线性分类器（2）

叶山 中国地质大学（北京）

yes@cugb.edu.cn

上期回顾

线性分类器：代数视角

$$
\begin{array}{c}
\text{权值 } \omega_i^T \\
\begin{array}{cc}
\text{狗 } \omega_1^T \\
\text{猫 } \omega_2^T \\
\text{鸟 } \omega_3^T
\end{array}
\begin{bmatrix}
0.2 & -0.5 & 0.1 & 2 \\
1.5 & 1.3 & 2.1 & 0 \\
0 & 0.25 & 0.2 & -0.3
\end{bmatrix}
\end{array}
\cdot
\begin{array}{c}
\text{图像表示} x \\
\begin{bmatrix} 56 \\ 231 \\ 24 \\ 2 \end{bmatrix}
\end{array}
+
\begin{array}{c}
\text{偏置} b_i \\
\begin{array}{c} b_1 \\ b_2 \\ b_3 \end{array}
\begin{bmatrix} 1.1 \\ 3.2 \\ -1.2 \end{bmatrix}
\end{array}
=
\begin{array}{c}
\text{得分} f_i \\
\begin{array}{c} f_1 \\ f_2 \\ f_3 \end{array}
\begin{bmatrix} -98 \\ 435 \\ 63 \end{bmatrix}
\end{array}
$$

