北京航空航天大学计算机学院

本科生毕业设计（论文）开题报告

论文题目：基于TVM的异构计算资源平台的设计与实现

学生姓名： 曲硕

学生学号： 17373482

专 业： 计算机科学与技术

指导教师： 李莹

学院(系)： 计算机学院

北京航空航天大学计算机学院

**2021年 2月 28日**

**填表说明**

1. 根据《北京航空航天大学关于本科生毕业设计(论文)手册》要求，每位学生必须认真撰写《本科生毕业设计（论文）开题报告》。
2. 每位学生应在指导教师的指导下认真、实事求是地填写各项内容。文字表达要明确、严谨，语句通顺，条理清晰。外来语要同时用原文和中文表达，第一次出现的缩写词，须注出全称。
3. 开题报告中引用的参考文献阅读数量不少于15篇，其中外文资料应不少于总篇数的30%。参考文献的书写请参照《北京航空航天大学关于本科生毕业设计(论文)手册》。
4. 本科生毕业设计（论文）开题报告总字数不少于3000字。
5. 请用宋体小四号字体填写，并用A4纸打印，于左侧装订成册。
6. 该表填写完毕后，须请指导教师审核，并签署意见。
7. 《北京航空航天大学计算机学院本科生毕业设计（论文）开题报告》将作为答辩资格审查的主要材料之一，占5分。
8. 本表格不够可自行扩页。

