

前置知识

矩阵

一个 n 行 m 列的矩阵就是由 $n \times m$ 个实数排列而成的矩形阵列，例如下面的矩阵 M 就是一个3行4列的矩阵

$$M = \begin{bmatrix} 1.1 & 1.4 & 2.1 & 4.2 \\ 0.9 & 1.9 & 2.2 & 0.8 \\ 0.7 & 1.8 & 1.2 & 1.3 \end{bmatrix}$$

我们用 A_{ij} 表示矩阵 A 的第 i 行第 j 列个元素，例如对于上述矩阵 M ， $M_{23} = 2.2$ 。

矩阵的加法

两个大小相同的矩阵才能够相加，我们定义如果 R, P, Q 是三个大小相同的矩阵，且 $R = P + Q$ ，那么

$$R_{ij} = P_{ij} + Q_{ij}$$

例如给定两个2行3列的矩阵 P, Q

$$P = \begin{bmatrix} 1.8 & 3.4 & 1.1 \\ 2.1 & 1.8 & 3.1 \end{bmatrix}, Q = \begin{bmatrix} 0.1 & 0.7 & 2.5 \\ 3.1 & 2.9 & 1.5 \end{bmatrix}$$

则

$$R = P + Q = \begin{bmatrix} 1.9 & 4.1 & 3.6 \\ 5.2 & 4.7 & 4.6 \end{bmatrix}$$

矩阵的乘法

矩阵的乘法稍微复杂一些，我们定义，对于两个矩阵 P, Q ，如果想要计算 $P \times Q$ ，则 P 的列数需要和 Q 的行数相等。

对于一个大小为 n 行 w 列的矩阵 P ，和一个大小为 w 行 m 列的矩阵 Q ，我们定义其结果 $R = P \times Q$ 是 n 行 m 列的，并且满足：

$$R_{ij} = \sum_{k=1}^w P_{ik} \times Q_{kj}$$

例如给定一个2行3列的矩阵 P 和一个3行4列的矩阵 Q

$$P = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix}, Q = \begin{bmatrix} 1 & 0 & 2 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 2 & 1 \end{bmatrix}$$

则

$$R = P \times Q = \begin{bmatrix} 1 & 3 & 6 & 2 \\ 0 & 3 & 5 & 2 \end{bmatrix}$$

比如其中 $R_{12} = P_{11} \times Q_{12} + P_{12} \times Q_{22} + P_{13} \times Q_{32} = 1 \times 0 + 2 \times 1 + 1 \times 1 = 3$ ，可以理解为 R_{ij} 就是 P 的第 i 行与 Q 的第 j 列的元素对应相乘的和。

需要注意的是，矩阵的乘法是不满足交换律的，也就是在大多数情况下， $A \times B \neq B \times A$ ，你需要注意乘法的顺序。

向量

我们称1行n列的矩阵为n维的行向量，n行1列的矩阵为n维的列向量，例如我们给了一个3维的行向量 \vec{a} 和一个2维度的列向量 \vec{b}

$$\vec{a} = [0.5 \quad 0.3 \quad 1.1], \vec{b} = \begin{bmatrix} 0.5 \\ 0.3 \end{bmatrix}$$

我们直接使用 \vec{v}_i 表示向量 \vec{v} 的第 i 个元素，例如对于上述向量 \vec{a} ， $\vec{a}_2 = 0.3$ 。

向量是一种矩阵，同样可以进行上面说明的矩阵加法和矩阵乘法。

RELU函数

RELU函数是一个矩阵函数，具体来说，如果输入是一个n行m列的矩阵 P ，则输出 $R = \text{RELU}(P)$ 同样是一个n行m列的矩阵，满足

$$R_{ij} = \max(0, P_{ij})$$

例如给定一个矩阵 P

$$P = \begin{bmatrix} -1.8 & 3.4 & -1.1 \\ 2.1 & -1.8 & -3.1 \end{bmatrix}$$

则

$$R = \text{RELU}(P) = \begin{bmatrix} 0 & 3.4 & 0 \\ 2.1 & 0 & 0 \end{bmatrix}$$

softmax函数

softmax函数是一个向量函数，具体来说，如果输入是一个n维的行/列向量 \vec{z} ，则输出 $\vec{r} = \text{softmax}(\vec{z})$ 同样也是一个n维的行/列向量，满足

$$\vec{r}_i = \frac{e^{\vec{z}_i}}{\sum_{j=1}^n e^{\vec{z}_j}}$$

例如给定一个向量 \vec{z}

$$\vec{z} = \begin{bmatrix} 0.5 \\ 0.3 \\ 1.1 \end{bmatrix}$$

则

$$\vec{r} = \text{softmax}(\vec{z}) = \begin{bmatrix} \frac{e^{0.5}}{e^{0.5} + e^{0.3} + e^{1.1}} \\ \frac{e^{0.3}}{e^{0.5} + e^{0.3} + e^{1.1}} \\ \frac{e^{1.1}}{e^{0.5} + e^{0.3} + e^{1.1}} \end{bmatrix}$$

简单的神经网络推理（题目中的）

题目中出现的神经网络仅仅包含2个全连接层和一个softmax输出头，假设我们有一个输入 x 和四个矩阵 W_1, b_1, W_2, b_2 ，则我们要求的结果就是

$$\vec{r} = \text{softmax}(\text{RELU}(x \times W_1 + b_1) \times W_2 + b_2)$$

神经网络的推理就是这样的一个计算过程，可以确定最后的结果是一个向量，用 \vec{r} 表示，代表了 x 的分类结果。其中 \vec{r}_i 表示了这个输入 x 属于第 i 个类别的概率。