Contenido

[Regresión Lineal con una variable 1](#_Toc504300682)

[1.1 Introducción 1](#_Toc504300683)

[1.2 Cost function 4](#_Toc504300684)

[1.3 Intuición 7](#_Toc504300685)

[Manteniendo los dos parámetros 9](#_Toc504300686)

[1.4 Estimación de parámetros 12](#_Toc504300687)

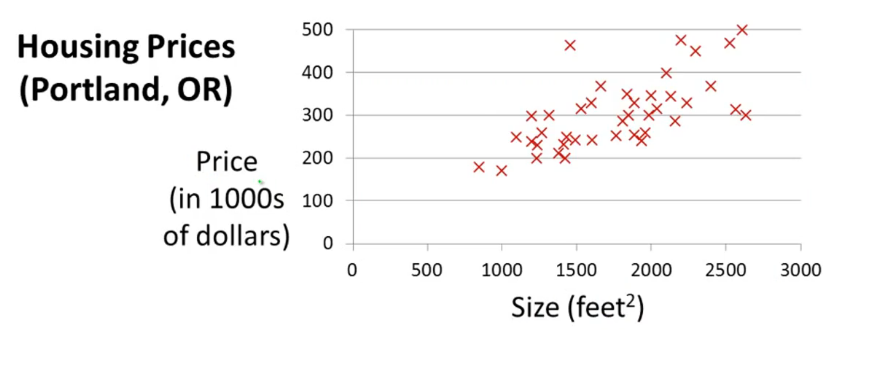
[1.5 Índice de aprendizaje y derivada. 17](#_Toc504300688)

[1.6 Descenso del gradiente para la regresión lineal. 20](#_Toc504300689)

# Regresión Lineal con una variable

## Introducción

Imaginemos que queremos predecir el precio de las viviendas de unan determinada zona según su tamaño de las cuales tenemos una muestra de las dos variables y representadas en un gráfico de dispersión quedarían de la siguiente manera:



Como se puede ver, en el eje horizontal tenemos distribuidas las casas según su tamaño (en pies cuadrados) y en el eje vertical tenemos el precio al que se vendieron. Con esta información podemos deducir diferentes ideas como por ejemplo que una casa en Portland que mide unos 1250 pies cuadrados se vende por alrededor de 200-300 mil dólares.

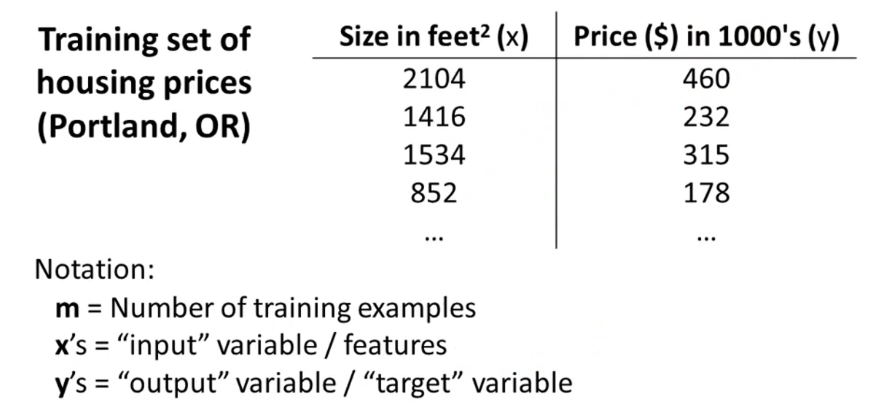
El objetivo principal es ajustar un modelo (en este caso podría ser una línea recta) en el que para cada tamaño elegido, dar un valor monetario exacto.

Este tipo de problemas se enmarca dentro del aprendizaje supervisado ya que para cada observación de la muestra tenemos la respuesta correcta para observarla. En este caso cual fuel el precio real de la casa para cada transacción.

Dentro del aprendizaje supervisado, si profundizamos más, a este tipo de modelos se les llama modelos de regresión ya que predecimos el resultado del valor real (outpout) a través de unos datos de entrada (input).

Más formalmente, en este tipo de problemas vamos a tener un conjunto de datos (dataset) que se llama conjunto de entrenamiento. Para este caso en particular un dataset con diferentes precios de viviendas y nuestro trabajo es aprender, a partir de estos datos, como predecir los precios de las viviendas.

**Ejemplo de dataset:**

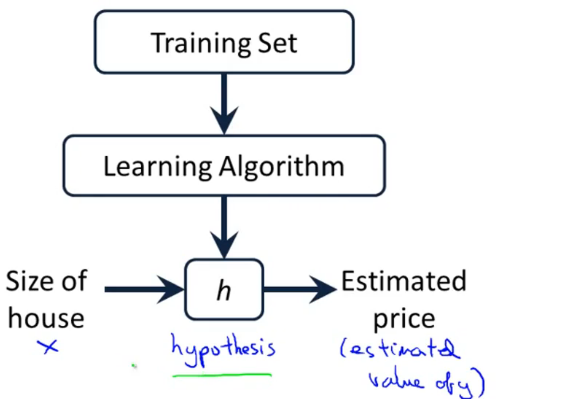


La notación son símbolos que hacen más fácil el trabajar con la teoría. En este curso se usará una notación que pasaremos a definir:

1. m: La letra m en minúscula representa durante este curso el número de ejemplos de entrenamiento. Si en el conjunto de datos tenemos 47 filas entonces tengo 47 ejemplos de entrenamiento y m=47.
2. xs: la letra x minúscula denota las variables de entrada o inputs que con frecuencia se llama features (funciones). En nuestro caso sería el tamaño de la vivienda.
3. y: La letra y denota las variables de salida u output, que volviendo a referirnos a este caso en particular sería el precio de las casas de nuestro conjunto de datos.
4. (x,y): Denota un solo ejemplo de entrenamiento. (2104,460) sería la casa que tiene de tamaño 2104 y precio 460 mil dólares.
5. (xi.yi): formalmente es equiparable al ejemplo inmediatamente superior. En este caso i= 1…47 y sirve para recorrer uno a uno el conjunto de entrenamiento y poder referirnos a uno o un conjunto de ellos fácilmente. Por lo tanto el súper índice i solo se referiría a una fila en concreto del conjunto de datos.

