目录

[生成式分类器 4](#_Toc167522514)

[贝叶斯最优分类器 4](#_Toc167522515)

[最小化错误率决策（最大后验概率） 4](#_Toc167522516)

[最小化风险决策 5](#_Toc167522517)

[引入拒识reject 6](#_Toc167522518)

[小结 6](#_Toc167522519)

[生成式分类器 7](#_Toc167522520)

[概率密度函数的参数估计 7](#_Toc167522521)

[极大似然估计 7](#_Toc167522522)

[贝叶斯估计 8](#_Toc167522523)

[常见分布的参数估计 8](#_Toc167522524)

[伯努利分布 8](#_Toc167522525)

[多项分布 9](#_Toc167522526)

[单变量高斯分布 10](#_Toc167522527)

[多元高斯分布 11](#_Toc167522528)

[朴素贝叶斯分类器（NBC） 11](#_Toc167522529)

[模型训练 11](#_Toc167522530)

[类先验分布 12](#_Toc167522531)

[类条件分布 12](#_Toc167522532)

[高斯判别分析 16](#_Toc167522533)

[假设 16](#_Toc167522534)

[判别函数 16](#_Toc167522535)

[两类 16](#_Toc167522536)

[两类且协方差矩阵相等 17](#_Toc167522537)

[模型训练 18](#_Toc167522538)

[判别式分类器 19](#_Toc167522539)

[线性判别函数 19](#_Toc167522540)

[两类分类 19](#_Toc167522541)

[多类分类 19](#_Toc167522542)

[(1)一对其他 19](#_Toc167522543)

[(2)一对一 20](#_Toc167522544)

[(3)一对一 20](#_Toc167522545)

[线性分类器的构造 20](#_Toc167522546)

[Fisher线性判别——有监督降维 21](#_Toc167522547)

[感知器 23](#_Toc167522548)

[准则函数 23](#_Toc167522549)

[批处理梯度下降法（BGD） 23](#_Toc167522550)

[随机梯度下降法（SGD） 24](#_Toc167522551)

[多类分类 24](#_Toc167522552)

[思考 25](#_Toc167522553)

[最小平方误差分类器 25](#_Toc167522554)

[SVM 25](#_Toc167522555)

[Logistic回归 25](#_Toc167522556)

[广义线性判别函数 25](#_Toc167522557)

[分段线性判别函数 25](#_Toc167522558)

[特征工程 26](#_Toc167522559)

[线性回归 27](#_Toc167522560)

[支持向量机 28](#_Toc167522561)

[统计学理论 29](#_Toc167522562)

[集成学习 30](#_Toc167522563)

[聚类 31](#_Toc167522564)

[降维 32](#_Toc167522565)

[半监督学习 33](#_Toc167522566)

[深度学习 34](#_Toc167522567)

[神经元结构 34](#_Toc167522568)

[典型神经元 34](#_Toc167522569)

[人工神经元（M-P模型） 35](#_Toc167522570)

[激活函数 35](#_Toc167522571)

[Relu变种 37](#_Toc167522572)

# 生成式分类器

* 模式分类：给定某个模式样本，确定其所属的类别（通过测量被识别对象的某些特征值，并将其作为一个判决规则的输入，按此规则来对样本进行分类）
* 确定性分类
  + 确定性现象：在获取模式的观测值时，有些事物具有确定的因果关系，即在一定的条件下，它必然会发生或必然不发生。
* 非确定性分类
  + 但在现实世界中，有许多客观现象的发生。就每一次观测来说，即使在基本条件保持不变的情况下也具有不确定性。只有在大量重复的观察下，其结果才能呈现出某种规律性，即对它们观察到的特征具有统计特性。
  + 特征的值不再是一个确定的向量，而是一个随机向量。此时，只能利用模式集合的统计特性来分类，以使分类器发生错误的概率最小

## 贝叶斯最优分类器

### 最小化错误率决策（最大后验概率）

* 希望决策的**平均错误率/平均误差概率**最小
* 如果对于每个样本，保证最小，则平均错误率就最小
* 给定观测值，判断其类别为的错误率是
* 要使错误率最小，则的概率最大。（就是找到样本最match的类别）
* 所以最小错误率决策等价于最大后验概率决策
* 根据贝叶斯规则
* 判别规则：
* 判别函数：
* 对数形式：
* 例题

地震，正常。表示出现生物异常。，。现假设观察到生物异常：

1. 直接求条件概率（后验）
2. 利用后验概率的等价形式

### 最小化风险决策

不同错误决策带来的损失可能不同（如癌症筛查） 引入**损失函数**或**代价函数**

* 损失函数：表示将本应属于类别的模式判别成属于类别的代价
* 平均损失
* 样本对应的条件风险：
* 期望风险：
* 对比平均错误率
  + 条件风险和错误率的作用相同，条件风险是错误率的推广
  + 选择对每个样本条件风险最小的分类规则，将使期望风险最小化
  + 当损失函数为0-1损失时，最小风险等价于最小错误率
* 最小化风险决策

**例题**

|  |  |  |
| --- | --- | --- |
| 课程质量 | 好（Y=1） | 差（Y=0） |
| 概率（先验） | 0.6 | 0.4 |
| P(X|Y) | 课程质量好 | 课程质量差 |
| 课堂有趣（X=1） | 0.8 | 0.1 |
| 课堂无聊（X=0） | 0.2 | 0.9 |

|  |  |  |
| --- | --- | --- |
|  | 课程质量好 | 课程质量差 |
| 选课（y=1） | 0 | 10 |
| 退课（y=0） | 20 | 0 |