这张图被划分为

猫

决策步骤：

1. 把图像展开为向量
2. 计算每个类别的分数
3. 查看得分最高的类别

$$f_i(x, \omega_i) = \omega_i^T x + b_i$$

$$i = 1, 2, 3$$

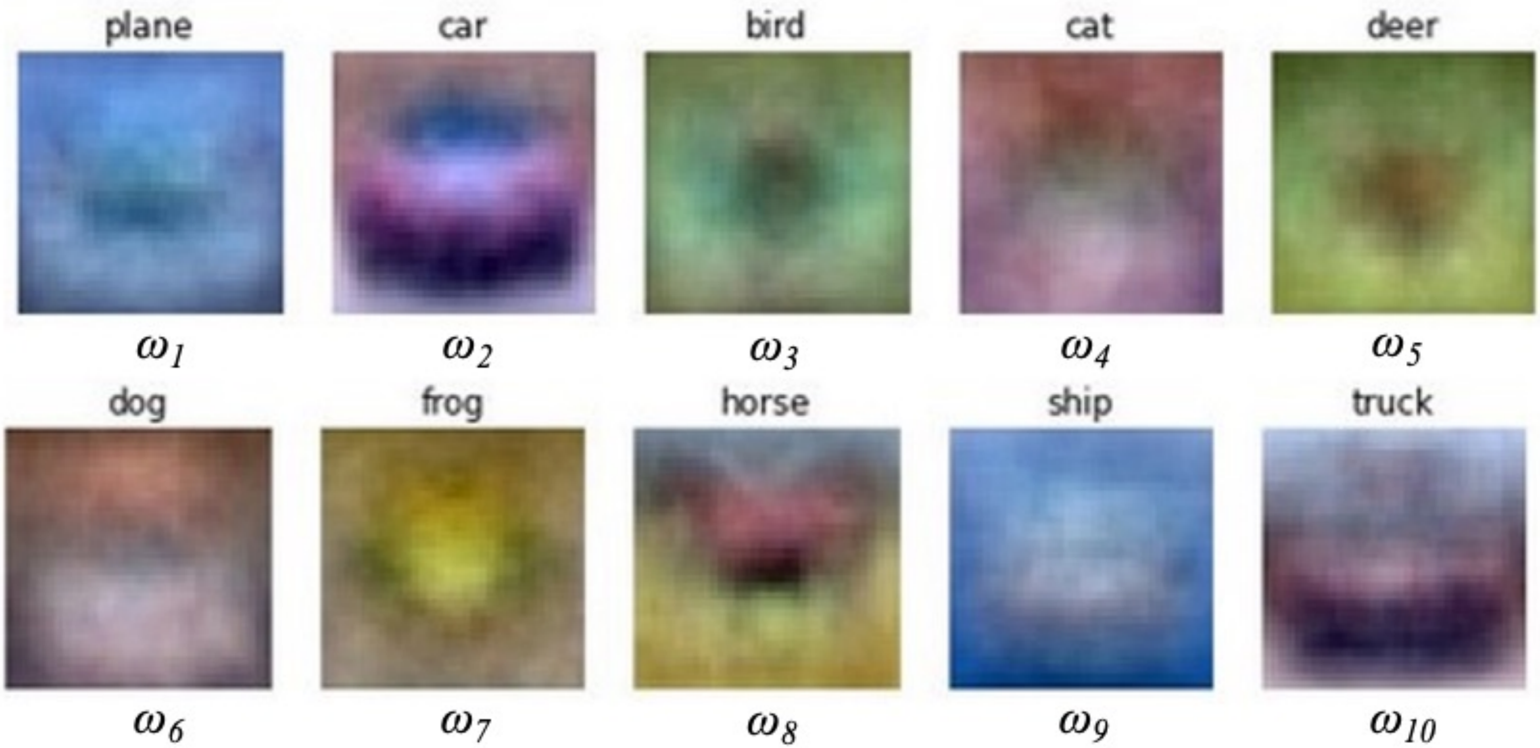
线性分类器：可视化视角

$f_i(x, \omega_i) = \omega_i^T x + b_i$

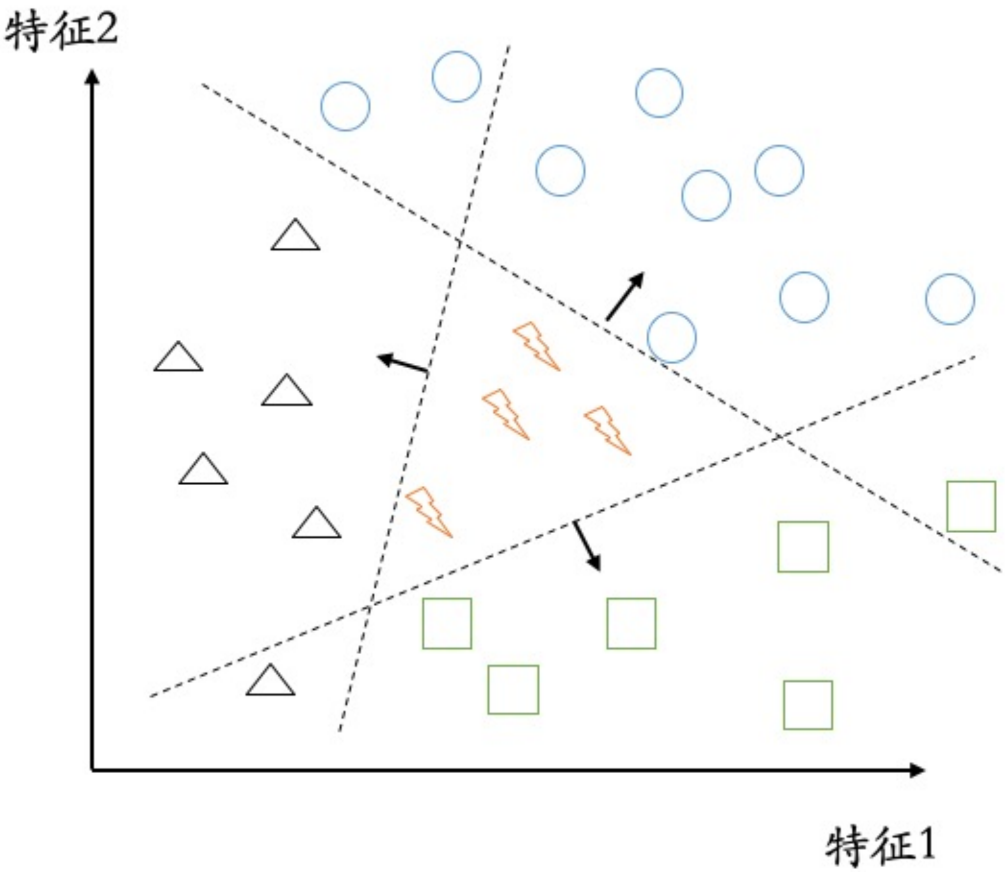
$i = 1, 2, \ldots, c$

$\omega_i = [\omega_{i1}, \omega_{i2}, \ldots, \omega_{id}]^T$为第$i$个类别的权值向量，$b_i$为第$i$个类别的偏置。

权值向量是记录该类图片宏观信息的模板。输入图像与模板的匹配度越高，点乘后得分越高。



线性分类器：几何视角



分类器的核心任务:
寻找决策边界（超平面）
$\omega_i^T x + b_i = 0$

ω：控制正方向（类比斜率）
b：控制位移（类比截距）

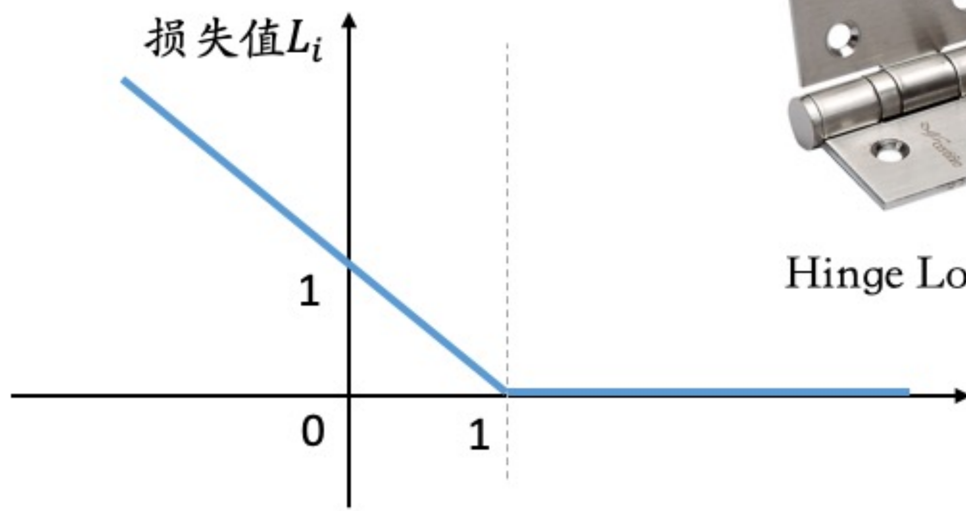
箭头：分数增加最快的方向，即正方向，沿此方向，距离决策面越远，分数就越高。

损失函数

$$L = \frac{1}{N}\sum_{i=1}^{N} L_i(f(x_i, W), y_i)$$

L：总损失值
L_i：当前样本的损失值
$f(x_i, W)$：当前样本的类别分数
y_i：当前样本的真实标签（整数）
W：权值（包含了偏置）
N：样本的个数

多类支持向量机损失



Hinge Loss 合页损失

第i个样本的多类支持向量机损失:

$$L_i = \sum_{j \neq y_i} \begin{cases} 0 & \text{如果 } s_{y_i} \geq s_{ij} + 1 \\ s_{ij} - s_{y_i} + 1 & \text{其他情况} \end{cases}$$

$$= \sum_{j \neq y_i} \max(0, s_{ij} - s_{y_i} + 1)$$

s_{y_i}: 第i个样本在真实类别里的预测分数

多类支持向量机损失

多类支撑向量机损失的最大和最小值是多少?

0到正无穷

如果s_{ij}是随机生成的较小值（模型未训练），则总损失L和类别数c的关系是什么?

$L \approx c - 1$

如果在计算单样本损失时，删除$j \neq y_i$的要求，结果会有明显变化吗?