**¿Cómo funciona el algoritmo de aprendizaje supervisado?**

En primer lugar tenemos un conjunto de datos de entrenamiento (**Trainning Set**), este conjunto lo utilizamos para que nuestro **algoritmo aprenda** una función **h** (de hipótesis) que se ajuste a nuestro datos a través de unos datos de entrada y otros de salida. Esta hipótesis h no es más que una función que en terminología matemática sería:

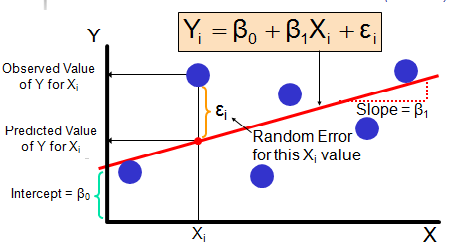


Puede que hipótesis no sea un buen nombre para este tipo de función pero fue el que se tomó en los principios de esta disciplina y por lo tanto es la terminología estándar que se utiliza.

Una vez nuestra función *h* está calculada el siguiente paso es decidir cómo se va a representar que será:

C:\Users\Diego\Desktop\Aprendiendo\Cursos\Machine learning Andrew NG\Teoría\apuntes ingles\Machine_learning_complete\01_02_Introduction_regression_analysis_and_gr_files\Image [7].png

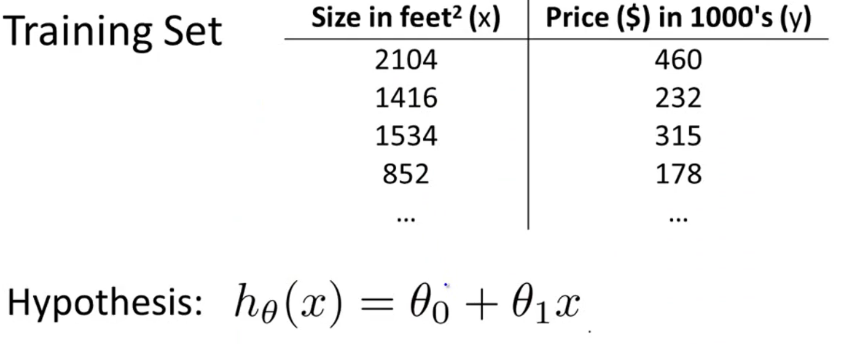
Esto no es más que una función lineal donde tenemos un término fijo que mide la altura de la función con respecto al origen y un parámetro que nos indica de la inclinación de la pendiente de la función.



## Cost function

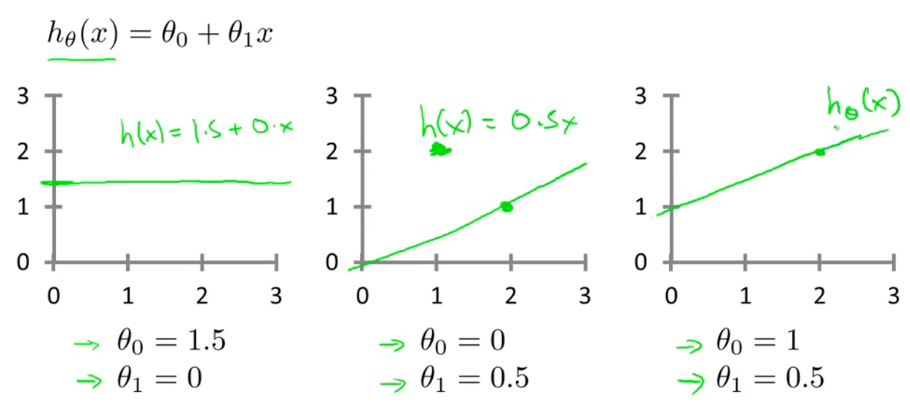
[chrome-extension://oemmndcbldboiebfnladdacbdfmadadm/http://wwwae.ciemat.es/~cardenas/docs/lessons/RegresionLineal.pdf](chrome-extension://oemmndcbldboiebfnladdacbdfmadadm/http:/wwwae.ciemat.es/~cardenas/docs/lessons/RegresionLineal.pdf)

En la regresión lineal tenemos un conjunto de entrenamiento como este. Donde tenemos una función *h* que nos permite hacer predicciones a través de unos datos de entrada.

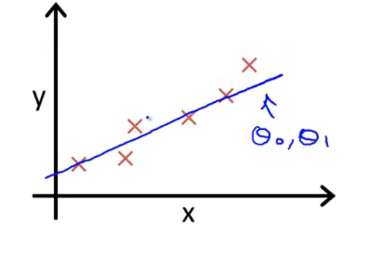


Las thetas son los denominados parámetros del modelo. Pero, ¿como determino yo cuales son los mejores valores de theta que se ajustan mejor a mi conjunto de datos?

Según escoja estos valores, obtendré diferentes tipos de hipótesis o funciones. Aquí tenemos diferentes ejemplos.



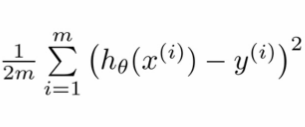
En la regresión lineal lo que queremos hacer es trazar una línea (matemáticamente obtener los parámetros θ de manera que la línea recta que obtenemos de alguna forma se ajuste bien a los datos.



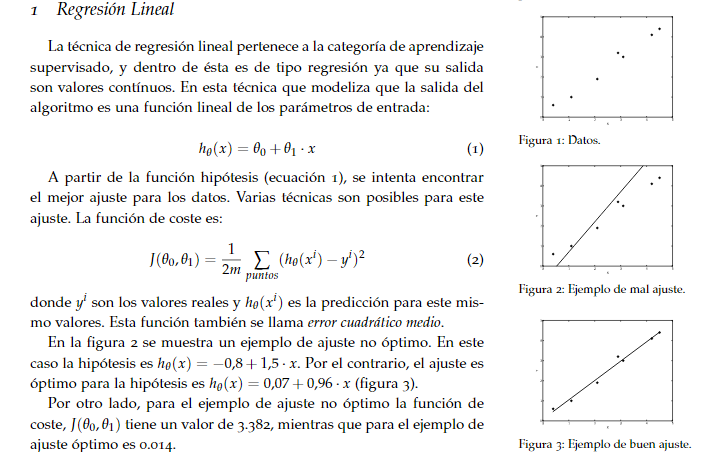
Formalmente, esto es un problema de minimización. Es decir quiero hallar el valor θ0 y θ1 que haga que los valores de entrada (x) de mi función, sean lo más cercanos posibles a los valores de salida que yo observo (y).

Por lo tanto, todas mis hθ(*xi)* y mis *yi* deben de ser lo más parecidas posibles.

