|  |  |  |
| --- | --- | --- |
| 第五章 自定义算子开发与集成 | | |
| 姓名： 吴文韬 | 学号：20373914 | 班级：212111 |
| **注意：课程实验代码和报告需独立完成，严禁抄袭!如发现有抄袭情况（同年级或者抄袭高年级），相应的实验或报告部分直接0分。** 实验4.4自定义TensorFlow CPU算子(50’’) **1.比较使用NumPy和C++进行算子实现有何不同？**  **C++实现的CPU算子性能更高，python在开发效率、易用性上有着巨大的优势，但作为一个解释性语言，在性能方面还是存在比较大的缺陷**  **2. 本实验中编程框架的内部核心代码为何使用C/C++语言，为何不使用Python等语言？**  **如一问中提到的，python在开发效率、易用性上有着巨大的优势，但作为一个解释性语言，在性能方面还是存在比较大的缺陷，在各类AI服务化过程中，采用python作为模型快速构建工具，使用高级语言(如C++，java)作为服务化程序实现是大势所趋（因为这里进行的是算子开发）** 实验5.1智能编程语言算子开发与集成实验(50’’) **1.请用自己的话解释：本次实验中把自定义算子集成到编程框架TensorFlow中的实现逻辑和流程。**   1. **算子实现：编写Kernel程序，实现PowerDifference算子；编写运行时程序，实现算子调用函数；编写Main程序，加载输入数据并调用MLU函数对数据进行计算。** 2. **框架集成：注册接口，实现PluginOp接口；进行DLP算子集成；实现算子类+封装函数+定义MLUOps接口** 3. **注册算子：在cwise\_op\_power\_difference.cc中添加DLP算子的注册信息即可** 4. **接口定义：定义MLULib、MLUOp、MLUStream层接口**   **2.在线推理和离线推理有什么区别？**  区别在于离线部署脱离了TensorFlow编程框架和高性能库CNML,仅与运行时库CNRT相关，减少了不必要的开销，提升了执行效率 | | |