变化不大

计算总损失L时，如果采用求和而非求平均，结果会有明显变化吗?

变化不大

若使用$L_i = \sum_{j \neq y_i} \max(0,\ s_{ij} - s_{y_i} + 1)^2$ 结果会有明显变化吗?

变化明显

线性分类器

$$s_{ij} = f_j(x_i, w_j, b_j) = w_j^T x_i + b_j$$

单样本的损失

$$L_i = \sum_{j \neq y_i} \max(0,\ s_{ij} - s_{y_i} + 1)$$

总损失

$$L = \frac{1}{N} \sum_{i=1}^{N} L_i(f(x_i, W),\ y_i)$$

其他常用损失函数

Regression Losses

Mean absolute error $\frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$

Mean square error $\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$

Gaussian negative log likelihood loss $\frac{1}{2N}\sum_{i=1}^{N}\left(\log(\max(\hat{\sigma},\epsilon) + \frac{(y_i - \hat{y}_i)^2}{\max(\hat{\sigma},\epsilon)})\right)$

Huber loss $\frac{1}{N}\sum_{i=1}^{N} l_i \text{ with } \begin{cases} l_i = \frac{1}{2}(y_i - \hat{y}_i)^2 & \text{if } |y_i - \hat{y}_i| < \delta \\ l_i = \delta(|y_i - \hat{y}_i| - \frac{1}{2}\delta) & \text{otherwise} \end{cases}$

Smooth L1 loss $\frac{1}{N}\sum_{i=1}^{N} l_i \text{ with } \begin{cases} l_i = \frac{1}{2\delta}(y_i - \hat{y}_i)^2 & \text{if } |y_i - \hat{y}_i| < \delta \\ l_i = \delta(|y_i - \hat{y}_i| - \frac{1}{2}\delta) & \text{otherwise} \end{cases}$

Log cosh $\frac{1}{N}\sum_{i=1}^{N}\log(\cosh(y_i - \hat{y}_i))$

Mean absolute percentage error $\frac{100}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$

Mean squared logarithmic error $\frac{1}{N}\sum_{i=1}^{N}(\log(y_i + 1) - \log(\hat{y}_i + 1))^2$

Poisson loss $\frac{1}{N}\sum_{i=1}^{N}\hat{y}_i - y_i\log(\hat{y}_i)$

Classification Losses

Cross-entropy $-\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{C} y_{ik}\log(\hat{y}_{ik})$

Squared Hinge $\max(0, 1 - y_i\hat{y}_i)^2$

Kullback-Leibler divergence loss $\frac{1}{N}\sum_{i=1}\sum_{k=1}^{C} y_{ik}\log\frac{y_{ik}}{\hat{y}_{ik}}$

Soft margin $\frac{1}{N}\sum_{i=1}^{N}\log\left(1 + \exp(-y_i\hat{y}_i)\right)$

Focal cross-entropy $-\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{C}(1 - \hat{y}_{ik})^{\gamma} y_{ik}\log(\hat{y}_{ik})$

Ranking Losses

Margin ranking loss $\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\max(0, (1 - 2\mathbb{I}_{y_1 > y_2})(\hat{y}_i - \hat{y}_j) + \delta)$

Soft pairwise Hinge $\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\log\left(1 + \exp((1 - 2\mathbb{I}_{y_1 > y_2})(\hat{y}_i - \hat{y}_j) + \delta)\right)$

Pairwise logistic $\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\log\left(1 + \exp((1 - 2\mathbb{I}_{y_1 > y_2})(\hat{y}_i - \hat{y}_j))\right)$

Contrastive Losses

Cosine embedding loss $\frac{1}{N}\sum_{i=1} l_i \text{ with } l_i = \begin{cases} 1 - \frac{\sum_{k=1}^{C} x_{1k}x_{2k}}{\sqrt{\sum_{k=1}^{C} x_{1k}^2}\sqrt{\sum_{k=1}^{C} x_{2k}^2}} & \text{if } y = 1 \\ \max\left(0, \frac{\sum_{k=1}^{C} x_{1k}x_{2k}}{\sqrt{\sum_{k=1}^{C} x_{1k}^2}\sqrt{\sum_{k=1}^{C} x_{2k}^2}} - \alpha\right) & \text{if } y = -1 \end{cases}$

Hinge embedding loss $\frac{1}{N}\sum_{i=1} l_i \text{ with } l_i = \begin{cases} \sum_{k-1}^{C}(x_{1k} - x_{2k})^2 & \text{if } y = 1 \\ \max\left(0, \delta - \sum_{k-1}^{C}(x_{1k} - x_{2k})^2\right) & \text{if } y = -1 \end{cases}$

Triplet margin loss $\frac{1}{N}\sum_{i=1}^{N}\max\left(\sum_{k=1}(a_{ik} - x_{1ik})^2 - (a_{ik} - x_{2ik})^2 + \delta, 0\right)$

Reinforcement Learning Losses

Q-value $\frac{1}{N}\sum_{i=1}^{N}\left(Q_i(s,a) - (r_i + \gamma\max_{a' \in A} Q_i(s', a'))\right)^2$

Policy gradient $-\frac{1}{N}\sum_{i=1}^{N} Q_i(s,a)\log(\pi_i(a|s))$

正则项

权值矩阵的问题

在用某个分类器进行某项任务时，如果我们发现存在权值矩阵W，使得总损失L为0，请问： W是唯一的吗?

