Gradient Descent

2020年8月9日 10:09

Gradient Descent

Review: Gradient Descent

- In step 3, we have to solve the following optimization problem:

$$\theta^* = \arg\min_{\theta} L(\theta) \qquad \text{L: loss function} \qquad \theta\text{: parameters}$$

loss 求最小

Suppose that θ has two variables $\{\theta_1, \theta_2\}$ 写得更简洁.

Randomly start at $\theta^0 = \begin{bmatrix} \theta_1^0 \\ \theta_2^0 \end{bmatrix}$ 就叫 Gradient → $\boxed{\nabla L(\theta) = \begin{bmatrix} \partial L(\theta_1)/\partial \theta_1 \\ \partial L(\theta_2)/\partial \theta_2 \end{bmatrix}}$

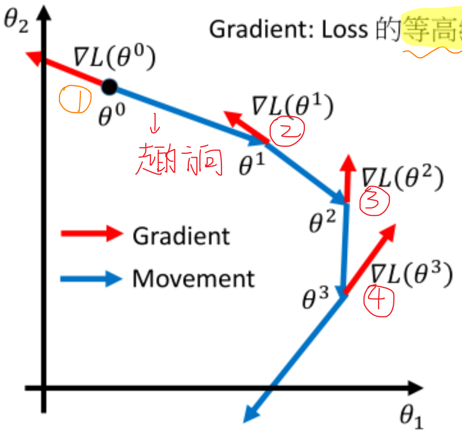
$$\begin{bmatrix} \theta_1^1 \\ \theta_2^1 \end{bmatrix} = \begin{bmatrix} \theta_1^0 \\ \theta_2^0 \end{bmatrix} - \eta \underline{\begin{bmatrix} \partial L(\theta_1^0)/\partial \theta_1 \\ \partial L(\theta_2^0)/\partial \theta_2 \end{bmatrix}} \Rightarrow \theta^1 = \theta^0 - \eta \underline{\nabla L(\theta^0)}$$

L在θ^0的参数

$$\begin{bmatrix} \theta_1^2 \\ \theta_2^2 \end{bmatrix} = \begin{bmatrix} \theta_1^1 \\ \theta_2^1 \end{bmatrix} - \eta \underline{\begin{bmatrix} \partial L(\theta_1^1)/\partial \theta_1 \\ \partial L(\theta_2^1)/\partial \theta_2 \end{bmatrix}} \Rightarrow \theta^2 = \theta^1 - \eta \underline{\nabla L(\theta^1)}$$

Review: Gradient Descent

Gradient: Loss 的等高線的法線方向



Start at position θ^0

Compute gradient at θ^0

Move to $\theta^1 = \theta^0 - \eta \nabla L(\theta^0)$

Compute gradient at θ^1

Move to $\theta^2 = \theta^1 - \eta \nabla L(\theta^1)$

⋮

Gradient Descent

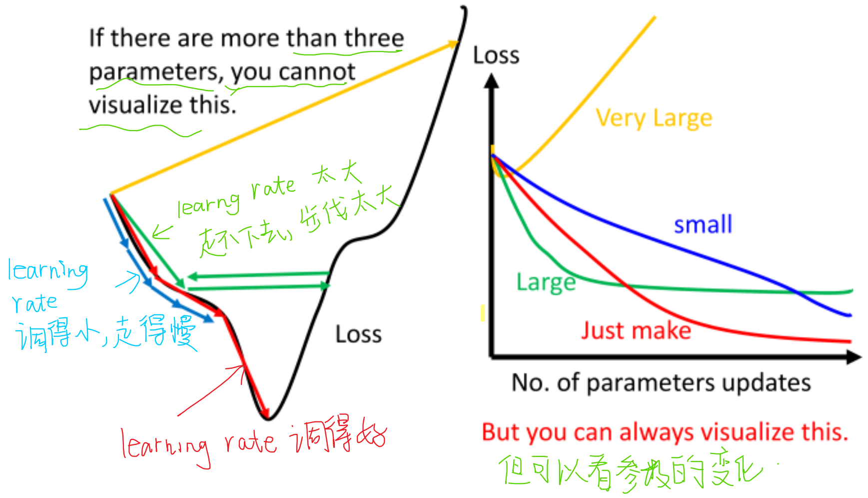
Tip 1: Tuning your learning rates

小心调

Learning Rate

$$\theta^i = \theta^{i-1} - \eta \nabla L\left(\theta^{i-1}\right)$$

Set the learning rate η carefully



But you can always visualize this.

