解决方案实践

天宽昇腾云行业大模型适配服务解决方案实践

文档版本 1.0

发布日期 2024-11-27

华为技术有限公司

版权所有 © 华为技术有限公司 2024。 保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明


和其他华为商标均为华为技术有限公司的商标。
本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

安全声明

漏洞处理流程

华为公司对产品漏洞管理的规定以“漏洞处理流程”为准，该流程的详细内容请参见如下网址：
https://www.huawei.com/cn/psirt/vul-response-process
如企业客户须获取漏洞信息，请参见如下网址：
https://securitybulletin.huawei.com/enterprise/cn/security-advisory

版权所有 © 华为技术有限公司

目 录

版权所有 © 华为技术有限公司

1 方案概述

应用场景

随着全球科技竞争的加剧和国际制裁背景下，中国企业对国产自主算力的需求迅速增长。昇腾行业大模型适配服务凭借其强大的高性能计算能力和深度学习算法优化，成为推动国内信创产业发展的关键力量。而各地国产化算力中心建设完成后，客户常因技术栈差异面临软硬件兼容性和使用困难，缺乏对华为昇腾AI平台的深入了解，遇到技术问题时响应不及时，影响项目推进和创新。

客户在使用昇腾算力开发模型时面临诸多挑战：

- 技术栈差异：各地国产化算力中心建设完成后，客户常因技术栈差异面临软硬件兼容性和使用困难，导致开发效率低下。
- 技术理解不足：部分客户缺乏对华为昇腾AI平台的深入了解，遇到技术问题时响应不及时，影响项目推进和创新。
- 迁移难度大：AI模型迁移面临算子层、框架层、模型层等多技术体系，迁移过程中遇到算子不适配场景难以解决，迁移后模型需要进行准确和性能调优，依赖专家经验进行模型分析与调优。
- 开发环境复杂：AI开发面临算子层、模型层、应用使能层等多技术体系的熟悉，学习难；AI现场开发过程中常会遇到难点问题、新特性理解不深入，问题求助响应慢；模型运行依赖多，开发环境搭建复杂；工具链种类多，学习周期长。
- 专业人才短缺：客户虽然有专业的AI算法工程师团队，但不了解CANN与昇腾底层，在开发过程中遇到底层问题疑难问题难以处理。算法工程师定位底层问题效率低，不了解昇腾有哪些可以利用依赖的工具链，疑难问题求助依赖社区途径。
- 调优经验不足：昇腾迁移调优经验少，CANN层问题不会处理，不了解昇腾的调度逻辑。缺乏大模型调优经验，针对模型性能与精度优化没有有效的方法，没有类似算子优化层面的高阶调优能力。

通过本方案实现的业务效果：

本章节介绍如何通过天宽昇腾云行业大模型适配服务解决方案，提供模型从开发到迁移的全流程支持，优化模型性能，确保业务平稳运行。

- 全栈式技术服务：提供算法框架、计算框架、加速框架、硬件组网以及芯片型号等组合的全栈支持能力，确保模型在不同硬件平台上的高效运行。
- 高效模型迁移适配：通过自动化迁移工具和专业的技术支持，实现模型从GPU平台快速、无缝地迁移到昇腾NPU平台，确保模型在新平台上的性能和精度不受影响；

版权所有 © 华为技术有限公司

- 多维度性能调优：提供从算子、内存、通信、调度等多维度的调优手段，提升模型的运行效率和性能，调优效率提升50%，平均模型性能提升20%以上；
- 专业服务团队支撑：具备经验丰富的现场工程师和远程专家团队，帮助客户快速定位精度问题，解决性能瓶颈，业务上线时间缩短25%。

解决方案实践的应用行业推荐：

通过华为云高性价比国产算力算力，结合天宽昇腾云行业大模型适配服务，为客户提供从模型设计、数据处理到训练、优化、部署的一站式AI模型服务，确保模型准确适配行业需求，快速实现业务落地。特别适合如下行业：

- 政府与公共服务：大量昇腾算力中心建设完成后，客户常因技术栈差异面临软硬件兼容性和使用困难的问题，需要专业技术团队为客户提供昇腾设备的使用支持服务，旨在提升昇腾开发效率、降低昇腾开发门槛，处理客户在开发过程中遇到的技术问题。
- 能源与电力：新能源的快速发展给电网稳定性带来巨大挑战，在各业务场景中迫切需要引入大模型提升管理效率，而通用基础模型往往无法直接使用，天宽深耕电力行业，具备丰富的技术实力和行业经验，通过对行业知识与场景需求的深度融合，为客户提供 NLP、CV、多模态等领域的模型应用解决方案，帮助企业解决特定的业务问题。

方案架构

天宽昇腾云行业大模型适配服务通过深度学习算法优化与高效计算，结合华为昇腾算力，为各行业提供全面的大模型迁移、适配与优化服务。天宽通过深度优化昇腾算力，结合大规模分布式训练、模型微调与部署等核心能力，针对不同行业的需求，为客户提供从模型设计、训练到部署的一站式服务，助力企业快速落地AI应用。

业务架构

图 1-1 业务架构图

版权所有 © 华为技术有限公司

行业大模型适配服务：

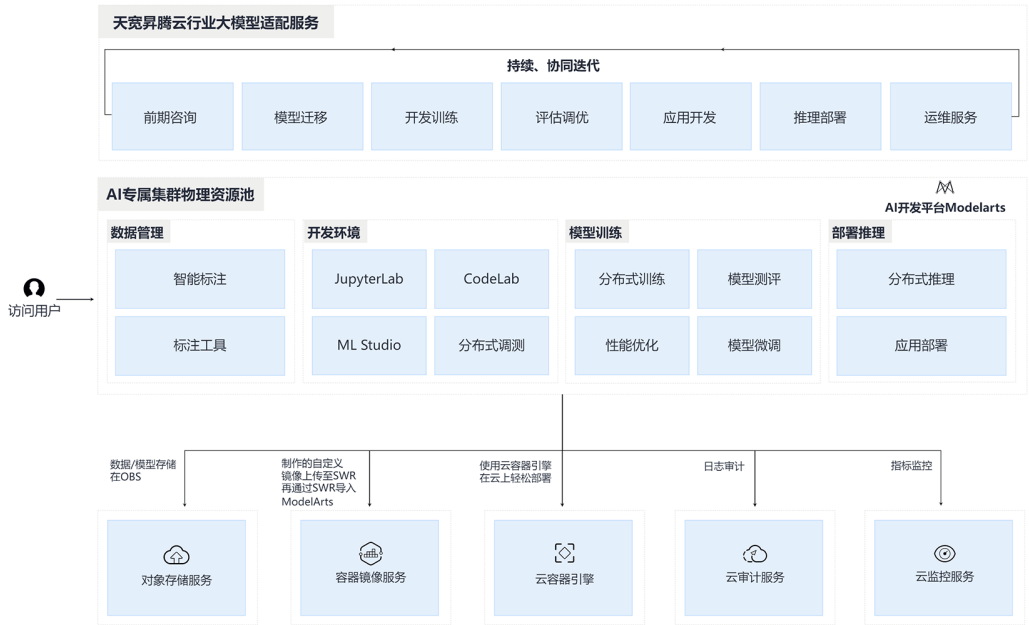
- 昇腾模型与应用开发支持：提供MindSpore 、Pytorch AI框架相关API的使用指导，支持客户基于昇腾平台进行模型开发和模型的并行化改造，解答模型开发训练过程中遇到的技术问题。提供昇腾编程语言ACL（Ascend Computing Language）或MindX SDK相关的API接口的使用指导，支持客户基于昇腾平台进行离线推理应用开发，支持客户使用昇腾ATC工具进行离线模型转换，解答客户在应用开发过程中遇到的技术问题。
- 昇腾工具链使用支持：提供昇腾AIT（Ascend Inference Tools）、ATT（Ascend Training Tools）、MindInsight、MindStudio等昇腾工具链的使用指导，支持客户使用昇腾官方提供的各类高阶组件进行模型迁移分析、模型算子精度采集与模型性能采集，支持客户调用工具实现精度、性能数据的可视化，处理客户在工具链使用过程中遇到的技术问题。

昇腾迁移&优化服务：

- 昇腾适配模型运行支持：基于昇腾已在ModelZoo上发布的模型，支持客户完成模型在昇腾平台上的部署与调测，获取模型网络权重，进行权重格式转换；支持客户进行数据集封装，打通适配模型的训练、微调、在线推理流程；支持客户进行模型的并行化改造，处理适配模型运行过程中的技术问题。
- 模型迁移与调优支持：调研客户业务场景，支持客户分析模型代码结构，分析迁移可行性，设计迁移方案。支持客户进行模型迁移环境部署与训练脚本改造。支持客户进行权重转换，打通在线推理流程，使用昇腾工具链进行调优。处理客户在模型迁移与调优过程中的技术问题。

部署架构

图 1-2 部署架构图



方案通过华为云提供的一站式AI开发平台ModelArts，对象存储服务OBS等服务，为客户提供从模型设计、训练到部署的一站式服务，助力企业快速落地AI应用。

版权所有 © 华为技术有限公司

- AI开发平台ModelArts：提供海量数据预处理及半自动化标注、大规模分布式训练、自动化模型生成及端-边-云模型按需部署能力，帮助用户快速创建和部署模型，管理全周期AI工作流。
- 对象存储服务：存储数据和模型，实现安全、高可靠和低成本的存储需求。
- 云容器引擎：ModelArts使用云容器引擎部署模型为在线服务，支持服务的高并发和弹性伸缩需求。
- 容器镜像服务：使用ModelArts不支持的AI框架构建模型时，可通过构建的自定义镜像导入ModelArts进行训练或推理。
- 云监控服务：使用云监控服务监控在线服务和对应模型负载，执行自动实时监控、告警和通知操作。
- 云审计服务：使用云审计服务记录ModelArts相关的操作事件，便于日后的查询、审计和回溯。

