机器学习相关线性代数简介

> 参考书籍：
>
> - [mathematics for machine learning]
> - [Introduction to Linear Algebra for Applied Machine Learning with Python]
>
> 起始编辑时间：2021-07-12

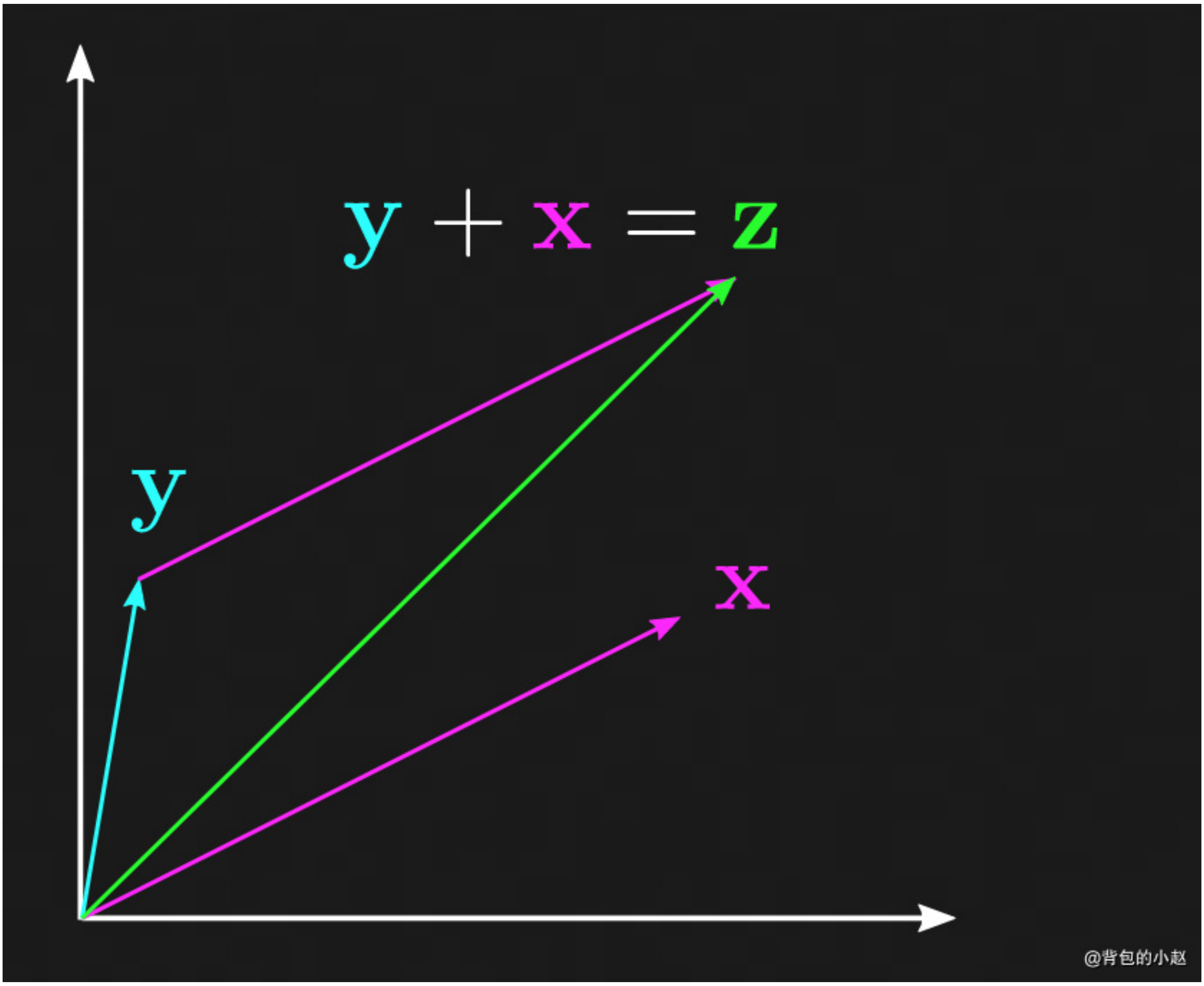
知识体系

一、向量（vectors）

1.1类型

几何向量：

我们高中时学到、见到的最多的类型，一般在二维或三维中作图就可以表示

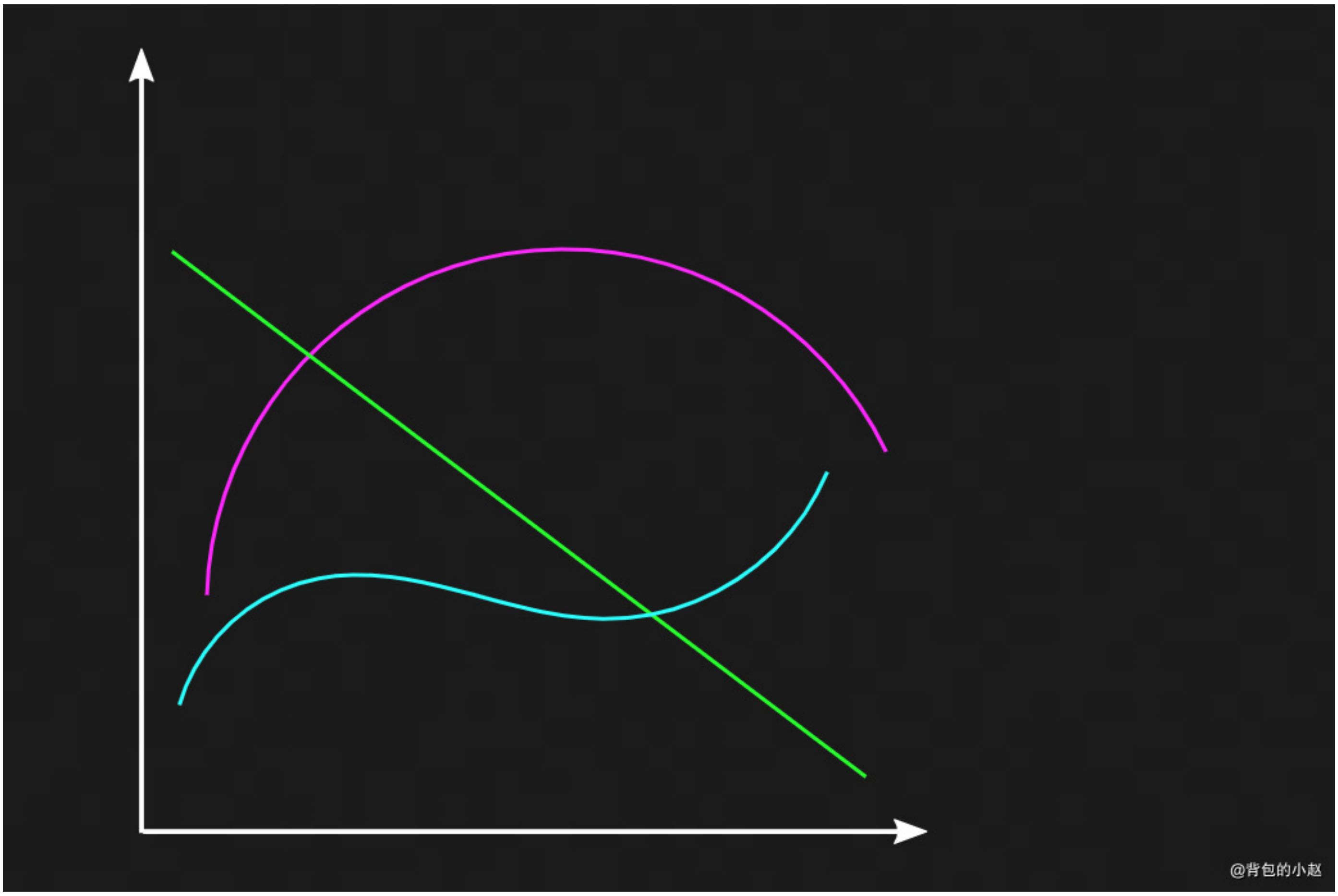


多项式：

多项式如$f(x)=x^2+y+1$，之所以也称为向量是因为它满足向量的定义：

- 可以用加法，得到新的多项式；
- 可以用乘法，得到新的多项式；

如$f(x)+g(x)$和$5\times f(x)$



实数集合：

广义地说，任意实数组成的集合$\mathbb{R}^n$也是向量，这是机器学习中最常见、最重要的向量：

$$X = \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ x_n \end{bmatrix} \in \mathbb{R}^n$$

如，一个$\mathbb{R}^3$ 3维的向量：

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \in \mathbb{R}^3$$

1.2向量的基本运算

1.2.1 加法

两个大小一样的向量的加法，需要向量的每个元素逐个相加（element-wise）：

