Aprendizaje Automático: 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:

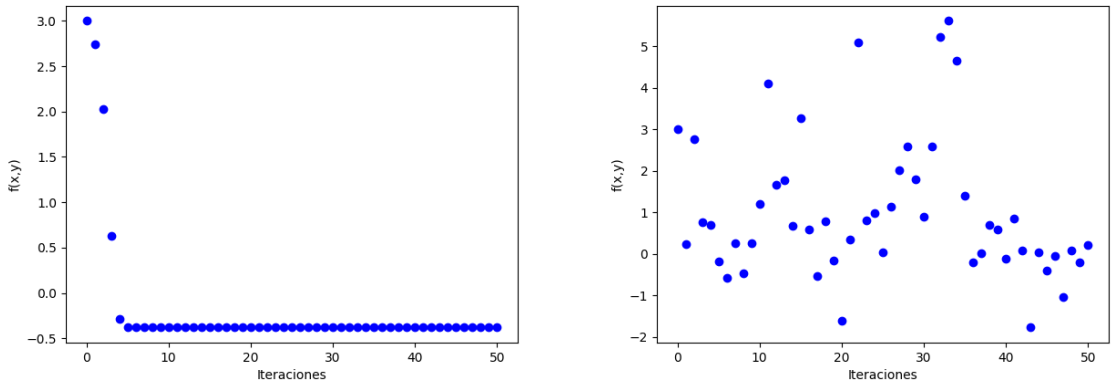
$$\nabla E(u,v) = \begin{pmatrix} \frac{\partial E}{\partial u}(u,v) \\ \frac{\partial E}{\partial v}(u,v) \end{pmatrix} = \begin{pmatrix} 2(ue^v - 2ve^{-u})(e^v + 2ve^{-u}) \\ 2(ue^v - 2ve^{-u})(ue^v - 2e^{-u}) \end{pmatrix}$$

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. Estudio de 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.

1.3. Estudio de la dependencia del punto inicial escogido

He ejecutado el algoritmo del gradiente descendiente para localizar mínimos locales en la función $f(x, y)$ partiendo de distintas condiciones iniciales. Como no se especifica, he realizado 50 iteraciones y he fijado el valor $\eta = 0.01$ como learning rate. He obtenido los siguientes resultados:

	(x, y) donde se alcanza el mínimo	$f(x, y)$
$(2.1, -2.1)$	-1.8200785415471563	$(2.2438049693647883, -2.2379258214861775)$
$(3, -3)$	-0.38124949743810005	$(2.730935648248105, -2.7132791261667033)$
$(1.5, 1.5)$	18.042078009957635	$(1.7779244744891156, 1.032056872669696)$
$(1, -1)$	-0.38124949743810027	$(1.269064351751895, -1.2867208738332965)$

El algoritmo ha encontrado mínimos locales con valores muy diversos, lo que deja claro la dependencia del punto inicial. Partiendo del punto $(1.5, 1.5)$ he obtenido a una solución pésima, ya que ha quedado atrapado en un mínimo local con un valor muy elevado. En cambio, partiendo del $(2.1, -2.1)$ he obtenido un valor bastante cercano al mínimo absoluto, valor que no he obtenido con ninguna de las otras dos condiciones iniciales.

En este problema sé qué soluciones son buenas y cuales son malas porque he podido visualizar la función, pero en el caso bastante común de que el espacio de funciones candidatas tenga dimensión mayor (haya que ajustar más de dos parámetros) habría que minimizar una función de error que no se puede visualizar.

Por tanto, el mínimo absoluto de una función de varias variables es muy difícil de encontrar con certeza, a no ser que la función sea convexa. Así que la dificultad radica en que es difícil saber cuando hemos elegido un punto inicial y un learning rate adecuados.

2. Ejercicio sobre regresión lineal

2.1. Dígitos manuscritos

Vamos a ajustar un modelo de regresión lineal a vectores de características extraídos de imágenes de dígitos manuscritos (sólo de los dígitos 1, con etiqueta -1 y 5, con etiqueta 1), concretamente el valor medio del nivel de gris y simetría del número respecto al eje vertical. Lo haremos tanto por el algoritmo de la pseudoinversa como por el gradiente descendente estocástico.

