智能之门

神经网络和深度学习入门

（基于Python的实现）

STEP 2 线性回归

第 7 章

多入多出的单层神经网络

线性多分类

本部分主要介绍线性多分类。作多分类任务时，可以采用一对一、一对多、多对多的方式，那么神经网络使用的是哪种方式呢？我们将探讨这个问题。此外，Softmax 函数是多分类问题的分类函数，通过对它的分析，我们能学习多分类的原理、实现、以及可视化结果，从而理解神经网络的工作方式。

7.1 线性多分类问题

上一章解决了楚汉相争的问题，本章来解决三国问题。

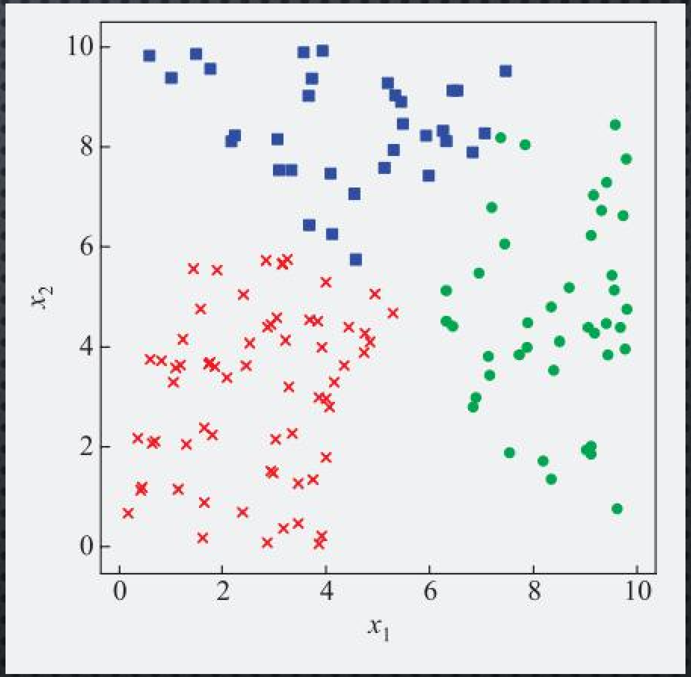
分类标签值的含义为：

魏国城池，标签为1，蓝色点；

蜀国城池，标签为2，红色点；

吴国城池，标签为3，绿色点。

样本序号	相对经度值 x_1	相对纬度值 x_2	分类 y
1	7.033	3.075	3
2	4.489	4.869	2
3	8.228	9.735	1
...	...	...	...
140	4.632	9.014	1