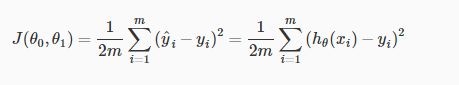
En el ejemplo de la casa, tengo que obtener una función que al pasarle los tamaños de las casas, me devuelva el precio más cercano posible al precio real de la casa para todo mi conjunto de datos. El problema formalmente sería asi:



Donde tenemos que encontrar los parámetros θ de manera que se reduzca el promedio de 1 dividido entre 2\*m veces la suma de los errores cuadráticos entre mis predicciones del conjunto de entrenamiento hθ(*xi)* menos los valores reales de las casas en el conjunto de entrenamiento *yi.*. Al poner 2 la constante de un medio hace que las matemáticas sean un poco más fáciles (al derivar, más adelante se verá). Entonces, al minimizar la mitad de algo, debe darte los mismos valores de los parámetros θ0 y θ1



Por lo tanto, con la función de costes podemos medir la precisión de nuestra función de hipótesis. Esto toma una diferencia promedio (en realidad, una versión más elegante de un promedio) de todos los resultados de la hipótesis con entradas de x y la salida real de y.

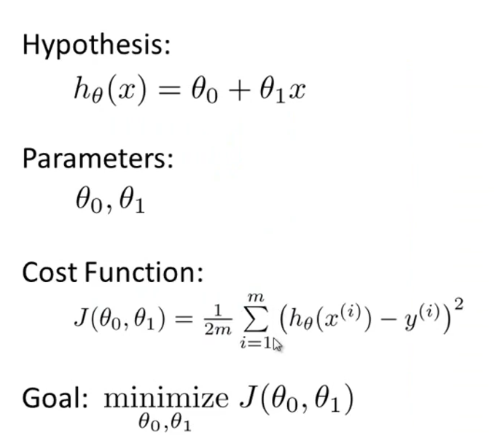


Aparte donde es la media de los errores de  o la diferencia entre el valor predicho y el valor real. Esta función es además llamada "Función de error cuadrado" o "Error cuadrático medio". La media se reduce a la mitad ) como una conveniencia para el cálculo del descenso del gradiente, ya que el término derivado de la función cuadrada cancelará la fracción.

## Intuición

Como explicamos anteriormente tenemos un problema donde queremos ajustar una línea recta a nuestros datos mediante una función llamada hipótesis que tiene dos parámetros θ0 y θ1.

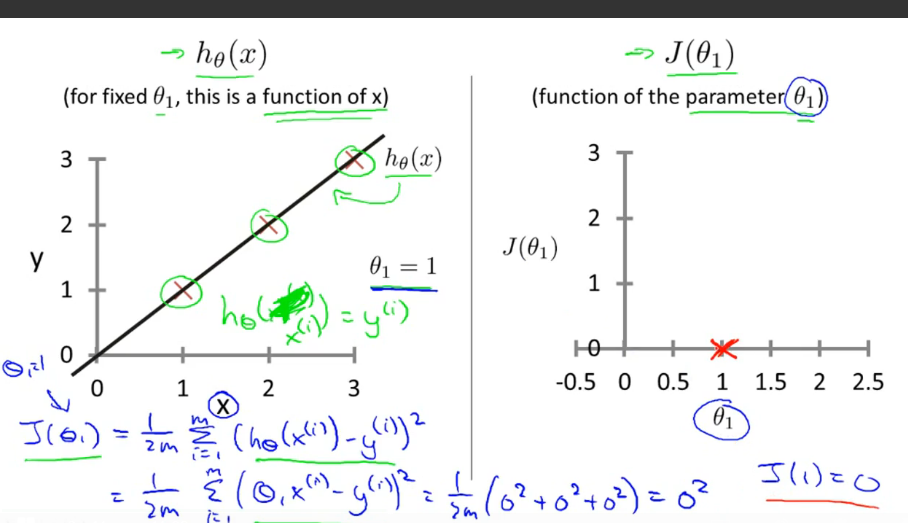
De ahí se obtiene un problema de minimización que se formaliza y se obtiene una función de costos *J* que depende de θ0 y θ1 cuyo objetivo es que sea lo más pequeña posible.



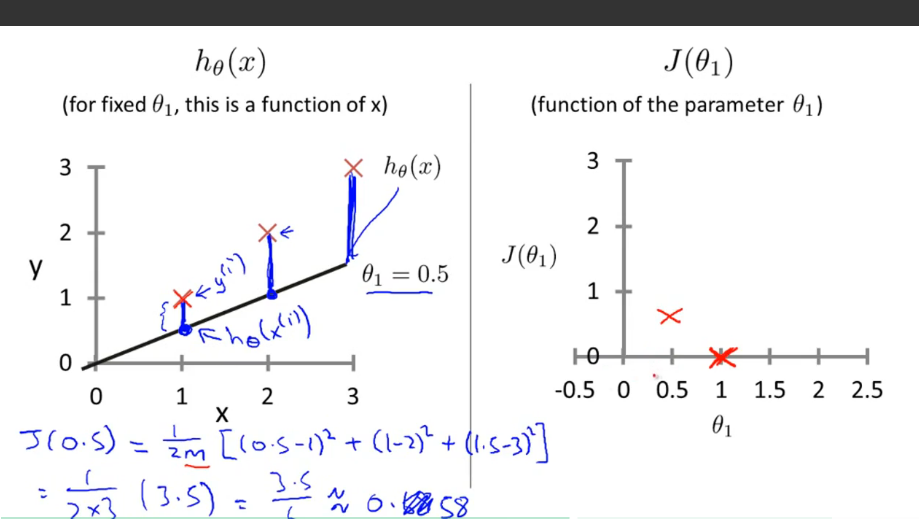
Imaginemos que el termino θ0 = 0, es decir, solo hemos impuesto que la línea recta pase si o si por el origen quedando hθ = θ1\*x. Recordemos que θ1 se encargaba de controlar la pendiente de la función.

