Terminology

数学部分

标量，向量，矩阵和张量的关系

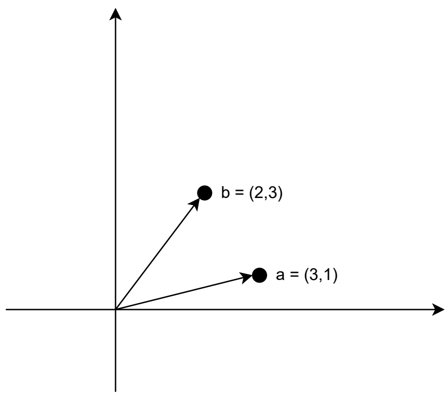
- 标量(scalar): 就是**0维**张量，代码里里用**变量**表示
- 向量(vertor): 就是**1维**张量，代码里里用**一维数组**表示
- 矩阵(matrix): 就是**2维**张量，代码里里用**二维数组**表示

几何向量

向量拥有大小和方向。

几何向量的表示法

- 用几何方法来表示向量，如下图这样用箭头来表示的二位向量。



- 用纵向排列方式表示向量，这样的向量被称为列向量。

  $\vec{a} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}, \vec{b} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$

向量的四则运算

$\vec{a} + \vec{b} = \begin{bmatrix} 3 \\ 1 \end{bmatrix} + \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 3+2 \\ 1+3 \end{bmatrix} = \begin{bmatrix} 5 \\ 4 \end{bmatrix}$

$\vec{a} - \vec{b} = \begin{bmatrix} 3 \\ 1 \end{bmatrix} + \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 3-2 \\ 1-3 \end{bmatrix} = \begin{bmatrix} 1 \\ -2 \end{bmatrix}$

$\vec{a} \cdot \vec{b} = a_1 b_1 + a_2 b_2 = 3 \times 2 + 1 \times 3 = 9$
Tips: 向量**点积**（dot product, 也称**内积**）后得到的已经不是向量了，而是一个**标量**，所以点积也称**标量积**。

余弦定理的向量表示法

我们先介绍几个概念，然后在推导出余弦定理。

- $||a||$表示向量的**长度**（也可以理解为**距离**）。
  - 假如二维向量$\vec{a} = (a_1, a_2)$，那么 $||a|| = \sqrt{a_1^2 + a_2^2}$
- 假设向量$\vec{a}$和$\vec{b}$之间的夹角为θ, 如何计算$\vec{a} \cdot \vec{b}$呢?
  - 先做个方向的转换，我们把$\vec{b}$投影到$\vec{a}$上，这样$\vec{b}$在$\vec{a}$方向上的投影就变成了$\left|\left|b\right|\right| \cos\theta$
  - $\vec{a}$在自己方向上的投影就是$||a||$
  - 这样，$\vec{a} \cdot \vec{b}$ 就等价于 $||a||||b|| \cos\theta$，即$\vec{a} \cdot \vec{b} = ||a||||b|| \cos\theta$

简单做一个等式变化，我们就得到了二维向量的余弦定理。

$$\cos\theta = \frac{\vec{a}\cdot\vec{b}}{\|a\|\|b\|} = \frac{a_1b_1 + a_2b_2}{\sqrt{a_1^2 + a_2^2}\sqrt{b_1^2 + b_2^2}}$$

推广到N维向量空间，就得到公式

$$\vec{a} = (a_1, a_2, \ldots, a_n); \qquad \vec{b} = (b_1, b_2, \ldots, b_n)$$

$$\cos\theta = \frac{\vec{a}\cdot\vec{b}}{\|a\|\|b\|} = \frac{a_1b_1 + a_2b_2 + \ldots + a_nb_n}{\sqrt{a_1^2 + a_2^2 + \ldots + a_n^2}\sqrt{b_1^2 + b_2^2 + \ldots + b_n^2}}$$

通常，我们用余弦定理来进行相似度的计算。

- 如果两个向量夹角很小，cos值大于0，接近1，说明他们很相似，即**正相关**。
- 如果两个向量夹角是90度，cos值为0，说明他们不相似，是正交的。
- 如果两个向量夹角大于90度，cos值为小于0，说明他们不相似，即**负相关**。

相似度(Similarity)或距离(distance)

详见《统计学习方法》P255

常用的距离量度

1. 闵可夫斯基距离
    1. 欧氏距离
    2. 曼哈顿距离
    3. 切比雪夫距离
2. 马哈拉诺比斯距离(Mahalanobis distance)
   常用的相似度量度
3. 相关系数(Correlation coefficient)
4. 夹角余弦(Cosine)

