Microsoft Reactor

# 一起学人工智能系列 - 线性分类2

2021-09-15

# Map


Redmond
Toronto
New York
San Francisco
Sao Paulo
Stockholm
COMING SOON
London
Tel Aviv
Abu Dhabi
Bangalore
Shanghai
Sydney

# 个人介绍


## Kinfey Lo – (卢建晖)

**Microsoft Cloud Advocate**

前微软MVP、Xamarin MVP和微软RD，拥有超过10年的云原生、人工智能和移动应用经验，为教育、金融和医疗提供应用解决方案。 Microsoft Iginte，Teched 会议讲师，Microsoft AI 黑客马拉松教练，目前在微软，为技术人员和不同行业宣讲技术和相关应用场景。

爱编程(Python， C#，TypeScript，Swift，Rust，Go ）

专注于人工智能，云原生，跨平台移动开发

**Github :** https://github.com/kinfey

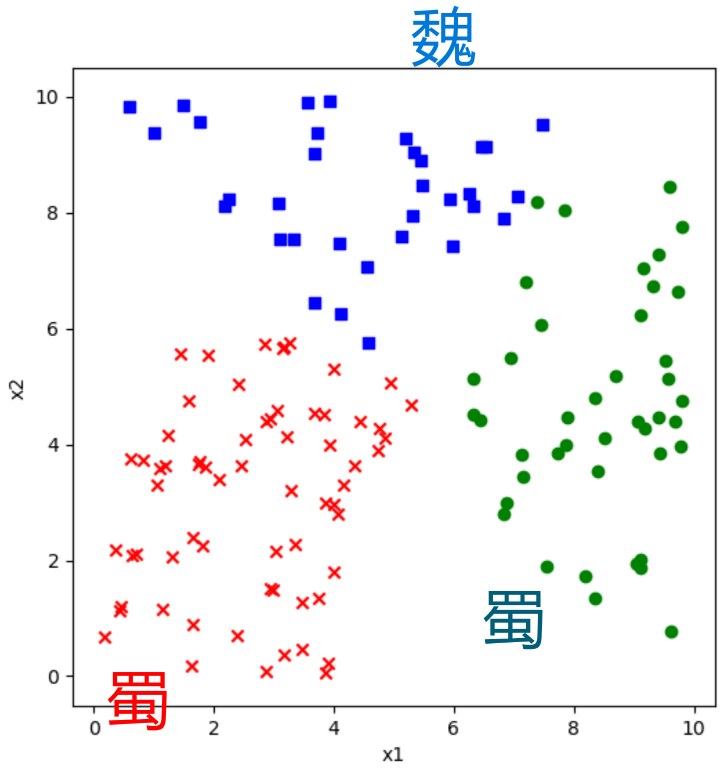
**Email :** kinfeylo@microsoft.com **Blog :** https://blog.csdn.net/kinfey

