上海交通大學

SHANGHAI JIAO TONG UNIVERSITY

智能信息处理课程论文

基于 LLM 的手写数字识别：从架构、训练方法到多模态

姓　　名：樊　奇

学　　号：124032910047

院　　系：自动化系

2025 年 2 月 8 日

摘 要

随着大语言模型（Large Language Model，LLM）的发展，其在架构、训练方法与多模态信息处理等方面有着丰富的先进经验并实现了显著的技术进步。同时，计算机视觉领域作为应用最广泛的领域，借鉴 LLM 的先进经验对其发展具有重要意义。因此，我们将在本报告中探索如何将 LLM 的先进经验与技术迁移到计算机视觉领域，并在手写数字识别任务上进行验证。具体地，我们参考了 LLM 对于变换器模型的改进，实现了更加高效的视觉变换器（Vision Transformer，ViT）。其次。受到 LLM 的直接偏好优化方法（Direct Preference Optimization，DPO）的启发，我们实现了新的训练方法，即分类偏好优化方法（Classification Preference Optimization，CPO）。此外，我们还基于开源 LLM 探索并构建了专注于手写数字识别任务的多模态大模型 MnistVL。最后，一系列实验结果说明了 ViT 的高效性以及 CPO 的有效性。相关代码已开源，链接为 https://github.com/fanqiNO1/AU7017。

关键词：手写数字识别，大语言模型，视觉变换器，多模态

目 录

第一章 绪论

1.1 引言

自从 ChatGPT[1] 的面世，大语言模型（Large Language Models，LLM）成为深度学习领域的一颗新星。LLM 的不断发展与研究不仅促使了众多的开源模型的涌现，如 LLaMA[2-3]、深度求索（DeepSeek）[4]、通义千问（Qwen）[5]、书生浦语（InternLM）[6]等；还在自然语言处理（Natural Language Processing，NLP）领域对变换器模型（Transformers）[7]的架构、训练方法、以及多模态信息处理方面做出了改进与创新。

相比于 NLP 领域，计算机视觉是起步较早且应用最广泛的研究领域，见证了从传统算法到深度学习方法的转变。近年来，部分工作已经表明，借鉴 NLP 领域的经验在一定程度上促进了计算机视觉领域的发展，比如视觉变换器（Vision Transformer，ViT）[8]、受到 Bert[9] 模型启发而来的 BEiT[10] 模型等。这种跨领域的知识迁移为解决计算机视觉问题提供了新的视角，并证明了不同领域间技术共享的重要性。

基于此，将 LLM 的先进经验迁移到计算机视觉领域显得尤为重要。手写数字识别任务作为计算机视觉领域最基本的任务之一，其目标是让模型能够自动识别并分类手写的数字字符。手写数字识别任务不仅仅是验证新算法正确性与高效性的理想测试环境，也是探索如何将 LLM 的先进技术整合进计算机视觉模型的起点。

1.2 主要内容

在本报告中，我们以 LLM 的架构、训练方法和对于多模态信息的处理方法为起点，探索了如何将这些先进技术应用到计算机视觉领域，并在手写数字识别任务上进行了实验。具体地，本报告的主要贡献如下：

- 我们参考 LLM 对于 Transformers 结构的改进，实现了更加高效的 ViT 模型。
- 受到 LLM 的训练方法中的直接偏好优化（Direct Preference Optimization，DPO）的启发，我们实现了分类偏好优化（Classification Preference Optimization，CPO）。
- 我们还探索并构建了专注于手写数字识别任务的多模态大模型 MnistVL。

本报告的其余部分安排如下：第 二章介绍了手写数字识别数据集，并回顾了 LLM 在架构、偏好优化以及多模态信息处理方面的相关工作；第 三章介绍了更高效的 ViT 的各模块实现细节；第 四章展开了 CPO 的原理与实现；第 五章介绍了多模态大模型 MnistVL 的具体内容；第 六章描述了实验设置与实验结果；第 七章总结了本报告。

第二章　相关工作

2.1　手写数字识别

在手写数字识别任务中，MNIST[11]（Modified National Institute of Standards and Technology）数据集是最广泛应用的数据集。该数据集包含 60000 个训练样本与 10000 个测试样本，每个样本使用了灰度图代表了 0 到 9 之间的一个数字。该数据集是更大的 NIST 特殊数据库 3（由美国人口普查局的员工书写的数字）和特殊数据库 1（由高中生书写的数字）的子集，这样多样化的来源确保了该数据集能够涵盖不同的书写风格和笔迹特征。该数据集的部分样本可视化如图 2–1所示。

