

# 神经网络模型初探

Introduction to Neural Network Model

王(云)斐, Michael

<https://www.michaelyunfei.com>

<https://github.com/Michael-yunfei/MDLforBeginners>

May 19, 2020

# 本节内容

- ① 课程设置简介
- ② 课前预习 C1
- ③ 人工智能的思维方式
- ④ 数据结构
- ⑤ 损失方程 (Loss Function)
- ⑥ 向量, 矩阵的相关运算
- ⑦ 卷积运算 (Convolution)

# 课程进度和相关考核

第三章内容跨度较大，我们分拆成两章来进行：

- 辅导教义的第三章为本节课的讲解内容（教材 44-56 页）
- 辅导教义的第三章为下周教授的内容（教材第 3 章余下内容）

辅导讲义的组成部分：

- 课前预习 C1
- 课堂讲义 C2
- 课后练习 C3

认真完成以上三个内容可以得到 A, 只完成 C1-C2 可以及格。根据自己的兴趣偏好和设定的目标合理安排学习时间（毕竟你不只是学习人工智能一门学科）。

# 课程进度和相关考核

考核机制的设计如下：

- 课后练习 C3 - 10%
- 期中考试 - 20 %
- 期末考试 - 30 %
- 案例报告 - 40 %

具体的考试形式和报告内容要求，后面会有具体的指示。

# 课前预习题目浅析

虽然科技已经飞速进步，但是人类至今仍然不能领会大脑运行的奥秘。但是对于记忆机制和神经触点已经有所了解。从生物意义上讲，

- 大脑是通过神经触点的关联来进行思考和指令下达的 - 脑电图

从哲学角度上看，我们的知识则是概念的关联和堆积：

- 大卫·休谟 (David Hume)
- 伊曼努尔·康德 (Immanuel Kant)

Geoffrey Hinton (杰弗里·辛顿) 等人正是在巨人的肩膀上进行最新的探索的。David Hume (大卫·休谟) 在剑桥和牛津是必修。其《人类理解研究》(An Enquiry Concerning Human Understanding) 的两章为：

- Of the origin of ideas (观念的起源)
- Of the association of ideas (观念的关联)

# 课前预习题目浅析

排列组合题目：

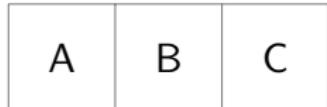
0/1	2		
0/1	0/1	$2^2$	
0/1	0/1	0/1	$2^3$

给定三个空位所能包含的信息单元总数为:  $2 + 2^2 + 2^3 = 14$ 。我们知道十进制是满 10 进一个位，二进制则是满 2 进一个空位。

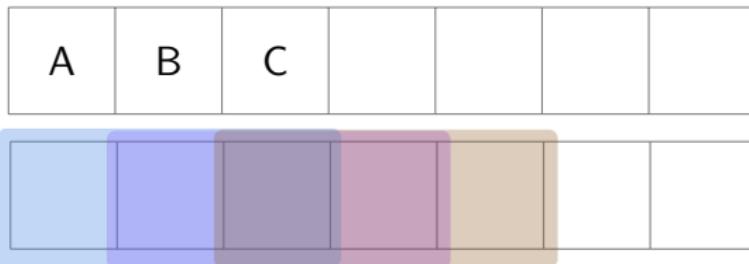
```
1 comper(32, 2) # 8589934590
comper(64, 2) # 36893488147419103230
3 comper(32, 4) # 24595658764946068820
comper(64, 4) # 453709822561251284617832809909024281940
```

# 课前预习题目浅析

排列组合题目 (捆绑法):



$$(7 - 3) + 1 = 5$$

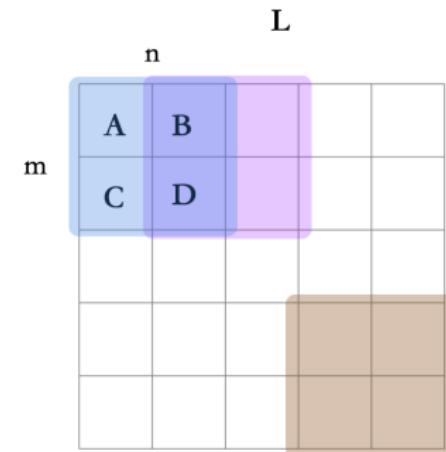


给定参数向量  $A$  长度为  $m$ , 数据向量  $B$  长度为  $n$ , 其中  $n > m$ , 我们共有的排列组合方式: $n - m + 1$

