2025 模式识别
作业一

人工智能学院 221300079 王俊童

2025.3.4

221300079 王俊童，人工智能学院

1 问题一

a. 由于只考虑实数且三次方根并不造成影响，我们只考虑其中的二次方根即可，解答如下：

$$\frac{8a-1}{3} \geq 0, \quad a \geq \frac{1}{8}$$

b. 当 $a = \frac{1}{8}$ 时，带入 1.1 可得到:

$$f(a) = f(\frac{1}{8}) = \sqrt[3]{\frac{1}{8}} + \sqrt[3]{\frac{1}{8}} = 1$$

c. 显然对于这个复杂方程可以找到其他也等于 1 的 x，可以解为 $\frac{1}{2}$ 或者 $\frac{13}{8}$（严格等于 1）

d. 根据书上的写法，matlab 给出答案如下:$1.2182 + 0.1260i$

e. 原因出在我们使用的是 $^(1/3)$ 这个写法，这导致其在复数域计算，应当使用 nthroot 或者 python 的 np.cbrt，我对以上两种都进行了验证，无论是 matlab 还是 python，在 $a > \frac{1}{8}$ 的时候，等于 1 都成立。

f. 为了证明大于 0.125 的时候都成立，我们令 $a = \frac{1}{8} + x^2, x \geq 0$, 可以解的：

$$f(a) = \sqrt[3]{\frac{1}{8} + x^2 + (\frac{1}{3}x^2 + \frac{3}{8}) * \sqrt{\frac{8}{3}x}} + \sqrt[3]{\frac{1}{8} + x^2 - (\frac{1}{3}x^2 + \frac{3}{8}) * \sqrt{\frac{8}{3}x}}$$

$$f(a) = \sqrt[3]{(\frac{1}{2} + \sqrt{\frac{2}{3}}x)^3} + \sqrt[3]{(\frac{1}{2} - \sqrt{\frac{2}{3}}x)^3}$$

$$f(a) = 1$$

g. 观察这个式子的形式，可以发现令 $a = 2$ 即可。

$$f(2) = \sqrt[3]{2 + \frac{2+1}{3} * \sqrt{\frac{16-1}{5}}} + \sqrt[3]{2 - \frac{2+1}{3} * \sqrt{\frac{16-1}{5}}}$$

$$f(2) = \sqrt[3]{2+\sqrt{5}} + \sqrt[3]{2-\sqrt{5}} = 1$$

h. 根据 cardano 公式所说, 对于 $x^3+px+q=0$, 实系数一元三次方程有:

$$x=\sqrt[3]{\frac{-q}{2}+\sqrt{\frac{p^3}{27}+\frac{q^2}{4}}}+\sqrt[3]{\frac{-q}{2}-\sqrt{\frac{p^3}{27}+\frac{q^2}{4}}}$$

这个跟 1.1 长得基本一样。我们猜测 1.1 其实是某一个 3 次方程的根，而我们经过上面推导得到 $a>0.125$ 的时候，恒为 1，所以 1 有可能就是一个根。那么我们使用待定系数法可以得到

$$a=\frac{-q}{2},(z-1)(z^2+bz+c)=z^3+(b-1)z^2+(c-b)z-c$$

由于要符合 cardano 三次方程的形式，可以得到 $b-1=0,-c=q$, 所以:

$$(z-1)(z^2+bz+c)=(z-1)(z^2+z-q)=z^3+(-q-1)z+q,q=-2a$$

所以:

$$z^3+(2a-1)z-2a=0$$

则是这个方程的一个根，并且在 a 比 0.125 大的时候恒为 1 成立。

2 问题二

a. 对于标准正态分布 $\mu=0,\sigma=1$, 所以可以得到:

$$P(X\geq\epsilon)=\int_\epsilon^\infty\frac{1}{\sqrt{2\pi}}e^{\frac{-t^2}{2}}\,dt=\int_0^\infty\frac{1}{\sqrt{2\pi}}e^{\frac{-(\epsilon+x)^2}{2}}\,dx\leq e^{\frac{-\epsilon^2}{2}}\int_0^\infty\frac{1}{\sqrt{2\pi}}e^{\frac{-x^2}{2}}\,dx=\frac{1}{2}e^{\frac{-\epsilon^2}{2}}$$

b. 对于标准正态分布，$f(x)'=-xf(x),f(x)=\frac{1}{\sqrt{2\pi}}e^{\frac{-x^2}{2}}$, 所以可得:

$$P(|X|\geq\epsilon)=2P(|x|\geq\epsilon)=2\int_\epsilon^\infty\frac{xf(x)}{x}\,dx\leq-2\int_\epsilon^\infty\frac{f(x)'}{\epsilon}\,dx=\frac{-2}{\epsilon}f(x)|_\epsilon^\infty=\sqrt{\frac{2}{\pi}}\frac{e^{\frac{-\epsilon^2}{2}}}{\epsilon}$$

