作者:
李明明

该书关于

# 深度学习

深度学习的理解及笔记

*2020* 年 *10* 月 *21* 日

# Dedication

该书关于对机器学习和机器学习的理解。

对我影响最大的为 Goodfellow 的《DEEP LEARNING》, 该书从理论上深刻的诠释了机器学习和机器学习，从中受益匪浅，该书中部分《DEEP LEARNING》就是对该书的翻译。

《deep learning for computer vision with python starter》,《deep learning for computer vision with python practitioner》与《deep learning for computer vision with python imagenet》这三本书从实践的角度来专门讲述计算机视觉。

《Case Studies》是 OpenCV 的基础, 可以用来入门 OpenCV。

李航的《统计学方法》中的概率的部分讲的也挺好。

# 网址

https://chyson.net

https://chyson.fun

# Contents

# 第一章　介绍

在早期的人工智能中，我们处理和解决对人类来说难处理，但对机器来说很简单的，可以用正式的数学规则描述的问题。而人工智能真正的挑战是那些对人类来说很简单，但难以正式的描述出的问题，比如识别说出的话或者图片中的人脸。

解决办法就是让机器从数据中进行学习，并以结构化的概念来理解这个世界，每个概念通过与更简单的概念的关系来定义。如果我们画一个图来描述这些概念是如果在其他概念上构成的，该图就会很深，有很多层。因此，也叫做深度学习。

最早的人工智能解决方式为：硬编码知识（hard-code knowledge）。这种方式有一个知识库和一个推理机，推理机由一系列逻辑推理规则组成，（人为设定）通过将推理机应用与知识库来对世界进行理解，这种方式未取得很大的成成功。

hard-code knowledge 面临的难题是，AI 系统需要有能力来获取知识，通过从原始数据中提取模式（pattern）。这种能力就是机器学习。一个简单的机器学习的算法为逻辑回归。

这些简单的机器学习算法的效率很大程度上依赖于数据的表征(representation),表征中的每一个信息都称为特征(feature),比如 [身高，体重，年龄，血压，心率...]，表征的选择对机器学

习的效率有巨大的影响。

然而对于许多任务来说，我们很难知道需要提取什么特征。比如，我们想识别图片中的汽车，我们可以使用汽车有轮子这个特征，但我们很难用像素来描述轮子，一个轮子有几何形状，但照片中的轮子受到光照，阴影，挡板和前景的障碍物的遮挡等等的影响。

一个解决方法为，使用机器学习来同时学习表征到输出的映射和表征本身，这个方法叫做“表征学习”。学习得到的表征常常比手动设计的表征的准确率要高。表征学习的经典算法为自编码器（autoencoder），自编码器由编码函数（将输入数据转化为不同的表征）和解码函数（将新的表征转化为原始数据格式）组成。

当我们设计特征或者算法来学习特征的时候，我们的目标通常是将变量的因素（factors of variation）进行分离，而这因素可以解释观察到的数据。在此处，因素指单独的影响源，该因素不是混合的，它们可以被想象为我们用来理解数据中的复杂变化性的一种概念或者抽象，例如，当我们分析一段录音，因素包括说话这的年龄，性别，口音和说话的内容等，当我们分析汽车的照片，因素有汽车的位置，颜色，角度和太阳的亮度等。

一个主要的问题为，许多变量的因素同时影响着我们观测到的数据中的所有单个部分。比如，红色轿车图片中的色素在晚上的时候非常接近黑色，阳光对整张图有着影响。

当然，我们很难从数据提取出高层次的抽象特征，比如口音，这需要对数据有专业的，人类水平的理解。当获取表征的难度和解决问题的难度相似的时候，表征学习似乎对我们是没有帮助的。

深度学习通过引入使用更简单的表征表示的表征来解决表征学习中的核心问题。深度学习是机器可以从更简单的概念中构建出更复杂的概念。经典的深度学习算法为，“前馈深度网络”（feedforward deep network）或者多层感知机（multilayer perception）。

