线性分类器

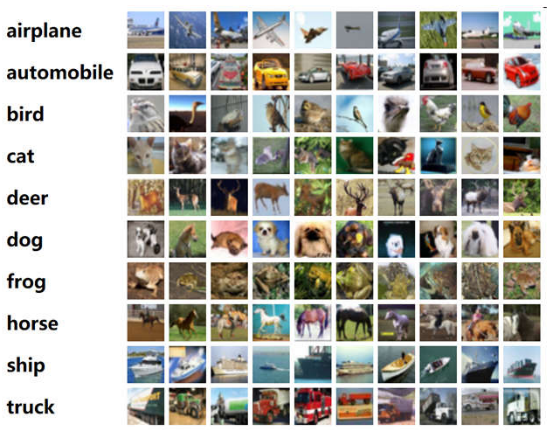
鲁 鹏
北京邮电大学 计算机学院 智能科学与技术中心

0

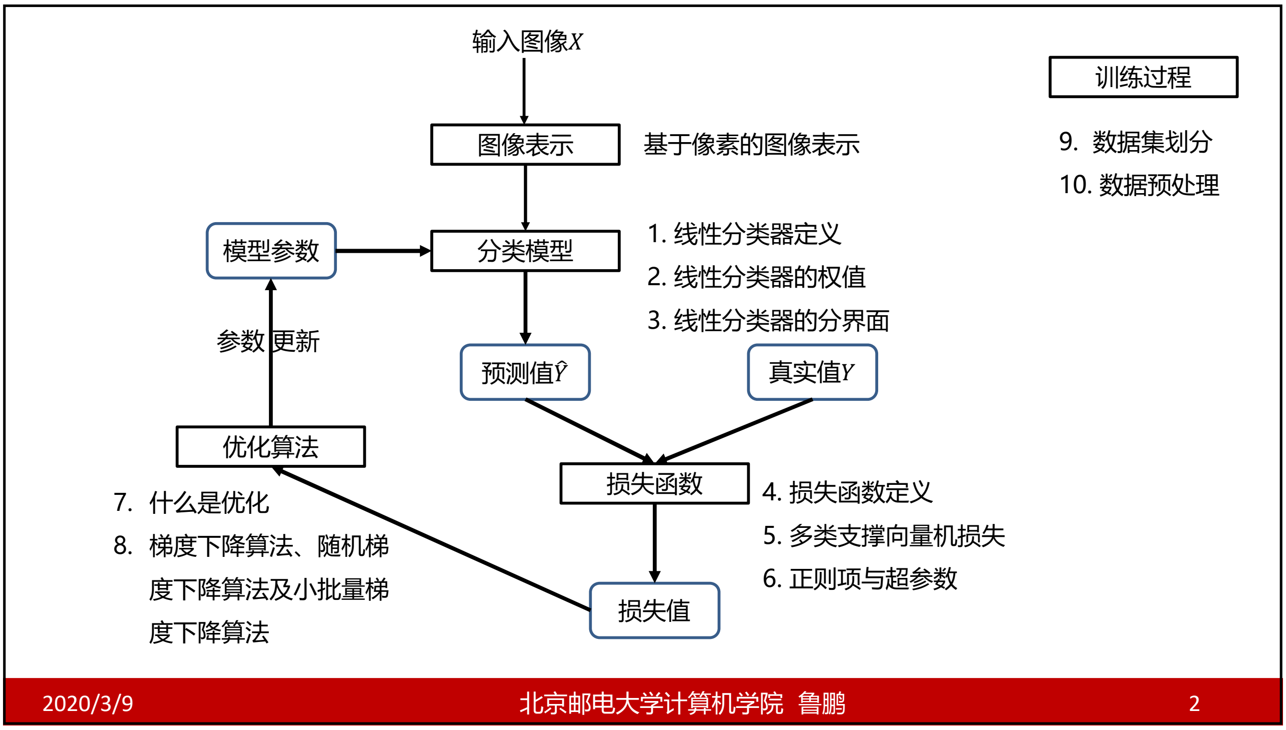
数据集介绍

- **CIFAR10数据集：**

  包含50000张训练样本、10000张测试样本分为飞机、汽车、鸟、猫、鹿、狗、蛙、马、船、卡车十个类别图像为彩色图像，其大小为32*32

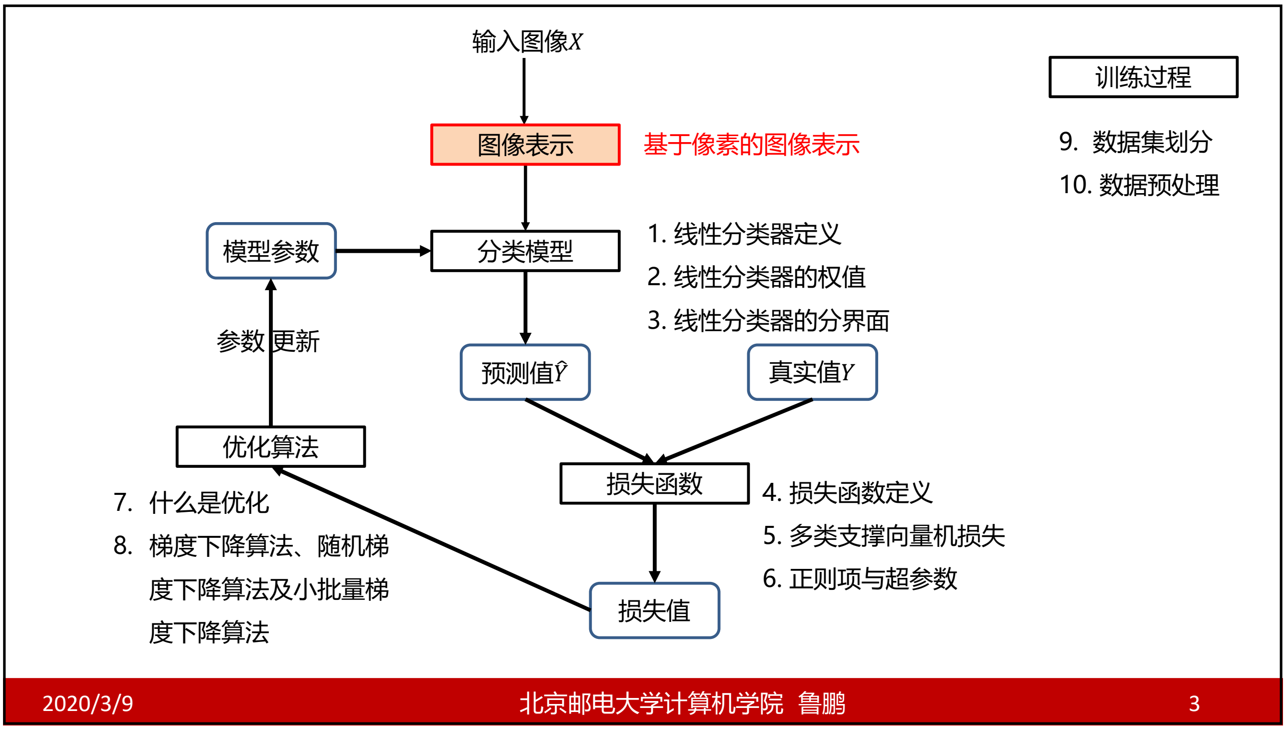


1



输入图像X
训练过程
图像表示
基于像素的图像表示
9. 数据集划分
10. 数据预处理
模型参数
分类模型
1. 线性分类器定义
2. 线性分类器的权值
3. 线性分类器的分界面
参数更新
预测值$\hat{Y}$
真实值Y
优化算法
损失函数
4. 损失函数定义
5. 多类支撑向量机损失
6. 正则项与超参数
7. 什么是优化
8. 梯度下降算法、随机梯度下降算法及小批量梯度下降算法
损失值
2020/3/9
北京邮电大学计算机学院 鲁鹏
2

2



输入图像X
训练过程
图像表示
基于像素的图像表示
9. 数据集划分
10. 数据预处理
模型参数
分类模型
1. 线性分类器定义
2. 线性分类器的权值
3. 线性分类器的分界面
参数更新
预测值$\hat{Y}$
真实值Y
优化算法
损失函数
4. 损失函数定义
5. 多类支撑向量机损失
6. 正则项与超参数
7. 什么是优化
8. 梯度下降算法、随机梯度下降算法及小批量梯度下降算法
损失值
2020/3/9
北京邮电大学计算机学院 鲁鹏
3

