

## 读程题

---

1.

```
1. int a,b,d=10;
   a = d | 1 % 9, b = d & (-1);
   printf(“%d,%d”,a,b);
```

结果: 1, 0

```
a = 10 | 1 = 1010 | 0001 = 1011 = 11
b = 10 & -1 = 1010 & 1111 = 1010 = 10
11,10
```

2.

```
2. int main(){
    int i;
    void f();
    for(i=0;i<5;i++)
        f();
    return 0;
}

void f(){
    static int cnt = 0;
    ++cnt;
    Printf(“%d”,cnt);
}
```

结果: 5

3.

```

3. int a[3][3];
   int *p = &a[0][0];
   for(int i=0;i<9;i++)
       p[i] = i+1;
   for(int i=0;i<3;i++)
       printf("%d",a[2][i]);

```

结果：789

4.

4.

```

#include<stdio.h>
#define M 5
#define N M+M
int main(void)
{
    int k;
    k=N*N*5;
    printf("%d\n",k);
    return 0;
}

```

结果：500

$$\text{int } k = N \times N \times 5 = M + M * M + M * 5 = 5 + 5 * 5 + 5 * 5 = 55$$

5.

5.

```
#include <stdio.h>
int main(void)
{
    int a = 6, b = 8;
    while (a--);
    while (--b);
    printf("a=%d b=%d\n", a, b);
    return 0;
}
```

结果: a=0 b=0

a = -1 b = 0

a 先判断再减1, 到0的时候不再进入循环, 之后退出来再减1

b 先减1再判断, 到1的时候, 先减1, 变为0之后判断再退出来

6.

6.

```
#include <stdio.h>
int main(void)
{
    int a[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, *p = a;
    printf("x:%d y:%d\n", *a, *(p + 5));
    return 0;
}
```

结果: x:1 y:6

## 编程题

---

1.

1. 题目：有5个学生坐在一起，问第5个学生多少岁，他说比第4个学生大2岁。问第4个学生岁数，他说比第3个学生大2岁。问第3个学生，又说比第2个学生大2岁。问第2个学生，说比第一个学生大2岁。最后问第一个学生， he 说是10岁。请问第5个学生多大。

答：

```
void five_student(int n){
    if(n == 1) return 10;
    return five_student(n-1) + 2;
}
```

2.

2. 编写一个函数，对一个无符号短整型数，取它的偶数位（即从左边起第2、4、6... 位）与奇数位（即从左边起第1、3、5...）分别组成新的无符短整型数并通过形参传回调用参数。

原型：

```
void Split(unsigned short a, unsigned short * pOdd, unsigned short * pEven);
```

其中pOdd代表奇数位，pEven代表偶数位。

答：

```
void Split(unsigned short a, unsigned short *pOdd, unsigned short *pEven){
}
}
```

灵活使用atoi, itoa函数

更正：

```

void Split(unsigned short a, unsigned short *pOdd, unsigned short *pEven) {
    char string[200], odd[100], even[100];
    int odd_count = 0, even_count = 0;

    itoa(a, string, 10);
    // printf("%s\n", string);

    for(int i=0; string[i] != '\0'; i++){
        if(i%2 == 0){
            odd[odd_count] = string[i];
            odd_count++;
        }else{
            even[even_count] = string[i];
            even_count++;
        }
    }

    odd[odd_count+1] = '\0';
    even[even_count+1] = '\0';

    *pOdd = atoi(odd);
    *pEven = atoi(even);
}

```

3.

3. 甲乙两队进行比赛，甲队有a、b、c三人，乙队有x、y、z三人，有人想知道比赛对手，a说不和x比赛，c说不和x、z比赛，编程找三对赛手名单。

解：

答：

```

void result(char jia[],int n){
    char yi[3] = {'x','y','z'},

    for(int i=0;i<n;i++){
        for(int j=0;j<n;j++){
            if(jia[i] = 'a' && yi[j] != 'x') printf("a = %c\n",yi[j]);
            else if(jia[i] = 'c' && yi[j] != 'x' && yi[j] != 'z')
                printf("c = %c\n",yi[j]);
            else
                printf("b = %c\n",yi[j]);
        }
    }
}

```

更正:

```

void result(){
    char x,y,z;

    for(x='a';x<='c';x++)
        for(y='a';y<='c';y++)
            for(z='a';z<='c';z++)
                if(x != y && x != z && y != z && x != 'a' && x != 'c' && z != 'c'){
                    printf("x -> %c\n",x);
                    printf("y -> %c\n",y);
                    printf("z -> %c\n",z);
                }
}

```

4.

