人工智能期末复习课短版

袁肖赟，上海交通大学计算机学院人工智能研究院，yuanxiaoyun@sjtu.edu.cn

Contents

1 使用方式

这份短版适合考前几天突击。它不替代详细版，而是把最容易考成选择题、判断题、填空题和简单计算题的内容压缩到一起。

短版与详细版的区别：详细版用于按章理解，短版用于考前突击。短版不放完整推导和大图，只保留公式、判断点、易混点和最小计算模板。

复习顺序：先背公式和关键结论，再做最小计算模板，最后用高频易混点和判断句查漏。

建议：

- 只剩 1 天：看“按题型复习”“公式与计算”“最小计算模板”“高频易混点”。
- 还有 2 到 3 天：按“基础与残差网络”到“强化学习”逐章过一遍。
- 遇到不理解的点，再回详细版对应章节看图和例子。

怎么用。看短版时不要只看名词。每看到一个概念，至少要能回答三件事：它是什么，它解决什么问题，它容易和什么概念混淆。

2 按题型复习

本课程题型以选择题、判断题和填空题为主，少量简单计算题。考前突击时，不需要把详细推导背下来，但要能识别概念、判断说法、套用最小公式。

选择题：重点看概念对比。常见问法是“哪个说法正确”“哪个模型适合某任务”“哪个模块负责某功能”。

判断题：重点看绝对化表述。遇到“一定、只能、不会、必须、所有”这类词，要特别小心。

填空题：重点背核心名词、公式符号含义和简单结论，例如 C_{out}、padding、stride、hidden state、Q/K/V、回报、Bellman 方程。

计算题：只准备最小模板：全连接参数量、卷积输出尺寸、卷积参数量、池化结果、RNN 参数量、回报和 Bellman 两状态方程。

- 选择题常考：CNN vs MLP，RNN vs Transformer，Encoder-only vs Decoder-only，监督学习 vs 强化学习。
- 判断题常考：参数量是否乘输出大小；Transformer 是否需要位置编码；LayerNorm 是否跨 token；残差是否减少参数。
- 填空题常考：卷积核个数决定输出通道数；RNN 靠参数共享处理变长序列；LSTM 的遗忘门、输入门、输出门；强化学习目标是期望回报。
- 计算题常考：把公式中的 $H, W, K, P, S, C_{in}, C_{out}$ 对上；先算空间大小，再算通道数，再算参数量。

3 公式与计算

3.1 基础公式

- **全连接层**：
$$y = Wx + b, \qquad W \in R^{d_{out} \times d_{in}}, \quad b \in R^{d_{out}}$$
参数量为 $d_{out}d_{in} + d_{out}$。
- **ReLU**：
$$\text{ReLU}(x) = \max(0, x)$$
负数变 0，正数不变；激活函数逐元素作用。
- **Softmax**：
$$p_i = \frac{e^{z_i}}{\sum_j e^{z_j}}$$
把 logits 变成概率，所有概率和为 1；大的更大，小的更小，更接近 one-hot。
- **二分类交叉熵**：
$$H(y, p) = -\frac{1}{n}\sum_{i=1}^{n} [y_i \log f_\theta(x_i) + (1 - y_i) \log(1 - f_\theta(x_i))]$$
- **多分类交叉熵**：
$$H(y, p) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{c=1}^{C} y_{i,c} \log p_{i,c}$$
真实类别概率越大，loss 越小。
- **残差连接**：
$$y = x + F(x)$$
保留原输入信息，使深层网络更容易训练。

3.2 CNN 公式

- **卷积输出大小**：
$$H_{out} = \left\lfloor \frac{H + 2P - K}{S} \right\rfloor + 1, \qquad W_{out} = \left\lfloor \frac{W + 2P - K}{S} \right\rfloor + 1$$
H, W 是输入高和宽，K 是卷积核大小，P 是 padding，S 是 stride。
- **输出通道数**：输出通道数由卷积核个数决定，通常记为 C_{out}。
- **卷积参数量**：
$$\#\text{params} = K^2 C_{in} C_{out} + C_{out}$$
最后的 C_{out} 是 bias；参数量不乘输出高和宽。
- **多通道卷积**：
$$Y_j = \sum_{c=1}^{C_{in}} X_c * K_{j,c} + b_j, \qquad j = 1, \ldots, C_{out}$$
$*$ 表示空间卷积；$K_{j,c} \in R^{K \times K}$ 是第 j 个 filter 在第 c 个输入通道上的卷积核切片。