方案优势

通过天宽昇腾云行业大模型适配服务，用户能够在华为云高性价比的昇腾算力支持下，克服技术栈差异、技术理解不足、迁移难度大、开发环境复杂、专业人才短缺和调优经验不足等痛点，实现高效、可靠的AI应用落地，推动企业的数字化转型。

- **高效模型迁移与适配：**支持模型从GPU平台快速迁移至昇腾NPU平台，提供自动化迁移工具与算子适配，确保模型无缝迁移。
- **定制化行业模型开发：**针对不同行业的特定业务场景，提供专属的模型设计与训练服务，满足复杂场景需求，实现准确适配。
- **高性能计算支持：**基于昇腾云的强大算力，通过算子优化、内存管理与梯度优化等技术，显著提升模型的训练效率和推理速度。
- **精度调优与性能优化：**提供专业的精度调试与性能调优服务，确保模型在迁移后能够保持与原平台一致的精度，并优化推理性能。

版权所有 © 华为技术有限公司

2 资源和成本规划

本节介绍解决方案实践中资源规划情况，配置为72B参数大模型开发通用配置，具体实施可根据选取的模型进行规格调整。

表 2-1 资源与成本规划如下表：

云资源	规格	数量	每月费用
AI开发平台 ModelArts	面向开发者的一站式AI开发平台	/	/
AI专属集群物理资源池	modelarts.bm.npu.arm.8sntgb3.d1个次)x 1	1	151620
对象存储服务OBS	标准存储单AZ存储包 | 5TB	1	456
容器镜像服务SWR	容器镜像全生命周期管理服务	/	/
云容器引擎 CCE	CCE容器集群 | 混合集群 | 50节点 | 是	/	/
云审计服务 CTS	云审计服务	/	/
云监控服务 CES	云监控服务	/	/

本案例所涉及的昇腾云行业大模型适配服务报价项如下，实际以收费账单为准：

版权所有 © 华为技术有限公司

表 2-2 昇腾云行业大模型适配服务报价

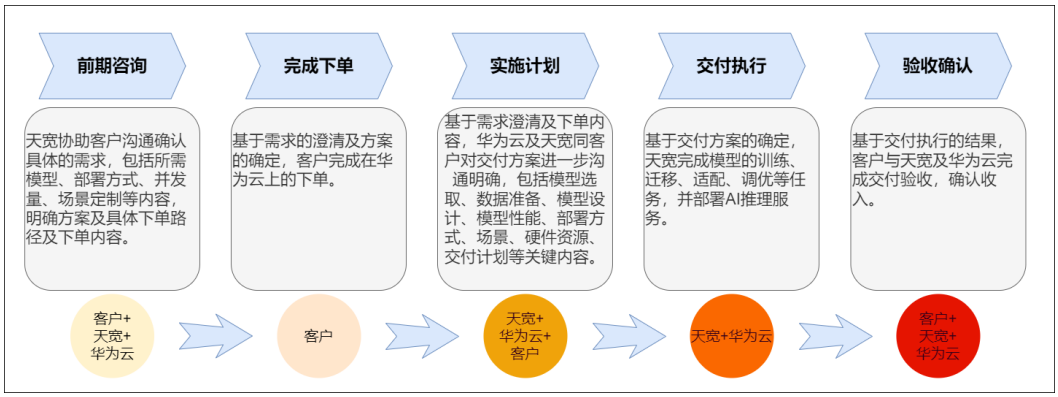
类别	服务内容	量纲
行业大模型适配服务	深入了解客户业务需求，提供各行业大模型开发和适配方案，通过性能评估和反馈机制，优化模型精度与性能，确保模型快速收敛并满足业务需求。	人天
昇腾模型迁移优化服务	调研客户业务场景，支持客户分析模型代码结构，分析迁移可行性，设计迁移方案。支持客户进行模型迁移环境部署与训练脚本改造。	人天

版权所有 © 华为技术有限公司

3 操作流程

天宽昇腾云行业大模型解决方案专业服务已经上架为联运商品，操作流程如下：

图 3-1 操作流程



各流程活动的具体工作和要求如下表格所示：

表 3-1 各流程活动的具体工作和要求

序号	工作内容	具体描述	责任人
1	前期模型开发咨询服务	天宽协助客户沟通确认具体的需求，包括所需模型、部署方式、并发量、场景定制等内容，明确方案及具体下单路径及下单内容。	客户+天宽+华为云
2	完成下单	基于需求的澄清及方案的确定，客户完成在华为云上的下单。	客户
3	实施计划	基于需求澄清及下单内容，华为云及天宽同客户对交付方案进一步沟通明确，包括模型选取、数据准备、模型设计、模型性能、部署方式、场景、硬件资源、交付计划等关键内容。	天宽+华为云+客户

版权所有 © 华为技术有限公司

序号	工作内容	具体描述	责任人
4	交付执行	基于交付方案的确定，天宽完成模型的训练、迁移、适配、调优等任务，并部署AI推理服务。	天宽+华为云
5	验收确认	基于交付执行的结果，客户与天宽及华为云完成交付验收，确认收入。	客户+天宽+华为云

版权所有 © 华为技术有限公司

4 实施步骤

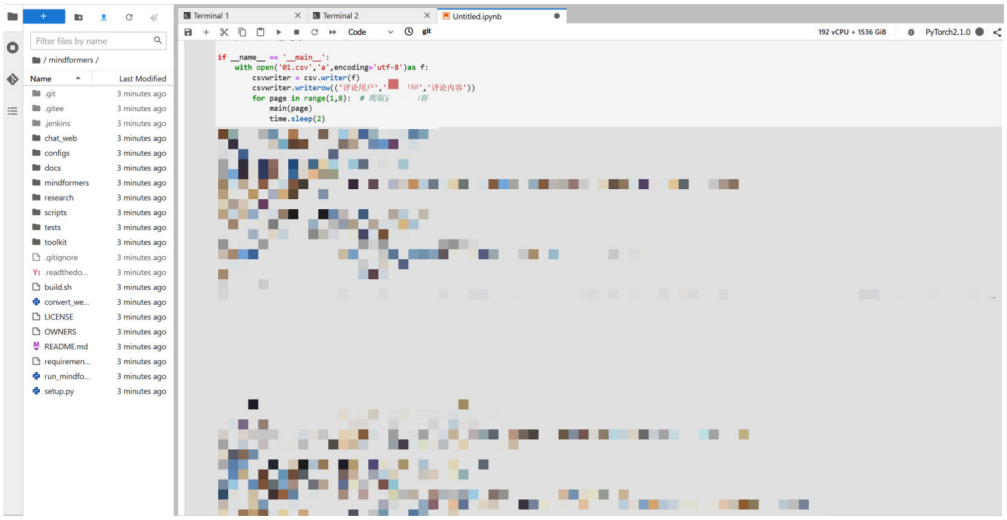
天宽行业大模型适配服务

前期咨询：天宽会深入了解客户所在行业的需求，评估业务场景中的具体问题和痛点。通过与客户的多轮沟通，明确所需解决的问题及目标，为客户量身定制相应的大模型解决方案。同时，天宽会结合模型的技术特点和行业实践，确定模型落地路径，并规划整个模型开发与实施的整体方案。

模型开发与训练：根据客户的具体业务需求及数据特性，天宽将设计出适合该业务场景的模型。此阶段会涉及数据预处理、特征工程及模型架构的选择。

- 天宽在数据采集领域拥有丰富的爬虫开发经验，能够熟练使用Python、JavaScript等编程语言，为客户定制高效的爬虫脚本，从指定的网站和平台采集所需数据。天宽团队在实际项目中曾广泛应用Scrapy、Beautiful Soup和Selenium等工具，确保数据采集的速度和质量。

图 4-1 模型开发与训练 1



- 天宽团队在数据处理方面具备深厚的专业技能，能够熟练运用Python的Pandas和NumPy等库进行高效的数据清洗与预处理。天宽团队掌握全面的数据清洗流程，包括去除重复值、处理缺失数据、检测和修正异常值等操作，确保数据的完整性和一致性。对于大规模数据集，天宽团队擅长使用Apache Spark等大数据处理工具，能够高效地对数据进行清洗、转换和优化。

版权所有 © 华为技术有限公司

图 4-2 天宽行业大模型适配服务 1

```
# 修改数据格式为适配大模型的格式，为下一步数据转换做准备
data_path = pathlib.Path(data_path)
with data_path.open() as f:
    data = json.load(f)

sources = []
for example in data:
    if example.get("input", "") == "":
        sources.append(example['instruction'])
    else:
        instruction = example['instruction']
        if instruction[-1] == ".":
            instruction = instruction[:-1]
        instruction = instruction + ": " + example['input']
        sources.append(instruction)

targets = []
for example in data:
    targets.append(example['output'])

new_data = []
for s, t in zip(sources, targets):
    new_data.append({
        "type": "chatml",
        "conversations": [
            {
                "from": "human",
                "value": s,
            },
            {
                "from": "gpt",
                "value": t,
            },
        ]
    })

flags_ = os.O_WRONLY | os.O_CREAT | os.O_TRUNC
with os.fdopen(os.open(output_path, flags_, 0o750), 'w', encoding='utf-8') as f:
    json.dump(new_data, f, ensure_ascii=False, indent=2)
```

