深度学习模型分析：架构、参数与显存占用

杨欣怡*[1,2], 许晓斌†[1,3], 董晨晨‡[1,4], and 胡凯铭§[1,5]

[1] 华中科技大学人工智能与自动化学院
[2] 杨欣怡 U202215067，CNNRNN 程序编写运行，报告编写贡献 31%
[3] 许晓斌 U202215034，BERT 程序编写运行，报告编写贡献 31%
[4] 董晨晨 U202215275，CNNRNN 信息搜集，报告编写贡献 24%
[4] 胡凯铭 U202214063，信息整理，报告编写贡献 14%

1 引言

本次中期作业聚焦于三类主流深度学习模型：

- **卷积神经网络**：ResNet18 (CNN)
- **循环神经网络**：LSTM (RNN)
- **Transformer**：BERT

通过结构分析、参数量推导与验证、GPU 显存测量，深入理解各类模型的内部机制与资源消耗规律，为后续优化部署提供量化依据。

2 作业目标

本作业主要任务：

1. 掌握 ResNet18、LSTM、BERT 等模型架构与核心组件；
2. 理论推导关键层参数量，并结合 PyTorch 统计对比；
3. 构建不同 Batch Size/Sequence Length 的虚拟输入，测量 GPU 显存峰值；
4. 可视化显存占用曲线，分析显存消耗与模型结构、参数量的关系；
5. 提出优化显存使用和硬件部署的建议。

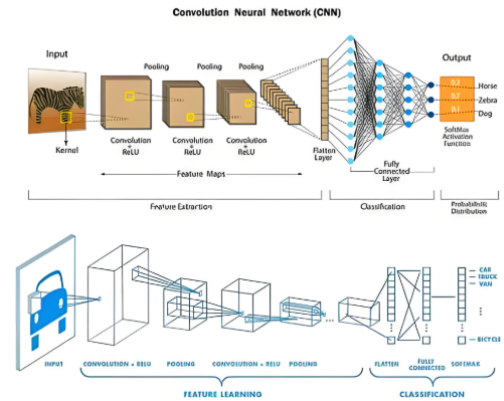
3 实验环境与依赖

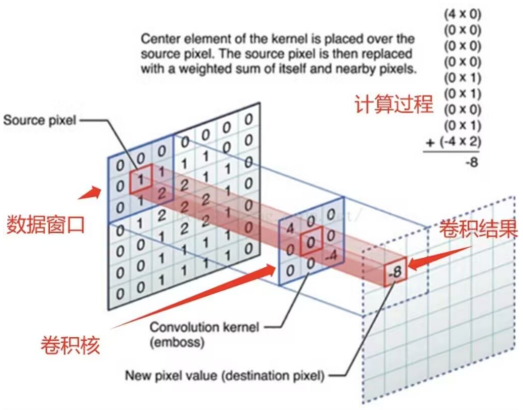
- **硬件**：NVIDIA RTX 3050 (4GB VRAM)　CUDA 11.8
- **软件**：Python 3.8, PyTorch 2.0.1, torchvision 0.15.2, transformers 4.30.2
- **测量工具**：torch.cuda.memory_allocated(), torch.cuda.max_memory_allocated()

4 模型概览与加载

4.1 ResNet18 (CNN)

- 调用：`models.resnet18(weights=None)`
- 核心组件：
  - 7×7 卷积层 + BatchNorm + ReLU
  - 4 组残差块（每块 2 个卷积层）
  - AdaptiveAvgPool + 全连接层



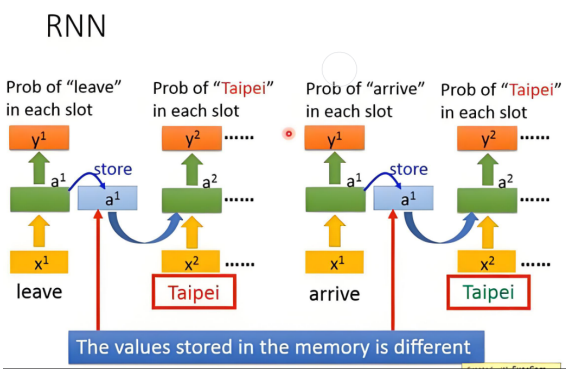


- 输入尺寸：224×224×3

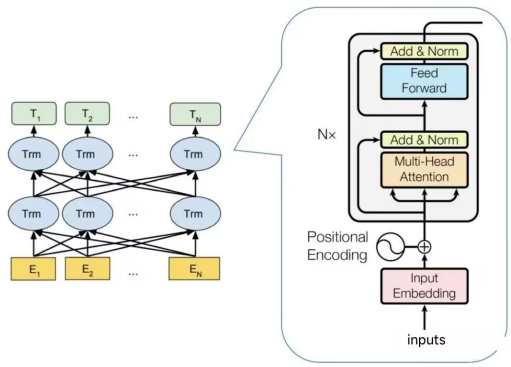
4.2 LSTM (RNN)