二、1. 字符串循环移动 （输入一个字符串和向左移动的位数n输出循环左移的结果）

答:

```

void result(){
    char str[100];
    int n;
    scanf("%s",str);
    scanf("%d",&n);

    for(int i=n;i<100;i++)
        printf("%c",str[i]);
    for(int i=0;i<n;i++)
        printf("%c",str[i]);

}

```

5.

2、输出以下的杨辉三角形（要求输出10行）（7分）

```

      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
 1 5 10 10 5 1
⋮ ⋮ ⋮ ⋮ ⋮ ⋮

```

答:

```

void result(){
    int data1=1,data2=2,data3;
    int n=10;

    printf("%d\n",data1);
    printf("%d %d\n",data1,data2);

    for(int i=3;i<=n;i++){
        data2 += data1;
        if(i%2 != 0) printf("")
    }
}

```

杨辉三角主要是用数组来实现的

更正：

```

void result_1(){
    int a[10][10];

    a[0][0] = 1;
    a[1][0] = a[1][1] = 1;

    for(int i=2;i<10;i++){
        for(int j=0;j<i;j++){
            if(j == 0 || j == i-1) a[i][j] = 1;
            else a[i][j] = a[i-1][j-1] + a[i-1][j];
        }
    }

    for(int i=0;i<10;i++) {
        for (int j = 0; j < i; j++)
            printf("%d ", a[i][j]);
        printf("\n");
    }
}

```

6.



### 3. 小顶堆:

有  $N(N \gg 10000)$  个整数, 求出其中的前  $K$  个最大的数。

为了方便测试我就输入 10 个数字求前 4 个最大的。

输入: (第一行  $n, k$  代表  $n$  个数字求  $k$  个最大的)

10 4  
1 2 3 4 5 6 7 8 9 10

输出:

7 8 9 10

答:

```
void result(int arr[],int n,int length){
    for(int i=0;i<n;i++){ // 排前n个数, 使前n个最大数在数组最前面
        int temp = arr[i];
        for(int j=i+1;j<length;j++){
            if(temp < arr[j]){
                temp = arr[j];
                arr[j] = arr[i];
                arr[i] = temp;
            }
        }
    }

    for(int i=n-1;i>=0;i--)
        printf("%d ",arr[i]);
}
```

7.

4、八皇后问题, 是一个古老而著名的问题, 是回溯算法的典型示例。该问题是国际西洋棋棋手马克斯·贝瑟尔于1848年提出: 在 $8 \times 8$ 格的国际象棋上摆放八个皇后, 使其不能互相攻击, 即任意两个皇后都不能处于同一行、同一列或同一斜线上, 问有多少种摆法。 (9分)

答:

```
void result(){  
  
}
```

更正：

```

int eight_queen_check(int row, int col) {

    for (int i = 0; i < 8; i++) if (chess[i][col] == 1) return 0;
    for (int i = row, j = col; i >= 0 && j >= 0; i--, j--) if (chess[i][j] == 1)
return 0;
    for (int i = row, j = col; i >= 0 && j < 8; i--, j++) if (chess[i][j] == 1) return
0;

    return 1;
}

void eight_queen_print() {
    printf("The %d method of eight queen\n", sum + 1);
    for (int i = 0; i < 8; i++) {
        for (int j = 0; j < 8; j++) {
            if (chess[i][j] == 1) printf("Q ");
            else printf("# ");
        }
        printf("\n");
    }
    printf("\n");
}

void eight_queen_search(int row) {
    if (row == 8) {
        eight_queen_print();
        sum++;
        return;
    }

    for (int col = 0; col < 8; col++) {
        if (eight_queen_check(row, col)) {
            chess[row][col] = 1;
            eight_queen_search(row + 1);
            chess[row][col] = 0;
        }
    }
}

int main() {

    eight_queen_search(0);
    printf("Total is: %d\n", sum);

    return 0;
}

```

1.

### 1. 基础原理

DeepSeek框架中默认的损失函数为交叉熵损失 (Cross-Entropy Loss)，请解释：

- 为什么交叉熵损失适合多分类任务？
- 如果遇到类别不平衡问题（如正负样本比例1:9），如何修改损失函数？

答：

改用Focal Loss 或 加权交叉熵损失

2.

### 2. 训练调优

在训练过程中发现验证集准确率波动较大，可能的原因有哪些？请至少列出3种并给出对应解决方案

答：

1. 学习率过高，使用学习率衰减
2. 小批量数据噪声，增大batch size
3. 模型过拟合，增加Dropout层

3.