3

图像类型

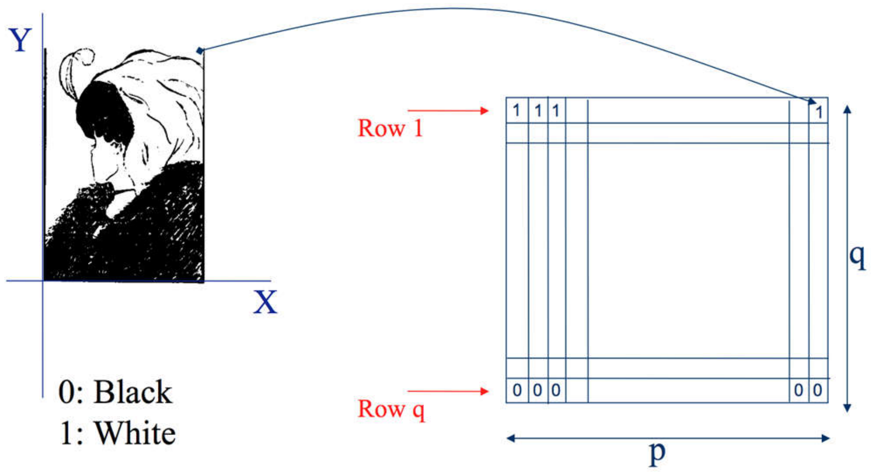
Binary

Gray Scale

Color

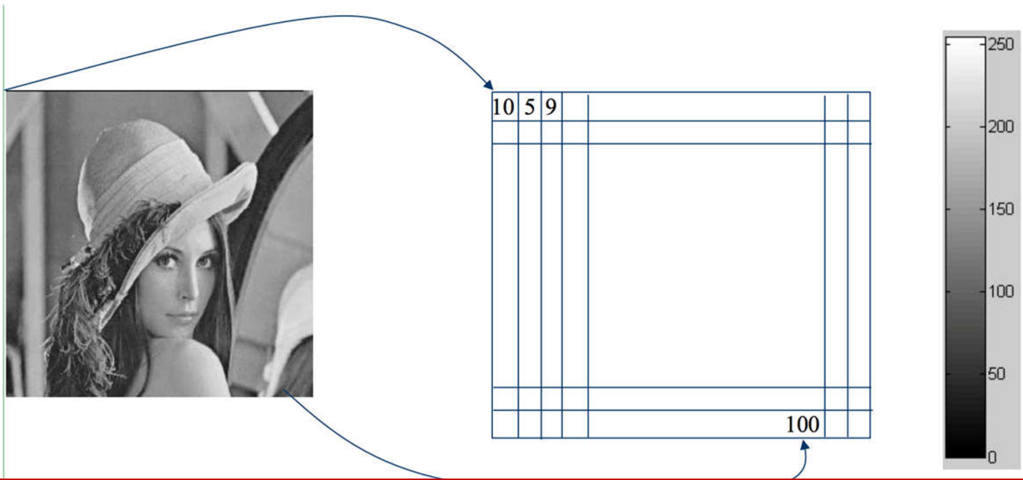
Source：Ulas Bagci

二进制图像



Source：Ulas Bagci

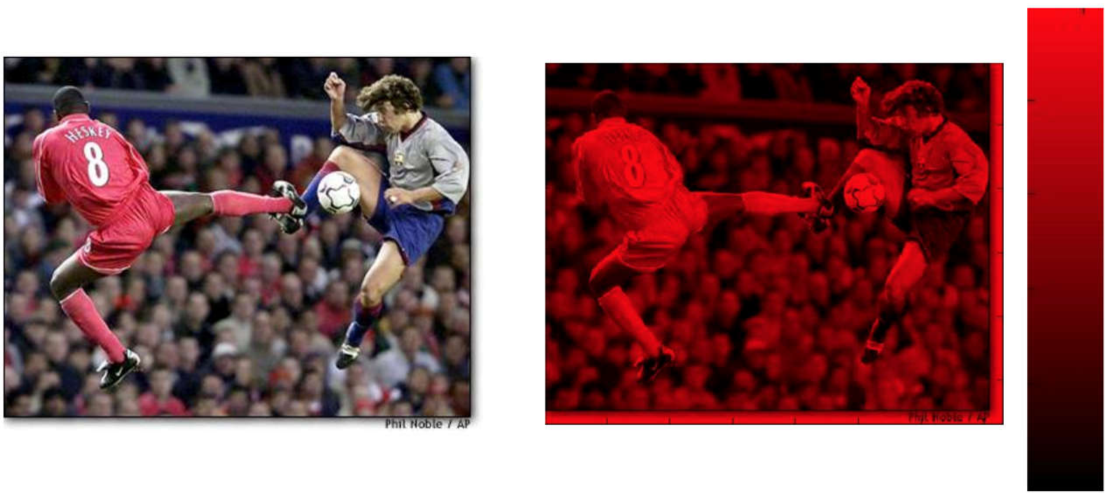
灰度图像



Source：Ulas Bagci

6

彩色图像



Source：Ulas Bagci

7

彩色图像

Source：Ulas Bagci

图像表示

大多数分类算法都要求输入向量！

图像表示

将图像转换成向量的方法有很多，这里我们用一种最简单的方法，直接将图像矩阵转换成向量

图像

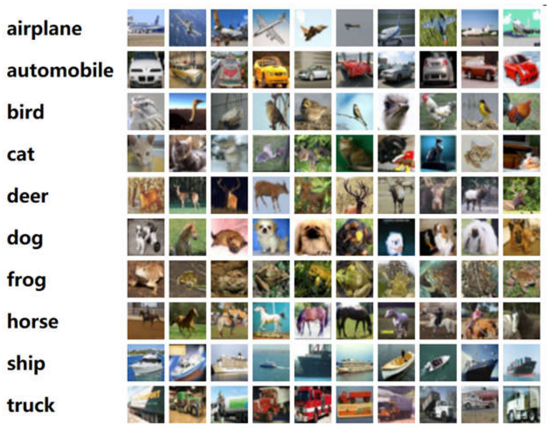
将矩阵转成列向量 →

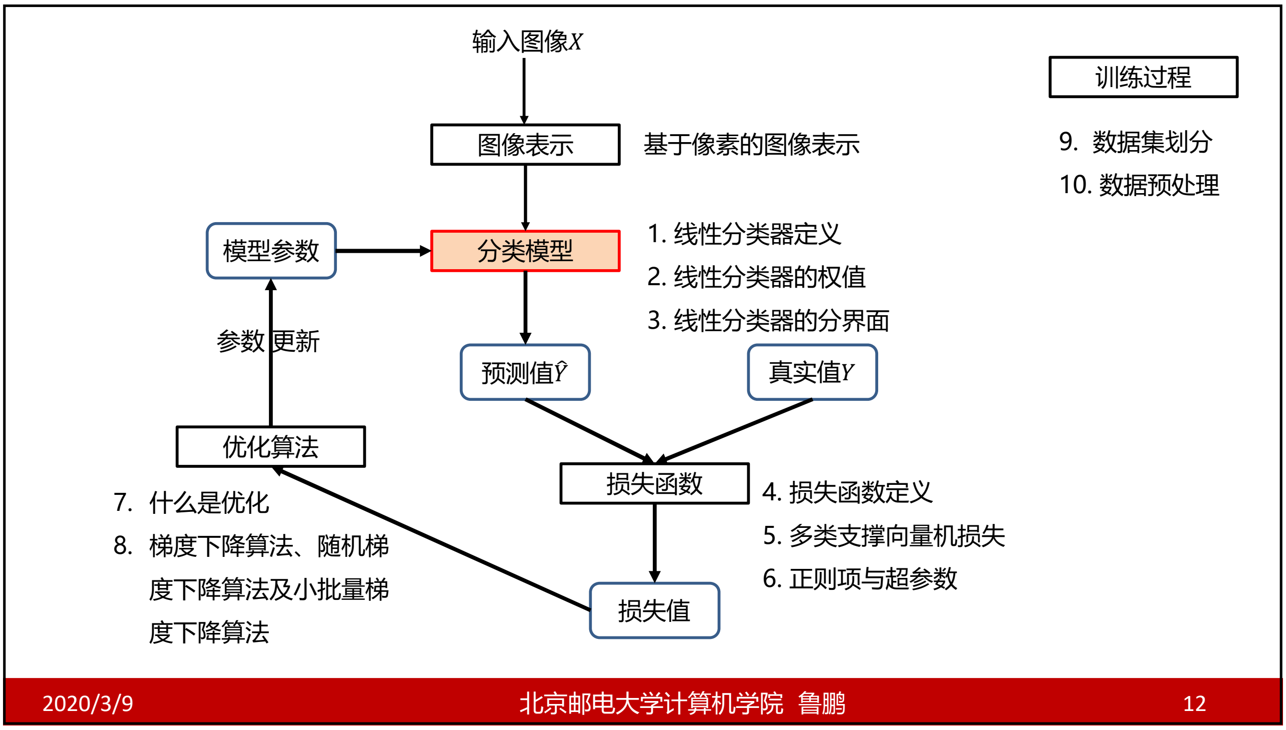
$$x = \begin{bmatrix} r_1 \\ g_1 \\ b_1 \\ \vdots \\ r_n \\ g_n \\ b_n \end{bmatrix}$$

图像表示

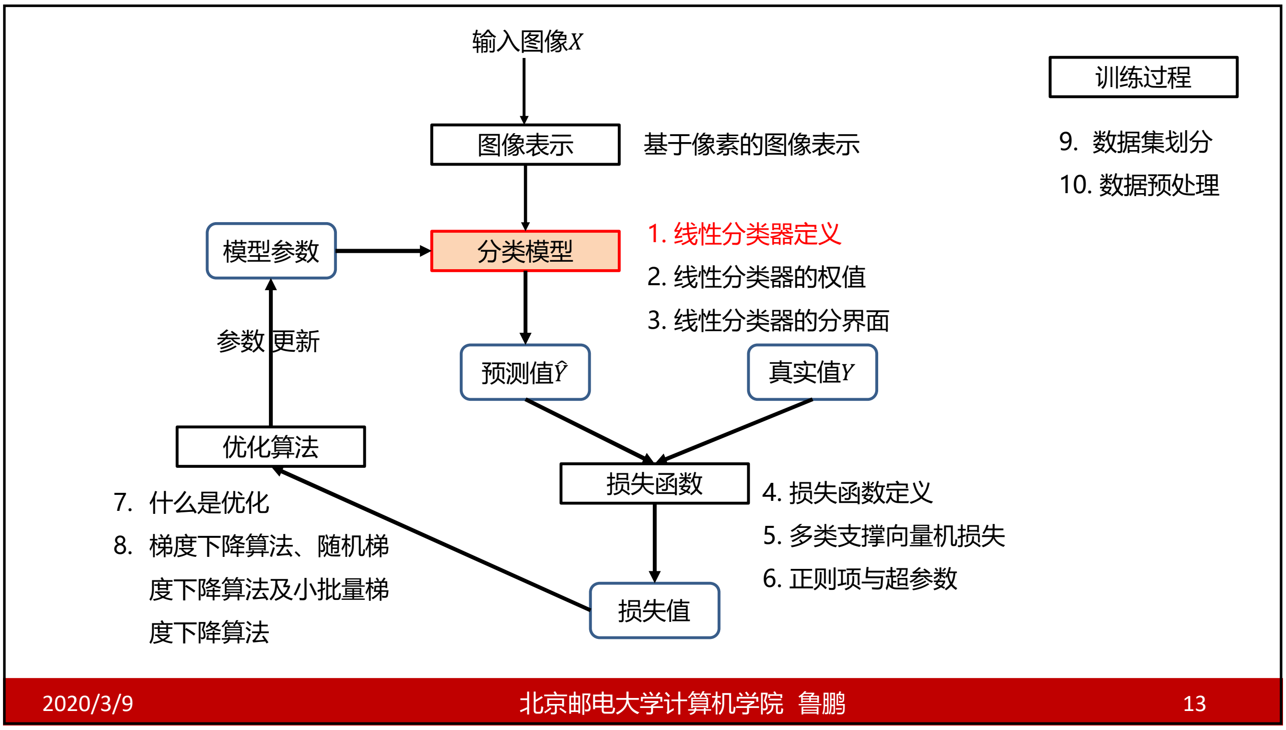
CIFAR10中每一张图像转换为向量是多少维？

答案：3072（=32*32*3）维列向量




输入图像X
训练过程
图像表示
基于像素的图像表示
9. 数据集划分
10. 数据预处理
模型参数
分类模型
1. 线性分类器定义
2. 线性分类器的权值
3. 线性分类器的分界面
参数更新
预测值Ŷ
真实值Y
优化算法
损失函数
4. 损失函数定义
5. 多类支撑向量机损失
6. 正则项与超参数
7. 什么是优化
8. 梯度下降算法、随机梯度下降算法及小批量梯度下降算法
损失值
2020/3/9
北京邮电大学计算机学院 鲁鹏
12

12


输入图像X
训练过程
图像表示
基于像素的图像表示
9. 数据集划分
10. 数据预处理
模型参数
分类模型
1. 线性分类器定义
2. 线性分类器的权值
3. 线性分类器的分界面
参数更新
预测值Ŷ
真实值Y
优化算法
损失函数
4. 损失函数定义
5. 多类支撑向量机损失
6. 正则项与超参数
7. 什么是优化
8. 梯度下降算法、随机梯度下降算法及小批量梯度下降算法
损失值
2020/3/9
北京邮电大学计算机学院 鲁鹏
13

13

为什么从线性分类器开始?

- 形式简单、易于理解
- 通过层级结构（神经网络）或者高维映射（支撑向量机）可以形成功能强大的非线性模型

什么是线性分类器

线性分类器是一种线性映射，将输入的图像特征映射为类别分数。

线性分类器定义

$\boldsymbol{x}$ 代表输入的 d 维图像向量，c 为类别个数

线性分类器定义

第i个类的线性分类器：

$$f_i(\boldsymbol{x}, \boldsymbol{w}_i) = \boldsymbol{w}_i^T \boldsymbol{x} + b_i\,,$$
$$\mathrm{i} = 1, \cdots, c$$

$\boldsymbol{x}$ 代表输入的 d 维图像向量，c 为类别个数

线性分类器定义

第i个类的线性分类器：

$$f_i(\boldsymbol{x}, \boldsymbol{w}_i) = \boldsymbol{w}_i^T \boldsymbol{x} + b_i \,, \quad \mathrm{i} = 1, \cdots, c$$

$\boldsymbol{x}$ 代表输入的 d 维图像向量，c 为类别个数

$\boldsymbol{w}_i = [w_{i1} \quad \cdots \quad w_{id}]^T$为第 i 个类别的权值向量，b_i为偏置

线性分类器定义

第i个类的线性分类器：

每个类都有自己的参数 $\boldsymbol{w}$ **和** b

$$f_i(\boldsymbol{x}, \boldsymbol{w}_i) = \boldsymbol{w}_i^T \boldsymbol{x} + b_i \,, \quad \mathrm{i} = 1, \cdots, c$$

$\boldsymbol{x}$ 代表输入的 d 维图像向量，c 为类别个数

$\boldsymbol{w}_i = [w_{i1} \quad \cdots \quad w_{id}]^T$为第 i 个类别的权值向量，b_i为偏置

线性分类器决策

第i个类的线性分类器：

每个类都有自己的参数 $\boldsymbol{w}$ 和 b

$$f_i(\boldsymbol{x}, \boldsymbol{w}_i) = \boldsymbol{w}_i^T \boldsymbol{x} + b_i\,,$$
$$\mathrm{i} = 1, \cdots, c$$

$\boldsymbol{x}$ 代表输入的 d 维图像向量，c 为类别个数

$\boldsymbol{w}_i = [w_{i1} \ \cdots \ w_{id}]^T$为第 i 个类别的权值向量，b_i为偏置

决策规则：

如果$f_i(\boldsymbol{x}) > f_j(\boldsymbol{x})$， $\forall j \neq i$,

则决策输入图像 $\boldsymbol{x}$ 属于第i类

线性分类器示例

任务：为图片分配类别标签（汽车类、猫类、鸟类）

图 片

线性分类器

$$\boldsymbol{w}_i^T \boldsymbol{x} + b_i = f_i, i = 1,2,3$$

?

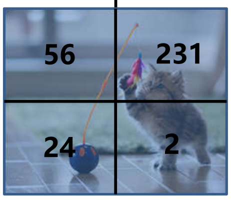
线性分类器示例

图像表示$\boldsymbol{x}$

$$\begin{bmatrix} 56 \\ 231 \\ 24 \\ 2 \end{bmatrix}$$

线性分类器决策步骤：

1.图像表示成向量



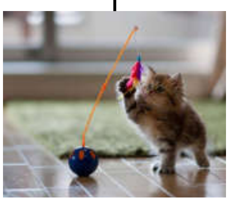
线性分类器示例

	权值 $\boldsymbol{w}_i^T$					图像表示$\boldsymbol{x}$		偏移 b_i				得分f_i
汽车类	$\boldsymbol{w}_1^T$	0.2	-0.5	0.1	2.0	56		1.1	b_1		f_1	-97.9
猫类	$\boldsymbol{w}_2^T$	1.5	1.3	2.1	0.0	231	+	3.2	b_2	=	f_2	434.7
鸟类	$\boldsymbol{w}_3^T$	0	0.25	0.25	-0.3	24		-1.2	b_3		f_3	63.15
						2						

线性分类器决策步骤：

1.图像表示成向量

2.计算当前图片每个类别的分数



线性分类器示例

		权值 $\boldsymbol{w}_i^T$				图像表示$\boldsymbol{x}$		偏移 b_i				得分f_i
汽车类	$\boldsymbol{w}_1^T$	0.2	-0.5	0.1	2.0	56		1.1	b_1		f_1	-97.9
猫类	$\boldsymbol{w}_2^T$	1.5	1.3	2.1	0.0	231	+	3.2	b_2	=	f_2	434.7
鸟类	$\boldsymbol{w}_3^T$	0	0.25	0.25	-0.3	24		-1.2	b_3		f_3	63.15
						2						

线性分类器决策步骤：

1.图像表示成向量

2.计算当前图片每个类别的分数

3.按类别得分判定当前图像

猫类

线性分类器示例

		权值 $\boldsymbol{w}_i^T$				图像表示$\boldsymbol{x}$		偏移 b_i				得分f_i
汽车类	$\boldsymbol{w}_1^T$	0.2	-0.5	0.1	2.0	56		1.1	b_1		f_1	-97.9
猫类	$\boldsymbol{w}_2^T$	1.5	1.3	2.1	0.0	231	+	3.2	b_2	=	f_2	434.7
鸟类	$\boldsymbol{w}_3^T$	0	0.25	0.25	-0.3	24		-1.2	b_3		f_3	63.15
						2						

权值矩阵 W　　偏移向量 $\boldsymbol{b}$　　得分向量 $\boldsymbol{f}$

线性分类器的矩阵表示

		权值 $\boldsymbol{w}_i^T$				图像表示$\boldsymbol{x}$		偏移 b_i				得分f_i
汽车类	$\boldsymbol{w}_1^T$	0.2	-0.5	0.1	2.0	56		1.1	b_1		f_1	-97.9
猫类	$\boldsymbol{w}_2^T$	1.5	1.3	2.1	0.0	231	+	3.2	b_2	=	f_2	434.7
鸟类	$\boldsymbol{w}_3^T$	0	0.25	0.25	-0.3	24		-1.2	b_3		f_3	63.15
						2						

权值矩阵 W　　偏移向量 $\boldsymbol{b}$　　得分向量 $\boldsymbol{f}$

线性分类器的矩阵表示:

$$\boxed{\boldsymbol{f}(\boldsymbol{x}, W) = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}}$$

其中， $\boldsymbol{x}$ 代表输入图像，其维度为 d ，

$\boldsymbol{f}$ 为分数向量，其维度等于类别个数 c ，

$\boldsymbol{W} = [\boldsymbol{w_1} \quad \cdots \quad \boldsymbol{w}_c]^T$为权值矩阵 ,

$\boldsymbol{w_i} = [w_i \quad \cdots \quad w_{id}]^T$为第 i 个类别的权值向量,

$\boldsymbol{b} = [b_1 \quad \cdots \quad b_c]^T$为偏置向量 , b_i为第 i 个类别的偏置。

26

线性分类器的矩阵表示

问题：CIFAR10 数据集分类任务的分类器， $\boldsymbol{W}$，$\boldsymbol{x}$，$\boldsymbol{b}$的维度是多少?

线性分类器的矩阵表示:

$$\boxed{\boldsymbol{f}(\boldsymbol{x}, W) = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}}$$

其中， $\boldsymbol{x}$ 代表输入图像，其维度为 d ，

$\boldsymbol{f}$ 为分数向量，其维度等于类别个数 c ，

$\boldsymbol{W} = [\boldsymbol{w_1} \quad \cdots \quad \boldsymbol{w}_c]^T$为权值矩阵 ,

$\boldsymbol{w_i} = [w_i \quad \cdots \quad w_{id}]^T$为第 i 个类别的权值向量,

$\boldsymbol{b} = [b_1 \quad \cdots \quad b_c]^T$为偏置向量 , b_i为第 i 个类别的偏置。

27

线性分类器的矩阵表示

问题：CIFAR10 数据集分类任务的分类器， $\boldsymbol{W}$， $\boldsymbol{x}$， $\boldsymbol{b}$的维度是多少?

