基本概念

2020年6月26日 14:11

什么是机器学习?

机器学习=寻找一种函数

Supervise learning 监督学习

Reinforcement learning强化学习

如何寻找这个函数?

①定一个函数集合

②判断函数的好坏

③选择最好的函数

机器学习套路

①设计模型model

②判断模型的好坏

③选择最好的函数，优化模型

3.1修改模型，增加数据维度

3.2增加正则因子，使函数更加平滑，让参数w取值更小。（x变化较小时，整个函数结果不会变化太大，结果更准）

学习路线

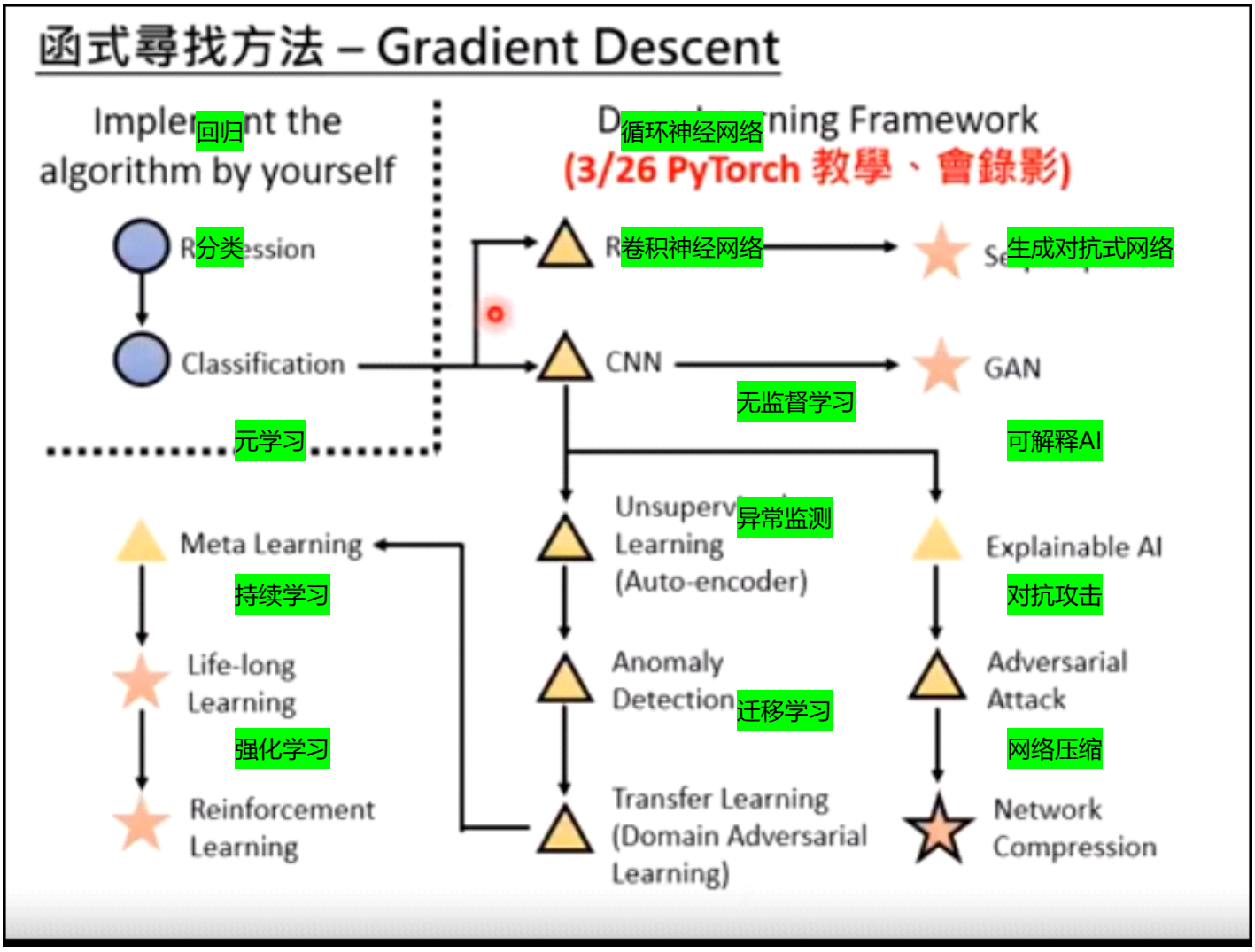
监督学习：有数据标注情况下学习（回归问题、分类问题）

半监督学习：训练数据中带标记的数据不够多

迁移学习：在已学习基础上，做看似和以前学习不相关的事情，但实际效果很好（在猫狗识别基础识别大象老虎等）

非监督学习：没有具体标注数据的情况下学习（机器阅读、机器绘画、聚类算法）梯度下降法

结构化学习：超越简单的回归和分类，产生结构化的结果（如图片、语言、声音）



回归：股票预测、自动驾驶、推荐系统

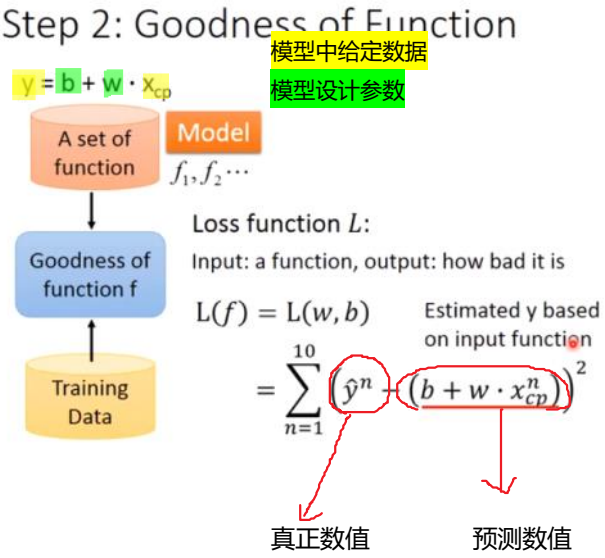
回归案例研究 Regression Case Study

2020年7月5日　21:40

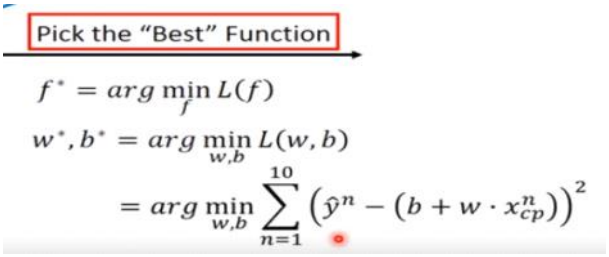
回归：股票预测、自动驾驶、推荐系统

模型设计 -->模型评估（右图）-->选出最优模型

设计评估函数，对线性方程的参数进行带入计算，根据已知数据对模型预测结果与真值之间的误差进行量化，评估出最优模型

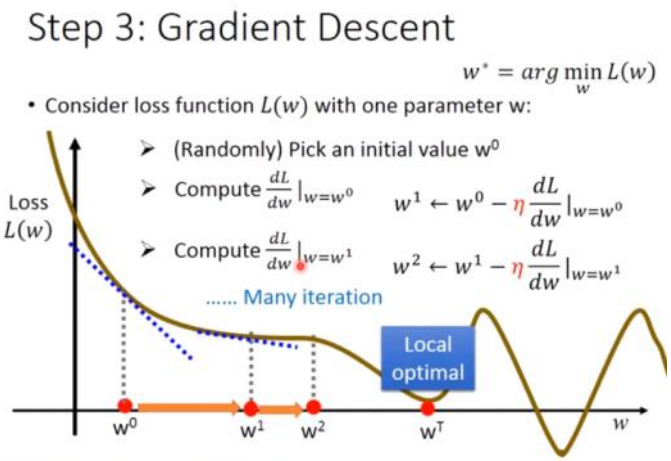