# 课前预习题目浅析

## 排列组合题目 (捆绑法):

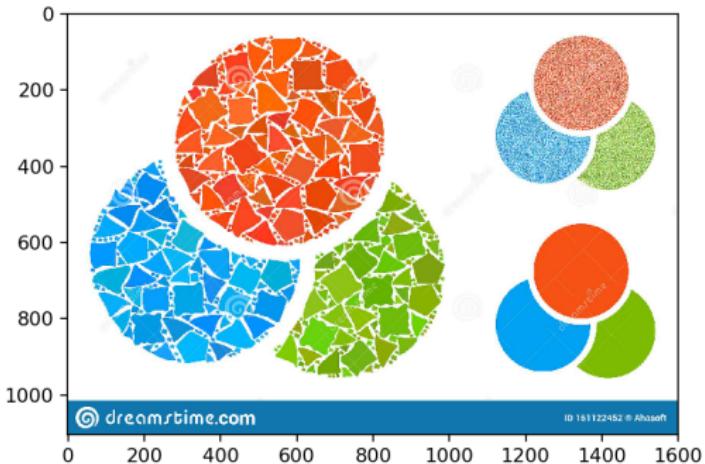
```
def matrixConv(kenl, dtm):
    # Assume size of kernel is less than datamatrix
    # 假设参数核矩阵小于数据矩阵
    m = np.shape(kenl)[0]
    n = np.shape(kenl)[1]
    k = np.shape(dtm)[0]
    l = np.shape(dtm)[1]
    resltmtx = np.zeros([k-m+1, l-n+1])
    for i in range(resltmtx.shape[0]):
        for j in range(resltmtx.shape[1]):
            temp = 0
            for u in range(m):
                for p in range(n):
                    temp += kenl[u, p] * dtm[u+i, p+j]
            resltmtx[i, j] = temp
    return(resltmtx)
```



给定参数核矩阵  $\mathbb{A}$ , 大小为  $m \times n$ , 数据矩阵  $\mathbb{B}$ , 大小为  $K \times L$ , 其中  $K > m, L > n$ , 我们可以得到的结果矩阵大小为  $(K - m + 1) \times (L - n + 1)$ 。