**Twitter :** @Ljh8304

# 引入

# 知识引入

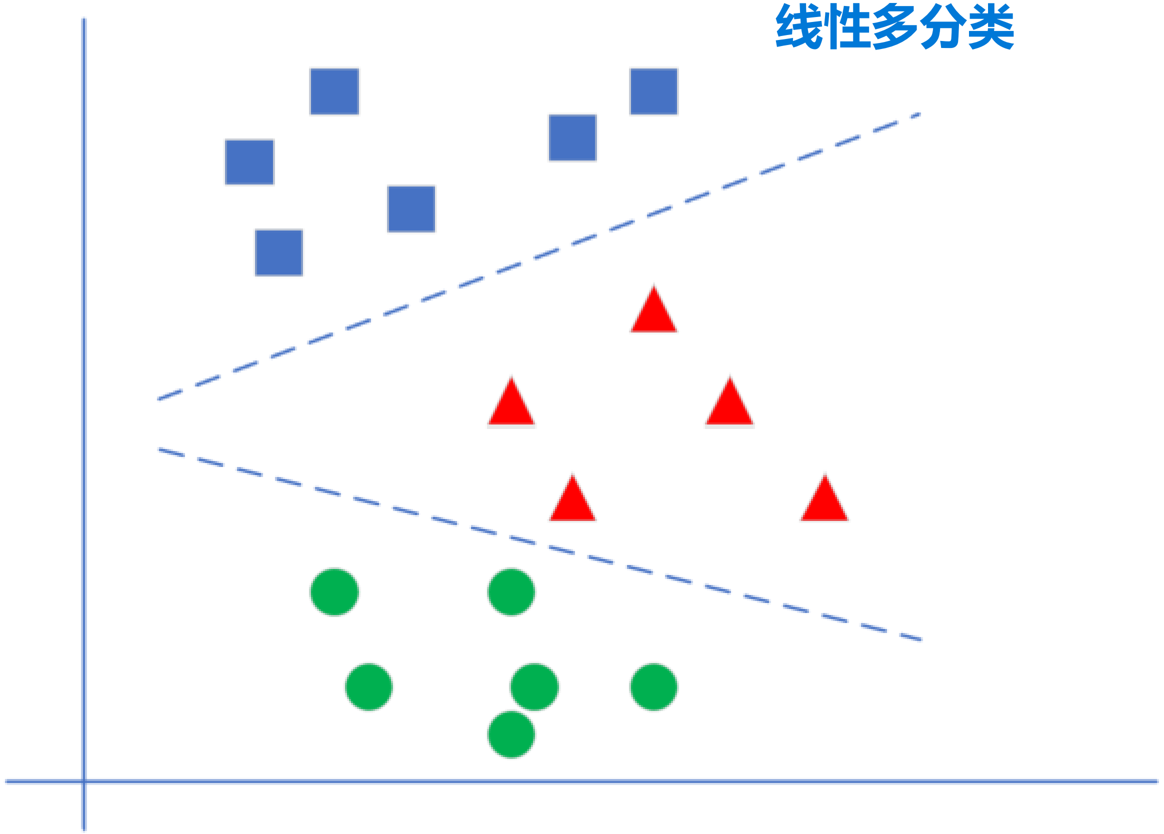
我们解决了公元前的楚汉相争的问题，现在看一下公元220年前后的三国问题。

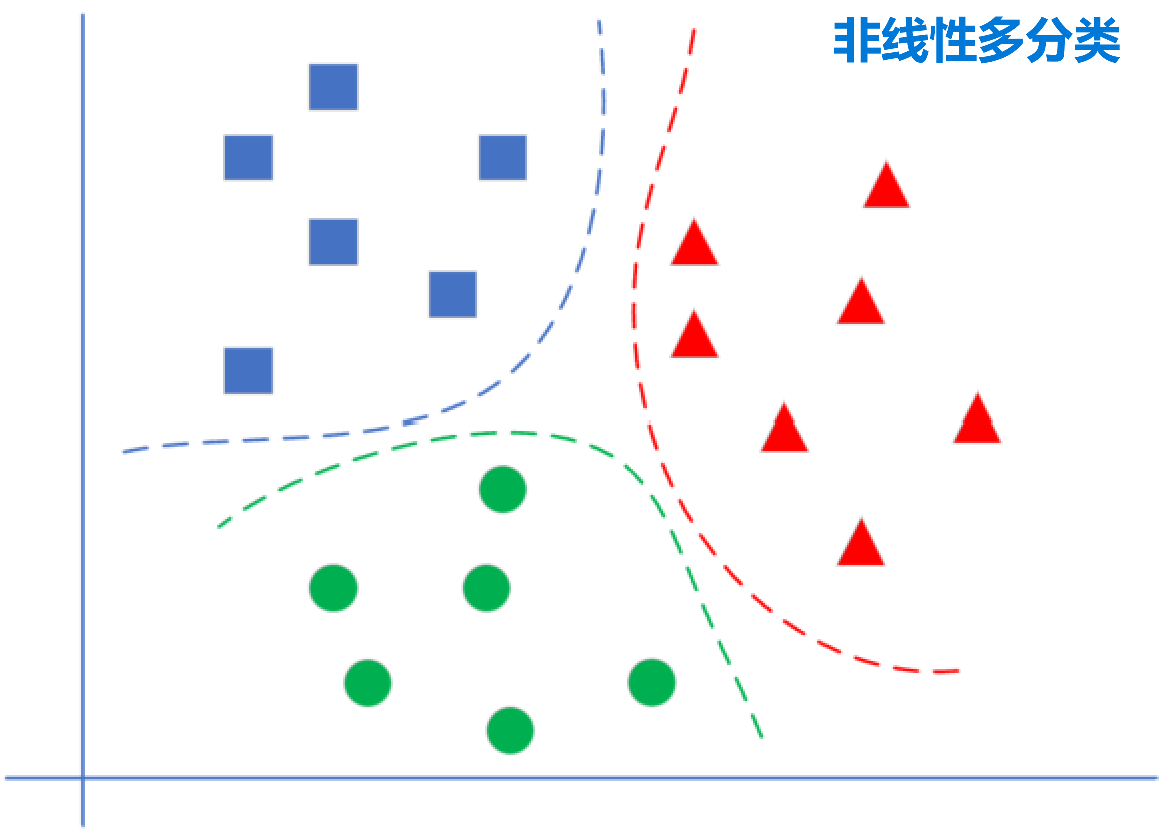



| 样本序号 | x1=经度相对值 | x2=纬度相对值 | y=分类 |
|---|---|---|---|
| 1 | 7.033 | 3.075 | 3 |
| 2 | 4.489 | 4.869 | 2 |
| 3 | 8.228 | 9.735 | 1 |
| ... | ... | ... | ... |
| 140 | 4.632 | 9.014 | 1 |


- 经纬度相对值为（5,1）时，属于哪个国?
- 经纬度相对值为（7,6）时，属于哪个国?
- 经纬度相对值为（5,6）时，属于哪个国?
- 经纬度相对值为（2,7）时，属于哪个国?

# 线性多分类和非线性多分类的区别



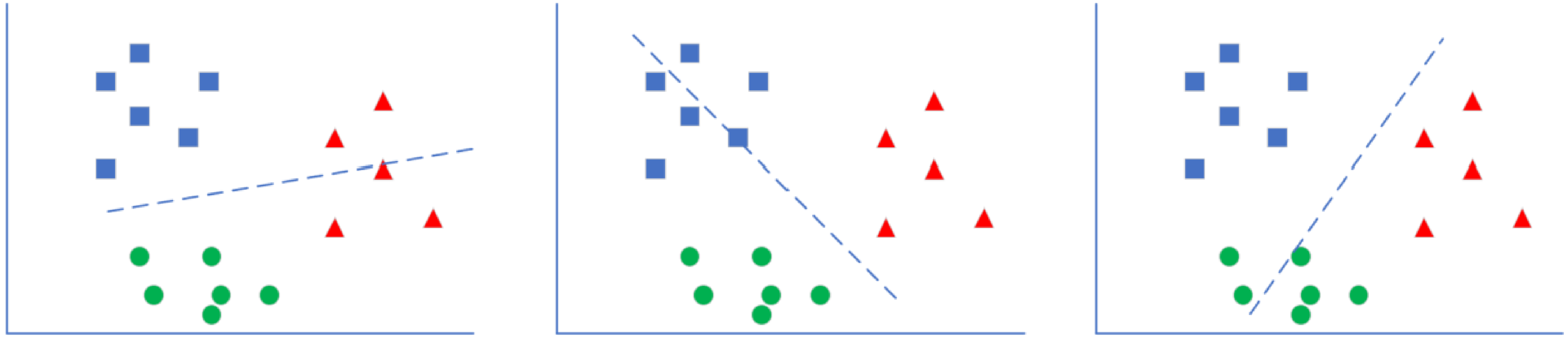

不同类别的样本点之间是否可以用一条直线来互相分割。

对神经网络来说，线性多分类可以使用单层结构来解决，而分线性多分类需要使用双层结构

神经网络做二分类的方法，它并不能用于多分类。

# 多分类问题解法 - 一对一方式

每次先只保留两个类别的数据，训练一个分类器。如果一共有N个类别，则需要训练$C_N^2$个分类器。以N=3时举例，需要训练A|B，B|C，A|CA|B，B|C，A|CA|B，B|C，A|C三个分类器。

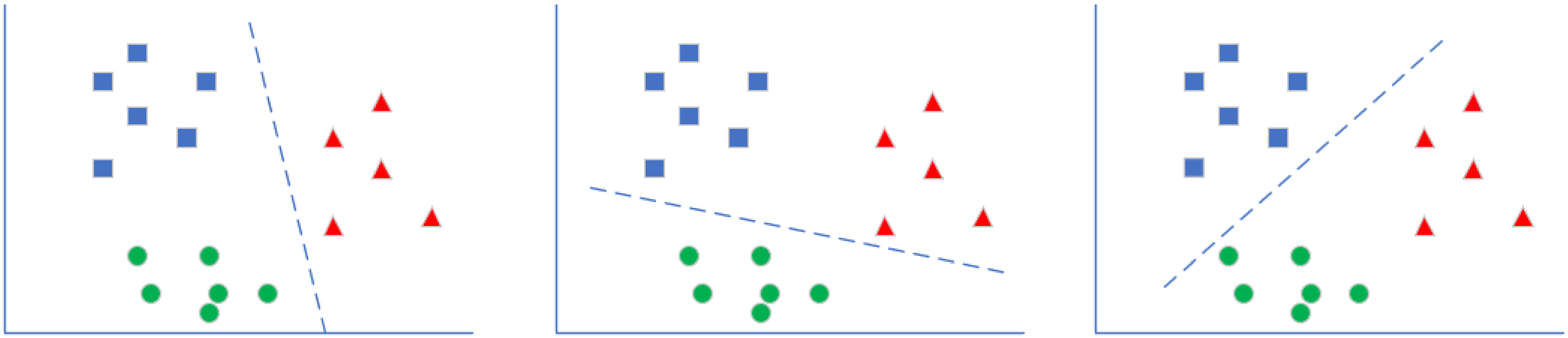


推理时，(A|B)分类器告诉你是A类时，需要到(A|C)分类器再试一下，如果也是A类，则就是A类。如果(A|C)告诉你是C类，则基本是C类了，不可能是B类，不信的话可以到(B|C)分类器再去测试一下。

# 多分类问题解法 - 一对多方式

处理一个类别时，暂时把其它所有类别看作是一类，这样对于三分类问题，可以得到三个分类器

**这种情况是在训练时，把红色样本当作一类，把蓝色和绿色样本混在一起当作另外一类。**



同时调用三个分类器，再把三种结果组合起来，就是真实的结果。比如，第一个分类器告诉你是“红类”，那么它确实就是红类；如果告诉你是非红类，则需要看第二个分类器的结果，绿类或者非绿类；依此类推。

# 多分类问题解法 – 多对多方式

假设有4个类别ABCD，我们可以把AB算作一类，CD算作一类，训练一个分类器1；再把AC算作一类，BD算作一类，训练一个分类器2。

推理时，第1个分类器告诉你是AB类，第二个分类器告诉你是BD类，则做“与”操作，就是B类。

# 多分类与多标签

多分类学习中，虽然有多个类别，但是每个样本只属于一个类别。
有一种情况也很常见，比如一幅图中，既有蓝天白云，又有花草树木，那么这张图片可以有两种标注方法：

•标注为“风景”，而不是“人物”，属于风景图片，这叫做分类

•被同时标注为“蓝天”、“白云”、“花草”、“树木”等多个标签，这样的任务不叫作多分类学习，而是“多标签”学习，multi-label learning。我们此处不涉及这类问题。

# 多分类函数

# 思考

Logistic函数可以得到诸如0.8、0.3这样的二分类概率值，前者接近1，后者接近0。那么多分类问题如何得到类似的概率值呢?