2.1.1. Por pseudoinversa

Con el algoritmo de la pseudoinversa, los pesos que he obtenido son $w = [-1.11588016, -1.24859546, -0.49753165]$ y el plano solución el de esta figura.

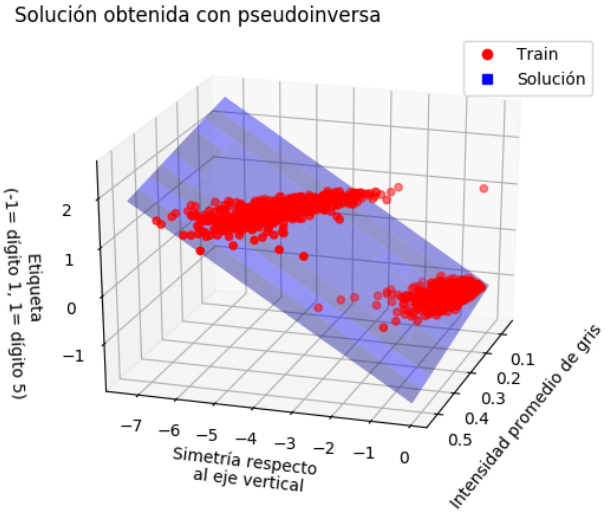


Figura 2: Plano obtenido con la pseudoinversa

Bondad del ajuste: $E_{in}(w) = 0.07918658628900388$, $E_{out}(w) = 0.13095383720052575$

2.1.2. Por gradiente descendente estocástico

Para el algoritmo del gradiente descendente estocástico, he usado el learning rate $\eta = 0.001$, he fijado el tamaño de minibatch a 32 y he realizado 1000 iteraciones. Los pesos que he obtenido son $w = [-1.23746596, -0.20592339, -0.44426966]$ y el plano solución obtenido es el de esta figura.

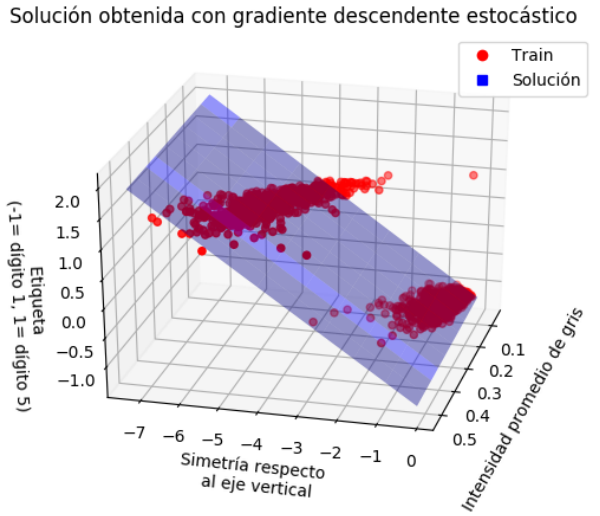


Figura 3: Plano obtenido con el gradiente descendente estocástico

Bondad del ajuste: $E_{in}(w) = 0.08341974836353012$, $E_{out}(w) = 0.13244457530997297$

Ligeramente peores, debido a que es un método aproximado, pero bastante buenos.

2.2. Ejercicio sobre la complejidad del modelo lineal

Generamos una muestra de 1000 puntos uniformemente distribuidos en el intervalo $\mathcal{X} = [-1, 1]\times[-1, 1]$, asignamos etiquetas con la función $f(x_1, x_2) = \text{sign}((x_1 - 0.2)^2 + x_2^2 - 0.6)$ y cambiamos el 10 % de las etiquetas para añadir ruido. La muestra queda de esta forma:

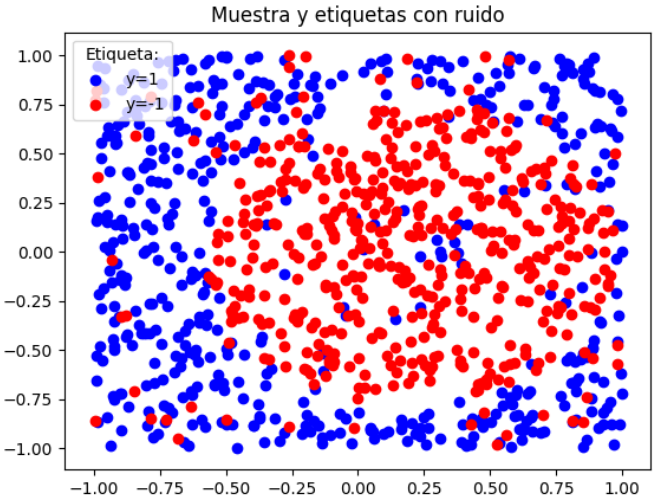


Figura 4: Muestra con ruido

Para los siguientes ajustes, he utilizado el algoritmo del gradiente descendente estocástico con learning rate $\eta = 0.001$, tamaño de minibatch 32 y 1000 iteraciones.