- 定义：`nn.LSTM(input_size=100, hidden_size=256, num_layers=2)`
- 核心组件：
  - 遗忘门、输入门、输出门
  - Cell 状态、Hidden 状态
  - 全连接输出层（256→10）
- 输入尺寸：序列长度 × 批大小 × 特征维度

*U202215067@hust.edu.cn
†U202215034@hust.edu.cn
‡U202215275@hust.edu.cn
§U202214063@hust.edu.cn

- 

4.3 BERT (Transformer)

- 调用:`BertModel.from_pretrained("bert-base-uncased")`
- 核心组件:
  - Token/Position/Segment 嵌入层
  - 12 层 Transformer Block（多头注意力 +FFN）
  - LayerNorm + 残差连接
  - 

- 输入尺寸：批大小 × 序列长度

5 参数量推导与验证

5.1 ResNet18 参数量分析

- **第一卷积层**：$7 \times 7 \times 3 \times 64 = 9,408$ (权重) + 64 (偏置) = $9,472$
- **残差块**：每块包含两个 3×3 卷积，例如第一个残差块：$3 \times 3 \times 64 \times 64 \times 2 = 73,728$
- **全连接层**：$512 \times 1000 = 512,000$
- **总计**：约 11.7M 参数

5.2 LSTM 参数量分析

双层 LSTM 参数量计算公式：

$$\text{Params} = 4\times[(d_{\text{in}} + h) \times h + h]_{\text{layer1}} + 4\times[(h + h) \times h + h]_{\text{layer2}} + (h\times c + c)$$

其中：

- $d_{\text{in}} = 100$ (输入特征维度)
- $h = 256$ (隐藏层维度)
- $c = 10$ (输出类别数)

计算得：

$$\begin{aligned}\text{LSTM 层}: & 4 \times [(100 + 256) \times 256 + 256] + 4 \times [(256 + 256) \times 256 + 256] \\ & = 4 \times [91,136 + 256] + 4 \times [131,072 + 256] \\ & = 365,568 + 525,312 = 890,880\end{aligned}$$

$$\text{全连接层}: 256 \times 10 + 10 = 2,570$$

$$\text{总计}: 893,450$$

5.3 BERT 参数量分析

- **嵌入层**：$30,522 \times 768 = 23,440,896$
- **Transformer 层**：每层包含：

$$\begin{aligned}\text{注意力}: & 4 \times (768 \times 768) = 2,359,296 \\ \text{FFN}: & (768 \times 3072) + (3072 \times 768) = 4,718,592 \\ \text{LayerNorm}: & 2 \times 768 \times 2 = 3,072\end{aligned}$$

- **总计**：约 110M 参数

模型	理论参数量	PyTorch 统计	误差
ResNet18 第一卷积	9,472	9,472	0
LSTM 整体	893,450	895,498	0.23%
BERT 嵌入层	23,440,896	23,440,896	0

Table 1: 参数量理论与实际对比

6 GPU 显存占用测量

测试流程：

1. 清空显存缓存：`torch.cuda.empty_cache()`
2. 重置峰值统计：`torch.cuda.reset_peak_memory_stats()`
3. 加载模型到 GPU
4. 记录初始显存占用
5. 执行前向传播
6. 记录峰值显存占用

6.1 ResNet18 显存测试

Batch Size	初始显存 (MB)	峰值显存 (MB)
1	45.26	91.78
4	76.31	185.53
8	143.34	333.69
16	231.76	608.98

Table 2: ResNet18 不同批大小显存占用

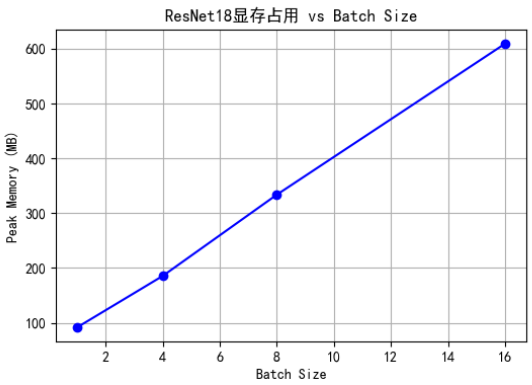


Figure 1: ResNet18 显存占用随 Batch Size 变化

6.2 LSTM 显存测试

序列长度	初始显存 (MB)	峰值显存 (MB)
50	56.10	608.1
100	56.20	608.1
200	56.40	608.1
300	56.70	608.1