但可以看参数的变化.

自动调 Learning Rate

Adaptive Learning Rates

- Popular & Simple Idea: Reduce the learning rate by some factor every few epochs.
  - At the beginning, we are far from the destination, so we use larger learning rate
  - After several epochs, we are close to the destination, so we reduce the learning rate　好几次后
  - E.g. 1/t decay: $\eta^t = \eta / \sqrt{t+1}$　η 与次数有关.
- Learning rate cannot be one-size-fits-all
  - Giving different parameters different learning rates

不同参数给不同的 Learning Rate

Adagrad

$$\eta^t = \frac{\eta}{\sqrt{t+1}} \qquad g^t = \frac{\partial L(\theta^t)}{\partial w}$$

- Divide the learning rate of each parameter by the ***root mean square of its previous derivatives***

之前　微分值

Vanilla Gradient descent

$$w^{t+1} \leftarrow w^t - \eta^t g^t$$

w

w is one parameters

Adagrad

$$w^{t+1} \leftarrow w^t - \frac{\eta^t}{\sigma^t} g^t$$

σ^t: ***root mean square*** of the previous derivatives of parameter w

Parameter dependent

parameter w

Parameter dependent

每个参数的σ^t都不同

Adagrad

σ^t: ***root mean square*** of the previous derivatives of parameter w

过去算过的微分值的

$$w^1 \leftarrow w^0 - \frac{\eta^0}{\sigma^0} g^0 \qquad \sigma^0 = \sqrt{(g^0)^2}$$

$$w^2 \leftarrow w^1 - \frac{\eta^1}{\sigma^1} g^1 \qquad \sigma^1 = \sqrt{\frac{1}{2}\left[(g^0)^2 + (g^1)^2\right]}$$

$$w^3 \leftarrow w^2 - \frac{\eta^2}{\sigma^2} g^2 \qquad \sigma^2 = \sqrt{\frac{1}{3}\left[(g^0)^2 + (g^1)^2 + (g^2)^2\right]}$$

$$\vdots$$

$$w^{t+1} \leftarrow w^t - \frac{\eta^t}{\sigma^t} g^t \qquad \sigma^t = \sqrt{\frac{1}{t+1}\sum_{i=0}^{t}(g^i)^2}$$

Adagrad

- Divide the learning rate of each parameter by the ***root mean square of its previous derivatives***

$$w^{t+1} \leftarrow w^t - \frac{\eta^t}{\sigma^t} g^t$$

$$\eta^t = \frac{\eta}{\cancel{\sqrt{t+1}}} \quad \text{1/t decay}$$

$$\sigma^t = \sqrt{\frac{1}{\cancel{t+1}}\sum_{i=0}^{t}(g^i)^2}$$

$$w^{t+1} \leftarrow w^t - \frac{\eta}{\sqrt{\sum_{i=0}^{t}(g^i)^2}} g^t$$

（化简后）

把 learning rate 写成这样

Contradiction?

$$\eta^t = \frac{\eta}{\sqrt{t+1}} \qquad g^t = \frac{\partial L(\theta^t)}{\partial w}$$

取决于 η 和 gradient

Vanilla Gradient descent

$$w^{t+1} \leftarrow w^t - \eta^t g^t$$

Larger gradient, larger step

参数更新↑

Adagrad

$$w^{t+1} \leftarrow w^t - \frac{\eta}{\ldots} g^t$$

Larger gradient, larger step

这样

Contradiction? $\eta^t = \frac{\eta}{\sqrt{t+1}}$ $g^t = \frac{\partial L(\theta^t)}{\partial w}$

取决于 η 和 gradient

Vanilla Gradient descent

$$w^{t+1} \leftarrow w^t - \eta^t \underline{g^t}$$

→ Larger gradient, larger step 参数更新↑

Adagrad

$$w^{t+1} \leftarrow w^t - \frac{\eta}{\sqrt{\underline{\sum_{i=0}^{t}(g^i)^2}}} \underline{g^t}$$

- g^t → Larger gradient, larger step
- $\sum_{i=0}^{t}(g^i)^2$ → Larger gradient, smaller step

和我们期望相反

Intuitive Reason $\eta^t = \frac{\eta}{\sqrt{t+1}}$ $g^t = \frac{\partial C(\theta^t)}{\partial w}$

