Universidad de Ingeniería y Tecnología - UTEC
Departamento de Ingeniería Mecatrónica

Curso: Tópicos de Ingeniería Mecatrónica (2020-2)
Prof. Oscar E. Ramos Ponce

Tarea 1 (parte 1)

Entrega: domingo 11 de octubre del 2020

La respuesta a cada pregunta debe tener un desarrollo que muestre el procedimiento seguido, los resultados obtenidos, y una descripción (explicación/justificación) de dichos resultados. Se debe adjuntar, además, el código utilizado: puede ser como anexo, al final del documento, o como archivos adjuntos al momento de realizar la entrega por canvas. Para todas las siguientes preguntas se debe implementar el proceso de entrenamiento y verificación de manera explícita (no se debe usar software adicional)

1. (3 pts) Considerar los datos que se brindan en el archivo `datos_regresion_train.csv`, donde la primera columna corresponde a los atributos, y la segunda columna a los valores deseados. Desarrollar un modelo de regresión que se ajuste a los datos. Mostrar la función de costo en función del número de iteraciones. Evaluar el desempeño del modelo usando los datos de `datos_regresion_test.csv`, a través del MSE. Dado que es un modelo de una sola variable, graficar, además, los datos reales junto con los datos que se predice.

   Nota: dada la naturaleza de los datos, puede ser útil utilizar alguna(s) base(s) no lineal(es) para un mejor ajuste.

2. (3 pts) Descargar los archivos `housing.data` y `housing.names` del repositorio *UCI Machine Learning Repository* (aquí) El primer archivo contiene varios atributos relacionados con el precio de casas en Boston; y el segundo archivo contiene la descripción de lo que representa cada columna. Los datos deben ser divididos de manera aleatoria en un 80 % para entrenamiento y un 20 % para prueba, aproximadamente. Usar regresión lineal (multivariable) para encontrar el modelo de predicción, visualizando el comportamiento de la función de costo. Utilizar el conjunto de prueba generado, para validar el funcionamiento del modelo, usando alguna métrica.

   Nota 1: en esta pregunta no es necesario usar alguna base no lineal, aunque de manera opcional podría realizarse.

   Nota 2: Existen varias maneras de dividir los datos aleatoriamente. Una forma simple de hacerlo, cuando no hay muchos datos, consiste en utilizar la función `random.shuffle` para primero aleatorizar los datos. Luego de aleatorizados, se puede separar el porcentaje deseado usando índices (*slicing*). En esta y otras preguntas, se puede usar este método o cualquier otro método.

3. (3 pts) El archivo `datos_clasificacion.csv` tiene tres columnas: las dos primeras representan los dos atributos x_1, x_2, y la última columna representa la clase y a la cual pertenece

la instancia, donde $y \in \{0, 1\}$. Dividir los datos en conjunto de entrenamiento y conjunto de prueba (usando el criterio 80 %-20 %). Entrenar un clasificador basado en regresión logística para clasificar los datos. Dado que la curva de decisión es no lineal, se usará bases polinomiales que incluirán combinaciones de ambos atributos hasta el sexto grado; es decir, se generará atributos tales que:

$$\Phi(x) = \begin{bmatrix} x_1 & x_2 & x_1^2 & x_1x_2 & x_2^2 & x_1^3 & \cdots & x_1^2x_2^4 & x_1x_2^5 & x_2^6 \end{bmatrix}^T.$$

Estos atributos se obtienen usando la función `generacion_bases`. Además, el clasificador debe tener un término de regularización L_2. Graficar los puntos y la frontera de decisión (se puede usar las funciones `plot_data` y `plot_frontera`). Igualmente, graficar la función de costo para verificar que converge.

4. (4 pts) Dados un conjunto de atributos relacionados con el género, la edad, condiciones como asma, hipertensión, etc. se desea predecir si una persona a quien se le detecta COVID-19 ingresará a cuidados intensivos (UCI). Se utilizará los datos llamados `covid_train.csv` y `covid_test.csv` (para entrenamiento y prueba, respectivamente), que son *datasets* reducidos del *dataset* completo que se encuentra aquí. Utilizando regresión logística, entrenar un sistema que detecte, en la medida de lo posible, el ingreso a UCI dados los atributos de entrada. Una vez que se tenga el sistema entrenado, calcular algunas métricas para evaluar el desempeño del sistema, e indicar si existe *overfitting* (sobreajuste) o *underfitting* (subajuste).

   Nota 1: Los archivos `Catalogs.xlsx` y `Description.xlsx` brindan información adicional sobre el significado de los atributos y sus valores.

   Nota 2: Se puede realizar mejoras al algoritmo básico de regresión logística, usando bases no lineales o regularización, por ejemplo.