

# 简历相关知识

---

## 简历相关知识

### AlexViT

Q1.主要创新是什么

Q2.ViT 与 AlexNet 的特性对比

Q3.如何结合 ViT 与 AlexNet?

1. 特征提取阶段：分层融合
2. 架构优化：轻量化设计

Q4.性能提升的原因

1. 准确度提升：互补特征增强诊断可靠性
2. 速度提升：结构优化降低冗余计算

Q5. 简单介绍一下AlexNet，并说一下为什么这篇论文里要用到AlexNet

Q6. Transformer的架构是什么

编码器 (Encoder)

- (1) 多头自注意力层
- (2) 前馈神经网络层

解码器 (Decoder)

- (1) Masked 自注意力层
- (2) 多头交叉注意力层
- (3) 前馈神经网络层与归一化

位置编码 (Positional Encoding)

输入与输出处理

Transformer 的核心优势

Q7.LN和BN的区别

### 边缘计算

Q1.什么是边缘计算

Q2. 什么是物联网

Q3. 什么是非正交多址NOMA

Q4. 什么是PPO

Q5.什么是Fisher 判别法

Q6.什么是模糊聚类

### 一、模糊聚类的核心原理

## AlexViT

---

### Q1.主要创新是什么

1. 首先是模型架构融合创新：创新性的把ViT 与 AlexNet 的跨架构整合：

- 需同时关注**局部细节**（如微动脉瘤、出血点的形态）和**全局结构**（如病变区域在视网膜中的分布范围）。
- 传统单一模型（如纯 CNN 或纯 Transformer）难以兼顾细节特征提取与全局特征整合，导致识别精度或速度受限。

该模型通过融合ViT（Vision Transformer）和AlexNet的优势，针对糖尿病视网膜病变（DR）等级诊断任务，在识别速度和准确度上实现了双重提升。以下从技术背景、融合逻辑、优势分析三方面展开解释：

2. 其次是交叉领域的融合，将这一全新模型运用到实际的医学交叉领域中，采用现实生活中采集到的数据集，具有现实意义

## Q2.ViT 与 AlexNet 的特性对比

技术	核心原理	优势	局限性
AlexNet (CNN)	基于卷积层提取图像局部特征，通过池化层降低维度，全连接层完成分类。	擅长捕捉图像局部细节（如视网膜血管纹理），计算效率高。	对全局特征建模能力弱，需堆叠多层卷积层提取高层语义。
ViT (Transformer)	将图像分块为序列，通过自注意力机制（Self-Attention）建模全局像素关联。	擅长捕捉长距离依赖（如视网膜全局病变分布），模型参数效率高。	对小尺寸图像（如医学切片）需大量计算资源，输入分辨率敏感。

## Q3.如何结合 ViT 与 AlexNet?

### 1. 特征提取阶段：分层融合

- **底层特征（AlexNet 主导）**：利用 AlexNet 的前几层的卷积层和池化层（如 Conv1-Conv2）提取视网膜图像的基础特征，如边缘、纹理、简单几何形状。这些局部特征对病变细节（如微小出血点）的捕捉至关重要。
- **高层特征（ViT 主导）**：将 AlexNet 输出的特征图输入 ViT 模块，通过自注意力机制建模不同区域特征的全局关联。例如，识别某个局部出血点是否与其他区域的渗出物存在关联，从而判断病变等级。

## 2. 架构优化：轻量化设计

- **减少 ViT 计算量：**对 AlexNet 输出的特征图进行拉伸，再输入 ViT 模块，避免直接处理高分辨率图像导致的计算成本激增，提升识别速度。
- **端到端训练：**融合后的模型采用统一的损失函数进行端到端训练，自动优化 CNN 与 Transformer 模块的特征交互权重。

## Q4.性能提升的原因

### 1. 准确度提升：互补特征增强诊断可靠性

AlexNet 的卷积层捕捉微动脉瘤等局部病变的精细结构，ViT 的自注意力机制分析这些局部特征在视网膜全局中的分布模式（如病变是否集中在黄斑区），两者结合可更全面地判断病变等级。