- How surprise it is 反差

特别大

g^0	g^1	g^2	g^3	g^4	……
0.001	0.001	0.003	0.002	0.1	……

g^0	g^1	g^2	g^3	g^4	……
10.8	20.9	31.7	12.1	0.1	……

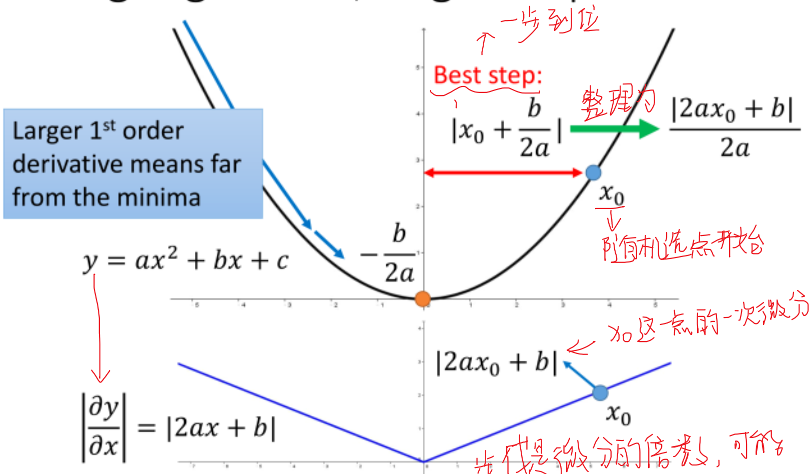
特别小

$$w^{t+1} \leftarrow w^t - \frac{\eta}{\sqrt{\underline{\sum_{i=0}^{t}(g^i)^2}}} g^t$$

$\sum_{i=0}^{t}(g^i)^2$ → 造成反差的效果 多少

过去的gradient有多大

Larger gradient, larger steps?