我们依然假设对于一个样本的分类值是用这个线性公式得到的：

$$z = x \cdot w + b$$

但是，我们要求 $z$ 不是一个标量，而是一个向量。如果是三分类问题，我们就要求 $z$ 是一个三维的向量，向量中的每个单元的元素值代表该样本分别属于三个分类的值，这样不就可以了吗？

具体的说，假设$x$是一个 (1x2) 的向量，把w设计成一个(2x3)的向量，b设计成(1x3)的向量，则z就是一个(1x3)的向量。我们假设z的计算结果是$[3, 1, -3]$，这三个值分别代表了样本x在三个分类中的数值，下面我们把它转换成概率值。

有的读者可能会有疑问：我们不能训练神经网络让它的z值直接变成概率形式吗？答案是否定的，因为z值是经过线性计算得到的，线性计算能力有限，无法有效地直接变成概率值。

# 演算

假设输入值是：[3, 1, -3]，如果取max操作会变成：[1,0,0]，这符合我们的分类需要。但是有两个不足：

1. 分类结果是[1, 0, 0]，只保留的非0即1的信息，没有各元素之间相差多少的信息，可以理解是“Hard-Max”
2. max操作本身不可导，无法用在反向传播中。

所以Softmax加了个"soft"来模拟max的行为，但同时又保留了相对大小的信息。

$$a_j = \frac{e^{z_j}}{\sum_{i=1}^{m} e^{z_i}} = \frac{e^{z_j}}{e^{z_1} + e^{z_2} + \cdots + e^{z_m}}$$

上式中:

- $z_j$ 是对第 j 项的分类原始值，即矩阵运算的结果
- $z_i$ 是参与分类计算的每个类别的原始值
- $m$ 是总的分类数
- $a_j$ 是对第 j 项的计算结果

# 演算

假设j=1，m=3，上式为：

$$a_1 = \frac{e^{z_1}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

用图7-5来形象地说明这个过程。

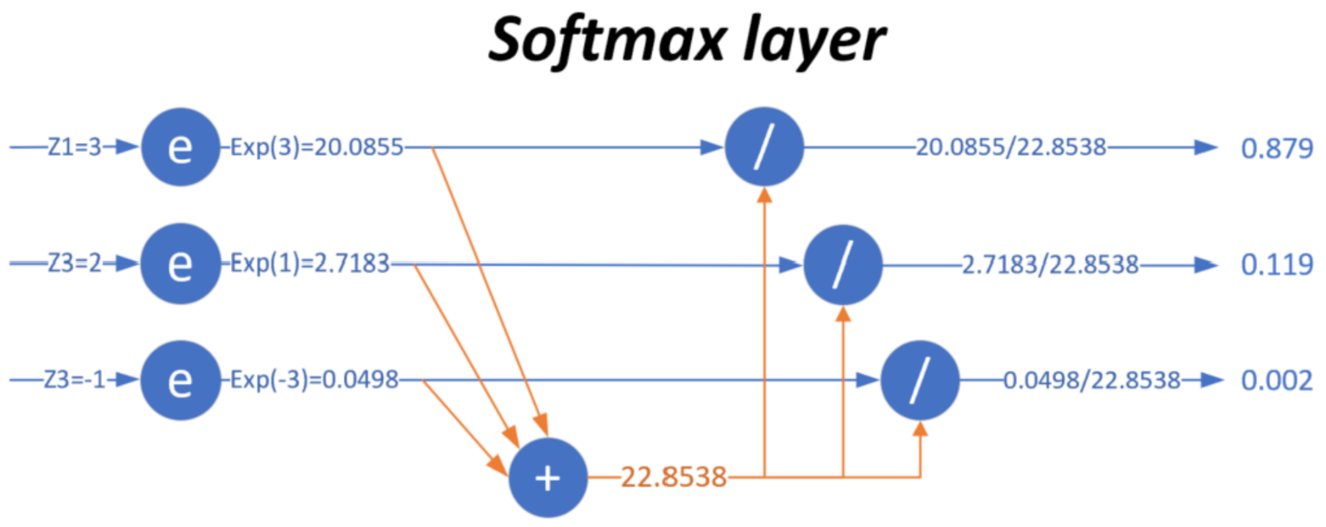



图7-5 Softmax工作过程

当输入的数据 $[z_1, z_2, z_3]$ 是 $[3, 1, -3]$ 时，按照图示过程进行计算，可以得出输出的概率分布是 $[0.879, 0.119, 0.002]$ 。

# MAX与Softmax的对比

| 输入原始值 | (3, 1, -3) |
|---|---|
| MAX计算 | (1, 0, 0) |
| Softmax计算 | (0.879, 0.119, 0.002) |

也就是说，在（至少）有三个类别时，通过使用Softmax公式计算它们的输出，比较相对大小后，得出该样本属于第一类，因为第一类的值为0.879，在三者中最大。注意这是对一个样本的计算得出的数值，而不是三个样本，亦即**softmax给出了某个样本分别属于三个类别的概率。**

它有两个特点：

1. 三个类别的概率相加为1
2. 每个类别的概率都大于0

# Softmax的工作原理

我们仍假设网络输出的预测数据是z=[3, 1, -3]，而标签值是y=[1, 0, 0]。在做反向传播时，根据前面的经验，我们会用z-y，得到：

$$z - y = [2, 1, -3]$$

这个信息很奇怪：

- 第一项是2，我们已经预测准确了此样本属于第一类，但是反向误差的值是2，即惩罚值是2
- 第二项是1，惩罚值是1，预测对了，仍有惩罚值
- 第三项是-3，惩罚值是-3，意为着奖励值是3，明明预测错误了却给了奖励

所以，如果不使用Softmax这种机制，会存在有个问题：

- z值和y值之间，即预测值和标签值之间不可比，比如z[0]=3与y[0]=1是不可比的
- z值中的三个元素之间虽然可比，但只能比大小，不能比差值，比如z[0]>z[1]>z[2]，但3和1相差2，1和-3相差4，这些差值是无意义的

# Softmax的工作原理

在使用Softmax之后，我们得到的值是a=[0.879, 0.119, 0.002]，用a-y：

$$a - y = [-0.121, 0.119, 0.002]$$

再来分析这个信息：

- 第一项-0.121是奖励给该类别0.121，因为它做对了，但是可以让这个概率值更大，最好是1
- 第二项0.119是惩罚，因为它试图给第二类0.119的概率，所以需要这个概率值更小，最好是0
- 第三项0.002是惩罚，因为它试图给第三类0.002的概率，所以需要这个概率值更小，最好是0

这个信息是完全正确的，可以用于反向传播。Softmax先做了归一化，把输出值归一到[0,1]之间，这样就可以与标签值的0或1去比较，并且知道惩罚或奖励的幅度。

Logistic函数也是给出了当前样本的一个概率值，只不过是依靠偏近0或偏近1来判断属于正类还是负类

从继承关系的角度来说，Softmax函数可以视作Logistic函数扩展，比如一个二分类问题：

$$a1 = \frac{e^{z_1}}{e^{z_1} + e^{z_2}} = \frac{1}{1 + e^{z_2 - z_1}}$$

**和Logistic函数形式非常像**

# 正向传播

## 正向传播

### 矩阵运算

$$z = x \cdot w + b \tag{1}$$

### 分类计算

$$a_j = \frac{e^{z_j}}{\sum\limits_{i=1}^{m} e^{z_i}} = \frac{e^{z_j}}{e^{z_1} + e^{z_2} + \cdots + e^{z_m}} \tag{2}$$