- 例如：单一 CNN 可能误将视网膜血管的正常褶皱识别为病变，而 ViT 通过全局分析可排除此类干扰；单一 ViT 可能因局部特征提取不足漏检微小病变，而 AlexNet 可增强细节辨识度。

### 2. 速度提升：结构优化降低冗余计算

- **卷积层的高效性保留：**AlexNet 的前向传播计算量低于纯 ViT 模型（尤其在低分辨率输入时），作为底层特征提取器可快速生成压缩后的特征图，减少后续 ViT 模块的处理压力。
- **注意力机制的针对性应用：**ViT 仅作用于高层语义特征（而非原始像素），自注意力计算的序列长度显著缩短。

## Q5. 简单介绍一下AlexNet，并说一下为什么这篇论文里要用到AlexNet

AlexNet 是深度学习领域的里程碑模型，它首次在 ImageNet 大规模视觉识别挑战赛中以显著优势夺冠，打破了传统机器学习方法的瓶颈，标志着深度学习在计算机视觉领域的崛起。

AlexNet 是一个深度卷积神经网络（CNN），共 8 层（5 层卷积层 + 3 层全连接层），包含约 6000 万个参数，其核心设计创新如下：

- **非线性激活函数 ReLU：**替代传统的 Sigmoid/Tanh 函数，缓解梯度消失问题，加速训练收敛。
- **多 GPU 并行训练：**利用 2 块 NVIDIA GTX 580 GPU 进行模型并行训练，突破当时单卡内存限制，提升计算效率。

- **局部响应归一化 (LRN)**：通过模拟生物神经元的侧抑制机制，增强特征的分度（后续研究表明该模块作用有限，现代模型已较少使用）。
- **Dropout 正则化**：在全连接层随机丢弃神经元，降低模型复杂度，避免过拟合。

在本篇论文中使用AlexNet的原因主要有

1. **经典**：AlexNet是首个成功的大规模深度CNN模型，结构相对简单（仅5层卷积+3层全连接），作为经典基准易于复现和对比，尤其适合验证新方法（如混合架构）的基础有效性。
2. **计算效率**：相比更深的VGG（19层）或ResNet（50+层），AlexNet参数量少（约6000万）、计算成本低，适合资源受限场景或需要快速迭代的实验设计。
3. **大核卷积特性**：其浅层的 $11 \times 11$ 、 $5 \times 5$ 大卷积核能更粗粒度地捕捉局部纹理特征（如边缘、颜色分布），而现代CNN（如VGG/ResNet）倾向于堆叠小核（ $3 \times 3$ ），可能过度抽象底层细节，不利于与全局模块（如Transformer）的特征互补。
4. **兼容性与可解释性**：AlexNet的层级较浅，特征图分辨率较高（如Conv1输出 $55 \times 55$ ），便于与后续模块（如Transformer）直接拼接，避免因多次下采样导致空间信息过度丢失；同时其结构透明性高，利于分析混合模型中各组件的作用。
5. **任务适配**：若论文任务侧重局部-全局特征融合（如细粒度分类），AlexNet的早期大核设计能保留更多空间细节，而深层CNN的强抽象能力可能削弱与Transformer的协同效果。

## Q6. Transformer的架构是什么

Transformer是一种基于自注意力机制的深度学习架构，由**Encoder**和**Decoder**堆组成。其核心组件包括：

### 编码器 (Encoder)

编码器由 **N 个相同的层**堆叠而成（论文中  $N=6$ ），每层包含两个子层：

- **多头自注意力层 (Multi-Head Self-Attention)**
- **前馈神经网络层 (Feed-Forward Neural Network)**

