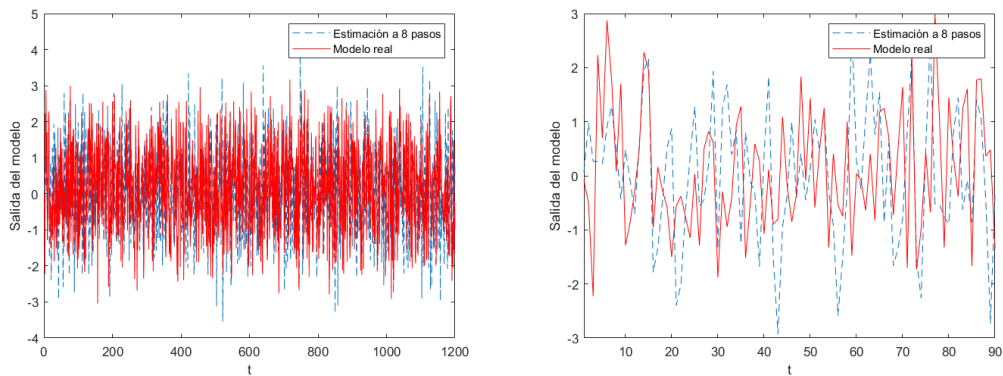
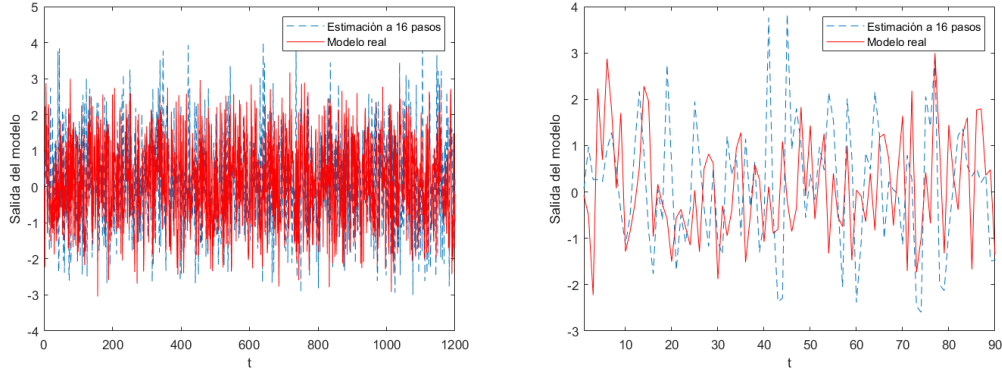


Figura 9: Salida del modelo a 16 paso.



2.2.3. Modelo de red neuronal

Para encontrar un buen modelo de red neuronal se propone seguir los pasos de identificación.

- **Obtención de datos:** Para ello se utiliza el set de datos creado en el punto a) del problema.
- **Selección de datos:** Análogo a los modelos anteriores, se utilizan los datos repartido con un 55 % en el conjunto de entrenamiento, 25 % en el conjunto de prueba y 20 % en el conjunto de validación.
- **Definición de la estructura de la red:** Se propone una red con una capa oculta, función de activación \tanh en la salida de la capa oculta y algoritmo de aprendizaje Levenberg-Marquardt. Las variables de entradas son iguales al número de regresores (4 entradas) como muestra la Figura 10. Por otro lado, el modelo matemático de la red, queda expresado como se muestra en la ecuación 10.

$$\hat{y}(k) = \sum_{i=1}^{N_h} rw_i \left(\tanh \left(\sum_{j=1}^{N_I} lw_{ji} x_j + b_i \right) \right) + c \quad (10)$$

donde, N_h es el número de neuronas en la capa oculta, N_I es el número de variables en la entrada, rw_i es el peso que conecta la i -ésima neurona de la capa oculta con el nodo de salida y lw_{ji} corresponde al peso que una la entrada j con la i -ésima neurona en la capa oculta. Los sesgos de cada neuronas en la capa oculta y para el nodo de salida son b_i y c respectivamente.

