

一. 单选题（20 道题目，每道 2 分，共 40 分）

1. 在 20 世纪 50 年代，艾伦·纽厄尔（Allen Newell）和赫伯特·西蒙（Herbert Simon）共同研发的“逻辑理论家”程序是一套基于公理和规则的推理系统。它成功地证明了数学名著《数学原理》一书第二章 52 个定理中的 38 个定理。这个程序是（ ）学派的典型成就。

- A. 符号主义（Symbolism）
- B. 连接主义（Connectionism）
- C. 行为主义（Behaviorism）
- D. 贝叶斯主义（Bayesism）

2. 某用户使用了人工智能助手作为办公辅助，他对应用程序说：“请提醒我下午三点开会。”人工智能助手随后确认了会议提醒。以下的人工智能领域在此应用程序中一定有所体现的是？（ ）

- A. 自然语言处理（Natural Language Processing, NLP）
- B. 计算机视觉（Computer Vision, CV）
- C. 生成式人工智能（Artificial Intelligence Generated Content, AIGC）
- D. 专家系统（Expert System）

3. 小爱是一位金融分析师，他的工作是为客户提供投资建议。为了提供更准确的建议，他考虑使用人工智能算法分析市场数据和识别潜在的投资机会。他希望这个系统能够处理大量的历史交易数据、新闻报道、经济指标以及社交媒体上的舆论，从而预测特定股票或资产的未来表现。在小爱的系统中，以下哪项人工智能技术应用最有可能被用来辅助分析和预测特定股票的未来表现？（ ）

- A. 生成式人工智能（Artificial Intelligence Generated Content, AIGC）用于产生股票说明文档
- B. 计算机视觉（Computer Vision, CV）用于解析图表和图形数据
- C. 深度学习（Deep Learning）用于时间序列预测（Time Series Prediction）
- D. 专家系统（Expert System）用于股票行业分类

4. 某网络购物平台使用了人工智能技术进行数据分析，使用的分析方法不恰当的是（ ）。

- A. 用已有用户注册信息和历史购买记录训练决策树算法（Decision Tree）对用户分类，并根据分类对用户进行商品推荐。

- B. 根据商家最近的经营及服务数据，用聚类算法（Clustering）判断出各商家在各自主营类目下所属的商家层级。
- C. 根据商家近几年的成交数据，用回归算法（Regression）拟合出用户未来一个月的消费金额。
- D. 用 K 近邻算法（K Nearest Neighbors, KNN）分析出是否应该对购买了鸡蛋的买家推荐购买西红柿。
5. 人工智能领域中，以下哪项技术主要用于从大量未标记数据中发现隐藏的模式或结构？（ ）
- A. 监督学习（Supervised Learning）
- B. 非监督学习（Unsupervised Learning）
- C. 强化学习（Reinforcement Learning）
- D. 自然语言处理（Natural Language Processing）
6. 在机器学习中，以下哪种方法是监督学习的典型方法？（ ）
- A. K-均值聚类（K-Means Clustering）
- B. 主成分分析（Principal Component Analysis, PCA）
- C. 决策树分类（Decision Tree）
- D. 自编码器（Autoencoder）
7. K 近邻算法（K-Nearest Neighbors, KNN）是一种常见的（ ）。
- A. 监督学习方法    B. 无监督学习方法    C. 强化学习方法    D. 聚类学习方法
8. 在机器学习时，根据用途的不同，会将原始数据集拆分为多个子集。其中用于在训练结束后，进行模型评估的子集是（ ）。
- A. 训练集（Training Set）
- B. 验证集（Validation Set）
- C. 测试集（Testing Set）
- D. 任务集（Task Set）