#### (1) 多头自注意力层

- **作用**：让模型在不同的子空间中捕捉序列内部的依赖关系，提升特征表达能力。

- **核心概念：**

- **Query (Q) 、Key (K) 、Value (V)：** 通过输入向量与三个可学习矩阵 ( $W^Q, W^K, W^V$ ) 相乘得到，用于计算注意力权重。
- **注意力分数计算：** 对每个 Query，计算其与所有 Key 的**相似度**（通常用点积），并通过 Softmax 归一化得到权重，再与 Value 加权求和，得到该位置的输出。
$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$
 ( $d_k$ 为 Key 的维度，缩放因子  $\sqrt{d_k}$  用于稳定梯度)
- **多头机制：** 将 Q、K、V 拆分为 h 个并行的头 (Head)，每个头独立计算注意力，最后将结果拼接并线性变换，增强模型捕捉多维度特征的能力。

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O$$
 ( $W^O$ 为拼接后的线性变换矩阵)

- **残差连接与层归一化：** 自注意力层输出后，通过**残差连接 (Skip Connection)** 与输入相加，再进行**层归一化 (Layer Normalization)**，缓解梯度消失并稳定训练。
- $\text{LayerNorm}(x + \text{MultiHead}(x))$

## (2) 前馈神经网络层

- **作用：** 对自注意力层的输出进行非线性变换，增强模型表达能力。
- **结构：** 两层全连接网络，中间用 ReLU 激活函数。
$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$
- **残差连接与层归一化：** 同自注意力层，输出经残差连接和层归一化后传入下一层编码器。

## 解码器 (Decoder)

解码器同样由 **N 个相同的层**堆叠而成（论文中 N=6），每层包含三个子层：

- **多头自注意力层 (Masked Multi-Head Self-Attention)**
- **多头交叉注意力层 (Multi-Head Cross-Attention)**
- **前馈神经网络层 (Feed-Forward Neural Network)**

## 2. 关键模块解析

### (1) Masked 自注意力层

- **作用：** 在生成目标序列时（如翻译），防止解码器提前看到未来的标签（自回归特性），确保预测仅依赖已生成的 tokens。

- **实现方式**：通过掩码（Mask）屏蔽当前位置之后的 tokens，使注意力计算时这些位置的分数为负无穷（Softmax 后趋近于 0）。

## (2) 多头交叉注意力层

- **作用**：让解码器利用编码器的输出（即输入序列的隐藏表示），建立输入与输出序列之间的依赖关系。
- **输入**
  - Query 来自解码器前一层的输出；
  - Key 和 Value 来自编码器的最终输出。
- **计算逻辑**：与多头自注意力类似，但 Q、K、V 分别来自不同的输入源。

## (3) 前馈神经网络层与归一化

- 结构和作用与编码器中的前馈层一致，同样采用残差连接和层归一化。

## 位置编码（Positional Encoding）

- **问题**：自注意力机制对序列顺序不敏感，需显式添加位置信息。
- **解决方案**：通过正弦和余弦函数生成位置编码向量，与输入嵌入（Token Embedding）相加后传入编码器 / 解码器。
- $PE_{(pos, 2i)} = \sin\left(pos/10000^{2i/d_{\text{model}}}\right)$   
 $PE_{(pos, 2i+1)} = \cos\left(pos/10000^{2i/d_{\text{model}}}\right)$   
(pos 为位置索引, i 为维度索引,  $d_{\text{model}}$  为嵌入维度)

## 输入与输出处理

### 输入处理（编码器侧）

- **文本场景**：输入序列经词嵌入（Word Embedding）转换为向量，再与位置编码相加。
- **图像 / 语音场景**：需先将数据转换为序列形式（如图像分块、语音特征帧），再进行嵌入和位置编码。

### 输出处理（解码器侧）

- 解码器最后一层输出经线性变换和 Softmax 层，生成目标序列的概率分布（如语言模型的下一词预测）。

## Transformer 的核心优势

- 1. **并行计算能力**：摒弃循环结构，所有位置的自注意力计算可并行进行，大幅提升训练效率（尤其适合长序列）。
- 2. **长距离依赖建模**：自注意力机制直接计算序列中任意两位置的关联，避免 RNN 因梯度消失导致的长序列建模缺陷。
- 3. **多任务泛化性**：通过预训练（如 BERT、GPT）可迁移至各类 NLP、CV 任务，成为现代大模型的基础架构。