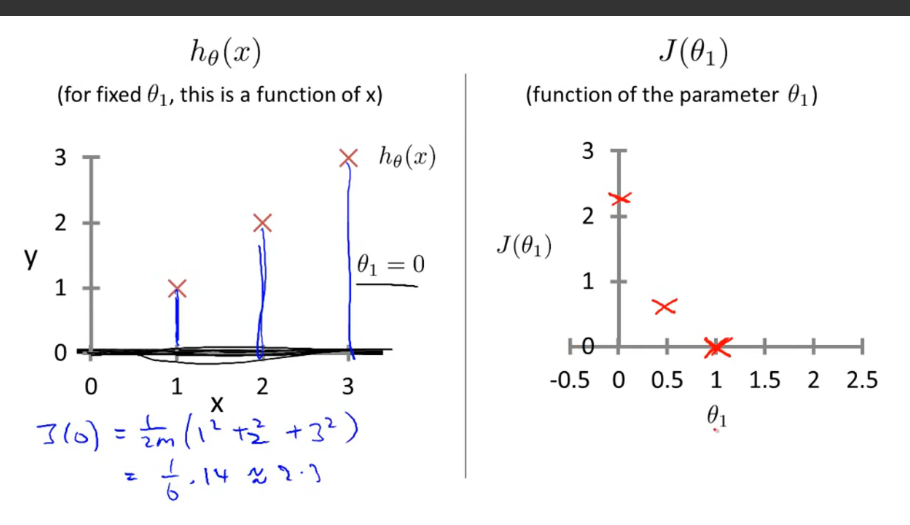
Entonces, imaginemos que tenemos tres valores (1,1), (2,2), (3,3).

Si establecemos un valor para θ1 = 1 la pendiente será igual a 1 y el ajuste será perfecto, siendo la función de coste = 0



Pero qué pasa con diferentes valores para el parámetro θ1



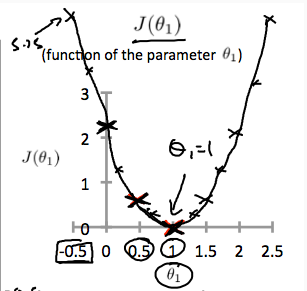


Si tratamos de pensarlo en términos visuales, nuestro conjunto de datos de entrenamiento está disperso en el plano xy. Estamos tratando de hacer una línea recta (definida por hθ(x)) que pasa a través de estos puntos de datos dispersos.

Nuestro objetivo es obtener la mejor línea posible. La mejor línea posible será tal que las distancias verticales cuadradas promedio de los puntos dispersos de la línea sean las mínimas. Idealmente, la línea debería pasar por todos los puntos de nuestro conjunto de datos de entrenamiento. En tal caso, el valor de J(θ0 ,θ1) será 0.

Cuando θ1 = 1 obtenemos una pendiente de 1 que pasa por cada punto de datos en nuestro modelo. Por el contrario, cuando θ1 = 0.5 vemos que la distancia vertical de nuestro ajuste a los puntos de datos aumenta.

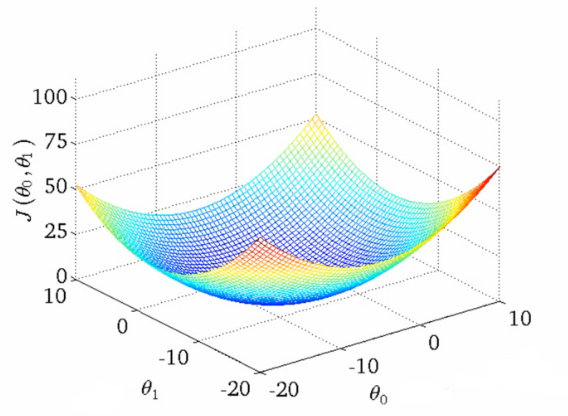
Esto aumenta nuestra función de costo a 0.58. Al trazar otros puntos se obtiene el siguiente gráfico:



Por lo tanto, como objetivo, debemos tratar de minimizar la función de costos. En este caso, θ1 = 1 es nuestro mínimo global.

### Manteniendo los dos parámetros

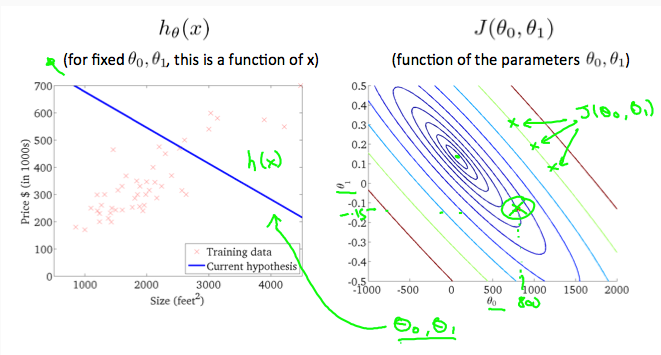
Ahora la función de costes es más complicada, ya que solo tenemos un parámetro, ahora hay dos, por lo tanto la representación gráfica pasa de lineal a 3D. La forma sigue siendo similar, una especie de arco que puede ser muy similar a esta:



Esto es una gráfica de superficie en 3D, donde los ejes tienen asignado el valor de θ0 yθ1. Los dos parámetros pueden cambiar en diferentes direcciones y diferentes combinaciones de ellos pueden marcar una altura diferente para nuestra función de costes.

Estas funciones 3D pueden pasar a ser funciones de superficie o lo que también se llama *“contour plots”* o figuras de contorno en español.

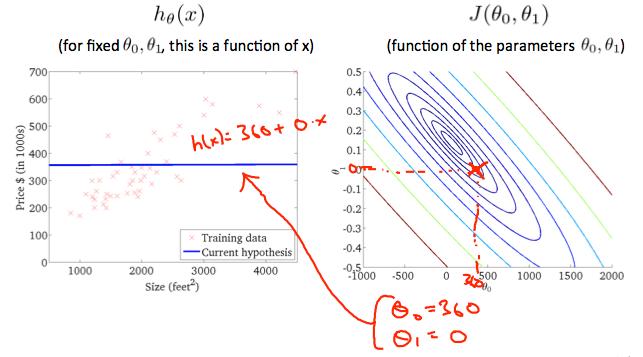
Un diagrama de contorno es un gráfico que contiene muchas líneas de contorno. Una línea de contorno de una función de dos variables tiene un valor constante en todos los puntos de la misma línea. Un ejemplo de dicho gráfico es el de la derecha a continuación.



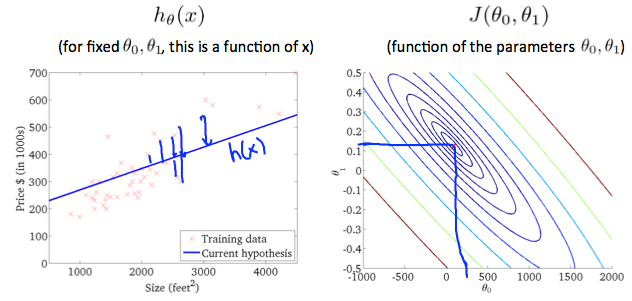
Tomando cualquier color y yendo a lo largo del 'círculo', uno esperaría obtener el mismo valor de la función de costo. Por ejemplo, los tres puntos verdes que se encuentran en la línea verde de arriba tienen el mismo valor para J(θ0 ,θ1) como resultado, se encuentran en la misma línea. El círculo x muestra el valor de la función de costo para el gráfico de la izquierda cuando θ0 = 800 y θ1 = -0.15.