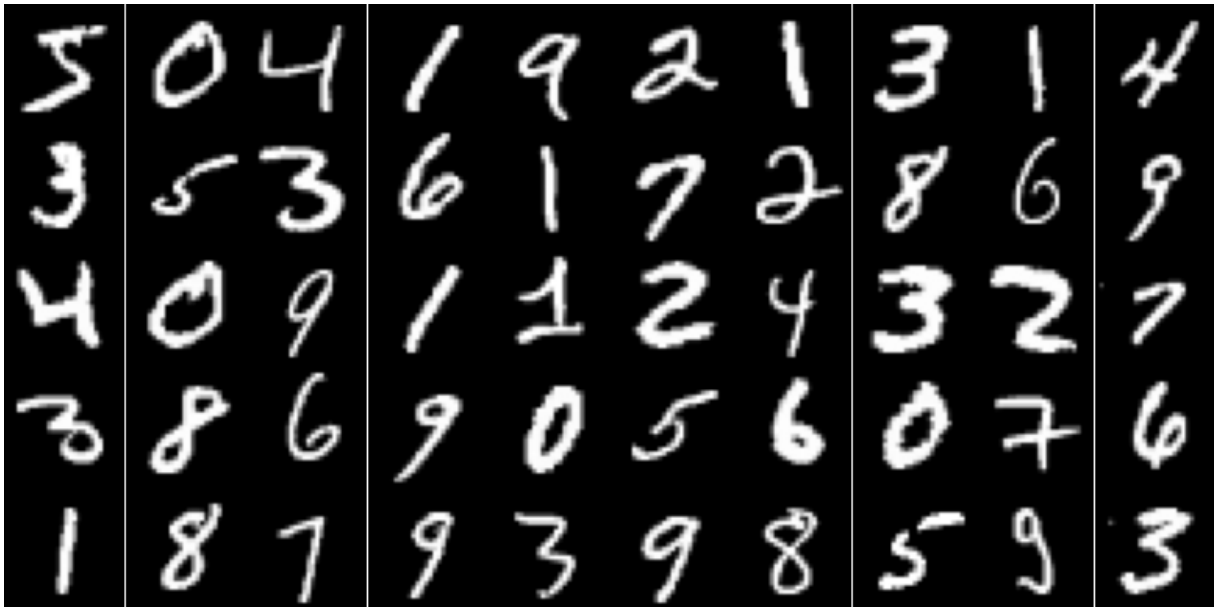

图 2–1 MNIST 数据集部分样本可视化结果

2.2　LLM 架构

由于 LLM 的架构是基于 Transformer 架构进行改进得到，故先介绍 Transformer 架构的基本组成部分：

- 词嵌入（Embedding）：将离散的词元（token）转换为连续的嵌入向量。
- 多层感知机（Multi-Layer Perception）：模型参数量的主要部分。
- 归一化（Normalization）：将向量进行归一化。

- 注意力机制（Attention Mechanism）：进行序列建模。
- 位置编码（Position Embedding）：对序列中各词元的位置进行编码。

随着 LLaMA[2] 第一代的放出，LLM 的架构基本确定。其中，相比于传统的 Transformer 模型架构，LLM 的架构在多层感知机部分、归一化部分、注意力机制与位置编码做出了改进。

在多层感知机方面，LLM 采用的主要分为两种，分别为稠密和稀疏。对于稠密的多层感知机，LLM 引入了门控层并使用 SwiGLU[12] 激活函数，以控制上投影层（Up projection）的输出尺度，具体数学表达将在第 三章展开；对于稀疏的多层感知机，LLM 引入了专家混合（Mixture of Expert，MoE），即使用一个门控单元控制多个专家的激活情况。

在归一化部分，LLM 通常使用均方根层归一化（Root Mean Square Layer Normalization，RMSNorm）[13]，其数学原理将会在第 三章展开。

在注意力机制方面，LLM 做出的改进较多，比如：分组查询注意力（Group Query Attention，GQA）[14]改进了多查询注意力（Multi Query Attention，MQA）[15]，从而实现了计算开销与性能损失之间的平衡；DeepSeek 所使用的多头隐式注意力（Multi Latent Attention，MLA）[16]进一步降低了计算与存储开销；闪电注意力（Lightning Attention）[17]更是将注意力机制计算复杂度由 $O(n^2)$ 降低至 $O(n)$，实现了线性复杂度。

在位置编码方面，LLM 普遍采用旋转位置编码（Rotary Position Embedding，RoPE），其最早应用于 RoFormer[18]。其数学原理将会在第 三章展开。

2.3 LLM 偏好优化

LLM 偏好优化是一种使 LLM 的输出更符合人类价值观的一种训练方法，主要可以分为两类，分别为基于强化学习的方法和直接优化的方法。

在基于强化学习的偏好优化算法中，最具代表性的是人类反馈的强化学习（Reinforcement Learning with Human Feedback，RLHF）与人工智能反馈的强化学习（Reinforcement Learning with AI Feedback，RLAIF）。这类方法依赖于预先训练得到的奖励模型，以对 LLM 的输出进行奖励建模，从而指导 LLM 的训练过程，以最大化预期奖励的同时保持与参考策略的分布相似性在一定范围内。

为了解决 RLHF 中的奖励模型问题，直接偏好优化（Direct Preference Optimization）方法被提出。其将模型本身作为隐式的奖励函数，避免了奖励模型的额外训练。DPO 使用了基于策略的方法，从而直接对 LLM 本身进行优化，简化了训练流程并减少了计算

负担。

此外，一系列基于 DPO 的直接优化方法被提出。如身份偏好优化（Identity Preference Optimization，IPO）为 DPO 引入了正则项以解决过拟合问题；Kahneman-Tversky 优化（Kahneman-Tversky Optimization，KTO）仅从输出是否可取的二进制信号中对 LLM 进行优化；单片偏好优化（Monolithic Preference Optimization）移除了 DPO 对于参考模型的依赖。

2.4 多模态大模型

随着 LLM 的快速发展，以视觉-语言模型（Vision-Language Model。VLM）为代表的多模态大模型也得到了充分发展。尽管早期工作如 CLIP（Contrastive Language-Image Pre-Training）[19]并未使用到 LLM，缺少深层的语义推理能力，但其提出了视觉-语言对比学习，为 VLM 的发展奠定了理论基础。现阶段 VLM 的结构主要可以分为三类，基于投影模块（projector）进行模态对齐的结构、基于交叉注意力（Cross Attention）进行模态对齐的结构、基于 Q-Former 进行模态对齐的结构。