### 损失函数计算

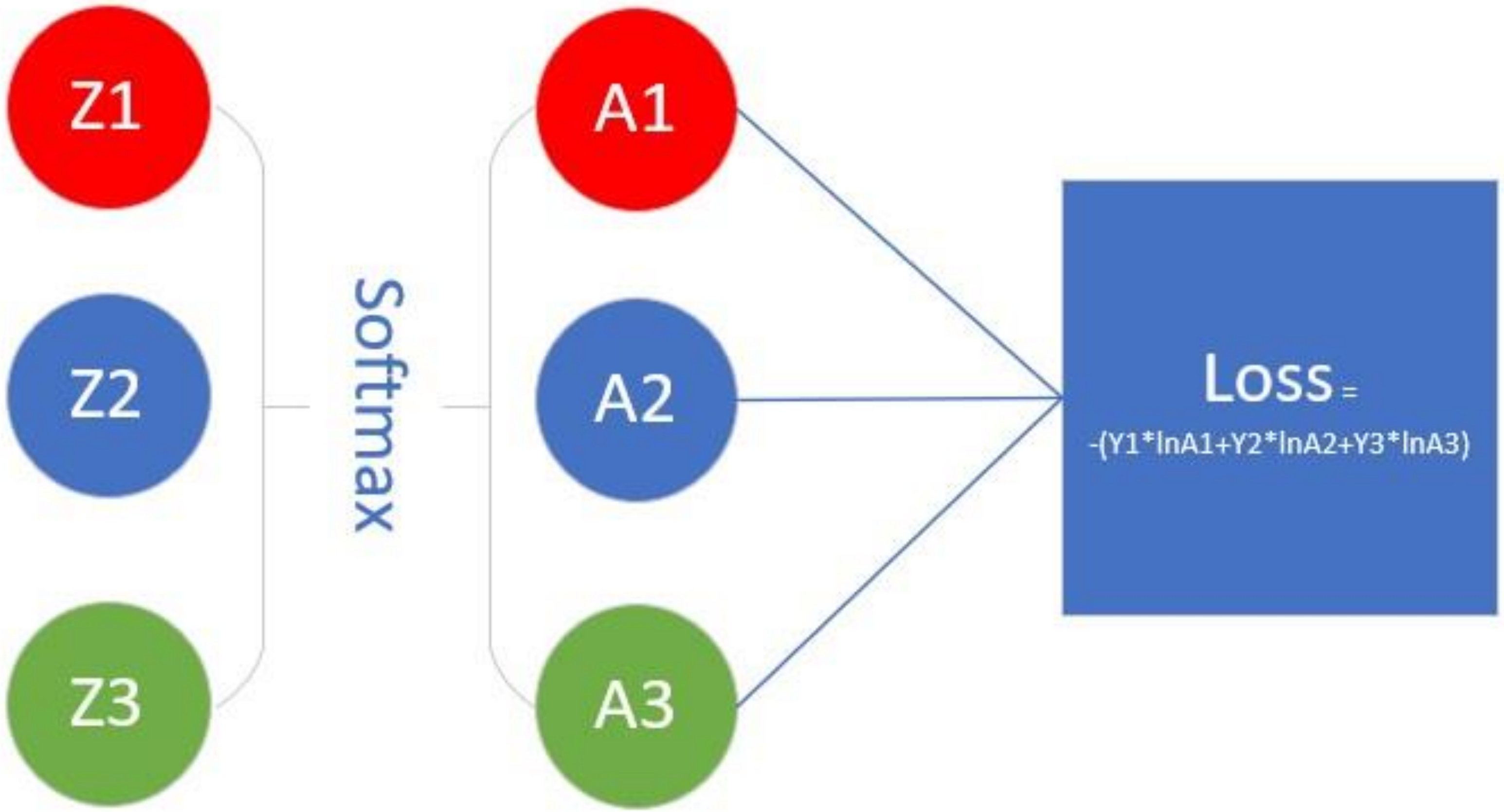
计算单样本时，m是分类数：

$$loss(w, b) = -\sum_{i=1}^{m} y_i \ln a_i \tag{3}$$

计算多样本时，m是分类数，n是样本数：

$$J(w, b) = -\sum_{j=1}^{n} \sum_{i=1}^{m} y_{ij} \log a_{ij} \tag{4}$$

# 正向传播



# 反向传播

我们先用实例化的方式来做反向传播公式的推导，然后再扩展到一般性上。假设有三个类别，则：

$$z_1 = x \cdot w + b_1 \tag{5}$$

$$z_2 = x \cdot w + b_2 \tag{6}$$

$$z_3 = x \cdot w + b_3 \tag{7}$$

$$a_1 = \frac{e^{z_1}}{\sum_i e^{z_i}} = \frac{e^{z_1}}{e^{z_1}+e^{z_2}+e^{z_3}} \tag{8}$$

$$a_2 = \frac{e^{z_2}}{\sum_i e^{z_i}} = \frac{e^{z_2}}{e^{z_1}+e^{z_2}+e^{z_3}} \tag{9}$$

$$a_3 = \frac{e^{z_3}}{\sum_i e^{z_i}} = \frac{e^{z_3}}{e^{z_1}+e^{z_2}+e^{z_3}} \tag{10}$$

为了方便书写，我们令：

$$E = e^{z_1} + e^{z_2} + e^{z_3}$$

$$loss(w,b) = -(y_1 \ln a_1 + y_2 \ln a_2 + y_3 \ln a_3) \tag{11}$$

$$\frac{\partial loss}{\partial z_1} = \frac{\partial loss}{\partial a_1}\frac{\partial a_1}{\partial z_1} + \frac{\partial loss}{\partial a_2}\frac{\partial a_2}{\partial z_1} + \frac{\partial loss}{\partial a_3}\frac{\partial a_3}{\partial z_1} \tag{12}$$

# 反向传播

$$\frac{\partial loss}{\partial a_1} = -\frac{y_1}{a_1} \tag{13}$$

$$\frac{\partial loss}{\partial a_2} = -\frac{y_2}{a_2} \tag{14}$$

$$\frac{\partial loss}{\partial a_3} = -\frac{y_3}{a_3} \tag{15}$$

$$\frac{\partial a_1}{\partial z_1} = (\frac{\partial e^{z_1}}{\partial z_1}E - \frac{\partial E}{\partial z_1}e^{z_1})/E^2 \qquad = \frac{e^{z_1}E - e^{z_1}e^{z_1}}{E^2} = a_1(1-a_1) \tag{16}$$

$$\frac{\partial a_2}{\partial z_1} = (\frac{\partial e^{z_2}}{\partial z_1}E - \frac{\partial E}{\partial z_1}e^{z_2})/E^2 \qquad = \frac{0 - e^{z_1}e^{z_2}}{E^2} = -a_1a_2 \tag{17}$$

$$\frac{\partial a_3}{\partial z_1} = (\frac{\partial e^{z_3}}{\partial z_1}E - \frac{\partial E}{\partial z_1}e^{z_3})/E^2 \qquad = \frac{0 - e^{z_1}e^{z_3}}{E^2} = -a_1a_3 \tag{18}$$

把公式13~18组合到12中：

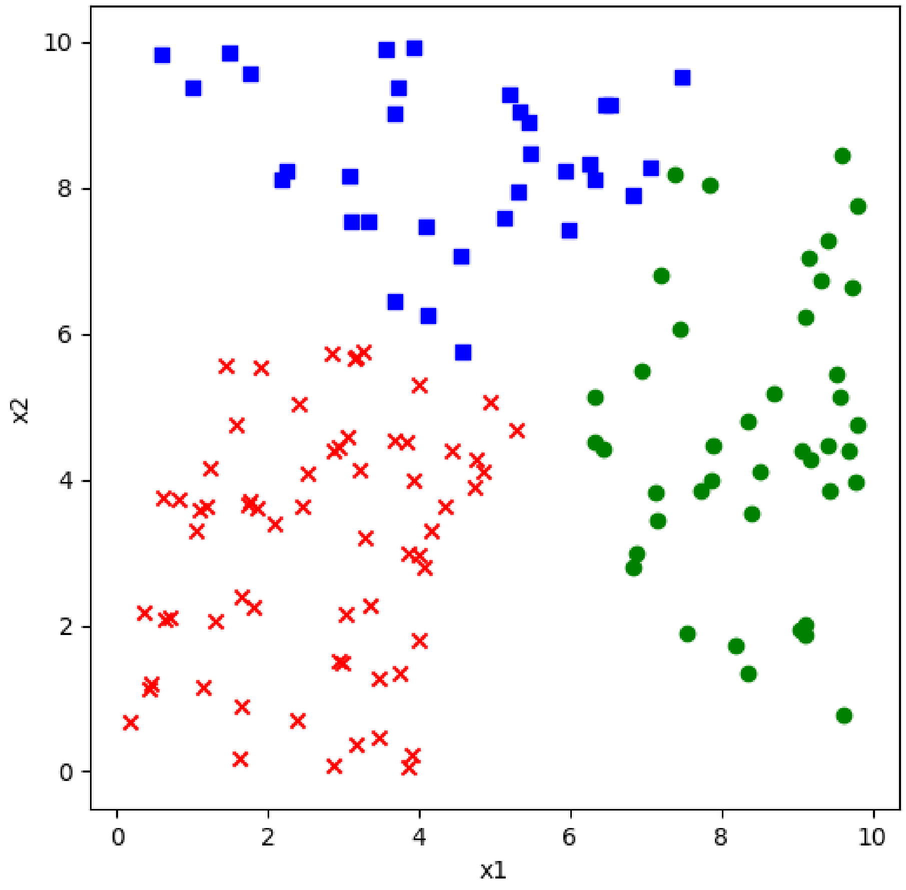
$$\begin{aligned}\tfrac{\partial\,\text{loss}}{\partial z_1} &= -\tfrac{y_1}{a_1}a_1\left(1-a_1\right) + \tfrac{y_2}{a_2}a_1a_2 + \tfrac{y_3}{a_3}a_1a_3 \\ &= -y_1 + y_1a_1 + y_2a_1 + y_3a_1 \\ &= -y_1 + a_1\left(y_1+y_2+y_3\right) \\ &= a_1 - y_1\end{aligned} \tag{19}$$