. Solo tenemos que imaginar la figura en 3D anterior que se ha aplastado y cada elipse tenía la misma altura o que estamos mirando desde arriba perdiendo la profundidad del eje vertical o los valores de J.

Tomando otra h (x) y trazando su gráfica de contorno, uno obtiene los siguientes gráficos:



Cuando θ0 = 360 y θ1 = 0 el valor para J(θ0 ,θ1) en el diagrama de contorno se acerca al centro, lo que reduce el error de la función de costo. Ahora, al darle a nuestra función de hipótesis una pendiente ligeramente positiva, se obtiene un mejor ajuste de los datos.



El gráfico anterior minimiza la función de costo tanto como sea posible y, en consecuencia, el resultado de θ0  y θ1  que tienden a ser alrededor de 0.12 y 250 respectivamente. Trazar esos valores en nuestro gráfico a la derecha parece poner nuestro punto en el centro del círculo más interno.

## Estimación de parámetros

**El algoritmo del descenso del gradiente**

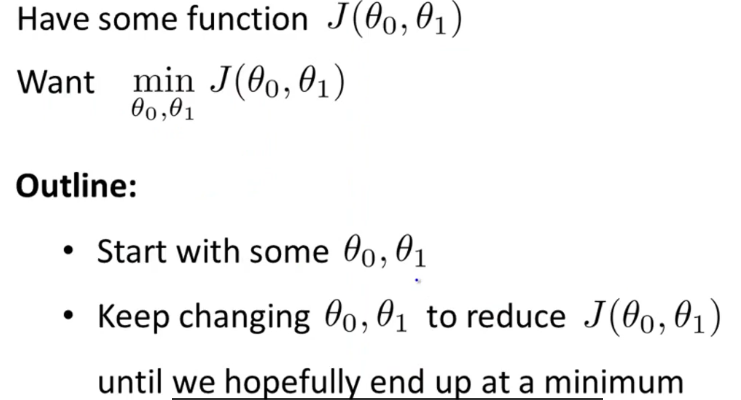
Este concepto puede ser uno de los más importantes del machine learning ya que de hecho, se utiliza en prácticamente todo el aprendizaje automático.

Tenemos una función hipóteis J(θ0 ,θ1) e incluso J(θ0 ,θ1…θn). El objetivo siempre será encontrar la combinación de parámetros para que nuestra función de coste J sea lo más pequeña posible.

Resulta que la gradiente de descenso es un algoritmo para resolver este problema más general pero de momento solo lo usaremos con dos parámetros.

Entonces, comenzamos con dos valores iniciales para θ0 y θ1, realmente no importa mucho cuales sean pero una opción común θ0 = 0 y θ1 = 0.

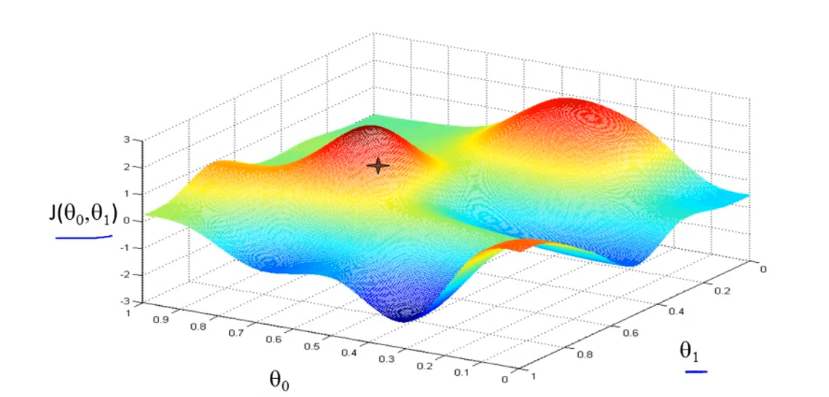
Lo que hace el algoritmo es mover el valor de los parámetros en pequeños saltos para tratar de reducir J hasta que con suerte obtengamos un mínimo, (Que puede ser local o global)



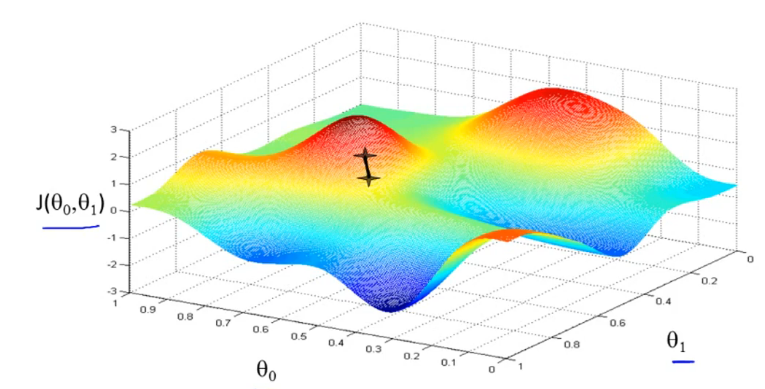
Imagine que graficamos nuestra función de hipótesis en función de sus campos θ0 y θ1 (en realidad estamos graficando la función de costo como una función de las estimaciones de los parámetros). No estamos graficando x e y en sí, sino el rango de parámetros de nuestra función de hipótesis y el costo resultante de seleccionar un conjunto particular de parámetros.

Ponemos θ0 en el eje x y θ1 en el eje y, con la función de costo en el eje z vertical. Los puntos en nuestro gráfico serán el resultado de la función de costo usando nuestra hipótesis con esos parámetros theta específicos. El siguiente gráfico muestra dicha configuración.