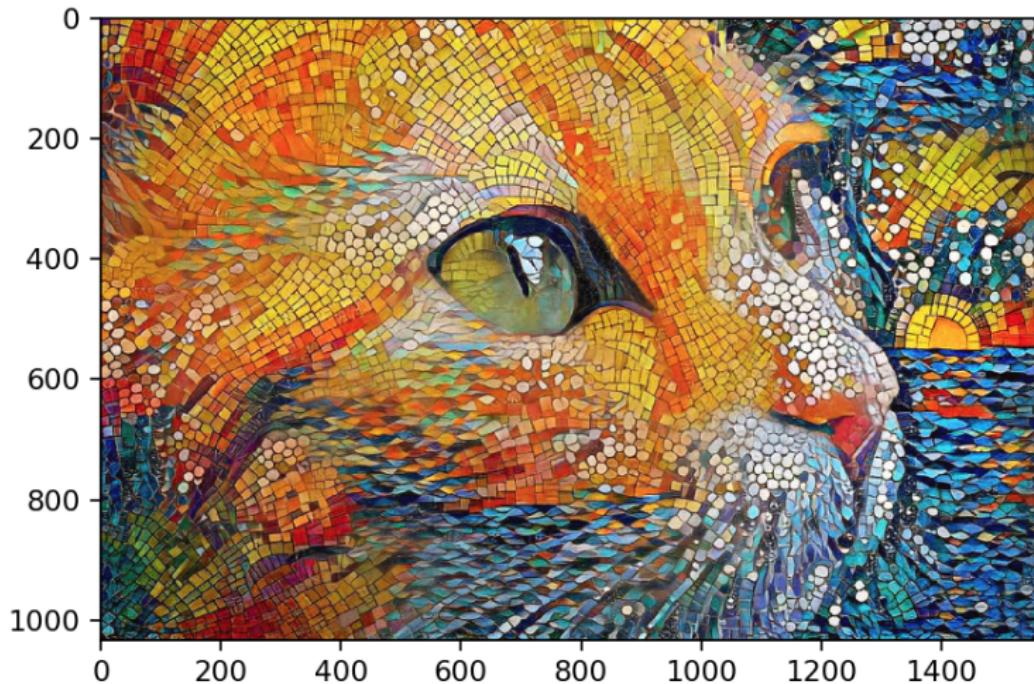
# 三色图结构

- 任何颜色都可以有红绿蓝 (R, G, B) 三色来调和构成
- 图像维度 (1104, 1600, 3)
- 注意右图的坐标



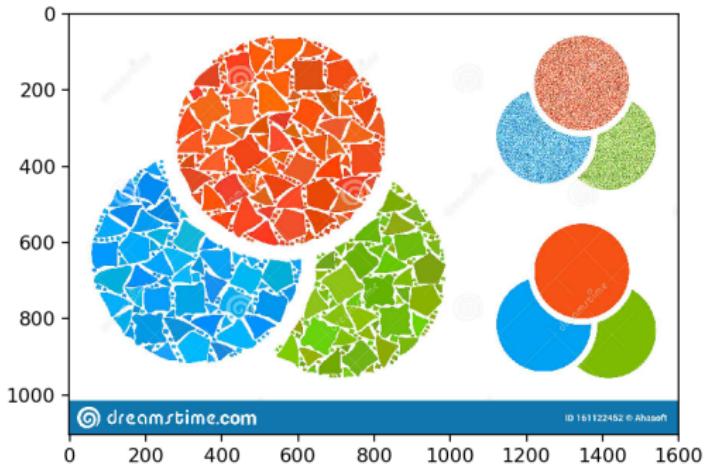
# 三色图结构

如何成为一个马赛克绘画艺术家?

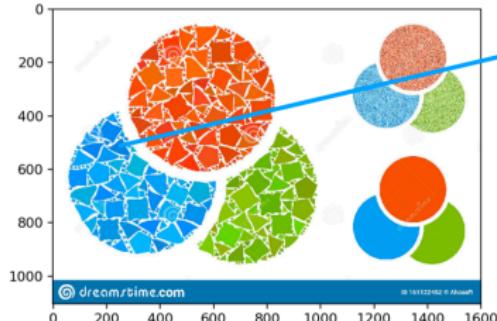


# 三色图结构

- 红-255, 绿-255, 蓝-255
- 图像维度 (1104, 1600, 3)
- (500 : 520, 600 : 620, 0)
- (500 : 520, 600 : 620, 1)
- (500 : 520, 600 : 620, 2)
- (500 : 520, 230 : 250, 0)
- (500 : 520, 230 : 250, 1)
- (500 : 520, 230 : 250, 2)



# 三色图结构



```
[224 115 46 20 4 3 0 0 0  
 4 4]  
[ 68 35 28 13 6 2 0 0 0  
 1 1]  
[ 33 16 12 2 0 0 0 0 0  
 0 0]  
[ 3 16 0 0 0 0 0 4 5  
 0 0]  
[ 5 0 0 0 0 3 9 16 18  
 0 0]  
[ 0 0 0 0 0 0 9 20 32]
```

(500:520, 230:250, 0)

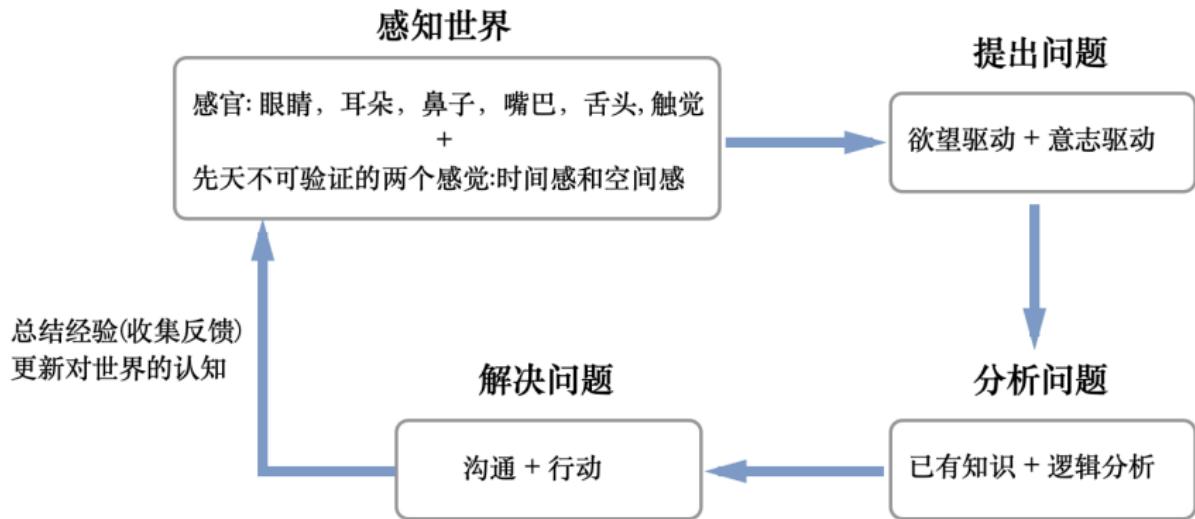
```
[255 166 115 109 115 134 147 147  
134 134]  
[127 111 120 123 132 143 145 141  
135 136]  
[121 119 130 129 132 135 136 137  
138 138]  
[118 145 136 135 136 137 136 135  
140 140]  
[143 135 143 145 145 142 137 132  
142 141]
```