* 听了一次课，觉得有趣，求最小化风险决策

退课风险远大于选课风险

### 引入拒识reject

在必要的情况下，分类器对某些样本可以拒绝给出一个输出结果，拒绝将样本判给C个类别中的任何一类

* 损失
* 拒识代价必须小于错分代价，否则永远不会对样本拒识
* 此时的条件风险为
* 引入拒识后的最小风险决策

### 小结

* 贝叶斯最优分类器：最小风险决策所决定出的贝叶斯分类器
* 相应地，风险被称为贝叶斯风险。
* 给定损失函数时，计算贝叶斯最优分类器的关键是计算后验概率

## 生成式分类器

* 计算后验概率，需要已知先验和类条件概率
* 此时亦被称为生成式分类器。因为已知 , 可得到联合分布，从而可以从联合分布通过采样生成数据
* 另一种方式是判别式分类器：直接计算后验概率或者判别函数
* 实际中，估计概率密度函数很困难。尤其是类条件概率， 因为通常是高维随机向量
* 类先验概率
* 类条件概率：由于为多维向量，条件分布建模困难，可根据数据的实际情况对其做适当假设或简化
* 朴素贝叶斯：在给定的情况下，的各维独立
* 高斯判别分析：在给定的情况下，为多元高斯分布

## 概率密度函数的参数估计

给定随机变量或随机向量的概率密度函数的形式，但其参数未知。如，但未知

估计模型参数的方法

* 矩方法
* 极大似然估计：频率学派
* 贝叶斯方法：贝叶斯学派

### 极大似然估计

令为来自分布的独立同分布的样本，定义似然函数：

似然函数在数值上是数据的联合密度，但它是参数的函数，不满足密度函数的性质（对积分不必为1）

极大似然估计是使得似然函数最大的，即

似然函数定义为似然函数的自然对数：

* 自然对数函数为单调增函数，所以和似然函数在相同的位置取极大值
* 数值计算更稳定：似然函数涉及多个小的概率值相乘，容易下溢出
* 计算更简单：很多概率密度函数是指数函数，取对数运算后更简单
* 在不引起混淆的情况下，有时记似然函数为似然函数

### 贝叶斯估计

* MLE认为参数只是一个值（点估计）
* 贝叶斯估计：：参数也是随机变量，亦可用概率分布描述其性质
  + 先验分布：在没有看到数据之前，参数的分布
    - 先验反映我们对参数取值的信念：通常偏好更简单或更光滑的模型
    - 为计算方便，我们一般采用共轭先验（先验分布与后验分布为同族分布）
  + 似然：同MLE相同，为
  + 后验分布： 在看到数据后，对参数分布的更新
    - 参数估计不再是一个点估计，而是一个分布（信息更多）
    - 也可以用后验分布的均值或众数得到参数的点估计

### 常见分布的参数估计

#### 伯努利分布

随机变量，

##### 极大似然估计

似然函数：

*令* ，求得

##### 贝叶斯估计

似然：

该似然对应的共轭先验为Beta分布：，假设

后验：

类先验的贝叶斯后验估计为：

伪先验的和称为先验的强度（先验的有效样本大小），与样本数的作用类似

参数的点估计可以取

* 最大后验估计（后验的众数，MAP）：（当时，退化成MLE）
* 后验的均值：（当时，，称为Laplace平滑）

#### 多项分布

*，*其中为指示函数，当成立时为1，否则为0

##### 极大似然估计

似然函数：

由于需要满足条件，采用拉格朗日乘子法：

分别求导：

得到：

##### 贝叶斯估计

似然：

该似然对应的共轭先验为Dirichlet分布：，假设

后验：

类先验的贝叶斯后验估计为：

参数的点估计可以取

* 最大后验估计（后验的众数，MAP）：（当所有时，MAP退化成MLE）
* 后验的均值：（当所有时，，称为Laplace平滑）

#### 单变量高斯分布

##### 极大似然估计

似然函数：

对参数求偏导：

得到：

##### 贝叶斯估计

（只讨论已知，的贝叶斯估计）

#### 多元高斯分布

##### 极大似然估计

##### 贝叶斯估计

## 朴素贝叶斯分类器（NBC）

假设有个类别，类别的先验分布

* 两类：
* 多类：

每个样本的特征为

朴素贝叶斯分类器：假设各维特征在给定类别标签的情况下条件独立

* 实际应用中即使特征条件独立的假设不严格满足，NBC性能也不错
* 因为NBC比较简单，不容易过拟合

### 模型训练

朴素贝叶斯模型的训练过程就是估计模型的参数

类先验分布：

类条件分布：

似然函数：

* 类先验分布的参数只与似然函数中第2项有关
* 类条件分布只与似然函数中第1项有关

#### 类先验分布

##### 两类分类任务

类先验分布为伯努利分布：

极大似然估计：

共轭先验，贝叶斯后验分布

点估计可取最大后验估计或后验均值估计

##### 多类分类任务

类先验分布为多项分布：

极大似然估计：

共轭先验，贝叶斯后验分布

点估计可取最大后验估计或后验均值估计

#### 类条件分布

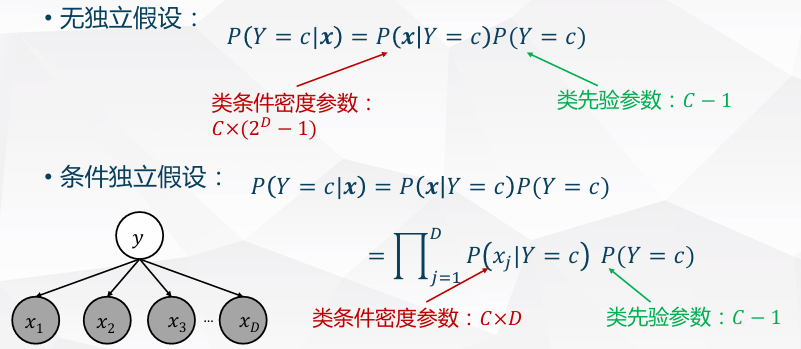
对每个类别𝑐的每维特征分别估计参数，与类先验分布的参数和其他类别、其他维特征分布的参数无关：将属于第𝑐类的样本的第𝑗维特征挑出来即可

