



# **TAREA 2ª: DEEP LEARNING**

**Prof. NIBALDO RODRÍGUEZ A.**



## OBJETIVO

- Implementar y evaluar el rendimiento de un modelo Aprendizaje Profundo (DL) usando un algoritmo Híbrido (Pseudo—inverse y Descenso Gradiente) para clasificar diez tipos de severidad de fallos de un motor de inducción.



## DATA : Train

- Formato: **train\_x.csv** :
  - ☐ **D-filas**: número de atributos
  - ☐ **N-columns**: números de muestras
- Formato: **train\_y.csv** :
  - ☐ **10-filas**: etiqueta binaria para cada clase
  - ☐ **N-columns**: números de muestras



## DATA: Test

- Formato: test\_x.csv :

- ☐ **D-filas:** número de atributos
- ☐ **N-columns:** números de muestras

- Formato: test\_y.csv :

- ☐ **10-filas:** etiqueta binaria para cada clases
- ☐ **N-columns:** números de muestras



## FASE 1: Pre-Tuning

- `train.py`
  - Archivos de Salida:
    - `costo_softmax.csv`. (N-filas por 1-columna)
    - Pesos del Deep Learning (`w_dl.npz`).
  
- `test.py`
  - Archivos de Salida:
    - `metrica_dl.csv`.
      - F-scores para cada una de las 10 clases.



## Configuración: AE-Apilados

### ■ param\_sae.csv

- ☐ Porcentaje de training : 0.80
- ☐ Tasa aprendizaje : 0.1
- ☐ Penalidad de P-inversa : 100
- ☐ Máximo Iteraciones : 100
- ☐ Número Nodos Oculto AE1 : 400
- ☐ Número Nodos Oculto AE2 : 200
- ☐ Número Nodos Oculto AE3 : 100
- ☐ ...
- ☐ .....



## Configuración : Softmax

### ■ param\_softmax.csv

- ☐ Máximo Iteraciones : 500
- ☐ Tasa aprendizaje ( $\mu$ ) : 0.1
- ☐ Penalidad ( $\lambda$ ) : 0.0001



## **ENTREGA**

- **Lunes 10/Mayo/2021**

- ☐ Hora : 9:00am

- ☐ Lugar : Aula Virtual del curso

- **Lenguaje Programación:**

- ☐ Python version: 3.7.6 window (anaconda)

- numpy

- panda





## **OBSERVACIÓN:**

- Si un Grupo no Cumple con los requerimientos funcionales y no-funcionales, entonces la nota máxima que obtendrá será igual a 3,5 (tres como a cinco).