(500:520, 230:250, 1)

```
[255 245 208 211 220 238 247 242  
232 230]  
[205 207 229 238 245 251 246 232  
230 228]  
[208 220 240 244 246 243 235 227  
226 226]  
[207 246 241 243 241 237 232 223  
225 225]  
[231 229 241 243 241 237 228 219  
223 225]  
[224 233 227 233 234 226 219 214  
223 223]
```

(500:520, 230:250, 2)

# 学习过程的一般性原理



# 人工智能取代人？

- 需求端 (demand): 我要，我要，我就要
- 供求端 (supply): 我能，我能，我很能



# 读点历史没坏处

- 需求端 (demand): 我要，我要，我就要 (1949-1979)
- 供求端 (supply): 我能，我能，我很能 (1979-2009)
- 70% 服务业 (13 亿工程师和全民皆兵的区别? e.g. 二战前的德国)
- 文官治国 (所以军队是得配个政委的)



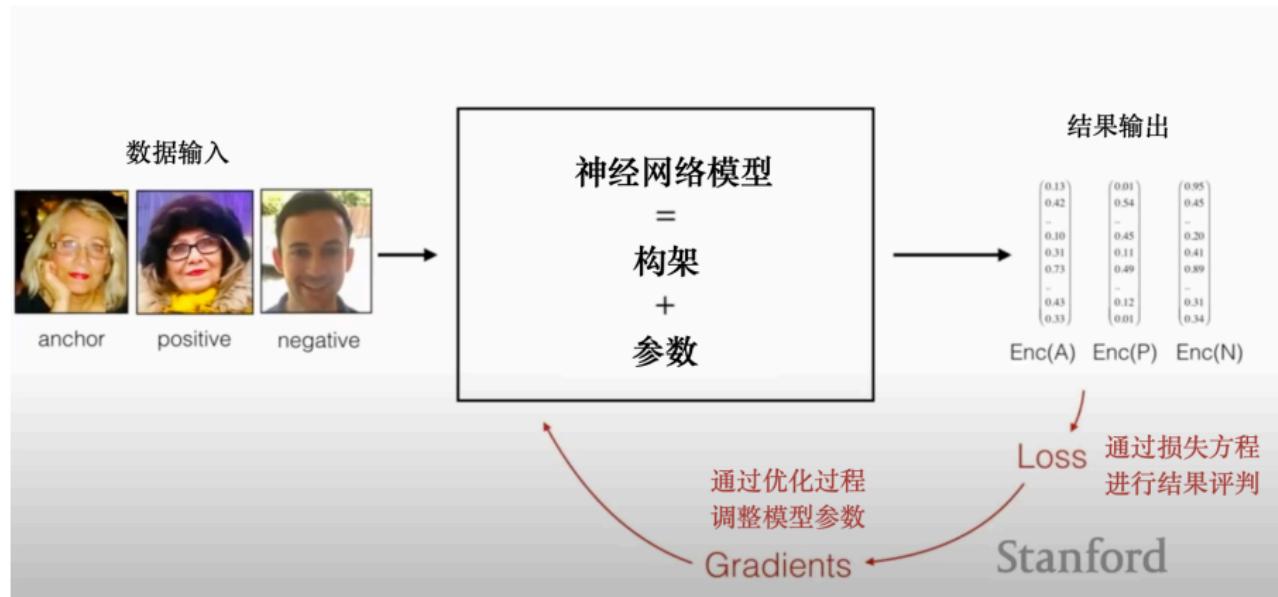
# 人工智能的思维方式

传统的问题解决方式和人工智能问题解决方式最大的不同体现在以下两个方面：

- 传统的问题解决方式着重对问题进行分步骤拆解和处理；
- 人工智能对问题的解决方式着重对整个流程的两个端口 进行数据和模型调试。