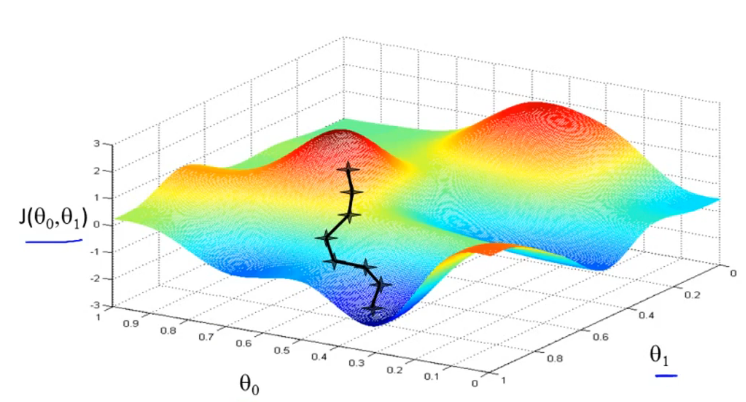
Un ejemplo, digamos que tenemos que minimizar la siguiente función J empezando por los valores (0, 0) representados por esa estrellita:



Imagina que esto es un paisaje de algún parque con zonas verdes y dos colinas. Estás parado físicamente en ese punto (0, 0) en esa pequeña colina roja del parque. En el gradiente de descenso, lo que haremos es girar 360 grados y ver todo nuestro alrededor y preguntarnos: ¿Si fuera a dar un pequeño paso en alguna dirección y quisiera bajar la colina lo más rápido posible ¿qué dirección tomaría ese pequeño paso si quiero ir para abajo, si quiero bajar físicamente de esta colina tan rápido como sea posible? Resulta que si estuvieras parado en ese punto de la colina, verías todo el alrededor y descubrirías que la mejor dirección para un pequeño descenso hacia abajo es más o menos en la siguiente dirección:

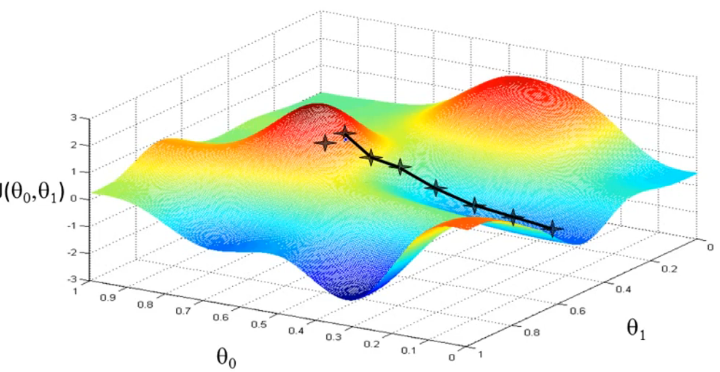


Ahora estás en un nuevo punto sobre tu colina. Una vez más, verás a tu alrededor y dirás: ¿Qué dirección debo tomar para da un pequeño paso cuesta abajo?, y asi sucesivamente hasta converger a algún mínimo local:



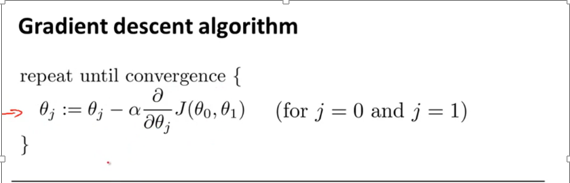
El mayor descenso tiene una propiedad interesante. Y es que la obtención de un mínimo local depende en gran medida del lugar en el que coloquemos los puntos de partida para empezar a iterar. Sabremos que hemos tenido éxito cuando nuestra función de costos se encuentra en el fondo de los pozos en nuestro gráfico, es decir, cuando su valor es el mínimo. Las flechas rojas muestran los puntos mínimos en el gráfico.

La forma en que hacemos esto es tomando la derivada (la línea tangencial de una función) de nuestra función de costo. La pendiente de la tangente es la derivada en ese punto y nos dará una dirección para avanzar. Bajamos la función de costos en la dirección con el descenso más inclinado. El tamaño de cada paso está determinado por el parámetro α, que se denomina tasa de aprendizaje. Por ejemplo, la distancia entre cada 'estrella' en el gráfico anterior representa un paso determinado por nuestro parámetro α. Un α más pequeño daría como resultado un paso más pequeño y un α más grande da como resultado un paso más grande.

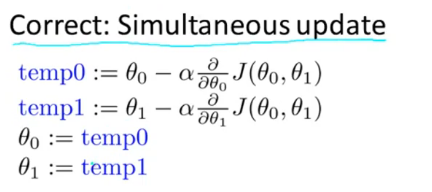


La dirección en la que se toma el paso está determinada por la derivada parcial de J(θ0, θ1). Dependiendo de dónde uno comienza en el gráfico, uno podría terminar en diferentes puntos. La imagen de arriba nos muestra dos puntos de partida diferentes que terminan en dos lugares diferentes.

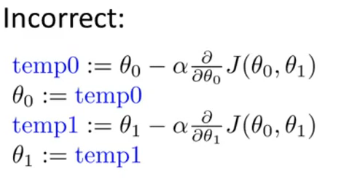
El gradiente de descenso te habría llevado al segundo óptimo local a la derecha.



* Usamos “:=” para denotar asignación. En concreto, si escribo “a:= b” lo que significa que en una computadora es tomar el valor b y usarlo para sobrescribir a. Es decir, establecemos a para que sea igual al valor de b (También puedo usar “a:= b+1” esto significa que a tomará siempre el valor de b más una unidad.). Si uso “a=b” será una afirmación verdadera, es decir, estoy afirmando que el valor de a es igual al valor de b. Así que el primero podíamos decir que es una operación de computadora donde establecemos cual será el valor de “a”. El segundo es una afirmación, solo afirmo que los valores a y b son iguales.
* Usamos α que es un número al que denominaremos índice de aprendizaje. Y lo que hace es básicamente, controlar que tan grande es un paso que damos cuesta abajo con el gradiente de descenso. Si alfa es muy grande, entonces corresponde a un descenso del gradiente muy agresivo, donde tratamos de dar pasos grandes cuesta abajo. Y si alfa es pequeño, entonces damos pasos muy pequeños.
*  Finalmente ese término indica la derivada, qué hablaremos de el más adelante.
* Existe una sutileza más sobre el gradiente de descenso. En gradiente de descenso, vamos a actualizar θ0  y θ1 esta actualización ocurre cuando j=0 y j=1. Así que vamos a actualizar J, θ0  y θ1. Y la sutileza de como aplicas el gradiente de descenso, para la expresión de arriba, es que quieres actualizar de forma simultanea θ0  y θ1 con esta me refiero que en esta ecuación vamos a actualizar “θ0:= θ0” y “θ1:= θ1” menos algo.



La forma de aplicarlo es que debes calcular el lado derecho de la ecuación (lo que está restando) tanto para θ0  como para θ1 y actualizar ambos al mismo tiempo. En la imagen superior establecemos temp0:= a θ0  menos el índice de aprendizaje multiplicado por la derivada de la función respecto de theta 0. Y lo mismo con temp1 pero usando la derivada de J con respecto de θ1. Así que básicamente calculamos los lados derechos. Y después de haber calculado los derechos y almacenarlos juntos en temp0 y temp1 actualizaremos θ0 y θ1 de forma simultánea.