回答：CIFAR10有10个类别且图像大小为32x32x3，因此:

$\boldsymbol{x}$ 是图像向量，其维度为3072维；

$\boldsymbol{W}$ 是权值矩阵，其维度为10x3072；

$\boldsymbol{b}$ 是偏置向量，其维度为10x1的向量；

$\boldsymbol{f}$ 是得分向量，其维度为10x1的向量；

线性分类器的矩阵表示：

$$\boxed{\boldsymbol{f}(\boldsymbol{x}, W) = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}}$$

其中， $\boldsymbol{x}$ 代表输入图像，其维度为 d ，

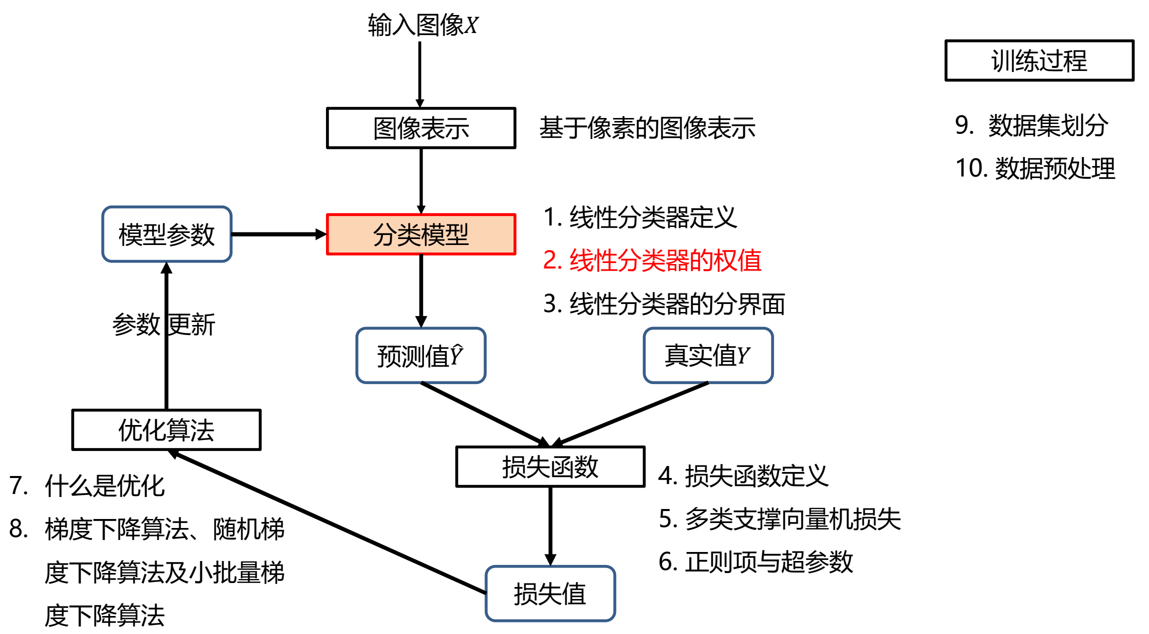
$\boldsymbol{f}$ 为分数向量，其维度等于类别个数 c ，

$\boldsymbol{W} = [\boldsymbol{w_1} \quad \cdots \quad \boldsymbol{w_c}]^T$为权值矩阵 ，

$\boldsymbol{w_i} = [w_i \quad \cdots \quad w_{id}]^T$为第 i 个类别的权值向量，

$\boldsymbol{b} = [b_1 \quad \cdots \quad b_c]^T$为偏置向量 , b_i为第 i 个类别的偏置。

28



29

线性分类器的权值向量

第i个类的线性分类器:

$$f_i(\boldsymbol{x},\boldsymbol{w}_i)=\boldsymbol{w}_i^T\boldsymbol{x}+b_i\ ,\quad \mathrm{i}=1,\cdots,c$$

$\boldsymbol{x}$ 代表输入的 d 维图像向量，c 为类别个数

$\boldsymbol{w}_i=[w_{i1}\ \ \cdots\ \ w_{id}]^T$为第 i 个类别的权值向量，b_i为偏置

w_1 w_2 w_3 w_4 w_5 w_6 w_7 w_8 w_9 w_{10}

汽车模板看起来记录的红色的汽车

马的模板看起来记录了两个马头

线性分类器的权值向量

第i个类的线性分类器:

$$f_i(\boldsymbol{x},\boldsymbol{w}_i)=\boldsymbol{w}_i^T\boldsymbol{x}+b_i\ ,\quad \mathrm{i}=1,\cdots,c$$

$\boldsymbol{x}$ 代表输入的 d 维图像向量，c 为类别个数

$\boldsymbol{w}_i=[w_{i1}\ \ \cdots\ \ w_{id}]^T$为第 i 个类别的权值向量，b_i为偏置

w_1 w_2 w_3 w_4 w_5 w_6 w_7 w_8 w_9 w_{10}

- 权值看做是一种模板

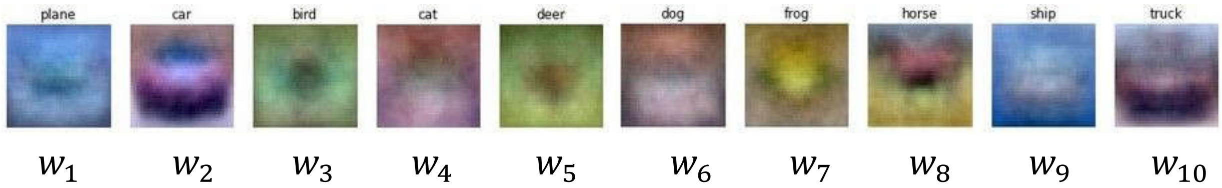
线性分类器的权值向量

第i个类的线性分类器：

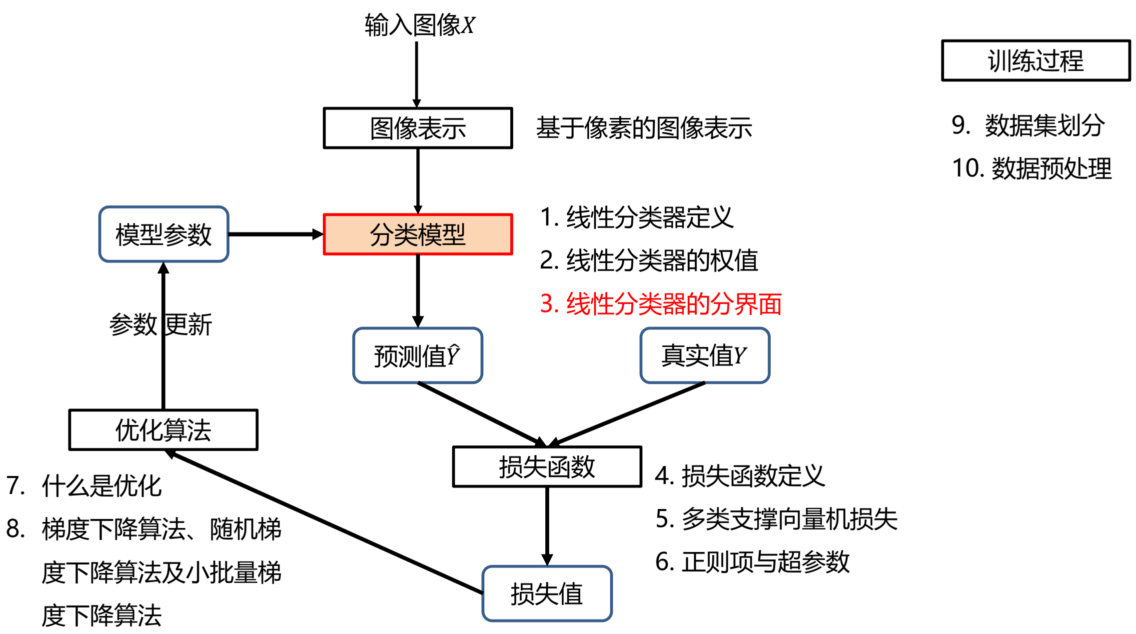
$$f_i(\boldsymbol{x}, \boldsymbol{w}_i) = \boldsymbol{w}_i^T \boldsymbol{x} + b_i\,,$$
$$\mathrm{i} = 1, \cdots, c$$

$\boldsymbol{x}$ 代表输入的 d 维图像向量，c 为类别个数

$\boldsymbol{w}_i = [w_{i1} \;\; \cdots \;\; w_{id}]^T$为第 i 个类别的权值向量，b_i为偏置

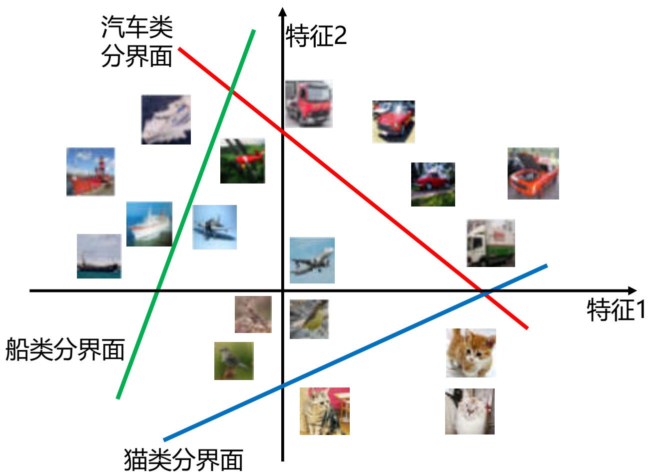


- 权值看做是一种模板
- 输入图像与评估模板的匹配程度越高，分类器输出的分数就越高



线性分类器的决策边界

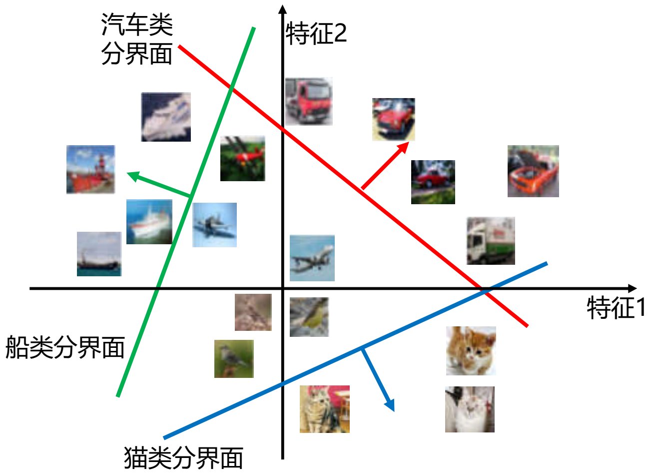
线性分类器的决策边界



- 分数等于0的线就是决策面

$$\boldsymbol{w}_i^T \boldsymbol{x} + b_i = 0, i = 1, \cdots, c$$

线性分类器的决策边界



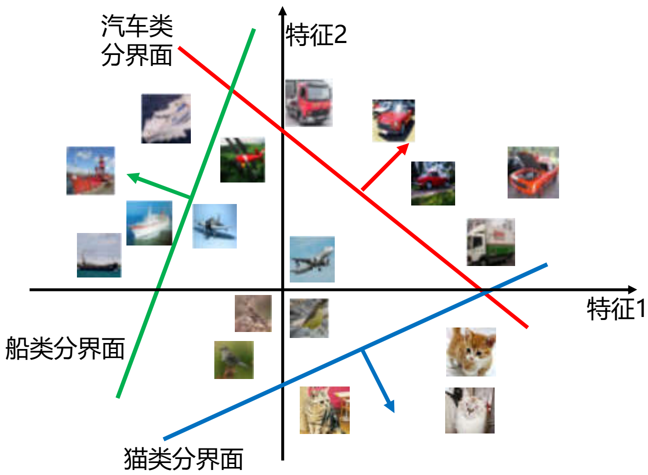
- 分数等于0的线就是决策面

  $\boldsymbol{w}_i^T\boldsymbol{x} + b_i = 0, i = 1, \cdots, c$

- $\boldsymbol{w}$ 控制着线的方向

36

线性分类器的决策边界



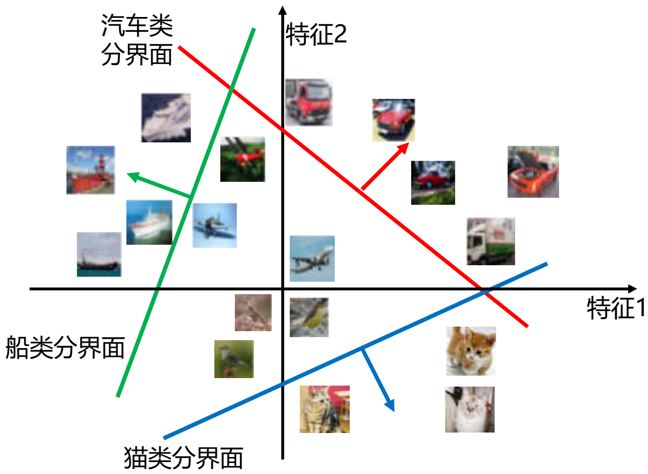
- 分数等于0的线就是决策面

  $\boldsymbol{w}_i^T\boldsymbol{x} + b_i = 0, i = 1, \cdots, c$

- $\boldsymbol{w}$ 控制着线的方向
- b 控制着线的偏移

37

线性分类器的决策边界



- 分数等于0的线就是决策面

  $w_i^T x + b_i = 0, i = 1, \cdots, c$

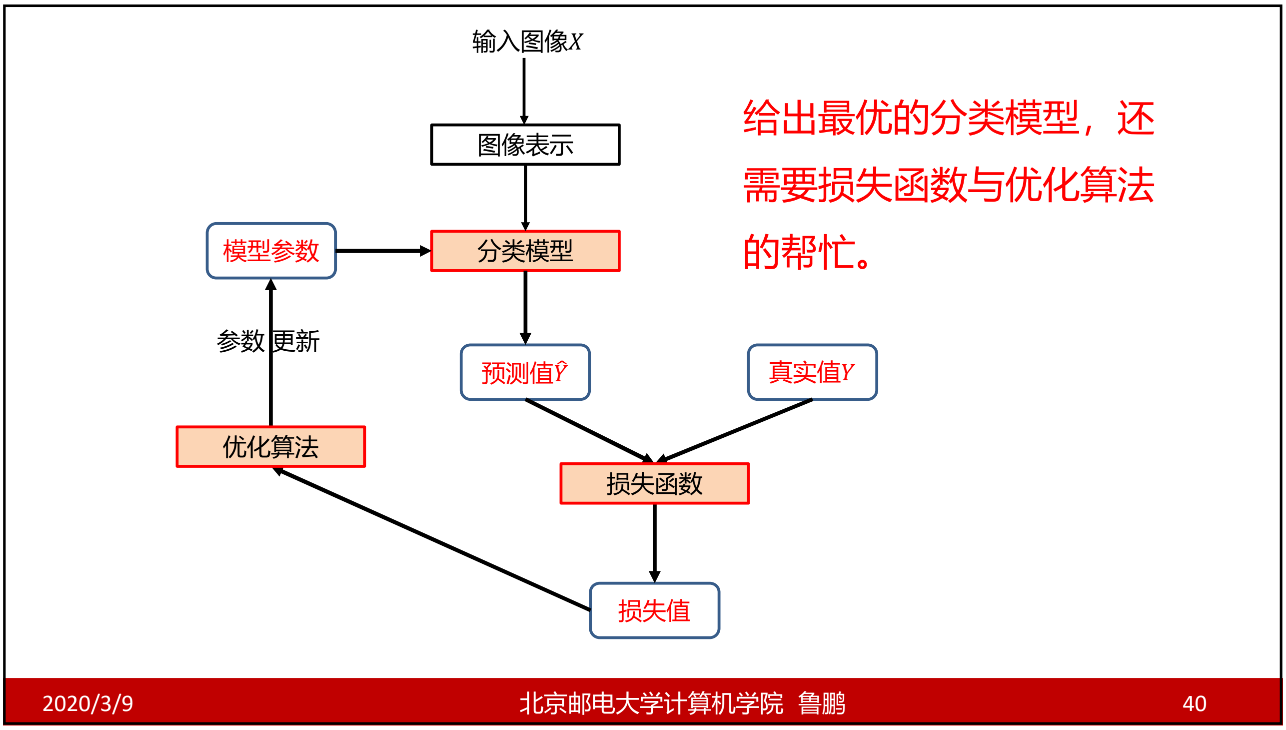
- w 控制着线的方向
- b 控制着线的偏移
- 箭头方向代表分类器的正方向，沿着箭头方向距离决策面越远分数就越高。