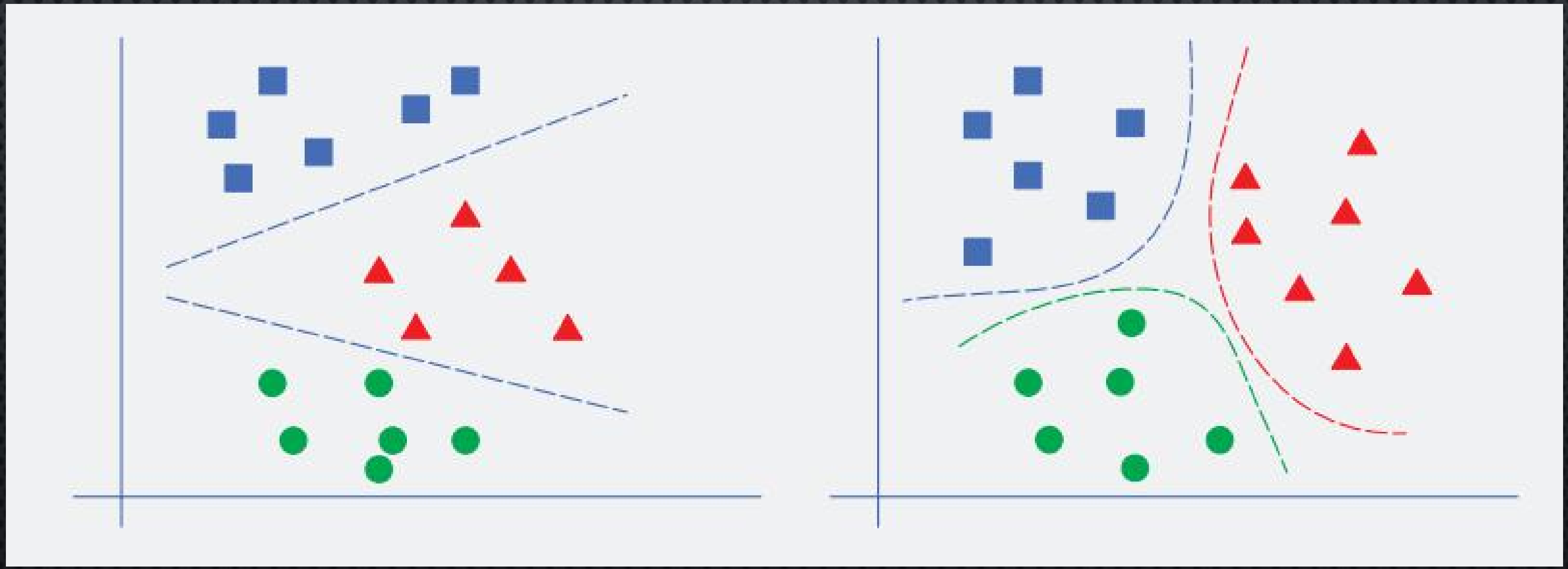


问题：经纬度相对坐标值分别为(5,1), (7,6), (5,6), (2,7)时，属于哪个国？

7.1 线性多分类问题

➢ 多分类学习策略

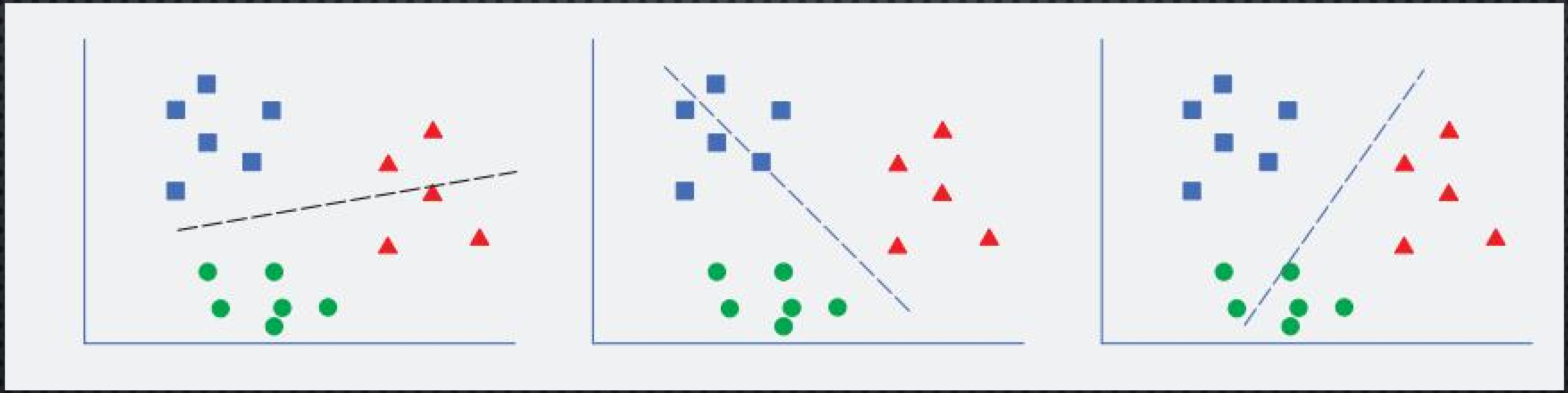
- 下图展示了线性多分类与非线性多分类的结果，左侧为线性多分类，右侧为非线性多分类。它们的区别在于不同类别的样本点之间是否可以用一条直线来互相分割。对神经网络来说，线性多分类可以使用单层结构来解决，而分线性多分类需要使用双层结构。