不是，比如$2W$的总损失L也为0。

线性分类器

$$s_{ij} = f_j(x_i, w_j, b_j) = w_j^T x_i + b_j$$

单样本的损失

$$L_i = \sum_{j \neq y_i} \max(0, s_{ij} - s_{y_i} + 1)$$

总损失

$$L = \frac{1}{N}\sum_{i=1}^{N} L_i(f(x_i, W), y_i)$$

权值矩阵的问题

$$s_{ij} = f_j(x_i, w_j, b_j) = w_j^T x_i + b_j$$

$$L_i = \sum_{j \neq y_i} \max(0, s_{ij} - s_{y_i} + 1)$$

cat	**3.2**	1.3	2.6	2.2
car	5.1	**4.9**	9.8	2.5
frog	-1.7	2.0	4.0	**-3.1**
		W_1	W_2	

$$L_{i,W1} = \max(0, 1.3 - 4.9 + 1) + \max(0, 2.0 - 4.9 + 1) = 0$$

$$L_{i,W2} = \max(0, 2.6 - 9.8 + 1) + \max(0, 4.0 - 9.8 + 1) = 0$$

W_1和W_2都能满足让损失为0。

问题：W_1和W_2之间该如何选择?

损失函数关注分类器模型在训练集上的效果，
但我们应该更关心模型在实战中的效果。

正则项

数据损失（data loss）　　正则损失（regularization）

$$L = \frac{1}{N}\sum_{i=1}^{N} L_i(f(x_i, W), y_i) + \lambda R(W)$$

分类器模型预测结果应和训练数据集相符

避免模型在训练数据集上的表现过于优秀

正则项又称惩罚因子，被用来抑制模型在训练集上的性能。

正则项$R(W)$函数的输出值只和W有关，和图像x无关。在添加常见的正则项以后，当总损失一定时（如$L = 0$），W通常有唯一解。

超参数λ控制正则损失$R(W)$占总损失L的占比。

模型参数和超参数

模型参数 model parameter，通常简称“参数”

- 模型在训练过程中，根据训练数据集习得的参数
- 可人为设置初始化值，但模型学习时会自动对其进行调整
- 举例：权值w，偏置b等（优化算法的优化对象）

超参数 hyperparameter

- 训练开始前，由模型的设计者手动设置的参数
- 模型运行时必须遵守超参数，没有改变超参数的权限
- 举例：正则化占比λ、神经元个数、学习率、训练批次等

模型参数和超参数

数据损失（data loss）　　正则损失（regularization）

$$L = \frac{1}{N}\sum_{i=1}^{N} L_i(f(x_i, W), y_i) + \lambda R(W)$$

超参数λ对L的影响

如果$\lambda = 0$?
只考虑数据损失，则W存在多解性。

如果$\lambda \to +\infty$?
数据损失难以影响优化结果；分类器在训练集上的学习情况被忽视。

模型参数和超参数

不同的超参数选择可能会导致完全不同的模型行为和性能。

超参数调优（调参）是调整机器学习模型以获得最佳性能的重要步骤之一。常见的超参数优化方法包括网格搜索、随机搜索、贝叶斯优化等。

合理的超参数设置可以使模型更好地学习和适应数据，从而提高模型的性能。调参的主要目标是通过最小化损失函数，找到最佳的超参数组合，以提升模型的预测能力和泛化性能。

“炼丹”