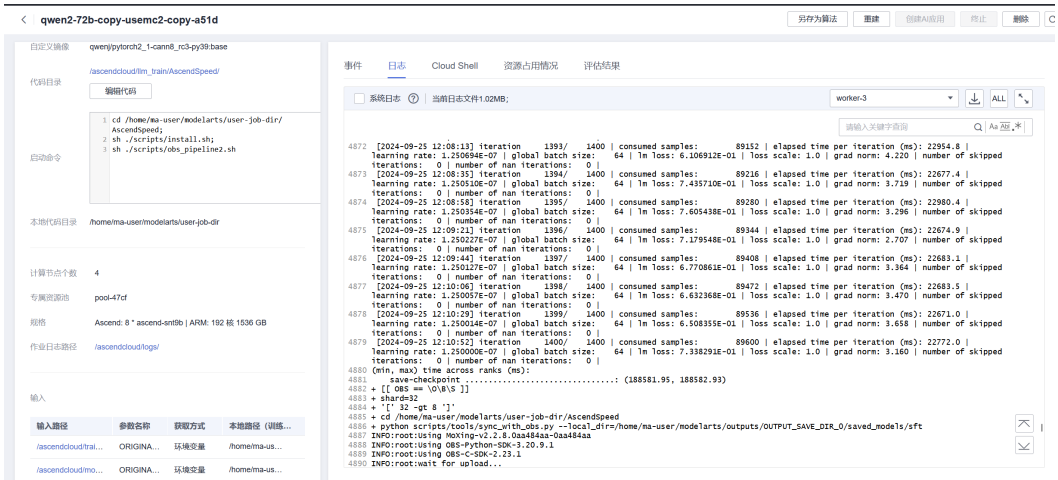

版权所有 © 华为技术有限公司

图 4-3 天宽行业大模型适配服务 2

```
INFO:modellink.tasks.preprocess.data_handler:Vocab size: 64000
INFO:modellink.tasks.preprocess.data_handler:Output prefix: /home/ma-user/modelarts/user-job-dir/processed_for_input/yi-34b/data/sft/ORIGINAL_TRAIN_DATA_PATH_0_packed
INFO:modellink.tasks.preprocess.data_handler:Time to startup:25.115968465805054
INFO:modellink.tasks.preprocess.data_handler:Processed 1000 documents (2536.4008214585483 docs/s, 3.7752261067073687 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 2000 documents (2552.3254204201053 docs/s, 3.8181773337248563 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 3000 documents (2570.5500705103364 docs/s, 3.8274076935632837 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 4000 documents (2478.518917502296 docs/s, 3.7362861459563965 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 5000 documents (2498.302718735448 docs/s, 3.7591019723248147 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 6000 documents (2499.4948533843235 docs/s, 3.7698374830310066 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 7000 documents (2466.1709763012586 docs/s, 3.7476469631160 37 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 8000 documents (2458.232022495823 docs/s, 3.7524882190780056 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 9000 documents (2461.4858143245924 docs/s, 3.7529151488626056 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 10000 documents (2433.963645927016 docs/s, 3.7248601790974933 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 11000 documents (2436.564042018204 docs/s, 3.7285188196591657 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 12000 documents (2423.410505653348 docs/s, 3.72178598302503 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 13000 documents (2428.292775428502 docs/s, 3.732468934093611 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 14000 documents (2423.916955117856 docs/s, 3.7297479773054336 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 15000 documents (2418.8456518085877 docs/s, 3.721411160923699 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 16000 documents (2417.148378648039 docs/s, 3.7246697849645205 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 17000 documents (2423.5640171456985 docs/s, 3.7309094691203057 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 18000 documents (2416.9478343406263 docs/s, 3.7221556605581148 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 19000 documents (2417.9809056282083 docs/s, 3.7313573548673364 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 20000 documents (2418.1632291197243 docs/s, 3.7342811108181353 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 21000 documents (2416.3429001678073 docs/s, 3.726695581269353 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 22000 documents (2418.396703054345 docs/s, 3.7282690498034152 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 23000 documents (2419.594652587273 docs/s, 3.7321582312550685 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 24000 documents (2416.9975102811472 docs/s, 3.7242943947758835 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 25000 documents (2418.2147442753176 docs/s, 3.7263627993217496 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 26000 documents (2412.912024639462 docs/s, 3.7217985127295647 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 27000 documents (2415.579042003239 docs/s, 3.7273267919040047 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 28000 documents (2417.41154481887 docs/s, 3.732477691941928 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 29000 documents (2412.8283673745277 docs/s, 3.7280125066167655 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 30000 documents (2415.380841210298 docs/s, 3.7322657440434712 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 31000 documents (2415.96868279475 docs/s, 3.736736881888219 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 32000 documents (2416.0778644764746 docs/s, 3.732671705857792 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 33000 documents (2418.65835508029 docs/s, 3.737027389326686 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 34000 documents (2418.4104833334745 docs/s, 3.736030318940083 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 35000 documents (2412.748420581174 docs/s, 3.729149790444345 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 36000 documents (2413.6694482181274 docs/s, 3.73366871354287 97 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 37000 documents (2416.099912902282 docs/s, 3.7363374095596753 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 38000 documents (2414.6939076288204 docs/s, 3.732466240489306 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 39000 documents (2415.233623540199 docs/s, 3.7335939067632005 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 40000 documents (2416.38726453418 docs/s, 3.7352572388784893 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 41000 documents (2414.882475292374 docs/s, 3.7314904973185996 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 42000 documents (2414.5509283809156 docs/s, 3.733003148771216 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 43000 documents (2411.5476549999894 docs/s, 3.7298429100393538 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 44000 documents (2412.1671297768735 docs/s, 3.7317570633766346 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 45000 documents (2412.072261689377 docs/s, 3.7304209979563345 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 46000 documents (2410.4517801536063 docs/s, 3.7267965855532545 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 47000 documents (2411.9021222742736 docs/s, 3.729644925708881 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 48000 documents (2409.3478678998727 docs/s, 3.7276142626808784 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 49000 documents (2406.8685583214137 docs/s, 3.7238117253781975 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 50000 documents (2407.083414821415 docs/s, 3.7259677802697 12 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 51000 documents (2405.49343058457 docs/s, 3.7237008595605762 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Processed 52000 documents (2402.4928297415886 docs/s, 3.719177439151047 MB/s).
INFO:modellink.tasks.preprocess.data_handler:Skip 0 sample exceeded seq-length(4096)
/home/ma-user/anaconda3/envs/PyTorch2.1.0/lib/python3.9/tempfile.py:830: ResourceWarning: Implicitly cleaning up <TemporaryDirectory '/tmp/tmpyoylryp9'>
  _warnings.warn(warn_message, ResourceWarning)
+ sleep 5s
++ ls -A /home/ma-user/modelarts/user-job-dir/processed_for_input/yi-34b/data/sft
++ grep 'ORIGINAL_TRAIN_DATA_PATH_0.*.idx'
+ '[' '!' 'ORIGINAL_TRAIN_DATA_PATH_0_packed_attention_mask_document.idx
```

在模型训练过程中，天宽通过配置和管理云资源，确保训练任务的高效运行。借助云计算实例（华为云ECS），天宽团队能够为训练任务分配合适的计算资源，同时利用存储服务（华为云OBS）来存储大规模数据和模型。训练期间，天宽团队使用云服务提供的监控工具，实时跟踪训练进度和资源使用情况，快速发现并解决潜在问题。通过Git等版本控制系统管理模型版本，天宽团队能够在基模型更新时自动触发集成和测试流程，确保训练过程中的稳定性与优化。