Una aplicación incorrecta sería aquella en la cual los parámetros no se actualizan de forma simultánea ya que estaríamos utilizando el valor de θ0 para calcular el nuevo θ1 y esto dará un valor diferente de temp1.

## Índice de aprendizaje y derivada.

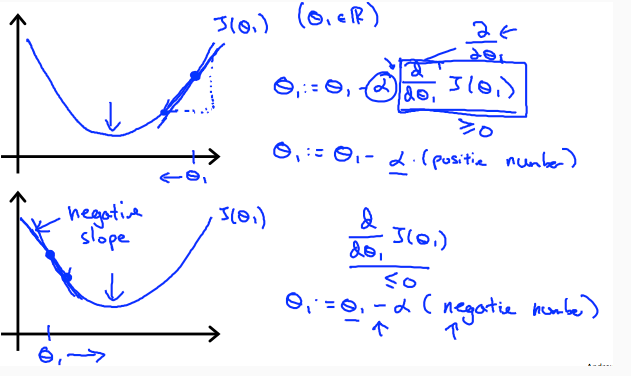
Exploramos el escenario donde usamos un parámetro θ1 que pertenece a los números reales con el fin de representar graficas en una dimensión y trazamos su función de costo para implementar un descenso de gradiente. Nuestra fórmula para un solo parámetro es:



Como se observa ahora usamos el símbolo d, ya que como solo hay una derivda estamos derivando respecto siempre al mismo término.

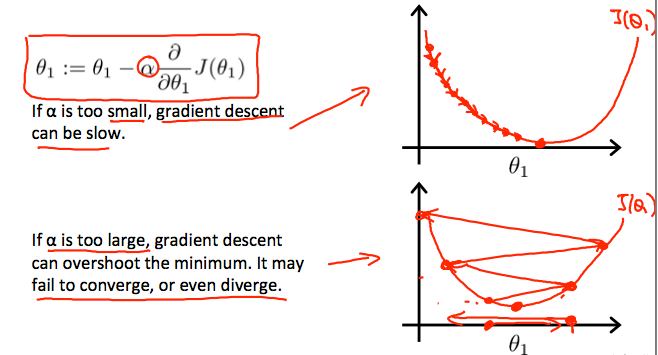
Recordemos que una derivada en un punto concreto es decirnos básicamente la tangente a ese punto. La tangente al final es una línea con una determinada pendiente que es donde está la derivada. Por lo tanto, la derivada nos dice cuál es la pendiente de esa línea tangente que pasa por ese punto en concreto de nuestra función. Si la línea tiene una pendiente positiva es que tiene una derivada positiva entonces será decrecer (menos por mas igual a menos)

Independientemente del signo de pendiente para θ1 finalmente converge a su valor mínimo. El siguiente gráfico muestra que cuando la pendiente es negativa, el valor de θ1 aumenta y cuando es positivo, el valor de θ1 disminuye.



Debemos ajustar nuestro parámetro α para garantizar que el algoritmo de descenso de gradiente converja en un tiempo razonable. La falta de convergencia o demasiado tiempo para obtener el valor mínimo implica que nuestro tamaño de paso es incorrecto.

Cuando fijamos un paso demasiado grande como en el segundo ejemplo aunque estemos muy cerca del mínimo damos un salto tan grande que empeoremos mi función de costes, entonces cambie de dirección pero otra vez saltaría a un más pequeño que el primero debido al elevado índice de aprendizaje y así sucesivamente hasta llegar a infinito,

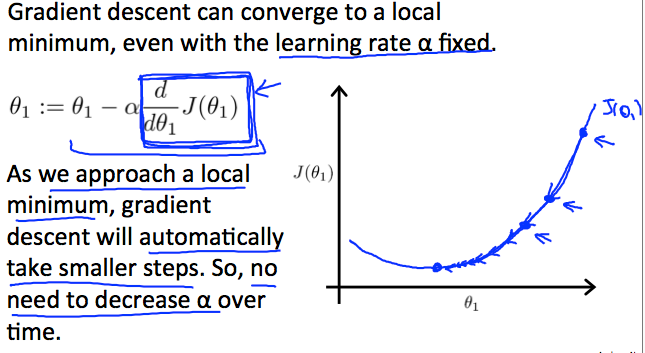


**¿Cómo converge el descenso de gradiente con un tamaño de paso fijo α?**

La intuición detrás de la convergencia es que se acerca a 0 a medida que nos acercamos al fondo de nuestra función convexa. Como mínimo, la derivada siempre será 0 y así obtenemos:



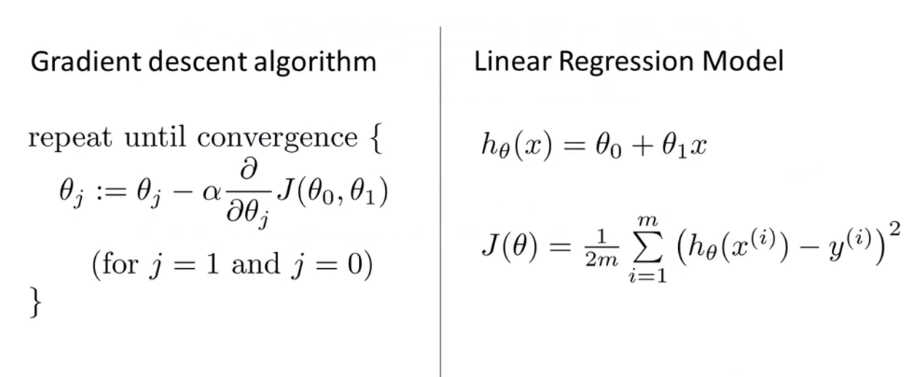
Recordamos que cuando la derivada es 0, la linea tangente pasa a ser plana, lo que la deja sin pendiente o lo que es lo mismo, la pendiente es igual a 0. Conforme se va acercando a ese punto 0 la pendiente va disminuyendo y por lo tanto aunque tengamos un paso fijo, la reducción (o aumento) del parámetro por cada interacción va decreciendo proporcinalmente.



## Descenso del gradiente para la regresión lineal.

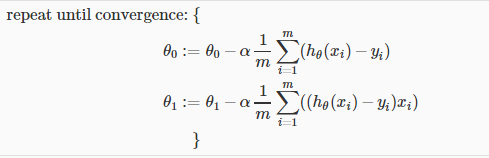
En esta sección, vamos a unir el gradiente de descenso con nuestra función de costes obteniendo un algoritmo para la regresión lineal que conseguirá ajustar una línea recta para nuestros datos.