38

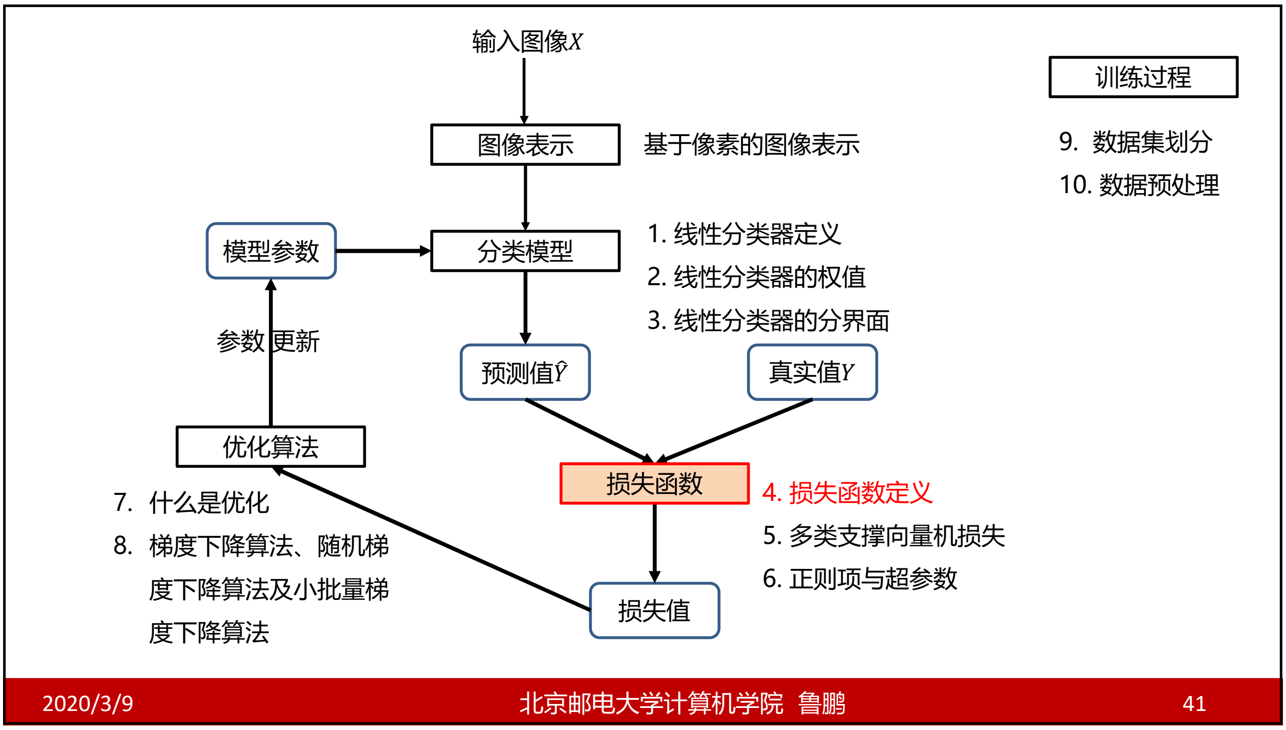
线性分类器小结

1. 线性分类器的定义
2. 线性分类器的决策
3. 线性分类器的矩阵表示
4. 线性分类器的权值向量
5. 线性分类器的决策边界

39


输入图像X
图像表示
模型参数
分类模型
参数更新
预测值$\hat{Y}$
真实值Y
优化算法
损失函数
损失值
给出最优的分类模型，还需要损失函数与优化算法的帮忙。
2020/3/9
北京邮电大学计算机学院 鲁鹏
40

40


输入图像X
训练过程
图像表示
基于像素的图像表示
9. 数据集划分
10. 数据预处理
模型参数
分类模型
1. 线性分类器定义
2. 线性分类器的权值
3. 线性分类器的分界面
参数更新
预测值$\hat{Y}$
真实值Y
优化算法
损失函数
4. 损失函数定义
7. 什么是优化
8. 梯度下降算法、随机梯度下降算法及小批量梯度下降算法
5. 多类支撑向量机损失
6. 正则项与超参数
损失值
2020/3/9
北京邮电大学计算机学院 鲁鹏
41

41

如何衡量分类器对当前样本的效果好坏?

分类器1

$$\underbrace{\begin{bmatrix} 0.2 & -0.5 & 0.1 & 2.0 \\ 1.5 & 1.3 & 2.1 & 0.0 \\ 0 & 0.25 & 0.25 & -0.3 \end{bmatrix}}_{W1} \begin{bmatrix} 56 \\ 231 \\ 24 \\ 2 \end{bmatrix} + \underbrace{\begin{bmatrix} 1.1 \\ 3.2 \\ -1.2 \end{bmatrix}}_{b1} = \underbrace{\begin{bmatrix} -97.9 \\ 434.7 \\ 63.15 \end{bmatrix}}_{f}$$

f	
-97.9	汽车类分数
434.7	猫类分数
63.15	船类分数

如何衡量分类器对当前样本的效果好坏?

分类器2

$$\underbrace{\begin{bmatrix} 0.2 & -0.1 & 0.1 & 1.7 \\ 1.5 & 0.2 & 2.1 & 5.0 \\ -1.2 & 1.5 & 0.25 & 0.5 \end{bmatrix}}_{W2} \begin{bmatrix} 56 \\ 231 \\ 24 \\ 2 \end{bmatrix} + \underbrace{\begin{bmatrix} 2.1 \\ -0.2 \\ 2.4 \end{bmatrix}}_{b2} = \underbrace{\begin{bmatrix} -6.1 \\ 190.6 \\ 286.3 \end{bmatrix}}_{f}$$

f	
-6.1	汽车类分数
190.6	猫类分数
286.3	船类分数

损失函数

对示例样本，分类器1与分类器2的分类谁的效果更好?

需要损失函数来帮忙

什么是损失函数?

损失函数搭建了模型性能与模型参数之间的桥梁，指导模型参数优化。

- 损失函数是一个函数，用于度量给定分类器的预测值与真实值的不一致程度，其输出通常是一个非负实值。
- 其输出的非负实值可以作为反馈信号来对分类器参数进行调整，以降低当前示例对应的损失值，提升分类器的分类效果。

什么是损失函数?

损失函数的一般定义：

$$L = \frac{1}{N}\sum_i L_i\left(f(x_i, W), y_i\right)$$

什么是损失函数?

损失函数的一般定义：

x_i 表示数据集中第 i 张图片；

$$L = \frac{1}{N}\sum_i L_i\left(f(x_i, W), y_i\right)$$

什么是损失函数?

损失函数的一般定义：

$$L = \frac{1}{N}\sum_i L_i\left(f(x_i, W), y_i\right)$$

x_i 表示数据集中第 i 张图片；

$f(x_i, W)$ 为分类器对 x_i 的类别预测；

什么是损失函数?

损失函数的一般定义：

$$L = \frac{1}{N}\sum_i L_i\left(f(x_i, W), y_i\right)$$

x_i 表示数据集中第 i 张图片；

$f(x_i, W)$ 为分类器对 x_i 的类别预测；

y_i为样本 i 真实类别标签（整数）；

什么是损失函数?

损失函数的一般定义:

$$L = \frac{1}{N}\sum_i L_i\left(f(x_i, W), y_i\right)$$

x_i 表示数据集中第 i 张图片;

$f(x_i, W)$ 为分类器对 x_i 的类别预测;

y_i为样本 i 真实类别标签（整数）;

L_i为第 i 个样本的损失当预测值;

什么是损失函数?

损失函数的一般定义:

$$L = \frac{1}{N}\sum_i L_i\left(f(x_i, W), y_i\right)$$

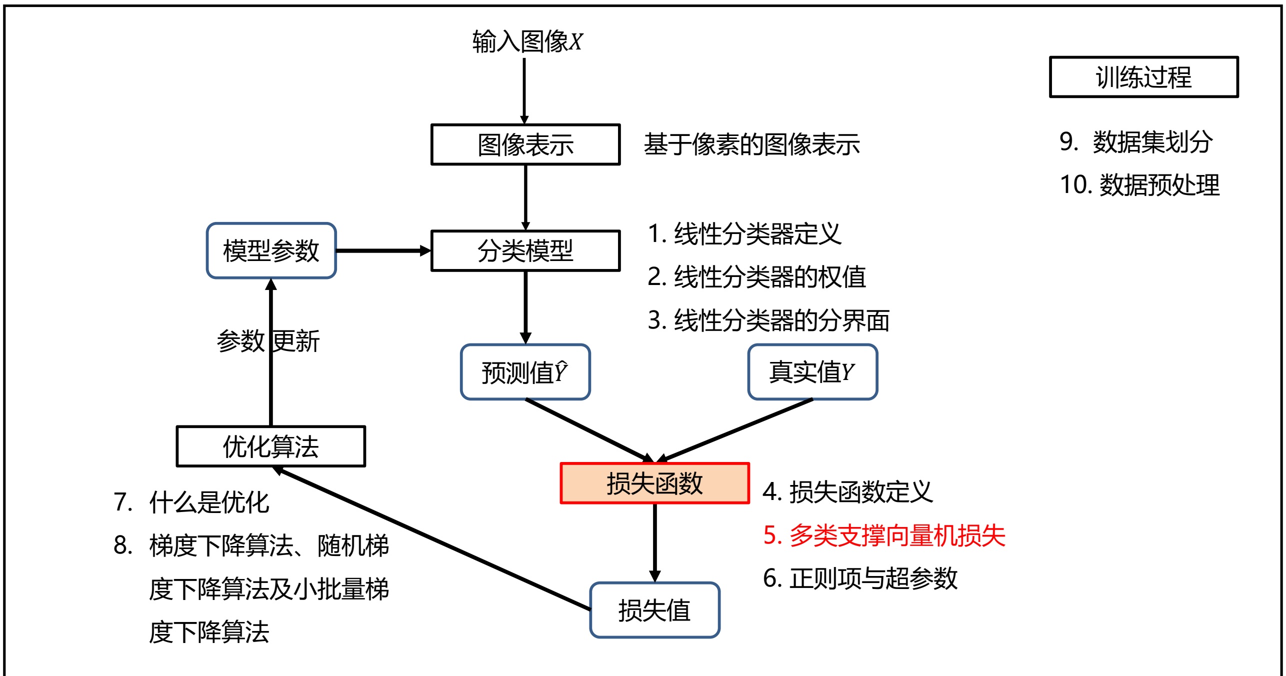
x_i 表示数据集中第 i 张图片;

$f(x_i, W)$ 为分类器对 x_i 的类别预测;

y_i为样本 i 真实类别标签（整数）;

L_i为第 i 个样本的损失当预测值;

L为数据集损失，它是数据集中所有样本损失的平均。


输入图像X
训练过程
图像表示
基于像素的图像表示
9. 数据集划分
10. 数据预处理
模型参数
分类模型
1. 线性分类器定义
2. 线性分类器的权值
3. 线性分类器的分界面
参数更新
预测值$\hat{Y}$
真实值Y
优化算法
损失函数
4. 损失函数定义
5. 多类支撑向量机损失
6. 正则项与超参数
7. 什么是优化
8. 梯度下降算法、随机梯度下降算法及小批量梯度下降算法
损失值

多类支撑向量机损失

多类支撑向量机损失

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

j ——类别标签，取值范围{1，2，…，c}；

多类支撑向量机损失

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

j ——类别标签，取值范围{1，2，…，c}；

w_j, b_j —— 第j个类别分类器的参数；

多类支撑向量机损失

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

j ——类别标签，取值范围{1，2，…，c}；

w_j, b_j —— 第j个类别分类器的参数；

x_i ——表示数据集中的第 i 个样本

多类支撑向量机损失

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

j ——类别标签，取值范围{1，2，…，c}；

w_j, b_j —— 第j个类别分类器的参数；

x_i ——表示数据集中的第 i 个样本

s_{ij} ——第i个样本第j类别的预测分数

多类支撑向量机损失

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

j ——类别标签，取值范围{1，2，…，c}；

w_j, b_j —— 第j个类别分类器的参数；

x_i ——表示数据集中的第 i 个样本

s_{ij} ——第i个样本第j类别的预测分数

第i个样本的多类支撑向量机损失定义如下：

$$L_i = \sum_{j \neq y_i} \begin{cases} 0 & if\ s_{y_i} \geq s_{ij} + 1 \\ s_{ij} - s_{y_i} + 1 & otherwise \end{cases}$$
$$= \sum_{j \neq y_i} \max(0, s_{ij} - s_{y_i} + 1)$$

多类支撑向量机损失

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

j ——类别标签，取值范围{1，2，…，c}；

w_j, b_j —— 第j个类别分类器的参数；

x_i ——表示数据集中的第 i 个样本

s_{ij} ——第i个样本第j类别的预测分数

s_{y_i} ——第i个样本真实类别的预测分数

第i个样本的多类支撑向量机损失定义如下：

$$L_i = \sum_{j \neq y_i} \begin{cases} 0 & if\ s_{y_i} \geq s_{ij} + 1 \\ s_{ij} - s_{y_i} + 1 & otherwise \end{cases}$$
$$= \sum_{j \neq y_i} \max(0, s_{ij} - s_{y_i} + 1)$$

多类支撑向量机损失

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

j ——类别标签，取值范围{1，2，…，c}；

w_j, b_j —— 第j个类别分类器的参数；

x_i ——表示数据集中的第 i 个样本

s_{ij} ——第i个样本第j类别的预测分数

s_{y_i} ——第i个样本真实类别的预测分数

第i个样本的多类支撑向量机损失定义如下：

$$L_i = \sum_{j \neq y_i} \begin{cases} 0 & if\ s_{y_i} \geq s_{ij} + 1 \\ s_{ij} - s_{y_i} + 1 & otherwise \end{cases} = \sum_{j \neq y_i} \max(0, s_{ij} - s_{y_i} + 1)$$

◆ 正确类别的得分比不正确类别的得分高出1分，就没有损失

多类支撑向量机损失

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

j ——类别标签，取值范围{1，2，…，c}；

w_j, b_j —— 第j个类别分类器的参数；

x_i ——表示数据集中的第 i 个样本

s_{ij} ——第i个样本第j类别的预测分数

s_{y_i} ——第i个样本真实类别的预测分数

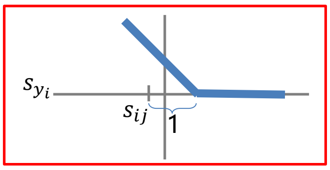
第i个样本的多类支撑向量机损失定义如下：

$$L_i = \sum_{j \neq y_i} \begin{cases} 0 & if\ s_{y_i} \geq s_{ij} + 1 \\ s_{ij} - s_{y_i} + 1 & otherwise \end{cases} = \sum_{j \neq y_i} \max(0, s_{ij} - s_{y_i} + 1)$$