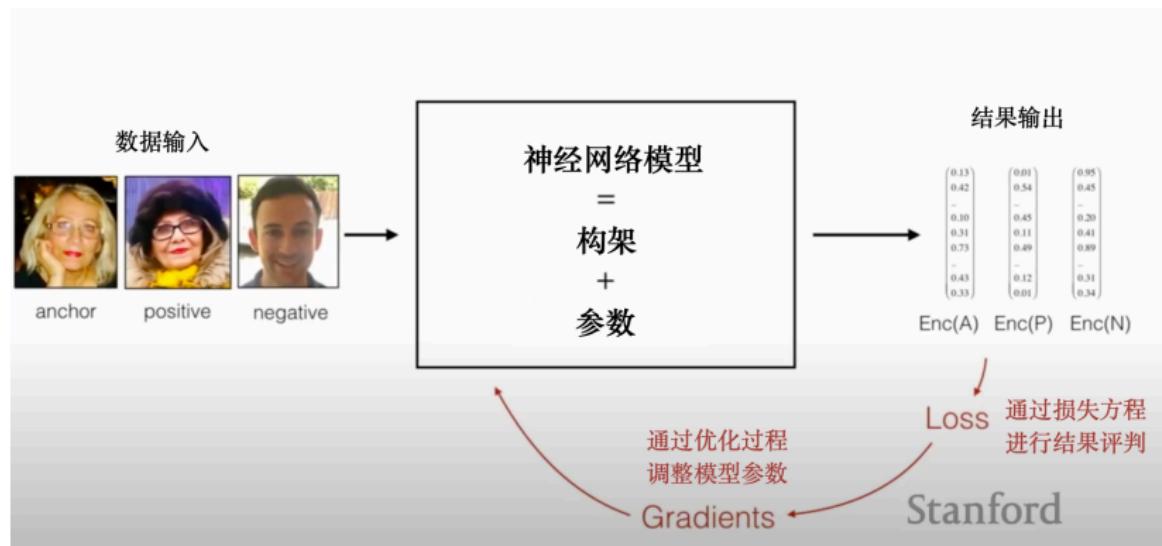
# 人工智能的思维方式

教材中有两张图片，即图 3-15 和图 3-19，给出了神经网络模型的工作流程。我们把两个图片中的模型进行简化，绘制下面这张流程图（来源：斯坦福大学深度学习官网）。



# 人工智能的思维方式

机器学习(人工智能): 通过对输入端的数据群(大数据)进行标注和整理后, 建立一个人工智能构架, 并设定目标函数(损失方程), 对所建立的构建(或模型)进行反复训练, 直到找到符合我们目标函数要求的参数核矩阵。



# 数据结构定义

因为机器学习和人工智能需要大量的数据，所以非常有必要对数据结构进行定义和分类。生活中最常见的数据形式应该是 Excel 表格。

## Definition

我们把由数字和对其的描述（有其根据）组成的信息形态，称为数据；我们将统一在一个描述下的数列，称为数据集；有多个数据集构成的一系列数据，称为数据阵。

姓名	身高	性别	成绩
Michael	176	1	89
Ani	163	0	93
Marina	161	1	82
Emmet	168	1	76
Lucy	165	0	67

# 数据结构定义

## Definition

我们把语言文本，网页信息，图片和影音等的复合数据，称为多媒体数据集。

## Definition

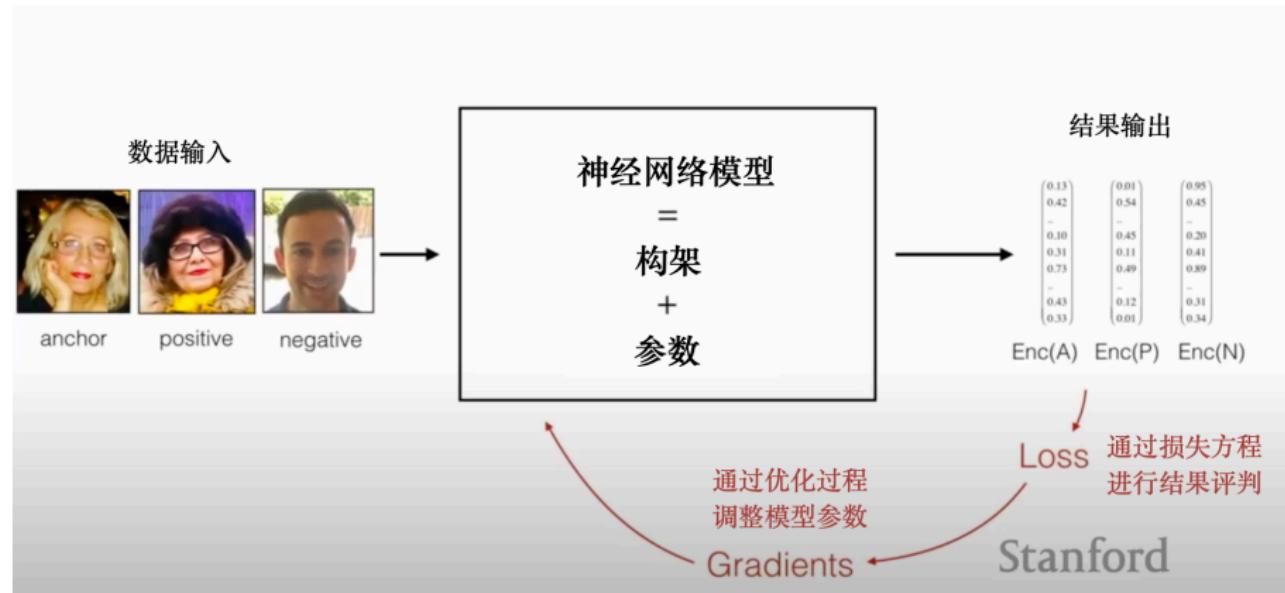
我们将多媒体数据集和数据阵组合构成的数据形态成为数据群或者大数据。数据群或大数据，在业界也被称为**有标注的数据群**。