### 3. 性能提升

现有DeepSeek模型在测试集上的F1-score为0.72，请设计一个优化方案流程图，要求包含以下环节

- 数据增强策略
- 模型结构改进方向
- 训练超参数调优方法

答：

1. 数据增强，随即裁剪+ Mixup
2. 模型改进，引入注意力机制模块
3. 超参数，网格搜索学习率/权重衰减组合

4.

4.

如果你是 DeepSeek 技术的开发者，你会如何平衡技术创新与社会责任？

答：

首先，从技术创新的角度来说，通过使用DeepSeek，可以带领各行各业在业务场景、文字工作、文案撰写、软硬件交互革新等等方面促进行业发展，提高行业的工作效率，减轻人为工作的负担，使工作的内容聚焦在更加需要想象力、自主设计的工作性质上来，将工作从繁杂冗余、重复性高的场景中解放出来，这是DeepSeek对于技术创新的有利支撑。

另外，从社会责任的角度来说，DeepSeek需要更多的学习工作者共同参与进来，制订一系列规范、合理、有助于社会发展的规则，技术创新不能跑偏，社会责任也离不开技术的创新，两者需要在相互促进、相互制约的情况下面合理地发展起来，帮助整个社会，乃至国家繁荣富强。

所以，总的来说，我的看法是技术创新与社会责任之间的平衡，应该是在社会责任的制约下去取得技术创新，同时再由技术创新来规范化社会责任。

5.

## 二、 题目 1

请详细阐述深度学习（Deep Learning）的概念，并分析它与机器学习（Machine Learning）的关系。要求至少从定义、方法、应用领域三个方面进行对比分析。

答：

在定义上，机器学习可以发现执行任务的规则，它是通过输入数据，然后根据预期输出的结果，衡量算法得到的结果与预期输出的结果之间的差异，也就是将输入数据变成有意义的输出结果。而深度学习是机器学习的一个分支，它是从数据中学习表示的一种新方法，强调从连续的层中学习，这些层会得到越来越有意义的表示。因此深度学习是指从更广泛、更深度的层中去不断获得越来越有意义的表示，以此来更加靠近预期结果的过程。

在方法上，深度学习的方法有Kreas、LightGBM等等

机器学习是一门多领域交叉学科，专门研究计算机怎样模拟或实现人类的学习行为，以获取新的知识或技能，重新组织已有的知识结构使之不断改善自身的性能。它基于数据，通过构建模型来进行预测和决策。深度学习是机器学习的一个分支领域，它是基于对数据进行表征学习的方法。深度学习通过构建具有很多层的神经网络模型，自动从大量数据中学习特征和模式。

机器学习方法包括决策树、支持向量机、朴素贝叶斯等传统算法。这些算法通常需要人工进行特征工程，即手动提取和选择对模型有意义的特征。深度学习主要依赖深度神经网络，通过构建包含多个隐藏层的神经网络结构，让模型自动从原始数据中学习不同层次的特征表示，减少了对人工特征工程的依赖。

机器学习应用在数据挖掘、生物信息学、金融风险评估等方面。深度学习应用在图像识别、语音识别、自然语言处理等对数据处理能力和模型复杂度要求较高的领域。

6.

#### 题目 2

什么是神经网络 (Neural Network)？请详细描述一个简单的前馈神经网络 (Feedforward Neural Network) 的结构，并解释各层的作用。

答：

神经网络是一种模仿生物神经网络的结构和功能的数学模型或计算模型。它由大量的节点和链接这些节点的边组成，通过对数据的学习来调整节点之间的连接权重，从而实现对数据的分类、预测、聚类等任务。

简单的前馈神经网络的结构包括输入层（网络与外部数据的接口）、隐藏层（输入层和输出层之间，实现对输入数据的非线性变换）、输出层（通过隐藏层传递过来的信息，产生最终的输出结果）

7.

#### 题目 3

解释卷积神经网络 (Convolutional Neural Network, CNN) 中的卷积层 (Convolutional Layer) 和池化层 (Pooling Layer) 的作用，并举例说明它们在图像识别任务中的应用。

答:

卷积层作用: 卷积层主要用于提取数据的局部特征, 在大幅减少参数数量的同时, 有效地提取图像的特征, 大大提高了模型的训练效率和泛化能力

池化层作用: 接在卷积层之后, 主要用于对特征图进行下采样操作, 常见的池化方法有最大池化和平均池化。

8.

#### 题目 4

电商平台上存在大量用户对商品的评价内容, 你的任务是构建一个模型, 判断这些评价中提及的商品属性 (如质量、外观、功能等) 是否符合用户期望, 判断结果分为符合、基本符合、不符合三种情况。设计这样一个商品属性期望判断模型需要考虑哪些因素?

