Práctica 1

David Cabezas Berrido

Índice

1. Ejercicio sobre la búsqueda iterativa de óptimos: Gradiente descendiente

1.1. Minimizar la función $E(u, v)$

La función a minimizar es $E(u,v) = (ue^v - 2ve^{-u})^2$, le aplicamos el algoritmo del gradiente descendente partiendo del punto $w = (1,1)$ con tasa de aprendizaje $\eta = 0.1$. La función es no negativa y sabemos que si encontramos un cero será un mínimo absoluto, aceptamos un margen de error $\varepsilon = 10^{-14}$ y nos interesa el punto en el que se alcanza y las iteraciones necesarias para alcanzarlo. También he fijado un máximo de 100 iteraciones, ya que no tengo asegurado encontrar un 0 y hay que añadir esa condición para asegurarnos de que va a parar en algún momento.

He usado la librería `sympy` para el cálculo de las derivadas parciales en el programa. También las he calculado analíticamente y el gradiente queda de este modo:

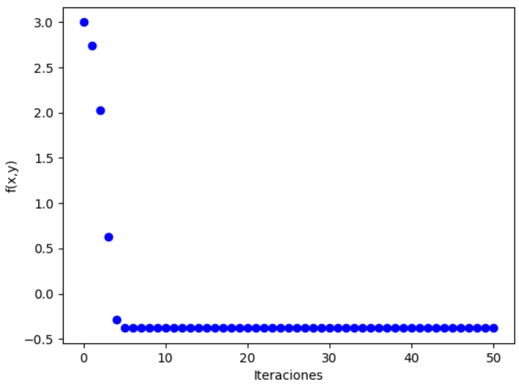
$$\nabla E(u,v) = \begin{pmatrix} \frac{\partial E}{\partial u}(u,v) \\ \frac{\partial E}{\partial v}(u,v) \end{pmatrix} = \begin{pmatrix} 2(ue^v - 2ve^{-u})(e^v + 2ve^{-u}) \\ 2(ue^v - 2ve^{-u})(ue^v - 2e^{-u}) \end{pmatrix}$$

He ejecutado el algoritmo y, en sólo 10 iteraciones, he encontrado un valor por debajo de ε, en el punto $w = (0.0447362903977822, 0.0239587140991418)$.

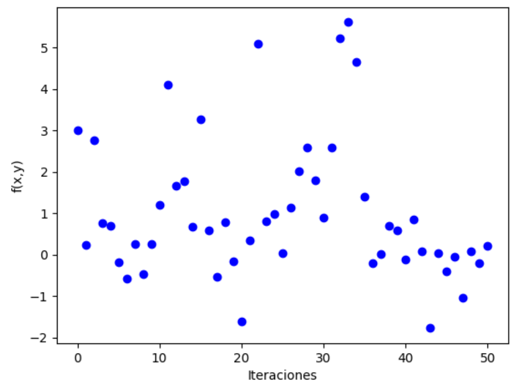
1.2. Estudiar la dependencia del learning rate (η)

He utilizado el algoritmo del gradiente descendente para buscar un mínimo local de la función $f(x,y) = (x-2)^2 + 2(y+2)^2 + 2\sin(2\pi x)\sin(2\pi y)$ partiendo desde el punto $(1,-1)$. Esta vez realiza un número de iteraciones fijo (50 iteraciones), ya que desconozco los valores que puede tomar la función y no puedo incorporar un criterio de parada como el de antes. Sí que podría incluir como criterio la norma del gradiente, ya que ésta es casi 0 cuando estamos muy cerca del mínimo local.

El objetivo ésta vez es estudiar cómo afecta a la eficacia del algoritmo el valor escogido para la tasa de aprendizaje η, para ello he ejecutado el algoritmo con dos valores diferentes de η (0.01 y 0.1) y he medido el valor de la función en cada iteración, para estudiar la velocidad a la que decrece o si se salta el mínimo y vuelve a crecer. En las siguientes gráficas se muestra cómo evoluciona el valor de la función en cada iteración.



(a) $\eta = 0.01$



(b) $\eta = 0.1$

Figura 1: Comparación de la eficacia del algoritmo para dos valores de η.

Lo deseable es que la función decrezca lo más rápido posible. En el primer caso ($\eta = 0.01$) se llega a un mínimo local bastante rápido (se queda muy cerca en la quinta iteración) en el que la función toma el valor -0.38124949743810027, en la segunda gráfica se aprecian valores de la función cercanos a -2, luego está claro que ese mínimo no es absoluto.

En cambio, la tasa de aprendizaje $\eta = 0.1$ es demasiado alta y esto provoca que se salte el punto donde se alcanza el mínimo local y el algoritmo no converja en el segundo caso. Es importante que la tasa de aprendizaje no sea demasiado alta para evitar esto.