对于基于投影模块进行模态对齐的 VLM，它们往往只使用一至两个线性层和非线性激活函数来实现视觉模态信息到文本模态信息的转换。具体地，在得到视觉编码器对视觉输入的处理结果后，projector 会将视觉结果进行投影，从而得到文本模态信息。这既是为了实现嵌入维度的转换，也是为了实现模态信息的对齐。这部分投影得到的文本模态信息会与经过词嵌入后的文本信息一起作为 LLM 的输入，从而进行推理过程。这类模型主要包括：LLaVA[20] 及其后续版本、Qwen2VL[21] 及其后续版本和 InternVL1.1[22] 及其后续版本等。

对于基于交叉注意力进行模态对齐的 VLM，它们虽然也包括 projector 模块，但其仅起到维度转换的作用。在视觉编码器处理视觉输入后，projector 将其转换到目标维度，后作为 LLM 中注意力机制模块的部分输入（Key 和 Value）参与到 LLM 的推理过程中。这类模型较少，主要包括 Flamingo[23]、OpenFlamingo[24] 及其后续版本和 LLaMA3.2-Vision[25] 等。

对于使用 Q-Former 进行模态对齐的 VLM，它们使用了 Q-Former 模块，其主要包括一系列可学习的查询（Query）。视觉处理结果会在 Q-Former 中进行交叉注意力的计算，从而得到文本模态信息，后参与 LLM 的推理过程。尽管同样使用了交叉注意力，但因交叉注意力的作用位置不同，其模态对齐策略自成一派。这类模型主要包括 BLIP2[26]、QwenVL[21] 和 InternVL1.0[27]。

第三章　视觉变换器

基于 LLM 对于 Transformer 架构的改进，我们实现了更加高效的 ViT，其结构如图 3–1所示，结构细节将在本章详细展开。

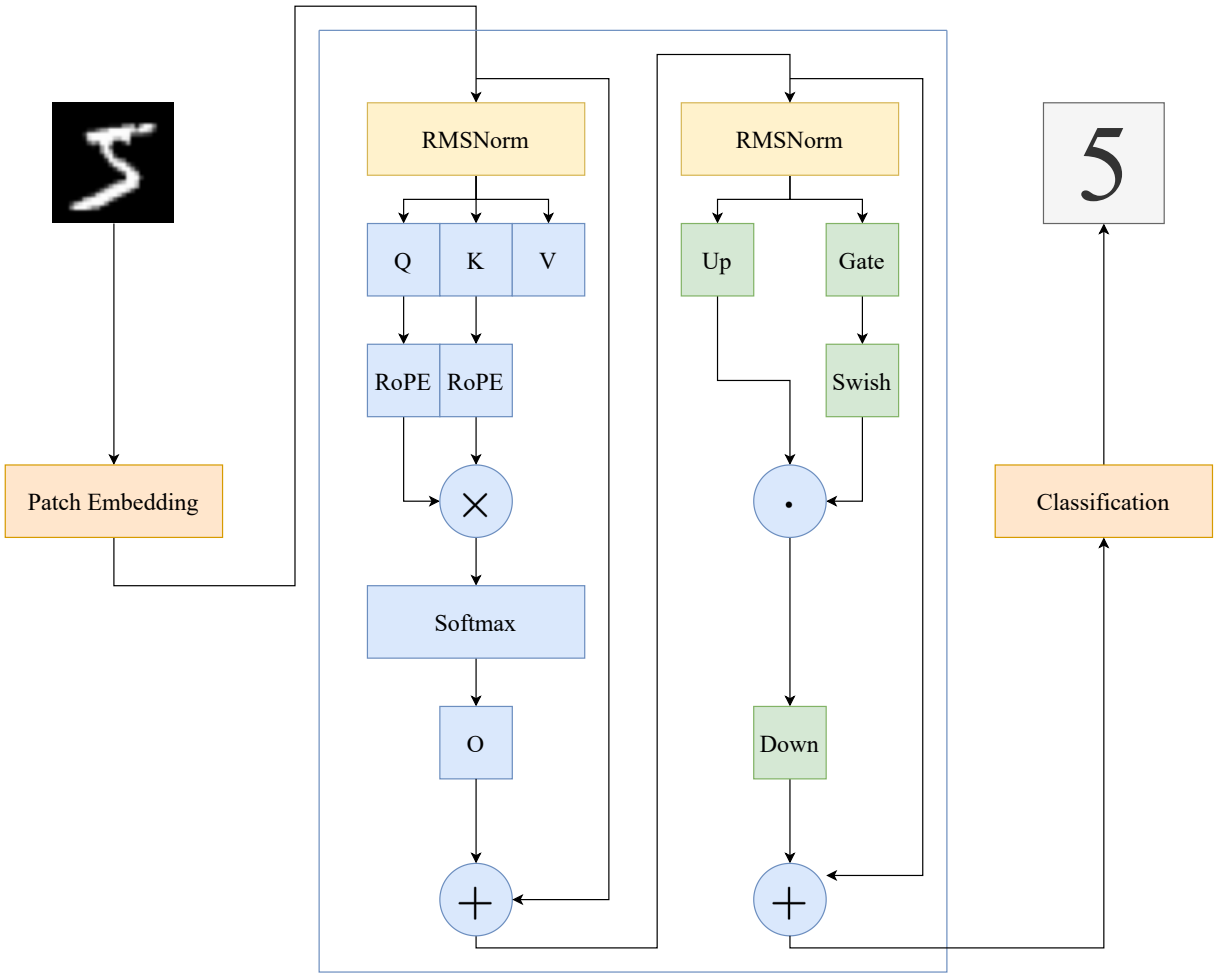

图 3–1　ViT 结构图

3.1　多层感知机

在多层感知机方面，我们采用了与 LLM 一致的稀疏多层感知机，其主要包括三个无偏置的线性层和一个非线性激活函数。其中，三个无偏置线性层可记为 W_{up}、W_{gate} 和 W_{down}，非线性激活函数选用了 SwiGLU 激活函数，其表达式如式 3–1所示。