图 4-4 训练日志实时跟踪

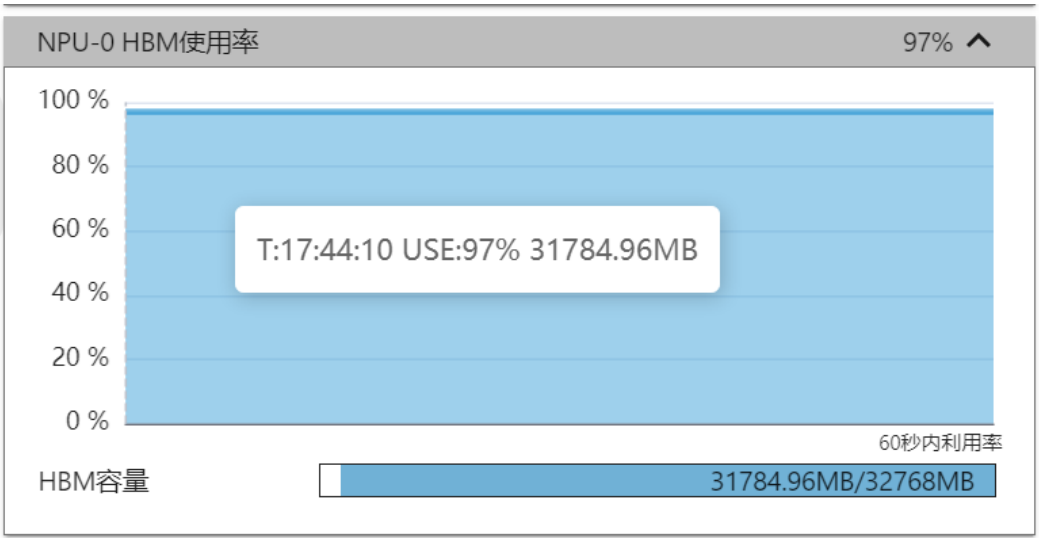


版权所有 © 华为技术有限公司

图 4-5 资源池状态监控

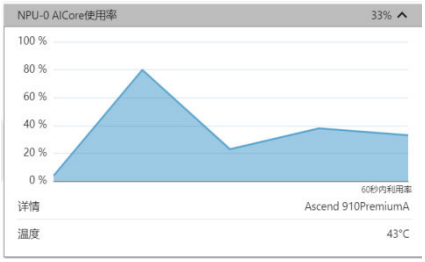
图 4-6 节点状态监控

图 4-7 NPU 状态监控 1



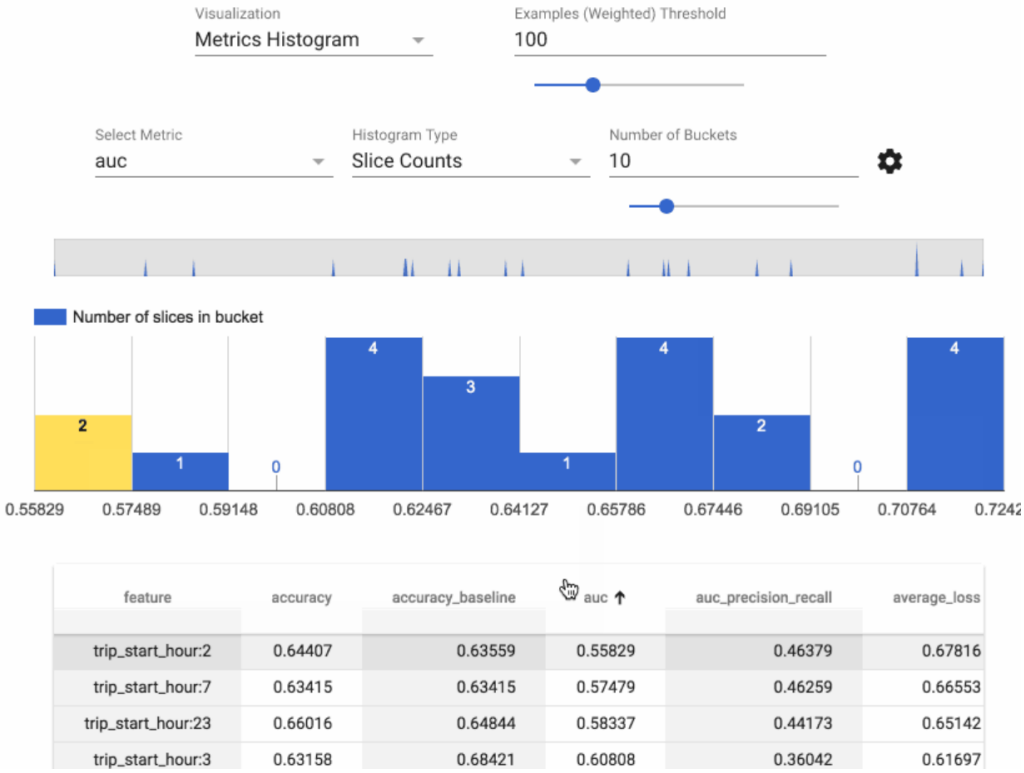
版权所有 © 华为技术有限公司

图 4-8 NPU 状态监控 2



模型评估调优：天宽凭借在多个项目中的实践，积累了丰富的大模型评测经验，能够高效且准确地定义性能指标，如准确性、召回率、精确度等标准评价维度。对于不同的业务场景，天宽还会根据具体需求设定与业务紧密相关的关键绩效指标（KPIs），如用户满意度、转化率或响应时间，确保评测结果能够直接反映模型在真实业务中的表现。在评测准备阶段，天宽特别注重测试集的创建与选择，力求测试数据具有高度的多样性和代表性，以真实反映模型的预期使用场景。这不仅能有效避免因数据偏差导致的评测失真，还能确保模型在不同环境和条件下的一致表现，从而为实际应用提供可靠的依据。在工具和框架的选择上，天宽充分考虑项目的具体需求，精心挑选支持范围广、精确度高、效率和易用性兼备的评测工具。例如，MLPerf作为广泛应用的行业标准工具，能够对多种模型和任务进行性能测试；而TensorFlow Model Analysis则适用于深入分析TensorFlow模型的行为。在需要定制化解决方案的场景下，天宽也会开发自定义评测脚本，确保评测方案能够全面覆盖项目的特殊需求，实现对模型表现的全方位评估和优化。通过这一系统化的评测流程，天宽确保模型能够在实际业务中达到最佳性能。

图 4-9 精度对比



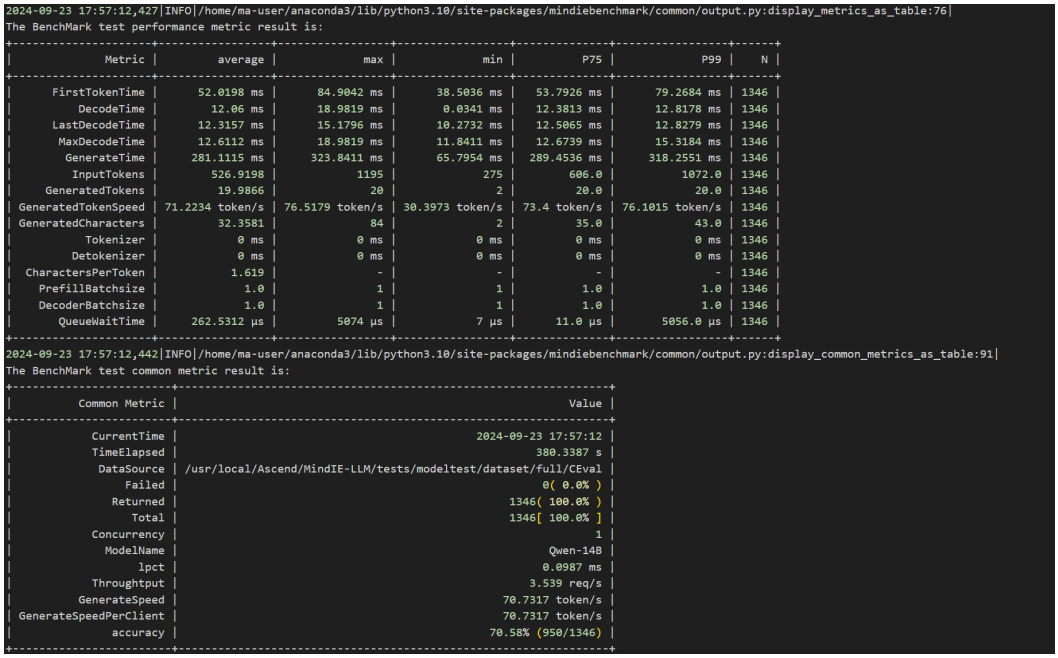
feature	accuracy	accuracy_baseline	auc	auc_precision_recall	average_loss
trip_start_hour:2	0.64407	0.63559	0.55829	0.46379	0.67816
trip_start_hour:7	0.63415	0.63415	0.57479	0.46259	0.66553
trip_start_hour:23	0.66016	0.64844	0.58337	0.44173	0.65142
trip_start_hour:3	0.63158	0.68421	0.60808	0.36042	0.61697

实施模型能力评测时，首先运行评测测试，执行模型在预设的测试集上的推理，并收集相关的性能数据。这一过程也可以通过在线评测来完成，模拟模型在真实环境中的

版权所有 © 华为技术有限公司

表现，从而获取更具参考价值的结果。随后，对测试结果进行统计和分析，运用统计方法来确定模型的性能是否达到了预期标准。如果条件允许，还可以进行A/B测试，以对比不同模型或不同版本的模型在实际场景中的表现，进一步评估其优劣。

