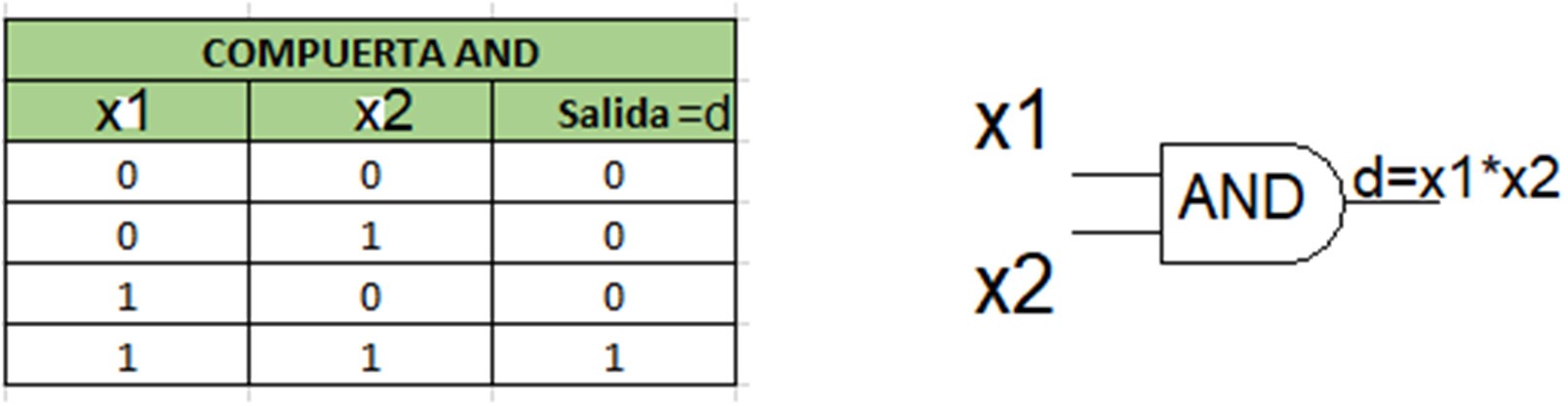
# Ejercicios 1 Machine Learning

**Prof. Gustavo Ceballos B. Fecha entrega: 08/08/2020 hasta las 20:00 hrs**

E1) En la figura, se muestra un diagrama circuital de una compuerta AND (**Y** lógico) con su respectiva tabla de verdad (o patrones de datos).

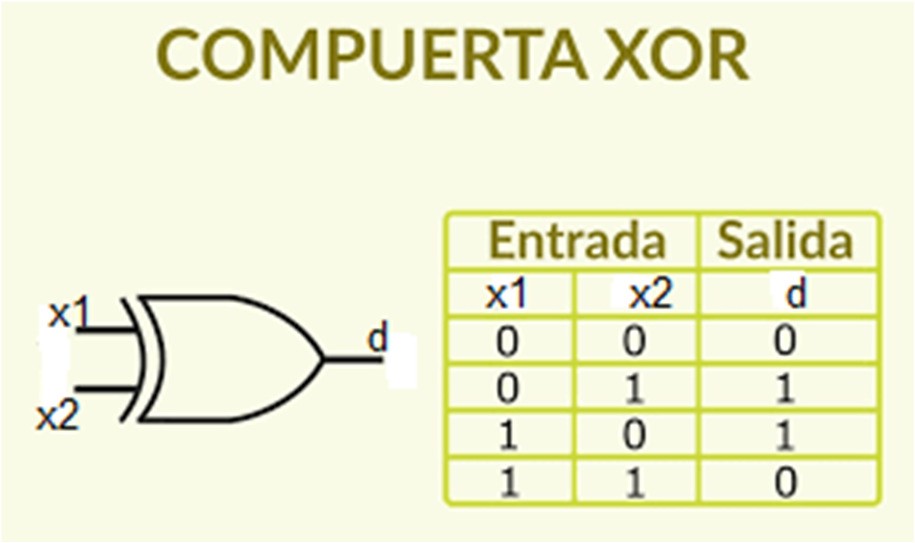


1. Usando el algoritmo de “corrección de error” visto en clases, se pide que implemente un programa en Matlab (**No puede usar ninguna función del Neural Network Toolbox de Matlab**), que sea capaz de entrenar una red neuronal (determinar los pesos sinápticos wi) Perceptrón Simple (1 Neurona) de tal manera que esta sea capaz de aprender el comportamiento de dicha compuerta AND.
2. Muestre los datos x2 vs x1 en un plano para luego determinar la recta de separación de los conjuntos cuya salida es 0 y 1 y grafíquela el dicho plano x2 vs x1.