从基于规则的系统 → 机器学习 → 表征学习 → 深度学习的进化如图1.1所示：

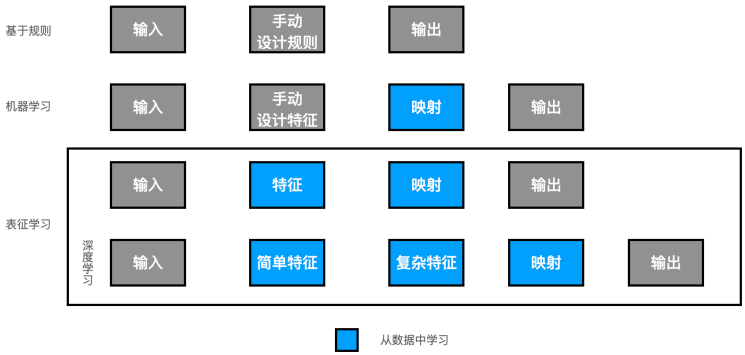


Figure 1.1: 深度学习的进化

# 第一部分

# 数学

# 第二章　线性代数

线性代数是数学的一个分之，它主要处理线性关系问题，即数学对象之间的关系以一次形式表达的，它的主要研究对象为张量（tensor）即张量空间。

## 2.1　标量，向量，矩阵和张量

### 2.1.1　标量

单个数就是标量，比如 3，5 等。

### 2.1.2　向量

一组数就是向量，比如

$$\begin{bmatrix} 1 \\ 2 \\ 5 \\ 3 \end{bmatrix} \tag{2.1}$$

### 2.1.3 矩阵

矩阵就是 2 维数组，每个元素使用两个索引表示，比如

$$\begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix} \tag{2.2}$$

$A_{1,1}$ 表示矩阵中的第一行和第一列处的元素。

### 2.1.4 张量

当需要大于两个的数组表示的时候，就是张量，张量中的元素需要用 $\geq 2$ 的索引来表示。

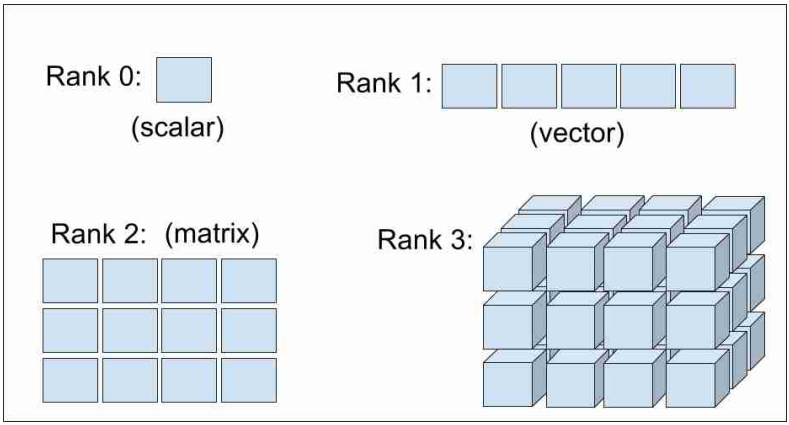


Figure 2.1: 标量，向量，矩阵和张量

### 2.1.5 转置 (transpose)

转置是矩阵的一种运算，即更换行和列。我们用 $A^\top$ 来表示矩阵 $A$ 的转置，则

$$(\boldsymbol{A}^\top)_{i,j} = A_{j,i} \tag{2.3}$$

向量可以看作为只含有一列的矩阵，向量的转置是只含有一行的矩阵。

标量可以当作只含有一个元素的矩阵，因此，标量的矩阵就是标量本身。

### 2.1.6 矩阵加法

$$\boldsymbol{C} = \boldsymbol{A} + \boldsymbol{B} \tag{2.4}$$

这里 $C_{i,j} = A_{i,j} + B_{i,j}$，即如果两个矩阵的形状相同，矩阵的加法就是将对应索引的元素进行相加。

### 2.1.7 矩阵的与标量的运算

矩阵和一个标量进行相乘，相当于标量乘矩阵的所有元素，矩阵和标量相加，相当于矩阵中的所有元素都加该标量，即

$$\boldsymbol{D} = a \cdot \boldsymbol{B} + c \tag{2.5}$$

这里 $D_{i,j} = a \cdot B_{i,j} + c$。

### 2.1.8 矩阵与向量的运算

在深度学习中，我们常常使用更简化的表示方式，允许将一个矩阵和一个向量进行相加，即

$$\boldsymbol{C} = \boldsymbol{A} + \boldsymbol{b} \tag{2.6}$$

这里 $C_{i,j} = A_{i,j} + b_j$，即向量 $b$ 被加到了矩阵的所有的列。

## 2.2 矩阵乘法

矩阵 $\boldsymbol{A}$ 与矩阵 $\boldsymbol{B}$ 相乘的前提是，$\boldsymbol{A}$ 的列数与 $\boldsymbol{B}$ 的行数要相同。假设 $\boldsymbol{A}$ 的形状为 $m \times n$，$\boldsymbol{B}$ 的形状为 $n \times p$，那么 $\boldsymbol{C}$ 的形状则为 $m \times p$。

$$\boldsymbol{C} = \boldsymbol{AB} \tag{2.7}$$

这里

$$C_{i,j} = \sum_k A_{i,k} B_{k,j} \tag{2.8}$$

换句话说，$\boldsymbol{A}$ 的第 $i$ 行乘 $\boldsymbol{B}$ 的第 $j$ 列，相乘的结果相加，得到 $\boldsymbol{C}$ 的索引为 $(i, j)$ 的元素。

