

## 神经网络试卷

### 一、选择题

1、人工智能从诞生至今，其发展历程大体可以分为（）

- A、推理期->学习期->知识期
- B、学习期->知识期->推理期
- C、知识期->推理期->学习期
- D、知识期->学习期->推理期

推理, 知识, 学习

2、哪种情况的模型为理想（）

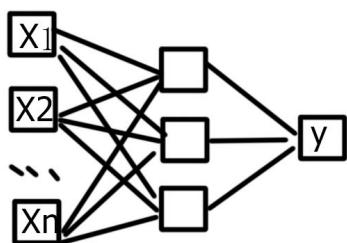
- A、高偏差，高方差
- B、低偏差，高方差
- C、高偏差，低方差
- D、低偏差，低方差

3、以下哪种关系正确描述卷积神经网络（）

- A、卷积神经网络是一种 one to one 模型
- B、卷积神经网络是一种 one to many 模型
- C、卷积神经网络是一种 many to one 模型
- D、卷积神经网络是一种 many to many 模型

4、所有的激活函数都是 Sigmoid，优化器为随机梯度下降，将所有的权重和偏置都初始化为

0，当输入  $x \in \mathbb{R}^{n-H}$  网络后，输出是多少？（）

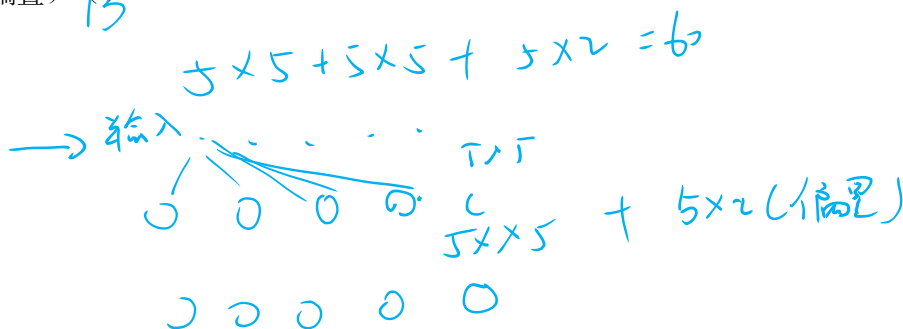


$$\text{Sigmoid}(x) = \frac{1}{1+e^x} = \frac{1}{2}$$

- A、-1
- B、0
- C、0.5
- D、1

5、考虑如下 2 层全连接神经网络，每一层都有五个神经元，请问网络的参数量有多少（包括权重和偏置）

- A、50
- B、60
- C、70
- D、80



6、以下关于丢弃法（Dropout）的说法，正确的是 **B**

- A、在训练过程中，Dropout 每次丢弃的神经元都是随机的
- B、Dropout 可以有解决模型的欠拟合现象 **→ 过拟合**
- C、Dropout rate 应当设置越高越好 **→ 欠拟合**
- D、以上描述都正确

7、以下关于梯度消失问题的哪项是正确的？ **C**

- A、可以用 Tanh 函数替代 Sigmoid 函数作为激活函数，缓解梯度消失现象。 **X**
- B、梯度消失可以使用网络的更深层更好地学习数据特征
- C、与 Sigmoid 函数相比，ReLU 函数可以有效解决梯度消失问题 **✓**
- D、以上描述均正确

8、以下哪种方法常用来解决梯度爆炸问题 **B**

- A、使用非线性激活函数，例如 ReLU
- B、梯度截断，例如按层截断
- C、优化网络结构，例如使用残差连接
- D、使用更好的参数初始化方法，例如使用单位矩阵初始化权重

9、考虑到非线性，以下哪个函数适用于作为神经网络的激活函数（ ）

- A、 $f(x) = \begin{cases} \min(x, 1x) & | x \geq 0 \\ \min(x, 1x) & | x < 0 \end{cases}$
- B、 $f(x) = \begin{cases} \min(x, 1x) & | x \geq 0 \\ \max(1, 1x) & | x < 0 \end{cases}$
- C、 $f(x) = x$
- D、 $f(x) = 0.9x + 1$

