

## 交叉科技 答题：

### 数学与统计：

1、24

2、

当 $N > 365$ ，概率为1，

当 $N \leq 365$ 时，概率为： $1 - \frac{365 \times 364 \times \dots \times (365 - N + 1)}{(365^N)}$

3、蒙特卡罗方法又称为统计模拟方法，是通过从概率模型的随机抽样进行近似数值计算的方法。

通过蒙特卡罗计算计算 $\pi$ 的实验方法：

1、在单位正方形区域进行随机均匀采样 $n$ 个点

2、统计落在该正方形内切圆的采样点个数 $m$ 个

3、则 $\pi$ 可以通过 $\pi = \frac{4 \times m}{n}$ 近似计算，采样点数 $n$ 越大，计算越准确

### 逻辑思维：

1\*\*、\*\*

### 编程题：

1.

```
def reverseBits(self, n):  
    temp = bin(n)[2:]  
    return** int(temp[::-1],2)
```

2、

```

def minimumTotal(self, triangle):
    """
        :type triangle: List[List[int]]
        :rtype: int
    """
    n=len(triangle)
    for row in range(n-2,-1,-1):
        for col in range(len(triangle[row])):
            triangle[row][col]+=min(triangle[row+1][col],triangle[row+1][col+1])
    return triangle[0][0]

```

3.

```

def trap(self, height):
    """
        :type height: List[int]
        :rtype: int
    """
    if len(height) <= 2:
        return 0
    nonZeroIdx = 0
    for i, hi in enumerate(height):
        if hi != 0:
            nonZeroIdx = i
            break

    height = height[nonZeroIdx:]
    startEnd = height[0]
    stopEnd = 0
    stopIdx = 1
    for i in range(1, len(height), ):
        hi = height[i]
        if hi >= startEnd:
            stopEnd = hi
            stopIdx = i
            break
        if hi >= stopEnd:
            stopEnd = hi
            stopIdx = i

    belowV = min(startEnd, stopEnd)
    curVol = 0
    for i in range(1, stopIdx):
        curVol += belowV - height[i]

    if stopIdx == len(height) - 1:
        return curVol
    else:

```

```
return curVol + self.trap(height[stopIdx:])
```

## 机器学习:

1. Explain how to view (soft-margin) SVM as a penalization method (i.e., explain the objective SVM as a Loss+Regularization term)

svm的一种解释是最小化如下的目标函数:

$$\sum_{i=1}^N [1 - y_i(wx_i + b)]_+ + \lambda \|w\|^2$$

上式第一项是经验损失, 又称为合页损失, 含义如下:

$$\mathcal{L}_\epsilon(Z) = \begin{cases} 0, & |Z| \leq \epsilon \\ |Z| - \epsilon, & |Z| > \epsilon \end{cases}$$

第二项就是对模型添加惩罚的二次正则化项。因此说SVM是一种penalization method。

2. Explain the principle of the stochastic gradient descent algorithm. What is the benefit of using stochastic gradient descent (instead of gradient descent)?

梯度下降以负梯度方向为搜索方向,

由于梯度下降每次更新时需要汇总所有训练样本的误差, 因此计算量比较大, 而随机梯度下降每次只需要汇总少量样本的误差来更新权重, 因此计算量小, 效率更高。另一方面如果标准误差曲面有多个局部极小值, 梯度下降容易陷入局部极小, 而随机梯度下降则可能避免陷入这些局部极小值中。

3. What is overfitting? How to prevent overfitting (in both traditional method and deep learning)?

过拟合是由于模型过于复杂或者是训练样本太少, 导致在模型在训练数据集上性能比较好, 而在测试集上性能较差, 模型方差较大, 模型泛化性能较差。

在传统学习方法中解决过拟合一般通过数据增强, 增大训练集, 给模型添加正则化项, 甚至降低模型复杂度(多项式拟合中降低多项式次数)等方法。

在深度学习中一般也可以增大训练数据集, 进行数据增强, 同时可以采用dropout, 还可以减少网络深度和宽度等方法, 同时也可以可以在损失函数中, 添加对参数的正则化项等。

4. Explain what are the gradient vanishing/explosion problems in RNN? (we need the underlying mathematical reason)

**梯度消失:** RNN梯度消失是因为激活函数tanh函数的导数在0到1之间, 反向传播时更新前面时刻的参数时, 当参数W初始化为小于1的数, 则多个(tanh函数导数 \* W)相乘, 将导致求得的偏导极小(小于1的数连乘), 从而导致梯度趋于0, 产生梯度消失现象。

**梯度爆炸：**而当参数初始化为足够大，使得tanh函数的导数乘以W大于1，则将导致偏导极大（大于1的数连乘），从而导致梯度爆炸。

5. Consider the image captioning problem. Given a picture as the input, your algorithm should output a sentence describing the picture.

(a) Describe a neural network structure that can do this. (if you use well known CNN or RNN structures, you don't need to go into the details. Just

say how to use them (what the input, what is the output).)

(b) Describing the training process (what is the training data? what is the objective?)

(c) Describe the test process (In the testing phase, give a picture as the input, how your algorithm output a sentence describing the picture.)

(d) Suppose we want to have a parameter to determine the diversity of the output sentence (on one extreme, the output is almost a deterministic sentence, and on the other extreme, the output is almost a random sequence) This can be done by adding a temperature parameter to the softmax function. How would you do it? (there may be more than one reasonable answers.)

a. 图像描述问题一般采用encoder-decoder网络结构。encoder和decoder一般都采用rnn模型，encode阶段，将图像按窗口大小，划分为按位置排列的图像块序列，输出则是每个图像块的一个语义编码向量。

在decoder阶段，初始输入为句子开始标志位，然后在预测 $t$ 时刻的单词时，考虑上一时刻 $t - 1$ 的状态然后和encoder阶段输出的图像块语义编码向量做attention操作，得到这一时刻输出的状态。

b. 训练阶段的loss函数是各个时刻预测的单词输出概率和真实句子单词的交叉熵损失的和。

c. 在测试阶段，将输入图像输入编码器，得到各个图像块的语义向量，然后输入进decoder，初始时刻，decoder输入句子开始标记，开始预测此时句子输出。预测 $t$ 时刻的输出，都是将 $t - 1$ 时刻的输出单词作为输入。也即是逐个单词进行预测输出。为了保证最终效果，一般也会采取beam search方式。

d.