选出最优模型 即确定一组b和w使得误差监测函数的值最小



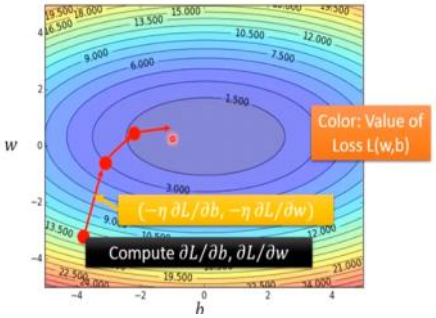
Gradient descent 梯度下降法

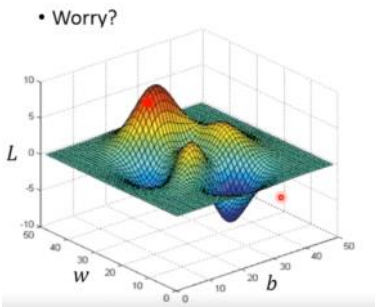
随机选取一个初始点，对此处w进行微分，根据微分值确定下一步要移动的方向和步长，在本例中因为要求最小点，所以微分值大于0，则减少w的值，反之增加



两个参数和一个参数类似，只是去求偏微分

偏微分思路： 右图





本例中偏微分计算：右图

$$L(w,b)=\sum_{n=1}^{10}\left(\hat{y}^n-\left(b+w\cdot x_{cp}^n\right)\right)^2$$

$$\frac{\partial L}{\partial w}=?\sum_{n=1}^{10}2\left(\hat{y}^n-\left(b+w\cdot x_{cp}^n\right)\right)\left(-x_{cp}^n\right)$$

$$\frac{\partial L}{\partial b}=?\sum_{n=1}^{10}2\left(\hat{y}^n-\left(b+w\cdot x_{cp}^n\right)\right)(-1)$$

可以通过增加高次项使得模型与训练值的匹配程度越高，但是
并不能代表与预测结果越接近，这种情况叫做“过适”overfitting
所以要根据训练数据和测试数据，选择误差相对最小的模式

右图为对模型增加选择条件后的计算

可以使不同种类选择不同的模型，只需将其它种类置零即可

Back to step 1:
Redesign the Model

$y=b+\sum w_i x_i$

Linear model?

$y=b_1\cdot\delta(x_s=\text{Pidgey})$
$+w_1\cdot\delta(x_s=\text{Pidgey})x_{cp}$
$+b_2\cdot\delta(x_s=\text{Weedle})$
$+w_2\cdot\delta(x_s=\text{Weedle})x_{cp}$
$+b_3\cdot\delta(x_s=\text{Caterpie})$
$+w_3\cdot\delta(x_s=\text{Caterpie})x_{cp}$
$+b_4\cdot\delta(x_s=\text{Eevee})$
$+w_4\cdot\delta(x_s=\text{Eevee})x_{cp}$

$\delta(x_s=\text{Pidgey})$

$\begin{cases}=1 & \text{If } x_s=\text{Pidgey}\\ =0 & \text{otherwise}\end{cases}$

If $x_s=$ Pidgey

回归

$y=b+\sum w_i x_i$

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

The functions with smaller w_i are better

$+\lambda\sum(w_i)^2$

- Why smooth functions are preferred?

$y=b+\sum w_i x_i$

$+w_i\Delta x_i$ $+\Delta x_i$

- If some noises corrupt input x_i when testing

A smoother function has less influence.