不失一般性，由公式19可得：

$$\frac{\partial loss}{\partial z_i} = a_i - y_i \tag{20}$$

# 线性多分类的神经网络实现

# 再来看看分地




似乎在三个颜色区间之间有两个比较明显的分界线，而且是直线，即线性可分的。我们如何通过神经网络精确地找到这两条分界线呢？

•从视觉上判断是线性可分的，所以我们使用单层神经网络即可

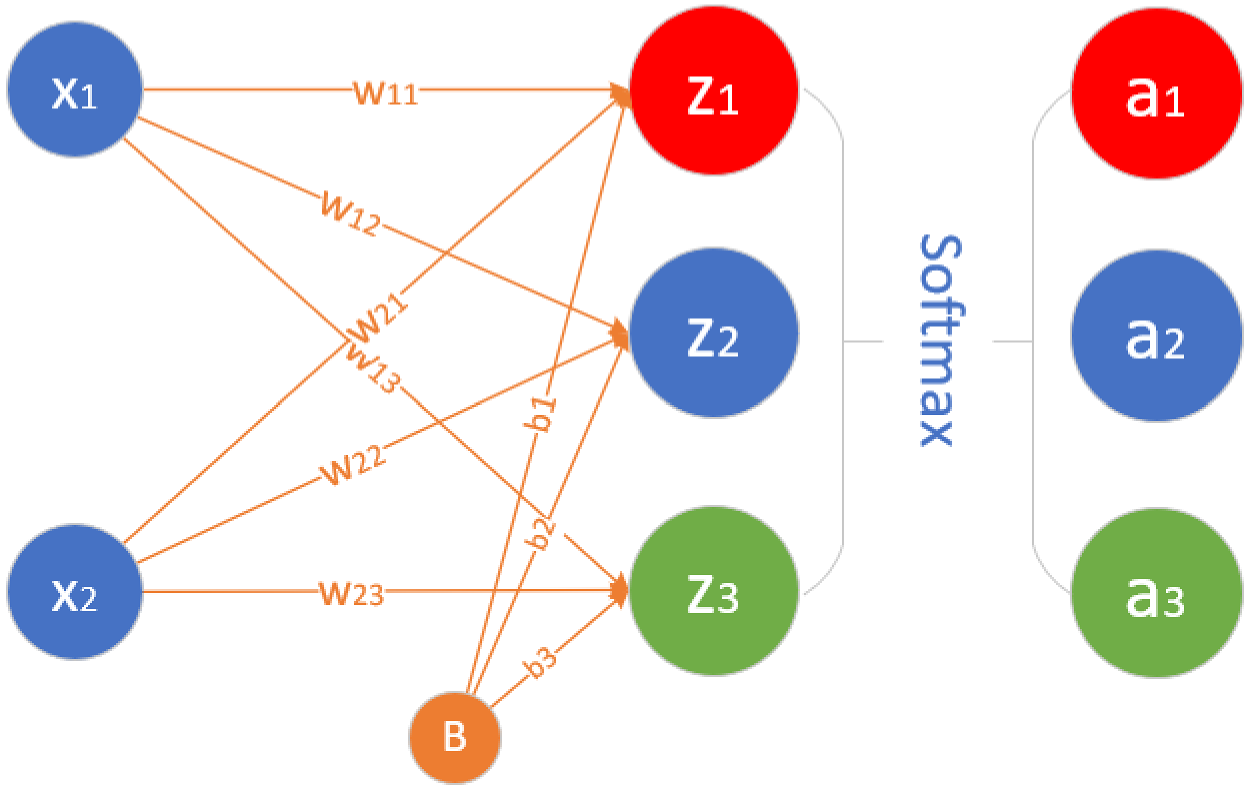
•输入特征是两个，X1=经度，X2=纬度

•最后输出的是三个分类，分别是魏蜀吴，所以输出层有三个神经元

如果有三个以上的分类同时存在，我们需要对每一类别分配一个神经元，这个神经元的作用是根据前端输入的各种数据，先做线性处理（Y=WX+B)，然后做一次非线性处理，计算每个样本在每个类别中的预测概率，再和标签中的类别比较，看看预测是否准确，如果准确，则奖励这个预测，给与正反馈；如果不准确，则惩罚这个预测，给与负反馈。两类反馈都反向传播到神经网络系统中去调整参数。

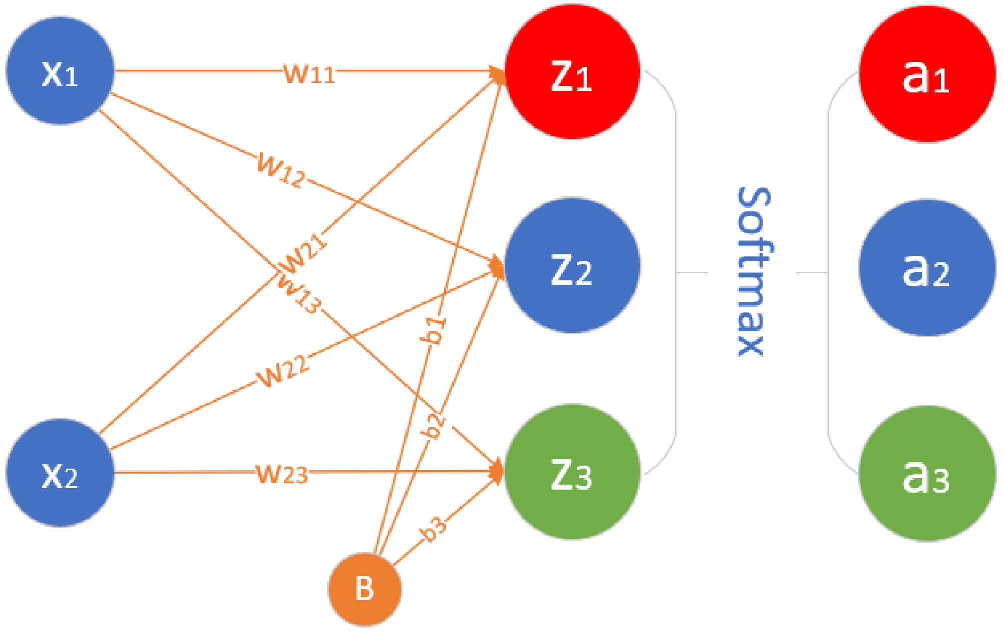
# 神经网络的转化

与前面的单层网络不同的是，输出层还多出来一个Softmax分类函数，这是多分类任务中的标准配置，可以看作是输出层的激活函数，并不单独成为一层，与二分类中的Logistic函数一样。




只有输入层和输出层，由于输入层不算在内，所以是一层网络

# 神经网络的转化




**输入层**

输入经度 $x_1$ 和纬度 $x_2$ 两个特征：

$$x = \begin{pmatrix} x_1 & x_2 \end{pmatrix}$$

**权重矩阵**

$W$ 权重矩阵的尺寸，可以从前往后看，比如：输入层是2个特征，输出层是3个神经元，则 $W$ 的尺寸就是2x3。

$$w = \begin{pmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \end{pmatrix}$$