答:

数据收集与整理, 实现多渠道收集、去噪处理

特征提取, 实现关键词提取、语义特征挖掘

模型选择与训练, 选择传统机器学习模型或者深度学习模型, 并进行训练

9.

#### 题目 5

在自然语言处理领域, 深度学习模型在机器翻译任务中发挥着关键作用。假设你要构建一个基于 Transformer 架构的深度学习模型, 实现英语到中文的翻译。请回答以下问题:

1. 简述 Transformer 架构中多头注意力机制 (Multi - Head Attention) 的工作原理及其在英译汉任务中的优势。(4 分)

答:

Transformer架构中的多头注意力机制工作原理及优势：

1. 工作原理：多头注意力机制将输入的查询、键和值分别通过多个不同的线性变换，得到多个不同的子查询、子键和子值。然后在每个子空间中分别计算注意力分数，即通过点积计算查询与键的相似度，通过Softmax归一化后得到注意力权重，再用注意力权重与值进行加权求和，得到每个子空间的输出。最后将多个子空间的输出拼接起来，再经过一次线性变换得到最终的输出。
2. 英译汉任务中的优势：多头注意力机制可以并行地从不同表示子空间中学习到不同的注意力信息，能够同时关注输入文本的不同部分，捕捉更丰富的语义和句法关系。

## 翻译题

In the realm of computer vision, deep learning has brought about revolutionary changes. Images, being a rich source of visual information, are composed of pixels arranged in a two - dimensional or multi - dimensional structure. Analyzing and understanding these complex visual patterns to extract meaningful insights has long been a goal in the field. Traditional methods for image recognition, such as using hand - crafted features like SIFT (Scale - Invariant Feature Transform) and HOG (Histogram of Oriented Gradients), have limitations in handling the vast diversity of images.

With the advent of deep learning, convolutional neural networks (CNNs) have emerged as a dominant approach. CNNs are designed to automatically learn hierarchical features from images through convolutional layers, pooling layers, and fully - connected layers. Over the years, researchers have been constantly improving CNN architectures. For example, the AlexNet, the first successful large - scale CNN, demonstrated the power of deep neural networks in image classification. Subsequently, more advanced architectures like VGGNet, ResNet, and Inception have been proposed. ResNet, in particular, introduced the concept of residual connections, which significantly alleviated the problem of vanishing gradients during training, enabling the training of much deeper neural networks.

Recently, the focus has shifted towards exploring the application of CNNs in more complex tasks, such as object detection in real - time scenarios, semantic segmentation for autonomous driving, and image generation using generative adversarial networks (GANs). As the amount of available image data continues to grow exponentially, and the computing power of GPUs (Graphics Processing Units) becomes more powerful, the potential of deep - learning - based computer vision techniques seems limitless. However, challenges still remain, such as dealing with limited data in certain specialized domains, and ensuring the interpretability of the models, especially in safety - critical applications.

答：

在计算机视觉领域，深度学习已经带来了革命性改变。图像，作为一种丰富的视觉信息资源，由两种不同维度结构的像素组成，分别是二维度结构、多维度结构。分析和理解这些复杂的视觉模型去抽取有意义的内容已经在这个领域里形成一个长期目标了。传统的图像认知方法，例如尺度不变特征变换（SIFT）和方向梯度直方图（HOG）等手工特征，已经限制了海量图像多样性的处理。

随着深度学习的出现，卷积神经网络（CNNs）成为了主流方法。CNNs被设计用于自动化学习层次化特征，从卷积层、池化层和全连接层上。这些年之后，研究者已经不断地提升了CNN的架



构。例如，AlexNet，首个成功的巨大规模模型的卷积神经网络，证明了深度神经网络在图像定义上的强大能力。随后，越来越多先进的架构，像VGGNet,ResNet和Inception被推出。特别是ResNet，引入了残差连接的概念，这具有重大意义地缓解了训练过程中梯度消失的问题，使许多更深层次的神经网络发挥作用。

最近，大众焦点转换到了朝着探索卷积神经网络处理更多复杂任务的方向出发，例如检测实时场景中的目标对象，自动驾驶中的语义分割和使用生成对抗网络进行图像生成。随着可利用图像数据的数量继续潜在的成长，并且图像处理单元的计算力量变得更强大，基于计算机视觉技术的深度学习潜力似乎是无穷无尽的。然而，挑战仍然很多，例如处理确定领域里的有限数据和确保模型的可预测性，尤其是安全性验证应用。