◆ 正确类别的得分比不正确类别的得分高出1分，就没有损失

◆ 否则，就会产生损失

多类支撑向量机损失



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

j ——类别标签，取值范围{1，2，…，c}；

w_j, b_j —— 第j个类别分类器的参数；

x_i ——表示数据集中的第 i 个样本

s_{ij} ——第i个样本第j类别的预测分数

s_{y_i} ——第i个样本真实类别的预测分数

第i个样本的多类支撑向量机损失定义如下：

$$L_i = \sum_{j \neq y_i} \begin{cases} 0 & if\ s_{y_i} \geq s_{ij} + 1 \\ s_{ij} - s_{y_i} + 1 & otherwise \end{cases}$$

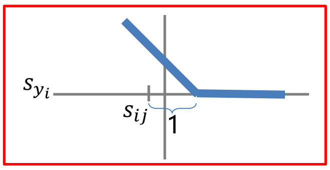
$$= \sum_{j \neq y_i} \max(0, s_{ij} - s_{y_i} + 1)$$

- 正确类别的得分比不正确类别的得分高出1分，就没有损失
- 否则，就会产生损失

2020/3/9 北京邮电大学计算机学院 鲁鹏 62

62

多类支撑向量机损失



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

j ——类别标签，取值范围{1，2，…，c}；

w_j, b_j —— 第j个类别分类器的参数；

x_i ——表示数据集中的第 i 个样本

s_{ij} ——第i个样本第j类别的预测分数

s_{y_i} ——第i个样本真实类别的预测分数

第i个样本的多类支撑向量机损失定义如下：

$$L_i = \sum_{j \neq y_i} \begin{cases} 0 & if\ s_{y_i} \geq s_{ij} + 1 \\ s_{ij} - s_{y_i} + 1 & otherwise \end{cases}$$

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

- 正确类别的得分比不正确类别的得分高出1分，就没有损失
- 否则，就会产生损失

$\max(0,\cdot)$损失——常被称为折页损失（hingeloss）

2020/3/9 北京邮电大学计算机学院 鲁鹏 63

63

示例：假设有3个类别的训练样本各一张，分类器是线性分类器 $f(x,W)=Wx+b$，其中权值W，b已知，分类器对三个样本的打分如下：

	bird	**cat**	**car**	**loss**
	**0.6**	-2.3	1.9	
	1.7	**2.9**	2.3	
	3.1	-2.6	**4.3**	

示例：假设有3个类别的训练样本各一张，分类器是线性分类器 $f(x,W)=Wx+b$，其中权值W，b已知，分类器对三个样本的打分如下：

	bird	**cat**	**car**	**loss**
	**0.6**	-2.3	1.9	2.3
	1.7	**2.9**	2.3	0.4
	3.1	-2.6	**4.3**	0

当前分类器对于鸟这张图像的损失：

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

= max(0, -2.3 – 0.6 + 1)

+max(0, 1.9 – 0.6 + 1)

= max(0, -1.9) + max(0, 2.3)

= 0 + 2.3

= 2.3

示例：假设有3个类别的训练样本各一张，分类器是线性分类器 $f(x,W)=Wx+b$，其中权值W，b已知，分类器对三个样本的打分如下：

	bird	cat	car	loss
	**0.6**	-2.3	1.9	2.3
	1.7	**2.9**	2.3	0.4
	3.1	-2.6	**4.3**	0

当前分类器对于鸟这张图像的损失：

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

= max(0, 1.7 – 2.9 + 1)

+max(0, 2.3 – 2.9 + 1)

= max(0, -0.2) + max(0, 0.4)

= 0 + 0.4

= 0.4

示例：假设有3个类别的训练样本各一张，分类器是线性分类器 $f(x,W)=Wx+b$，其中权值W，b已知，分类器对三个样本的打分如下：

	bird	cat	car	loss
	**0.6**	-2.3	1.9	2.3
	1.7	**2.9**	2.3	0.4
	3.1	-2.6	**4.3**	0

当前分类器对于鸟这张图像的损失：

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

= max(0, 3.1 – 4.3 + 1)

+max(0, -2.6 – 4.3 + 1)

= 0

示例：假设有3个类别的训练样本各一张，分类器是线性分类器 $f(x, W) = Wx + b$，其中权值W，b已知，分类器对三个样本的打分如下：

	bird	cat	car	loss
	**0.6**	-2.3	1.9	2.3
	1.7	**2.9**	2.3	0.4
	3.1	-2.6	**4.3**	0

当前分类器对于整个数据集图像的损失：

$$L = \frac{1}{N}\sum_{i=1}^{N} L_i$$ （整个数据集损失的平均值）

L = (2.3 + 0.4 + 0)/3 = **0.9**

问题抢答

- 损失函数：

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

- 单样本的多类支撑向量机损失：

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

- 线性分类器：

$$s_{ij} = w_j^T x_i + b_j$$

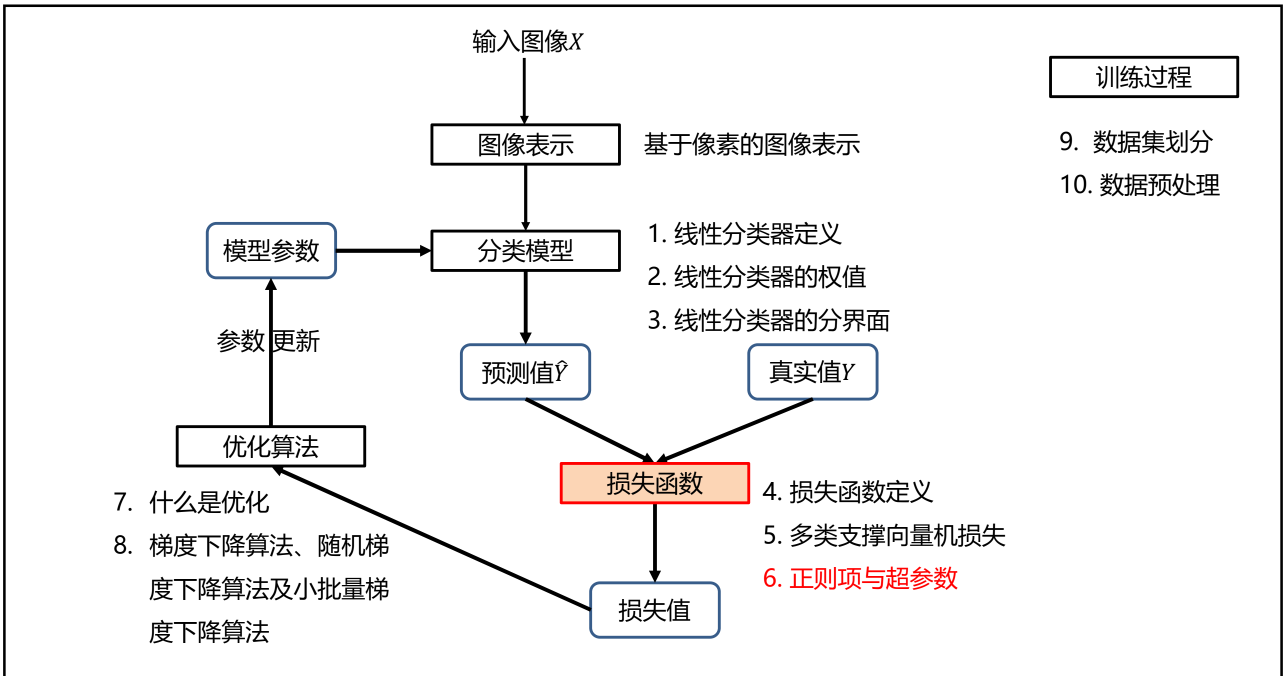
1:多类支撑向量机损失L_i的最大/最小值会是多少？

2:如果初始化时w和b很小，损失L会是多少？

3: 考虑所有类别（包括 $j = y_i$），损失 L_i 会有什么变化？

4: 在总损失L计算时，如果用求和代替平均？

5: 如果使用$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)^2$



70

再谈损失函数

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

问题：假设存在一个W使损失函数L=0，这个W是唯一的吗？

71

再谈损失函数

示例：假设两个线性器分类$f_1(x, W_1) = W_1 x$，$f_2\,(x, W_2) = W_2 x$，其中，$W_2 = 2\,W_1$。对于下面图像，已知分类器1的打分结果（如表所示），请计算分类器2的打分结果，以及两个分类器对当前样本的多类支撑向量机损失：

	鸟	猫	汽车	损失
分类器1	3.1	-2.6	4.3	?
分类器2	?	?	?	?

第i个样本的多类支撑向量机损失：

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

再谈损失函数

示例：假设两个线性器分类$f_1(x, W_1) = W_1 x$，$f_2\,(x, W_2) = W_2 x$，其中，$W_2 = 2\,W_1$。对于下面图像，已知分类器1的打分结果（如表所示），请计算分类器2的打分结果，以及两个分类器对当前样本的多类支撑向量机损失：

	鸟	猫	汽车	损失
分类器1	3.1	-2.6	4.3	?
分类器2	6.2	-5.2	8.6	?

第i个样本的多类支撑向量机损失：

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

再谈损失函数

示例：假设两个线性器分类 $f_1(x, W_1) = W_1 x$ ，$f_2(x, W_2) = W_2 x$ ，其中， $W_2 = 2W_1$ 。对于下面图像，已知分类器1的打分结果（如表所示），请计算分类器2的打分结果，以及两个分类器对当前样本的多类支撑向量机损失：

	鸟	猫	汽车	损失
分类器1	3.1	-2.6	4.3	0
分类器2	6.2	-5.2	8.6	?

第i个样本的多类支撑向量机损失：

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

分类器1损失：

= max(0, 3.1 – 4.3 + 1)

+max(0, −2.6 – 4.3 + 1)

= max(0, −0.2) + max(0,−5.9)

= 0

再谈损失函数

示例：假设两个线性器分类 $f_1(x, W_1) = W_1 x$ ，$f_2(x, W_2) = W_2 x$ ，其中， $W_2 = 2W_1$ 。对于下面图像，已知分类器1的打分结果（如表所示），请计算分类器2的打分结果，以及两个分类器对当前样本的多类支撑向量机损失：

	鸟	猫	汽车	损失
分类器1	3.1	-2.6	4.3	0
分类器2	6.2	-5.2	8.6	0

第i个样本的多类支撑向量机损失：

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

分类器1损失：

= max(0, 3.1 – 4.3 + 1)

+max(0, −2.6 – 4.3 + 1)

= max(0, −0.2) + max(0,−5.9)

= 0

分类器2损失:

= max(0, 6.2 – 8.6 + 1)

+max(0, -5.2 - 8.6 + 1)

= max(0, -1.4) + max(0, -12.8)

= 0

再谈损失函数

$$L(W) = \frac{1}{N}\sum_i L_i\left(f(x_i, W), y_i\right)$$

问题：假设存在一个W使损失函数L=0，这个W是唯一的吗?

答：不唯一，因为$\boldsymbol{W_2}$同样有L = 0

应该如何在$\boldsymbol{W_1}$和$\boldsymbol{W_2}$之间做出选择?

正则项

$$L(W) = \frac{1}{N}\sum_i L_i\left(f(x_i, W), y_i\right) + \lambda R(W)$$

数据损失

$$L(W) = \underbrace{\frac{1}{N}\sum_i L_i\left(f(x_i, W), y_i\right)} + \lambda R(W)$$

数据损失: 模型预测需

要和训练集相匹配

正则项损失

$$L(W) = \underbrace{\frac{1}{N}\sum_i L_i\left(f(x_i, W), y_i\right)} + \underbrace{\lambda R(W)}$$

数据损失: 模型预测需

要和训练集相匹配

正则损失：防止模型在训

练集上学习得“太好”。

正则项损失

$$L(W) = \underbrace{\frac{1}{N}\sum_i L_i\left(f(x_i, W), y_i\right)}_{\text{数据损失: 模型预测需要和训练集相匹配}} + \underbrace{\lambda R(W)}_{\text{正则损失：防止模型在训练集上学习得“太好”。}}$$

- $R(W)$ 是一个与权值有关，跟图像数据无关的函数

正则项损失

$$L(W) = \underbrace{\frac{1}{N}\sum_i L_i\left(f(x_i, W), y_i\right)}_{\text{数据损失: 模型预测需要和训练集相匹配}} + \underbrace{\lambda R(W)}_{\text{正则损失：防止模型在训练集上学习得“太好”。}}$$

- $R(W)$ 是一个与权值有关，跟图像数据无关的函数
- λ 是一个超参数控制着正则损失在总损失中所占的比重

什么是超参数?

- 在开始学习过程之前设置值的参数，而不是学习得到。
- 超参数一般都会对模型性能有着重要的影响。

$$L(W) = \underbrace{\frac{1}{N}\sum_i L_i\left(f(x_i, W), y_i\right)}_{\text{数据损失: 模型预测需要和训练集相匹配}} + \underbrace{\lambda R(W)}_{\text{正则损失：防止模型在训练集上学习得“太好”。}}$$

- λ =0 优化结果仅与数据损失相关
- $\lambda = \infty$ 优化结果与数据损失无关，仅考虑权重损失。此时，系统最优解为$W = 0$。

如何设置一个好的超参数呢，这个问题我们会在后面的课程中专门探讨。

L2正则项

$$L(W) = \underbrace{\frac{1}{N}\sum_i L_i\left(f(x_i, W), y_i\right)}_{\text{数据损失: 模型预测需要和训练集相匹配}} + \underbrace{\lambda R(W)}_{\text{正则损失：防止模型在训练集上学习得“太好”。}}$$

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

L2正则项

L2损失示例

样 本: $x = [1,1,1,1]$

分类器1: $w_1 = [1,0,0,0]$

分类器2: $w_2 = [0.25,0.25,0.25,0.25]$

$$L(W) = \underbrace{\frac{1}{N}\sum_i L_i\left(f(x_i, W), y_i\right)}_{\text{数据损失: 模型预测需要和训练集相匹配}} + \underbrace{\lambda R(W)}_{\text{正则损失: 防止模型在训练集上学习得“太好”。}}$$

数据损失: 模型预测需要和训练集相匹配

正则损失：防止模型在训练集上学习得“太好”。

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

L2正则项

L2损失示例

样 本: $x = [1,1,1,1]$

分类器1: $w_1 = [1,0,0,0]$

分类器2: $w_2 = [0.25,0.25,0.25,0.25]$

$$L(W) = \frac{1}{N}\sum_i L_i\left(f(x_i, W), y_i\right) + \lambda R(W)$$

数据损失: 模型预测需要和训练集相匹配

正则损失：防止模型在训练集上学习得“太好”。

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

L2正则项

$$L(W) = \underbrace{\frac{1}{N}\sum_i L_i\left(f(x_i, W), y_i\right)}_{\text{数据损失: 模型预测需要和训练集相匹配}} + \underbrace{\lambda R(W)}_{\text{正则损失：防止模型在训练集上学习得“太好”。}}$$