##### 二值

特征值只有两种可能

可以用伯努利分布表示，表示在类别Y=c的情况下，特征的概率

朴素贝叶斯大大减少了模型的参数量



极大似然估计：

Laplace平滑估计：

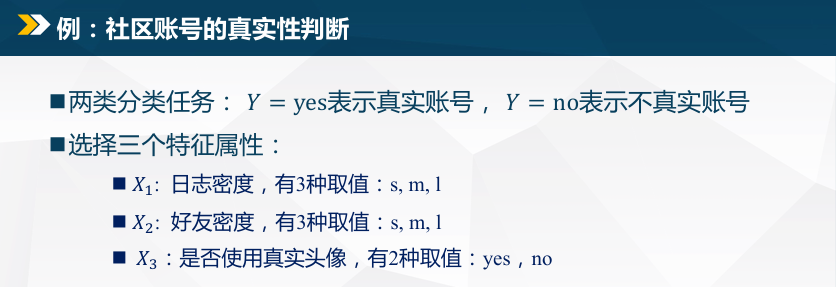
##### 多个离散值

特征值只有M种可能

可用分布表示，表示在类别Y=c的情况下，特征的概率

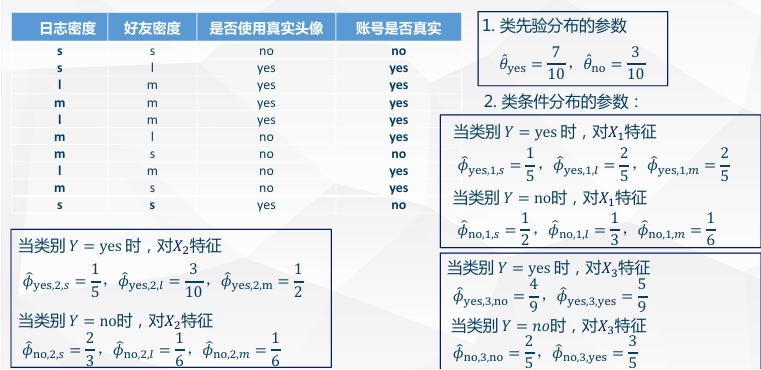
极大似然估计：

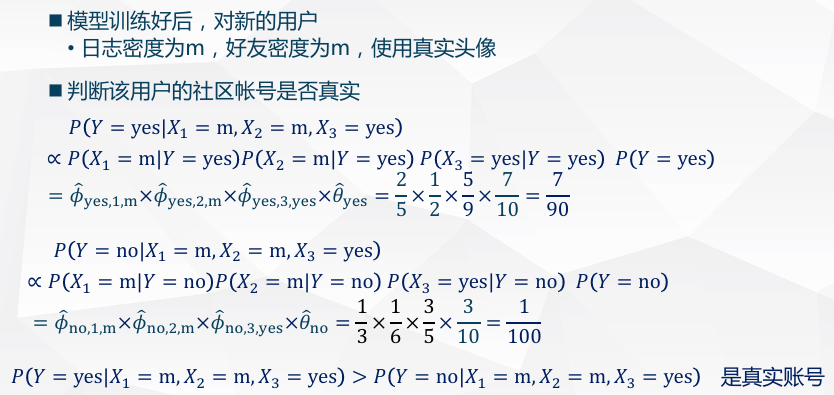
Laplace平滑估计：











##### 计数

多项分布：在多次Multinoulli试验中，每种结果出现的次数

令字典中有𝐷个单词，则对每个类别𝑐，分布参数为1个𝐷维向量：

极大似然估计：

##### 连续值

当特征取值为连续值，且在类边缘分布为高斯分布时，

注意：不是所有的连续特征都可假设为高斯分布

极大似然估计：

## 高斯判别分析

### 假设

假设：每类数据由一个多元高斯分布产生，即

类别的先验分布

* 两类：
* 多类：

根据贝叶斯公式，可计算给定特征时类别的后验概率为

### 判别函数

类别c的判别函数：

去掉与参数无关的项

#### 两类

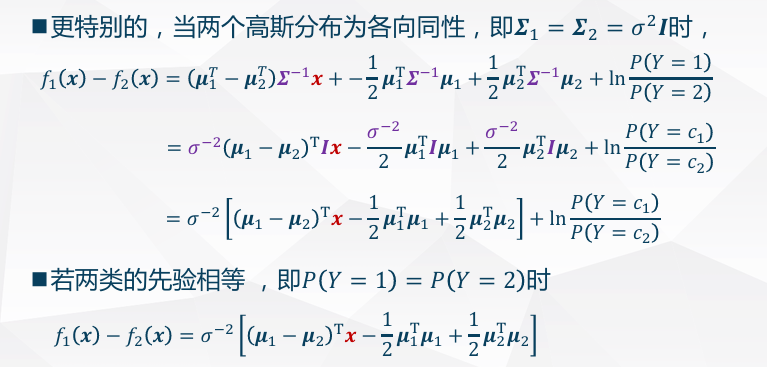
两个类别的判别函数分别为：

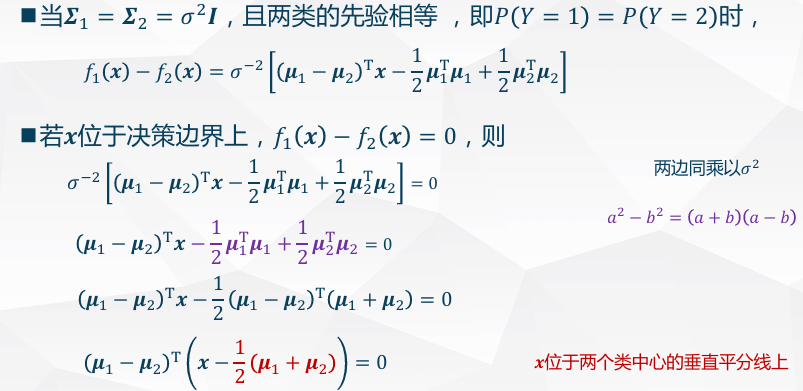
判别函数为的二次函数，因此高斯判别分析被称为二次判别分析

其中与无关

#### 两类且协方差矩阵相等

当两类的协方差矩阵时，判别函数为线性函数：（线性判别分析）





### 模型训练

* 类先验：Bernoulli分布或Multinoulli分布，同朴素贝叶斯分类器
* 类条件分布的参数：每个类别的均值向量和协方差矩阵
* 根据概率密度函数参数估计部分关于多元高斯分布参数估计的结论