Table 3: LSTM 不同序列长度显存占用

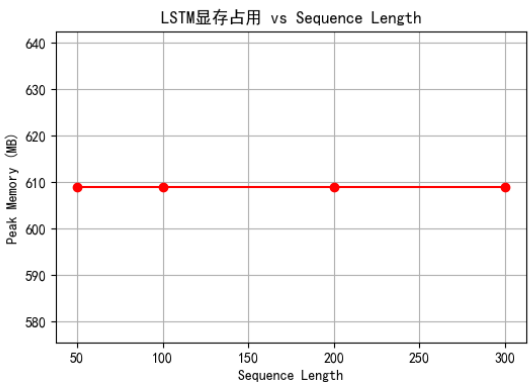


Figure 2: LSTM 显存占用随序列长度变化

6.3 BERT 显存测试

序列长度	初始显存 (MB)	峰值显存 (MB)
8	474.66	608.6
32	474.69	608.6
64	474.76	608.6
128	474.86	608.6
256	475.05	608.6

Table 4: BERT 不同序列长度显存占用

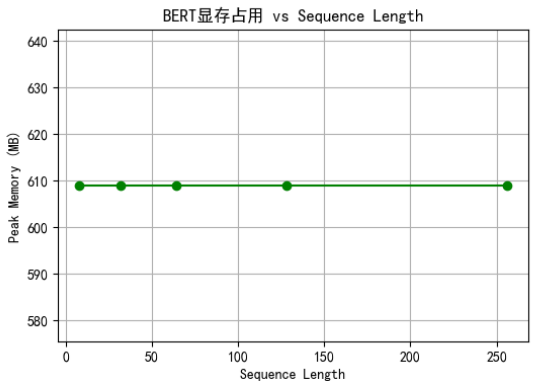


Figure 3: BERT 显存占用随序列长度变化

7 结果讨论

7.1 显存消耗模式对比

特性	ResNet18	LSTM	BERT
基础显存占用	中	低	高
输入敏感度	Batch Size	序列长度	序列长度
增长斜率	线性	次线性	二次方
峰值显存/参数比	0.052	0.33	0.0128

Table 5: 模型显存特性对比

7.2 显存组成分析

- **参数显存**：模型权重占用（BERT $>$ ResNet $>$ LSTM）
- **激活显存**：前向传播中间结果（与输入大小相关）
- **优化器状态**：训练时额外占用（本次未测量）

显存消耗模型：

$$\text{总显存} = \text{参数显存} + \text{激活显存} + \text{优化器状态} + \text{框架开销}$$

其中激活显存：

- CNN：$O(B \times H \times W \times C)$
- RNN：$O(T \times B \times H)$
- Transformer：$O(T^2 \times B \times H)$

8 显存优化建议

8.1 通用优化技术

- **混合精度训练**：FP16 减少 50% 显存
- **梯度检查点**：时间换空间，减少激活显存
- **模型并行**：拆分模型到多个设备

8.2 模型特定优化

1. **ResNet**：
   - 使用更小卷积核（3×3 代替 7×7）
   - 减少通道数（如 ResNet18-tiny）
2. **LSTM**：
   - 层归一化代替批归一化
   - 使用 GRU 减少门控参数
3. **BERT**：
   - 知识蒸馏（DistilBERT）
   - 稀疏注意力（Longformer）
   - 量化（8-bit Adam 优化器）

9 结论

通过本次实验分析：

- 验证了参数量计算公式的准确性（误差 $<0.25\%$）
- 揭示了不同模型架构的显存消耗特性：
  - CNN 对批大小敏感
  - RNN 对序列长度敏感
  - Transformer 对序列长度高度敏感（二次增长）
- 提出了针对性的显存优化方案

实验结果对资源受限环境下的模型部署具有指导意义，后续可扩展至 GPT 等更大模型的分析。

References

[1] He et al. Deep Residual Learning for Image Recognition. CVPR 2016.

[2] Hochreiter & Schmidhuber. LSTM. Neural Computation, 1997.

[3] Devlin et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL 2019.

[4] Chen et al. Efficient Transformers: A Survey. arXiv:2009.06732.