数据损失: 模型预测需要和训练集相匹配

正则损失：防止模型在训练集上学习得“太好”。

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

L2损失示例

样 本: $x = [1,1,1,1]$

分类器1: $w_1 = [1,0,0,0]$

分类器2: $w_2 = [0.25,0.25,0.25,0.25]$

分类器输出: $w_1^T x = w_2^T x = 1$

L2正则项

$$L(W) = \underbrace{\frac{1}{N}\sum_i L_i\left(f(x_i, W), y_i\right)}_{\text{数据损失: 模型预测需要和训练集相匹配}} + \underbrace{\lambda R(W)}_{\text{正则损失：防止模型在训练集上学习得“太好”。}}$$

数据损失: 模型预测需要和训练集相匹配

正则损失：防止模型在训练集上学习得“太好”。

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

L2损失示例

样 本: $x = [1,1,1,1]$

分类器1: $w_1 = [1,0,0,0]$

分类器2: $w_2 = [0.25,0.25,0.25,0.25]$

分类器输出: $w_1^T x = w_2^T x = 1$

正则损失:

$\mathrm{R}(w_1) = 1$ $\mathrm{R}(w_2) = 0.25$

L2正则项

$$L(W) = \underbrace{\frac{1}{N}\sum_i L_i\left(f(x_i, W), y_i\right)}_{\text{数据损失: 模型预测需要和训练集相匹配}} + \underbrace{\lambda R(W)}_{\text{正则损失：防止模型在训练集上学习得“太好”。}}$$

数据损失: 模型预测需要和训练集相匹配

正则损失：防止模型在训练集上学习得“太好”。

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

L2损失示例

样 本: $x = [1,1,1,1]$

分类器1: $w_1 = [1,0,0,0]$

分类器2: $w_2 = [0.25,0.25,0.25,0.25]$

分类器输出: $w_1^T x = w_2^T x = 1$

正则损失:

$R(w_1) = 1$ $R(w_2) = 0.25$

w_2总损失小

L2正则项

$$L(W) = \underbrace{\frac{1}{N}\sum_i L_i\left(f(x_i, W), y_i\right)}_{\text{数据损失: 模型预测需要和训练集相匹配}} + \underbrace{\lambda R(W)}_{\text{正则损失：防止模型在训练集上学习得“太好”。}}$$

数据损失: 模型预测需要和训练集相匹配

正则损失：防止模型在训练集上学习得“太好”。

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

L2损失示例

样 本: $x = [1,1,1,1]$

分类器1: $w_1 = [1,0,0,0]$

分类器2: $w_2 = \boxed{[0.25,0.25,0.25,0.25]}$

分类器输出: $w_1^T x = w_2^T x = 1$

正则损失:

$R(w_1) = 1$ $R(w_2) = 0.25$

w_2总损失小

L2正则项

L2损失示例

$$L(W) = \underbrace{\frac{1}{N}\sum_i L_i\left(f(x_i, W), y_i\right)}_{\text{数据损失: 模型预测需要和训练集相匹配}} + \underbrace{\lambda R(W)}_{\text{正则损失：防止模型在训练集上学习得“太好”。}}$$

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

L2正则损失对大数值权值进行惩罚，喜欢分散权值，鼓励分类器将所有维度的特征都用起来，而不是强烈的依赖其中少数几维特征

样 本: $x = [1,1,1,1]$

分类器1: $w_1 = [1,0,0,0]$

分类器2: $w_2 = \boxed{[0.25,0.25,0.25,0.25]}$

分类器输出: $w_1^T x = w_2^T x = 1$

正则损失:

$R(w_1) = 1$ $R(w_2) = 0.25$

w_2总损失小

L2正则项

L2损失示例

$$L(W) = \underbrace{\frac{1}{N}\sum_i L_i\left(f(x_i, W), y_i\right)}_{\text{数据损失: 模型预测需要和训练集相匹配}} + \underbrace{\lambda R(W)}_{\text{正则损失：防止模型在训练集上学习得“太好”。}}$$

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

L2正则损失对大数值权值进行惩罚，喜欢分散权值，鼓励分类器将所有维度的特征都用起来，而不是强烈的依赖其中少数几维特征

样 本: $x = [1,1,1,1]$

分类器1: $w_1 = [1,0,0,0]$

分类器2: $w_2 = \boxed{[0.25,0.25,0.25,0.25]}$

分类器输出: $w_1^T x = w_2^T x = 1$

正则损失:

$R(w_1) = 1$ $R(w_2) = 0.25$

w_2总损失小

正则项让模型有了偏好！！！

常用的正则项损失

$$L(W)=\underbrace{\frac{1}{N}\sum_i L_i\left(f(x_i,W),y_i\right)}_{\text{数据损失: 模型预测需要和训练集相匹配}}+\underbrace{\lambda R(W)}_{\text{正则损失：防止模型在训练集上学习得“太好”。}}$$

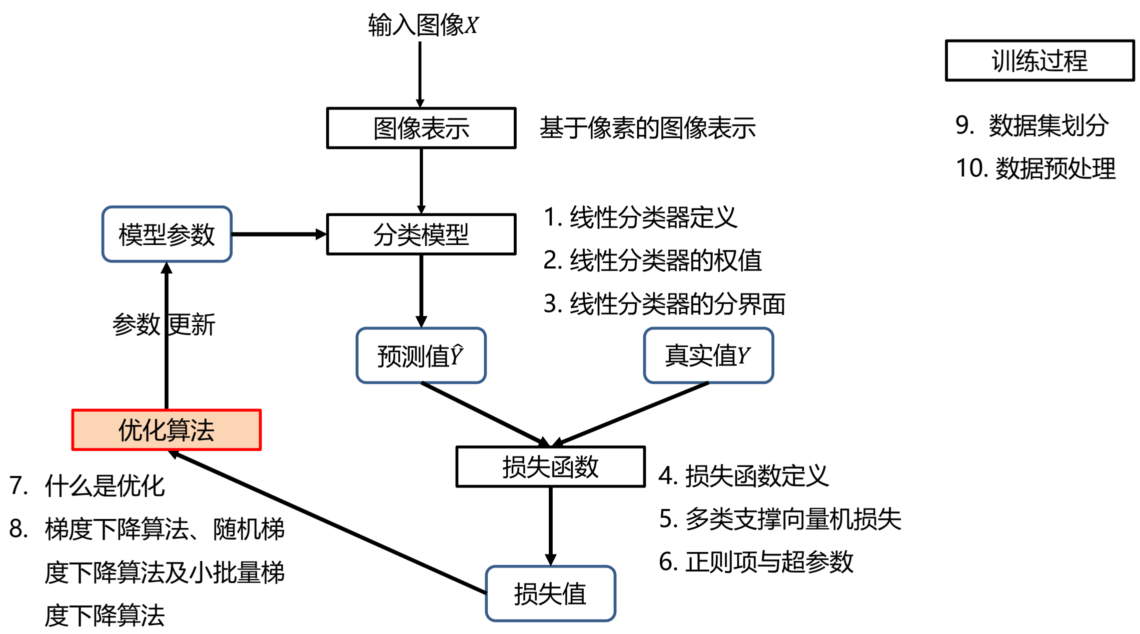
数据损失: 模型预测需要和训练集相匹配

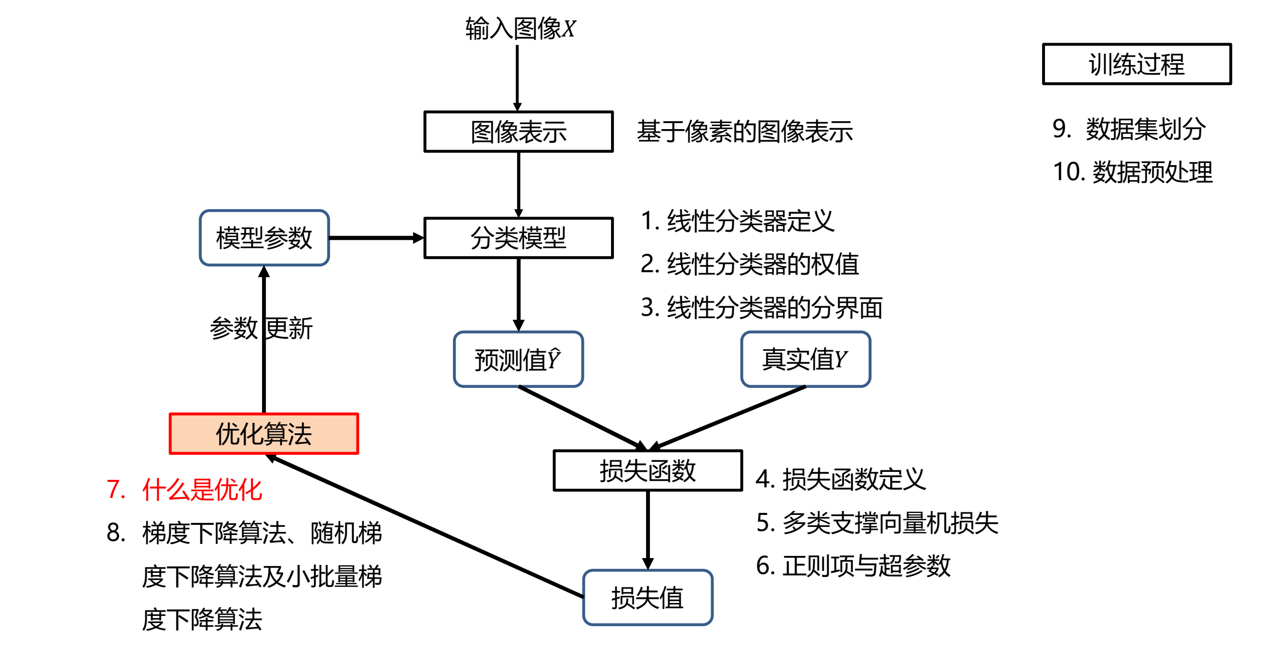
正则损失：防止模型在训练集上学习得“太好”。

L1正则项: $R(W)=\sum_k\sum_l |W_{k,l}|$

L2正则项: $R(W)=\sum_k\sum_l W_{k,l}^2$

Elastic net(L1+L2): $R(W)=\sum_k\sum_l \beta W_{k,l}^2+|W_{k,l}|$





94

什么是参数优化?

参数优化是机器学习的核心步骤之一，它利用损失函数的输出值作为反馈信号来调整分类器参数，以提升分类器对训练样本的预测性能。

95

优化算法目标

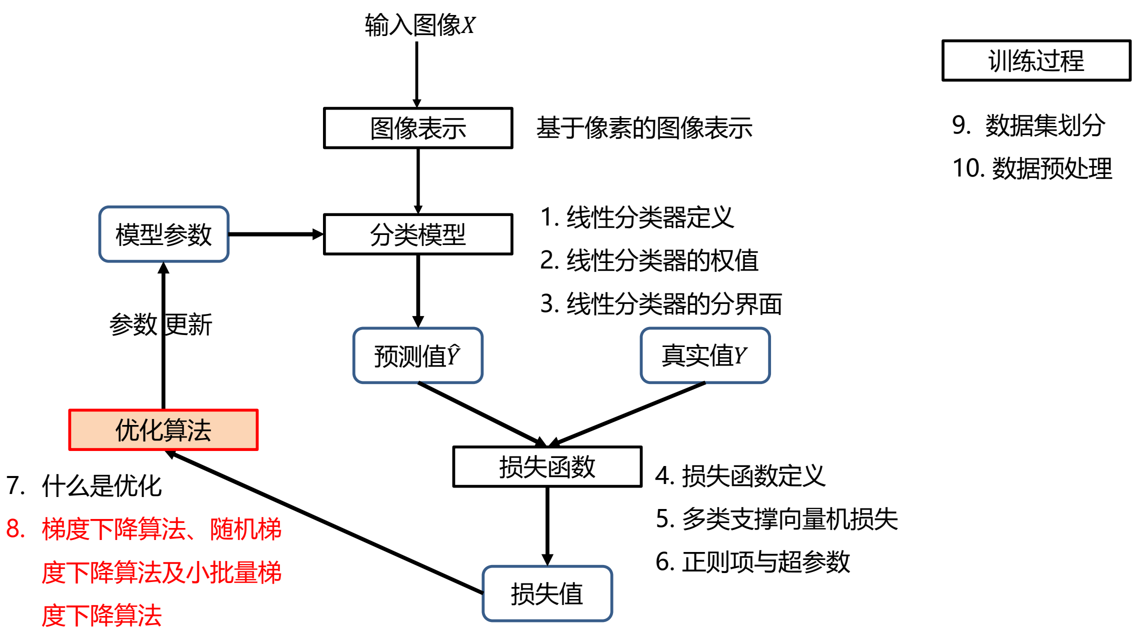
损失函数 $L = \frac{1}{N}\sum_{i=1}^{N} L_i + \lambda R(W)$

损失函数L是一个与参数W有关的函数，优化的目标就是找到使损失函数L达到最优的那组参数W 。

直接方法：

$$\frac{\partial \boldsymbol{L}}{\partial \boldsymbol{W}} = \mathbf{0}$$

通常，L形式比较复杂，很难从这个等式直接求解出 W!



梯度下降算法

一种简单而高效的迭代优化方法!

梯度下降算法

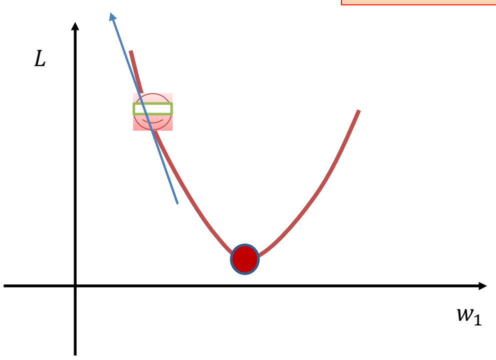
$$L(W) = \frac{1}{N}\sum_i L_i\left(f(x_i, W), y_i\right) + \lambda R(W)$$

- 往哪儿走?

  答：负梯度方法

- 走多远?

  答：步长来决定



梯度下降算法

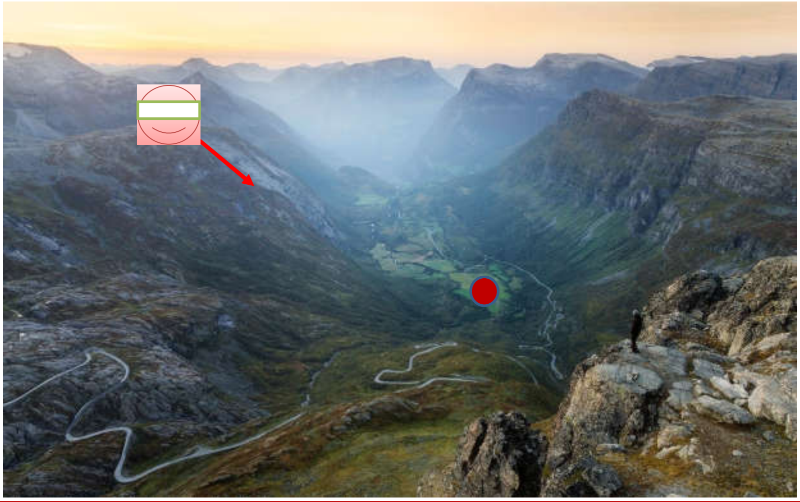
$$L(W) = \frac{1}{N}\sum_i L_i(f(x_i, W), y_i) + \lambda R(W)$$

- 往哪儿走?