10、以下哪项不是神经网络的超参数

## 二、判断题

- 1、
- 2、
- 3、
- 4、
- 5、

### 三、名词解释

1、感受野：

2、转置卷积（反卷积）：*用于上采样的-种操作，一定程度上实现特征图尺寸的放大*

3、平均汇聚：

4、损失函数：

5、过拟合：

*一种池化操作，计算特征图局部区域  
内的平均值，从而得到一个新的  
尺寸更小的特征图*

### 四、简答题

1、激活函数需要具备的性质，并简要说明为什么

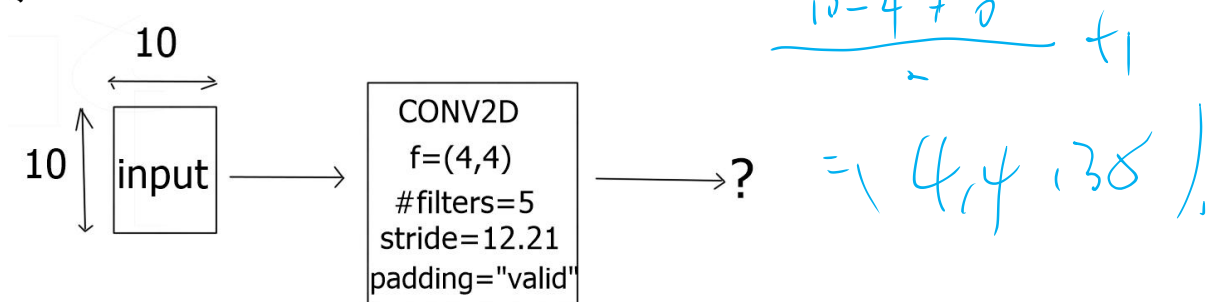
2、现有数据集包括若干幅灰度图，你想要设计一个分类器完成 5 分类任务，下属两种的方案你会选择哪种，请简述说明为什么

①将输入图像开成一个 100 维的向量，取得其输入到一个包含 5 个神经元的全连接层

②将灰度图直接输入到一个卷积层，实验 5 个尺寸为  $10 \times 10$  的卷积核运算

*1. 卷积层可以自动提取局部特征，不用手动设计特征提取  
2. 局部连接可以减少参数量，但牺牲与...  
3. 卷积层...降低计算复杂度与存储量...  
4. 卷积层...对图像平移不变性*

4、考虑将一个  $10 \times 10$  的图像输入到以下卷积模块，其中输入的形状为  $(nH, nW, nC) = (10, 10, 1)$ ，CONV2D 中实验 35 个尺度为  $4 \times 4$  的卷积核，且  $\text{padding}=0$ ，补偿设置为  $(2, 2)$ ，请问当输入图像经过该卷积层后，输出图像的尺寸是什么？请以  $(nH, nW, nC)$  的形式表示