9. 在机器学习中，梯度下降算法（Gradient Descent Method）的主要目的是什么？（     ）

- A. 计算损失函数（Loss Function）的最小值
- B. 确定数据集的分布
- C. 评估模型的复杂度
- D. 计算损失函数的最大值

10. 混淆矩阵（Confusion Matrix）是机器学习中用于评估模型性能的一种表格工具，特别适用于监督学习中的分类问题。它用  $n$  行  $n$  列的矩阵形式表示了模型对于样本分类的情况，并将模型预测的结果与样本的实际标签进行对比，其中  $n$  为类别的数量。一个包含“正”、“负”两种类别的典型的二分类问题的混淆矩阵如下：

	预测为正类别	预测为负类别
实际为正类别	True Positive（TP）	False Negative（FN）
实际为负类别	False Positive（FP）	True Negative（TN）

其中 True Positive（TP）表示实际是正类别，分类系统也预测为正类别的样本数量；False Negative（FN）表示实际是正类别，分类系统错误预测为负类别的样本数量。

由此可以推断，False Positive（FP）和 True Negative（TN）分别代表（     ）

- A. FP 表示实际是负类别，分类系统错误预测为正类别的样本数量；TN 表示实际是负类别，分类系统也预测为负类别的样本数量。
- B. FP 表示实际是负类别，分类系统正确预测为负类别的样本数量；TN 表示实际是负类别，分类系统错误预测为正类别的样本数量。
- C. FP 表示实际是负类别，分类系统无法给出预测的样本数量；TN 表示实际是负类别，分类系统错误预测为正类别的样本数量。
- D. FP 表示实际是负类别，分类系统错误预测为正类别的样本数量；TN 表示实际是负类别，分类系统无法给出预测的样本数量。

11. 应用混淆矩阵，我们可以计算出一系列的评估指标以对模型进行性能评估。其中，精确率（Precision）指标被定义为“模型预测正确的正样本占有所有预测为正样本数的比例”。以下哪个公式最符合精确率指标定义？（     ）

- A.  $\frac{TP}{TP + TN}$
- B.  $\frac{TP}{TP + FP}$
- C.  $\frac{TP}{TP + TN + FP + FN}$
- D.  $\frac{TP + TN}{TP + TN + FP + FN}$

12. 某城市的环保部门为提高垃圾分类效率，采用智能垃圾分拣系统对垃圾样本进行分类。在系统测试阶段，研究人员进行了垃圾样本收集并借助混淆矩阵来评估系统的准确性。该矩阵显示了各类垃圾被正确及错误分类的数量。基于这一数据，系统的性能可以通过计算诸如

预测 \ 真实	可回收垃圾	有害垃圾	厨余垃圾	其他垃圾
可回收垃圾	250	50	30	10
有害垃圾	20	180	10	10
厨余垃圾	20	10	350	50
其他垃圾	10	10	20	420

准确率（Accuracy）和召回率（Recall）等指标来分析。混淆矩阵如下所示：

其中，准确率的定义为：

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

其中 TP, TN, FP, FN 的定义与二分类问题的定义类似。例如，类别 1 的 FN 表示实际为类别 1，分类系统错误预测为不是类别 1 的样本数。

根据混淆矩阵数据，可以计算智能垃圾分拣系统对可回收垃圾的识别准确率约为（ ）。

- A. 67%
- B. 76%
- C. 83%
- D. 90%

13. 在训练神经网络（Neural Network）时，反向传播算法（Back Propagation）主要的作用

是（ ）。

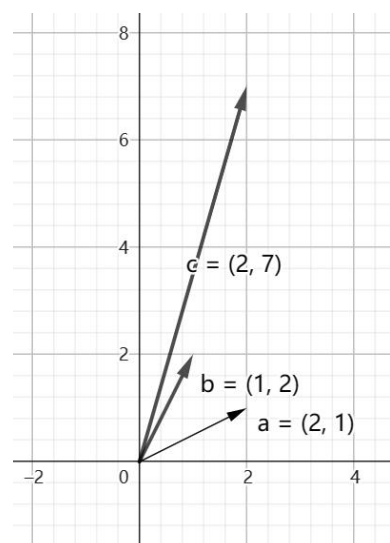
- A. 计算网络的输出
- B. 调整网络的权重（Weight）和偏置项（Bias）
- C. 评估网络的性能
- D. 确定网络的结构