3.3 RNN、Attention 与 RL 公式

- **RNN 状态更新**：
$$S_t = f(Ux_t + WS_{t-1} + b), \qquad y_t = g(VS_t + c)$$
U, W, V, b, c 在所有时间步共享，所以序列变长时参数量不随时间步增加。
- **缩放点积注意力**：
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$
先用 QK^T 算相关性，再 softmax 成权重，最后对 V 加权求和。
- **折扣回报**：
$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots$$
γ 越大越重视长期奖励。
- **价值函数**：
$$V^\pi(s) = \mathbb{E}_\pi[G_t \mid S_t = s], \qquad Q^\pi(s, a) = \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a]$$
- **Bellman 方程**：
$$V = R + \gamma PV$$
V 是状态价值向量，R 是奖励向量，P 是状态转移矩阵。

4 最小计算模板

这一节专门用来应对填空题和最简单计算题。考试不会考复杂推导，但可能考“形状怎么算、参数量怎么算、公式里每个量是什么意思”。

4.1 基础层计算

例子 1：**全连接层参数量**。输入维度 $d_{in} = 4$，输出维度 $d_{out} = 3$：

$$W \in R^{3\times 4}, \qquad b \in R^3$$

权重参数量 $3 \times 4 = 12$，偏置参数量 3，总参数量 15。

$$\#\text{params} = d_{out}d_{in} + d_{out} = 3 \times 4 + 3 = 15$$

例子 2：**激活函数逐元素作用**。对向量 $x = [-2, 0, 3]^T$：

$$\text{ReLU}(x) = [0, 0, 3]^T$$

激活函数不是只作用在整个向量上，而是对每个元素分别作用。

例子 3：**Softmax 与交叉熵直觉**。二分类真实标签是正类，交叉熵可看成

$$H = -\log p_{\text{true}}$$

若模型给正类概率 0.9，则 $H = -\log 0.9$；若只给 0.6，则 $H = -\log 0.6$，loss 更大。交叉熵惩罚的是“真实类别概率太小”。

4.2 CNN 计算

例子 4：卷积输出尺寸。输入 $32\times32\times3$，卷积核 $K=3$，padding $P=1$，stride $S=1$，卷积核个数 $C_{out}=16$：

$$H_{out}=W_{out}=\left\lfloor\frac{32+2\times1-3}{1}\right\rfloor+1=32$$

输出尺寸为 $32\times32\times16$。

例子 5：卷积参数量。上一题中，参数量为

$$\#\text{params}=K^2C_{in}C_{out}+C_{out}=3^2\times3\times16+16=448$$

注意最后 $+16$ 是 bias；不要再乘 32×32。

例子 6：stride 改变空间大小。输入 $32\times32\times3$，$K=3$，$P=1$，$S=2$，$C_{out}=16$：

$$H_{out}=W_{out}=\left\lfloor\frac{32+2-3}{2}\right\rfloor+1=16$$

输出为 $16\times16\times16$。这说明 stride 为 2 的卷积可以实现下采样。

例子 7：池化。对局部区域

$$\begin{bmatrix}1&3\\2&0\end{bmatrix}$$

max pooling 输出 3，mean pooling 输出 $(1+3+2+0)/4=1.5$。

4.3 序列与注意力计算

例子 8：Embedding 矩阵大小。词表大小 $|\mathcal{V}|=10000$，embedding 维度 $d=128$：

$$E\in R^{|\mathcal{V}|\times d}=R^{10000\times128}$$

一个 one-hot 向量乘这个矩阵后得到 128 维词向量。

例子 9：RNN 参数量。输入维度 $d_x=4$，隐藏维度 $d_h=3$，输出维度 $d_y=2$：

$$\#\text{params}=d_hd_x+d_hd_h+d_yd_h+d_h+d_y$$

$$=3\times4+3\times3+2\times3+3+2=32$$

这些参数在所有时间步共享。

例子 10：Self-attention 形状。输入 $X\in R^{n\times d_m}$，若 $W^Q,W^K\in R^{d_m\times d_k}$，$W^V\in R^{d_m\times d_v}$，则

$$Q,K\in R^{n\times d_k},\qquad V\in R^{n\times d_v}$$

$$QK^T\in R^{n\times n},\qquad \text{softmax}(QK^T/\sqrt{d_k})V\in R^{n\times d_v}$$