机器学习部分

均方误差(MSE:Mean Square Error)

公式： $\frac{1}{n}\sum_{i=1}^{n}(y^{(i)} - f_\theta(x^{(i)}))^2$

Accuracy，Precision和Recall

在分类问题中，我们经常需要计算Accuracy的值来评估模型训练的结果。

分类	结果标签-True	结果标签-False
Positive	True Positive(TP)	False Positive(FP)
Negative	False Negative(FN)	True Negative(TN)

- 正确率(Accuracy)

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

- 精确率(Precision)

$$Precision = \frac{TP}{TP + FP}$$

- 召回率(Recall)

$$Recall = \frac{TP}{TP+FN}$$

归一化（Normalization）

归一化是一种常用的数据预处理技术，主要用于消除数据特征之间的量纲和数值范围差异，使得不同特征具有相同的尺度。

归一化的基本思想是将原始数据按比例缩放，使之落入一个小的特定区间。这样做可以使得模型训练更加稳定，收敛更快，同时可以防止模型在训练过程中产生过大或过小的数值。

常见的归一化方法有以下几种：

1. **最大最小归一化**：这种方法将原始数据线性变换到[0,1]区间，计算公式为：

$$x' = \frac{x - min}{max - min}$$

其中x是原始数据，x'是归一化后的数据，min和max分别是数据集中的最小值和最大值。
2. **Z-Score归一化**：这种方法将原始数据变换为均值为0，标准差为1的数据，计算公式为：

$$x' = \frac{x - \mu}{\sigma}$$

其中x是原始数据，x'是归一化后的数据，μ是数据集的均值，σ是数据集的标准差。
3. **单位长度归一化**：这种方法将原始数据变换为单位长度，即每个数据点都在单位球面上。
4. **批归一化**（Batch Normalization）是一种在深度学习中常用的技术，主要用于解决深度神经网络训练过程中的梯度消失和梯度爆炸问题。Batch Normalization的基本思想是对每个小批量（mini-batch）的数据进行归一化处理，使得结果（输出信号各个维度）的均值为0，方差为1。Batch Normalization的计算过程如下：

- 计算均值和方差。
  - 均值公式：

$$\mu = \frac{1}{D}\sum_{i=1}^{D} x_i$$

  - 方差公式：

$$\sigma = \sqrt{\frac{1}{D}\sum_{i=1}^{D}(x_i - \mu)^2}$$

- 进行归一化：通过均值和方差，可以得到归一化后的值, 公式：

$$\hat{x} = \frac{x - \mu}{\sqrt{\sigma^2 + \varepsilon}}$$

其中，ε是一个很小很小的数，用于防止分母为0这种情况。

- 线性变换：在Layer Normalization中，我们还需要一组参数来保证归一化操作不会破坏之前的信息。这组参数叫做增益（gain）γ和偏置（bias）β。
  输出公式：

$$y = \gamma\hat{x} + \beta$$

其中γ和β是可学习的参数，可以通过反向传播进行优化的。
5. **层归一化**（Layer Normalization）与Batch Normalization不同，Layer Normalization是在特征维度上进行标准化的，而不是在数据批次维度上。具体的计算过程如下：

- 计算均值和方差。
  - 均值公式：

$$\mu = \frac{1}{D}\sum_{i=1}^{D} x_i$$

  - 方差公式：

$$\sigma = \sqrt{\frac{1}{D}\sum_{i=1}^{D}(x_i - \mu)^2}$$

- 进行归一化：通过均值和方差，可以得到归一化后的值, 公式：

$$\hat{x} = \frac{x-\mu}{\sqrt{\sigma^2+\varepsilon}}$$

其中，ε是一个很小很小的数，用于防止分母为0这种情况。

- 线性变换：在Layer Normalization中，我们还需要一组参数来保证归一化操作不会破坏之前的信息。这组参数叫做增益（gain）g和偏置（bias）b（等同于Batch Normalization中的γ和β）。
  输出公式：

$$h = f(\frac{g}{\sqrt{\sigma^2+\varepsilon}} \odot (x-\mu) + b)$$

其中f是激活函数，$\odot$ 表示Hadamard Product，也就是操作矩阵中对应的元素相乘，因此要求两个相乘矩阵是同型的。