L1和L2正则项

$$L = \frac{1}{N}\sum_{i=1}^{N} L_i(f(x_i, W), y_i) + \lambda R(W)$$

$$R(W) = \sum_k \sum_l |W_{k,l}|$$ L1正则项

$$R(W) = \sum_k \sum_l W_{k,l}^2$$ L2正则项

$$R(W) = \sum_k \sum_l \beta W_{k,l}^2 + |W_{k,l}|$$ Elastic Net

k, l：权值矩阵W的维度，相当于线性分类器任务中的类别数c，以及输入图像维度d。

L1和L2正则项

$$L = \frac{1}{N}\sum_{i=1}^{N} L_i(f(x_i, W), y_i) + \lambda R(W)$$

$$L_i = \sum_{j \neq y_i} \max(0, s_{ij} - s_{y_i} + 1)$$

$$R(W) = \sum_k \sum_l W_{k,l}^2 \quad \text{L2正则项}$$

作用：抑制较大权值的作用，从而促使权值的分散，鼓励模型综合使用所有的特征，而非过度依赖少数权值较大的特征，从而让模型更稳健，减少噪声的影响。

鼓励分类器2的权值分布（权值平均分布，分类时考虑所有特征维度），不鼓励分类器1的权值分布（权值集中，分类时只参考了少量特征维度）。

<u>L2正则项计算举例</u>

样本数据 $x = [1,1,1,1]$

分类器1权值 $W_1 = [1,0,0,0]$

分类器2权值 $W_2 = [0.25,0.25,0.25,0.25]$

分类器的输出 $W_1^T x = W_2^T x = 1$

忽略偏置，假设$\lambda = 1$

两个分类器数值损失相同

分类器1的正则损失:
$R(W_1) = 1^2 + 0^2 + 0^2 + 0^2 = 1$

分类器2的正则损失:
$R(W_2) = 0.25^2 + 0.25^2 + 0.25^2 + 0.25^2 = 0.25$

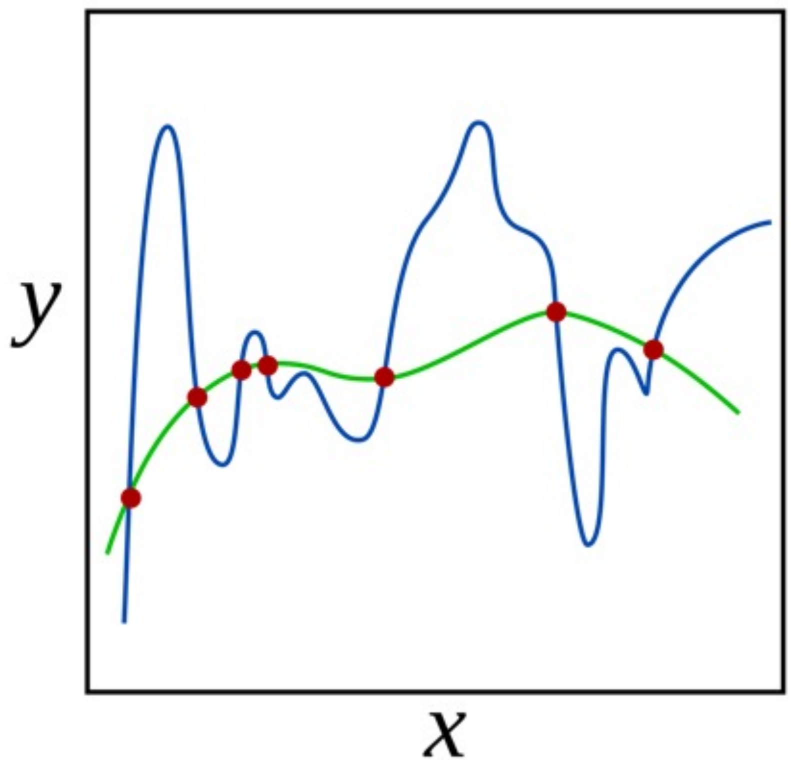
分类器2的总损失小

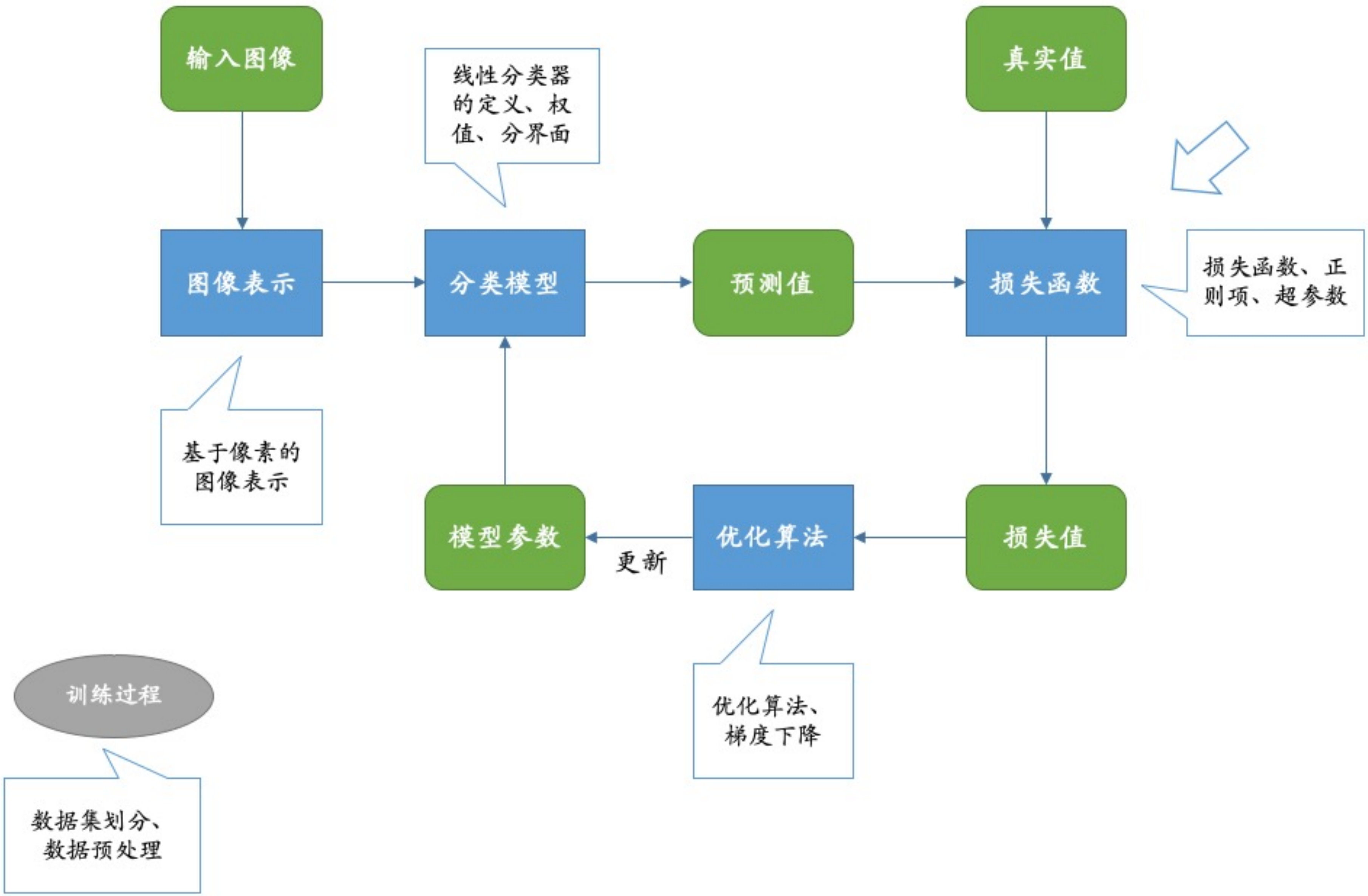
正则项的意义

引导分类器模型“兼听则明”，综合考虑所有特征维度的信息，防止出现“一言堂”现象（即部分特征维度的“话语权”过高）。

正则项的意义

抑制过拟合现象：不要“死记硬背”训练数据集中的局部规律，鼓励模型去学习宏观规律。




输入图像
线性分类器的定义、权值、分界面
真实值
图像表示
分类模型
预测值
损失函数
损失函数、正则项、超参数
基于像素的图像表示
模型参数
更新
优化算法
损失值
训练过程
优化算法、梯度下降
数据集划分、数据预处理