例子 **11**：**Causal mask**。对序列“我 爱 小 猫”，第 2 个位置“爱”只能看“我、爱”，不能看“小、猫”。所以生成任务中当前位置不能提前看到未来 token。

4.4 强化学习计算

例子 **12**：折扣回报。若未来三个奖励为 $1, 0, 2$，$\gamma = 0.5$，则

$$G_t = 1 + 0.5 \times 0 + 0.5^2 \times 2 = 1.5$$

例子 **13**：**Bellman** 两状态方程。若

$$P = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \qquad R = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \qquad \gamma = 0.5$$

则

$$V(s_1) = 1 + 0.5V(s_2), \qquad V(s_2) = 2 + 0.5V(s_1)$$

解得 $V(s_1) = 8/3,\ V(s_2) = 10/3$。

5 基础与残差网络

参数：神经网络中通过训练学习的数，典型是权重 W 和偏置 b；偏置不是手工常数，而是可学习参数。

激活函数：激活函数提供非线性；如果没有非线性，多层线性变换整体仍等价于一层线性变换。

- 单个神经元常写成 $z = w^T x + b$，再经过激活函数输出。
- 全连接层可以看成多个并行神经元。
- 深层网络可能出现梯度消失、梯度爆炸和退化；退化指更深的网络训练效果反而更差。
- 残差连接不是为了减少参数，也不是直接解决过拟合；它的核心作用是让信息和梯度更容易传播。

6 卷积神经网络 CNN

CNN：CNN 利用图像的局部关联性和参数共享，用局部卷积窗口提取空间特征。

- 图像有局部关联性：相邻像素关系强，距离越远关系越弱；图像数据更接近幂律衰减关联。
- 图像任务常希望对平移、旋转、缩放有一定不变性，但图像像素顺序不能随便打乱。
- CNN 与 MLP 的区别：MLP 通常全连接，参数多；CNN 使用局部连接和参数共享，更适合图像。
- 卷积核是可训练小窗口，同一个卷积核在整张图上滑动。
- Padding 控制边缘和输出大小，不增加可学习参数。
- Stride 是滑动步长；stride 越大，输出空间尺寸越小。
- Stride 为 2 的卷积可以做可学习下采样，也就是卷积和类似 pooling 的降采样合在一起。
- 池化通常没有可学习参数；max pooling 取最大值，mean pooling 取平均值。
- GAP 把每个通道压成 1 个数，改变空间尺寸、不改变通道数，常用于减少后续全连接层参数量。
- 1×1 卷积主要改变通道数或做通道间线性组合。
- 卷积、激活、池化、全连接等层可以灵活搭配，没有固定网络格式。

7 RNN 与 LSTM

RNN： RNN 按时间步处理序列，用隐藏状态保存历史信息，同一组参数在不同时间步共享。

LSTM： LSTM 是改进的 RNN，用记忆状态和门控机制缓解普通 RNN 难以保留长期信息的问题。

- 文本是离散符号序列，长度可变，有顺序、上下文和长程依赖。
- RNN 能处理变长序列的核心是参数共享：序列变长只增加计算步数，不增加参数量。
- Tokenization 把原始文本变成 token 序列。
- One-hot 高维稀疏，只表示编号；embedding 低维稠密、可训练，可表达语义相似性。
- Embedding 矩阵大小为 $|\mathcal{V}| \times d$，one-hot 乘这个矩阵得到对应词向量。
- Next Token Prediction 用前文预测下一个 token，标签来自文本本身，属于自监督学习，常用交叉熵。
- BPTT 是沿时间展开后的网络反向传播；长序列完整反传开销大，常用截断 BPTT；很长时间步上可能出现梯度消失或梯度爆炸。
- LSTM 三个门：遗忘门保留旧记忆，输入门写入新信息，输出门控制输出信息。
- LSTM 常见设计原则：先做线性变换；门用 sigmoid，输出 0 到 1；候选信息常用 tanh。