7.1 线性多分类问题

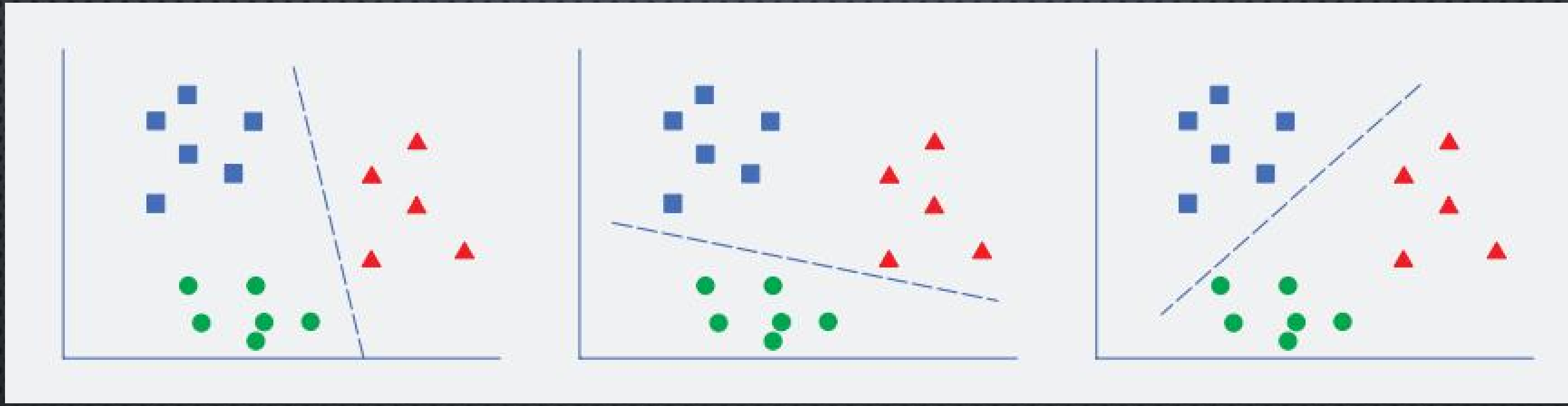
➢ 多分类问题的三种解法

- 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。如果数据有N类，则需要训练 C_N^2 个分类器。



7.1 线性多分类问题

- 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。



- 多对多方式：多个类别数据组合，作二分类，预测时结合多个分类结果进行逻辑运算。

➢ 多标签学习

- 同时被标注多个标签，区别于多分类问题。

7.2 多分类函数

➢ SOFTMAX 函数

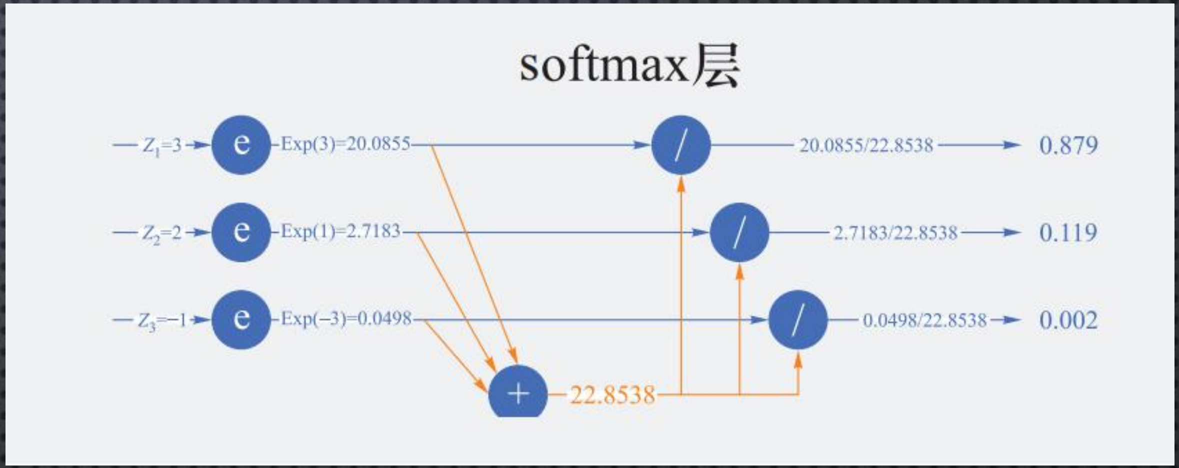
- 假设输入值是 $[3,1,-3]$，如果直接进行取 MAX 操作会变成 $[1,0,0]$，这符合我们的分类需要，但是有两个不足：
  - ✓ 分类结果缺少各元素间相差多少的信息，可以理解为“HARD-MAX”。
  - ✓ MAX 函数本身不可导，无法应用于反向传播。
- 所以 SOFTMAX 函数加了个“SOFT”来模拟 MAX 的行为，同时保留了相对大小的信息：

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

7.2 多分类函数

➢ Softmax 函数的特点

- 各个类别的概率相加为1。
- 各个类别的概率均非负。



输入原始值	(3, 1, −3)
max 运算	(1, 0, 0)
softmax 运算	(0.879, 0.119, 0.002)