图 4-10 C_eval 精度测试 1

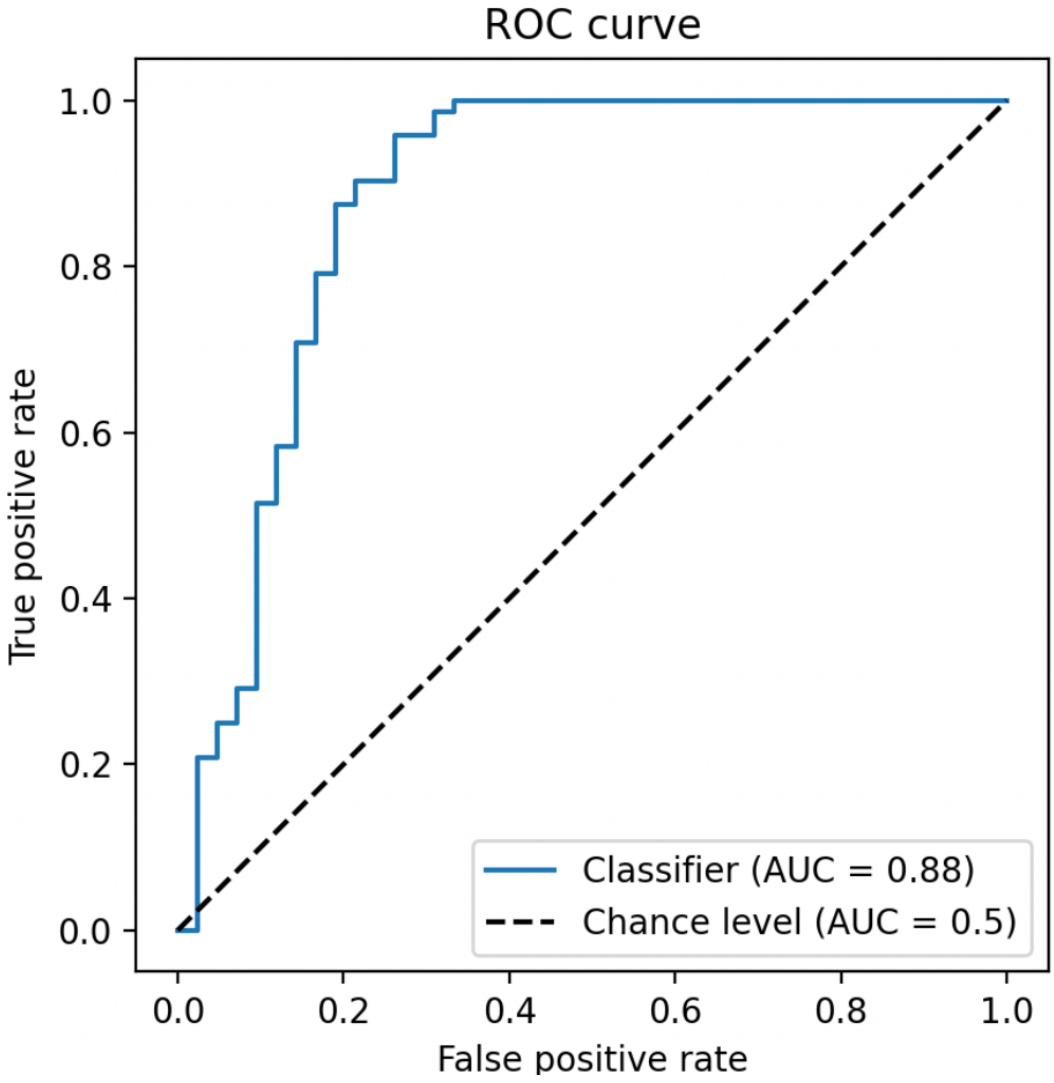


```
2024-09-23 17:57:12,427|INFO|/home/ma-user/anaconda3/lib/python3.10/site-packages/mindiebenchmark/common/output.py:display_metrics_as_table:76|
The BenchMark test performance metric result is:
+--------------------+----------------+----------------+----------------+-------------+----------------+------+
|             Metric |        average |            max |            min |         P75 |            P99 |    N |
+--------------------+----------------+----------------+----------------+-------------+----------------+------+
|      FirstTokenTime |      52.0198 ms |      84.9042 ms |      38.5036 ms |   53.7926 ms |      79.2684 ms | 1346 |
|         DecodeTime |        12.06 ms |      18.9819 ms |       0.0341 ms |   12.3813 ms |      12.8178 ms | 1346 |
|      LastDecodeTime |      12.3157 ms |      15.1796 ms |      10.2732 ms |   12.5065 ms |      12.8279 ms | 1346 |
|       MaxDecodeTime |      12.6112 ms |      18.9819 ms |      11.8411 ms |   12.6739 ms |      15.3184 ms | 1346 |
|        GenerateTime |     281.1115 ms |     323.8411 ms |      65.7954 ms |  289.4536 ms |     318.2551 ms | 1346 |
|         InputTokens |        526.9198 |            1195 |             275 |        606.0 |          1072.0 | 1346 |
|     GeneratedTokens |         19.9866 |              20 |               2 |         20.0 |            20.0 | 1346 |
| GeneratedTokenSpeed | 71.2234 token/s | 76.5179 token/s | 30.3973 token/s | 73.4 token/s | 76.1015 token/s | 1346 |
| GeneratedCharacters |        32.3581 |              84 |               2 |         35.0 |            43.0 | 1346 |
|           Tokenizer |           0 ms |           0 ms |           0 ms |         0 ms |            0 ms | 1346 |
|         Detokenizer |           0 ms |           0 ms |           0 ms |         0 ms |            0 ms | 1346 |
|  CharactersPerToken |          1.619 |              - |              - |           - |              - | 1346 |
|    PrefillBatchsize |            1.0 |              1 |              1 |         1.0 |            1.0 | 1346 |
|    DecoderBatchsize |            1.0 |              1 |              1 |         1.0 |            1.0 | 1346 |
|       QueueWaitTime |     262.5312 µs |        5074 µs |           7 µs |      11.0 µs |      5056.0 µs | 1346 |
+--------------------+----------------+----------------+----------------+-------------+----------------+------+
2024-09-23 17:57:12,442|INFO|/home/ma-user/anaconda3/lib/python3.10/site-packages/mindiebenchmark/common/output.py:display_common_metrics_as_table:91|
The BenchMark test common metric result is:
+------------------------+-------------------------------------------------------------------+
|          Common Metric |                                                             Value |
+------------------------+-------------------------------------------------------------------+
|            CurrentTime |                                               2024-09-23 17:57:12 |
|            TimeElapsed |                                                        380.3387 s |
|             DataSource | /usr/local/Ascend/MindIE-LLM/tests/modeltest/dataset/full/CEval |
|                 Failed |                                                      0( 0.0% ) |
|               Returned |                                                   1346( 100.0% ) |
|                  Total |                                                   1346[ 100.0% ] |
|            Concurrency |                                                                1 |
|              ModelName |                                                         Qwen-14B |
|                   lpct |                                                       0.0987 ms |
|             Throughtput |                                                     3.539 req/s |
|           GenerateSpeed |                                                  70.7317 token/s |
| GenerateSpeedPerClient |                                                  70.7317 token/s |
|               accuracy |                                                70.58% (950/1346) |
+------------------------+-------------------------------------------------------------------+
```

在结果解读阶段，对于未达到标准的指标，需要深入分析可能的原因。常见的问题可能包括数据质量的不足、模型过拟合或欠拟合等。通过混淆矩阵、ROC曲线等工具，可以更深入地理解模型的行为，找到其潜在的弱点，并据此进行相应的改进或优化。

版权所有 © 华为技术有限公司

图 4-11 C_eval 精度测试 2



模型应用开发：基于大模型框架，天宽团队将训练好的模型集成到实际应用中，使其能够在具体的业务场景中发挥作用。例如在自动化流程、预测分析等应用中，构建智能体以应对复杂场景。同时，天宽团队会确保该系统在实际应用中的性能、稳定性及可扩展性。对需要部署在不同环境中的模型，会进行针对性的适配和优化。

模型推理部署：完成模型训练和优化后，进入推理部署阶段。天宽团队将模型打包部署为可供API调用的AI应用，使客户能够在自己的业务场景中方便地集成模型推理服务。通过API接口，客户可以实现与其他应用系统的集成，完成对大规模数据的实时处理和推理操作。天宽团队会确保部署过程中的高效性与稳定性，以应对业务中的并发需求和大数据量处理。

