

实验报告

课程名称: _____

实验类型: _____ 操作实验 _____

实验项目名称: _____ 实验二: 基于 AgileNN 的云边协同推理实验 _____

姓名: _____ 学号: _____

QQ 号码: _____

(高校联合班成员填写: 学校_____姓名_____学号_____)

实验日期: _____年 _____月 _____日

了解云边协同计算的基本原理和架构

一、实验目的和要求:

- 了解 AgileNN 云边协同推理框架的基本原理和架构
- 理解基于可解释 AI (XAI) 的特征重要性评估机制
- 掌握 AgileNN 与传统模型 (MobileNetV2) 的性能对比方法
- 学习云边协同推理中的动态负载分配策略
- 分析不同部署模式下的推理延迟和准确率表现

二、实验内容

基于 AgileNN 框架和 MobileNetV2 深度学习模型, 根据实验代码库 (<https://github.com/csmVlc/edge-cloud-experiment>) 完成以下实验

AgileNN 云边协同推理性能对比实验

- ✓ 配置不同的边缘端 CPU 资源, 例如 (0.3, 0.5, 0.6, 0.7)
- ✓ 通过 tc 工具调节网络带宽, 例如 (5mbit/s, 7mbit/s, 10mbit/s)
- ✓ 对比 AgileNN 与 MobileNetV2 在不同部署模式下的性能
- ✓ 测试 only-edge (纯边缘)、only-cloud (纯云端) 和云边协同三种模式
- ✓ 分析端到端延迟、推理准确率
- ✓ 记录实验内容, 找到在哪一种设备以及网络带宽下, 云边协同的端到端延迟最低

三、实验背景

- AgileNN 框架介绍

AgileNN 是一种新型的神经网络卸载技术, 通过利用可解释 AI (eXplainable AI,

XAI) 技术实现极弱设备上的实时神经网络推理。该框架的核心思想是将神经网络卸载中所需的计算从在线推理迁移到离线学习阶段, 通过在训练过程中显式强制特征稀疏性来最小化在线计算和通信成本。

AgileNN 采用轻量级特征提取器在本地嵌入式设备上提供特征输入: 重要性高的 top-k 特征由本地神经网络保留以进行本地预测, 然后与来自其他不太重要特征的远程神经网络预测相结合, 产生最终推理输出。

- MobileNetV2 模型架构

MobileNetV2 是一个专为移动和嵌入式设备设计的轻量级卷积神经网络架构。它使用深度可分离卷积和倒置残差结构, 能够在保持较高精度的同时显著减少计算复杂度和模型大小。在本实验中, MobileNetV2 作为对比基准, 用于评估 AgileNN 框架的性能优势。

四、 主要仪器设备

- 边缘计算设备
高性能边缘设备香橙派
- 云端服务器
高性能服务器
- 软件环境
代码运行环境: PyTorch
部署环境: Kubernetes, Docker.

五、 实验题目简答

- a) AgileNN 相比传统神经网络分区方法有哪些创新点? 请分析其核心技术优势。
- b) 在云边协同推理中, only-edge、only-cloud 和协同模式各有什么特点? 请分析各自的适用场景。
- c) 可解释 AI (XAI) 技术在 AgileNN 中起到什么作用? 请说明特征重要性

评估的工作原理。

六、实验数据记录和处理

以下实验记录均需结合屏幕截图，进行文字标注和描述（看完请删除本句）。

AgileNN 性能数据

1) 实验环境配置截图

```
- name: edge-test
  image: agilenn-edge:v2.0
  securityContext:
    privileged: true
  resource:
    limits:
      cpu: "0.3"
    requests:
      cpu: "0.3"
  env:
    - name: PYTHONUNBUFFERED
      value: "1"
  args: ["/bin/bash", "-c", "tc qdisc add dev eth0 root tbf rate 5mbit
```

2) 元物云平台终端运行截图

```
edge-test      cloud-test

2025-07-07 11:22:06: 样本 98: 预测=42 | 真实=25 | x | 端到端= 36.34ms | 压缩时间= 0.62ms | 压缩
2025-07-07 11:22:06: 样本 99: 预测=89 | 真实=89 | ✓ | 端到端= 35.82ms | 压缩时间= 0.67ms | 压缩
2025-07-07 11:22:06: 样本 100: 预测=17 | 真实=17 | ✓ | 端到端= 37.06ms | 压缩时间= 0.65ms | 压缩
2025-07-07 11:22:21: AgileNN 测试结果
2025-07-07 11:22:21: 总样本数: 100
2025-07-07 11:22:21: 推理正确率: 0.7200 (72/100)
2025-07-07 11:22:22:
2025-07-07 11:22:22: 推理延迟统计 (ms/sample):
2025-07-07 11:22:22: 端到端时间:      36.51 ± 1.64
2025-07-07 11:22:22: 本地推理:        5.28 ± 0.44
2025-07-07 11:22:22: 压缩时间:        0.65 ± 0.05
2025-07-07 11:22:22: 结果融合:        0.28 ± 0.02
2025-07-07 11:22:22: 压缩前后统计 :
2025-07-07 11:22:22: 压缩前= 778248
2025-07-07 11:22:22: 压缩后=3346.10 ± 338.21B
2025-07-07 11:22:22: 压缩比= 23.5 ± 2.6x
websocket disconnect
>>
```

```
edge-test      cloud-test

2025-07-07 11:22:06: 云端推理:      20.36 | 解压缩:      0.65 | 传输时延:      5.61
2025-07-07 11:22:06: 云端推理:      20.58 | 解压缩:      0.66 | 传输时延:      5.11
2025-07-07 11:22:06: 云端推理:      20.60 | 解压缩:      0.66 | 传输时延:      4.66
2025-07-07 11:22:06: 云端推理:      22.02 | 解压缩:      0.67 | 传输时延:      4.85
2025-07-07 11:22:06: 云端推理:      20.69 | 解压缩:      0.74 | 传输时延:      4.68
2025-07-07 11:22:06: 云端推理:      20.82 | 解压缩:      0.66 | 传输时延:      4.71
2025-07-07 11:22:06: 云端推理:      20.86 | 解压缩:      0.69 | 传输时延:      7.93
2025-07-07 11:22:06: 云端推理:      20.96 | 解压缩:      0.74 | 传输时延:      4.83
2025-07-07 11:22:06: 云端推理:      24.07 | 解压缩:      0.66 | 传输时延:      4.29
2025-07-07 11:22:06: 云端推理:      20.52 | 解压缩:      0.67 | 传输时延:      4.62
2025-07-07 11:22:06: 云端推理:      20.72 | 解压缩:      0.76 | 传输时延:      5.25
2025-07-07 11:22:06: 云端推理:      20.88 | 解压缩:      0.66 | 传输时延:      5.35
2025-07-07 11:22:21: Received END signal, stopping inference...
2025-07-07 11:22:22: 推理延迟统计 (ms/sample):
2025-07-07 11:22:22: 云端推理:      20.99 ± 1.24
2025-07-07 11:22:22: 传输时间:      5.03 ± 0.84
websocket disconnect
>>
```

3) 数据分析记录截图

本地cpu	带宽	准确率	end-end	本地	传输	云端
-------	----	-----	---------	----	----	----

0.6	10mbit/s	≈ 72	34.13ms	6	5	20
0.3	10mbit/s	≈ 72	36.01	6	5	20

MobileNetV2 性能数据

七、实验结果与分析

通过上述实验和相关资料学习，分别解答以下问题（看完请删除本句）：

- 基于该实验，谈谈你对云边协同推理计算的理解
- 分析 AgileNN 云边协同推理框架的技术优势和应用前景。