7.2 多分类函数

➢ 工作原理

- 若不使用 SOFTMAX 机制，会导致：
  - ✓ 预测值与标签值之间不可比。
  - ✓ 预测向量中的三个元素只可比较大小，差值无意义。

- 若使用 SOFTMAX 机制，得到 $a-y=(-0.121,0.119,0.002)$，很好地表示了奖励或惩罚的幅度，并可以用于反向传播。
- SOFTMAX 函数可以视作 LOGISTIC 函数的扩展，如二分类问题：

$$a_1=\frac{e^{z_1}}{e^{z_1}+e^{z_2}}=\frac{1}{1+e^{z_2-z_1}}$$

7.2 多分类函数

➢ 正向计算

- 矩阵运算

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

- 分类计算

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

- 损失函数计算

$$loss(w,b) = -\sum_{i=1}^{m} y_i \ln a_i\,, \qquad J(w,b) = -\sum_{i=1}^{m}\sum_{j=1}^{n} y_{ij} \ln a_{ij}$$

7.2 多分类函数

➢ 反向传播

- 假设有三个类别，为了方便书写，令

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

- 由于

$$\frac{\partial loss}{\partial a_1} = -\frac{y_1}{a_1}, \qquad \frac{\partial loss}{\partial a_2} = -\frac{y_2}{a_2}, \qquad \frac{\partial loss}{\partial a_3} = -\frac{y_3}{a_3}$$

$$\frac{\partial a_1}{\partial z_1} = \frac{e^{z_1}E - e^{z_1}e^{z_1}}{E^2} = a_1(1 - a_1), \qquad \frac{\partial a_2}{\partial z_1} = \frac{0 - e^{z_1}e^{z_2}}{E^2} = -a_1a_2, \qquad \frac{\partial a_3}{\partial z_1} = \frac{0 - e^{z_1}e^{z_3}}{E^2} = -a_1a_3$$

- 由链式法则可算得

$$\frac{\partial loss}{\partial z_1} = \frac{\partial loss}{\partial a_1} \cdot \frac{\partial a_1}{\partial z_1} + \frac{\partial loss}{\partial a_2} \cdot \frac{\partial a_2}{\partial z_1} + \frac{\partial loss}{\partial a_3} \cdot \frac{\partial a_3}{\partial z_1} = a_1 - y_1$$

7.2 多分类函数

- **反向传播的一般性推导**
  - 假设有 m 个类别，令 $E = e^{z_1} + e^{z_2} + \cdots + e^{z_m}$。
  - 当 $i = j$ 时，有

$$\frac{\partial a_j}{\partial z_i} = \frac{e^{z_j}E - e^{z_j}e^{z_j}}{E^2} = a_j(1 - a_j)$$

  - 当 $i \neq j$ 时，有

$$\frac{\partial a_j}{\partial z_i} = \frac{-e^{z_i}e^{z_j}}{E^2} = -a_i a_j$$

  - 由链式法则可算得

$$\frac{\partial loss}{\partial z_i} = \sum_{j=1}^{m} \frac{\partial loss}{\partial a_j} \cdot \frac{\partial a_j}{\partial z_i} = a_i - y_i$$

7.3 线性多分类的实现

看魏蜀吴城池示意图，每两个颜色区域之间似乎存在一条分割直线，即线性可分的。

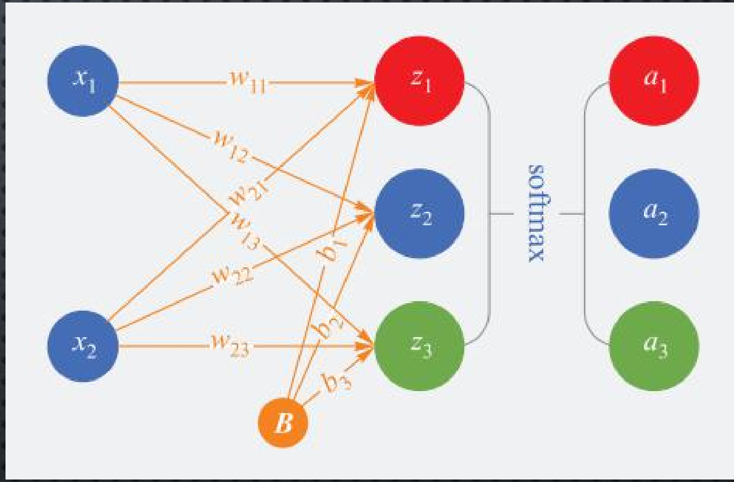
➢ 神经网络结构

- 从视觉上判断是线性可分的，使用单层神经网络；
- 输入层设置两个输入单元，表示经纬度：$X = (x_1, x_2)$。
- 输出层设置三个单元，表示地盘所属阵营：

$$z = XW + B, a = Softmax(z)$$

- 权重参数：

$$W = \begin{pmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \end{pmatrix}, \qquad B = (b_1 \quad b_2 \quad b_3)$$