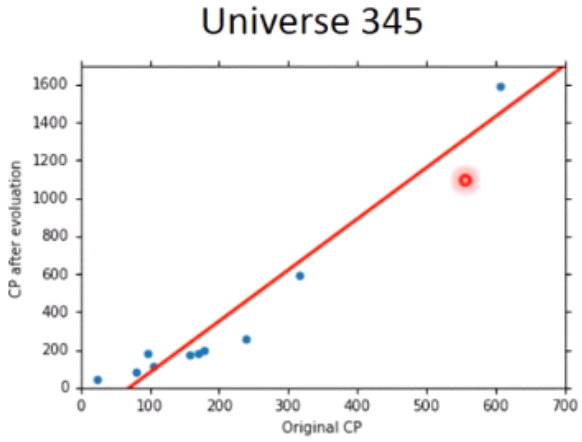
错误的来源

2020年7月6日　19:41

简单的model结果分布比较集中，但是离最优情况或许有一定差异
复杂的model离最优情况较为接近，但是分布比较发散

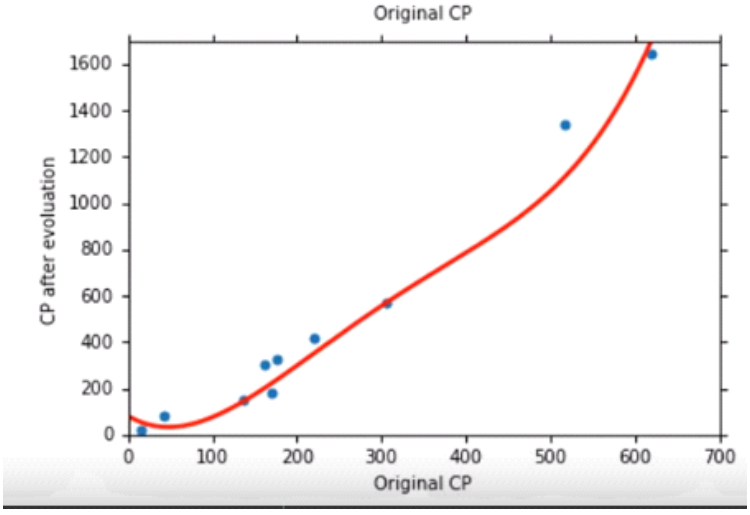
误差 = 偏差 + 方差

首先三者之间的联系是 Error = Bias + Variance （这里应该是忽略的噪音）。Error反映的是整个模型的准确度，说白了就是你给出的模型，input一个变量，和理想的output之间吻合程度，吻合度高就是Error低。Bias反映的是模型在样本上的输出与真实值之间的误差，即模型本身的精准度，其实Bias在股票上也有应用，也可以反映股价在波动过程中与移动平均线偏离程度。其实通过这个我感觉可以更容易的理解这个概念，我们知道Bias是受算法模型的复杂度决定的，假设下图的红线是我们给出的模型，蓝色的点就是样本，这是一个最简单的线性模型，这个时候Bias就可以通过这些蓝色的点到红线沿Y轴的垂直距离来反映（即真实值与模型输出的误差），距离越大说明Bias越大，也说明拟合度更低。