# 判别式分类器

## 线性判别函数

若𝒙是二维模式样本，用和为坐标轴，得到模式的平面图。若这两类模式可用一条直线来划分：

则称则为判别函数，称为决策面/判别界面方程。

若不同类别的模式之间可用一个线性函数来划分，则这些模式是线性可分的，否则就是非线性可分的。

一旦线性判别函数的权重向量𝒘被确定，这些函数就可用作模式分类。

模型训练：根据训练样本集确定权重向量𝒘

例如：Fisher判别分析、感知器、Logistic回归、线性SVM……

### 两类分类

### 多类分类

转化为多个两类分类任务

#### (1)一对其他

用线性判别函数将属于第𝑐类的模式与不属于第𝑐类的模式分开（转化为C个两类问题），判别函数：

不确定区域：若对某一模式区域，的超过一个，或者全都，则分类失败

#### (2)一对一

用线性判别函数将任意两类（第𝑐类和第𝑘类）的模式分开（共有个判别函数），判别函数为

对，若，则

不确定区域：对任意的，都有，

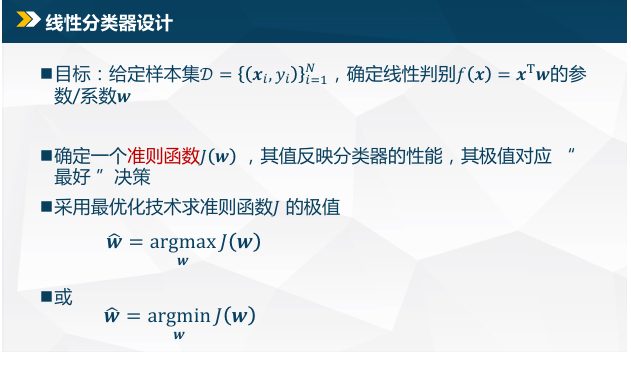
#### (3)一对一

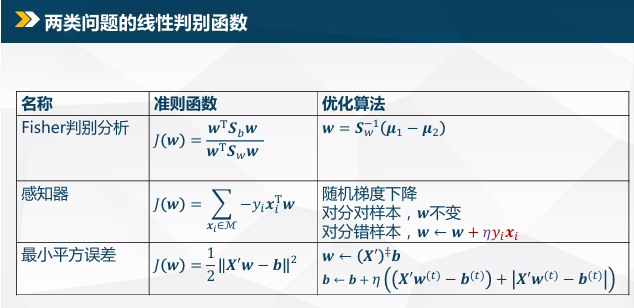
没有不确定域。对C类多类问题，共有C个判别函数

判别规则：

直接使用判别函数做差求解

## 线性分类器的构造



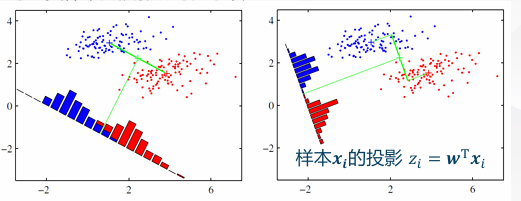


### Fisher线性判别——有监督降维

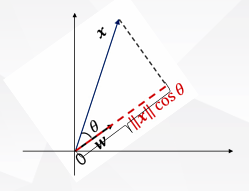
将原始𝐷维空间的样本投影到一条直线上，形成一维空间𝑍

寻找一个最合适的投影轴𝒘，使得：

* 两类样本在该轴上投影之间的距离尽可能远
* 每一类样本的投影尽可能紧凑



假设投影方向为𝒘。由于我们只关心方向，可令模长为1，𝒘为单位向量， ：



投影前X空间样本分布的描述量

* 第c类样本均值向量：
* 第c类样本类内散度矩阵：
* 总类内散度矩阵：
* 样本类间散度矩阵：

投影后Z空间样本分布的描述量

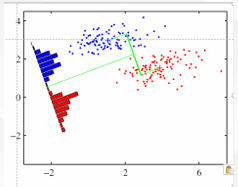
* 第c类样本均值：
* 第c类样本类内散度：
* 总类内散度：
* 样本类间散度：