5、一些卷积神经网络会使用到 Concatenate 层，请简述该层的作用

*张量拼接，融合特征*

6、超参数对网络性能的影响很大，不同的机器学习任务往往需要不同的超参数，请分析超参数优化的信息，并举例至少 3 个超参数的配置方法

*1. 网格搜索  
2. 随机搜索  
3. 贝叶斯优化*

7、非特殊说明默认填充为 1，步长为 1。

CONVx-N 表示一个卷积层，该层使用了 N 个卷积核，每个卷积核的高度和宽度都为 x

Pool-n 表示一个形状为  $n \times n$  的池化池，填充为 0，步长为 2

FLATTEN 表示将上一层的输入展开

FC-N 表示一个具有 N 个神经元的全连接层

层	维度	参数数量
---	----	------

in (BATCHNDRM)  
速度进行倍数和位移, 增加 2 个参数 (倍数和偏移),  
6 × 8,  
= 16 (每个通道有 2 个参数)。

Input	$32 \times 32 \times 3$	0
CONV3-8	$32 \times 32 \times 8$	$3 \times 3 \times 3 \times 8 + 8$
Leaky ReLU <i>激活函数</i>	$32 \times 32 \times 8$	0
Pool-2 <i>池化无参数</i>	$16 \times 16 \times 8$	0
BATCHNDRM	$16 \times 16 \times 8$	$8 \times 2 = 16$
CONV3-16	$16 \times 16 \times 16$	$3 \times 3 \times 8 \times 16 + 16$
Leaky ReLU	$16 \times 16 \times 16$	0
Pool-2	$8 \times 8 \times 16$	0
FLATTEN <i>展开成一维向量</i>	$(8 \times 8 \times 16) = 1024$ <i>写成这样</i>	0
FC-10	10	$1024 \times 10 + 10$

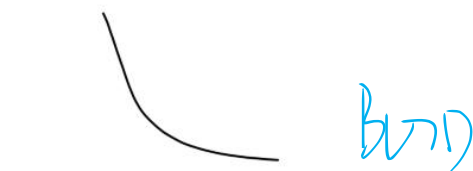
## 五、大题

1、你想要设计一个分类器用于预测给定音频中的乐器种类, 总共 50 种乐器, 你首先创建了一个数据集, 该数据集包含了每种乐器的音频数据

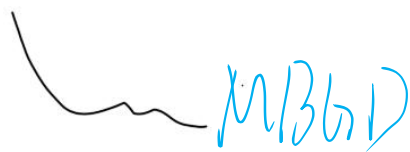
①你是要批量梯度下降法 (BGD) 优化损失函数, 但是模型训练损失很高, 检查代码后发现, 为随机打乱训练数据, 轻微在实验该算法时, 随机打乱训练集能否提高模型性能表现, 请简述理由

②你发现, 使用 BGD 算法优化损失函数始终不能达到令人满意的结果, 所以你决定使用其他梯度下降算法, 例如小批量梯度下降算法 (MBGD) 和随机梯度下降算法 (SGD), 请分析为什么使用 MBGD 算法优化损失函数的效果比 BGD 算大更好 ① 泛化 ② 平稳收敛

③下图中的 3 个损失函数值得曲线分别对应着 SGD, MBGD, BGD 算法训练过程, 请在每张图的下方填写相应的算大名称



( )



( )



( )

④过大的学习率和过小的学习率分别会导致模型的损失函数出现什么变化

⑤你决定使用动量梯度下降算法根据动量公式  $V_{\partial w} = \beta V_{\partial w} + (1 - \beta) \partial J$ , 写出权重  $w$  的更新规则, 其中学习率用  $\alpha$  表示, 动量超参数  $\beta$  表示, 损失函数用  $J$  表示, 加权平均用  $V$  表示

随机打乱训练集可以提高模型的性能表现, 原因如下:

- 避免样本顺序对模型学习的影响:
  - 如果训练集中样本的顺序是有规律的 (例如按类别或按时间顺序), 模型可能会学习到非真正的特征, 导致过拟合。
  - 通过随机打乱训练集, 保证每个批次中样本的多样性, 从而提高模型对未知数据的泛化能力。
- 提升梯度更新的随机性:
  - 随机打乱训练集会改变每次梯度下降中所选样本, 从而使梯度更新更加随机, 避免陷入局部最优。
  - 这种随机性特别适合小批量梯度下降 (MBGD) 和随机梯度下降 (SGD), 可以使梯度更新更加鲁棒和高效。
- 提高模型的泛化能力:
  - 随机打乱训练集会使模型更难以记住样本顺序, 从而降低过拟合风险, 从而提升在测试集上的性能。

### 第4题: 学习率的影响

#### 1. 学习率过大:

- 梯度更新幅度过大, 可能导致损失函数在最优值附近不断震荡, 无法收敛。
- 甚至可能导致损失函数发散, 无法下降。

#### 2. 学习率过小:

- 梯度更新幅度太小, 优化过程非常慢, 可能导致收敛时间过长。
- 甚至可能陷入局部最优或停滞不前。

#### 1. 动量更新:

$$V_{\partial w} = \beta V_{\partial w} + (1 - \beta) \frac{\partial J}{\partial w}$$

#### 2. 权重更新:

