**Práctica 5 (Train & Cross Validation)**

¿Cómo he planteado la práctica?

Primero necesitamos los datos, en este caso leemos el .MAT ‘ex5data1’, y separamos los datos en X e Y, Xval e Yval, Xtest e Ytest.

Posteriormente hay que implementar la regresión lineal multivariable regularizada, que devolverá el coste y la gradiente.

Finalmente hay que dibujar la recta de resultados y la comparativa de los errores de aprendizaje.

Código comentado y gráficas

1. Importar las librerías

Texto

Descripción generada automáticamente

1. Obtener los datos del CSV:

Texto

Descripción generada automáticamente

1. Ahora implementamos la regresión lineal multivariable, por un parte el coste, por otra parte la gradiente, y finalmente regularizamos ambas

Texto

Descripción generada automáticamente

1. Y comprobamos los resultados

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

1. Posteriormente minimizamos con la función MINIMIZE de Scipy, con

, el método escogido en este, y los siguientes puntos es ‘TNC’ Truncated Newton

Texto

Descripción generada automáticamente

1. Finalmente mostramos la recta de resultados, como en anteriores prácticas, nos quedamos con el mínimo y máximo, y predecimos la Y con las

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

Podemos apreciar un alto sesgo (high bias), ya que los datos no se ajustan a la gráfica

1. Ahora calculamos la tasa de error de aprendizaje, acumulando todos los ejemplos de X (), el error se calcula con la regresión lineal multivariable y con . Esta función devuelve un array concatenado de todos los errores.

Texto

Descripción generada automáticamente

1. Finalmente, dibujamos el error de aprendizaje apoyándonos en la función anterior

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Como habíamos comentado anteriormente, tenemos un alto sesgo, ya que el error de la Cross Validation es relativamente alto.

**Práctica 5 (Regresión polinomial)**

¿Cómo he planteado la práctica?

Primero necesitamos los datos, en este caso leemos el .MAT ‘ex5data1’, y separamos los datos en X e Y, Xval e Yval, Xtest e Ytest.

Posteriormente hay que implementar una función que convierte los datos en vectores polinómicos , y las funciones de normalización, una para normalizar un valor, y otra para normalizar a partir de unos valores y .

En el siguiente paso, calculamos las predicciones y comprobamos si se ajusta a los datos.

Finalmente probamos varios parámetros y dibujamos el error que tiene respecto a su incremento, para así sacar un .

Código comentado y gráficas

1. Importar las librerías

Texto

Descripción generada automáticamente

1. Obtener los datos del CSV:

Texto

Descripción generada automáticamente

1. Creamos la función polinómica que nos devuelve un vector con :

Texto, Carta

Descripción generada automáticamente

1. Implementamos una función de normalización a partir de un vector, con la siguiente fórmula:

Además, añade una columna de 1s.

Texto

Descripción generada automáticamente

Por otro lado, implementamos una función de normalización a partir de unos valores y

Texto

Descripción generada automáticamente

1. Ahora usamos la función de MINIMIZE para obtener las , para ello primero convertimos la X en polinómica y después la normalizamos (Recordar que también se añade la columna de 1s).

Texto

Descripción generada automáticamente

1. Y dibujamos la recta de resultados para comprobar si la polinomización se ajusta a los datos:

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Y podemos comprobar que sí, se ajusta a los datos, quizás demasiado, por ello vamos a dibujar la curva de aprendizaje.

1. Dibujamos la curva de aprendizaje, para ello necesitamos polinomizar y normalizar ambas X, la y la .

Gráfico

Descripción generada automáticamente con confianza media

La curva de aprendizaje en el Train set es bajo y estable, lo cual está bien, por otro lado, la Cross Validation es baja pero irregular, por lo que necesitaríamos más datos para conseguir que bajase, o más iteraciones.

**NOTA: Recordar que en la curva de aprendizaje siempre debe ser 0**

1. Ahora dibujamos varias y su error, para conocer la , al igual que antes, polinomizamos y normalizamos.

Texto

Descripción generada automáticamente

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Podemos observar que el valor está en el intervalo 2 < < 3

1. Finalmente, el valor más cercano a que hemos probado es 3, por lo que comprobamos el error, y en este caso es de 3.5720, como pide el enunciado.

Texto

Descripción generada automáticamente