两个空间之间统计量之间的关系：

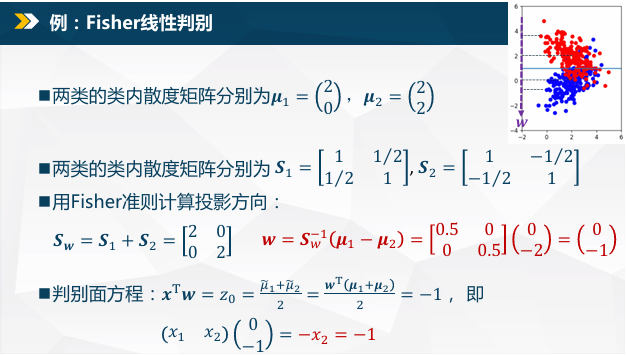
Fisher准则函数：

最佳投影方向

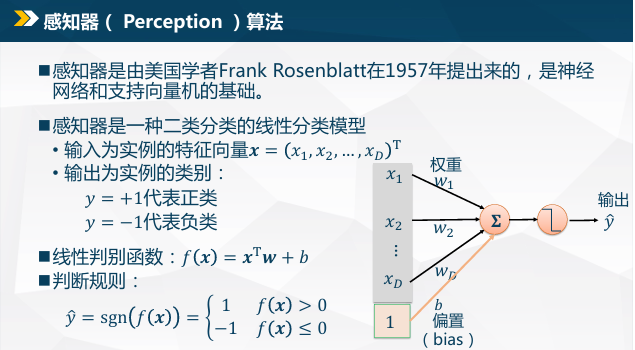
和𝒘垂直的两类的中心线为两类的判别直线



判别函数为（判别面为投影后中线的垂直平分线）



### 感知器



样本分对：

#### 准则函数

分错的样本尽可能少，令M为错误分类样本的集合

最佳

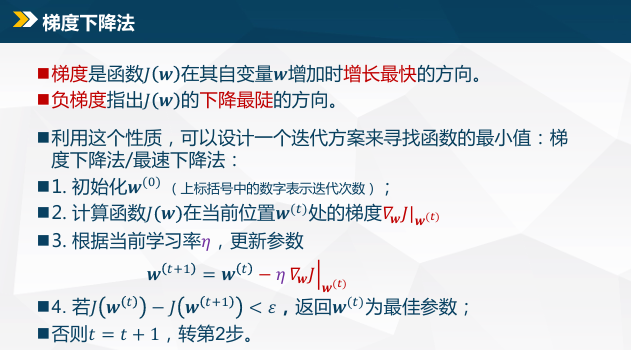
#### 批处理梯度下降法（BGD）

在梯度计算中使用成批的所有样本

将目标函数写成增广向量的形式，采用梯度下降法求解参数w

根据优化理论，函数极值点只能在边界点、不可导点或导数为0的点，其中导数为0的点称为函数的临界点。对多元函数，临界点满足所有自变量的偏导数均为0，即梯度为𝟎向量：

梯度是函数在其自变量𝒘增加时增长最快的方向。



#### 随机梯度下降法（SGD）

每次梯度计算只使用一个样本，更高效，但梯度震荡大，所有样本都用过一次称为一轮（epoch）迭代

目标函数：

梯度：

1. 随机初始化（如0向量）
2. 选择一个样本，
   * 若，样本被正确分类，梯度为0，不变
   * 若，样本被分错，梯度为
3. 对分错样本，更新模型参数
4. 如果训练集没有误分类样本，结束训练；否则转第2步

#### 多类分类

多类情况3，将感知器算法推广到多类模式

对𝐶类分类任务，存在𝐶个判别函数

若，则对，

在训练过程的第次迭代时，一个属于第𝑐类的样本送入分类器，计算出𝐶个判别函数：

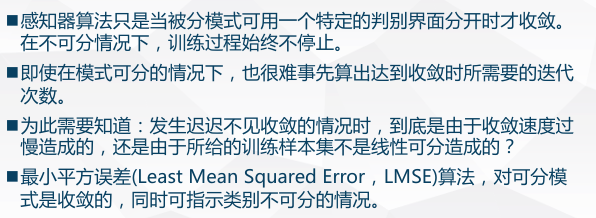
若对对，成立，分类正确，权向量不变；

若其中第个权向量使得，则相应的权向量应做调整：

思考：感知器算法存在许多解？

* 初值的选择
* 迭代过程中误分类点的选择顺序

### 最小平方误差分类器



两类分类，实例的特征向量，标签代表正类，代表负类

线性判别函数：

分类正确：

对N个训练样本，

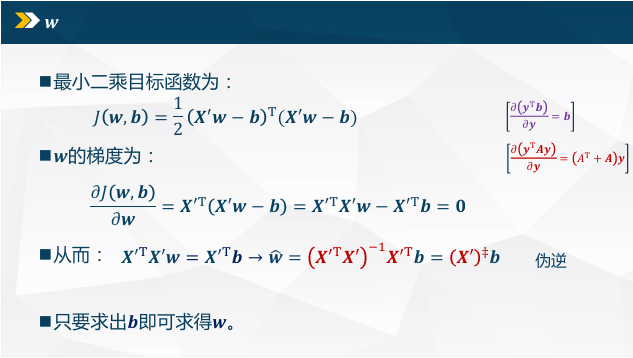
如果所有训练样本分类正确，有

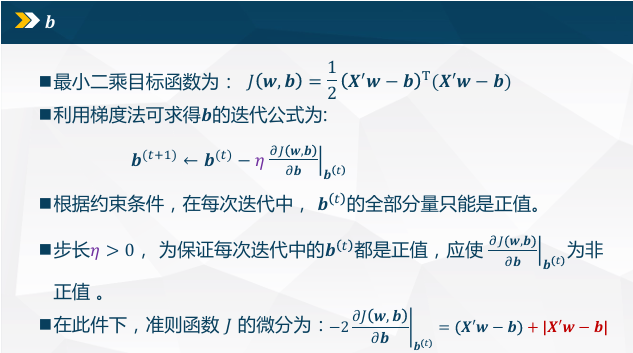
将不等式组的求解转换为对线性方程组，其中 ,

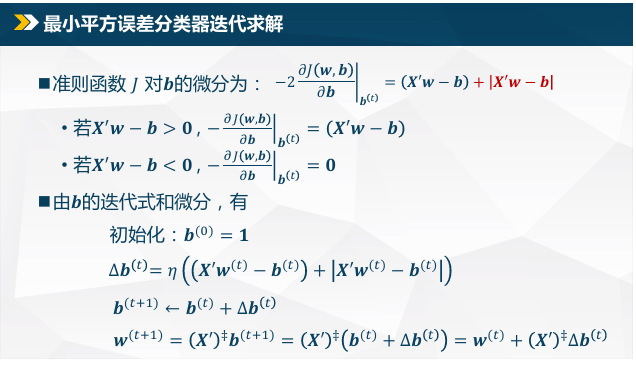
𝑿′不是𝑁×𝑁的方阵，通常是行数（𝑁）多于列数（𝐷+1）的长方阵，所以𝑿′𝒘=𝒃是超定方程，可求其最小二乘解