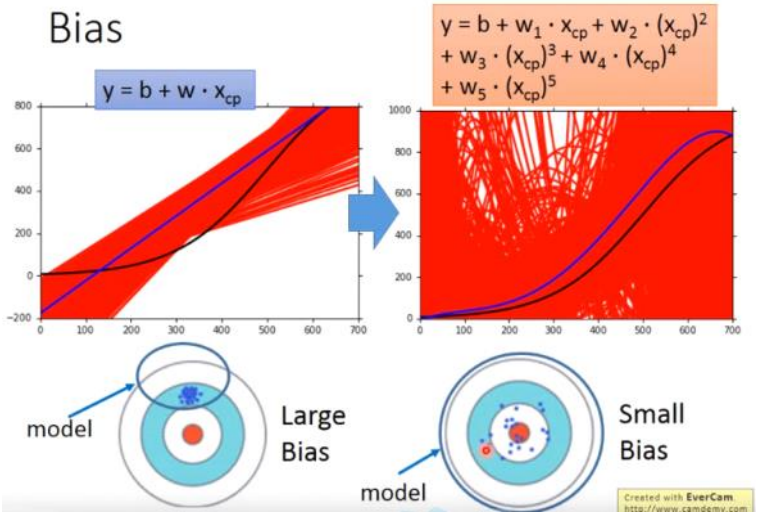
$$y = b + w \cdot x_{cp}$$

当我们增加模型的复杂度，刚刚是一个线性的模型，现在是一个四次方的模型，可以明显看出点到模型的沿Y轴的垂直距离更少了，即拟合度更高了，所以Bias也更低了。所以这样我们就可以很容易理解Bias和模型复杂度之间的关系了。给出结论：当模型复杂度上升时，Bias减小。当模型复杂度降低时，Bias增加。这里就涉及到了欠拟合(unfitting)和过度拟合(overFitting)的问题了。



Variance（方差）反映的是模型每一次输出结果与模型输出期望之间的误差，即模型的稳定性。在概率论和统计学中方差是衡量随机变量或一组数据时离散程度的度量。下图中红线就是每一组样本对应的模型，想象一下真实数据有无限多，我们以10个样本为一组，选取了500个样本组，然后在线性模型下，针对这500个样本组，我们会有500组不同的b和w值组成

的线性模型，最后构成左图的样子。当我们的模型升级成5次方的复杂程度时，针对这500个样本组，我们会有右边这张图显示的500组不同的参数构成的模型。可以看出，明显右边的图比左边的图更离散一些，试想一个极端情况，当模型就是一个常数时，这个时候模型复杂度最低，同时Variance也为0。所以我们可以得出结论：当模型复杂度低时，Variance更低，当模型复杂度高时，Variance更高。

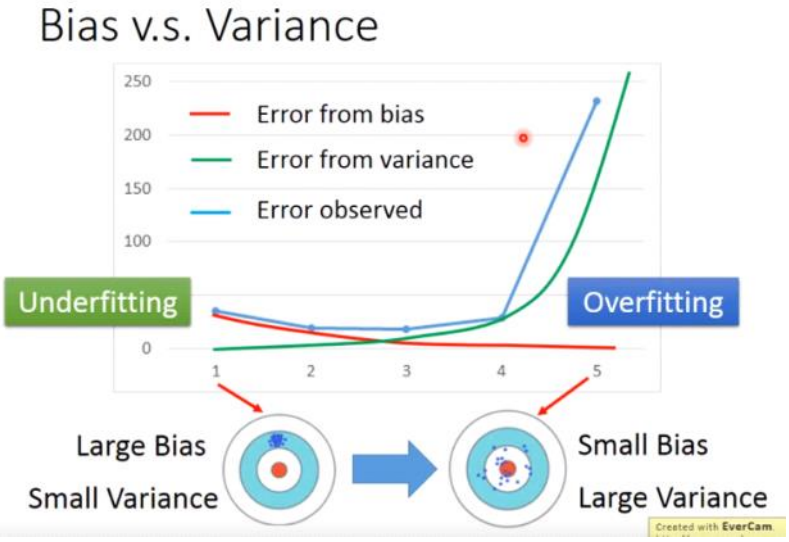


到这里我们可以给出两个结论。

一、Bias和模型复杂度的关系：当模型复杂度上升时，Bias减小。当模型复杂度降低时，Bias增加。（反比关系）

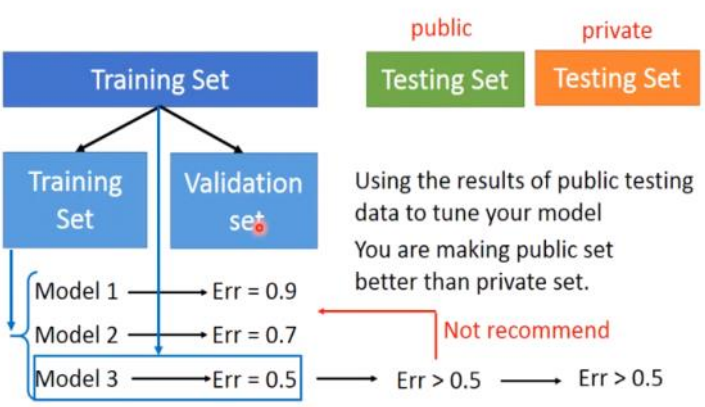
二、Variance和模型复杂度的关系：当模型复杂度低时，Variance更低，当模型复杂度高时，Variance更高。（正比关系）