$B$ 的尺寸是1x3，列数永远和神经元的数量一样，行数永远是1。

$$B = \begin{pmatrix} b_1 & b_2 & b_3 \end{pmatrix}$$

**输出层**

输出层三个神经元，再加上一个Softmax计算，最后有 $A1, A2, A3$ 三个输出，写作：

$$Z = \begin{pmatrix} z_1 & z_2 & z_3 \end{pmatrix}$$

$$A = \begin{pmatrix} a_1 & a_2 & a_3 \end{pmatrix}$$

其中，$Z = X \cdot W + B$，$A = Softmax(Z)$

# 线性多分类原理

# 多分类的过程

1. 线性计算

$$z_1 = x_1 w_{11} + x_2 w_{21} + b_1 \tag{1}$$

$$z_2 = x_1 w_{12} + x_2 w_{22} + b_2 \tag{2}$$

$$z_3 = x_1 w_{13} + x_2 w_{23} + b_3 \tag{3}$$

# 多分类的过程

2. 分类计算

$$a_1 = \frac{e^{z_1}}{\sum_i e^{z_i}} = \frac{e^{z_1}}{e^{z_1}+e^{z_2}+e^{z_3}} \tag{4}$$

$$a_2 = \frac{e^{z_2}}{\sum_i e^{z_i}} = \frac{e^{z_2}}{e^{z_1}+e^{z_2}+e^{z_3}} \tag{5}$$

$$a_3 = \frac{e^{z_3}}{\sum_i e^{z_i}} = \frac{e^{z_3}}{e^{z_1}+e^{z_2}+e^{z_3}} \tag{6}$$

# 多分类的过程

3. 损失函数计算

单样本时，$n$ 表示类别数，$j$ 表示类别序号：

$$loss(w,b) = -(y_1 \ln a_1 + y_2 \ln a_2 + y_3 \ln a_3) = -\sum_{j=1}^{n} y_j \ln a_j \tag{7}$$

批量样本时，$m$ 表示样本数，$i$ 表示样本序号：

$$J(w,b) = -\sum_{i=1}^{m} (y_{i1} \ln a_{i1} + y_{i2} \ln a_{i2} + y_{i3} \ln a_{i3}) = -\sum_{i=1}^{m} \sum_{j=1}^{n} y_{ij} \ln a_{ij} \tag{8}$$

损失函数计算在交叉熵函数一节有详细介绍。

# 数值计算举例

假设对预测一个样本的计算得到的 $z$ 值为：

$$z = [z_1, z_2, z_3] = [3, 1, -3]$$

则按公式4、5、6进行计算，可以得出Softmax的概率分布是：

$$a = [a_1, a_2, a_3] = [0.879, 0.119, 0.002]$$

# 数值计算举例

**如果标签值表明此样本为第一类**

即：

$$y = [1, 0, 0]$$

则损失函数为：

$$loss_1 = -(1 \times \ln 0.879 + 0 \times \ln 0.119 + 0 \times \ln 0.002) = 0.123$$

反向传播误差矩阵为：

$$a - y = [-0.121, 0.119, 0.002]$$

因为 $a_1 = 0.879$，为三者最大，分类正确，所以 $a - y$ 的三个值都不大。

# 数值计算举例

### 如果标签值表明此样本为第一类

即：

$$y=[1,0,0]$$

则损失函数为：

$$loss_1=-(1\times\ln 0.879+0\times\ln 0.119+0\times\ln 0.002)=0.123$$

反向传播误差矩阵为：

$$a-y=[-0.121,0.119,0.002]$$

因为 $a_1=0.879$，为三者最大，分类正确，所以 $a-y$ 的三个值都不大。

### 如果标签值表明此样本为第二类

即：

$$y=[0,1,0]$$

则损失函数为：

$$loss_2=-(0\times\ln 0.879+1\times\ln 0.119+0\times\ln 0.002)=2.128$$

可以看到由于分类错误，$loss_2$ 的值比 $loss_1$ 的值大很多。

反向传播误差矩阵为：

$$a-y=[0.879,0.881,0.002]$$

本来是第二类，误判为第一类，所以前两个元素的值很大，反向传播的力度就大。

# 多分类的几何原理

假设一共有三类样本，蓝色为1，红色为2，绿色为3，那么Softmax的形式应该是:

$$a_j = \frac{e^{z_j}}{\sum_{i=1}^{3} e^{z_i}} = \frac{e^{z_j}}{e^{z_1} + e^{z_2} + {}^{z_3}}$$

# 多分类的几何原理

## 当样本属于第一类时

把蓝色点与其它颜色的点分开。

如果判定一个点属于第一类，则 $a_1$ 的概率值一定会比 $a_2$、$a_3$ 大，表示为公式：

$$a_1 > a_2 \text{且} a_1 > a_3 \tag{9}$$

由于Softmax的特殊形式，分母都一样，所以只比较分子就行了。而分子是一个自然指数，输出值域大于零且单调递增，所以只比较指数就可以了，因此，公式9等同于下式：

$$z_1 > z_2 \text{且} z_1 > z_3 \tag{10}$$

# 多分类的几何原理

把公式1、2、3引入到10：

$$x_1w_{11}+x_2w_{21}+b1>x_1w_{12}+x_2w_{22}+b_2 \tag{11}$$

$$x_1w_{11}+x_2w_{21}+b1>x_1w_{13}+x_2w_{23}+b_3 \tag{12}$$

变形：

$$(w_{21}-w_{22})x_2>(w_{12}-w_{11})x_1+(b_2-b_1) \tag{13}$$

$$(w_{21}-w_{23})x_2>(w_{13}-w_{11})x_1+(b_3-b_1) \tag{14}$$

我们先假设：

$$w_{21}>w_{22}\text{，且}w_{21}>w_{23} \tag{15}$$

所以公式13、14左侧的系数都大于零，两边同时除以系数：

$$x_2>\frac{w_{12}-w_{11}}{w_{21}-w_{22}}x_1+\frac{b_2-b_1}{w_{21}-w_{22}} \tag{16}$$

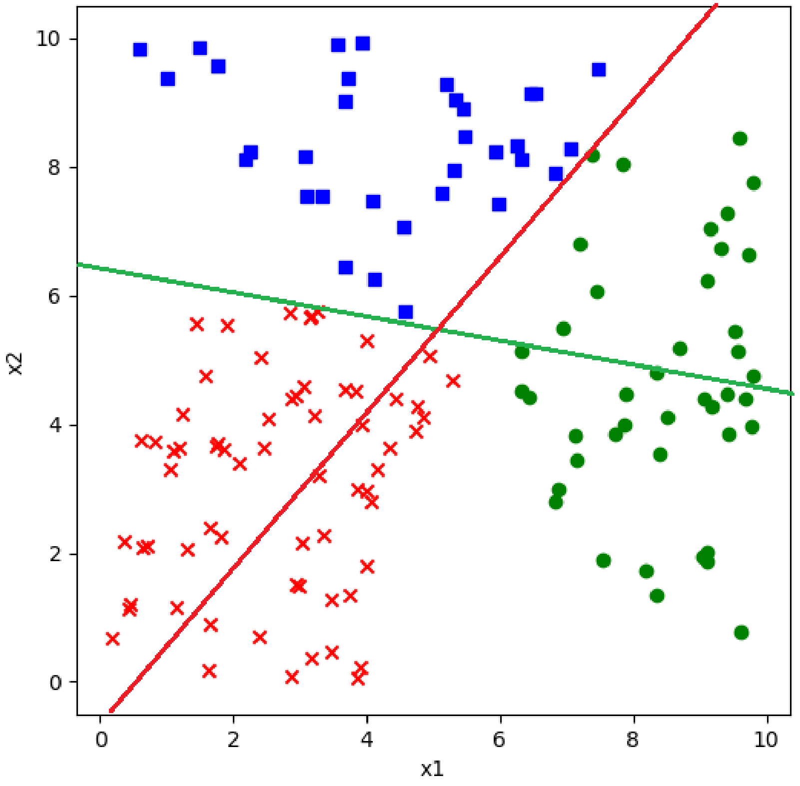
$$x_2>\frac{w_{13}-w_{11}}{w_{21}-w_{23}}x_1+\frac{b_3-b_1}{w_{21}-w_{23}} \tag{17}$$