- **Selección de entradas relevantes:** De los 4 regresores presente en el sistema se debe analizar cuál tiene mayor peso en el modelo. Un método para encontrar

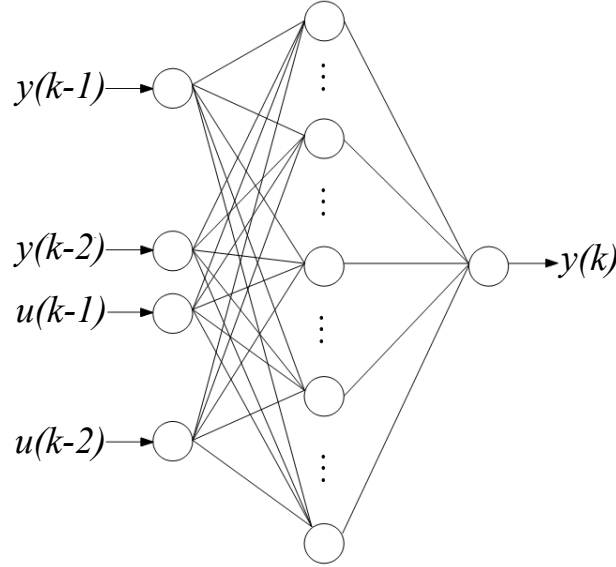


Figura 10: Estructura de la red neuronal perceptrón

dichos regresores es mediante un análisis de sensibilidad evaluando la derivada de la salida de la red por cada premisa de nuestros datos, es decir,

$$\xi_j = \frac{\partial \hat{y}(k)}{\partial x_j} \quad (11)$$

Como la funcion de activacion es \tanh y en la salida es lineal, se tiene,

$$\begin{aligned} \xi_j &= \frac{\partial \hat{y}(k)}{\partial x_j} \\ &= \sum_{i=1}^{Nh} r w_i \left(1 - \tanh \left(\sum_{k=1}^{N_I} l w_{ki} x_k + b_i \right)^2 \right) l w_{ji} \end{aligned} \quad (12)$$

Como se tendrá un valor de ξ_j para cada dato vector de entrada, se genera un vector $\boldsymbol{\xi}_j$ del mismo largo que el número de datos de cada variable.

Luego, se hace uso de un *indicador* I_j para cada entrada j definido como,

$$I_j = \mu^2(\boldsymbol{\xi}_j) + \sigma^2(\boldsymbol{\xi}_j) \quad (13)$$

donde μ es la media del vector de datos y σ^2 es la varianza para cada entrada j .

- **Optimizan paramétrica y estructural:** Para encontrar los valores óptimos de los parámetros peso y sesgo de la red neuronal se utiliza el algoritmo de Levenberg-Marquardt backpropagation [1]. Por otro lado, para encontrar el óptimo de la estructura se analiza cuantas neuronas debe tener la capa oculta. Para ello se estudia el RMSE (Raíz del error cuadrático medio) de la salida para un número de neuronas entre $[2 - 21]$. Los resultados de sensibilidad para cada neurona en la capa oculta se muestran en la Figura 11 y el RMSE evaluado en los 3 conjuntos se muestra en la Figura 12. Se puede ver que el mínimo RMSE para el conjunto de prueba es para 5 neuronas y que el modelo es sensible a la entrada $y(k - 1)$ mayoritariamente.

Nota: El entrenamiento se configura a una velocidad inicial de aprendizaje de la red de 0,05 y un valor de épocas de 2000 sin evaluación de *Overfitting* (esa es la razón de que exista una diferencia tan grande entre el número de neuronas óptimo definido por el conjunto de entrenamiento (20 neuronas) y por el conjunto de prueba (5 neuronas)).