$$\begin{aligned} \mathrm{SwiGLU}(x, W, V, b, c\beta) &= \mathrm{Swish}_\beta(xW + b) \otimes (xV + c) \\ \mathrm{Swish}_\beta(x) &= x \cdot \mathrm{sigmoid}(\beta x) = \frac{x}{1 + e^{-\beta x}} \end{aligned} \tag{3–1}$$

具体地，记输入维度为 d_{in}，隐藏层维度为 d_{hidden}，则有 $W_{\text{up}}, W_{\text{gate}} \in \mathbb{R}^{d_{\text{in}} \times d_{\text{hidden}}}, W_{\text{down}} \in \mathbb{R}^{d_{\text{hidden}} \times d_{\text{in}}}$。设输入矩阵 $x \in \mathbb{R}^{n \times d_{\text{in}}}$，其中 n 为序列长度，则多层感知机部分的整体计算公式如式 3–2所示：

$$\mathrm{MLP}(x) = (\mathrm{SwiGLU}(xW_{\text{gate}}) \otimes xW_{\text{in}})W_{\text{down}} \tag{3–2}$$

3.2 归一化层

在归一化层方面，我们直接采用了均方根层归一化（Root Mean Square Layer Normalization，RMSNorm）。相比于普通的层归一化，均方根层归一化移除了对于输入样本的均值的依赖。设输入矩阵为 $x \in \mathbb{R}^{n \times d} = (x_1, x_2, \ldots, x_n)$，则其计算公式如式 3–3所示：

$$\begin{aligned} \mathrm{RMSNorm}(x) &= (\mathrm{RMSNorm}(x_1), \mathrm{RMSNorm}(x_2), \ldots, \mathrm{RMSNorm}(x_n)) \\ \mathrm{RMSNorm}(x_i) &= \frac{x_i}{\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2 + \epsilon}} g_i \end{aligned} \tag{3–3}$$

其中，ϵ 是一个很小的数，防止分母为 0；g_i 是均方根归一化层的参数。

3.3 注意力机制

注意力机制方面，我们采用了分组注意力查询（Group Query Attention）。具体地，在多头注意力机制的查询、键和值（Query, Key and Value）中，query 占有 n_q 个头，而 key 和 value 均只有 n_{kv} 个头，其中 n_q 是 n_{kv} 的整数倍。这样一来，n_q 个头被分为 n_{kv} 个组，组内共享同一份 key 和 value，组间使用不同的 key 和 value。其伪代码如代码 3–1所示。

相比于传统的多头注意力，key 和 value 的复用有效降低了计算开销。而相比于所有头共用一份 key 和 value 的多查询注意力（Multi Query Attention），分组操作缓解了模型性能的下降幅度。这样一来，GQA 平衡了计算开销与性能损失。

算法 3–1 GQA 算法伪代码

Data: 输入矩阵 $x \in \mathbb{R}^{n\times d}$

Data: 输出矩阵 $x_a \in \mathbb{R}^{n\times d}$

$q = xW_q + b_q; k = xW_k + b_k; v = xW_v + b_v;$

将 q, k, v 变成 $(n_{\text{head}}, n, d_{\text{head}})$ 的形状；

对 q, k 应用旋转位置嵌入；

对 k, v 根据 $\frac{n_q}{n_{kv}}$ 进行复制；

计算注意力分数 $a = \text{softmax}\left(\frac{qk^\top}{\sqrt{d_{\text{head}}}}\right)\text{v}$；

将注意力分数变成 (n, d) 形状；

经过输出层 $x_a = aW_o + b_o$

3.4　旋转位置嵌入

位置嵌入方面，我们采用了旋转位置嵌入（Rotary Position Embedding，RoPE）。

旋转位置嵌入通过将向量旋转某个角度，以为其赋予位置信息。具体地，其计算公式如式 3–4所示：

$$\begin{pmatrix} \cos m\theta_0 & -\sin m\theta_0 & 0 & 0 & \cdots & 0 & 0 \\ \sin m\theta_0 & \cos m\theta_0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cos m\theta_1 & -\sin m\theta_1 & \cdots & 0 & 0 \\ 0 & 0 & \sin m\theta_1 & \cos m\theta_1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \cos m\theta_{d/2-1} & -\sin m\theta_{d/2-1} \\ 0 & 0 & 0 & 0 & \cdots & \sin m\theta_{d/2-1} & \cos m\theta_{d/2-1} \end{pmatrix} \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \\ \vdots \\ q_{d-2} \\ q_{d-1} \end{pmatrix} \tag{3–4}$$

也可以写作 $R_m q$，其中 R_m 是旋转矩阵。为了实现远程衰减性，即距离越远的两个元素注意力分数越低，RoPE 选用 $\theta_i = 10000^{-2i/d}$。

