

Práctica 1

Aprendizaje Automático

Ignacio Aguilera Martos

25 de Marzo de 2019

Índice

| | |
|---------------------------------|----------|
| 1. Ejercicio 1 | 2 |
| 1.1. Apartado 1 | 2 |
| 1.2. Apartado 2 | 2 |
| 1.2.1. Apartado a | 2 |
| 1.2.2. Apartado b y c | 3 |

1. Ejercicio 1

1.1. Apartado 1

Se nos pide implementar el algoritmo de gradiente descendente, veamos como funciona para justificar la implementación.

El algoritmo de gradiente descendente aplicado a minimización de funciones toma la idea del ajuste de una función lineal que hemos estudiado en teoría. Ahora en vez de actualizar w_j de forma proporcional a la derivada parcial j -ésima de el error dentro de la muestra vamos a actualizarla mediante la derivada parcial j -ésima de la función que queremos minimizar.

El algoritmo parte de un punto w inicial, que se irá actualizando hasta aproximar el mínimo de la función. El valor de w en la iteración $i+1$ -ésima será el valor de w en la iteración i -ésima menos una constante de proporcionalidad que llamamos tasa de aprendizaje por el gradiente de la función a minimizar, esto es:

$$w_{i+1} = w_i - \eta \cdot \nabla f(x_1, \dots, x_n)$$

Donde $f(x_1, \dots, x_n)$ es la función que queremos minimizar.

De esta forma lo que vamos a hacer en la práctica es comprobar lo siguiente: vamos a actualizar de esta forma el valor de w hasta que agotemos un número de iteraciones máximas fijadas o hasta que el valor absoluto de la diferencia de $f(w_i)$ y $f(w_{i+1})$ sea menor que una cierta tolerancia, esto es que la imagen de los w_i y w_{i+1} no hayan experimentado un cambio apreciable entre dos iteraciones consecutivas.

Cabe destacar que en mi caso he generalizado la implementación del algoritmo mediante el uso de la librería SymPy. Mediante esta librería he implementado una función que generaliza el cálculo del gradiente en un punto, de tal forma que no tengo que programar explícitamente la función que calcula el gradiente de la función que queremos minimizar. Esto se plasma en las funciones “evaluate” y “gradiente” de mi código.

1.2. Apartado 2

1.2.1. Apartado a

En este apartado se nos proporciona la siguiente función a minimizar:

$$E(u, v) = (u^2 e^v - 2v^2 e^{-u})^2$$

Vamos a calcular analíticamente la expresión del gradiente como se nos pide en el ejercicio. Lo primero que tenemos que hacer es ver cómo sería la expresión del gradiente en nuestro caso:

$$\nabla E(u, v) = \left(\frac{\partial E(u, v)}{\partial u}, \frac{\partial E(u, v)}{\partial v} \right)$$

Calculemos pues cada una de las dos parciales.

$$\frac{\partial E(u, v)}{\partial u} = 2(u^2 e^v - 2v^2 e^{-u})(2ue^v + 2v^2 e^{-u})$$

$$\frac{\partial E(u, v)}{\partial v} = 2(u^2 e^v - 2v^2 e^{-u})(u^2 e^v - 4ve^{-u})$$

Por tanto:

$$\nabla E(u, v) = \left(\frac{\partial E(u, v)}{\partial u}, \frac{\partial E(u, v)}{\partial v} \right) = 2(u^2 e^v - 2v^2 e^{-u}) \cdot [(2ue^v + 2v^2 e^{-u}), (u^2 e^v - 4ve^{-u})](u, v)$$

Cabe decir que esta es una función $\nabla E(u, v) : \mathbb{R}^2 \rightarrow \mathbb{R}^2$, es decir que toma dos variables y que obtiene un vector de dimensión 2.

1.2.2. Apartado b y c

En este apartado se nos pide dar el punto en el que el método de gradiente descendente ha alcanzado por primera vez un valor de la función menor que 10^{-14} y el número de iteraciones que ha consumido hasta obtener dicho valor.

En este caso el punto en el que se ha encontrado un valor que cumple esta restricción es $w = (0,619207678450638, 0,968448269010049)$ y para ello ha consumido 33 iteraciones.