所以对于正态分布有:

$$P(|X|\geq\epsilon)\leq min\{1,\sqrt{\frac{2}{\pi}}\frac{e^{\frac{-\epsilon^2}{2}}}{\epsilon}\}$$

3 问题三

a. 全连接层主要提供了一个简单的方法去从特征空间中学习非线性的映射，将学习的权重进行整合，提取好的特征组合。比如一个输入向量 $\mathbf{x}$, 经过一个权重矩阵 W 加上一个 bias 项就可以得到一个输出向量 $\mathbf{y}$。

$$\mathbf{y}=\mathbf{W}\mathbf{x}+b$$

BN 层呢主要是对每一个 batch 进行输入归一化，将数据变为均值 0，标准差 1 的高斯分布或者在 0 附近的分布。如果不处理的话样本可能过于分散导致学习速度慢或者不能学习的问题，也有可能后续导致梯队消失和梯度爆炸。同时 BN 层还会应用一个可训练的缩放参数尺度因子 γ 和平滑因子 β 来解决被限制在正态分布下网络表达能力下降的问题。具体公式如下:

$$\mathbf{y}=\frac{\mathbf{x}-\mu}{\sqrt{\sigma^2+\epsilon}}\gamma+\beta$$

b. 假设第一个 FC 层为 $\mathbf{W}_1$,bias 为 $\mathbf{b}_1$. 则:

$$\mathbf{y}_1=\mathbf{W}_1\mathbf{x}+\mathbf{b}_1$$

第二个也一样可以表示为:

$$\mathbf{y}_2 = \mathbf{W}_2\mathbf{y}_1 + \mathbf{b}_2$$

带入可以得到:

$$\mathbf{y}_2 = \mathbf{W}_2\mathbf{W}_1\mathbf{x} + \mathbf{W}_2\mathbf{b}_1 + \mathbf{b}_2$$

这表明可以和为一个新的 FC 层，新的权重是 $\mathbf{W}_2\mathbf{W}_1$,bias 为 $\mathbf{W}_2\mathbf{b}_1 + \mathbf{b}_2$

c.
- 如果两个全连接层要向前传的话，这个是可以减少层数达到你要的效果的，降低了计算量。在前向传递的时候，减少了矩阵乘法的次数。
- 同时还加速了模型速度，减少了数据的存储，这样是合适的。

d.
- 丢失部分第一层次的信息。一般全连接层后面会有 relu 这种激活函数，合并为一个的话可能丢失一部分非线性变换，从而导致表达能力下滑。
- 合并之后层变得比较复杂，可解释性会降低因为本身是一个线性模型，而且如果你要对于单独层进行分析，合并为一个显然是不好的，因为这样解释不了。

e. 当 FC 层后面跟了一个 BN 层之后，可以合并为一个 FC 层，证明如下:

$$\mathbf{y}_1 = \mathbf{W}\mathbf{x} + \mathbf{b}, \quad \hat{\mathbf{y}}_1 = \frac{\mathbf{y}_1 - \mu}{\sqrt{\sigma^2 + \epsilon}}, \quad \mathbf{y}_2 = \gamma\hat{\mathbf{y}}_1 + \beta$$

所以新的权重和 bias 可以表示为:

$$\mathbf{W}' = \frac{\gamma\mathbf{W}}{\sqrt{\sigma^2 + \epsilon}}, \mathbf{b}' = \frac{\gamma(\mathbf{b} - \mu)}{\sqrt{\sigma^2 + \epsilon}} + \beta$$

这样就可以合并为一个新的 FC 层了，这样有很多好处的，比如每一次提取之后归一化，这样数据会很规整，而且可以极尽其表达能力，这样保证了数据既规范又能够经过缩放和平滑展现其原有特征，对于之后要做的步骤，无疑是很好的省略开销和简化。

f. 我们假设模型的卷积层输出为 $y = Conv(x)$, 然后 BN 层的式子跟上面一样，所以很快就可以得到一个关系如下:

$$y = \gamma\frac{Conv(x) - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta$$

很自然的，其中如果转换成 FC 层的话，就可以:$W' = \frac{W}{\sqrt{\sigma^2+\epsilon}}, b' = \beta - \frac{\gamma\mu}{\sqrt{\sigma^2+\epsilon}}$ 然后我们用 pytorch 实现一下.