第四章　分类偏好优化

受到直接偏好优化的启发，我们实现了新的训练方法，分类偏好优化。本章详细介绍了分类偏好优化的原理，并附上了伪代码实现。

4.1　原理

首先先来介绍 DPO 的基本原理，其损失函数如式 4–1所示：

$$\mathcal{L}_{\text{DPO}}(\pi_\theta;\pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y_w\mid x)}{\pi_{\text{ref}}(y_w\mid x)} - \beta\log\frac{\pi_\theta(y_l\mid x)}{\pi_{\text{ref}}(y_l\mid x)}\right)\right]. \tag{4–1}$$

其中，σ 为 sigmoid 函数；β 为超参数，用于控制输出策略（logits）范围；y_w 为偏好数据中好的数据，y_l 为偏好数据中差的数据；$\pi(y_w\mid x)$ 为模型 π 在输入为 x 时输出 y_w 的概率。

从 DPO 的损失函数中可以看到，DPO 的优化目标为增大模型得到好输出的概率，同时降低得到差输出的概率，以最大化两者之间的差距。在这一过程中，y_w 和 y_l 的选择至关重要。

因此，在实现 CPO 时，首先先要确定 y_w 与 y_l。假设数据集 $\mathcal{D}=\{x,y\}$，其中 x 为输入数据，y 为标签；假设模型为 π。那么可以得到模型在输入为 x 时的输出 $\pi(x)$。由于标签在训练时已知，那么自然有 $y_w=y$。对于所有标签中除 y 外，我们便可动态地选取 $\pi(x)$ 最大的标签为 y_l，即 $y_l=\arg\max\pi(x)'$，其中 $\pi(y_w\mid x)'=-\inf,\pi(y_i\mid x)'=\pi(y_i\mid x),y_i\neq y_w$。同时，$y_l$ 也是模型在分类时最容易混淆的，即将 y_w 错误分类为 y_l。值得注意的是，计算 y_l 的过程应当在不求导的情况下完成，以避免对模型的训练造成干扰。

在确定 y_w 和 y_l 后，便可得到 CPO 的损失函数，如式 4–2所示：

$$\mathcal{L}_{\text{CPO}}(\pi) = -\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi(y\mid x)}{\pi(\arg\max\pi(x)')\mid x)}\right)\right] \tag{4–2}$$

通过公式可以看到，CPO 与 DPO 的损失函数较为相似，但移除了对于参考模型的依赖，并且引入了差样本的动态选择逻辑。

4.2　实现

根据以上分析，我们可以得到 CPO 算法的伪代码，如代码 4–1所示。

算法 4–1 CPO 算法伪代码

Data: 模型 π，输入样本 x，标签 y
Data: 损失函数计算结果 loss

得到模型输出 $o = \pi(x)$;
得到输出策略 $l = \log(\text{softmax}(o))$;
在不求导条件下初始化 $l(x)' = l(x)$;
在不求导条件下将 y 的概率置为负无穷 $l(y \mid x') = -\inf$;
在不求导条件下计算 $y_l = \arg\max l(x)'$;
计算输出策略差值 $l_d = l(y \mid x) - l(y_l \mid x)$;
计算损失 $\text{loss} = -\log \sigma(l_d)$

第五章　多模态大模型

5.1　原理

由于第 二章中已经介绍了多模态大模型的三种主要架构，故此处不再展开。我们参考基于投影模块进行模态对齐的 VLM 实现了专注于手写数字识别任务的多模态大模型 MnistVL。LLM 部分我们选用了开源的 `Qwen2.5-1.5B-Instruct`。

由于手写数字识别任务相对简单，且图片分辨率较低，我们并没有使用视觉编码器来对图片做进一步编码，而是将图片展开到一维后直接作为 Projector 层的输入，既进行维度转换，同时进行视觉信息的转换。其结构如图 5–1所示。

