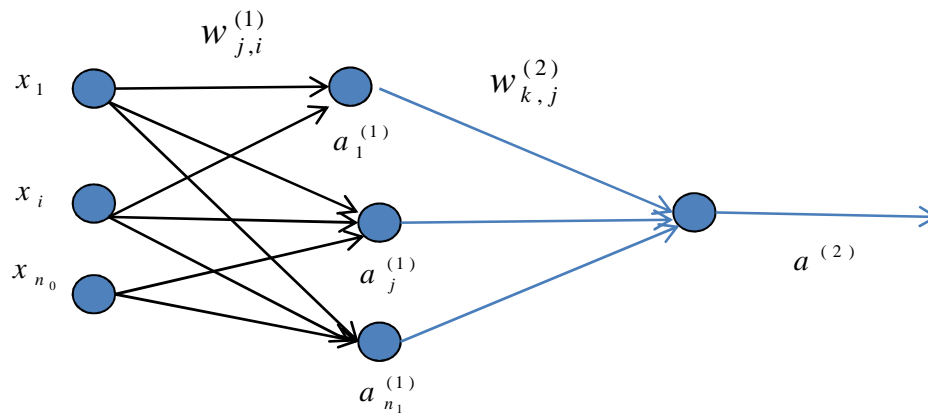


TAREA #1:

Regresión Neuronal

$$a^{(0)} = x$$



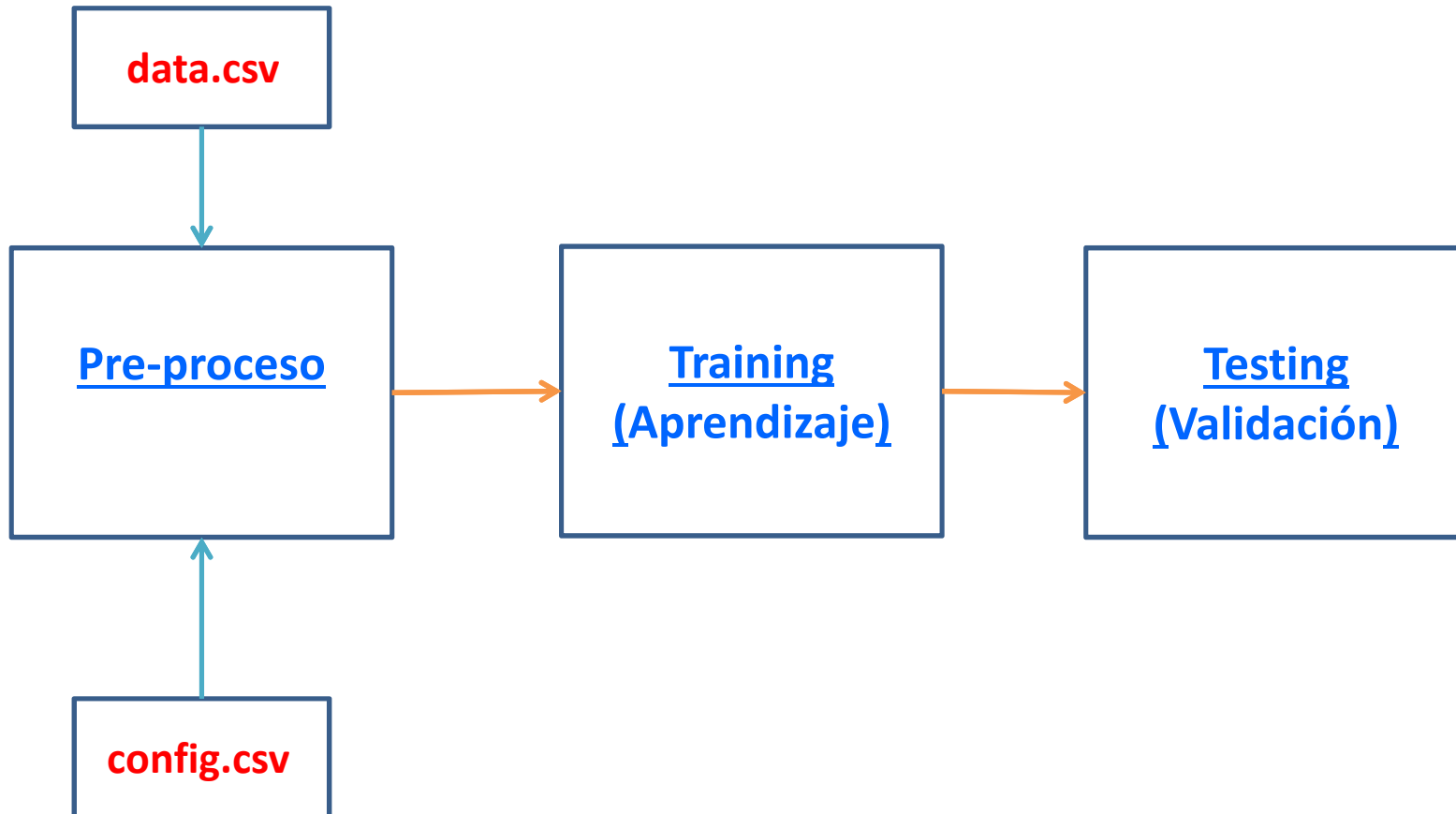
$$z^{(l)} = w^{(l)} \times a^{(l-1)}$$

$$a^{(l)} = f(z^{(l)}) = \frac{1}{1 + \exp(-z^{(l)})}$$

OBJETIVO

Implementar y Evaluar un Modelo de Regresión No-Lineal usando una Red Neuronal Artificial con aprendizaje supervisado vía algoritmo Back-propagation.