  答：负梯度方法

- 走多远?

  答：步长来决定



100

梯度下降算法

梯度下降：利用所有样本计算损失并更新梯度

```
while True
    权值的梯度 ← 计算梯度(损失, 训练样本, 权值)
    权值  ← 权值 – 学习率 * 权值的梯度
```


101

梯度下降算法

梯度下降：利用所有样本计算损失并更新梯度

while True
　权值的梯度 ← 计算梯度(损失, 训练样本, 权值)
　权值 ← 权值 – 学习率 * 权值的梯度

梯度下降算法

梯度下降：利用所有样本计算损失并更新梯度

while True
　权值的梯度 ← 计算梯度(损失, 训练样本, 权值)
　权值 ← 权值 – 学习率 * 权值的梯度

梯度下降算法

梯度下降：利用所有样本计算损失并更新梯度

```
while True
    权值的梯度 ← 计算梯度(损失, 训练样本, 权值)
    权值  ← 权值 – 学习率 * 权值的梯度
```

更新后的权值

当前的权值

梯度下降算法

梯度下降：利用所有样本计算损失并更新梯度

```
while True
    权值的梯度 ← 计算梯度(损失, 训练样本, 权值)
    权值  ← 权值 – 学习率 * 权值的梯度
```

更新后的权值

当前的权值

梯度下降算法

梯度下降：利用所有样本计算损失并更新梯度

```
while True
    权值的梯度 ← 计算梯度(损失, 训练样本, 权值)  ?
    权值  ← 权值 – 学习率 * 权值的梯度
```


梯度计算

1.数值法 计算量大，不精确！

一维变量，函数求导：

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

示例：损失函数 $L(w) = w^2$ 求w=1点处的梯度

$$\frac{dL(w)}{dw} = \lim_{h \to 0} \frac{L(w+h) - L(w)}{h} \approx \frac{L(1+0.0001) - L(1)}{0.0001} = 2.0001$$

梯度计算

2.解析法　精确，速度快，导数函数推导易错！

示例：损失函数 $L(w) = w^2$ 求w=1点处的梯度

$\nabla L(w) = 2w$

$\nabla_{w=1} L(w)$ =2

108

梯度计算总结

- 数值梯度: 近似, 慢, 易写
- 解析梯度: 精确, 快, 易错

数值梯度有什么作用?

答：求梯度时一般使用解析梯度，而数值梯度主要用于解析梯度的正确性校验（梯度检查）。

109

作业：梯度计算

如何计算多类支撑向量机损失的导数函数?

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

$$s_{ij} = w_j^T x_i + b_j$$

$$L_i = \sum_{j \neq y_i} \max(0, w_j^T x_i + b_j - (w_{y_i}^T x_i + b_{y_i}) + 1)$$

梯度下降算法的计算效率

梯度下降：利用所有样本计算损失并更新梯度

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

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

while True
    权值的梯度 ← 计算梯度(损失, 训练样本, 权值)
    权值 ← 权值 – 学习率 * 权值的梯度

当N很大时，权值的梯度计算量很大!

随机梯度下降算法

随机梯度下降: 每次随机选择一个样本x_i,计算损失并更新梯度

$$L(W) = L_i(x_i, y_i, W) + \lambda R(W)$$

$$\nabla_W L(W) = \nabla_W L_i(x_i, y_i, W) + \lambda \nabla_W R(W)$$

```
while True
   数据 ← 从训练数据采样(训练数据,1)
   权值的梯度 ← 计算梯度(损失, 数据, 权值)
   权值 ← 权值 – 学习率 * 权值的梯度
```

单个样本的训练可能会带来很多噪声，不是每次迭代都向着整体最优化方向，

小批量梯度下降算法

小批量随机梯度下降: 每次随机选择m（批量的大小）个样本,计算损失并更新梯度

超参数

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

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

```
while True
   数据 ← 从训练数据采样(训练数据,批量大小)
   权值的梯度 ← 计算梯度(损失, 数据, 权值)
   权值 ← 权值 – 学习率 * 权值的梯度
```

- iteration：表示1次迭代，每次迭代更新1次网络结构的参数；
- batch-size：1次迭代所使用的样本量；
- epoch：1个epoch表示过了1遍训练集中的所有样本。

tip:通常使用2 的幂数作为批量大小，比如每次选取32或64或128个样本

总结

梯度下降方法

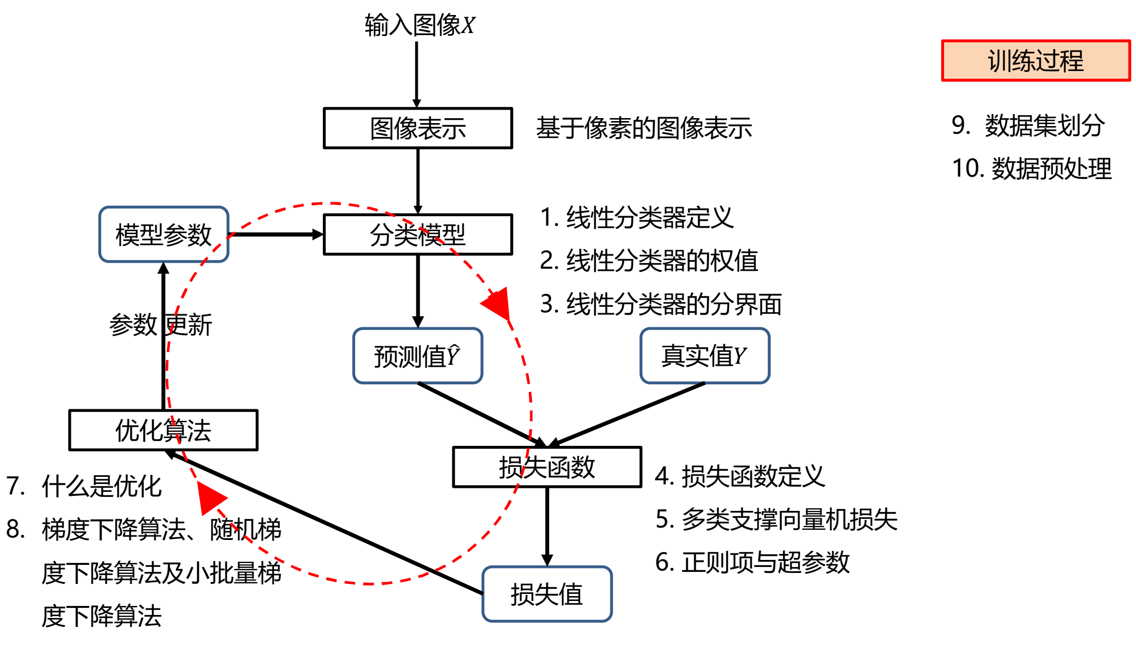
```
while True
    权值的梯度 ← 计算梯度(损失, 训练数据, 权值)
    权值 ← 权值 – 学习率 * 权值的梯度
```

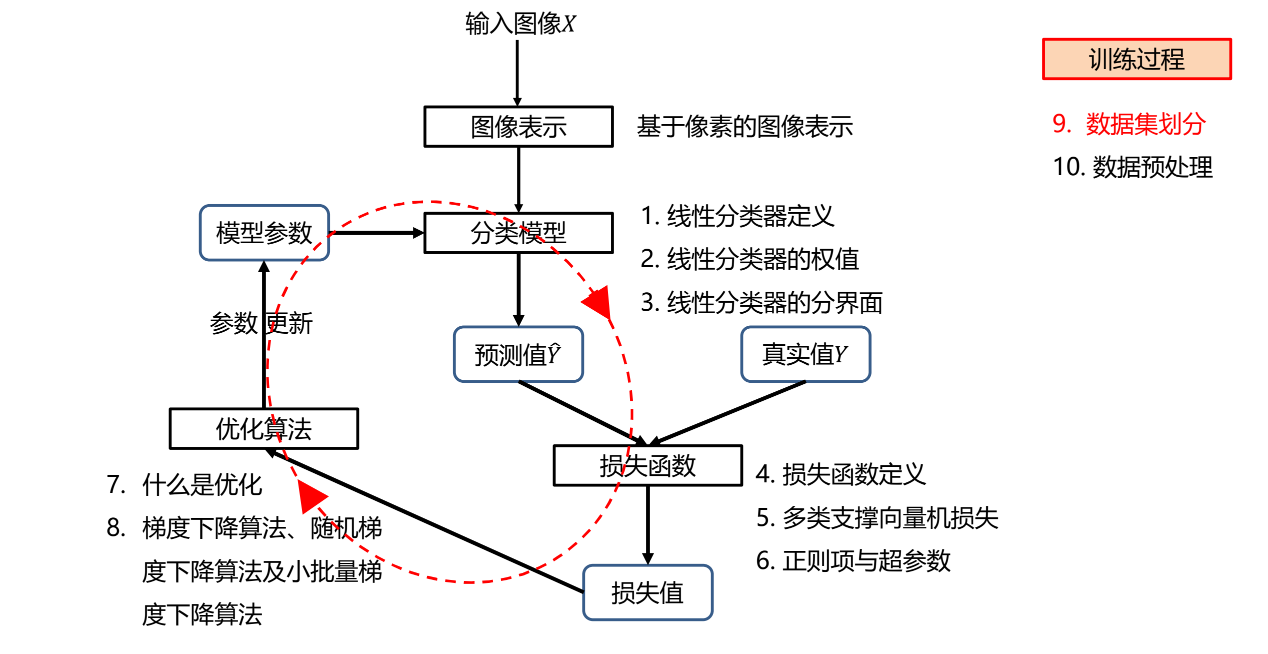
随机梯度下降算法

```
while True
    数据 ← 从训练数据采样(训练数据, 1)
    权值的梯度 ← 计算梯度(损失, 数据, 权值)
    权值  ← 权值 – 学习率 * 权值的梯度
```

小批量梯度下降算法

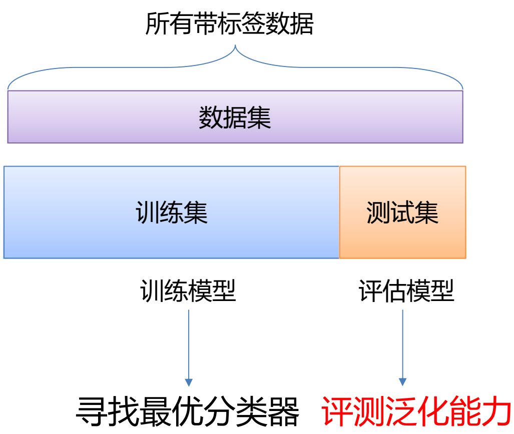
```
while True
    小批量数据 ← 从训练数据采样(训练数据, 批量大小)
    权值的梯度 ← 计算梯度(损失, 小批量数据, 权值)
    权值  ← 权值 – 学习率 * 权值的梯度
```



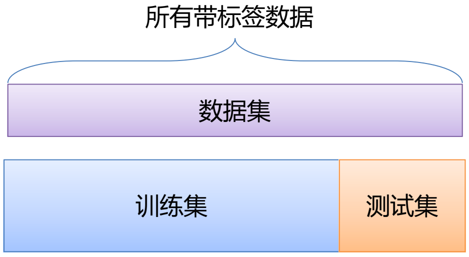
116

数据集划分



117

数据集划分

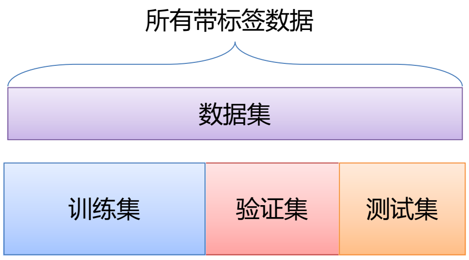


问题：如果模型含有超参数（比如正则化强度），如何找到泛化能力最好的超参数?

回答：使用验证集

训练集调超参数，测试集选超参数?

数据集划分



- 训练集用于给定的超参数时分类器参数的学习；
- 验证集用于选择超参数；
- 测试集评估泛化能力；

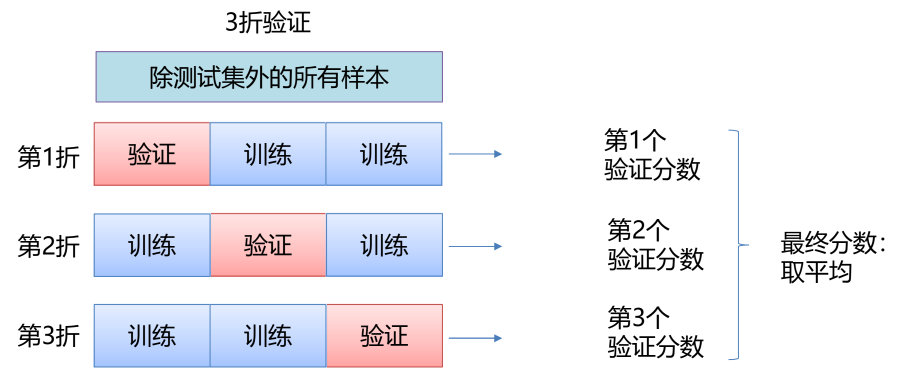
K折交叉验证

问题：如果数据很少，那么可能验证集包含的样本就太少，从而无法在统计上代表数据。

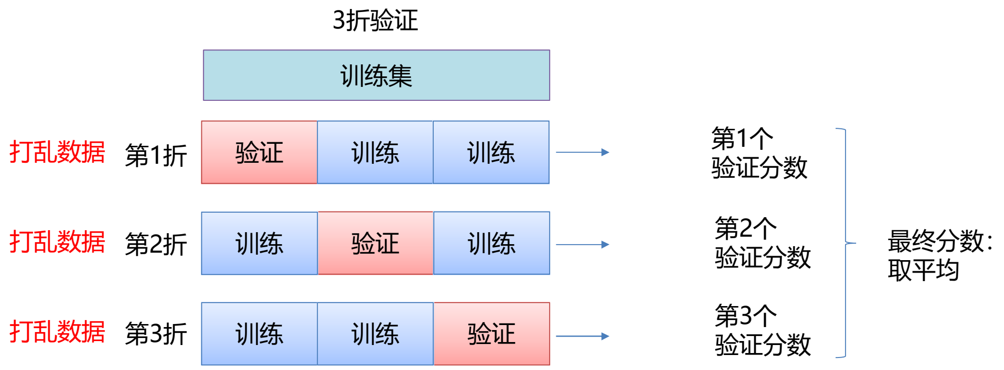
这个问题很容易发现：如果在划分数据前进行不同的随机打乱，最终得到的模型性能差别很大，那么就存在这个问题。