### 2.2.1 分配律 (distributive)

$$\boldsymbol{A}(\boldsymbol{B} + \boldsymbol{C}) = \boldsymbol{AB} + \boldsymbol{AC} \tag{2.9}$$

### 2.2.2 结合律 (associative)

$$\boldsymbol{A}(\boldsymbol{BC}) = (\boldsymbol{AB})\boldsymbol{C} \tag{2.10}$$

### 2.2.3 交换律 (commutative)

矩阵乘法不满足交换律。

### 2.2.4 矩阵乘法的转置

$$(\boldsymbol{AB})^\top = \boldsymbol{B}^\top \boldsymbol{A}^\top \tag{2.11}$$

## 2.3 线性方程的矩阵表达

假设有如下线性方程组:

$$\begin{cases} a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,n}x_n = b_1 \\ a_{2,1}x_1 + a_{2,2}x_2 + \cdots + a_{2,n}x_n = b_2 \\ \qquad\qquad \cdots \\ a_{m,1}x_1 + a_{m,2}x_2 + \cdots + a_{m,n}x_n = b_m \end{cases} \tag{2.12}$$

线性方程组中，所有的系数 $a$ 组成一个矩阵 $\boldsymbol{A}$:

$$\boldsymbol{A} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ & & \cdots & \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix} \tag{2.13}$$

方程组中所有等号右边的数组成一个向量 $\boldsymbol{b}$:

$$\boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ \cdots \\ b_n \end{bmatrix} \tag{2.14}$$

所有的待求系数 $x$ 组成向量 $\boldsymbol{x}$:

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{bmatrix} \tag{2.15}$$

根据矩阵乘法定义，则线性方程组可以写为:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \tag{2.16}$$

## 2.4 单位矩阵 (identity matrix)

单位矩阵就是，当单位矩阵和一个向量相乘时，结果还是该向量本身，即

$$\forall \boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{I}_n = \boldsymbol{x}. \tag{2.17}$$

有定义可知，单位矩阵主对角线上的元素都为 1，其余都为 0，如：

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.18}$$

表示 $\boldsymbol{I}_3$。

## 2.5 逆矩阵 (inverse matrix)

我们用 $\boldsymbol{A}^{-1}$ 来表示矩阵 $\boldsymbol{A}$ 的逆矩阵，定义如下：

$$\boldsymbol{A}^{-1}\boldsymbol{A} = \boldsymbol{I}_n \tag{2.19}$$

## 2.6 扩张空间 (span)

对于(2.16)所示的矩阵表达的线性方程组的解，将矩阵 $\boldsymbol{A}$ 的每一列当作一个向量，表示从原点指向特定的方向，则方程组的解的个数表示有多少种方式到达 $\boldsymbol{b}$。从这个观点来看，$\boldsymbol{x}$ 的每个