14. 在神经网络中，以下哪个概念指的是将线性求和的结果进行非线性（non-linear）映射？

（ ）

- A. 激活函数（Activation Function）
- B. 权重（Weight）
- C. 偏置项（Intercept）
- D. 层间连接（Connection）

15. 在自然语言处理中，词嵌入技术（Word Embedding）是指用一个低维向量（Vector）来表示一个对象（例如单词）的方法，这个向量能够表达相应对象的某些特征，且不同对象对应的向量之间的某种度量方法能够反应对象之间的相似性。现已知在某一种词嵌入技术下，单词 X 对应的向量为  $\vec{a} = (2, 1)$ ，单词 Y 对应的向量为  $\vec{b} = (1, 2)$ ，单词 Z 对应的向量为  $\vec{c} = (2, 7)$ 。在该技术下，向量夹角的余弦值（即方向相似性）可用来度量两个向量的相似程度，余弦值越大，表示两个向量越相似，其表征的两个词也越相近。在这种度量方法下，单词 X, Y, Z 最有可能对应的单词是（ ）。



- A. X-猫, Y-狗, Z-游泳
- B. X-游泳, Y-狗, Z-猫
- C. X-猫, Y-游泳, Z-狗
- D. X-狗, Y-游泳, Z-猫

16. 小爱同学在一次针对分类问题的人工智能训练和测试过程中,生成了一个神经网络模型,此神经网络模型在训练集上得到的分类准确率(Accuracy)为 97%。但是,当小爱同学把这个模型放到测试集上测试时,分类准确率只有 48%。这种表现说明神经网络模型出现了( )。

- A. 欠拟合(Under-fitting)
- B. 梯度消失(Vanishing Gradient)
- C. 过拟合(Over-fitting)
- D. 训练不收敛(Convergence)

17. 面对上述这种情况,下列做法中可能会让模型效果提升的方案是( )。

- A. 增加神经网络的层数
- B. 添加参数丢弃层(Dropout)以丢弃部分神经网络的参数
- C. 增加每层神经元的个数
- D. 增加训练轮次(Epoch)

请阅读下面一段材料,回答 18, 19, 20 题。 卷积神经网络(Convolutional Neural Networks, CNNs)是一类包含卷积计算且具有深度结构的前馈神经网络,是深度学习(Deep Learning)的代表模型之一,已经被广泛应用于图像处理(Image Processing)等场景中。图像在计算机中一般用矩阵(二维数表)进行储存,矩阵(Matrix)中的每一个元素值代表图像中对应点颜色的深浅,数值越大的地方颜色越深。例如:  $A$  就是一个  $3 \times 3$  的矩阵,其中值为 128 的位置颜色最深,值为 1 的位置颜色最浅。

$$A = \begin{pmatrix} 1 & 5 & 2 \\ 128 & 3 & 21 \\ 2 & 7 & 1 \end{pmatrix}$$

为了更好地引入矩阵的卷积运算,我们首先引入矩阵的内积运算(记作  $\odot$ ),用来表示两个矩阵所有对应项的乘积的和。对于两个  $n \times n$  的矩阵  $A$  和  $B$ , 如下所示:

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix} \quad B = \begin{pmatrix} b_{11} & \cdots & b_{1n} \\ \vdots & \ddots & \vdots \\ b_{n1} & \cdots & b_{nn} \end{pmatrix}$$