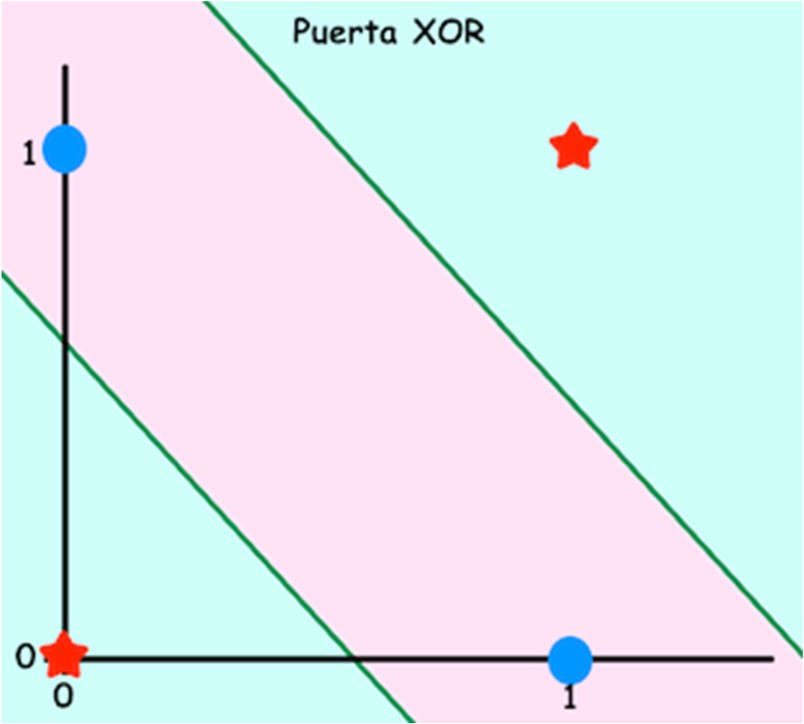
**Indicación:** (debe usar los comandos **gscatter**(x2,x1,…) o bien **plot**(x2,x1,….) y luego los comandos **hold on** para graficar la recta encima del gráfico anterior y luego termina con **hold off** para soltar ese gráfico o figure.

1. Realice un gráfico del número de pasadas (o épocas) por la data completa hasta lograr el aprendizaje de la red neuronal **vs** el coeficiente o razón de aprendizaje **γ**. Comente al respecto.

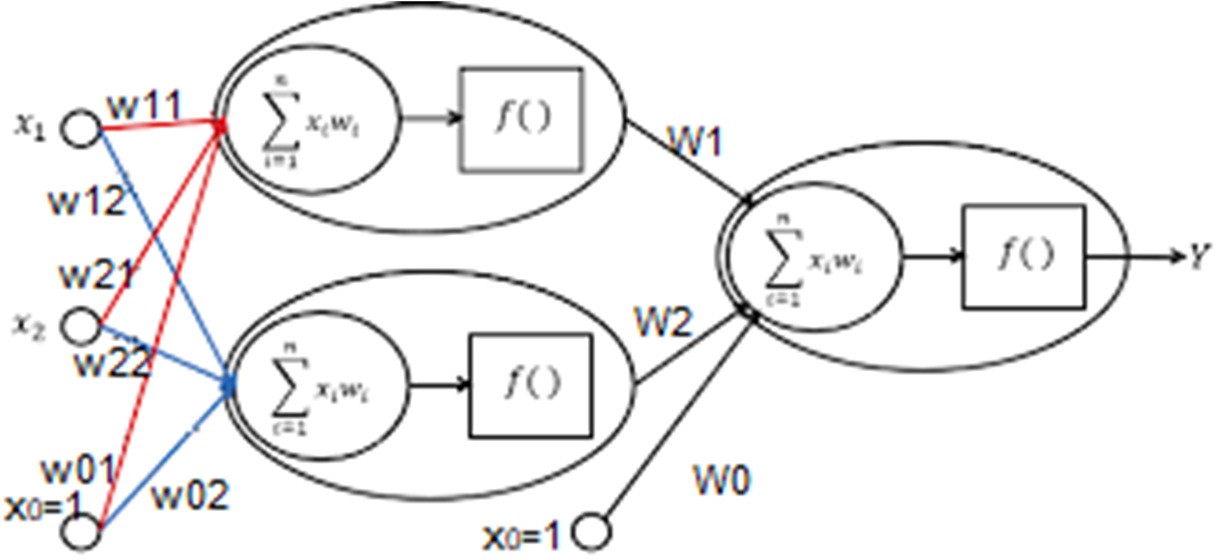
P2) En la figura, se muestra un diagrama circuital de una compuerta XOR (**O exclusivo** lógico) con su respectiva tabla de verdad (o patrones de datos).



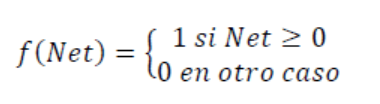
Si uno grafica los patrones en el plano x1-x2, se da cuenta que estos conjuntos NO son linealmente separables (es decir, no se puede construir una línea recta que separe los conjuntos) como se muestra en la figura siguiente.



Los vectores (x1,x2) en rojo corresponden a la salida d=0 y los azules a d=1.

Es por este motivo que 1 solo perceptrón no sirve para abordar este problema. Se propone entonces, usar 1 capa oculta con 2 perceptrones, más el perceptrón de la capa de salida. Es decir:

En que la función de activación f( ), es la función escalón unitario, es decir:



a) Usando el algoritmo de “corrección de error” visto en clases, se pide que implemente un programa en Matlab (**No puede usar ninguna función del Neural Network Toolbox de Matlab**), que sea capaz de entrenar esta red neuronal (determinar los pesos sinápticos wij y Wj). Es decir, necesita determinar 9 pesos sinápticos (incluido los 3 bias; w01 w02 y W0).

**Nota:** La Prueba debe incorporar su código de Matlab en Anexos. Además, junto con el envío de la Prueba Solemne 1, debe adjuntar el archivo del programa construido por usted en Matlab (programa**.m** por ejemplo).