元素表示在每个方向上走多远，$x_i$ 表示在 $i$ 列方向上走多远：

$$\boldsymbol{Ax} = \sum_i x_i \boldsymbol{A}_{:,i}. \tag{2.20}$$

(2.20)表示的操作称为“线性组合”。

正式的，一组向量集 $\{v^{(1)}, ..., v^{(n)}\}$ 的线性组合通过将每一个向量乘一个标量得到：

$$\sum_i c_i \boldsymbol{v}^{(i)} \tag{2.21}$$

一组向量的“扩展空间”是通过对该组向量的线性组合而可以达到的所有点的集合。

## 2.7 线性相关，线性无关

对于一组向量组成的集合，如果集合中的任一向量都无法用其余向量的线性组合进行表示，则该组向量是线性无关的，否则为线性相关的。

## 2.8 模

在机器学习中，我们用模（norm）来表示向量的大小。

常用的 $L^p$ 模定义如下：

$$||\boldsymbol{x}||_p = \left( \sum_i |x_i|^p \right)^{\frac{1}{p}} \tag{2.22}$$

对 $p \in \mathbb{R}, p > 1$。

模，包括 $L^p$ 模，是这样一类函数，映射向量到非负数。更严格的讲，模函数需要满足如下特性：

- $f(x) = 0 \Rightarrow x = 0$
- $f(x + y) \leq f(x) + f(y)$
- $\forall \alpha \in \mathbb{R}, f(\alpha x) = |\alpha| f(x)$

其中 $L^2$ 模也常称为欧式模。

机器学习中常用的一个模为 $L^\infty$ 模，也叫最大模：

$$||\boldsymbol{x}||_\infty = \max_i |x_i|. \tag{2.23}$$

深度学习中，我们有时希望测量矩阵的大小，常用 Frobenius 模来表示：

$$||A||_F = \sqrt{\sum_{i,j} A_{i,j}^2} \tag{2.24}$$

## 2.9 对称矩阵

对阵矩阵（symmetric matrix），顾名思义，就是转置矩阵等于本身的矩阵：

$$\boldsymbol{A} = \boldsymbol{A}^\top \tag{2.25}$$

例如：

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{bmatrix} \tag{2.26}$$

## 2.10 正交矩阵（orthogonal matrix）

单位向量就是模为 1 的向量：

$$||\boldsymbol{x}||_2 = 1. \tag{2.27}$$

如果满足 $\boldsymbol{x}^\top \boldsymbol{y}$，则我们说向量 $\boldsymbol{x}$ 和向量 $\boldsymbol{y}$ 相互正交。

如果向量不仅正交，而且模等于 1，则称它们为标准正交的。

正交矩阵为这样的矩阵，它的行是相互标准正交的，它的列也是相互标准正交的：

$$\boldsymbol{A}^\top \boldsymbol{A} = \boldsymbol{A}\boldsymbol{A}^\top = \boldsymbol{I} \tag{2.28}$$

因此可知：

$$\boldsymbol{A}^{-1} = \boldsymbol{A}^\top \tag{2.29}$$

## 2.11 特征分解

为什么要进行矩阵分解?

将数学物体 (mathematical object) 分解成组成要素，而不仅仅是它原始的表现形式，有时候方便我们更好的理解它们或者发现它们通用的特性。比如：12 这个数字，如果用十进制表示，则为 12，如果用 2 进制表示，则为 1100，如果用 16 进制表示，则为 C，但是，12 可以分解为 $12 = 2 \times 2 \times 3$ 却总是真的，因而我们可以知道，12 可以被 2 整除，不能被 5 整除。矩阵分解就类似于数的因式分解，通过对矩阵进行矩阵分解，我们可以更好的理解矩阵的特性。

mathematical object，我不喜欢用数学对象来表述，而更喜欢用数学物体，这个物体是一种统称，而不仅仅是现实生活中的一种实物，数学符号也是一种数学物体，