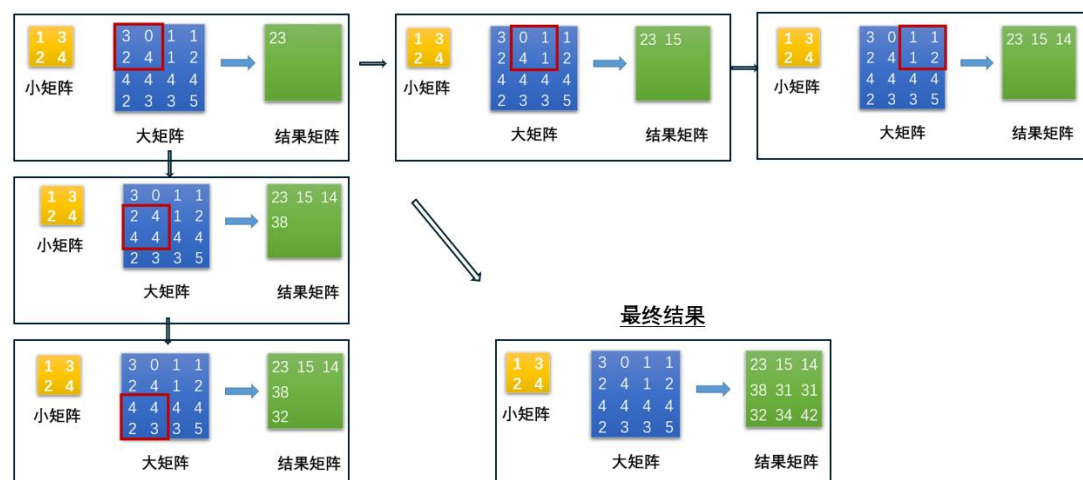
内积运算定义如下： $A \odot B = a_{11}b_{11} + a_{12}b_{12} + \cdots + a_{1n}b_{1n} + a_{21}b_{21} + \cdots + a_{nn}b_{nn} = \sum_{i=1}^n \sum_{j=1}^n a_{ij}b_{ij}$

下图是一个简单的两个  $2 \times 2$  的矩阵内积运算过程如下：

$$\begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix} \odot \begin{pmatrix} 0 & 3 \\ 5 & 1 \end{pmatrix} = 1 \times 0 + 3 \times 3 + 2 \times 5 + 4 \times 1 = 23$$

卷积运算可以在内积运算基础上进行定义。对于一个  $m \times m$  的矩阵  $A$  和一个  $n \times n$  的矩阵  $B$  ( $m \leq n$ )； $A$  与  $B$  的卷积运算记作  $A * B$ 。定义如下：

$$C = A * B = \begin{pmatrix} c_{1,1} & \cdots & c_{1,n-m+1} \\ \vdots & \ddots & \vdots \\ c_{n-m+1,1} & \cdots & c_{n-m+1,n-m+1} \end{pmatrix} \quad \text{其中, } c_{i,j} = A \odot \begin{pmatrix} b_{i,j} & b_{i,j+1} & \cdots & b_{i,j+m-1} \\ b_{i+1,j} & b_{i+1,j+1} & & \vdots \\ \vdots & & \ddots & \\ b_{i+m-1,j} & \cdots & & b_{i+m-1,j+m-1} \end{pmatrix}.$$