比如，下面这个数据群，就是一个有标注的数据群。

图片 (1400, 1400, 3)	年份	地点	备注 1	备注 2
图 1	2016	桥	青岛	红色
图 2	2020	伦敦	人	女
图 3	2013	成都	熊猫	国宝

# 数据结构定义

多数情况下，机器学习需要有标注的数据群（大数据）。模型的训练和数据的质量存在正相关关系。



# 损失方程

拥有了标注的数据，就有了对比依据<sup>1</sup>，就可以训练人工智能模型，从而可以调整模型参数，来提升预测准确度。而用来衡量准确度的方程就需要损失方程。

## Definition

假设数据群标注过的数据准确可靠，在人工智能模型中，模型在整合输入数据且加工计算后输出的结果与数据群标注的数据差别的方程称为损失方程 (*Loss Function*)。损失方程的一般形式为：

$$L : (Y, \hat{Y}) : \rightarrow \mathbb{R}$$

---

<sup>1</sup>对，没有对比就没有伤害。

# 损失方程

比如我们想要通过人工智能来判定图片中人类的性别，下面的表格中给出了最为简单的一种损失方程。

图片 (1400, 1400, 3)	标注 (1-男,0-女)	模型预测	差异
图 1	1	0	1
图 2	0	1	1
图 3	1	1	0

用  $\hat{Y}$  来代表预测的结果， $Y$  来代表标注结果，那么损失方程为：

$$L(Y, \hat{Y}) : |Y - \hat{Y}|$$

# 向量和矩阵

无论是多媒体数据集还是数据阵，除去其标注的信息，具体的数都是储存在矩阵中的。

## Definition

矩阵（Matrix）是一个按照长方阵列排列的复数或实数集合。对一个矩阵的简单描述可以为  $m \times n$  的矩阵，即该矩阵有  $m$  行， $n$  列。 $m \times n$  也被称为矩阵的大小。

## Example

比如，我们有  $3 \times 3$  的矩阵  $A$  和  $2 \times 3$  的矩阵  $B$ 。

$$A = \begin{bmatrix} 1 & 3 & 4 \\ 2 & 5 & 8 \\ 3 & 9 & 7 \end{bmatrix}, \quad B = \begin{bmatrix} 3.2 & 1.9 & 5.7 \\ 4.1 & 5.8 & 9.3 \end{bmatrix}$$

# 向量和矩阵

## Definition

矩阵中单独的一行被成为行向量 (row vector), 单独的一列别成为列向量 (column vector)。

## Example

比如在上面的矩阵  $A$  中, 我们有以下单独的向量:

$$A_{r2} = [2 \ 5 \ 8], \quad A_{c2} = \begin{bmatrix} 3 \\ 5 \\ 9 \end{bmatrix}$$

# 矩阵运算

## Definition

现有同样大小的  $m \times n$  的矩阵  $A, B$ , 我们定义矩阵的加法 (+) 运算为:

$$A + B = \begin{bmatrix} (a_{11} + b_{11}) & (a_{12} + b_{12}) & \cdots & (a_{1n} + b_{1n}) \\ a_{21} + b_{21} & a_{22} + b_{22} & \cdots & a_{2n} + b_{2n} \\ \vdots & \ddots & & \vdots \\ a_{m1} + b_{m1} & a_{m2} + b_{m2} & \cdots & a_{mn} + b_{mn} \end{bmatrix}$$

其中,  $a_{ij}$  为矩阵  $A$  中第  $i$  行第  $j$  列元素,  $b_{ij}$  为矩阵  $B$  中第  $i$  行第  $j$  列元素。

注意: 矩阵的加法只能在同样大小的矩阵中进行, 即同位元素相加。

# 矩阵运算

矩阵最早的提出是为了解决多项式的求解，比如下面方程组的转换。

$$2x_1 + 3x_2 + 7x_3 = 9$$

$$4x_1 + 5x_2 + 8x_3 = 10$$

$$\begin{bmatrix} 2 & 3 & 7 \\ 4 & 5 & 8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 9 \\ 10 \end{bmatrix}$$