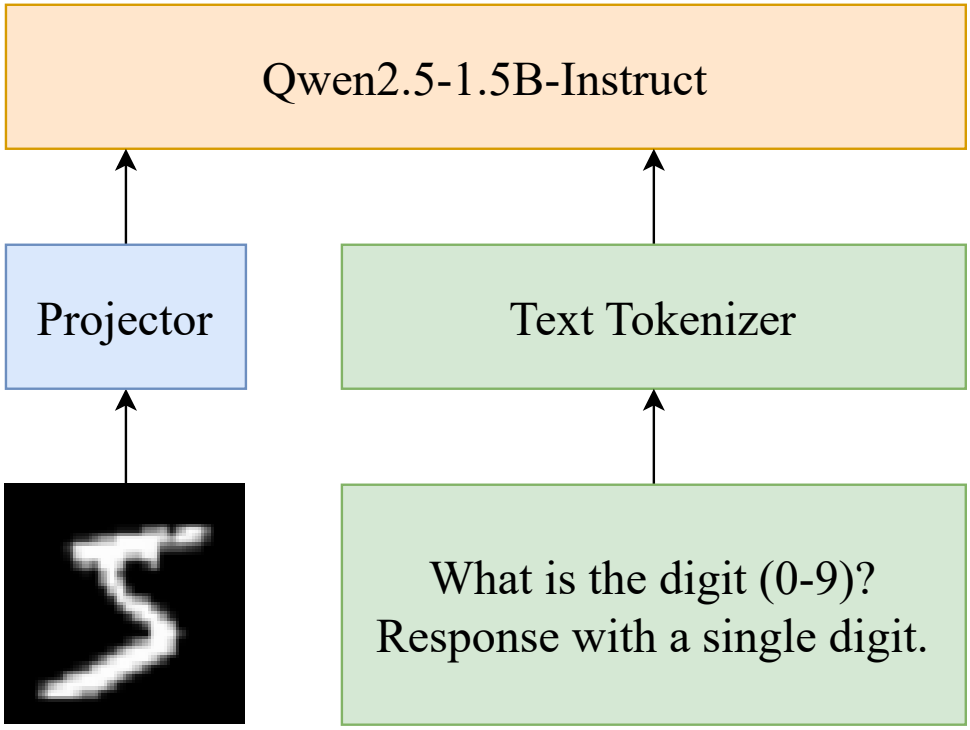

图 5–1 MnistVL 结构图

对于训练策略部分，我们采用了参数高效微调（Parameter Efficient Fine-Tuning）方法中的量化低秩适应（Quantized Low Rank Adaptation，QLoRA）[28]方法，即只训练 projector 和 LLM 的部分参数。此处不对 QLoRA 方法进行展开。

第六章 实验

6.1 实验设置

由于计算资源与时间限制等问题，本报告实验并未采用 MindSpore 以及华为🅴腾计算资源。

我们采用了 PyTorch 2.1.2[29]，Cuda 12.1 来实现我们的 ViT 与 CPO。此外，为了快速迭代模型与算法，我们并没有花费过多时间在实现训练过程上。我们使用了 MMEngine 0.10.6[30] 来实现训练过程 ①。

在训练细节方面，ViT 部分与 CPO 所采用的训练细节一致。训练轮数为 10，训练与测试时批处理大小为 32。优化器为带有权重衰减的自适应动量估计（Adaptive moment estimation with Weight decay，AdamW）[31]。学习率设置为 1×10^{-3}，前 2 轮训练中逐步从 1×10^{-6} 线性增大到 1×10^{-3}，并在后面的 8 轮训练中按余弦下降到 0。

6.2 实验结果

6.2.1 ViT 部分

实验结果如表 6–1所示。

表 6–1 ViT 实验结果

层数 num_layers	块大小 patch_size	嵌入维度 dim	隐藏层维度 hidden_dim	查询头数 n_q	键值头数 n_kv	参数量 #parameters	准确率
6	7	64	128	8	2	0.21M	99.12
6	7	64	128	8	8	0.25M	99.03
12	4	128	256	16	4	1.68M	99.35
24	4	512	1024	32	8	53.53M	99.34

从表格中可以看到，GQA 有效降低了参数量（降低约 16%）。此外，该结构的 ViT 可以在 0.21M 可学习参数量的情况下，实现 99.21% 的准确率，这也进一步说明了该 ViT 的高效性。

此外，我们还探索了进一步增大 ViT 的参数量对于准确率的影响。可以看到，参数量在增大到 1.68M 时，准确率有较为明显的提升。而当参数量进一步提升时，准确率有

① 笔者为 MMEngine 主要贡献者之一，故使用此包来实现训练过程。

所下降，这可能是学习率导致的问题。

6.2.2 CPO 部分

实验结果如表 6–2所示。

表 6–2 CPO 实验结果

层数 num_layers	块大小 patch_size	嵌入维度 dim	隐藏层维度 hidden_dim	查询头数 n_q	键值头数 n_kv	参数量 #parameters	准确率
6	7	64	128	8	2	0.21M	99.01
6	7	64	128	8	8	0.25M	98.95
12	4	128	256	16	4	1.68M	99.21
24	4	512	1024	32	8	53.53M	98.95

表中的高准确率首先说明了 CPO 方法的有效性。但是相比于表 6–1中数据，可以看到 CPO 训练方法得到的准确率略低于传统训练方法（交叉熵损失函数）。

这可能是因为，CPO 训练方法虽然最大化了最优标签与最易混淆标签的距离，但是忽略了其他标签产生的影响。而交叉熵损失函数仅最大化最优标签的概率，也就是最小化其他所有标签的概率，所以交叉熵损失函数训练出的模型的准确率略胜一筹。

6.2.3 MnistVL 部分

为了测试 MnistVL 的有效性，我们将 `Qwen2.5-VL-3B-Instruct` 作为基线模型。由于时间关系，MnistVL 的实验结果暂时缺少数据。

值得注意的是，我们采用的输入为：

```
What is the digit (0-9)? Response with a single digit.
```

表 6–3 多模态大模型实验结果

模型	参数量	准确率
Qwen2.5-VL-3B-Instruct	3B	83.18
MnistVL	1.5B	

可以看到，`Qwen2.5-VL-3B-Instruct` 的准确率并不如传统分类模型。此外，我们还分析了 `Qwen2.5-VL-3B-Instruct` 的混淆矩阵，如图 6–1所示：