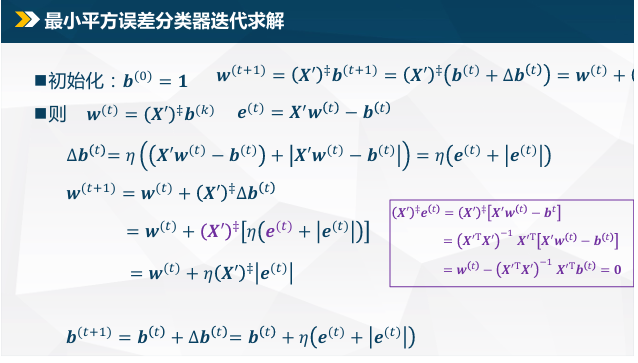
设的最小二乘解为，因此取极小

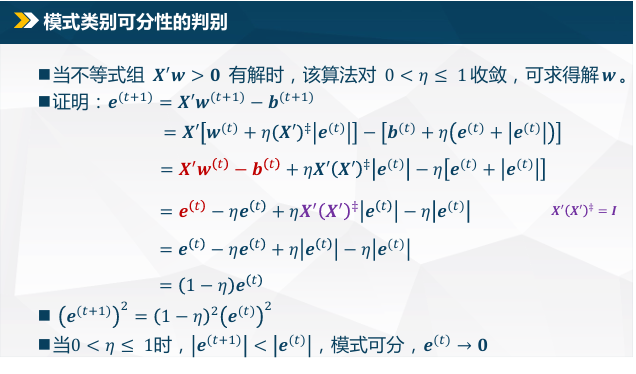
准则函数：

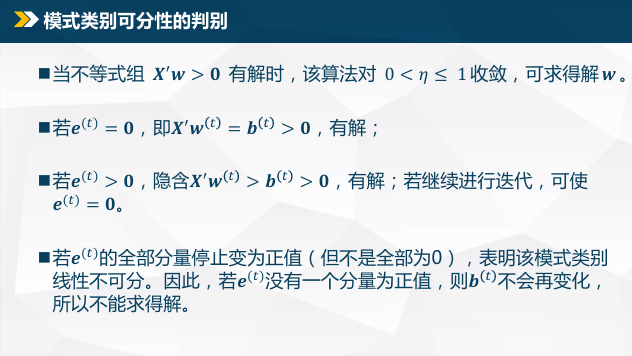


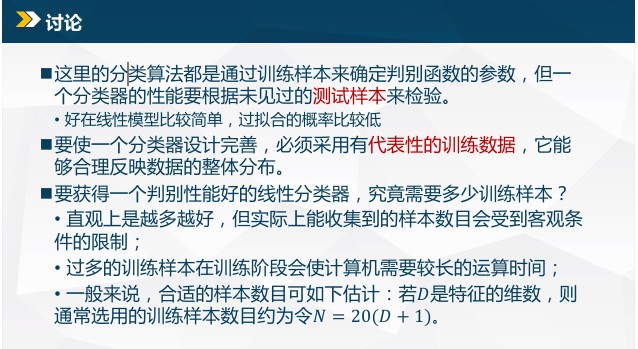












### SVM

### Logistic回归

## 广义线性判别函数

## 分段线性判别函数

# 特征工程

# 线性回归