**本科生毕业设计(论文)开题报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学生姓名 | 曲硕 | 学号 | 17373482 | 专业班级 | 170611 |
| 指导教师 | 李莹 | 职称 | 副教授 | 单 位 | 计算机教学实验中心 |
| 题目类型 | □理论研究 应用研究 □综述 □其它  如选“其它”，请自定义题目类型 | | | | |
| 题目来源 | 校内 □校外 | | | | |
| 毕业设计（论文）题目 | 基于TVM的异构计算资源平台的设计与实现 | | | | |
| 开题报告（阐述课题的目的、意义、研究现状、研究内容、研究方案、进度安排、预期结果、参考文献等）   1. **研究背景和意义**    1. 研究目的   实现一个用户友好的，易用的平台，使得用户能够通过该平台把神经网络模型部署到不同的硬件设备上。而不需要了解底层具体的编译，部署流程。简化模型部署的难度，提高效率。   * 1. 研究背景   近些年来，以神经网络[1]为工具的深度学习在众多领域都取得了重大的成功，如图像的识别，自然语言的处理，以及在一些策略游戏上战胜了人类。随着神经网络模型的研究深入和训练速度的加快，产生了大量的模型。传统的模型部署的方式是把模型部署在服务器上，应用通过网络来使用模型。但是，把模型部署在移动设备上可以降低网络延迟，降低网络负载，同时可以提高用户隐私。基于这些优点，把神经网络模型部署在移动设备上的需求越来越多。但是真正把模型落地，面临着一个难以解决的问题。目前存在着众多的深度学习的框架，如Pytorch[3]，Tensorflow[2]，Caffe[5]等，得到的模型结构有所区别，同时，众多设备类型如CPUs，GPUs，FPGAs，TPUs[6]，在指令架构上有所不同。所以，需要一个统一的方式把众多的模型结构部署到众多的设备类型上。  同时，神经网络模型需要占用庞大的计算资源也是一个不可忽视的问题，而在许多的设备上计算资源比较匮乏，如移动手机和一些嵌入式设备。所以在部署过程中还需要对模型进行优化。目前的模型优化大多基于深度学习框架在计算图上的优化，这样的优化层级太高，不能处理和具体硬件相关的算子层面上的优化。其次，深度学习框架大多只针对了服务器端的GPU进行了优化，而没有针对每种不同的硬件设备进行优化。  所以，研究人员希望能够通过编译器来解决这个问题，通过编译的技术把不同深度学习框架的模型结构部署到不同架构的异构设备上，同时处理高层面的计算图优化和面向具体设备的优化。目前，随着深度学习的不断发展，神经网络编译器[11]得到了越来越多的关注和重大的发展。   * 1. 研究意义   首先，神经网络模型的功能日益丰富，多种硬件设备对模型的需求日益增多。传统的模型部署到异构计算设备往往需要很多的人工的调整，浪费了大量的人力。所以，提供一个统一的，易于使用的异构计算平台的部署，同时保持和人工调整相近或更优的准确度，能够节省大量的人力。  其次，TVM[7]是一个神经网络模型的编译器，能够实现不同的深度学习框架模型到不同硬件设备的部署同时执行计算图和面向硬件的优化。所以基于TVM来实现异构计算资源的平台能够更好的支持多种深度学习框架和众多的硬件设备，同时使得部署的模型效率更高。但是，TVM的使用涉及到复杂的环境部署，对于多种框架和多种硬件设备的使用复杂。所以，在TVM基础上封装异构计算资源的平台能够大大简化模型编译，部署的流程。  最终，通过封装的平台，能够使得用户更简洁，高效的实现神经网络模型到不同设备的部署。     1. **研究现状**   2.1 神经网络模型的部署  神经网络模型的部署往往都是与训练该模型的框架耦合在一起的，需要深度学习框架实现对应硬件平台的runtime。如Tensorflow Lite可以把Tensorflow的模型部署在Android设备上。Pytorch目前也推出了自己的模型部署到移动设备的runtime，但是这些解决方案只支持本身框架训练出的模型，且只支持移动设备。同时这些方式缺少针对具体硬件的优化，还需要人工对模型进行调整。  2.2 神经网络模型编译器  人工调整模型并不高效，所以产生了神经网络模型的编译器，自动化的生产可以直接在特定平台上运行的网络模型。如，GLOW[4]，TVM等。  为了同时支持众多的深度学习框架和广泛的硬件平台，目前主流的神经网络编译器都包含两部分，编译前端和编译后端[14]。同时包含两个中间表示，High-Level IR，Low-Level IR。编译前端能够处理多种主流的深度学习框架，如Tensorflow，Pytorch，MXNet等。编译前端把不同框架的不同模型结构转化为统一的计算图表示，High-Level IR。同时针对计算图进行优化，如计算图改写，算子融合等。编译后端把High-Level IR转换为Low-Level IR，并进行面向具体硬件的优化，如硬件元语对应，内存分配，并行计算等。  2.3 TVM  TVM是一个端到端的神经网络模型的编译器工具链，支持目前主流的前端的深度学习框架，如Pytorch，Tensorflow，Caffe等，同时支持部署到广泛的后端硬件设备，如CPU，服务器端GPU，移动端GPU，FPGA[12]等。在编译过程中TVM同时对模型进行优化，分别进行高层次的计算图优化和低层次的硬件相关的算子优化来保证部署到硬件设备上的模型的效率。    图 1 TVM系统结构  TVM的具体执行流程为，对于不同深度学习框架训练得到的模型转化为统一的计算图表示，然后对该计算图进行数据流的重写，得到一个优化的计算图。之后对该计算图进行算子层面的优化，对该计算图进行算子的合并，同时针对不同不同硬件设备的内存和指令结构进行具体的优化。   1. **研究内容** 2. 绪论 3. 研究背景和意义 4. 国内外研究现状 5. TVM的研究 6. TVM的总体架构 7. TVM编译的具体流程和实现细节 8. TVM部署的具体流程和实现细节 9. 异构计算平台的设计 10. 