从图中可以看到，模型将图片预测为 1 的次数较多，导致了真实标签为 0 或 9 的分类效果不佳，同样导致了模型在真实标签为 1 的测试集上表现良好。

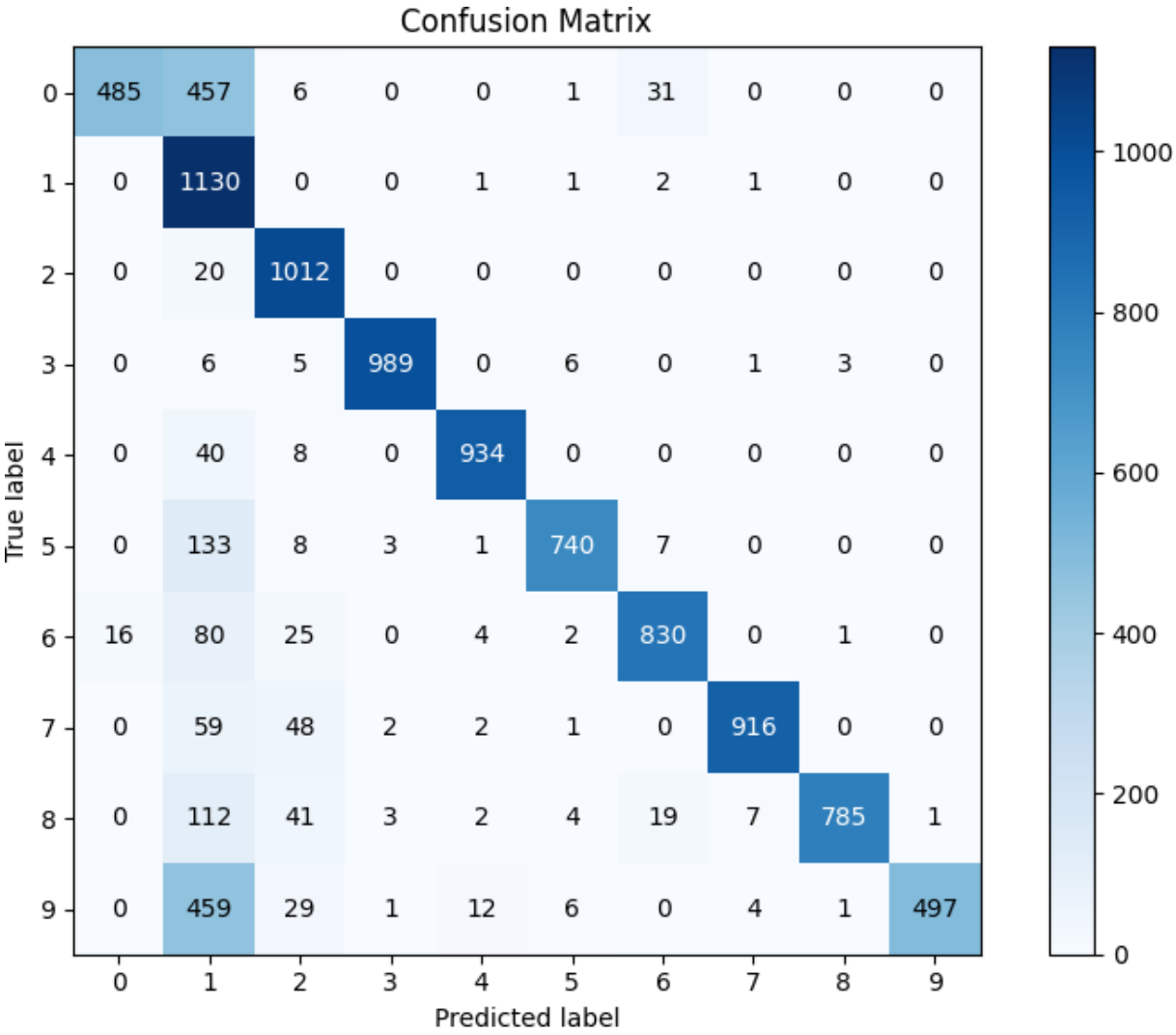

图 6–1 Qwen2.5-VL-3B-Instruct 混淆矩阵

第七章　全文总结

7.1　主要结论

本报告基于对大语言模型（LLM）的深入研究，探索了如何将这些模型中的先进经验迁移到计算机视觉领域，特别是针对手写数字识别这一基础任务。以下是本报告得出的主要结论：

- 改进的视觉变换器（ViT）：通过借鉴 LLM 对于 Transformer 结构的优化方法，我们成功实现了更加高效的视觉变换器（ViT）。实验结果表明，这种改进不仅保持着模型在手写数字识别任务上较高的性能，同时也减少了计算资源的需求，证明了跨领域知识迁移的有效性。
- 分类偏好优化（CPO）：受 LLM 训练方法中直接偏好优化（DPO）的启发，我们提出了分类偏好优化（CPO），并应用于手写数字识别任务。尽管 CPO 所得到的模型准确率低于传统训练方法，但其有效性表明了从 LLM 中提炼出的训练策略同样适用于计算机视觉模型的训练过程。
- 多模态大模型 MnistVL：我们构建了一个专注于手写数字识别任务的多模态大模型 MnistVL。虽然由于时间关系，MnistVL 暂无实验结果，但其仍然可以作为入门多模态大模型的重要一步。

综上所述，本研究表明，将 LLM 的技术进步应用于计算机视觉领域是可行且有效的，不仅有助于解决特定任务（如手写数字识别）中的挑战，还可能为其他计算机视觉任务带来新的解决方案和技术突破。

7.2　未来展望

尽管本报告已经应用了一部分 LLM 的先进技术到计算机视觉任务中，但仍旧存在着一些不足。因此，本报告所涉及的研究的未来展望包括以下几个方向：

- 更多的视觉任务：将 LLM 的先进技术应用到目标检测任务中，实现更高效且支持任意分辨率的目标检测模型。
- 探索混合专家的有效性：使用稀疏的多层感知机，进一步改善 ViT 的计算资源需求。
- 进一步优化 CPO：完善 CPO，使其得到的模型可以优于传统训练方法。
- 探索国产算力：使用 MindSpore 完成代码，并使用华为□腾算力训练。

参考文献

[1] ACHIAM J, ADLER S, AGARWAL S, et al. Gpt-4 technical report[J]. arXiv preprint arXiv:2303.08774, 2023.