优化算法

参数优化

利用损失函数的输出值（损失值）作为反馈信号来调整分类器的模型参数，从而提升分类器对训练样本的预测性能。

目标：通过调整模型参数，让损失值尽量变小。

数学意义：机器学习中，需要用一个数学模型来解释训练数据集的规律。该数学模型过于复杂，难以直接求得最优解，但可以通过多次迭代，逐渐靠近最优解。

参数优化

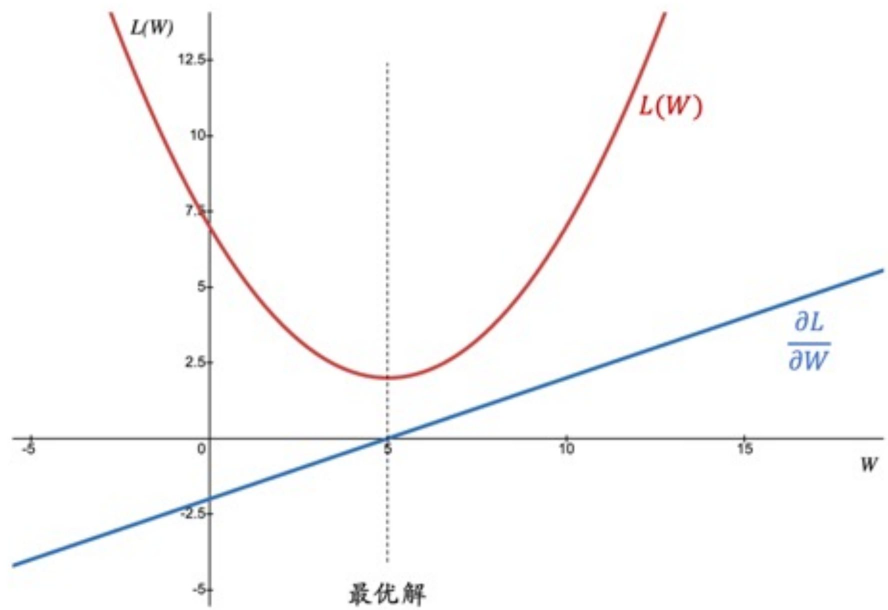
损失函数是一个与参数W（即权值矩阵，并且融合了偏置b）有关的函数，而参数优化的目标是找到能让损失函数L达到最小的参数W 。

损失函数 $L = \frac{1}{N}\sum_{i=1}^{N} L_i(f(x_i, W), y_i) + \lambda R(W)$

参数优化 $W^* = arg\min_{W} L(W)$

当损失函数L为凸函数时（常见情况），优化寻找的目标为让L的一阶导数为0的W：

$$\frac{dL}{dW} = 0$$



参数优化

通常情况下，L很复杂，难以直接求出让其导数为0的参数W（以及偏置b）。

以CIFAR-10的分类任务为例：

$$\underbrace{\begin{cases} \dfrac{dL}{d\mathrm{W}_1} = 0 \\ \dfrac{dL}{d\mathrm{W}_2} = 0 \\ \quad\vdots \\ \dfrac{dL}{d\mathrm{W}_{10}} = 0 \end{cases} \qquad \begin{cases} \dfrac{dL}{db_1} = 0 \\ \dfrac{dL}{db_2} = 0 \\ \quad\vdots \\ \dfrac{dL}{db_{10}} = 0 \end{cases}}$$