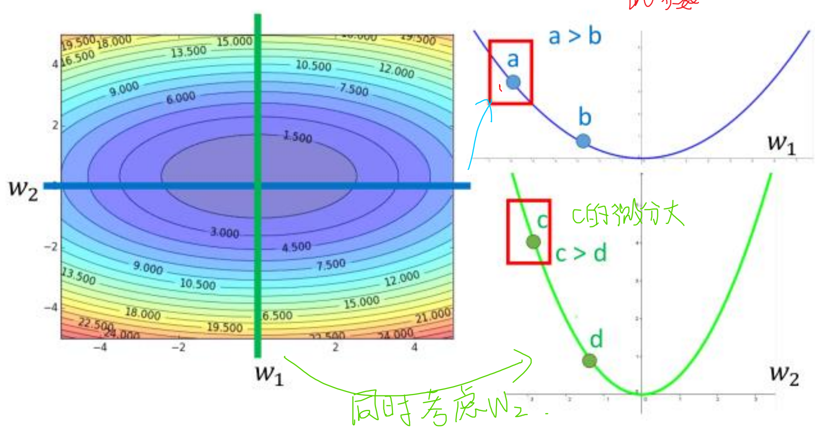
不考虑跨参数，只有一个参数

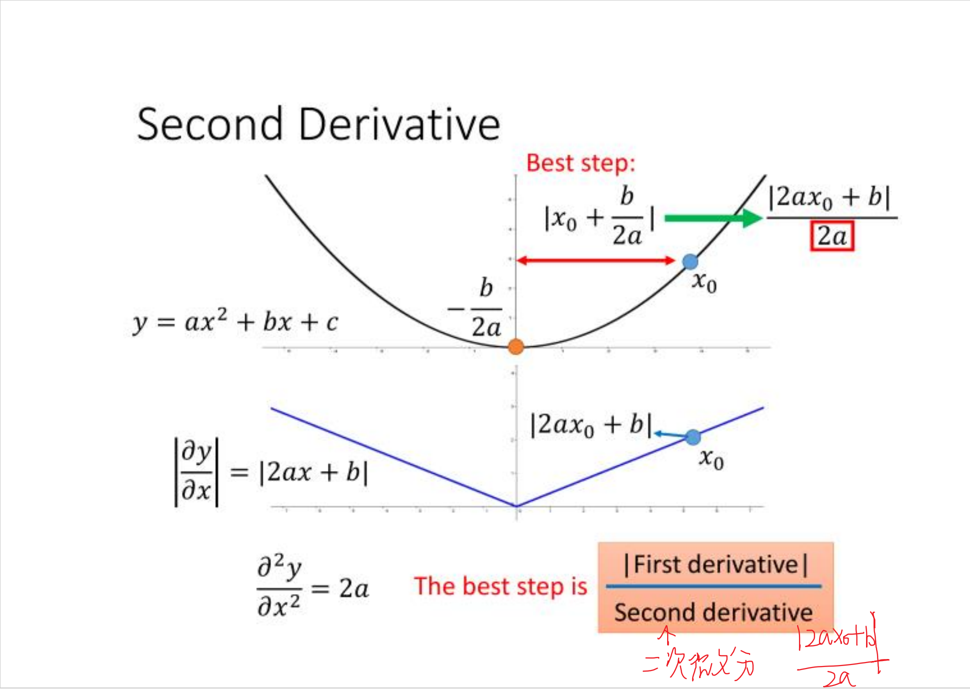
Comparison between different parameters

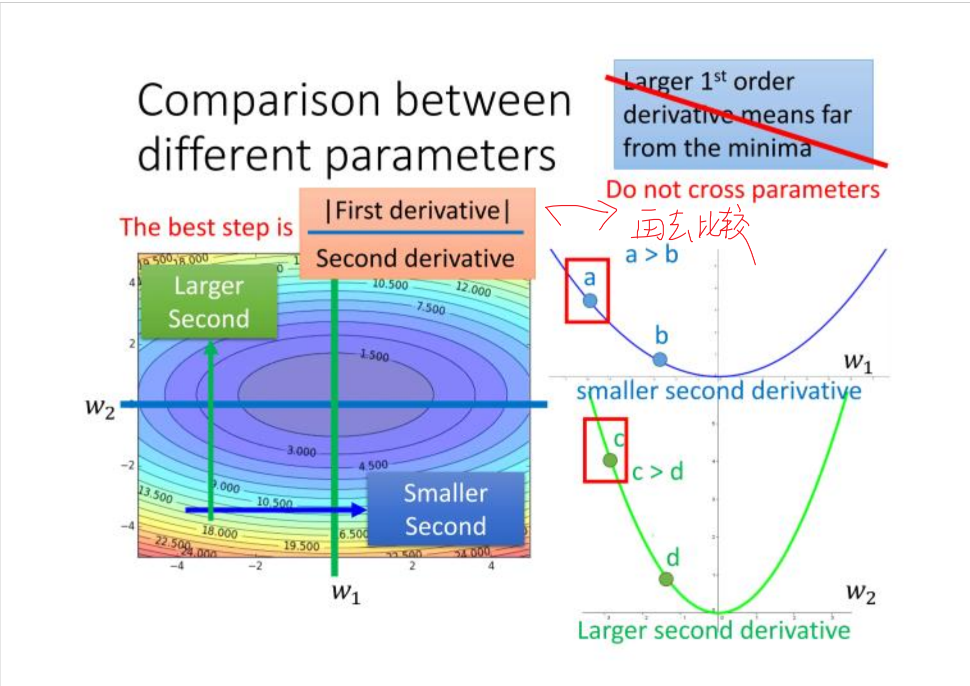
~~Larger 1st order derivative means far from the minima~~

Do not cross parameters

跨参数.




Second Derivative
Best step:
$|x_0 + \frac{b}{2a}|$
$\frac{|2ax_0 + b|}{\boxed{2a}}$
$-\frac{b}{2a}$
x_0
$y = ax^2 + bx + c$
$|2ax_0 + b|$
$\left|\frac{\partial y}{\partial x}\right| = |2ax + b|$
$\frac{\partial^2 y}{\partial x^2} = 2a$
The best step is
|First derivative|
Second derivative
二次微分
$\frac{|2ax_0+b|}{2a}$


Comparison between different parameters
~~Larger 1st order derivative means far from the minima~~
Do not cross parameters
The best step is
|First derivative|
Second derivative
Larger Second
Smaller Second
w_1
w_2
a > b
a
b
smaller second derivative
c
c > d
d
Larger second derivative