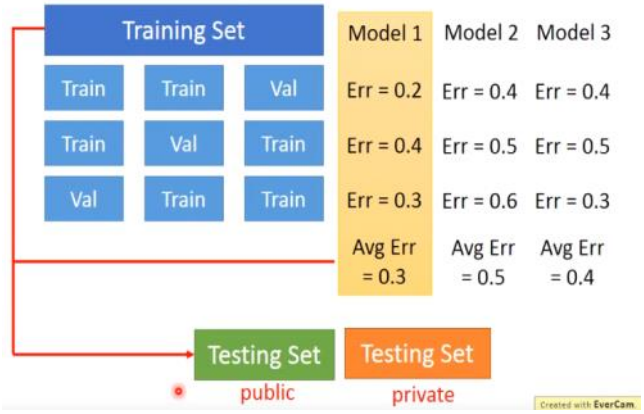
一开始我们就知道Error = Bias + Variance。整个模型的准确度和这两个都有关系，所以这下看似是有些矛盾的。如何才能取到最小的Error呢，看下图，蓝线就是Error的伴随Bias和Variance的变化情况，可以看出横坐标3应该是一个较好的结果，所以我们需要找到一个平衡点取得最优解



实际情景中我们怎么判断自己的模型是Bias大还是Variance大呢，这个就要看到底是你的模型无法尽量大的拟合你的样本还是你的模型高度拟合你的样本但是用测试数据算时误差右很大。前者就是应该bias大导致的，也就是模型复杂度太低导致的。后者就是因为模型复杂度高导致Variance高导致的。
简单模型，variance小。复杂模型，variance大。简单模型，bias大。复杂模型，bias小。

训练数据集的使用：分成训练集和验证集，通过训练集获取模型，使用验证机进行验证


Training Set
Training Set
Validation set
Model 1
Model 2
Model 3
Err = 0.9
Err = 0.7
Err = 0.5
public
private
Testing Set
Testing Set
Using the results of public testing data to tune your model
You are making public set better than private set.
Not recommend
Err > 0.5
Err > 0.5


Training Set
Train
Train
Val
Train
Val
Train
Val
Train
Train
Model 1
Model 2
Model 3
Err = 0.2
Err = 0.4
Err = 0.4
Err = 0.4
Err = 0.5
Err = 0.5
Err = 0.3
Err = 0.6
Err = 0.3
Avg Err = 0.3
Avg Err = 0.5
Avg Err = 0.4
Testing Set
Testing Set
public
private
Created with EverCam

梯度下降 Gradient Descent

2020年7月8日 9:23

解决最优化问题 optimization

Gradient 梯度：梯度的定义

设二元函数 $z=f(x,y)$ 在平面区域D上具有一阶连续偏导数，则对于每一个点P（x，y）都可定出一个向量 $\left\{\frac{\partial f}{\partial x},\frac{\partial f}{\partial y}\right\}=f_x(x,y)\bar{i}+f_y(x,y)\bar{j}$，该函数就称为函数 $z=f(x,y)$ 在点P（x，y）的梯度，记作gradf（x，y）或 $\nabla f(x,y)$，即有：

gradf（x，y）$=\nabla f(x,y)=\left\{\frac{\partial f}{\partial x},\frac{\partial f}{\partial y}\right\}=f_x(x,y)\bar{i}+f_y(x,y)\bar{j}$

其中 $\nabla=\frac{\partial}{\partial x}\bar{i}+\frac{\partial}{\partial y}\bar{j}$ 称为（二维的）向量微分算子或Nabla算子，$\nabla f=\frac{\partial f}{\partial x}\bar{i}+\frac{\partial f}{\partial y}\bar{j}$。