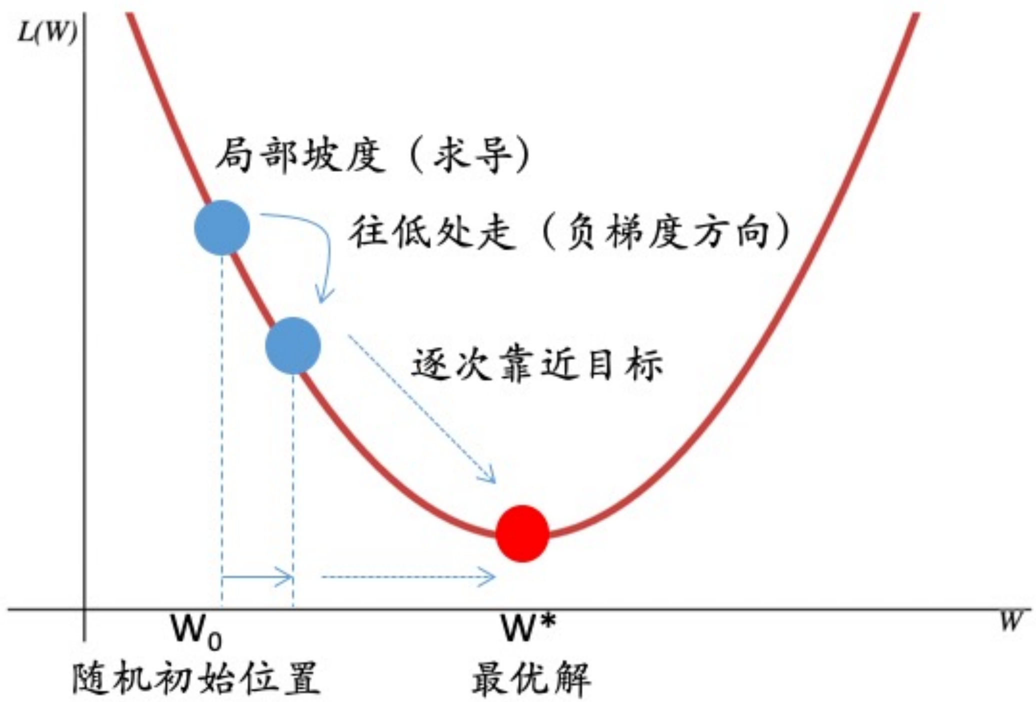
20个式子联立求解
直接求解难度通常很大
需要采用其他手段

梯度下降
gradient descent

梯度

损失函数 $L = \frac{1}{N}\sum_{i=1}^{N} L_i(f(x_i, W), y_i) + \lambda R(W)$

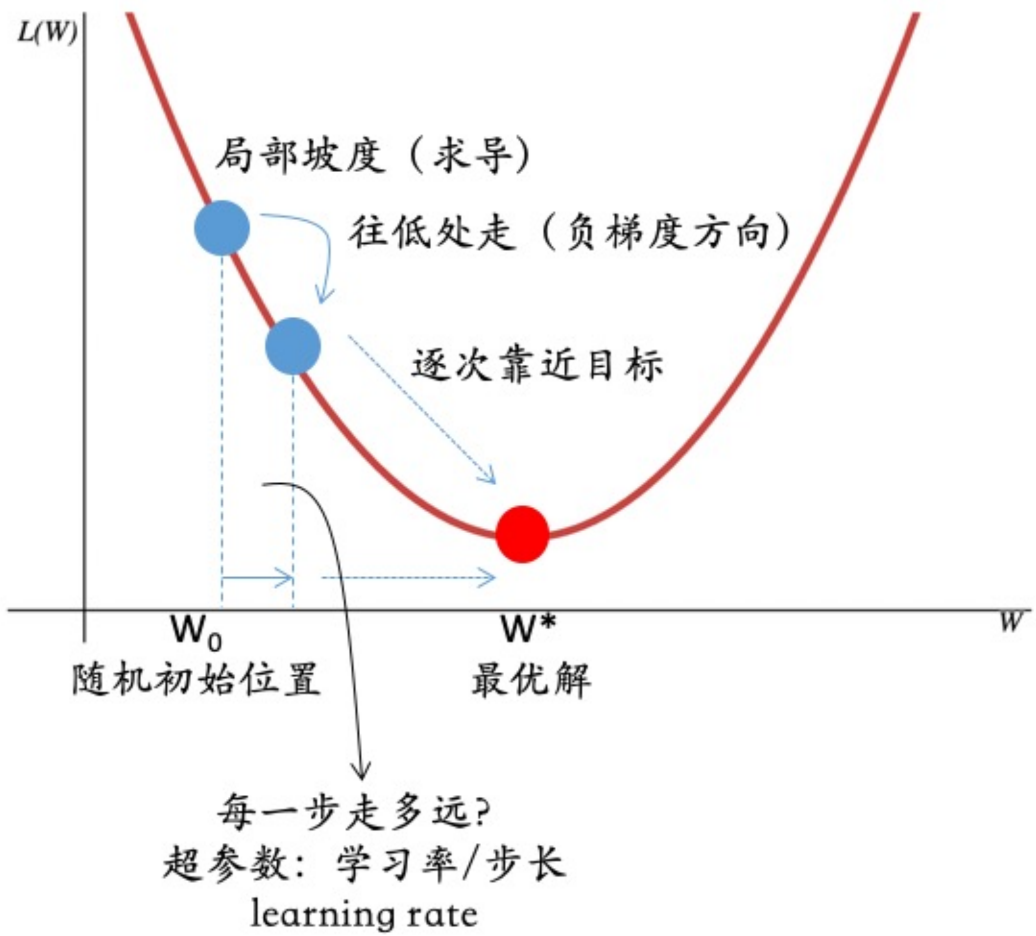
参数优化 $W^* = arg\min_{W} L(W)$



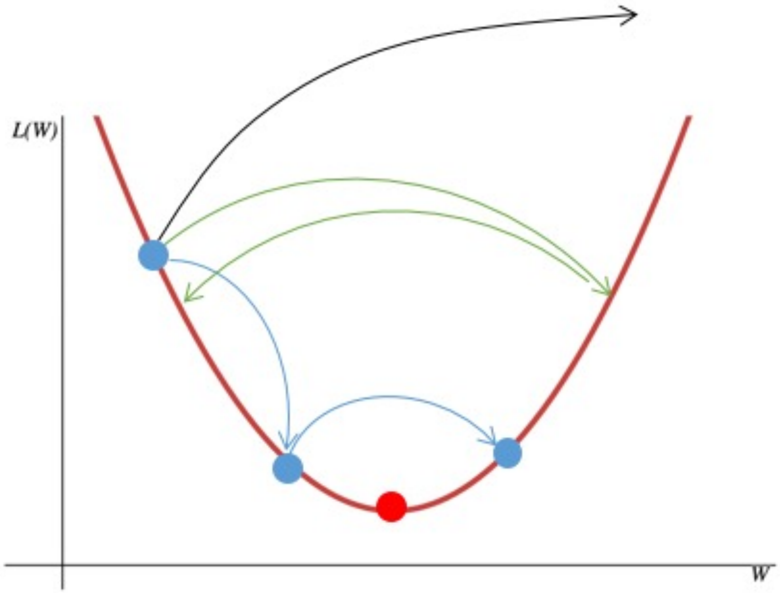
梯度

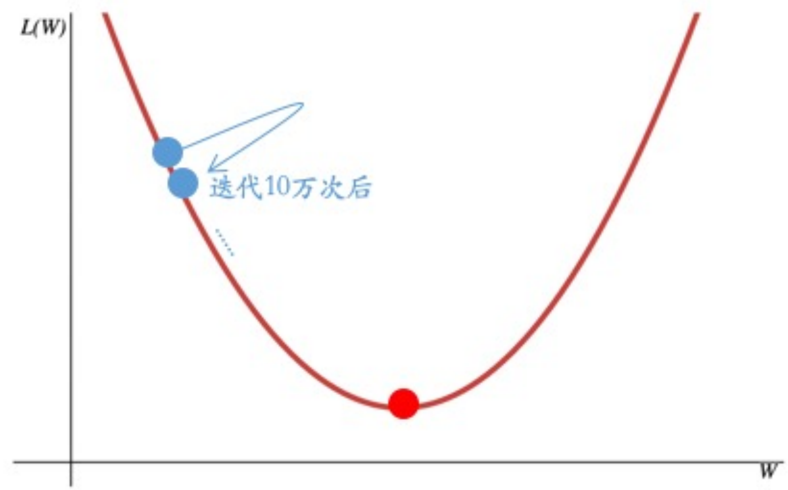
损失函数 $L = \frac{1}{N}\sum_{i=1}^{N} L_i(f(x_i, W), y_i) + \lambda R(W)$