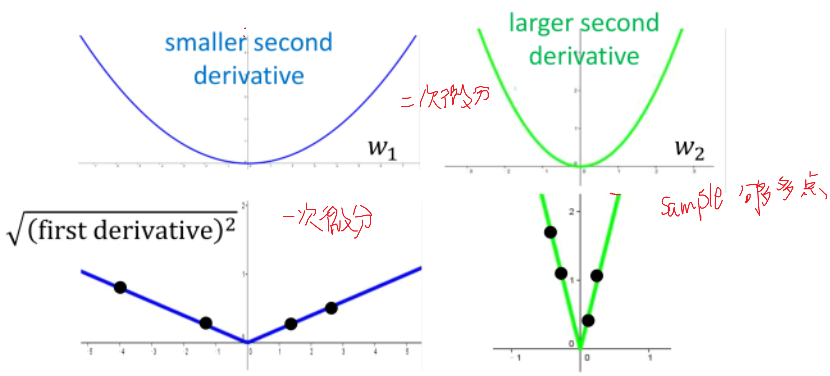
$$w^{t+1} \leftarrow w^t - \frac{\overset{\nearrow \text{常数.}}{\eta}}{\underbrace{\sqrt{\sum_{i=0}^{t}(g^i)^2}}_{\text{想要代表二次微分}}} g^t$$

?

The best step is

$$\frac{|\text{First derivative}|}{\text{Second derivative}}$$

Use *first derivative* to estimate *second derivative*



Gradient Descent

Tip 2: Stochastic Gradient Descent

Make the training faster

让 traing 更快

Stochastic Gradient Descent

Regression的 Loss

$$L = \sum_n \left(\hat{y}^n - \left(b + \sum w_i x_i^n \right) \right)^2$$

Loss is the summation over all training examples

◆ **Gradient Descent** $\theta^i = \theta^{i-1} - \eta \nabla L\left(\theta^{i-1}\right)$

◆ ***Stochastic Gradient Descent*** Faster!

Pick an example x^n

$$L^n = \left(\hat{y}^n - \left(b + \sum w_i x_i^n \right) \right)^2 \qquad \theta^i = \theta^{i-1} - \eta \nabla L^n\left(\theta^{i-1}\right)$$

Loss for only one example

考虑第n个example的Loss

每次拿一个example出来 { 随机取 / 顺序取 }

不用累加了

- Demo

Stochastic Gradient Descent

Gradient Descent

Update after seeing all examples

比较稳定

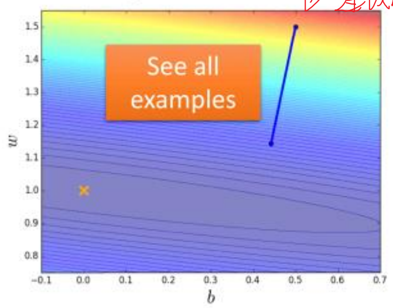
走的方向就是gradient建议的方向

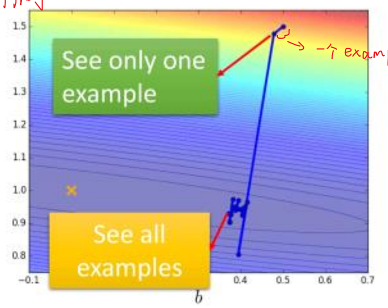
SGD

Stochastic Gradient Descent

Update for each example

If there are 20 examples, 20 times faster.