版权所有 © 华为技术有限公司

图 4-12 MindIE 推理服务部署

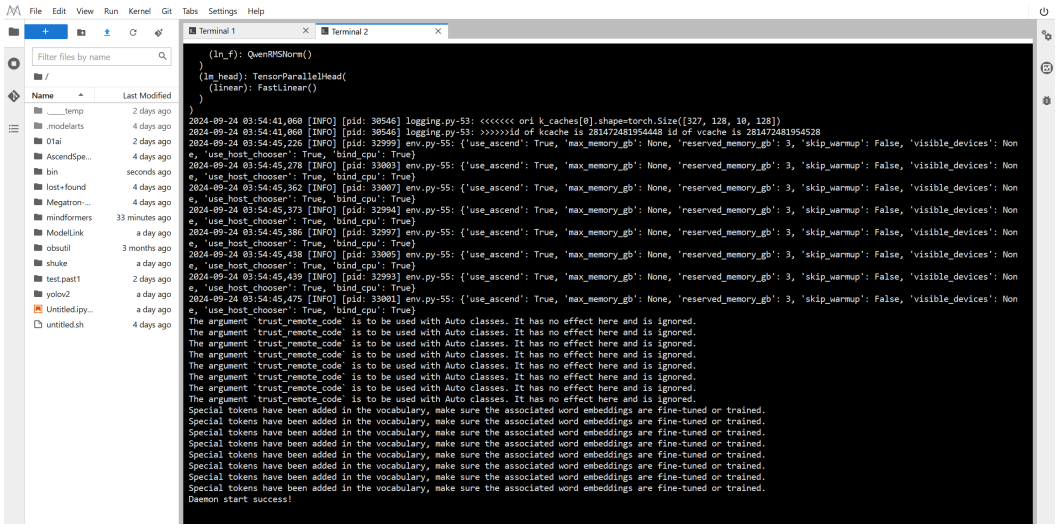


图 4-13 Npu 占用

```
+------------------------------------------------------------------------------------------------+
| npu-smi 24.1.t15                 Version: 24.1.t15                                             |
+---------------------------+---------------+----------------------------------------------------+
| NPU   Name                | Health        | Power(W)    Temp(C)           Hugepages-Usage(page)|
| Chip                      | Bus-Id        | AICore(%)   Memory-Usage(MB)  HBM-Usage(MB)        |
+===========================+===============+====================================================+
| 0     910B1               | OK            | 94.9        30                0    / 0         |
| 0                         | 0000:C1:00.0  | 0           0    / 0          23983/ 65536         |
+===========================+===============+====================================================+
| 1     910B1               | OK            | 99.9        32                0    / 0         |
| 0                         | 0000:01:00.0  | 0           0    / 0          20502/ 65536         |
+===========================+===============+====================================================+
| 2     910B1               | OK            | 95.7        30                0    / 0         |
| 0                         | 0000:C2:00.0  | 0           0    / 0          3168 / 65536         |
+===========================+===============+====================================================+
| 3     910B1               | OK            | 98.7        32                0    / 0         |
| 0                         | 0000:02:00.0  | 0           0    / 0          3167 / 65536         |
+===========================+===============+====================================================+
| 4     910B1               | OK            | 98.6        30                0    / 0         |
| 0                         | 0000:81:00.0  | 0           0    / 0          3333 / 65536         |
+===========================+===============+====================================================+
| 5     910B1               | OK            | 97.0        32                0    / 0         |
| 0                         | 0000:41:00.0  | 0           0    / 0          3333 / 65536         |
+===========================+===============+====================================================+
| 6     910B1               | OK            | 92.7        31                0    / 0         |
| 0                         | 0000:82:00.0  | 0           0    / 0          3333 / 65536         |
+===========================+===============+====================================================+
| 7     910B1               | OK            | 94.1        32                0    / 0         |
| 0                         | 0000:42:00.0  | 0           0    / 0          3333 / 65536         |
+===========================+===============+====================================================+
+---------------------------+---------------+----------------------------------------------------+
| NPU     Chip              | Process id    | Process name             | Process memory(MB)      |
+===========================+===============+====================================================+
| 0       0                 | 1915448       |                          | 1720                    |
| 0       0                 | 471139        |                          | 17222                   |
| 0       0                 | 884549        |                          | 1722                    |
+===========================+===============+====================================================+
| 1       0                 | 471140        |                          | 17224                   |
+===========================+===============+====================================================+
| No running processes found in NPU 2                                                            |
+===========================+===============+====================================================+
| No running processes found in NPU 3                                                            |
+===========================+===============+====================================================+
| No running processes found in NPU 4                                                            |
+===========================+===============+====================================================+
| No running processes found in NPU 5                                                            |
+===========================+===============+====================================================+
| No running processes found in NPU 6                                                            |
+===========================+===============+====================================================+
| No running processes found in NPU 7                                                            |
+===========================+===============+====================================================+
```

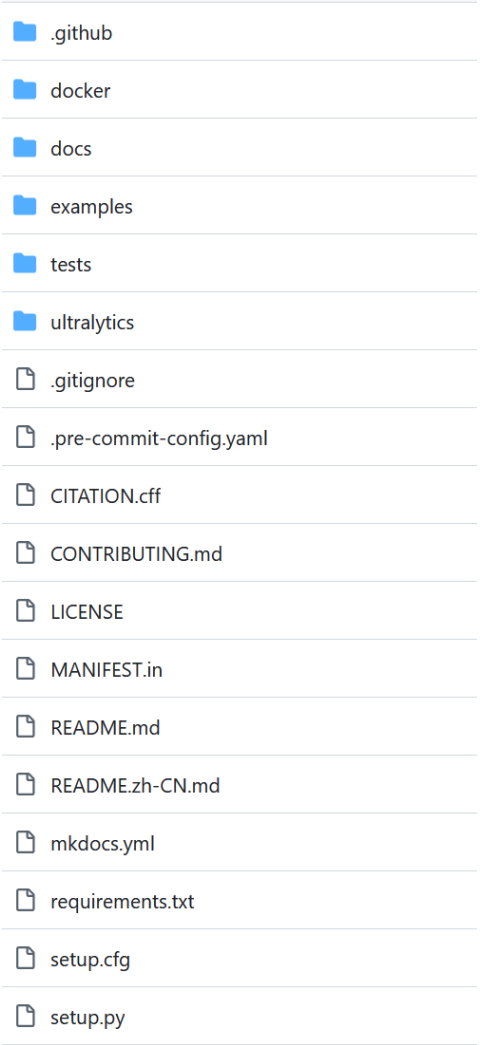

版权所有 © 华为技术有限公司

模型运维服务：部署完成后，天宽团队为客户提供完善的运维服务。通过现场或远程的方式，天宽团队会为客户提供后续支持，包括模型的日常巡检、性能监控、技术指导等。同时，还将提供模型升级服务，确保模型能够与最新的业务需求和技术发展同步。在遇到模型性能下降或业务调整时，天宽团队会迅速响应，并提供针对性的调优或升级方案，保障模型的长期稳定运行。

天宽科技昇腾迁移&优化服务

前期咨询：天宽具备丰富的技术实力和专业经验，可以为客户提供 NLP、CV、多模态等领域 L0 级别大模型的服务部署方案的全面规划设计。将利用大模型（商用大模型、经典开源大模型）、计算机视觉算法（例如 ResNet、YOLO 等）、以及多模态融合技术（如 CLIP 等），为客户量身定制符合其业务需求的部署方案。天宽将综合考虑模型选择、性能优化、部署架构设计、系统可扩展性以及高可用性等方面因素，确保客户能够在实际应用中充分发挥大模型的潜力，实现业务目标的有效实施。

图 4-14 获取模型权重及源码



迁移可行性分析：天宽提供全面的迁移分析服务，帮助客户将基于其他平台（如GPU）的PyTorch训练脚本顺利迁移至昇腾AI处理器。迁移前，天宽会借助msFmkTransplt工具，对客户的PyTorch训练脚本进行全面分析，确保迁移过程的高效性和成功率。该工具能够深入分析脚本中使用的算子、三方库套件、亲和API以及动态

版权所有 © 华为技术有限公司

shape等方面的适配情况，并对模型迁移到昇腾平台的可行性做出详细评估。通过迁移分析，天宽团队能够快速识别训练脚本中不支持的torch API和cuda API，提供针对性优化建议，帮助提升模型在昇腾平台上的精度和性能。此外，针对三方库套件的分析，也可以帮助用户快速发现代码中不支持的第三方库API及其相关依赖项。三方库中的函数如果包含了不被支持的算子或cuda自定义算子，天宽会根据分析结果提供替代方案或进行适配优化，以保证整体系统的兼容性和稳定性。

图 4-15 工具分析

```
(PyTorch2.1.0) [ma-user ms_fmk_transplt]$./pytorch_analyse.sh -i ~/work/yolov5/ -o ~/work/result -v 2.1.0
2024-10-12 16:28:27 [INFO]    Start to check input path...
2024-10-12 16:28:27 [INFO]    PyTorch analysis start working now, please wait for a moment.
2024-10-12 16:28:27 [INFO]    Analysis start...
2024-10-12 16:28:27 [INFO]    [Progress:  0.00%]  Start analysis hubconf.py.
2024-10-12 16:28:35 [INFO]    [Progress:  0.00%]  line: 98 ~ 98 Operation Type: SUGGESTION Message: to has a suggestion about performance
2024-10-12 16:28:35 [INFO]    [Progress:  0.00%]  Analysis hubconf.py complete.
2024-10-12 16:28:35 [INFO]    [Progress:  1.96%]  Start analysis detect.py.
2024-10-12 16:28:40 [INFO]    [Progress:  1.96%]  line: 187 ~ 187 Operation Type: SUGGESTION Message: to has a suggestion about performance
2024-10-12 16:28:40 [INFO]    [Progress:  1.96%]  Analysis detect.py complete.
2024-10-12 16:28:40 [INFO]    [Progress:  3.92%]  Start analysis export.py.
2024-10-12 16:28:46 [WARNING] [Progress:  3.92%]  line: 278 ~ 278 Operation Type: UNSUPPORTED Message: NA
2024-10-12 16:28:49 [INFO]    [Progress:  3.92%]  line: 1381 ~ 1381 Operation Type: SUGGESTION Message: to has a suggestion about performance
2024-10-12 16:28:49 [INFO]    [Progress:  3.92%]  Analysis export.py complete.
2024-10-12 16:28:49 [INFO]    [Progress:  5.88%]  Start analysis benchmarks.py.
2024-10-12 16:28:50 [INFO]    [Progress:  5.88%]  Analysis benchmarks.py complete.
2024-10-12 16:28:50 [INFO]    [Progress:  7.84%]  Start analysis train.py.
2024-10-12 16:28:58 [INFO]    [Progress:  7.84%]  line: 216 ~ 216 Operation Type: SUGGESTION Message: to has a suggestion about performance
2024-10-12 16:28:58 [INFO]    [Progress:  7.84%]  line: 223 ~ 223 Operation Type: SUGGESTION Message: to has a suggestion about performance
2024-10-12 16:28:58 [INFO]    [Progress:  7.84%]  line: 281 ~ 281 Operation Type: SUGGESTION Message: to has a suggestion about performance
2024-10-12 16:28:58 [INFO]    [Progress:  7.84%]  line: 343 ~ 343 Operation Type: SUGGESTION Message: to has a suggestion about performance
2024-10-12 16:28:58 [INFO]    [Progress:  7.84%]  line: 414 ~ 414 Operation Type: SUGGESTION Message: to has a suggestion about performance
2024-10-12 16:28:58 [INFO]    [Progress:  7.84%]  Analysis train.py complete.
2024-10-12 16:28:58 [INFO]    [Progress:  9.80%]  Start analysis val.py.
2024-10-12 16:29:01 [WARNING] [Progress:  9.80%]  line: 290 ~ 290 Operation Type: UNSUPPORTED Message: NA
2024-10-12 16:29:02 [INFO]    [Progress:  9.80%]  line: 334 ~ 334 Operation Type: SUGGESTION Message: to has a suggestion about performance
2024-10-12 16:29:02 [INFO]    [Progress:  9.80%]  Analysis val.py complete.
2024-10-12 16:29:02 [INFO]    [Progress: 11.76%]  Start analysis segment/predict.py.
```

