PRÁCTICA 1 - REGRESIÓN LINEAL Y DESCENSO SEGÚN EL GRADIENTE

Javier Sáez Aprendizaje Automático

Ejercicio 1

En este ejercicio, se implementará el algoritmo de descenso de gradiente y se aplicará este sobre varias funciones con el objetivo de estudiar cómo afecta tanto el punto inicial como la tasa de aprendizaje η a la solución que este algoritmo encuentra.

Lo primero que debemos hacer es recordar en qué consiste el algoritmo de descenso de gradiente, veámoslo en pseudocódigo:

Algorithm 1: Descenso de gradiente

Result: Mínimo local de una función
parametros: η, f, $max\_iteraciones$,$criterio\_parada$, $punto\_inicial$;
$punto = punto\_inicial$;
while *No (Criterio de parada)* **do**
  $gradiente \leftarrow \nabla f(punto)$;
  $direccion \leftarrow -gradiente$;
  $punto \leftarrow punto - \eta \cdot direccion$
end

En la implementación en el lenguaje de programación escogido, *python*, se podrá hacer todo el contenido de este bucle while en una única línea. Sin embargo, se añadirá contenido extra al cuerpo de la función para que nos devuelva no solo el mínimo, sino también toda la sucesión de puntos que se han ido obteniendo así como el número de iteraciones que se han dado. En este caso, consideraremos una iteración claramente como una actualización del punto mínimo de la función.

```
def gradient_descent(eta,fun,grad_fun,maxIter,error2get,initial_point):
    iterations = 0
    w_t = initial_point
    all_w = []
    all_w.append(w_t)

    while iterations < maxIter and fun(w_t) > error2get:
        # All gradient descent in 1 line
        w_t = w_t - eta*grad_fun(w_t)
        all_w.append(w_t)
        # sum iterations
        iterations += 1
```

```
    return np.array(all_w),w_t, iterations
```

Consideremos ahora la función

$$E(u,v) = \left(u^3 e^{(v-2)} - 2v^2 e^{-u}\right)^2.$$

Gradiente de E

Nuestra función E es claramente derivable , así que procedemos a obtener sus derivadas parciales para poder aplicarle el algoritmo. Obtenemos que

$$\frac{\partial E}{\partial u} = 2\left(u^3 e^{(v-2)} - 2v^2 e^{-u}\right)\left(3u^2 e^{(v-2)} + 2v^2 e^{-u}\right)$$

y

$$\frac{\partial E}{\partial v} = 2\left(u^3 e^{(v-2)} - 2v^2 e^{-u}\right)\left(u^3 e^{(v-2)} - 4v e^{-u}\right).$$

Tras definir una función para cada una de estas derivadas parciales y una para darnos el gradiente $\nabla E = \left(\frac{\partial E}{\partial u}, \frac{\partial E}{\partial v}\right)$ de E en un punto , podemos aplicar el algoritmo. Los parámetros de la ejecución son los siguientes:

- Tasa de aprendizaje $\eta = 0.1$.
- Punto inicial $(u,v) = (1,1)$.
- Condición de parada: Error E inferior a 10^{-14} (esto está programado directamente en la función).
- Máximo de iteraciones: prácticamente ilimitado (10000000000) para este primer ejercicio.

Tras la obtención del gráfico, se ha programado también una pequeña función que formatea la salida de algunos datos que pueden ser relevantes. En particular, dada la propia función en un string, los parámetros del gradiente descendente y los resultados obtenidos del mismo, esta función imprimirá estos parámetros, y los resultados obtenidos entre ellos el número de iteraciones y el punto final. Esta función se llama `print_output_e1` y el resultado sobre este problema es:

```
Gradiente descendente sobre la funcin: E(u,v) = (u^3 e^(v-2) - 2v^2
   e^(-u))^2
Punto inicial: [1. 1.]
Tasa de aprendizaje: 0.1
Numero de iteraciones: 10
Coordenadas obtenidas: ( 1.1572888496465497 , 0.9108383657484799 )
```