$$x+y=\begin{bmatrix}x_1\\ .\\ .\\ .\\ x_n\end{bmatrix}+\begin{bmatrix}y_1\\ .\\ .\\ .\\ y_n\end{bmatrix}=\begin{bmatrix}x_1+y_1\\ .\\ .\\ .\\ x_n+y_n\end{bmatrix}$$

如

$$x+y=\begin{bmatrix}1\\ 2\\ 3\end{bmatrix}+\begin{bmatrix}3\\ 4\\ 5\end{bmatrix}=\begin{bmatrix}1+3\\ 2+4\\ 3+5\end{bmatrix}=\begin{bmatrix}4\\ 6\\ 8\end{bmatrix}$$

向量加法满足以下性质：

- 交换律（Commutativity）：$x+y=y+x$
- 结合律（Associativity）：$(x+y)+z=x+(y+z)$
- 零向量$\vec{0}$无效：$x+0=x$
- 减去自身等于零向量$\vec{0}$：$x-x=\vec{0}$

在`numpy`中，使用运算符`+`或`add`方法计算两个向量的和

```
import numpy as np
# 赋值x,y为向量[1,2,3]
x = y = np.array([[1],
                  [2],
                  [3]])
>>> x + y
array([[2],
       [4],
       [6]])

>>> np.add(x,y)
array([[2],
       [4],
       [6]])
```