上图是一个更加直观的卷积运算示例。

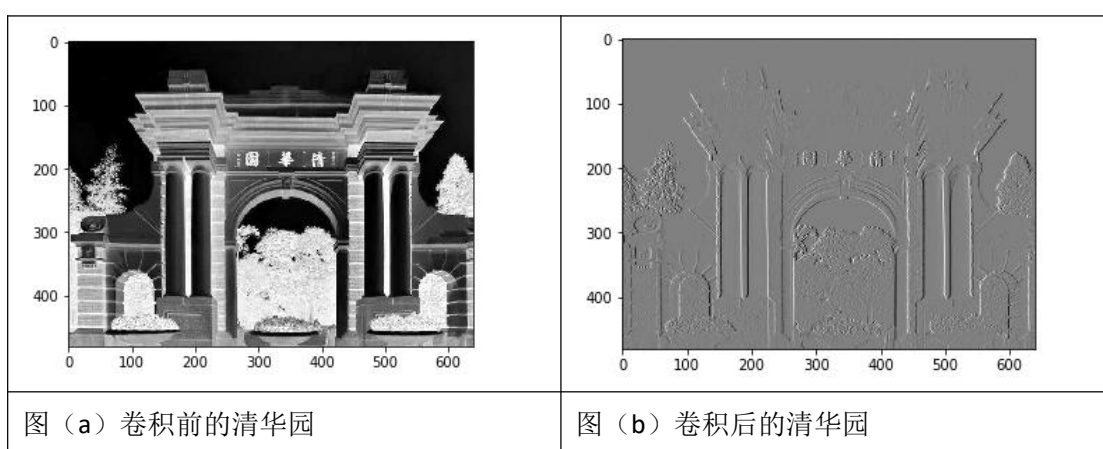
18. 请根据上面的材料回答下列问题：

已知  $A = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ ,  $B = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ , 则  $A \odot B = ( \quad )$

- A.  $\begin{pmatrix} 1 & 2 \\ 3 & -4 \end{pmatrix}$     B.  $\begin{pmatrix} 4 & -2 \\ 6 & -2 \end{pmatrix}$     C. 2    D. 10

19. 已知  $A = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ ,  $B = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix}$ , 则  $A * B =$  ( )
- A.  $\begin{pmatrix} 1 & 3 \\ 3 & 3 \end{pmatrix}$     B.  $\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$     C.  $\begin{pmatrix} 1 & 3 & 1 \\ 3 & 3 & 1 \\ 1 & 1 & 1 \end{pmatrix}$     D.  $\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$

20. 下图中图 (a) 为清华园的图片, 记为矩阵  $B$ , 经过与矩阵  $A$  卷积运算 ( $A * B$ ) 后, 得到的图片效果如图 (b) 所示, 则可以推断矩阵  $A$  可能是 ( )



- A.  $\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$     B.  $\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$     C.  $\begin{pmatrix} 1 & -1 \\ 1 & -1 \end{pmatrix}$     D.  $\begin{pmatrix} -1 & -1 \\ 1 & 1 \end{pmatrix}$

## 二. 概念解释 (每题 10 分, 共计 20 分)

21. 用简要语言描述“有监督学习 (Supervised Learning)”和“无监督学习 (Unsupervised Learning)”的概念, 同时列举出各自的一个实际应用问题。

22. 正则化 (Regularization) 是机器学习中用于防止模型过拟合的一种技术。 $l_0$  正则化、 $l_1$  正则化和  $l_2$  正则化是三种常见的正则化方法。下面是三种正则化的公式, 其中  $\mathbf{w}$  为模型中的权重:

- $l_0$  正则化:  $l_0 = \sum_{i=1}^n |w_i|^0$ , 实际上  $|w_i|^0$  被定义为  $I_{w_i \neq 0}$  (即当  $w_i$  非零时为 1, 否则为 0);
- $l_1$  正则化:  $l_1 = \sum_{i=1}^n |w_i|$ ;



- $l_2$  正则化:  $L_2 = \sum_{i=1}^n w_i^2$ 。

- (1) 请简述正则化的作用。
- (2) 请比较三种正则化方法的特点。

### 三. 程序补全 (10 分)

23. 请结合下面对二分查找算法的描述, 补全一段使用 Python 语言编写的算法代码。二分查找算法 (Binary Search Algorithm), 也称为对数查找算法, 是一种在有序数组 (Sorted Array) 中查找特定元素的高效算法, 其中有序数组是数组中的数按照从小到大的顺序排列 (升序) 或者从大到小的顺序排列 (降序)。二分查找的基本原理是通过不断将数组分为两半, 缩小搜索范围, 直至找到目标元素或确定元素不存在于数组中。

算法步骤:

- ① 确定搜索范围: 设定两个指针 (也可理解为数组索引, Index), 一个指向数组的起始位置 (通常称为左指针), 另一个指向数组的结束位置 (通常称为右指针)。
- ② 计算中间位置: 计算当前搜索范围的中间位置的索引 (通常取左指针和右指针的中点)。
- ③ 比较中间元素: 检查中间位置的元素是否等于目标值。
  - a) 如果中间元素等于目标值, 返回该元素的索引, 查找成功;
  - b) 如果中间元素不等于目标值, 根据中间元素与目标值的大小关系, 确定目标值位于当前中间位置的左侧还是右侧。
- ④ 更新搜索范围:
  - a) 如果目标值小于中间元素, 说明目标值在左侧子数组中, 更新右指针位置为中间位置索引减一;
  - b) 如果目标值大于中间元素, 说明目标值在右侧子数组中, 更新左指针位置为中间位置索引加一。
- ⑤ 重复步骤: 重复步骤②至④, 直到找到目标元素或左指针位置大于右指针位置 (此时搜索范围为空, 说明目标元素不存在于数组中)。

因为每次迭代都会将搜索范围减半, 二分查找算法的时间复杂度为  $O(\log n)$ , 其中  $n$  是数组的长度。上述算法的 Python 代码如下:

```

# 二分查找，A为一个有序列表（严格升序，即没有重复元素），x为要查找的列表元素。
# 如果 x 在列表 A 中存在返回其在 A 中对应元素位置的下标。
# 如果不存在返回 -1
def BinarySearch(A, x):
    L = 0
    R = len(A) - 1
    while ____ (1) ____:
        mid = ____ (2) ____
        if ____ (3) ____:
            R = mid - 1
        elif A[mid] < x:
            ____ (4) ____
        else:
            return mid
    return ____ (5) ____

```

#### 四. 编程解决问题（每题 15 分，共 30 分）

24. Python 中使用 print 函数可以将字符串输出到屏幕上，例如：

```

print('*')
print('*' * 10)

*
*****

```

其中，`'*' * 10`得到的是将`'*'`重复 10 次的字符串。请用 Python 语言编写一个名为 `PrintDiamond` 的函数，它接受一个正奇数作为参数。例如：当用行数 11 作为参数调用它时，它将在屏幕上输出以下图形（此图形共计 11 行，中间无空行）：

```

      *
     ***
    *****
   *********
  ***********
 *************
*****
*****
 *****
  *****
   *****
    *****
     *****
      *

```

请写出完整的函数定义和函数调用。

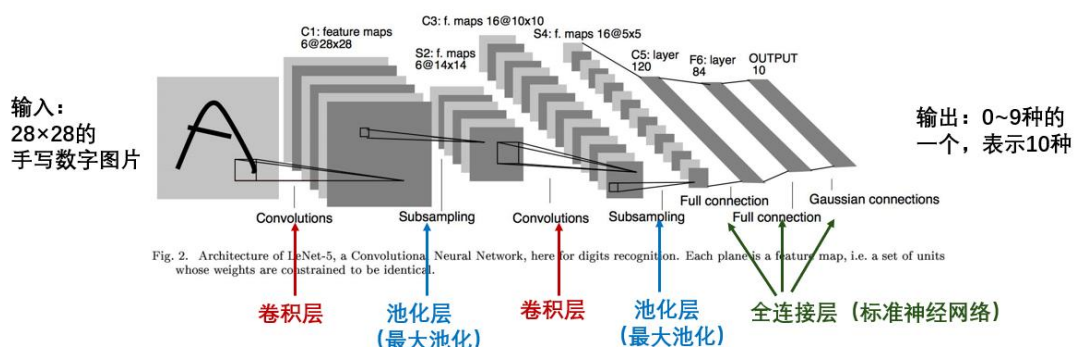
25. 下图是 LeCun Yann 团队于 1998 年提出的 LeNet-5 卷积神经网络（Convolutional Neural Network）结构，用于识别手写数字。其中网络的输入为  $28 \times 28$  的手写数字图片，输出为 0-9 数字中的 1 个数字。整个网络由 2 个卷积层（Convolutional Layer）、2 个池化层（Pooling Layer）和 3 个全连接层（Fully Connected Layer）构成。卷积层的输出采用 ReLU 函数作为激活函数（Activation Function）。通过查阅 PyTorch 文档，小爱学习了二维卷积层、激活函数和池化层的定义方法，文档显示如下：

```
class torch.nn.Conv2d(in_channels, out_channels, kernel_size, stride=1, padding=0, dilation=1, groups=1, bias=True)
部分参数：
·in_channels(int) - 输入信号的通道
·out_channels(int) - 卷积产生的通道
·kernel_size(int or tuple) - 卷积核的尺寸
·stride(int or tuple, optional) - 卷积步长
·padding(int or tuple, optional) - 图像填充模式
```

```
class torch.nn.ReLU(inplace=False)
参数：
·inplace - 选择是否进行覆盖运算
```

```
class torch.nn.MaxPool2d(kernel_size, stride=None, padding=0, dilation=1, return_indices=False, ceil_mode=False)
部分参数：
·kernel_size(int or tuple) - max pooling的窗口大小
```