[2] TOUVRON H, LAVRIL T, IZACARD G, et al. Llama: Open and efficient foundation language models[J]. arXiv preprint arXiv:2302.13971, 2023.

[3] DUBEY A, JAUHRI A, PANDEY A, et al. The llama 3 herd of models[J]. arXiv preprint arXiv:2407.21783, 2024.

[4] LIU A, FENG B, XUE B, et al. Deepseek-v3 technical report[J]. arXiv preprint arXiv:2412.19437, 2024.

[5] YANG A, YANG B, ZHANG B, et al. Qwen2. 5 technical report[J]. arXiv preprint arXiv:2412.15115, 2024.

[6] CAI Z, CAO M, CHEN H, et al. Internlm2 technical report[J]. arXiv preprint arXiv:2403.17297, 2024.

[7] VASWANI A. Attention is all you need[J]. Advances in Neural Information Processing Systems, 2017.

[8] DOSOVITSKIY A. An image is worth 16x16 words: Transformers for image recognition at scale[J]. arXiv preprint arXiv:2010.11929, 2020.

[9] DEVLIN J. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.

[10] BAO H, DONG L, PIAO S, et al. Beit: Bert pre-training of image transformers[J]. arXiv preprint arXiv:2106.08254, 2021.

[11] LECUN Y, BOTTOU L, BENGIO Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[12] SHAZEER N. Glu variants improve transformer[J]. arXiv preprint arXiv:2002.05202, 2020.

[13] ZHANG B, SENNRICH R. Root mean square layer normalization[J]. Advances in Neural Information Processing Systems, 2019, 32.

[14] AINSLIE J, LEE-THORP J, de JONG M, et al. Gqa: Training generalized multi-query transformer models from multi-head checkpoints[J]. arXiv preprint arXiv:2305.13245, 2023.

[15] SHAZEER N. Fast transformer decoding: One write-head is all you need[J]. arXiv preprint arXiv:1911.02150, 2019.

[16] LIU A, FENG B, WANG B, et al. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model[J]. arXiv preprint arXiv:2405.04434, 2024.

[17] QIN Z, SUN W, LI D, et al. Lightning attention-2: A free lunch for handling unlimited sequence lengths in large language models[J]. arXiv preprint arXiv:2401.04658, 2024.

[18] SU J, AHMED M, LU Y, et al. Roformer: Enhanced transformer with rotary position embedding[J]. Neurocomputing, 2024, 568: 127063.

[19] RADFORD A, KIM J W, HALLACY C, et al. Learning transferable visual models from natural language supervision[C]//International conference on machine learning. 2021: 8748-8763.

[20] LIU H, LI C, WU Q, et al. Visual instruction tuning[J]. Advances in neural information processing systems, 2024, 36.

[21] WANG P, BAI S, TAN S, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution[J]. arXiv preprint arXiv:2409.12191, 2024.

[22] CHEN Z, WANG W, TIAN H, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites[J]. Science China Information Sciences, 2024, 67(12): 220101.

[23] ALAYRAC J B, DONAHUE J, LUC P, et al. Flamingo: a visual language model for few-shot learning[J]. Advances in neural information processing systems, 2022, 35: 23716-23736.

[24] AWADALLA A, GAO I, GARDNER J, et al. Openflamingo: An open-source framework for training large autoregressive vision-language models[J]. arXiv preprint arXiv:2308.01390, 2023.

[25] META L. Llama 3.2: Revolutionizing edge ai and vision with open, customizable

models, 2024[J]. URL: https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices,

[26] LI J, LI D, SAVARESE S, et al. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models[C]//International conference on machine learning. 2023: 19730-19742.

[27] CHEN Z, WU J, WANG W, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024: 24185-24198.

[28] DETTMERS T, PAGNONI A, HOLTZMAN A, et al. Qlora: Efficient finetuning of quantized llms[J]. Advances in Neural Information Processing Systems, 2024, 36.

[29] PASZKE A, GROSS S, MASSA F, et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library[C]//WALLACH H, LAROCHELLE H, BEYGELZIMER A, et al. Advances in Neural Information Processing Systems 32. Curran Associates, Inc., 2019: 8024-8035.

[30] CONTRIBUTORS M. MMEngine: OpenMMLab Foundational Library for Training Deep Learning Models[J]. 2022.

[31] LOSHCHILOV I, HUTTER F. Fixing Weight Decay Regularization in Adam[J/OL]. CoRR, 2017, abs/1711.05101. arXiv: 1711.05101. http://arxiv.org/abs/1711.05101.

[32] LEE H, PHATALE S, MANSOOR H, et al. Rlaif: Scaling reinforcement learning from human feedback with ai feedback[J]. 2023.

[33] RAFAILOV R, SHARMA A, MITCHELL E, et al. Direct preference optimization: Your language model is secretly a reward model[J]. Advances in Neural Information Processing Systems, 2024, 36.

[34] AZAR M G, GUO Z D, PIOT B, et al. A general theoretical paradigm to understand learning from human preferences[C]//International Conference on Artificial Intelligence and Statistics. 2024: 4447-4455.

[35] ETHAYARAJH K, XU W, MUENNIGHOFF N, et al. Kto: Model alignment as prospect theoretic optimization[J]. arXiv preprint arXiv:2402.01306, 2024.

[36] HONG J, LEE N, THORNE J. Orpo: Monolithic preference optimization without reference model[C]//Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing. 2024: 11170-11189.

[37] BAI J, BAI S, YANG S, et al. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond[J]. arXiv preprint arXiv:2308.12966, 2023, 1(2): 3.