使用最广泛的矩阵分解为特征值分解 (eigendecomposition)，即将矩阵分解为一组特征向量和特征值。方阵 $\boldsymbol{A}$ 的特征向量为非零向量 $\boldsymbol{v}$ 满足：

$$\boldsymbol{A}\boldsymbol{v} = \lambda\boldsymbol{v}. \tag{2.30}$$

标量 $\lambda$ 为特征向量对应的特征值。

假设矩阵 $\boldsymbol{A}$ 有 $n$ 个线性无关的的特征向量 $\{\boldsymbol{v}^{(1)}, ..., \boldsymbol{v}^{(n)}\}$，对应的特征值为 $\{\lambda_1, ..., \lambda_n\}$，将所有的特征向量组成矩阵 $\boldsymbol{V} = [\boldsymbol{v}^{(1)}, ..., \boldsymbol{v}^{(n)}]$，将所有的特征值组成向量 $\boldsymbol{\lambda} = [\lambda_1, ..., \lambda_n]$，则 $\boldsymbol{A}$ 的特征分解为：

$$\boldsymbol{A} = \boldsymbol{V}\text{diag}(\lambda)\boldsymbol{V}^{-1} \tag{2.31}$$

### 2.11.1 正定矩阵

如果一个矩阵的所有特征值都为正数，则该矩阵为“正定矩阵”。

$$\boldsymbol{A}\boldsymbol{v} = \lambda \boldsymbol{v} \tag{2.32}$$

$$\Downarrow$$

$$\begin{cases} \boldsymbol{A}\boldsymbol{v}_1 = \lambda_1 \boldsymbol{v}_1 \\ \dots \\ \boldsymbol{A}\boldsymbol{v}_n = \lambda_n \boldsymbol{v}_n \end{cases} \tag{2.33}$$

$$\Downarrow$$

$$\boldsymbol{A}[\boldsymbol{v}_1, \dots, \boldsymbol{v}_n] = [\lambda_1 \boldsymbol{v}_1, \dots, \lambda_n \boldsymbol{v}_n] \tag{2.34}$$

$$\Downarrow$$

$$\boldsymbol{A}[\boldsymbol{v}_1, \dots, \boldsymbol{v}_n] = [\boldsymbol{v}_1, \dots, \boldsymbol{v}_n] \begin{bmatrix} \lambda_1 & \dots & 0 \\ & \vdots & \\ 0 & \dots & \lambda_n \end{bmatrix} \tag{2.35}$$

令 $\boldsymbol{V} = [\boldsymbol{v}_1, \dots, \boldsymbol{v}_n]$, $\boldsymbol{\lambda} = [\lambda_1, ..., \lambda_n]$, 则:

$$\boldsymbol{A}\boldsymbol{V} = \boldsymbol{V}\mathrm{diag}(\lambda) \tag{2.36}$$

$$\boldsymbol{A} = \boldsymbol{V}\mathrm{diag}(\lambda)\boldsymbol{V}^{-1}$$

## 2.12 奇异值分解（singular value decompositon）

奇异值分解可以提供给我们特征值分解相同的信息，但其应用更广泛，因为可以使用奇异值分解的矩阵的范围更广，比如，如果矩阵中的元素都是实数，该矩阵不是方阵，则不用使用特征值分解，但却可是使用奇异值分解。任一实数矩阵都有对应的奇异值分解。

假设矩阵 $\boldsymbol{A}$ 的形状为 $m \times n$，则

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^{\top} \tag{2.37}$$

$\boldsymbol{U}$ 为 $m \times m$ 的正交矩阵，$\boldsymbol{D}$ 为 $m \times n$ 的对角矩阵，$\boldsymbol{V}$ 为 $n \times n$ 的正交矩阵。

# 第三章　概率

概率论是表示不确定事件的数学，它提供了量化不确定的方法，以及推导新的不确定事件的理论。

人工智能中为什么要使用概率论?

- 用来解释人工智能的原理。
- 用来分析人工智能的表现。

> 其实，所有的科学都应包含概率论，确定的事情是很少的。概率可以看作处理不确定性的逻辑。逻辑根据已有的假设的真假，来判断推理的真或假；概率论根据已有假设的概率，来决定推理的概率。

## 3.1　随机变量

随机变量是可以随机取不同的值的变量，它是对可能的状态的一种描述，它必须和表示该状态可能性的概率分布一起使用。

## 3.2 概率分布

# 第二部分

# 机器学习

# 第三部分

# 深度学习

# 第四部分

# 卷积网络

# 第五部分

# 循环网络

# List of Tables

# List of Figures