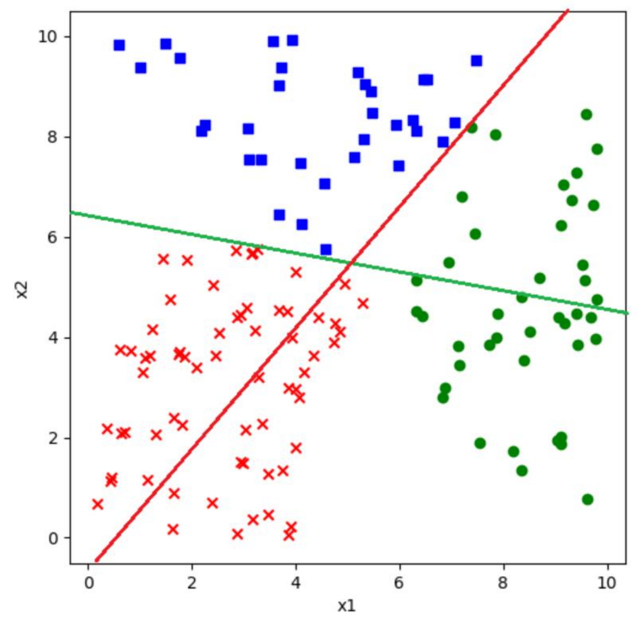


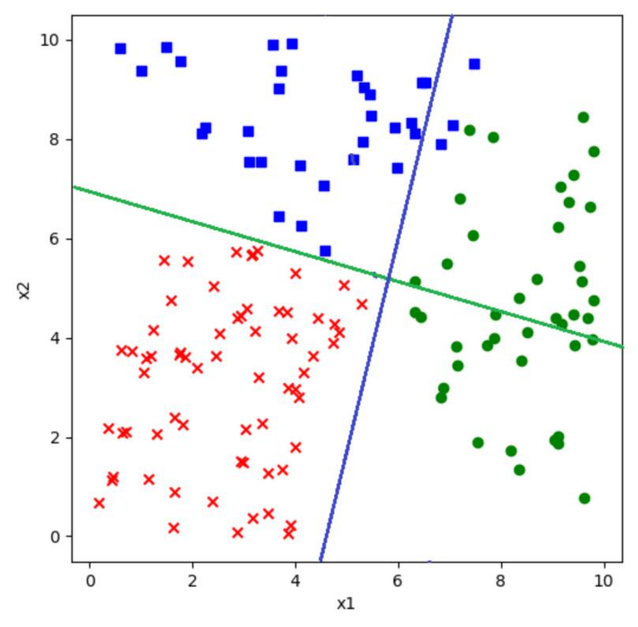
分类结果：经纬度相对坐标值分别为(5,1), (7,6), (5,6), (2,7)时，属于蜀国、吴国、魏国、魏国。

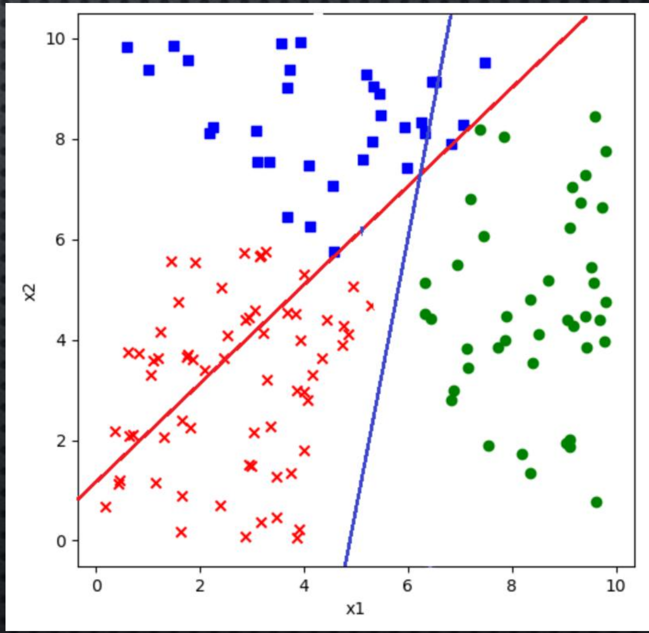
7.4 线性多分类的工作原理

➢ 几何原理

- 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；第二类、第三类的情况道理类同。简单的代数变换可以分离出单类样本的范围。







THE END

谢谢！