Cabe mencionar que no siempre el conjunto de prueba entrega el valor óptimo de neuronas exactamente igual a 5 ya que depende levemente del valor inicial de los pesos y sesgos, los resultados mostraron que entre 5 y 7 neuronas está el mínimo. Finalmente, se selecciona un número de neuronas en la capa oculta igual a 5 con solo una entrada, $y(k - 1)$.

Ejercicio N°1

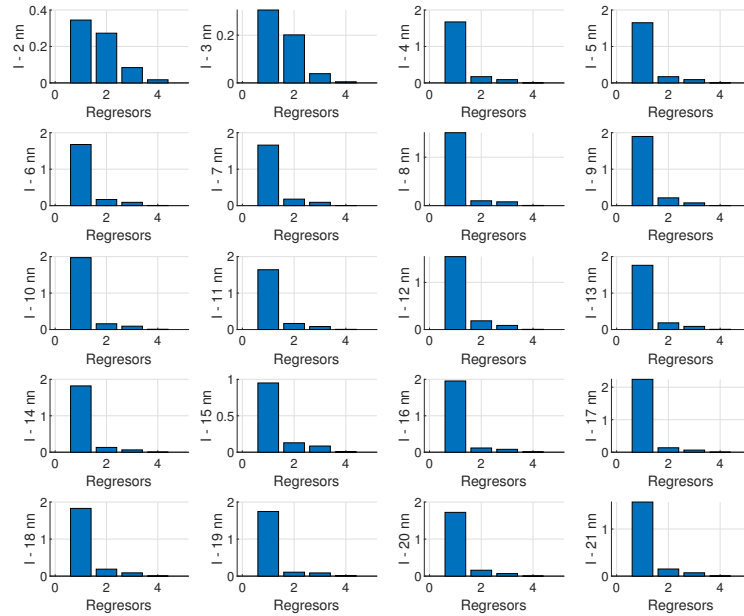


Figura 11: Sensibilidad para diferente número de neuronas en la capa oculta. *Sin Overfitting.*

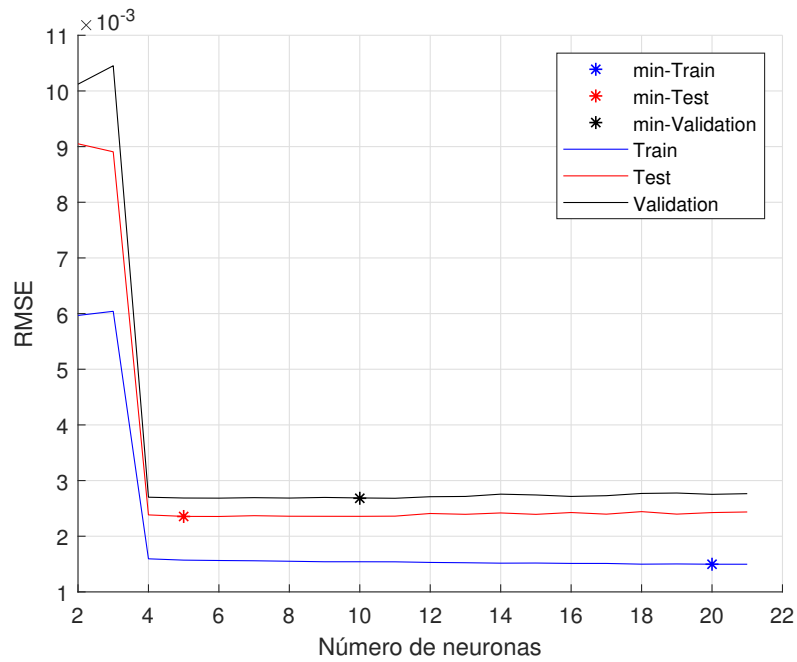


Figura 12: RMSE para diferente número de neuronas. *Sin Overfitting.*

Ejercicio N°1

- sadsadas

3. Resultados

4. Conclusión

Referencias

- [1] Matlab, “trainlm, levenberg-marquardt backpropagation.” [Online]. Available: <https://www.mathworks.com/help/deeplearning/ref/trainlm.html>