图 4-16 生成结果

```
2024-10-12 16:31:07 [WARNING] [Progress: 98.04%]  line: 127 ~ 127 Operation Type: UNSUPPORTED Message: NA
2024-10-12 16:31:13 [INFO]    [Progress: 98.04%]  line: 124 ~ 124 Operation Type: SUGGESTION Message: torch.nn.Linear has a suggestion about performance
2024-10-12 16:31:13 [INFO]    [Progress: 98.04%]  line: 125 ~ 125 Operation Type: SUGGESTION Message: torch.nn.Linear has a suggestion about performance
2024-10-12 16:31:13 [INFO]    [Progress: 98.04%]  line: 126 ~ 126 Operation Type: SUGGESTION Message: torch.nn.Linear has a suggestion about performance
2024-10-12 16:31:13 [INFO]    [Progress: 98.04%]  line: 128 ~ 128 Operation Type: SUGGESTION Message: torch.nn.Linear has a suggestion about performance
2024-10-12 16:31:13 [INFO]    [Progress: 98.04%]  line: 129 ~ 129 Operation Type: SUGGESTION Message: torch.nn.Linear has a suggestion about performance
2024-10-12 16:31:13 [INFO]    [Progress: 98.04%]  line: 149 ~ 149 Operation Type: SUGGESTION Message: torch.nn.Linear has a suggestion about performance
2024-10-12 16:31:13 [INFO]    [Progress: 98.04%]  line: 580 ~ 580 Operation Type: SUGGESTION Message: to has a suggestion about performance
2024-10-12 16:31:13 [INFO]    [Progress: 98.04%]  line: 763 ~ 763 Operation Type: SUGGESTION Message: to has a suggestion about performance
2024-10-12 16:31:13 [INFO]    [Progress: 98.04%]  line: 866 ~ 866 Operation Type: SUGGESTION Message: to has a suggestion about performance
2024-10-12 16:31:13 [INFO]    [Progress: 98.04%]  line: 890 ~ 890 Operation Type: SUGGESTION Message: to has a suggestion about performance
2024-10-12 16:31:13 [INFO]    [Progress: 98.04%]  line: 1103 ~ 1103 Operation Type: SUGGESTION Message: torch.nn.Linear has a suggestion about performance
2024-10-12 16:31:13 [INFO]    [Progress: 98.04%]  Analysis models/common.py complete.
2024-10-12 16:31:13 [INFO]    [Progress:100.00%]  Analyse run success, welcome to the next use.
2024-10-12 16:31:13 [INFO]    The detailed transplant result files are in the output path you defined, the relative path is yolov5_analysis.
+---------------------------+------------+
|           files           | statistics |
+---------------------------+------------+
| api_performance_advice.csv |     57     |
|      cuda_op_list.csv      |     0      |
|    unsupported_api.csv     |     5      |
|      unknown_api.csv       |     19     |
|  api_precision_advice.csv  |     1      |
+---------------------------+------------+
```


版权所有 © 华为技术有限公司

图 4-17 不支持算子列表

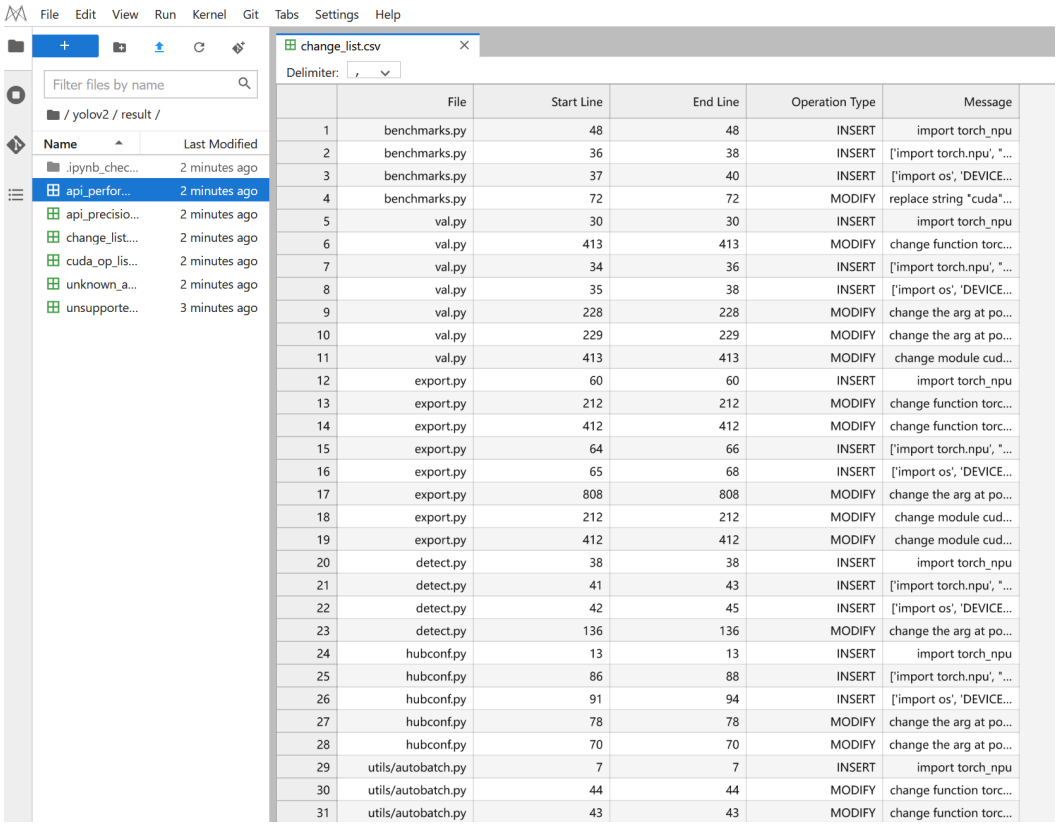
Delimiter: ,

	File	Start Line	End Line	OP	Tips
1	export.py	158	158	torch.jit.trace	
2	export.py	162	162	torch.utils.mobile_op...	
3	export.py	309	309	torch.jit.trace	
4	utils/dataloaders.py	136	136	torch.Generator	
5	utils/dataloaders.py	202	202	torch.Generator	
6	utils/dataloaders.py	1360	1360	torch.Generator	
7	utils/loss.py	169	169	torch.full_like	
8	utils/general.py	288	288	torch.use_determinis...	
9	utils/segment/datalo...	69	69	torch.Generator	
10	utils/segment/loss.py	83	83	torch.full_like	
11	utils/loggers/__init_....	460	460	torch.jit.trace	
12	segment/train.py	236	236	torch.nn.DataParallel	
13	segment/train.py	240	240	torch.nn.SyncBatchN...	
14	models/common.py	475	475	torch.jit.load	
15	.ipynb_checkpoints/t...	246	246	torch.nn.DataParallel	
16	.ipynb_checkpoints/t...	250	250	torch.nn.SyncBatchN...	

模型迁移：天宽通过三种方式完成模型迁移任务。导入import torch_npu和from torch_npu.contrib import transfer_to_npu库，可以实现自动迁移。在这种方法下，训练脚本会在运行过程中自动将CUDA接口替换为昇腾AI处理器支持的NPU接口，整个流程是在训练中动态完成转换，简化了操作，提升了效率。使用迁移工具ms_fmk_transplt是另一种迁移方式。通过这个工具，训练脚本中的CUDA接口会被自动替换为NPU接口，并生成迁移报告，其中包括脚本转换日志、不支持的算子列表和脚本修改记录。完成脚本转换后，可直接运行转换后的脚本进行训练，实现快速迁移。在手工迁移中，天宽团队通过分析模型，对比GPU和NPU接口，对训练脚本进行手动调整，以支持昇腾AI处理器的运行。手工迁移的核心在于将训练设备切换至NPU，并手动替换脚本中适配GPU的接口。在涉及多卡分布式训练时，还需要修改芯片间的通信方式，使用昇腾支持的hccl。通过这些灵活的迁移方式，天宽能够高效地满足客户不同场景下的迁移需求，并优化模型性能。

版权所有 © 华为技术有限公司

图 4-18 工具迁移列举出修改的算子列表



	File	Start Line	End Line	Operation Type	Message
1	benchmarks.py	48	48	INSERT	import torch_npu
2	benchmarks.py	36	38	INSERT	['import torch.npu', "...
3	benchmarks.py	37	40	INSERT	['import os', 'DEVICE...
4	benchmarks.py	72	72	MODIFY	replace string "cuda"...
5	val.py	30	30	INSERT	import torch_npu
6	val.py	413	413	MODIFY	change function torc...
7	val.py	34	36	INSERT	['import torch.npu', "...
8	val.py	35	38	INSERT	['import os', 'DEVICE...
9	val.py	228	228	MODIFY	change the arg at po...
10	val.py	229	229	MODIFY	change the arg at po...
11	val.py	413	413	MODIFY	change module cud...
12	export.py	60	60	INSERT	import torch_npu
13	export.py	212	212	MODIFY	change function torc...
14	export.py	412	412	MODIFY	change function torc...
15	export.py	64	66	INSERT	['import torch.npu', "...
16	export.py	65	68	INSERT	['import os', 'DEVICE...
17	export.py	808	808	MODIFY	change the arg at po...
18	export.py	212	212	MODIFY	change module cud...
19	export.py	412	412	MODIFY	change module cud...
20	detect.py	38	38	INSERT	import torch_npu
21	detect.py	41	43	INSERT	['import torch.npu', "...
22	detect.py	42	45	INSERT	['import os', 'DEVICE...
23	detect.py	136	136	MODIFY	change the arg at po...
24	hubconf.py	13	13	INSERT	import torch_npu
25	hubconf.py	86	88	INSERT	['import torch.npu', "...
26	hubconf.py	91	94	INSERT	['import os', 'DEVICE...
27	hubconf.py	78	78	MODIFY	change the arg at po...
28	hubconf.py	70	70	MODIFY	change the arg at po...
29	utils/autobatch.py	7	7	INSERT	import torch_npu
30	utils/autobatch.py	44	44	MODIFY	change function torc...
31	utils/autobatch.py	43	43	MODIFY	change function torc...