Gradient Descent

Tip 3: Feature Scaling

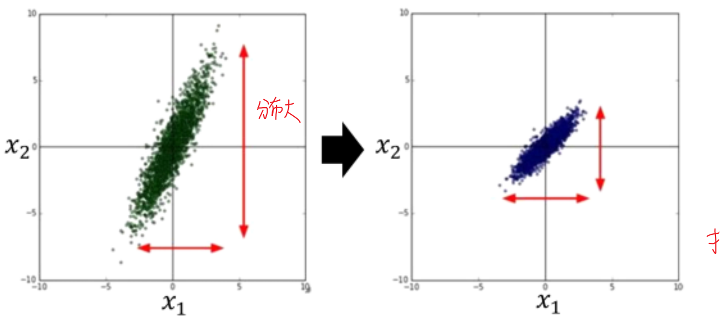
特征缩放

特征归一化

Feature Scaling

Source of figure: http://cs231n.github.io/neural-networks-2/

$$y = b + w_1x_1 + w_2x_2$$



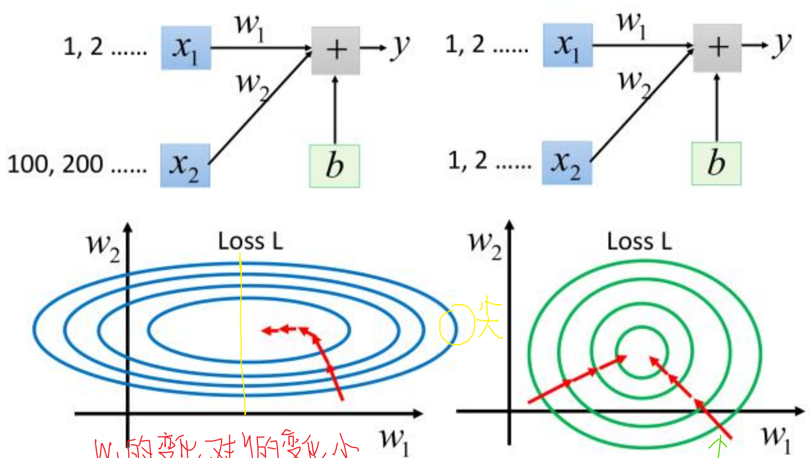
Make different features have the same scaling

把他们的range 改到一样

为什么要做

Feature Scaling 呢？

$$y = b + w_1x_1 + w_2x_2$$



w_1 w_2 对y有差不多的影响

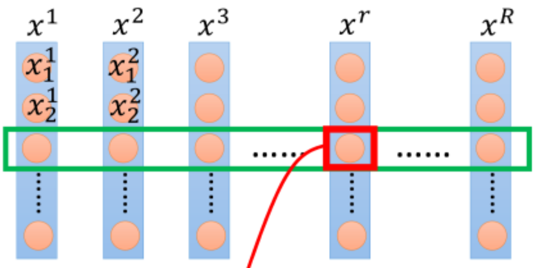
w_1的变化对y的变化小

w_2的变化对y的变化大

都是向着圆心走

Feature Scaling

R个example，每个example都有一组feature



For each dimension i:

mean: m_i

standard deviation: σ_i

$$x_i^r \leftarrow \frac{x_i^r - m_i}{\sigma_i}$$

The means of all dimensions are 0, and the variances are all 1

$\sim (0,1)$

Gradient Descent

Theory

理论基础

Question

- When solving:

$$\theta^* = \arg\min_{\theta} L(\theta) \quad \text{by gradient descent}$$

- Each time we update the parameters, we obtain θ that makes $L(\theta)$ smaller.

$$L(\theta^0) > L(\theta^1) > L(\theta^2) > \cdots$$

Is this statement correct?

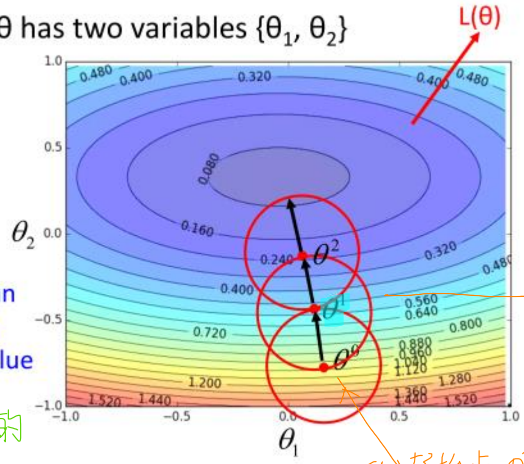
这个陈述是对的吗?　不对

更新后，Loss不一定下降

Warning of Math

Formal Derivation

- Suppose that θ has two variables $\{\theta_1, \theta_2\}$



Given a point, we can easily find the point with the smallest value nearby. How?

如何快速找到最小的那个点呢？

再画圆，再在圆圈内找最低点 θ^2.

给起始点 θ^0，在红圈内找一个最低点 θ^1

Taylor Series

泰勒展开 泰勒级数

- **Taylor series**: Let h(x) be any function infinitely differentiable around $x = x_0$.

把h(x)写成

$$h(x)=\sum_{k=0}^{\infty}\frac{h^{(k)}(x_0)}{k!}(x-x_0)^k$$

$$=h(x_0)+h'(x_0)(x-x_0)+\frac{h''(x_0)}{2!}(x-x_0)^2+\dots$$

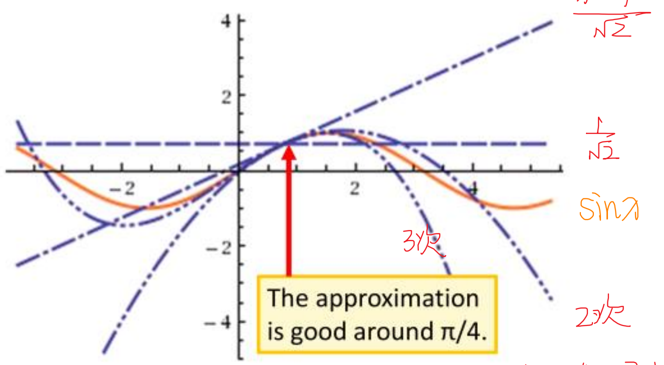
When x is close to x_0 ➡ $h(x)\approx h(x_0)+h'(x_0)(x-x_0)$