例子：**为什么 RNN 能处理变长文本**。同一个 RNN 单元可以处理第 1 个 token、第 2 个 token、一直到第 t 个 token。句子变长时，只是重复更多时间步，参数 U, W, V, b, c 不会随着句子长度增加。

8 Transformer

Transformer： Transformer 以注意力机制为核心，能够并行处理 token，并直接建模 token 之间的关系。

位置编码：注意力机制本身不关心顺序，所以要给 token 加入位置信息；输入通常是 embedding 加位置编码。

- RNN/LSTM 通常串行处理 token；Transformer 可以并行输入、并行输出。
- Q 是 Query，表示要查什么；K 是 Key，表示可被匹配的标识；V 是 Value，表示被汇总的信息。
- 在 self-attention 中，Q, K, V 都由同一个输入 X 乘可训练矩阵得到。
- Q 和 K 的最后一维必须相同，因为要做点积；K 和 V 的数量必须相同，因为一个 key 对应一个 value。
- 课件重点： self-attention 中 Q, K, V 来自同一段序列，所以 token 数相同；扩展到 cross-attention 时，Q 的数量可以和 K/V 不同。
- 注意力四步：线性映射得到 Q/K/V；QK 点积算关联强度；softmax 得到权重；权重对 V 加权求和。
- W^O 把 attention 输出映射回模型维度，方便和输入做残差相加。
- Causal mask 遮住未来 token，保证生成当前位置时不能提前看到答案。
- LayerNorm 是逐 token 在特征维度内归一化，不在 token 之间算均值方差。
- FNN 是逐 token 的 MLP；token 之间的信息交互主要发生在 attention 中。
- 贪婪解码每步取最大概率 token；采样解码按概率抽样，同一输入可能有不同输出。

例子：**Causal mask 的判断**。生成“我 爱 小 猫”时，在“爱”这个位置预测下一个 token，模型只能看“我、爱”，不能提前看“小、猫”。所以“Decoder-only 可以看到未来 token”是错误说法。

9 训练现象：频率、凝聚、平坦性

频率原则：神经网络训练通常先学习低频成分，再学习高频成分。

平坦解：参数附近一片区域 loss 都较低，扰动后仍稳定，通常泛化更好。

- 函数可看成不同频率成分叠加；低频变化慢、平滑，高频变化快、振荡、细节多。
- 判断频率原则时要控制幅值影响，比较同幅值、不同频率的成分。
- 早停可以减少对高频噪声的过度拟合。
- 两种频率要区分：图像频率描述像素变化快慢；响应频率描述模型输出对输入扰动的敏感程度。
- 过参数化模型参数很多，但训练仍可能得到有规律、可泛化的解；这和优化隐式偏置有关。
- 全零初始化通常不可取，因为同层神经元会保持对称。
- 小初始化下，同层 ReLU 神经元方向和转折点可能趋同，这叫参数凝聚。
- 转折点相同的 ReLU 神经元可以合并理解；不同转折点提供不同非线性位置。
- 尖锐解附近 loss 变化快，扰动后不稳定；平坦解更鲁棒。
- 小批量梯度噪声有助于跳出尖锐区域；合理较大的学习率更容易得到平坦解，但太大会不收敛。

例子：**早停为什么可能有用**。如果主趋势像低频的 $\sin x$，噪声像高频抖动，神经网络训练早期先学主趋势，继续训练才更容易拟合高频噪声；早停就是在过度拟合噪声前停下来。