图 4-19 修改不支持的算子

算子名称	处理方法
torch.full_like	支持的算子
torch.Generator	不支持，但是随机类算法可以放在 cpu 侧执行，torch.Generator(device='cpu')
torch.jit.load	支持的算子
torch.jit.trace	支持的算子
torch.nn.DataParallel	不支持，更改使用 torch.nn.parallel.DistributedDataParallel
torch.nn.SyncBatchNorm.convert_sync_batchnorm	支持的算子
torch.use_deterministic_algorithms	不支持，查看代码后发现与 torch1.12 版本相关，没有影响，直接删除相关代码块
torch.utils.mobile_optimizer.optimize_for_mobile	不支持，但是只在 export.py 中，不影响训推任务

模型评估与调优

天宽凭借在多个项目中的实践，积累了丰富的大模型评测经验，能够高效且准确地定义性能指标，如准确性、召回率、精确度等标准评价维度。对于不同的业务场景，天宽还会根据具体需求设定与业务紧密相关的关键绩效指标（KPIs），如用户满意度、转化率或响应时间，确保评测结果能够直接反映模型在真实业务中的表现。在评测准备阶段，天宽特别注重测试集的创建与选择，力求测试数据具有高度的多样性和代表性，以真实反映模型的预期使用场景。这不仅能有效避免因数据偏差导致的评测失真，还能确保模型在不同环境和条件下的一致表现，从而为实际应用提供可靠的依据。在工具和框架的选择上，天宽充分考虑项目的具体需求，精心挑选支持范围广、精确度高、效率和易用性兼备的评测工具。例如，MLPerf作为广泛应用的行业标准工具，能够对多种模型和任务进行性能测试；而TensorFlow Model Analysis则适用于深

版权所有 © 华为技术有限公司

入分析TensorFlow模型的行为。在需要定制化解决方案的场景下，天宽也会开发自定义评测脚本，确保评测方案能够全面覆盖项目的特殊需求，实现对模型表现的全方位评估和优化。通过这一系统化的评测流程，天宽确保模型能够在实际业务中达到最佳性能。

实施模型能力评测时，首先运行评测测试，执行模型在预设的测试集上的推理，并收集相关的性能数据。这一过程也可以通过在线评测来完成，模拟模型在真实环境中的表现，从而获取更具参考价值的结果。随后，对测试结果进行统计和分析，运用统计方法来确定模型的性能是否达到了预期标准。如果条件允许，还可以进行A/B测试，以对比不同模型或不同版本的模型在实际场景中的表现，进一步评估其优劣。

图 4-20 评估脚本

```
check_requirements(exclude=('tensorboard', 'thop'))

if opt.task in ('train', 'val', 'test'):  # run normally
    if opt.conf_thres > 0.001:
        LOGGER.info(f'WARNING ⚠ confidence threshold {opt.conf_thres} > 0.001 produces invalid results')
    if opt.save_hybrid:
        LOGGER.info('WARNING ⚠ --save-hybrid will return high mAP from hybrid labels, not from predictions alone')
    run(**vars(opt))

else:
    weights = opt.weights if isinstance(opt.weights, list) else [opt.weights]
    opt.half = torch.npu.is_available() and opt.device != 'cpu'  # FP16 for fastest results
    if opt.task == 'speed':  # speed benchmarks
        # python val.py --task speed --data coco.yaml --batch 1 --weights yolov5n.pt yolov5s.pt...
        opt.conf_thres, opt.iou_thres, opt.save_json = 0.25, 0.45, False
        for opt.weights in weights:
            run(**vars(opt), plots=False)

    elif opt.task == 'study':  # speed vs mAP benchmarks
        # python val.py --task study --data coco.yaml --iou 0.7 --weights yolov5n.pt yolov5s.pt...
        for opt.weights in weights:
            f = f'study_{Path(opt.data).stem}_{Path(opt.weights).stem}.txt'  # filename to save to
            x, y = list(range(256, 1536 + 128, 128)), []  # x axis (image sizes), y axis
            for opt.imgsz in x:  # img-size
                LOGGER.info(f'\nRunning {f} --imgsz {opt.imgsz}...')
                r, _, t = run(**vars(opt), plots=False)
                y.append(r + t)  # results and times
            np.savetxt(f, y, fmt='%10.4g')  # save
        os.system('zip -r study.zip study_*.txt')
        plot_val_study(x=x)  # plot
```


版权所有 © 华为技术有限公司

图 4-21 结果显示

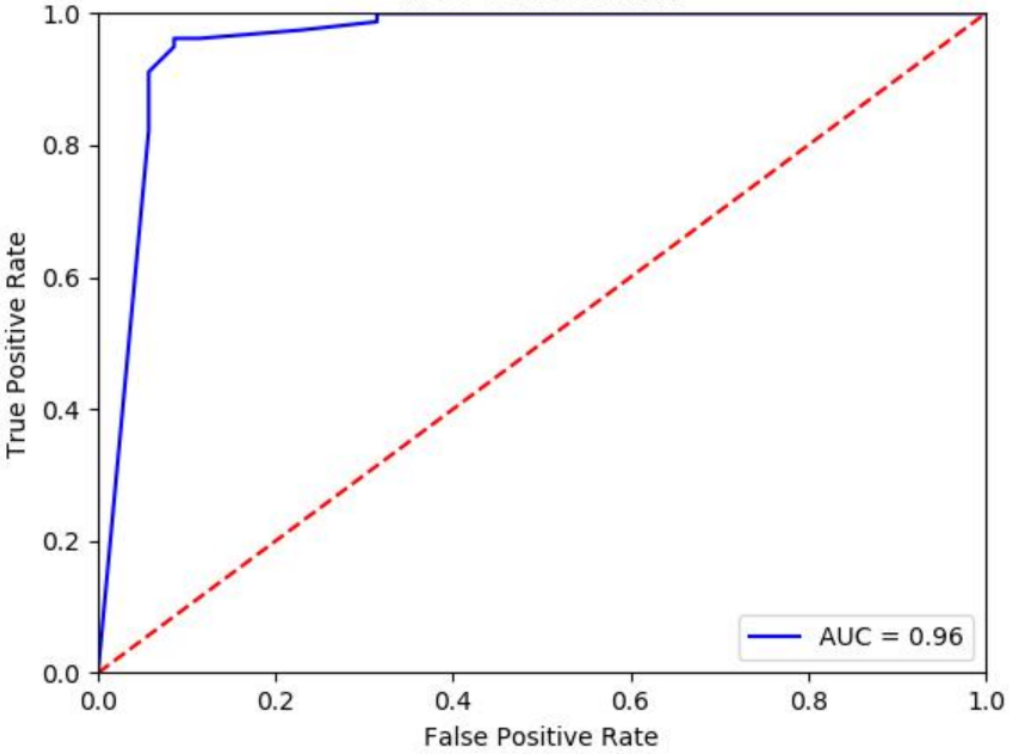
```
Evaluating pycocotools mAP... saving runs/val/exp3/yolov5s_predictions.json...
Results saved to •[1mruns/val/exp3•[0m
loading annotations into memory...
Done (t=0.85s)
creating index...
index created!
Loading and preparing results...
DONE (t=37.09s)
creating index...
index created!
Running per image evaluation...
Evaluate annotation type *bbox*
DONE (t=83.31s).
Accumulating evaluation results...
DONE (t=35.06s).
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.002
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.004
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.002
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.004
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.004
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.003
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.042
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.084
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.092
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.053
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.100
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.152
```

在结果解读阶段，对于未达到标准的指标，需要深入分析可能的原因。常见的问题可能包括数据质量的不足、模型过拟合或欠拟合等。通过混淆矩阵、ROC曲线等工具，可以更深入地理解模型的行为，找到其潜在的弱点，并据此进行相应的改进或优化。

模型交付：在交付阶段准备详细的评测报告，清晰地描述评测过程、结果以及优化建议。同时，提供可交互的仪表板，使非技术利益相关者也能够理解评测结果。基于评测反馈，模型架构可能需要通过增加或减少层次来进行调整，或者通过引入更多的数据预处理步骤来提升输入数据的质量。此外，自动化测试流程的设立，能够确保模型定期接受性能评估，持续满足业务需求。

版权所有 © 华为技术有限公司

图 4-22 测评结果展示



版权所有 © 华为技术有限公司

5 修订记录

表 5-1 修订记录

发布日期	修订记录
2024-11-27	第一次正式发布。

版权所有 © 华为技术有限公司