# 支持向量机

# 统计学理论

# 集成学习

# 聚类

# 降维

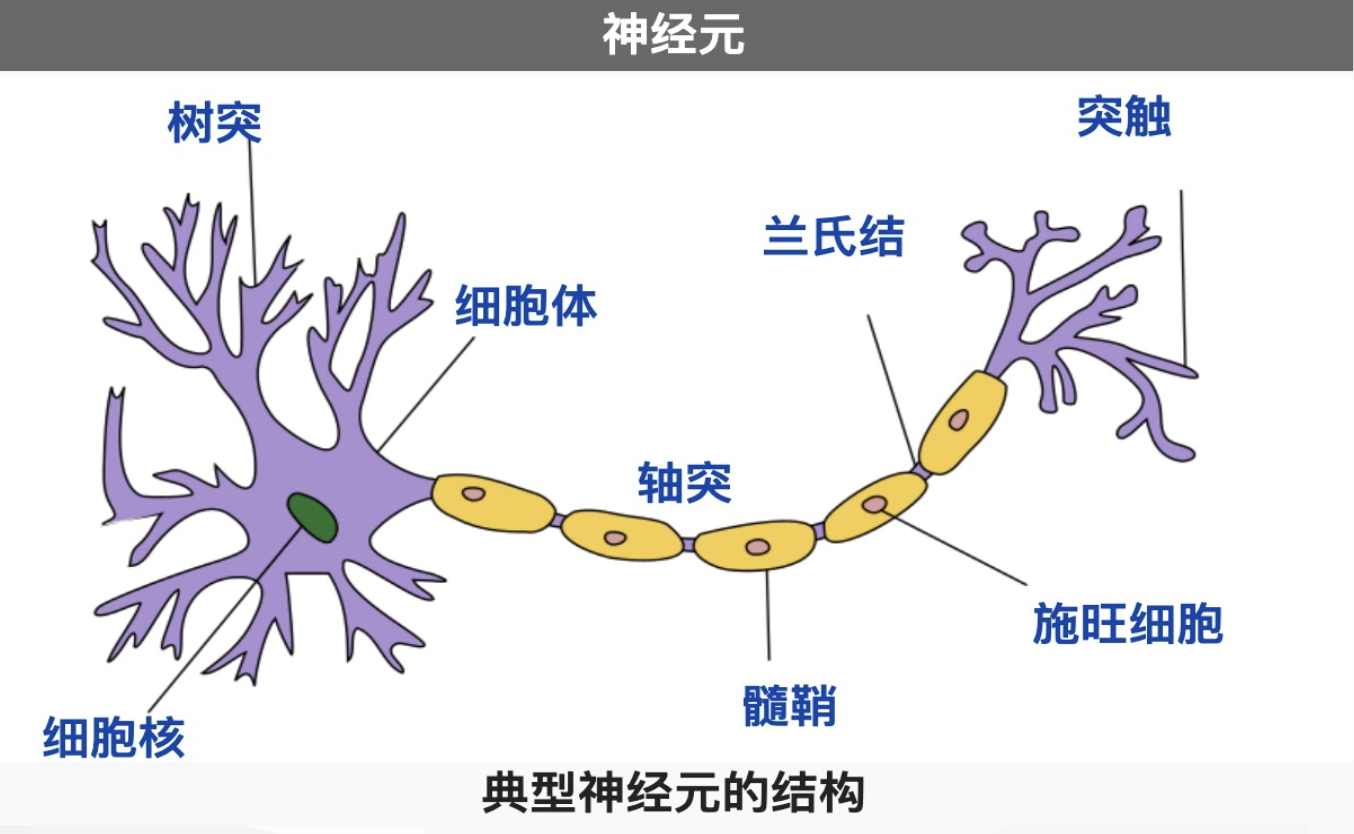
# 半监督学习

# 深度学习

## 神经元结构

### 典型神经元

* 神经细胞之间通过突触连接，突触通过复杂精巧的电化学过程传递信息
* 接收前面神经元的输入，汇总决策传递



### 人工神经元（M-P模型）

* 权重、偏置

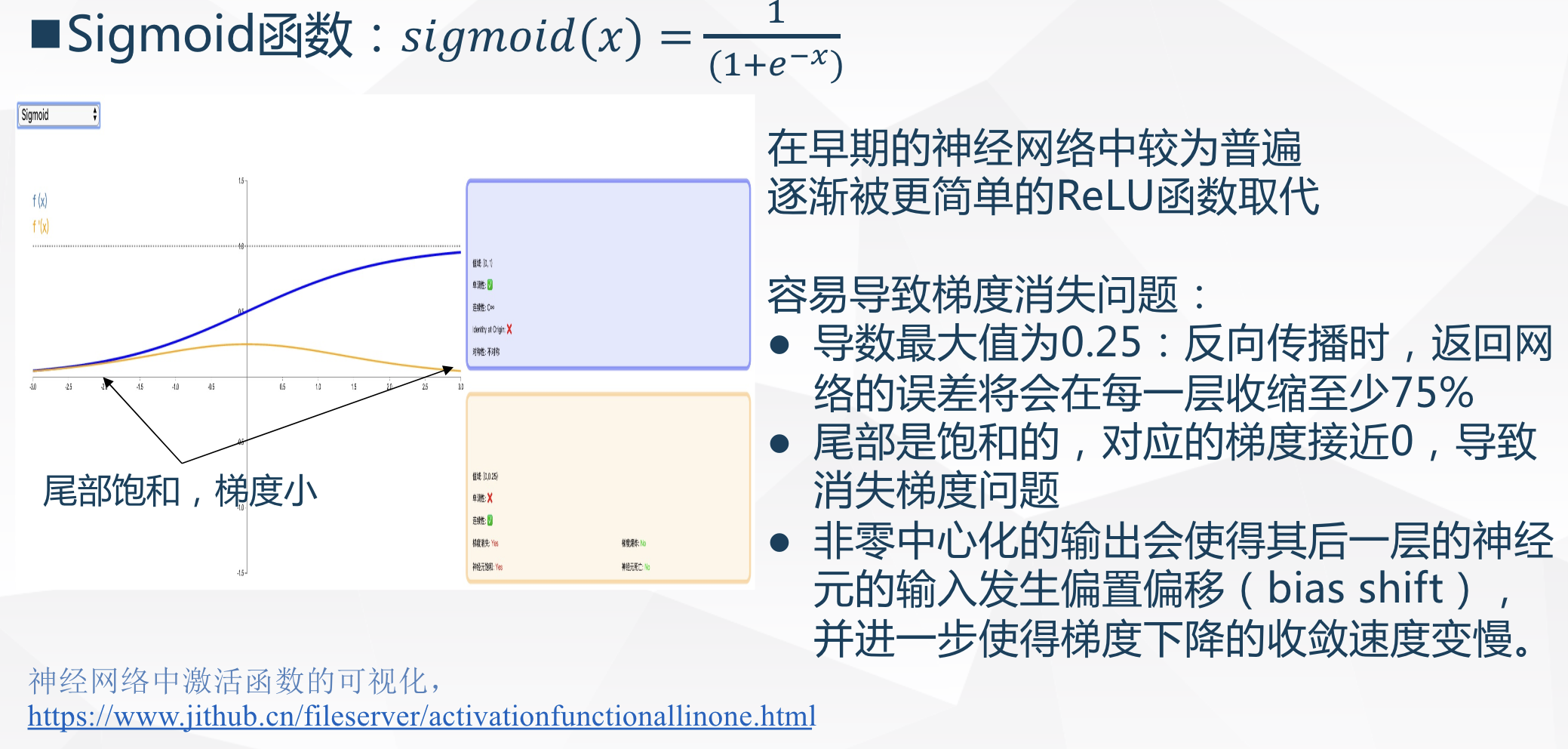
### 激活函数

激活函数性质