10 大模型

基础模型：基础模型是参数量大、经过大规模数据训练、可适应多种下游任务的模型。

- 大语言模型以文本 token 为主要输入输出；多模态数据也可以 token 化。
- Encoder-only 看双向上下文，适合理解任务，代表 BERT。
- Decoder-only 使用 causal mask，只看当前及以前 token，适合生成任务，代表 GPT、Llama、DeepSeek。
- Encoder-Decoder 先编码输入再解码输出，适合翻译、摘要等任务。
- 预训练学习通用语言和知识能力；SFT 用标注指令数据继续训练。
- 强化学习后训练可增强推理或对齐偏好，不是用来取代预训练，也不是直接缩短响应时间。
- 提示词工程改输入提示，通常不更新模型参数。
- 知识蒸馏让 Student model 学习 Teacher model，用于压缩和部署。
- Scaling law 描述规模、数据、算力增加时测试损失通常按幂律下降。
- 上下文学习不更新参数；思维链输出中间推理步骤；幻觉是看似合理但实际错误的生成内容。
- 初始化会影响训练路径和推理能力；课件例子中，小初始化通常更有利于学习可泛化规律。

例子：**提示词工程和微调**。给模型输入“请一步一步思考”属于提示词工程，通常不更新参数；用一批数学题继续训练模型属于微调或后训练，会更新参数。

11 强化学习

强化学习：强化学习是智能体在环境中采取动作、获得奖励并转移到新状态，通过试错学习最大化期望回报的方法。

- 强化学习是决策型任务；监督学习是预测型任务。
- 强化学习依赖奖励反馈，不依赖每一步的标注标签；奖励可能延迟出现。
- 状态是环境完整信息；观察是智能体实际看到的信息，可能只是状态的一部分。
- 动作空间可以离散，也可以连续。
- 情节型任务有终止状态；连续型任务没有固定终止状态。
- 探索是尝试未知动作获取信息；利用是选择当前认为更优的动作。
- 马尔可夫性：给定当前状态后，未来状态分布只依赖当前状态；有动作时，下一状态只依赖当前状态和当前动作。
- MDP 常包含状态、动作、转移概率、奖励和折扣因子。
- 策略 $\pi(a|s)$ 表示在状态 s 下选择动作 a 的规则。
- $V^\pi(s)$ 是从状态 s 出发的期望回报；$Q^\pi(s,a)$ 是先采取动作 a 后的期望回报。
- DP 通常需要环境模型和转移概率；MC 通过多次采样取平均，方差较大。
- 深度强化学习用神经网络近似价值函数或策略；DQN 用神经网络拟合 Q 值。
- RLHF 用人类偏好构造奖励信号，使大模型输出更符合人类偏好和价值观。

例子：**探索和利用**。去常吃且评价稳定的餐厅是利用；尝试一家没去过的新餐厅是探索。只利用可能错过更好的选择，只探索又可能长期得不到稳定回报。

12 高频易混点

这一节不是增加新知识，而是把最容易在判断题和选择题里混淆的地方放在一起。

- **正确说法**：卷积参数共享；RNN 时间步共享参数；Transformer 需要位置编码；LayerNorm 逐 token 归一化；强化学习最大化期望回报。
- **错误说法**：卷积参数量要乘输出高宽；残差连接减少参数；attention 自动知道顺序；FNN 负责 token 间交互；提示词工程通常更新参数。
- **常见选择题问法**：哪个模型适合图像？CNN。哪个模型适合自回归生成？Decoder-only。哪个模块遮住未来 token？Causal mask。
- **常见填空题问法**：输出通道数由卷积核个数决定；RNN 处理变长序列靠参数共享；Bellman 方程为 $V = R + \gamma PV$。

- **参数量 vs 输出大小**：输出尺寸由输入大小、卷积核、padding、stride 决定；卷积参数量由 K, C_{in}, C_{out} 和 bias 决定。
- **卷积核个数 vs 卷积核大小**：卷积核大小是 $K \times K$；卷积核个数通常是 C_{out}，决定输出通道数。
- **Padding vs stride**： padding 主要补边和控制输出大小；stride 是滑动步长，通常会让空间尺寸变小。
- **池化 vs stride 卷积**：池化通常不可学习；stride 卷积可以在提取特征时完成可学习下采样。
- **Max pooling vs mean pooling**： max 取局部最大值，mean 取局部平均值。
- **GAP**：全局平均池化把每个通道压成 1 个数，减少后续全连接参数量。