简化：

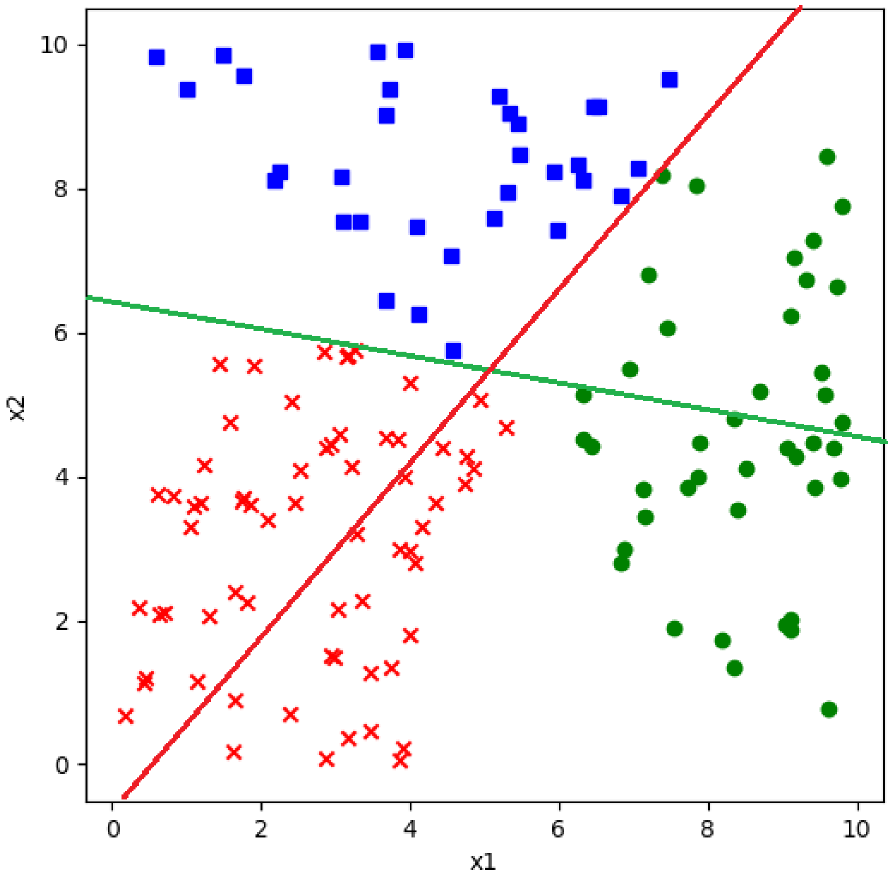
$$y>W_{12}\cdot x+B_{12} \tag{18}$$

$$y>W_{13}\cdot x+B_{13} \tag{19}$$

此时y代表了第一类的蓝色点。



# 多分类的几何原理




$$y > W_{12} \cdot x + B_{12} \tag{18}$$

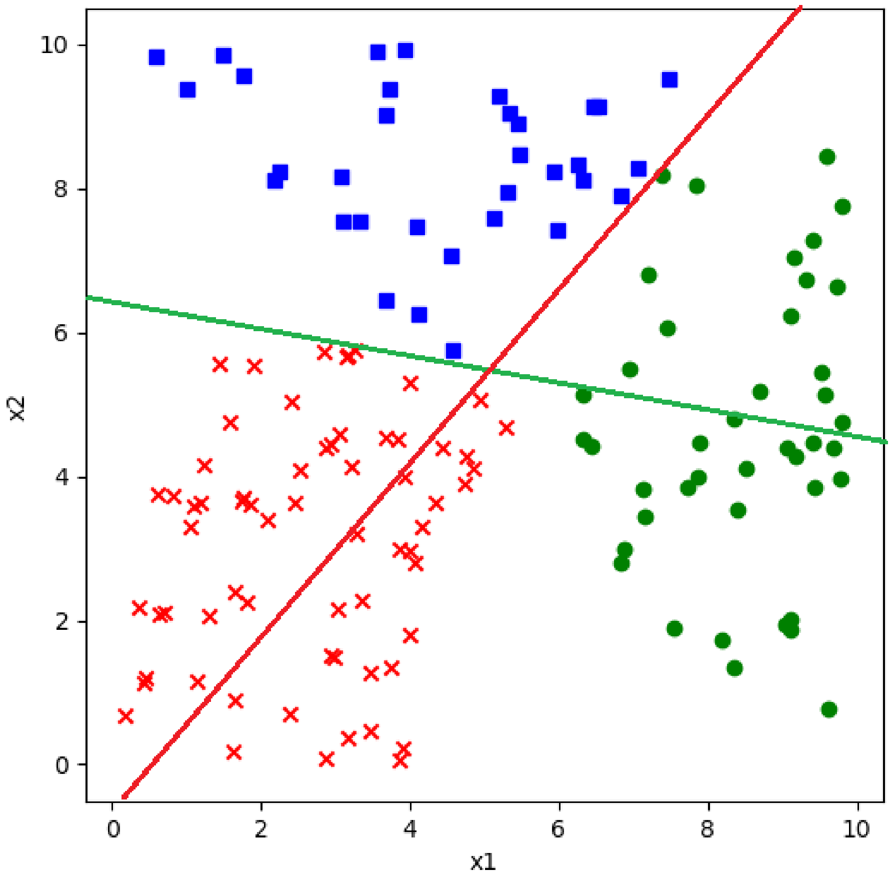
$$y > W_{13} \cdot x + B_{13} \tag{19}$$

借用二分类中的概念，公式18的几何含义是：有一条直线可以分开第一类（蓝色点）和第二类（红色点），使得所有蓝色点都在直线的上方，所有的红色点都在直线的下方。于是我们可以画出图7-9中的那条**绿色直线。**

而公式19的几何含义是：有一条直线可以分开第一类（蓝色点）和第三类（绿色点），使得所有蓝色点都在直线的上方，所有的绿色点都在直线的下方。于是我们可以画出图7-9中的那条红色直线。

也就是说在图中画两条直线，所有蓝点都同时在红线和绿线这两条直线的上方。

# 多分类的几何原理




$$y > W_{12} \cdot x + B_{12} \tag{18}$$

$$y > W_{13} \cdot x + B_{13} \tag{19}$$

借用二分类中的概念，公式18的几何含义是：有一条直线可以分开第一类（蓝色点）和第二类（红色点），使得所有蓝色点都在直线的上方，所有的红色点都在直线的下方。于是我们可以画出图7-9中的那条**绿色直线。**
而公式19的几何含义是：有一条直线可以分开第一类（蓝色点）和第三类（绿色点），使得所有蓝色点都在直线的上方，所有的绿色点都在直线的下方。于是我们可以画出图7-9中的那条红色直线。
也就是说在图中画两条直线，所有蓝点都同时在红线和绿线这两条直线的上方。