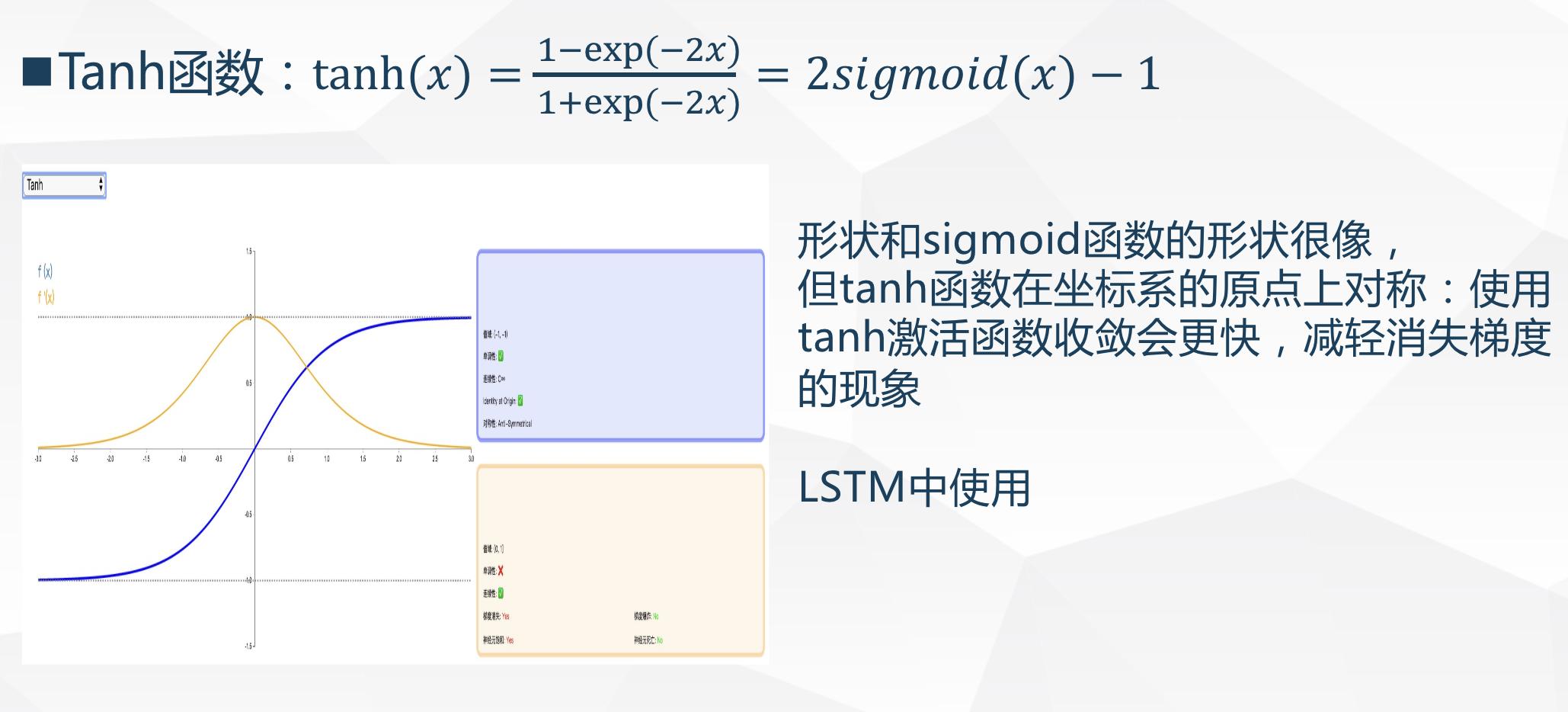
* 连续并可导（允许少数点上不可导）的非线性函数
  + 可导的激活函数可以直接利用数值优化的方法来学习网络参数
* 激活函数及其导函数要尽可能的简单
  + 有利于提高网络计算效率
* 导函数的值域要在一个合适的区间内
  + 不能太大也不能太小，否则会影响训练的效率和稳定性。

**例子**

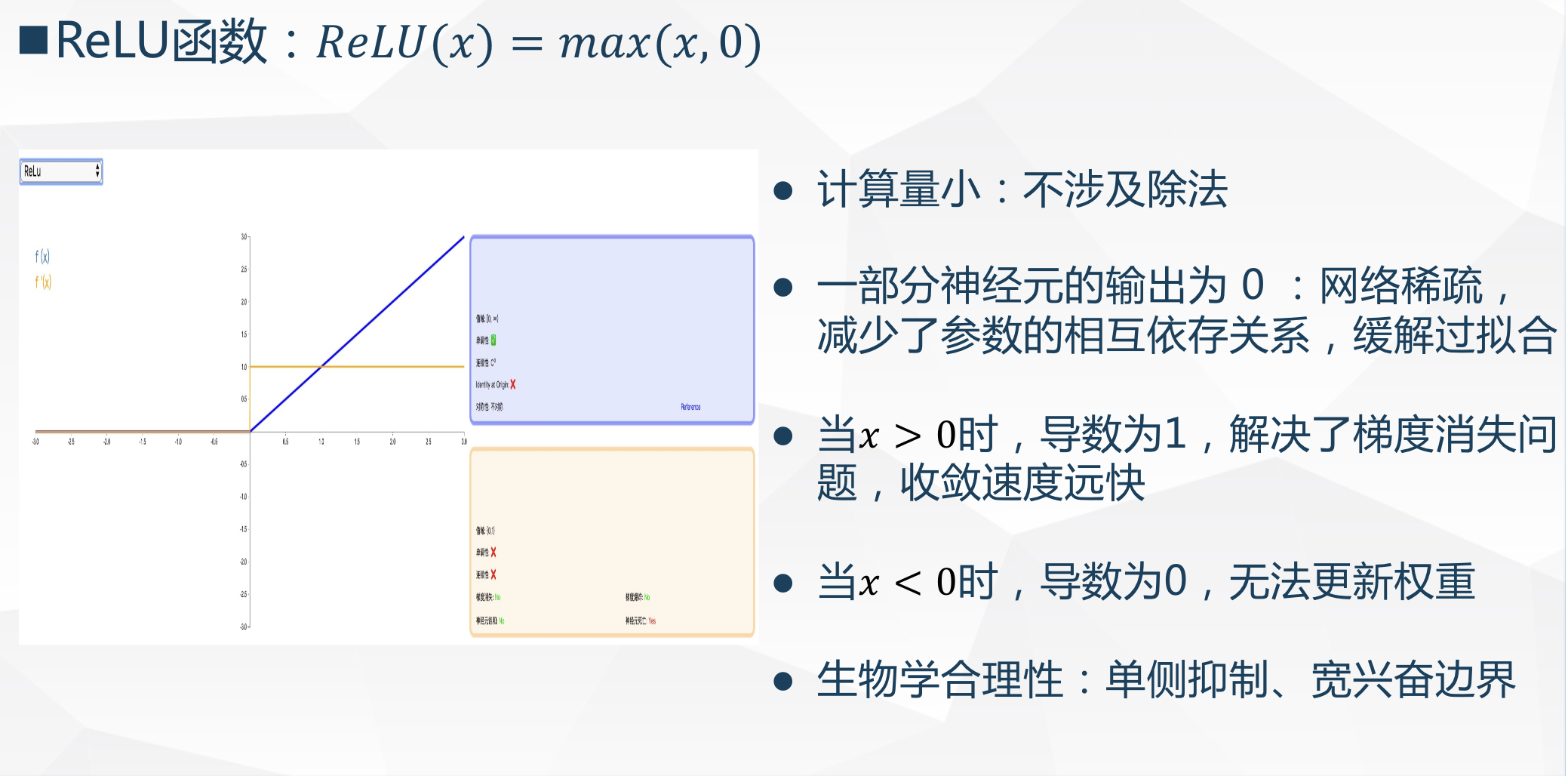
* Sigmoid



* Tanh



* Relu



### Relu变种