## Definition

现有  $m \times n$  的矩阵  $A$  和  $n \times k$  的矩阵  $B$ , 我们定义矩阵的乘法 · 为

$$(A \cdot B)_{ij} = \sum_{r=1}^n a_{ir} b_{rj} = a_{i1} b_{1j} + a_{i2} b_{2j} + \cdots + a_{in} b_{nj}$$

其中  $(AB)_{ij}$  是两个矩阵相乘后的第  $i$  行第  $j$  列元素。

# 矩阵运算

注意：取任意  $a \times b$  矩阵 A 和  $c \times d$  矩阵 B，如果  $b \neq c$ ，则两个矩阵不可以相乘。

## Example

计算下列矩阵的乘法

$$\begin{bmatrix} 2 & 3 & 7 \\ 4 & 5 & 8 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} = ? \quad \begin{bmatrix} 1 & 3 & 3 \\ 2 & 8 & 4 \\ 3 & 9 & 1 \end{bmatrix} \begin{bmatrix} 3 & 2 \\ 1 & 0 \\ -1 & 8 \end{bmatrix} = ?$$

# 卷积 (Convolution)

卷积运算 (Convolution) 是图形处理中特有的一种运算方式。其目的是系统性一次性得对原始图片进行快速处理。我们之前已经讲解过图形的储存形式，而且在课前预习中的习题中也特别对卷积运算进行了铺垫，接下来我们就更加规范得对这个概念进行学习。

# 向量卷积 (Vector Convolution)

## Definition

给定一个长度为  $m$  的向量  $A$  和一个长度为  $n$  的向量  $B$ , 其中  $n > m$ :

$$A = [a_1, a_2, \dots, a_m]$$

$$B = [b_1, b_2, \dots, \dots, b_n]$$

我们称  $A$  为参数 (核) 向量 (kernel vector),  $B$  为数据向量, 其卷积运算 (convolution)\* 定义为

---

### Algorithm 1: Convolution Operation

---

初始化长度为  $n - m + 1$  的向量  $C = [c_1, c_2, \dots, c_{n-m+1}]$  ;

$C = A * B$  ;

其中, 每一个元素的计算方法为:

**for**  $i$  in  $[1 : n - m + 1]$  **do**

$$| \quad c_i = \sum_{k=1}^m a_k b_{k+i-1} = a_1 b_i + a_2 b_{i+1} + \dots + a_m b_{i+m-1}$$

**end**

---

# 向量卷积 (Vector Convolution)

## Example

现有参数向量  $A = [1, 3, 4]$  和数据向量  $B = [5, 6, 7, 8, 9]$ , 那么

$$A * B = [34, 40, 46]$$

$$34 = 1 \times 5 + 3 \times 6 + 4 \times 7$$

$$40 = 1 \times 6 + 3 \times 7 + 4 \times 8$$

# 矩阵卷积 (Matrix Convolution)

## Definition

给定一个  $m \times n$  的矩阵  $A$ , 和一个  $k \times l$  的矩阵  $B$ , 并且  $k > m, l > n$ , 那么矩阵  $A$  被叫作参数 (核) 矩阵 (kernel matrix), 矩阵  $B$  被称为数据矩阵。那么矩阵  $A$  和  $B$  的卷积运算定义为

$A * B =$  移动参数向量对数据向量进行对位求积后求和

运算过程如下图所示 (来源: Irhum Shafkat, 2018)。

矩阵的卷积有很多种, 更多的介绍请参考该网站:

<https://towardsdatascience.com/intuitively-understanding-convolutions-for-deep-learning-1f6f>

# 矩阵卷积 (Matrix Convolution)

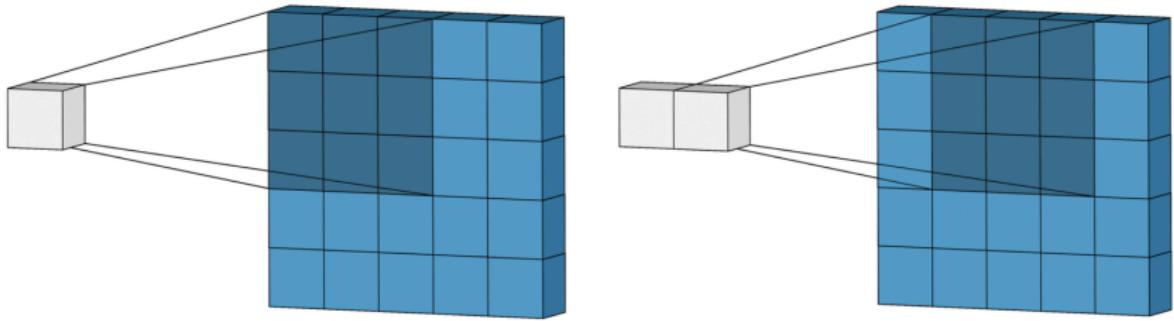


Figure 1: 矩阵卷积演示 (Irhum Shafkat, 2018)