2.2.1. Ajuste de un modelo de regresión con características lineales

He ajustado un modelo de regresión lineal utilizando el vector de características $(1, x_1, x_2)$, los pesos que he obtenido son $w = [0.04968622, -0.51886574, -0.0205383]$ y el plano solución es el siguiente.

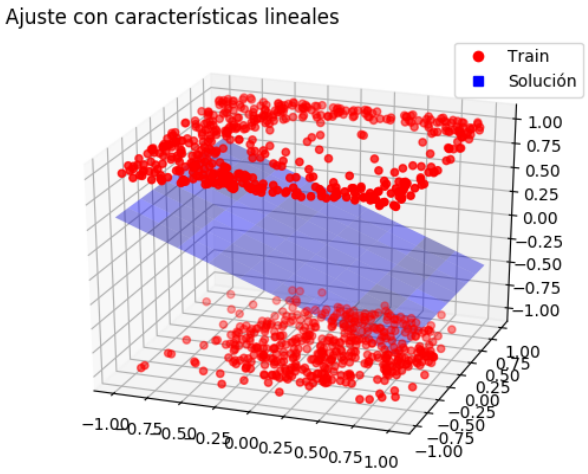


Figura 5: Plano obtenido utilizando características lineales

He repetido el experimento 1000 veces y he obtenido los siguiente errores medios:
E_{in} medio = 0.9240616367833707 , E_{out} medio = 0.929279797769683

E_{in} y E_{out} están muy cercanos (lo que cabe esperar por la simplicidad del modelo). No obstante, son muy altos, cosa que también esperaba tras observar la distribución, difícilmente iba a ser capaz de aproximar esa distribución con un plano.

2.2.2. Ajuste de un modelo de regresión con características no lineales (cuadráticas)

Esta vez, he ajustado un modelo de regresión lineal utilizando el vector de características $(1, x_1, x_2, x_1x_2, x_1^2, x_2^2)$, los pesos que he obtenido son $w = [-0.83490846, -0.42438055, 0.05310049, 0.02361122, 1.1982358, 1.4037777]$ y la cuádrica solución es el siguiente.

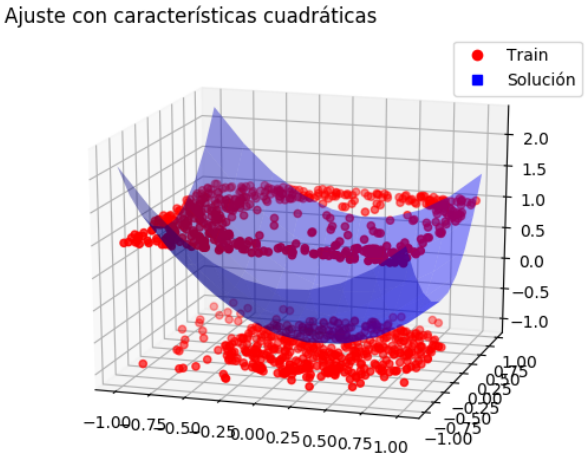


Figura 6: Plano obtenido utilizando características lineales

He repetido el experimento 1000 veces y he obtenido los siguiente errores medios:
E_{in} medio $= 0.5643661965437399$, E_{out} medio $= 0.5705303213340673$

En esta ocasión, E_{in} y E_{out} no están significativamente más distantes que en el modelo anterior a pesar del aumento de complejidad, y son bastante menores. En la figura se aprecia que la cuádrica obtenida (un paraboloide elíptico) aproxima mucho mejor la distribución que el plano del caso anterior.

Por tanto, es obvio que **el modelo con características no lineales es más adecuado**, ya que obtiene mayor precisión en el ajuste. Consigue un E_{out} menor, que es el principal objetivo de la regresión.