Etapas del Modelo:



Configuración del SNN:

- Crea archivo: **config.csv**

- Nodos Ocultos :
- Número Máx. Iteraciones :
- Tasa Aprendizaje :

Pre-Proceso:

- Datos Originales: **train.csv**:
 - **N**-filas por **D**-columnas
 - Las primeras (D-1)-columnas representan las variables explicativas del modelo
 - La última columna representa el valor deseado del modelo
- Re-ordenar aleatoriamente la data **train.csv**.
- Crear nuevos datos desde la data **train.csv**
 - **Xe**: N-filas por (D-1)-columnas.
 - **Ye**: N-filas por 1-coumna.

Pre-Proceso:

- 1.- Normalizar cada columna de **Xe**:

$$x = \frac{(x - x_{\min})}{(x_{\max} - x_{\min})} \times (b - a) + a, \quad a = 0.01 \quad b = 0.99$$

- Normalizar la data **Ye** usando ecuación (1).

Pre-Proceso:

- Datos Originales: **test.csv**:
 - **N**-filas por **D**-columnas
 - Las primeras (**D**-1)-columnas representan las variables explicativas del modelo
 - La última columna representa el valor deseado del modelo
- Re-ordenar aleatoriamente la data **test.csv**.
- Crear nuevos datos desde la data **test.csv**
 - **X_v**: **N**-filas por (**D**-1)-columnas.
 - **Y_v**: **N**-filas por 1-coumna.

Pre-Proceso:

- 2.- Normalizar cada columna de **Xv**:

$$x = \frac{(x - x_{\min})}{(x_{\max} - x_{\min})} \times (b - a) + a, \quad a = 0.01 \quad b = 0.99$$

- Normalizar la data **Yv** usando ecuación (2).

train.py

- Cargar datos de configuración de la SNN.
- Cargar datos de train.
- Realizar proceso de aprendizaje supervisado.
 - Crear archivo de costo :
 - **costo.csv:**
 - Columna 1: MSE.
 - Crear archivo de pesos:
 - **pesos.npz:**
 - w1 w2.

test.py

- Cargar data de test.
- Cargar pesos entrenados.
- Realizar proceso de forward.
 - Crear archivo de métricas:
 - **metrica.csv:**
 - MAE, RMSE , R2.
 - Crear archivo de estimación:
 - **estima.csv**
 - Columna 1: valor real
 - Columna 2: valor estimado.

