***PaddleSOT***

**需求说明书**

|  |  |  |  |
| --- | --- | --- | --- |
| 日 期 | 版 本 | 说明 | 作 者 |
| 2025/4/13 | 1.0 | 关于项目的需求说明 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

[1引言 3](#_Toc33515629)

[1.1编写目的 3](#_Toc33515630)

[1.2背景 3](#_Toc33515631)

[1.3定义 3](#_Toc33515632)

[2任务概述 3](#_Toc33515633)

[2.1目标 3](#_Toc33515634)

[2.2用户的特点 4](#_Toc33515635)

[3需求规定 4](#_Toc33515636)

[3.1对功能的规定 4](#_Toc33515637)

[3.2对性能的规定 6](#_Toc33515638)

[3.2.1精度 6](#_Toc33515639)

[3.2.2时间特性要求 6](#_Toc33515640)

[3.2.3灵活性 6](#_Toc33515641)

[3.3其他要求 7](#_Toc33515642)

[3.3.1 输人输出要求 7](#_Toc33515643)

[3.3.2 数据管理能力要求 7](#_Toc33515644)

[3.3.3 故障处理要求 7](#_Toc33515645)

*3.3.4 其他专门要求*..................................................................................................................8

[4运行环境规定 8](#_Toc33515647)

[4.1设备 8](#_Toc33515648)

[4.2支持软件](#_Toc33515649) 9

[4.3接口 9](#_Toc33515650)

# 1引言

## 1.1编写目的

本软件需求说明书旨在系统性地描述PaddlePaddle框架中 Symbolic Opcode Translator（以下简称 PaddleSOT）的功能、性能及运行环境需求，为后续的系统设计、开发和测试提供依据。

预期读者包括：项目开发人员、测试人员、产品经理、维护工程师以及用户代表。

## 1.2背景

本项目为PaddlePaddle孵化项目，旨在通过符号方式翻译Python Bytecode，实现动态图静态图转换（dy2static）过程中的统一中间表示（IR）生成，提升静态图模式的兼容性与性能。

## 1.3定义

PaddleSOT：Paddle Symbolic Opcode Translator  
IR：Intermediate Representation，中间表示  
dy2static：动态图转静态图  
Bytecode：Python编译后的字节码指令集

# 2任务概述

## 2.1目标

本软件的目标是构建一个基于字节码级别的转换器，将Python动态计算图中的执行逻辑翻译为 IR，从而用于图优化与部署阶段，兼容更多语法结构，提高运行性能。该系统将作为PaddlePaddle框架的组件之一，与原有动态图、静态图子系统协作工作。

## 2.2用户的特点

用户主要为深度学习模型开发者、框架维护人员及研究人员。他们具备Python编程能力，有一定的深度学习框架使用经验，对代码语义及性能有较高要求。

# 3需求规定

## 3.1对功能的规定

以下为PaddleSOT系统的核心功能列表及其详细说明：

|  |  |  |  |
| --- | --- | --- | --- |
| 功能编号 | 1.1 | 功能简述 | 解析Python字节码 |
| 前置条件 | 接收到函数对象 | | |
| 触发因素 | 调用入口函数 | | |
| 执行动作 | 1. 使用dis获取Python字节码序列 | | |
| 例外情况 | 无 | | |
| 后置条件 | 获取字节码及操作数序列 | | |
| 备注 | 使用Python官方dis模块 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 功能编号 | 1.2 | 功能简述 | 建立执行帧信息 |
| 前置条件 | 获取字节码 | | |
| 触发因素 | 分析栈结构 | | |
| 执行动作 | 构建Frame对象并记录变量名、堆栈、控制流等 | | |
| 例外情况 | 控制流异常或闭包嵌套 | | |
| 后置条件 | 构造Frame对象 | | |
| 备注 | 无 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 功能编号 | 1.3 | 功能简述 | 转换字节码为IR操作 |
| 前置条件 | 拥有Frame对象 | | |
| 触发因素 | 调用转换器 | | |
| 执行动作 | 逐条翻译为 IR 节点 | | |
| 例外情况 | 不支持指令 | | |
| 后置条件 | 构建IR节点序列 | | |
| 备注 | 无 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 功能编号 | 1.4 | 功能简述 | 构造控制流图 |
| 前置条件 | 获取IR节点序列 | | |
| 触发因素 | 控制流语句识别 | | |
| 执行动作 | 识别jump、branch语句生成控制流图 | | |
| 例外情况 | 异常跳转结构 | | |
| 后置条件 | 输出可视化/序列化 CFG | | |
| 备注 | 无 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 功能编号 | 1.5 | 功能简述 | 返回最终IR表示 |
| 前置条件 | CFG 完整 | | |
| 触发因素 | 请求IR输出 | | |
| 执行动作 | 整合并优化IR表达形式 | | |
| 例外情况 | IR不完整 | | |
| 后置条件 | 返回用于静态图构建的IR表达式 | | |
| 备注 |  | | |