## Q7.LN和BN的区别

对比维度	Batch Normalization (BN)	Layer Normalization (LN)
归一化维度	按特征维度 (channel) 归一化，跨 batch 样本	按样本维度归一化，不跨样本 (每个样本独立计算)
参数数量	每个特征维度 (channel) 有一组 $\gamma, \beta$	每个样本有一组 $\gamma, \beta$ (与特征维度无关)
Batch Size 敏感性	高度依赖 batch size，小 batch 时效果差。	不依赖 batch size，适合动态序列或小 batch 场景。
适用场景	图像、CNN (特征维度固定)	NLP (序列长度可变)、RNN/LSTM/Transformer
训练与推理一致性	训练和推理时统计量不同 (需维护全局统计量)	训练和推理过程完全一致 (无需维护全局统计量)

## 边缘计算

### Q1.什么是边缘计算

边缘计算是在靠近物联网设备或用户设备等数据源头的网络边缘侧（如基站、边缘服务器、智能终端等），就近提供实时数据存储、计算和智能处理的分布式计算模式，旨在减少数据传输延迟、降低云端负载并提升系统响应效率。

## Q2. 什么是物联网

物联网 (IoT) 是通过传感器、射频识别 (RFID)、摄像头等设备, 将物理世界中的物体 (如家电、车辆、基础设施等) 连接到互联网, 使其具备数据采集、传输和交互能力, 从而实现物与物、物与人之间智能连接和协同工作的网络系统。

## Q3. 什么是非正交多址NOMA

NOMA (非正交多址接入, Non-Orthogonal Multiple Access) 是一种**频谱高效的通信技术**, 允许多个用户在相同频段、相同时间资源上通过**功率复用或信号叠加的方式共享信道**, 区别于传统正交多址技术 (如TDMA、FDMA需划分独立时隙或频段)。其核心原理是利用接收机的**串行干扰消除 (SIC) 技术**分离叠加信号, 从而在不增加频谱资源的前提下提升系统容量和接入用户数, 尤其适用于高密度设备连接场景如5G/6G通信、物联网、无人机网络

## Q4. 什么是PPO

PPO (Proximal Policy Optimization, 近端策略优化算法) 是一种**基于策略梯度的强化学习算法**, 旨在解决传统策略梯度算法中 “策略更新幅度过大导致训练不稳定” 的问题。其核心思想是通过限制新旧策略之间的差异 (即 “近端约束”), 使策略更新更加平滑、稳定, 从而提升算法在复杂环境中的收敛效率和鲁棒性。

### 1. 策略梯度框架

- 强化学习中, 智能体通过策略函数 $\pi_{\theta}(a|s)$  (输入状态 $s$ , 输出动作 $a$ 的概率) 与环境交互, 目标是最大化累计奖励的期望。
- 策略梯度算法通过计算奖励对策略参数 $\theta$ 的梯度 $\nabla J(\theta)$ , 迭代更新参数以优化策略。

### 2. 近端约束 (关键创新)

- 传统策略梯度算法 (如 REINFORCE) 每次更新策略时可能导致参数变化过大, 引发训练震荡甚至发散。
- PPO 引入 **KL 散度 (Kullback-Leibler divergence)** 作为新旧策略 $\pi_{\theta_{old}}$ 和 $\pi_{\theta}$ 之间的差异度量, 并在目标函数中添加惩罚项, 强制限制策略更新幅度:

$$\text{目标函数} = \mathbb{E}_s [\min (r_t(\theta) \cdot A_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon) \cdot A_t)] - \beta \cdot \text{KL}[\pi_{\theta},$$

其中:

- $r_t(\theta)$ 为新旧策略的动作概率比值, 用于衡量策略变化;



- clip函数限制 $r_t(\theta)$ 的范围（如 $\epsilon = 0.2$ ），避免更新幅度过大；
- $\beta$ 为 KL 散度的惩罚系数，动态调整策略更新的保守程度。

### 3. 优势函数 (Advantage Function)

- 引入优势函数( $A_t$ )评估动作的相对价值（即该动作相较于平均策略的优劣），通过广义优势估计（GAE）降低方差，提升训练效率。

## Q5.什么是Fisher 判别法

Fisher 判别法 (Fisher Discriminant Analysis, FDA)，又称**线性判别分析 (LDA)**，是一种经典的统计模式识别方法，属于**有监督学习算法**。其核心思想是通过**线性投影**将高维数据降维到低维空间（通常为一维或二维），使得不同类别的数据在新空间中尽可能**类内紧凑、类间分离**，从而实现对未知样本的分类。

假设存在两类样本（多类可扩展） $C_1$ 和 $C_2$ ，原始特征空间为d维，目标是找到一个投影方向 $w$ ，将样本从d维投影到一维（直线）上，使得：

1. **类间距离尽可能大**：投影后两类样本的均值差异显著；
2. **类内距离尽可能小**：投影后同一类样本的分布尽可能紧凑。

通过最大化 **类间散布矩阵 (Between-Class Scatter)** 与**类内散布矩阵 (Within-Class Scatter)** 的比值，求解最优投影方向 $w$ ，数学表达式为：

$$w^* = \arg \max_w \frac{|\mu_1 - \mu_2|^2}{w^T(\Sigma_1 + \Sigma_2)w} \text{ 其中:}$$

- $\mu_1, \mu_2$ 为两类样本的均值向量；
- $\Sigma_1, \Sigma_2$ 为两类样本的协方差矩阵。

该比值称为**Fisher 准则函数**，本质上是寻找一个“最具判别性”的投影方向，使降维后的数据最易区分。

## Q6.什么是模糊聚类

模糊聚类 (Fuzzy Clustering) 是一种**无监督学习算法**，用于将数据点划分到多个聚类中，允许每个数据点以不同的隶属度属于多个聚类，而非传统硬聚类（如 K-means）中“非此即彼”的明确归属。其核心思想是通过**模糊数学理论**描述数据点与聚类中心的不确定性关联，适用于数据边界模糊、类别重叠的场景。

## 一、模糊聚类的核心原理

1. **隶属度矩阵** 用一个 $n \times c$ 的矩阵 $U = [u_{ij}]$ 表示数据点与聚类的关联程度，其中：
  - $n$ 为数据点数量， $c$ 为聚类数；

- $u_{ij} \in [0, 1]$ 表示第*i*个数据点属于第*j*个聚类的隶属度，且满足 $\sum_{j=1}^c u_{ij} = 1$ （每个数据点的隶属度之和为 1）。

2. **目标函数优化** 通常采用**模糊 C 均值算法 (FCM, Fuzzy C-Means)**，其目标

3. 函数为： $J = \sum_{i=1}^n \sum_{j=1}^c u_{ij}^m \|x_i - v_j\|^2$  其中：

- $x_i$ 为第*i*个数据点， $v_j$ 为第*j*个聚类中心；
- $m \geq 1$ 为模糊系数（控制模糊程度，( $m = 1$ )时退化为硬聚类）；
- 目标是最小化数据点与聚类中心的加权距离平方和，通过迭代更新U和 $v_j$ 求解。

#### 4. 模糊性的意义

- 允许数据点属于多个类别（如“一个样本 60% 属于聚类 A，40% 属于聚类 B”），更符合现实世界的复杂性（如基因表达数据中的细胞类型重叠、图像像素的模糊边界）。
- 隶属度可作为数据点分类的置信度，辅助后续决策（如异常检测中隶属度均较低点可能为离群值）。