Métricas de Rendimiento

1. Error del modelo SNN:

$$e_i = y_i - \hat{y}_i, \quad i = 1, \dots, N$$

2. Error absoluto Medio:

$$MAE = \frac{1}{N} \sum_{i=1}^N |e_i|$$

3. Raíz Error cuadrático Medio:

$$RMSE = \sqrt{MSE}$$
$$MSE = \frac{1}{N} \sum_{i=1}^N e_i^2$$

4. Coeficiente de determinación:

$$R^2 = 1 - \frac{\text{var}(e)}{\text{var}(y)}$$

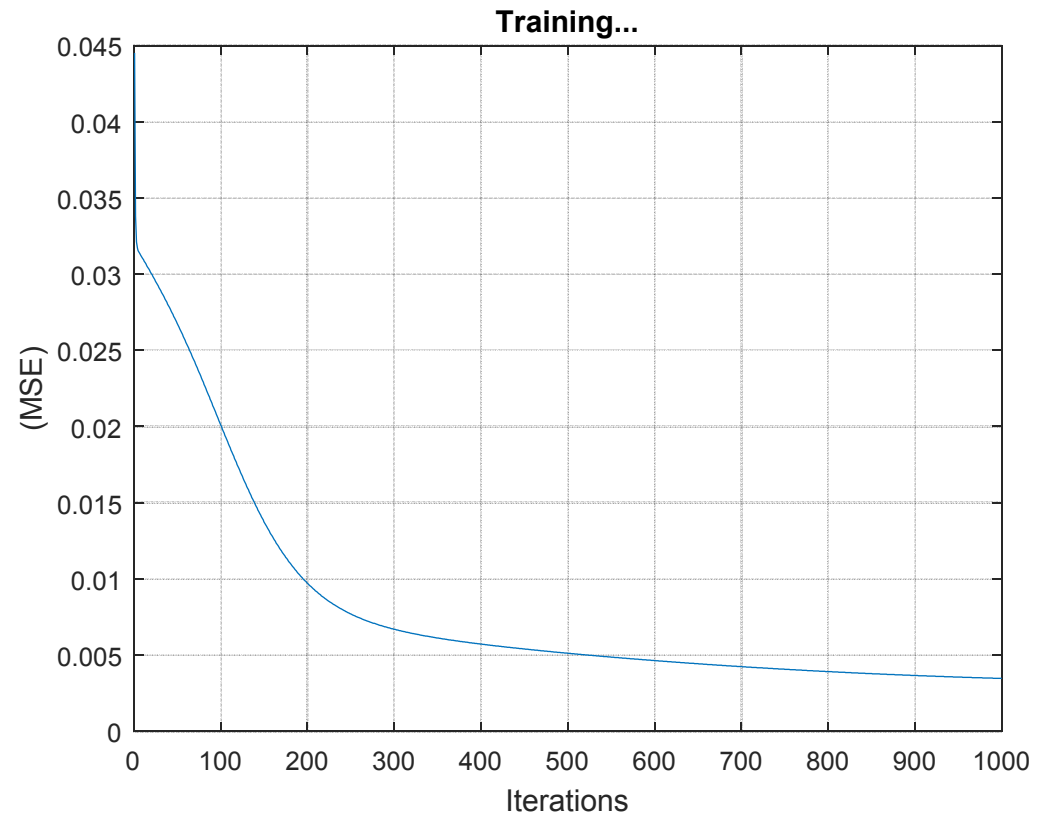
Lenguaje Programación : Python

Numpy/Panda

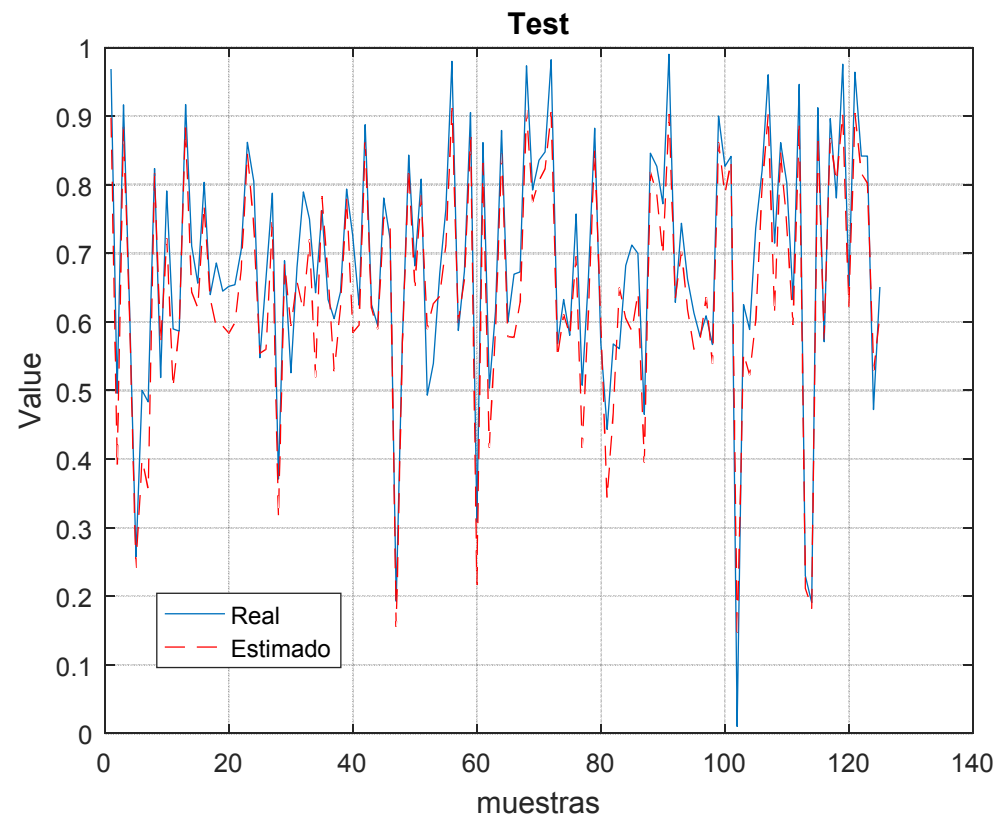
RESULTADOS :

Train/Test

Training:



Testing:



MAE: 0.0486

RMSE: 0.0600

R2 (%): 92.79

FECHA ENTREGA:

ENTREGA

- Lunes 06/Septiembre/2021, Hora: 9:00
 - Lugar: Aula Virtual del curso
- **Programas fuentes:**
 - config.csv,
 - train.py, test.py
- **Archivos Resultados:**
 - costos.csv
 - metricas.csv
 - estima.csv

OBSERVACIÓN:

- Si un Grupo no Cumple con los requerimientos funcionales y no-funcionales, entonces la nota máxima será igual a 4.0 .

CONTINUARÁ....