参数优化 $W^* = arg\min_{W} L(W)$



梯度





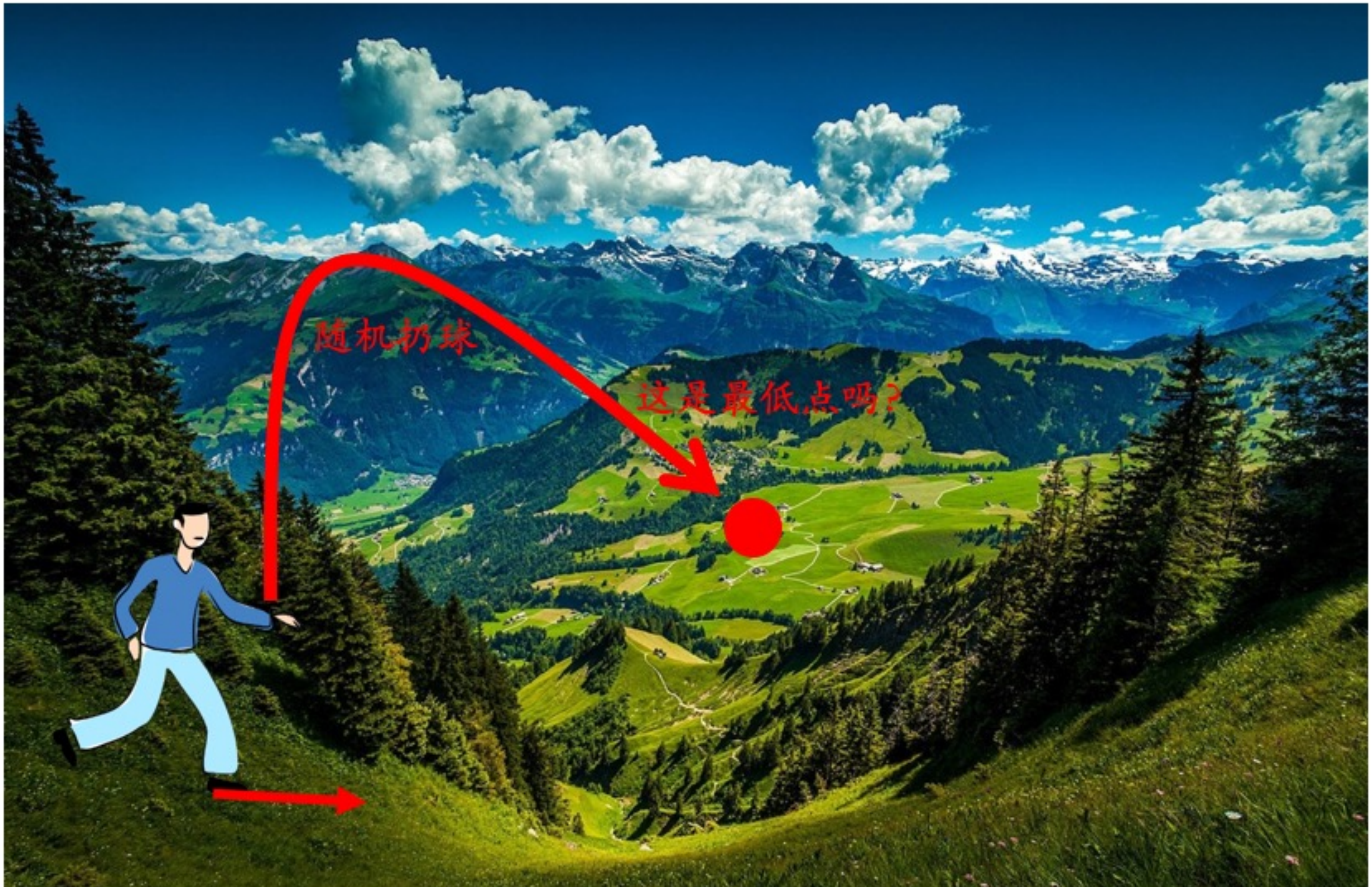
学习率过大
模型不稳定、难以收敛

学习率过小
训练速度慢、可能被困于局部最优解

梯度


while True:
  weights_grad = evaluate_gradient(loss_fun, data, weights)
  weights += - step_size * weights_grad # perform parameter update

梯度的计算：随机搜索法



梯度的计算：数值近似法

$$\frac{dL(w)}{dw} = \lim_{h \to 0} \frac{L(w+h) - L(w)}{h}$$

数值梯度 numeric gradient

优点：计算简单易懂

缺点：不精确、计算量大

例题：求损失函数$L(w) = w^2 + 3w + 5$在$w = 0.5$处的梯度。

解：假设h为0.0001，代入$w = 0.5$可得

$$\frac{dL(w)}{dw} \approx \frac{L(0.5 + 0.0001) - L(0.5)}{0.0001}$$

$$= \frac{(0.5001^2 + 3 \times 0.5001 + 5) - (0.5^2 + 3 \times 0.5 + 5)}{0.0001}$$

$$= 4.0001$$

梯度的计算：梯度解析法

解析梯度：analytical gradient

用微积分的方法算梯度

优点：精确值、计算量小

缺点：计算复杂、易错

例题：求损失函数$L(w) = w^2 + 3\mathrm{w} + 5$在$w = 0.5$处的梯度。

解：对$L(w) = w^2 + 3\mathrm{w} + 5$求导

$\nabla L(w) = 2w + 3$

代入$w = 0.5$可得

$\nabla L(0.5) = 2 \times 0.5 + 3 = 4$

梯度检验

数值梯度：简单不易错的近似值，速度慢

解析梯度：复杂易错的精确值，速度快

梯度检验：计算解析梯度，但用数值梯度对其计算的正确性进行验算。