# 多分类的几何原理

**当样本属于第二类时**

即如何把红色点与其它两色点分开。

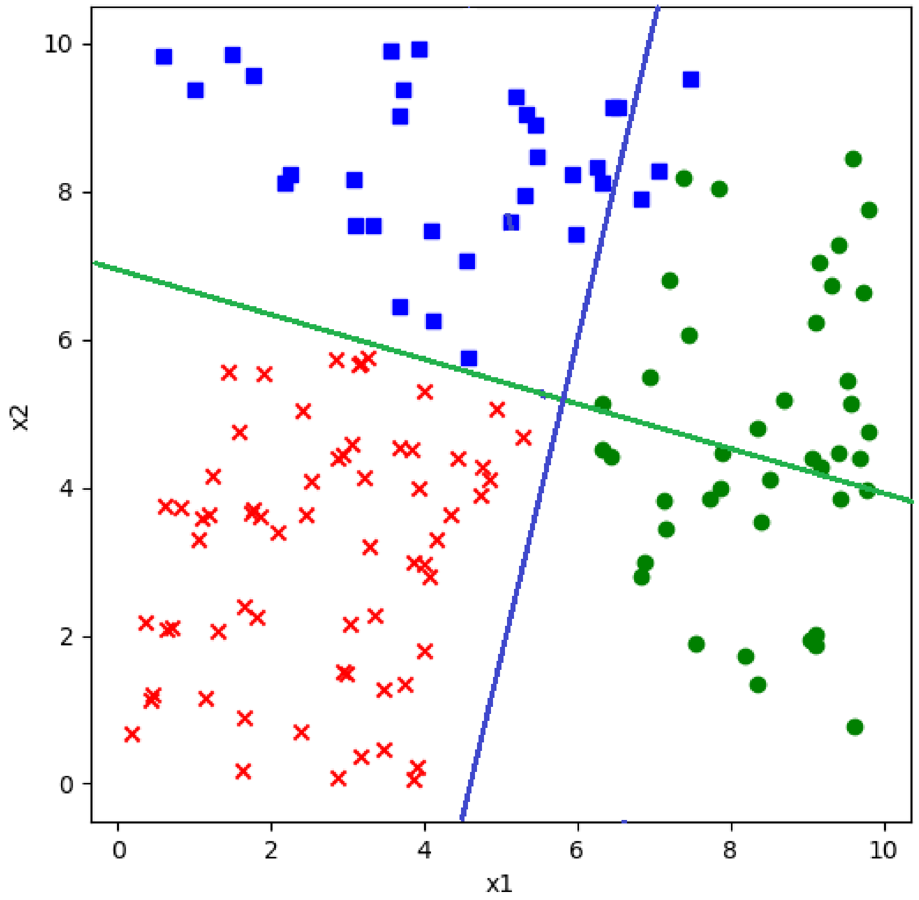
$$z_2 > z_1 且 z_2 > z_3 \tag{20}$$

同理可得

$$y < W_{12} \cdot x + B_{12} \tag{21}$$

$$y > W_{23} \cdot x + B_{23} \tag{22}$$

此时yyy代表了第二类的红色点。
公式21和公式18几何含义相同，不等号相反，代表了图7-10中绿色直线的分割作用，即红色点在绿色直线下方。
公式22的几何含义是，有一条蓝色直线可以分开第二类（红色点）和第三类（绿色点），使得所有红色点都在直线的上方，所有的绿色点都在直线的下方。



# 多分类的几何原理

即如何把绿色点与其它两色点分开。

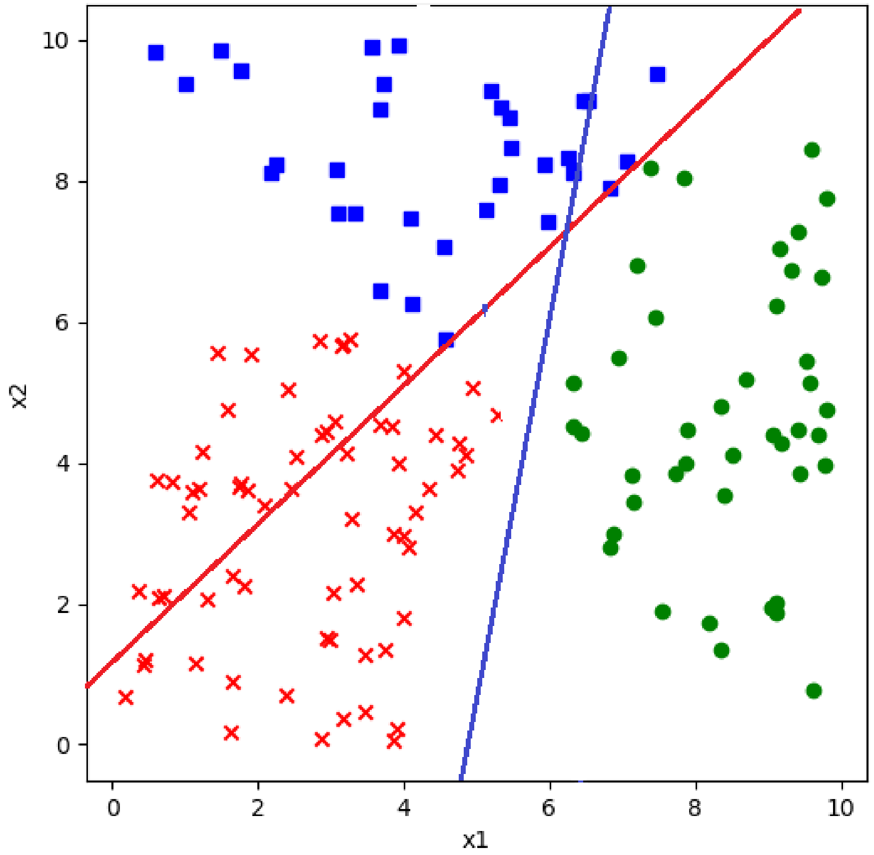
$$z_3 > z_1 且 z_3 > z_2 \tag{22}$$

最后可得：

$$y < W_{13} \cdot x + B_{13} \tag{23}$$

$$y < W_{23} \cdot x + B_{23} \tag{24}$$

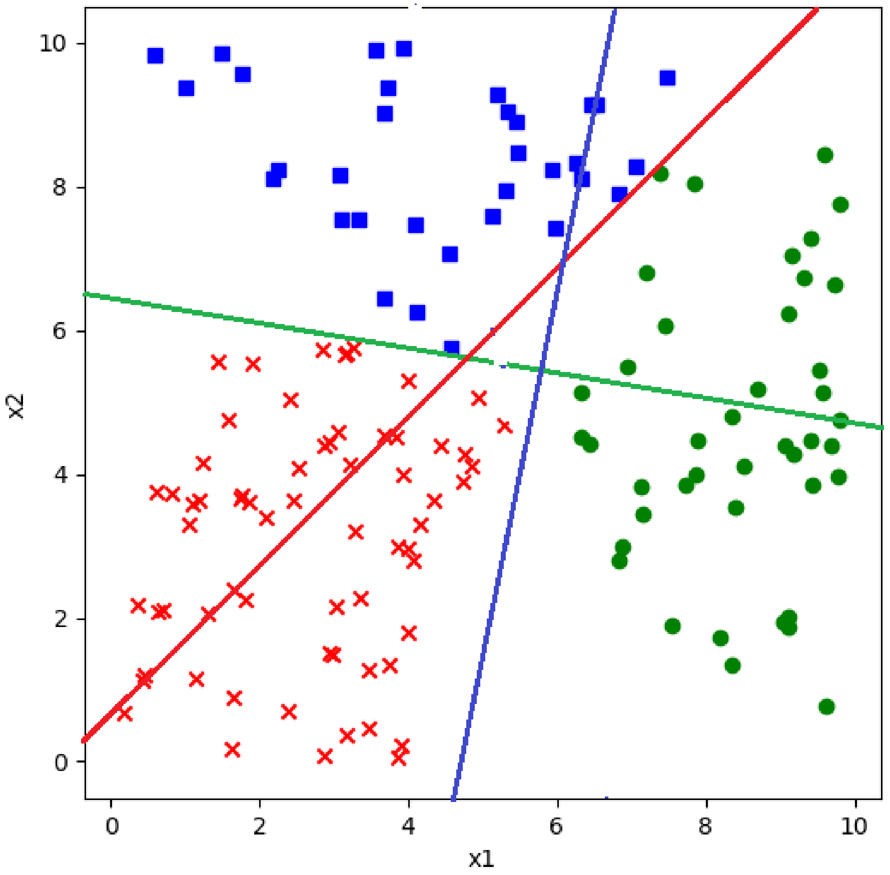
此时 $y$ 代表了第三类的绿色点。




公式23与公式19不等号相反，几何含义相同，代表了图7-11中红色直线的分割作用，绿色点在红色直线下方。

公式24与公式22不等号相反，几何含义相同，代表了图7-11中蓝色直线的分割作用，绿色点在蓝色直线下方。

# 多分类的几何原理



# 示例

Microsoft

Reactor

# Thank You!