$x \to x_0$

例

E.g. Taylor series for h(x)=sin(x) around $x_0=\pi/4$

$$\sin(x)=\frac{1}{\sqrt{2}}+\frac{x-\frac{\pi}{4}}{\sqrt{2}}-\frac{\left(x-\frac{\pi}{4}\right)^2}{2\sqrt{2}}-\frac{\left(x-\frac{\pi}{4}\right)^3}{6\sqrt{2}}+\frac{\left(x-\frac{\pi}{4}\right)^4}{24\sqrt{2}}+\frac{\left(x-\frac{\pi}{4}\right)^5}{120\sqrt{2}}-\frac{\left(x-\frac{\pi}{4}\right)^6}{720\sqrt{2}}-\frac{\left(x-\frac{\pi}{4}\right)^7}{5040\sqrt{2}}+\frac{\left(x-\frac{\pi}{4}\right)^8}{40\,320\sqrt{2}}+\frac{\left(x-\frac{\pi}{4}\right)^9}{362\,880\sqrt{2}}-\frac{\left(x-\frac{\pi}{4}\right)^{10}}{3\,628\,800\sqrt{2}}+\ldots\ldots$$



$x \to \frac{\pi}{4}$时，后面高次项都很小

多元

Multivariable Taylor Series

$(x,y) \to (x_0,y_0)$

$$h(x,y)=\underline{h(x_0,y_0)}+\frac{\partial h(x_0,y_0)}{\partial x}\underline{(x-x_0)}+\frac{\partial h(x_0,y_0)}{\partial y}\underline{(y-y_0)}$$

+ something related to $(x-x_0)^2$ and $(y-y_0)^2$ + ...

When x and y is close to x_0 and y_0

$$h(x,y)\approx h(x_0,y_0)+\frac{\partial h(x_0,y_0)}{\partial x}(x-x_0)+\frac{\partial h(x_0,y_0)}{\partial y}(y-y_0)$$

Back to Formal Derivation

Based on Taylor Series:

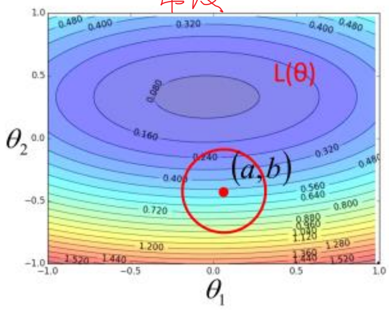
If the red circle is ***small enough***, in the red circle 红圈足够小

$$\mathrm{L}(\theta) \approx \underbrace{\mathrm{L}(a,b)}_{\text{常数}} + \underbrace{\frac{\partial \mathrm{L}(a,b)}{\partial \theta_1}}_{\text{常数}}(\theta_1 - a) + \underbrace{\frac{\partial \mathrm{L}(a,b)}{\partial \theta_2}}_{\text{常数}}(\theta_2 - b)$$ ← 红圈

$$s = \mathrm{L}(a,b)$$

$$u = \frac{\partial \mathrm{L}(a,b)}{\partial \theta_1}, v = \frac{\partial \mathrm{L}(a,b)}{\partial \theta_2}$$

$$\mathrm{L}(\theta) \approx s + u(\theta_1 - a) + v(\theta_2 - b)$$



Back to Formal Derivation

Based on Taylor Series:

If the red circle is ***small enough***, in the red circle

$$\mathrm{L}(\theta) \approx s + u(\theta_1 - a) + v(\theta_2 - b)$$

constant 常数

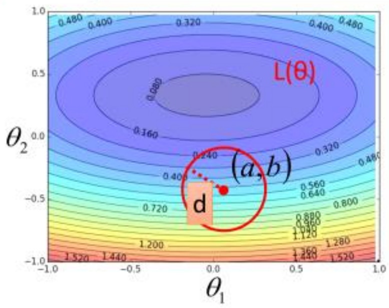
$$s = \mathrm{L}(a,b)$$

$$u = \frac{\partial \mathrm{L}(a,b)}{\partial \theta_1}, v = \frac{\partial \mathrm{L}(a,b)}{\partial \theta_2}$$

Find θ_1 and θ_2 in the red circle ***minimizing*** L(θ)

$$(\theta_1 - a)^2 + (\theta_2 - b)^2 \le d^2$$

Simple, right?

