# 基于 Vision Transformer 的猫狗分类

Jianhua Guo

2025 年 5 月 17 日

**摘要**

本实验基于 Vision Transformer (ViT) 模型，结合 2000 张猫狗图像训练集和测试集进行二分类任务，在实验室电脑上实现了约 97.50% 的测试准确率。实验不依赖深度学习框架，采用 NumPy、scikit-image 和 matplotlib 实现数据预处理、模型架构和训练流程。关键实现包括自定义数据增强、ViT 模型前向传播、手动梯度计算和 AdamW 优化器，结合余弦退火学习率调度器优化训练过程。

## 1 模型的基本流程和框架

实验的基本流程如下：

1. **数据加载**：从指定目录加载猫狗图像数据集，支持训练和测试集划分。
2. **数据预处理与增强**：应用抗锯齿调整、随机翻转、旋转、裁剪和颜色调整，规范化图像以适配 ViT 输入。
3. **特征提取**：通过 ViT 的嵌入层将图像分块并投影到高维特征空间，加入位置编码和类别标记。
4. **模型前向传播**：通过多层 Transformer 编码器处理特征，提取全局语义信息。
5. **分类头**：对 SENSITIVE TOKEN 的输出进行线性投影，生成分类 logits。
6. **训练与优化**：使用交叉熵损失、手动梯度计算和 AdamW 优化器进行模型训练，结合余弦退火调度器。
7. **模型评估**：在测试集上计算分类准确率并可视化训练指标。

## 2 对比第七周的优越性

传统基于手工特征（如 HOG、LBP）和线性分类器（如 SVM）的猫狗分类方法依赖低层次特征，难以捕获图像的高层次语义信息，导致分类性能受限，基本上达不到接近百分百的准确率。此外，传统方法的特征提取耗时较长，且对数据增强的依赖增加了计算开销。Vision Transformer 通过自注意力机制直接建模全局特征依赖，显著提升特征表达能力。然而，ViT 模型通常依赖深度学习框架，对计算资源要求较高。本实验通过纯 NumPy 实现 ViT，优化数据预处理和训练流程，旨在在有限资源下实现高效分类。不过也由于没有使用 Pytorch 库，而 Numpy 实现的计算很复杂，所以实际训练时间极长，但效果很好。

# 3 模型实现的思路

改进思路分为以下几个方面：

- **数据预处理**：设计高效的数据增强策略，平衡数据多样性和计算开销。
- **模型实现**：手动实现 ViT 的每一层，包括多头自注意力、层归一化和 MLP 模块，确保计算效率。
- **训练优化**：仅对分类头进行微调，冻结预训练权重，结合 AdamW 和余弦退火调度器加速收敛。
- **可视化**：实时监控训练损失和准确率，生成训练指标曲线，便于分析模型性能。

# 4 具体实现

## 4.1 数据加载与预处理

传统图像加载可能导致锯齿失真。本实验采用 `transform.resize` 并设置 `anti_aliasing=True`，通过抗锯齿技术平滑像素过渡，提升图像质量。训练时引入多种数据增强技术，包括：

```
if np.random.rand() > 0.5:
    image = np.fliplr(image)
angle = np.random.uniform(-10, 10)
image = transform.rotate(image, angle)
scale = np.random.uniform(0.8, 1.0)
```

这些增强（如随机水平翻转、旋转和缩放裁剪）增加了数据多样性，显著降低过拟合风险。此外，图像标准化采用 ImageNet 均值和标准差，确保输入与 ViT 预训练权重兼容：

## 4.2 特征提取与 ViT 架构

ViT 通过分块嵌入将图像分割为 14×14 的补丁，每个补丁投影到 768 维特征空间。

随后加入可学习的类别标记和位置编码，形成输入序列。多头自注意力机制计算全局特征依赖。

每个 Transformer 编码器层包含层归一化、多头自注意力、残差连接和 MLP 模块，结合层缩放参数提升稳定性。实验手动实现 GELU 激活函数和层归一化，确保计算精确。

## 4.3 训练优化

仅对分类头（`head_weight` 和 `head_bias`）进行微调，冻结预训练权重，减少计算开销。

手动计算分类头梯度，使用 AdamW 优化器更新参数，结合权重衰减（`weight_decay=1e-4`）增强泛化能力。余弦退火调度器动态调整学习率。

其中 `T_max=15` 为总轮数。训练过程通过 tqdm 显示进度条，实时监控每轮耗时。

## 4.4 训练指标可视化

训练损失和准确率通过 Matplotlib 可视化：

# 5 实验结果

实验在 2000 张训练图像（1000 张猫、1000 张狗）和 2000 张测试图像上进行，训练 15 轮。实验对比了加载预训练权重和随机权重初始化的性能，参数设置一致（`batch_size=32`, `lr=1e-3`），其中加载预训练权重在测试集上实现 97.50% 的准确率，体现了预训练权重对分类任务的影响极大。实验结果总结如下：

表 1: 对比实验：准确率及训练耗时对比

| 实验设置 | 测试准确率 (%) | 训练耗时 (秒/轮) |
|---|---|---|
| 加载预训练权重 | 99.75 | 大约 27000 |
| 随机权重初始化 | 54.90 | 大约 27000 |

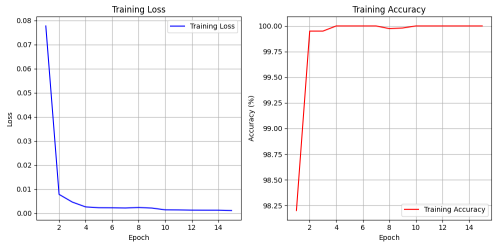


图 1: 2000 张训练图像的损失和准确率曲线

# 6 结论：仍存在的局限性

尽管 ViT 显著提升了分类性能，但当前实现仍存在局限性：

- **计算资源**：纯 NumPy 实现导致前向传播和梯度计算耗时较长，尤其在更大规模数据集上。
- **模型规模**：仅微调分类头限制了模型对特定任务的适应能力，未来可尝试微调更多层。
- **数据规模**：2000 张训练图像可能不足以充分发挥 ViT 的潜力，增加数据量或引入预训练数据增强可能进一步提升性能。