1.2.2向量与标量相乘（vector-scalar multiplication）

向量与标量的乘法也遵循逐个元素操作的原则（element-wise）：

$$ax=\begin{bmatrix}ax_1\\ .\\ .\\ .\\ ax_n\end{bmatrix}$$

假设$a=3$，$x=[1,2,3]^T$，则

$$ax = 3\begin{bmatrix}1\\2\\3\end{bmatrix} = \begin{bmatrix}3\times 1\\3\times 2\\3\times 3\end{bmatrix} = \begin{bmatrix}3\\6\\9\end{bmatrix}$$

向量与标量的乘法满足以下性质：

- 结合律（Associativity）：$(\alpha\beta)x = \alpha(\beta x)$
- 左分配率（Left-distributive）：$(\alpha+\beta)x = \alpha x + \beta x$
- 右分配率（Right-distributive）：$x(\alpha+\beta) = x\alpha + x\beta$
- 其他：$\alpha(x+y) = \alpha x + \alpha y$

在`numpy`中，使用运算符`*`计算向量与标量乘积

```
import numpy as np

x = 3
y = np.array([
    [3],
    [4],
    [5],
    [6]
])

>>> x*y
array([[ 9],
       [12],
       [15],
       [18]])
```

1.2.3向量的线性组合

$$\alpha x + \beta y = \alpha\begin{bmatrix}x_1\\x_2\end{bmatrix} + \beta\begin{bmatrix}y_1\\y_2\end{bmatrix} = \begin{bmatrix}\alpha x_1 + \beta y_1\\\alpha x_2 + \beta y_2\end{bmatrix}$$

如 $\alpha = 2, \beta = 3, x = [2,3]^T, y = [4,5]^T$，则

$$\alpha x + \beta y = 2\begin{bmatrix}2\\3\end{bmatrix} + 3\begin{bmatrix}4\\5\end{bmatrix} = \begin{bmatrix}2\times 2 + 3\times 4\\2\times 3 + 3\times 5\end{bmatrix} = \begin{bmatrix}16\\21\end{bmatrix}$$

线性组合的另一种方式是用求和公式，假设$x_i, \ldots x_k$是一个向量，标量集合$\beta_i, \ldots, \beta_k \in \mathbb{R}$，则

$$\sum_{i=1}^{k}\beta_i x_i := \beta_1 x_1 + \ldots + \beta_k x_k$$

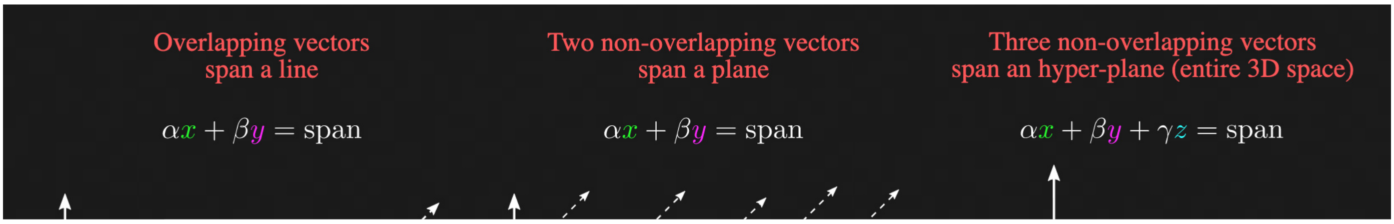
公式中:=表示“定义为”的意思。

在`numpy`中，计算向量的线性组合

```
import numpy as np
a,b = 2,3
x,y = np.array([[2],[3]]), np.array([[4],[5]])

>>> a * x + b * y
array([[16],
       [21]])
```

1.3向量空间



1.4向量子空间

1.5 向量范数（vector norms）

线性方程组（systems of linear equations）

一般线性方程组如下：

$$
\begin{aligned}
x_1 \quad & +x_2 + x_3 = 3 \\
x_1 \quad & -x_2 + 2x_3 = 2 \\
2x_1 \quad & \qquad + 3x_3 = 5
\end{aligned}
$$

将方程组系数a_{ij}挑出，组成向量（vectors）形式：

$$\begin{bmatrix} a_{11} \\ . \\ . \\ . \\ a_{m1} \end{bmatrix} x_1 + \begin{bmatrix} a_{12} \\ . \\ . \\ . \\ a_{m2} \end{bmatrix} x_2 + \ldots + \begin{bmatrix} a_{1n} \\ . \\ . \\ . \\ a_{mn} \end{bmatrix} x_n = \begin{bmatrix} b_1 \\ . \\ . \\ . \\ b_m \end{bmatrix}$$

再将向量变成矩阵（matrices）形式：

$$\begin{bmatrix} a_{11} & \ldots & a_{1n} \\ . & & . \\ . & & . \\ . & & . \\ a_{m1} & \ldots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ . \\ . \\ . \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ . \\ . \\ . \\ b_n \end{bmatrix}$$

矩阵

加法和乘法