En las anteriores secciones teníamos nuestro algoritmo para el gradiente de descenso y el modelo de regresión lineal con nuestra hipótesis lineal y nuestra función de error cuadrática:



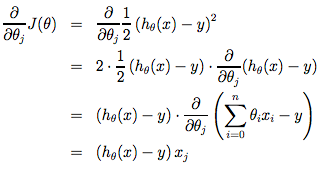
Lo que vamos a hacer es unirlas para minimizar nuestras función del error al cuadrado.

Cuando se aplica específicamente al caso de la regresión lineal, se puede derivar una nueva forma de ecuación de descenso del gradiente. Podemos sustituir nuestra función de costo real y nuestra función de hipótesis real y modificar la ecuación para:



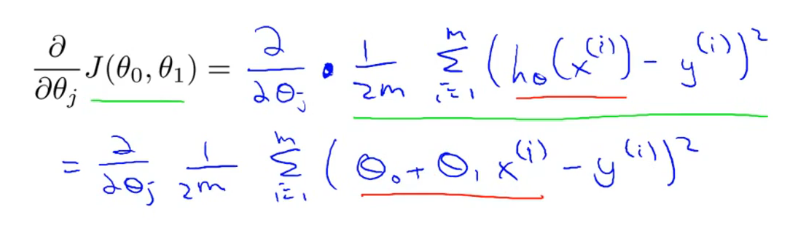
Donde m es el tamaño del conjunto de entrenamiento, θ0 una constante que cambiará simultáneamente con θ1 y xi, yi son valores del conjunto de entrenamiento dado (datos).

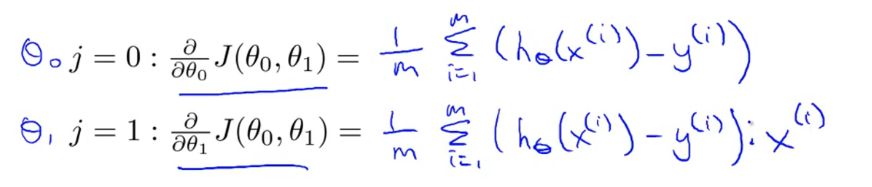
Tenga en cuenta que hemos separado los dos casos para θj en ecuaciones separadas para θ0 y θ1; y que para θ1 estamos multiplicando xi al final debido a la derivada. Lo siguiente es una derivación de J (θ) para un solo ejemplo:



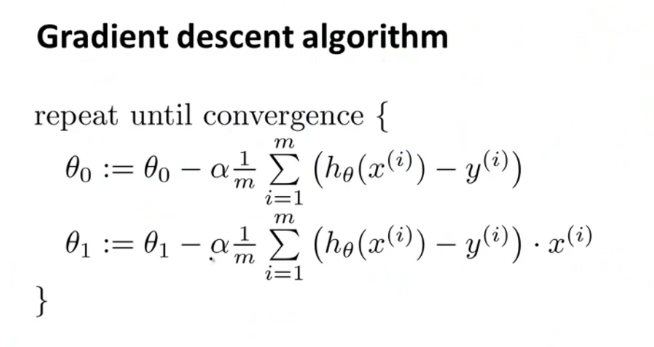
El objetivo de todo esto es que si comenzamos con una suposición para nuestra hipótesis y luego aplicamos repetidamente estas ecuaciones de descenso de gradiente, nuestra hipótesis será cada vez más precisa.

Entonces, la cuestión principal es obtener el valor de la derivada parcial, que se hace derivando nuestra función de costos J respecto de sus parámetros.

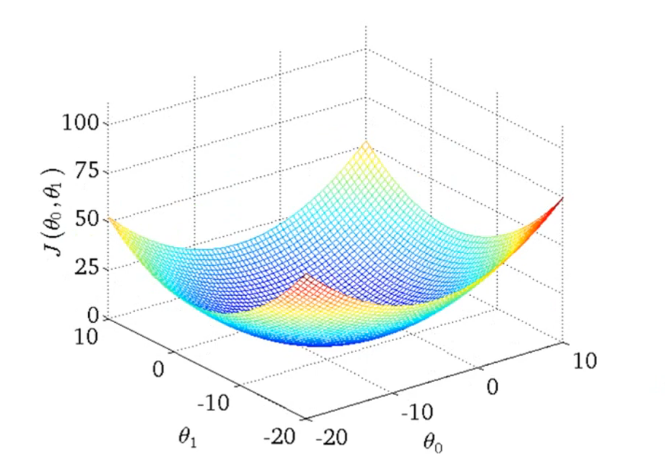




Ahora, nuestro algoritmo de gradiente de descenso quedaría:

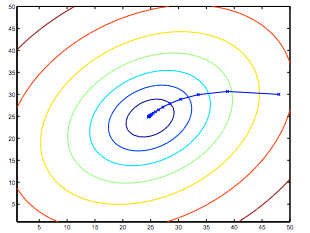


Por lo tanto, esto es simplemente un descenso en pendiente en la función de costo original J. Este método analiza cada ejemplo en el conjunto de entrenamiento completo en cada paso, y se denomina descenso de gradiente por lotes. Tenga en cuenta que, aunque el descenso del gradiente puede ser susceptible a los mínimos locales en general, el problema de optimización que hemos planteado aquí para la regresión lineal tiene solo una óptima global, y ningún otro local, ya que la regresión lineal siempre tiene forma de arco:



El término técnico para esto es función convexa, que de maneara informal es una función con forma curva, como de arco que hace que el óptimo local siempre sea global (porque solo hay uno).

Por lo tanto, el descenso gradual siempre converge (suponiendo que la tasa de aprendizaje α no es demasiado grande) al mínimo global. De hecho, J es una función cuadrática convexa. Aquí hay un ejemplo de descenso de gradiente a medida que se ejecuta para minimizar una función cuadrática.



Las elipses que se muestran arriba son los contornos de una función cuadrática. También se muestra la trayectoria tomada por el descenso del gradiente, que se inicializó en (48,30). Las x en la figura (unidas por líneas rectas) marcan los valores sucesivos de θ que el descenso del gradiente pasó a medida que convergía a su mínimo.

Como dato, hay que saber que a veces, el algoritmo de descenso de gradiente a veces se llama descenso de gradiente por lotes (batches) y resulta que en el aprendizaje automático y se refiere que en cada paso vemos todos los ejemplos de entrenamiento.