实验结果表明, 有提升，加速了 1.14 倍

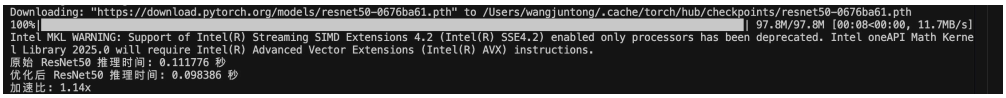

Figure 1: 实验对比

4 问题四

a.
- 缩放法，比如使用数值分析中的最近邻差值，把 $(4i+1, 4j+1)$ 的像素点作为最近邻差值。存为 (i, j) 的像素值。
- 双线性插值, 将 $(4i+1, 4j+1)(4i+2, 4j+1), (4i+1, 4j+2), (4i+2, 4j+2)/4$ 这四个作为 (i, j)

b. 如果两两接近的一个 2*2 的小矩阵具有相似性的话，将他们差值按照上面的第一问说法变成一个 1*1 的像素点储存不就行了。

c. 训练集为 $acc_{train} = \frac{9900}{9900+100} * 100\% = 99\%$ 测试集为:$acc_{test} = \frac{5000}{5000+5000} * 100\% = 50\%$

d. micro 方法指的是总正确分类样本除以总样本数量,$micro - acc = \frac{\sum TP_i}{Sum\ of\ instances}$ 而 macro 方法是计算出每一类比的 acc 之后求平均:$macro - acc = \frac{1}{N}\sum acc_i$. 根据分析，c 中我们采取的是 micro 方法，是有偏见的。

e. 显然经过上面的分析，我们发现样本其实真正的表现好是在 test 上面表现好，那么其实我们应该选 macro，因为这样不均衡样本的一些问题和表现不好可以被考虑进来，针对不均衡进行修改。一些可以采用的训练方法是，针对训练样本进行重采样，这样可以缓解不均衡问题，比如说给 B 类多生成 9800 个样本。或者就是人为的设计一个根据样本有关的均衡权重，比如可以为多类别/少类别这样的权重，然后赋权给少样本来增加其影响力。

5 问题五

首先这些样本分布如下:

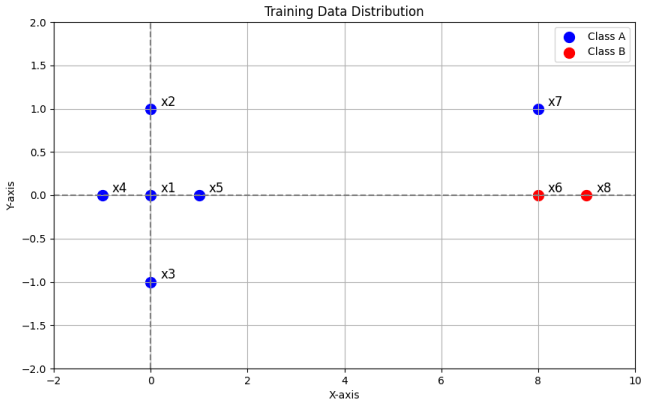

Figure 2: distribution

a.
- z_1 的 1- NN 为 A 类
- z_2 的 1- NN 为 A 类

b.
- z_1 的 1- NN 为 A 类
- z_2 的 1- NN 为 B 类

c. 因为在第一次 1—NN 分类里面，就是只看最近的点，明显是 x_3, x_7 两个起了主导，但是 3-NN 的话，就需要考虑 x_6, x_8 所以导致了第二个 z 被分为 B 类别

d. 可能啊，因为 x_7 从图上看起来还是蛮离群的，所以有可能是标签打错了之类的问题。k-NN 比 1-NN 更有容错啊，加入你的决策边界并不是一个线性的，显然 k-NN 给了更大的容错空间，看的是主导 label 的个数从而来决定你的类别。

6 问题六

a. 已提供

b. 已提供

c. 已完成

d. 已完成

e. 已完成

f. 已完成

g. 已完成

h. 已完成

i. 收获在 jupyter notebook（pdf）里面。