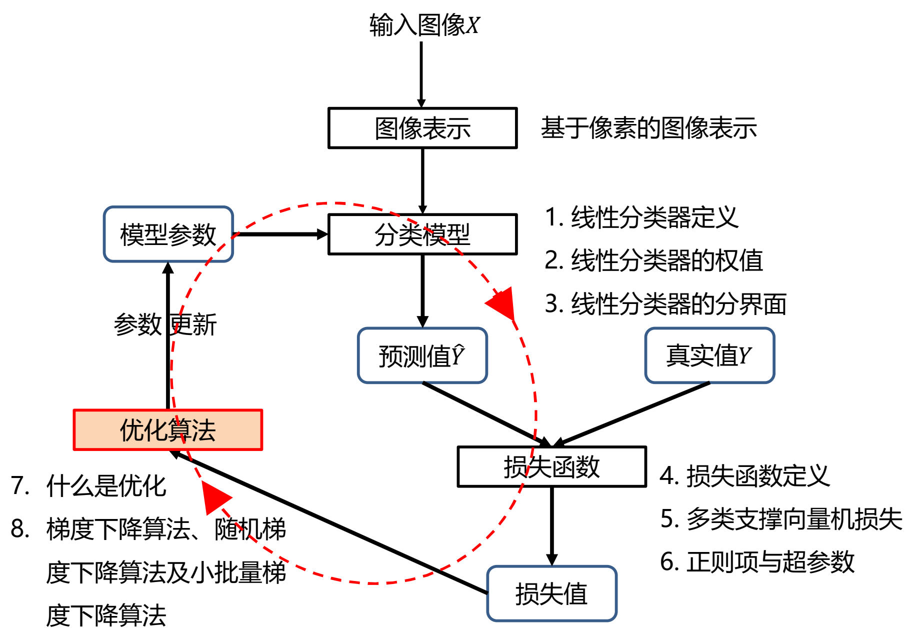
接下来会介绍*K*折验证与重复的*K*折验证，它们是解决这一问题的两种方法。

K折交叉验证



带有打乱数据的重复K折验证



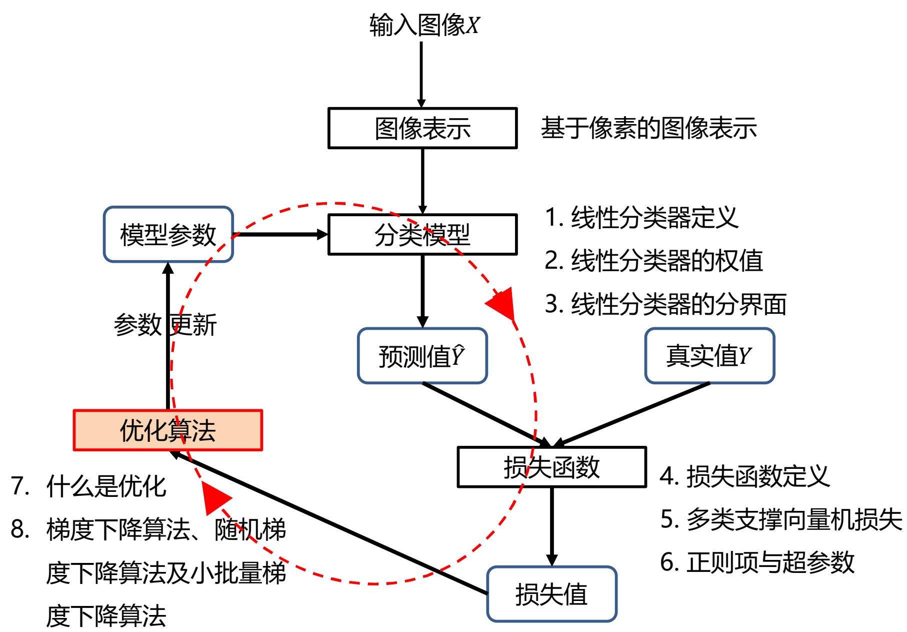


训练过程

9. 数据集划分
10. 数据预处理

数据预处理

数据能否直接使用？有哪些处理方式？

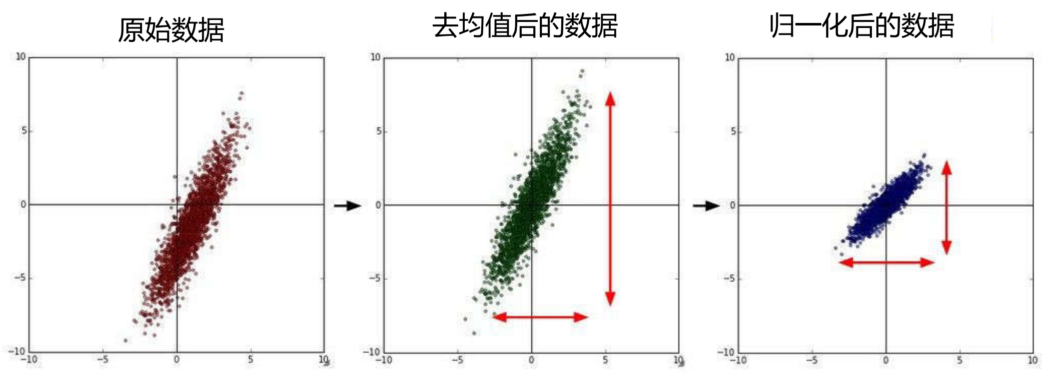


训练过程

9. 数据集划分

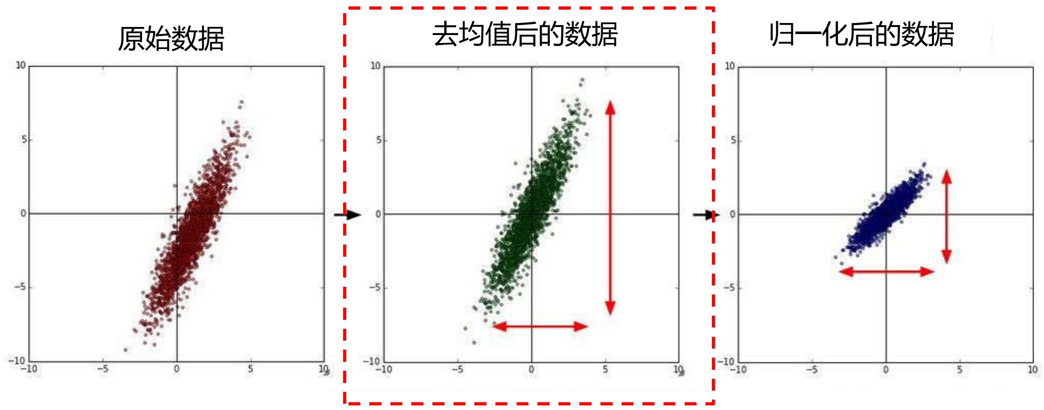
10. 数据预处理

数据预处理-1



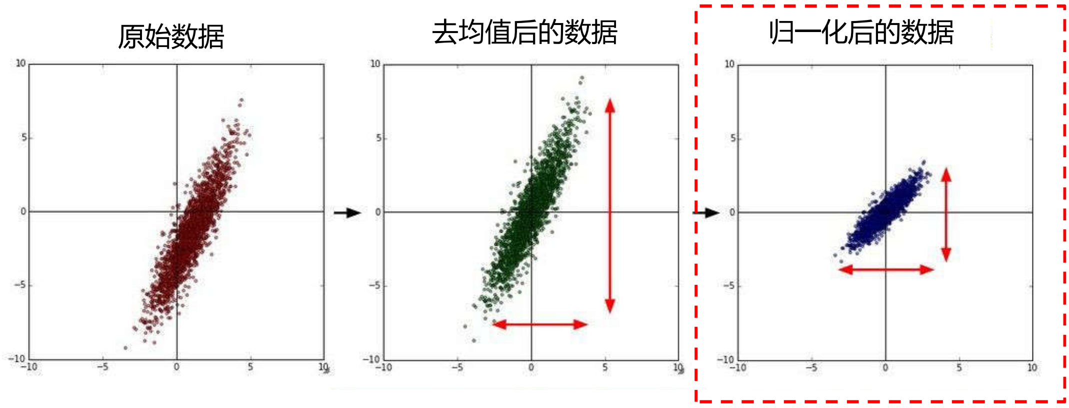
126

数据预处理-1

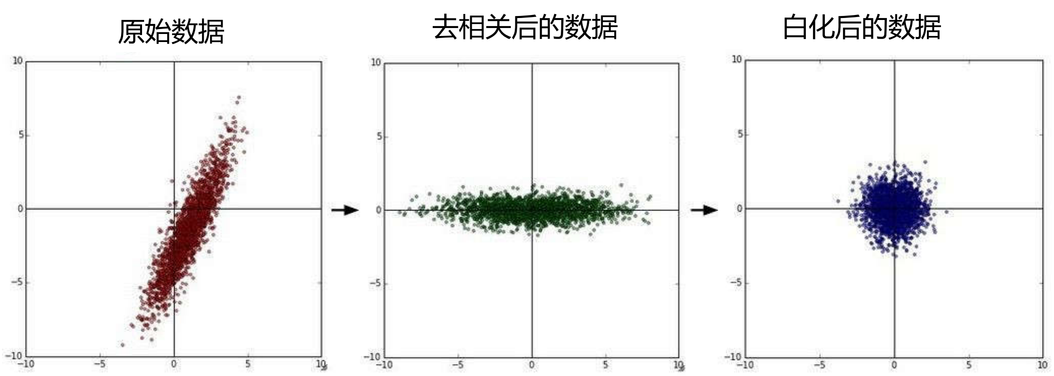


127

数据预处理-1

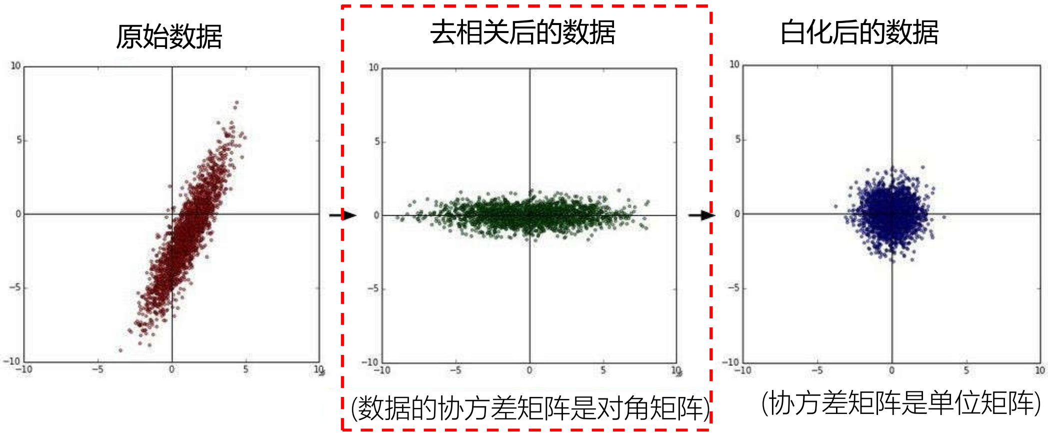


数据预处理-2



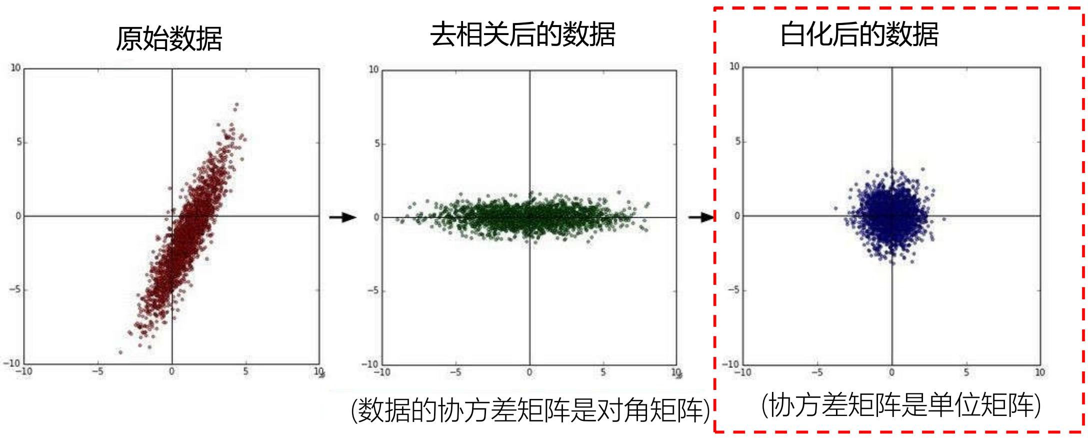
(数据的协方差矩阵是对角矩阵) (协方差矩阵是单位矩阵)

数据预处理-2

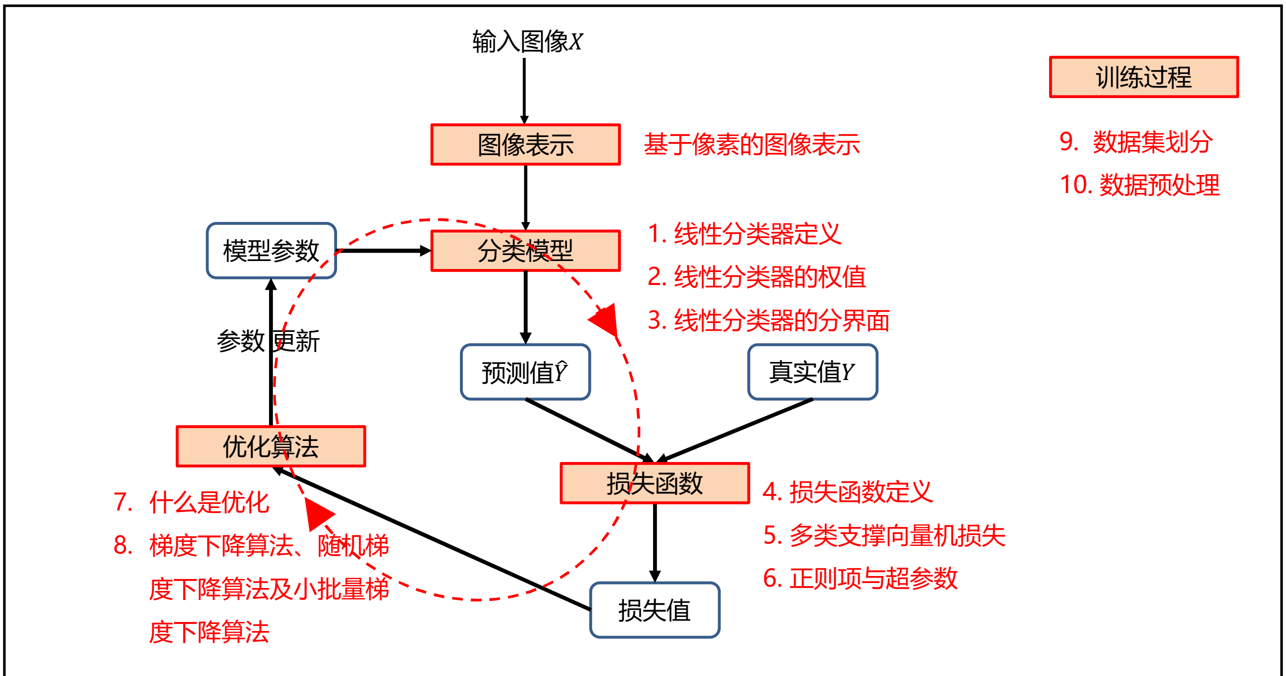


130

数据预处理-2



131



132

线性分类器体验

http://vision.stanford.edu/teaching/cs231n-demos/linear-classify/

133