## 

## 3.2对性能的规定

### 3.2.1精度

翻译结果需在语义上等价于原始Python函数，确保IR能准确表示变量状态与控制流逻辑。应支持复杂语法结构，如try-except、with、yield、闭包等语义还原。

### 3.2.2时间特性要求

系统应具备高效的处理能力，确保动态图转换过程中性能可接受：

* 1.小型函数（10-50字节码）转换耗时 < 100ms
* 2.中等复杂度函数（50-150字节码）耗时 < 500ms
* 3.每条字节码平均转换耗时 < 5ms

系统应提供基准测试框架，以验证处理性能并支持优化方向评估。

### 3.2.3灵活性

PaddleSOT 应具有良好的适配性与可扩展性，满足以下灵活性要求：

* 1.支持多版本 Python字节码差异
* 2.支持插件式指令扩展机制
* 3.与Paddle动态图/静态图模块解耦，可独立部署
* 4.可适配未来动态图Trace重构或IR优化器改动

## 3.3其他要求

### 3.3.1 输入输出要求

输入输出要求如下：

**1.输入类型**：

用户定义的Python函数对象（Callable），可包含控制流、嵌套结构、闭包、异常处理等语法。

**2.输入格式**：

支持标准Python语言格式，无需AST或源码级修改。

**3.输出类型**：

IR（中间表示）节点序列，可视为内部图结构。

**4.输出格式**：

默认为Python对象结构，支持序列化为JSON、Graphviz DOT格式或 protobuf。

### 3.3.2 数据管理能力要求

要求如下：

**1.缓存能力**：

对已转换函数IR可选择性缓存，加快后续重复调用的转换速度。

支持基于函数签名与结构的哈希缓存键生成策略。

**2.数据规模预估**：

单个中等复杂函数IR节点数目约为 100～1000。

典型项目中IR存储总量应可控制在数MB～数十MB范围内。

**3.配置与管理**：

支持设置缓存清理策略（如 LRU、TTL）与最大占用空间阈值。

### 3.3.3 故障处理要求

为保证系统稳定性，PaddleSOT应提供完善的故障处理机制：

**1.指令不支持**：

遇到不支持的 Python 字节码（Opcode）时，抛出 UnsupportedOpcodeError，并记录详细错误日志。

**2.控制流异常：**

对于非法的跳转或异常嵌套结构，应中止当前转换并提供用户友好的报错提示。

**3.栈不一致**：

在执行帧过程中检测堆栈不平衡时立即中止并回滚状态。

**4.调试辅助：**

在失败节点附近提供源码位置信息、高亮可能的问题段。

### **3.3.4 其他专门要求**

**1.安全与隔离性**：

PaddleSOT不执行用户函数，仅对其字节码符号分析，具备良好的执行隔离性，避免运行时副作用。

**2.可维护性**：

模块结构清晰，指令翻译器采用注册机制，便于维护与新增。

**3.可移植性**：

核心组件基于纯Python实现，可跨平台部署。

**4.易用性**：

提供统一入口API和命令行工具，支持调试参数和 IR可视化选项。

**5.可测试性**：

每条字节码指令的转换逻辑均有对应的单元测试，保证翻译逻辑正确性。

# 4运行环境规定

## 4.1设备

PaddleSOT 的运行对硬件资源要求较低，可在普通工作站或服务器上运行。

建议配置：

**1.CPU**：支持AVX指令集的x86\_64架构处理器

**2.内存**：推荐≥ 4GB，处理大型函数时建议≥ 8GB

**3.磁盘**：最低500MB空间用于缓存与日志

**4.显卡**：非必须，PaddleSOT处理阶段无需 GPU 参与

**5.网络**：如需远程调试或调用分布式API，建议具备外网能力

## 4.2支持软件

所需软件依赖如下：

**1.操作系统**：支持 Linux（Ubuntu 18.04+）、Windows 10、macOS 12+

**2.Python 环境：**Python 3.8 - 3.11，建议使用virtualenv或conda管理

**3.框架依赖**：PaddlePaddle 2.5+ 版本

**4.其他工具库**：

dis（Python 标准库）— 字节码解析

graphviz — 控制流图可视化

astunparse — Python AST 反解析（用于展示和调试）

## 4.3接口

提供的主要接口和其对接模块：

**1.输入接口**：

dy2ir(func: Callable) 接收用户定义的 Python 函数，进行符号执行

**2.中间交互接口**：

与 Paddle 动态图 Trace 模块集成，支持 fallback 触发

**3.输出接口**：

生成结构化 IR 表示，支持 JSON、Graph、结构体等格式

**4.第三方兼容**：

IR 可接入 Paddle 静态图构建系统或定制后端优化器