# 矩阵卷积 (Matrix Convolution)

矩阵的卷积运算目的是想要通过参数矩阵来对数据矩阵进行进一步的处理。之所以需要这个处理过程是因为

- 原始多媒体数据集维数较大，比如图 3.4 中的猫有  $(1400, 1400, 3)$ ，即  $1400 \times 1400 \times 3 = 588000$  个数（元素）组成。通过卷积运算后数据集的维数会有所下降，从而节省模型训练时间；
- 原始多媒体数据集的特征不明显，需要通过参数矩阵的卷积运算后来进行优化；
- 卷积在图像识别中的应用原因还有其它的，以后我们会结合具体案例进一步讲解。卷积运算主要应用在图像识别中，在其它领域，比如金融数据的人工智能模型，则没有被那么广泛得使用

# 矩阵卷积 (Matrix Convolution)

## Example

我们手动计算一种标准的卷积形式，给定参数核矩阵  $A$  和数据矩阵  $B$  为

$$A = \begin{bmatrix} 1 & -1 \\ 1 & 2 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 3 & 6 \\ 0 & 5 & 7 \end{bmatrix}$$

$C = A * B$  的最终结果为

$$C = \begin{bmatrix} 7 & 16 \\ 9 & 16 \end{bmatrix}$$

你会发现  $C$  中元素最大值是 14，刚好在  $B$  的右下角。我们演示计算结果矩阵 ( $C = A * B$ ) 的第一个元素：

$$C_{11} = 1 \times 1 + (-1) \times 2 + 1 \times 2 + 2 \times 3 = 7$$

# 卷积案例

下面的表格是一个班级内几名同学的不同学科的成绩表，我们来计算不同的参数卷积矩阵可以给出不同的‘评估’结果（这个评估结果可以理解成我们机器学习的一个运算结果）。

Arts	姓名	语文	数学	英语	物理	历史	人工智能
	Michael	90	60	80	60	100	70
	Ani	90	60	80	60	100	70
	Marina	90	60	80	60	100	70
	Emmet	90	60	80	60	100	70
	Lucy	90	60	80	60	100	70
	Kara	90	60	80	60	100	70
Science	姓名	语文	数学	英语	物理	历史	人工智能
	Michael	60	90	60	80	70	100
	Ani	60	90	60	80	70	100
	Marina	60	90	60	80	70	100
	Emmet	60	90	60	80	70	100
	Lucy	60	90	60	80	70	100
	Kara	60	90	60	80	70	100

# 卷积案例

其中参数核 (kernel) 矩阵给定为

$$K = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

将文科生 (arts) 和理科生 (science) 的数据阵进行两次卷积运算：

$$A \Rightarrow \begin{bmatrix} 30 & 0 & -60 & -30 \\ 30 & 0 & -60 & -30 \\ 30 & 0 & -60 & -30 \\ 30 & 0 & -60 & -30 \end{bmatrix} \Rightarrow \begin{bmatrix} 270 & 90 \\ 270 & 90 \end{bmatrix} \quad S \Rightarrow \begin{bmatrix} 0 & 30 & -30 & -60 \\ 0 & 30 & -30 & -60 \\ 0 & 30 & -30 & -60 \\ 0 & 30 & -30 & -60 \end{bmatrix} \Rightarrow \begin{bmatrix} 90 & 270 \\ 90 & 270 \end{bmatrix}$$

两个矩阵看起来很相似，只是列向量顺序不同。

# 卷积案例

如果我们将两个矩阵按照此前的顺序进行标注，可以得出下面的表格：

学生/成绩集成	文科成绩	理科成绩
文科生	270 270	90 90
理科生	90 90	270 270

复杂神经学习模型要比以上的例子复杂的多，但是基本流程却非常相似，具体的计算和演化逻辑也可以说‘如出一辙’。我们可以把神经网络深度学习的模型再一次概况为：通过对输入端的数据群（大数据）进行标注和整理后，建立一个人工智能构架，并设定目标函数（损失方程），对所建立的构建（或模型）进行反复训练，直到找到符合我们目标函数要求的参数核矩阵。

# 再见

Thank you ! 请完成课后习题 C3 !



Figure 2: C3C1-问卷



Figure 3: C3C2-问卷

# Reference I

I will add reference list later, in which only two websites should be listed.