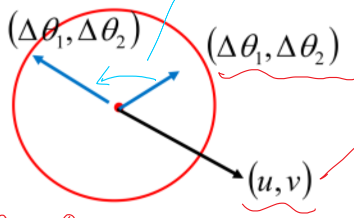


Gradient descent – two variables

Red Circle: (If the radius is small)

$$L(\theta) \approx \cancel{s} + u(\underbrace{\theta_1 - a}_{\Delta\theta_1}) + v(\underbrace{\theta_2 - b}_{\Delta\theta_2})$$

转到反向，做内积是 -1



u (Δθ1)
v (Δθ2)

Find θ_1 and θ_2 in the red circle *minimizing* L(θ)

$$(\underbrace{\theta_1 - a}_{\Delta\theta_1})^2 + (\underbrace{\theta_2 - b}_{\Delta\theta_2})^2 \le d^2$$

选什么样的Δθ1，Δθ2

To minimize L(θ)

$$\begin{bmatrix} \Delta\theta_1 \\ \Delta\theta_2 \end{bmatrix} = -\eta \begin{bmatrix} u \\ v \end{bmatrix} \Rightarrow \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix} - \eta \begin{bmatrix} u \\ v \end{bmatrix}$$

$$\begin{bmatrix} \theta_1 - a \\ \theta_2 - b \end{bmatrix} = -\eta \begin{bmatrix} u \\ v \end{bmatrix}$$

Back to Formal Derivation

Based on Taylor Series:

If the red circle is ***small enough***, in the red circle

$$L(\theta) \approx s + u(\theta_1 - a) + v(\theta_2 - b)$$

前题该式成立

constant

$$s = L(a,b)$$

$$u = \frac{\partial L(a,b)}{\partial \theta_1}, v = \frac{\partial L(a,b)}{\partial \theta_2}$$

Find θ_1 and θ_2 yielding the smallest value of $L(\theta)$ in the circle

$$\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix} - \eta \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix} - \eta \begin{bmatrix} \frac{\partial L(a,b)}{\partial \theta_1} \\ \frac{\partial L(a,b)}{\partial \theta_2} \end{bmatrix}$$

This is gradient descent.

代进去就是gradient descent

Not satisfied if the red circle (learning rate) is not small enough → learning rate 要设好 → 使 circle 足够小.
使上面的式子成立

You can consider the second order term, e.g. Newton's method.

二次式

牛顿法

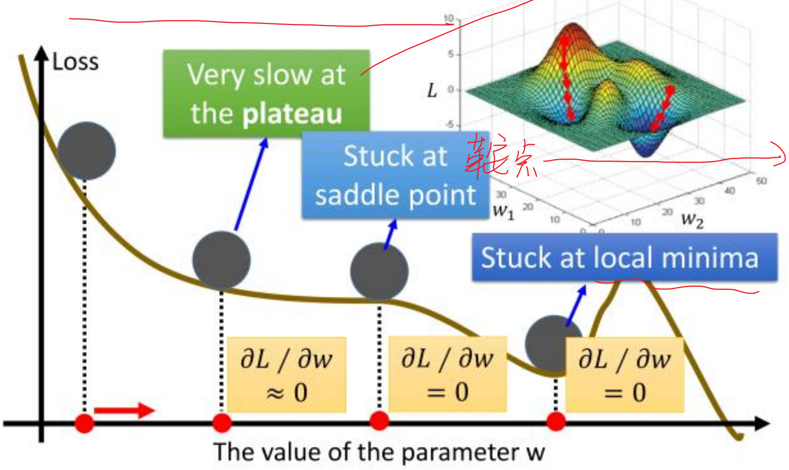
↳ 二次微分

End of Warning

More Limitation of Gradient Descent

的限制

微分值很小但是是高原



不是局部最小值,但是微分值为0

卡在局部最小值

Acknowledgement

- 感謝 Victor Chen 發現投影片上的打字錯誤