- **One-hot vs embedding**： one-hot 只是编号，高维稀疏；embedding 是可训练低维向量。
- **RNN 变长能力**：靠同一组参数在时间步共享，不是靠为每个长度重新训练模型。
- **RNN vs Transformer**： RNN 通常串行处理；Transformer 可并行处理，但需要位置编码。
- **BPTT vs 普通反向传播**：普通反向传播沿网络层反传；BPTT 沿时间展开后的网络反传。
- **LSTM 三个门**：遗忘门保留旧记忆，输入门写入新信息，输出门控制输出信息。

- **Q/K/V**：Q 查，K 被匹配，V 被汇总；真正被加权求和的是 V。
- **QK 维度**：Q 和 K 的最后一维必须相同，因为要做点积。
- **K/V 数量**：K 和 V 的数量必须相同，因为每个 key 要对应一个 value。
- **Q 数量**：课件重点是 self-attention，Q、K、V 的 token 数相同；扩展到 cross-attention 时，Q 的数量可以和 K/V 不同。
- **Causal mask**：用于生成任务，防止当前位置看到未来 token。
- **LayerNorm**：对每个 token 内部的特征维度归一化，不在 token 之间归一化。
- **FNN**：Transformer 里的 FNN 是逐 token 的 MLP，token 之间的信息交互主要在 attention 中完成。

- **频率原则**：通常先学低频，再学高频；不要把幅值影响和频率影响混在一起。
- **图像频率 vs 响应频率**：图像频率看像素变化；响应频率看模型输出对输入扰动的敏感程度。
- **凝聚 vs 小网络**：凝聚后的大网络可等效更少神经元，但直接训练小网络不一定同样容易。
- **平坦解 vs 尖锐解**：平坦解附近 loss 变化慢，更稳定；尖锐解附近 loss 变化快，更不稳定。
- **大学习率**：合理较大学习率更容易得到平坦解，但太大会震荡或不收敛。

- **Encoder-only vs Decoder-only**：Encoder-only 看双向上下文，适合理解；Decoder-only 看当前及以前 token，适合生成。
- **预训练 vs SFT vs RL 后训练**：预训练学基础能力；SFT 学会按指令回答；RL 后训练增强推理或对齐偏好。
- **提示词工程**：改输入，不更新模型参数。
- **上下文学习**：不更新参数，靠任务描述和示例适应当前任务。
- **幻觉**：输出看似合理但实际错误，不等于模型“不流畅”。

- **强化学习 vs 监督学习**：强化学习是决策型任务，用奖励反馈；监督学习是预测型任务，用标签。
- **状态 vs 观察**：状态是环境完整信息；观察是智能体看到的信息，可能只是状态的一部分。
- **回报 vs 奖励**：奖励是一步反馈；回报是从当前时刻开始的累计折扣奖励。
- **探索 vs 利用**：探索尝试未知动作；利用选择当前认为更优动作。
- V **vs** Q：$V^\pi(s)$ 只给状态；$Q^\pi(s,a)$ 给状态和第一个动作。
- **DP vs MC**：DP 通常需要环境模型和转移概率；MC 通过采样取平均。

13 最后只背这些

- 基础：参数、偏置、激活函数、softmax、交叉熵、残差连接。
- CNN：局部连接、参数共享、padding、stride、输出尺寸、参数量、池化、1×1 卷积。
- RNN/LSTM：变长序列、参数共享、tokenization、embedding、RNN 公式、BPTT、LSTM 三个门。
- Transformer：并行、位置编码、Q/K/V、scaled dot-product attention、causal mask、LayerNorm、FNN。
- 训练现象：频率原则、早停、参数凝聚、平坦解、小批量和大学习率。
- 大模型：三种架构、预训练、SFT、RL 后训练、知识蒸馏、scaling law、上下文学习、思维链、幻觉。
- 强化学习：状态、动作、奖励、回报、探索利用、MDP、策略、价值函数、Bellman、DP/MC/DQN/RLHF。