小爱同学接着使用 PyTorch 在 Python 中定义了 LeNet-5 的神经网络结构，请你帮他补全缺失的部分。（提示：第二个卷积层设定  $\text{stride} = 1$ ,  $\text{padding} = 0$ ）



```
class LeNet(nn.Module):
    def __init__(self):
        super(LeNet, self).__init__()
        self.conv = nn.Sequential(
            nn.Conv2d(1, 6, 5, stride=1, padding=2), # padding=2 保证输入尺寸匹配
            nn.ReLU(True), # 激活函数是 ReLU
            nn.MaxPool2d(2, 2), # 最大池化
            

缺失的部分


            nn.Conv2d(16, 16, 5, stride=1, padding=0), # stride=1, padding=0
            nn.ReLU(True), # 激活函数是 ReLU
            nn.MaxPool2d(2, 2), # 最大池化
            nn.Linear(120, 120),
            nn.Linear(120, 84),
            nn.Linear(84, 10))
        self.fc = nn.Sequential(
            nn.Linear(400, 120),
            nn.Linear(120, 84),
            nn.Linear(84, 10))
    def forward(self, x):
        out = self.conv(x) # 卷积层
        out = out.view(out.size(0), -1) # 展平层
        out = self.fc(out) # 全连接层
        return out
```

NOAI 2024 水平测试轮  
答案

一. 单选题（20 道题目，每道 2 分，共 40 分）

1.	A	2.	A	3.	C	4.	D
5.	B	6.	C	7.	A	8.	C
9.	A	10.	A	11.	B	12.	D
13.	B	14.	A	15.	B	16.	C
17.	B	18.	C	19.	A	20.	C

NOAI 2024 水平测试轮  
答案

二. 概念解释（每题 10 分，共计 20 分）

21.

有监督学习概念解释（2 分）

● 答案示例

在有监督学习中，模型通过训练数据集进行学习，该数据集包含输入特征和对应的标签（输出）。通过这些带标签的数据，模型学习如何将输入映射到正确的输出，从而在给定新的未标记数据时能够做出预测。这种方法类似于有教师指导的学习过程。

● 评分细则

1. 需要明确答出模型训练时，数据集有标签或需要标注数据或数据集包括输入特征和对应的标签（输出）。未答出这点的不得分，答出的得 1.5 分。
2. 在 1 得分的基础上答出对于未标记数据能做出预测（推理）的再得 0.5 分。

有监督学习举例（3 分）

● 答案示例

**邮件分类。**在这个问题中，**邮件（输入特征）**被标记为“垃圾邮件”或“非垃圾邮件”（**输出标签**）。有监督学习算法可以训练来识别新邮件是否为垃圾邮件。

● 评分细则

1. 答出任何具体的应用并对其输入输出和解决问题过程进行了合理描述的得 2 分。
2. 包含对于该应用中的有标签数据集描述的得 1 分。

无监督学习概念解释（2 分）

● 答案示例

监督学习涉及处理没有标签的数据。模型试图在数据中找到模式或结构，如聚类或异常点。这种方法更像是让模型自己探索和发现数据的内在关系，而没有人告诉它什么是正确的答案。

● 评分细则

1. 需要明确答出模型训练时，数据集无标签或不需要标注数据。未答出这点的不得分，答出的得 1.5 分。
2. 在 1 得分的基础上答出模型自己探索发现数据内部模式/结构的再得 0.5 分。

无监督学习举例（3 分）

● 答案示例

**无监督学习实际应用问题：市场细分。**在这个问题中，**消费者（输入特征）**的数据没有预先定义的标签。无监督学习算法可以帮助识别消费者群体中的不同细分市场，基于他们的购买行为、偏好等。

● 评分细则

1. 答出任何具体的应用并对其输入输出和解决问题过程进行了合理描述的得 2 分。
2. 包含对于该应用中的无标签数据进行了描述的得 1 分。

22.

正则化的作用（6分）：

● 答案示例

正则化方法是一种在机器学习模型训练过程中常用的防止过拟合的技术。其基本思想是在经验损失的基础上增加一个结构化损失或惩罚项，即参数的  $L_0$ ,  $L_1$  或  $L_2$  范数（分别为非零参数个数、参数绝对值的和、参数平方和），作为正则项来限制模型的复杂程度，提升模型的泛化能力。

● 评分细则

答出以下关键点的每个得 2 分：① 防止过拟合；② 惩罚项；③ 限制模型复杂度，提升泛化能力。

比较三种正则化方法的特点（4分）：

● 答案示例

$l_0$  正则化：能够产生稀疏权重矩阵，即许多权重为零，这样可以减少模型的复杂度并提高可解释性。但是在优化过程中是非光滑和不可微的，这使得传统的梯度下降方法难以应用。

$l_1$  正则化：对比  $l_2$  正则化，倾向于产生稀疏解，即许多权重参数会变为零，从而使得模型权重稀疏化。这不仅有助于降低模型的复杂度，还有助于特征选择，因为它可以自动地从模型中排除不重要的特征。

$l_2$  正则化：倾向于使权重均匀地分布，而不是选择性地将某些权重设为零。这有助于模型的平滑性和泛化能力。

● 评分细则

按以下关键点进行评分：

①  $l_0$  正则化——稀疏、不光滑难以使用各 1 分；

②  $l_1$  正则化——稀疏 1 分；

③  $l_2$  正则化——平滑 1 分。

23.

# ( 1 )	1 ← P
---------	-------

# (1)	L <= R      #或 R>=L, 缺少=不得分
# (2)	(L+R) // 2      #其他计算中点写法均可, 但必须考虑到整数类型, 如 int(L+(R-L)/2)
# (3)	A[mid] > x      # 或 x < A[mid] 也可
# (4)	L=mid+1
# (5)	-1

24.

## ● 完整程序示例

```
[1]: def PrintDiamond(lines):  
    l = int(lines/2)  
    for i in range(l):  
        s = ' ' * (l-i)  
        s = s + '*' * (2*i+1)  
        print(s)  
    for i in range(l+1):  
        s = ' ' * i  
        s = s + '*' * (lines - 2 * i)  
        print(s)
```

```
[2]: PrintDiamond(11)
```

```

      *
    ***
  *****
*****
*****
*****
*****
*****
*****
*****
*****

```

### ● 评分细则

1. 函数定义格式正确的得 2 分。
2. 函数调用格式正确的得 1 分。
3. 能正确打印出图形的得 12 分；程序逻辑正确，但是不能正确打印出图形的得 2 分，例如：对于整数的处理错误等。

25.

● 正确答案

```
nn.Conv2d(6, 16, 5, stride=1, padding=0),  
nn.ReLU(True),  
nn.MaxPool2d(2, 2)  
)
```

● 评分细则

1. 答对第一行的得 7 分，后两行各 4 分（缺括号或者逗号不扣分）
2. 第一行，数字 6，16，5 各 2 分，stride=1，padding=0 得 1 分；
3. 第二行，只要选择 ReLU 得 4 分；
4. 第三行，选择 MaxPool2d 得 2 分，尺寸填写（2,2）得 2 分。