设 $e=\{cos\alpha,\cos\beta\}$ 是方向l上的单位向量，则

$$\frac{\partial f}{\partial l}=\frac{\partial f}{\partial x}\cos\alpha+\frac{\partial f}{\partial y}\cos\beta=\left\{\frac{\partial f}{\partial x},\frac{\partial f}{\partial y}\right\}\{\cos\alpha,\cos\beta\}$$

$$=gradf(x,y)\,e=|gradf(x,y)|\,|e|\cos[gradf(x,y),e]$$

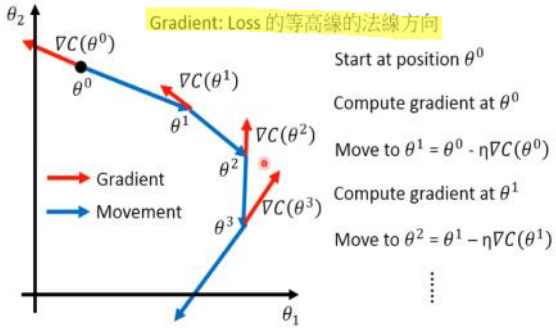
由于当方向l与梯度方向一致时，有

$$\cos[gradf(x,y),e]=1$$

所以当l与梯度方向一致时，方向导数 $\frac{\partial f}{\partial l}$ 有最大值，且最大值为梯度的模，即

$$|gradf(x,y)|=\sqrt{\left(\frac{\partial f}{\partial x}\right)^2+\left(\frac{\partial f}{\partial y}\right)^2}$$

因此说，函数在一点沿梯度方向的变化率最大，最大值为该梯度的模。[1]

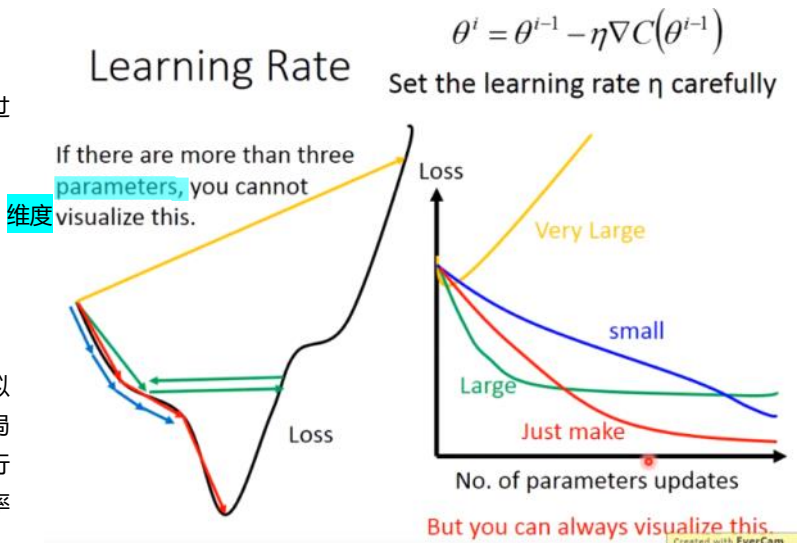


Learning rate 学习率 能不能理解成拟合的步长

首先我们简单回顾下什么是学习率，在梯度下降的过程中更新权重时的超参数，即下面公式中的α

$$\theta=\theta-\alpha\frac{\partial}{\partial\theta}J(\theta)$$

学习率越低，损失函数的变化速度就越慢，容易过拟合。虽然使用低学习率可以确保我们不会错过任何局部极小值，但也意味着我们将花费更长的时间来进行收敛，特别是在被困在局部最优点的时候。而学习率过高容易发生梯度爆炸，loss振动幅度较大，模型难以收敛。下图是不同学习率的loss变化，因此，选择一个合适的学习率是十分重要的。



随机梯度下降 Stochastic Gradient Descent