平台的总体结构和接口设计 11. 平台编译部署的使用流程 12. 异构计算平台的测试 13. 测试的环境 14. 异构计算平台和TVM使用流程的对比 15. 多硬件环境部署模型准确度的对比   第五章 结论  **四、研究方案和技术路线**  4.1 研究方案  研究方案总体依赖TVM的官方文档和相关论文，了解TVM内部实现原理和实现细节。  具体对TVM对每种深度学习框架的编译和每种硬件后端的部署进行具体实践，记录其过程。对于其中发现的问题依赖TVM的讨论区和搜索来解决。着重注意TVM在编译每种框架模型时需要手动调整的部分，通过实现额外的辅助代码来实现这部分的自动化。  4.2 技术路线  本课题的技术路线总体围绕TVM来展开，对TVM编译，部署的流程进行封装，并解决其中的一些问题，实现更高的自动化使用。  首先，在具体设备上，具体实践TVM对不同框架的编译流程，对不同硬件的部署流程。记录其过程。  其次，对编译和部署的流程进行封装，实现更加简洁的接口，让用户更加方便的使用。并且，选用目前各个深度学习领域主流的模型，如ResNet[15]，MoblieNet[14]等，对封装的接口进行测试，保证功能的正确性,同时验证模型的准确度。  最后，采用WEB实现用户良好的使用界面。  **五、关键技术或难点**  1. 多种框架模型结构的理解。该平台和TVM需要支持目前的多种主流的深度学习框架，所以需要对多种模型结构进行了解。  2. TVM框架的使用和源码的了解。该平台需要对TVM进行封装，所以需要对TVM的架构非常熟悉，对源码有一定的了解，能够对源码进行额外的功能补充和细节的更改。  3. 多种硬件设备环境的使用和计算体系的掌握。由于TVM支持多种硬件设备，如CPU，Serve GPU，Mobile GPU等，同时需要对Android[10]，树梅派等设备进行部署，所以需要对这些硬件和环境有充分的理解。  4. 多种语言的混合使用，由于TVM的地层细节是采用C++进行编写，前端采用python，同时还需要支持Cuda等计算平台，所以多种语言的混合使用和编译也需要很好的掌握。  **六、预期结果**  本课题的目标是对TVM的编译部署流程进行封装，支持多种深度学习框架和多种硬件平台。最终，基于TVM的异构计算资源平台应该提供统一，简洁，易用的接口，应当至少支持目前主流的深度学习框架，Pytorch，Tensorflow，Caffe，MXNet[13]等，至少支持Android，树莓派等设备。同时，保证基于平台接口部署的模型准确度同基于TVM部署的准确度没有差别。  **七、进度安排**  2020年12月~2021年1月：阅读相关学术论文，深入国内外相关领域的研究。  2021年2月：完成系统设计方案；  2021年3月~ 2016年4月：初步实现系统设计  2021年5月：测试和完善系统；撰写毕业论文。  2021年6月：总结毕业设计，完成最终答辩。  **八、参考文献**   1. Schmidhuber, J. (2015). Deep learning in neural networks: An overview. Neural networks, 61, 85-117. 2. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Zheng, X. (2016). Tensorflow: A system for large-scale machine learning. In 12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16) (pp. 265-283). 3. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. arXiv preprint arXiv:1912.01703. 4. Rotem, N., Fix, J., Abdulrasool, S., Catron, G., Deng, S., Dzhabarov, R., ... & Wang, M. (2018). Glow: Graph lowering compiler techniques for neural networks. arXiv preprint arXiv:1805.00907. 5. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., ... & Darrell, T. (2014, November). Caffe: Convolutional architecture for fast feature embedding. In Proceedings of the 22nd ACM international conference on Multimedia (pp. 675-678). 6. Jouppi, N. P., Young, C., Patil, N., Patterson, D., Agrawal, G., Bajwa, R., ... & Yoon, D. H. (2017, June). In-datacenter performance analysis of a tensor processing unit. In Proceedings of the 44th annual international symposium on computer architecture (pp. 1-12). 7. Chen, T., Moreau, T., Jiang, Z., Shen, H., Yan, E. Q., Wang, L., ... & Krishnamurthy, A. (2018). TVM: end-to-end optimization stack for deep learning. arXiv preprint arXiv:1802.04799, 11, 20. 8. Ignatov, A., Timofte, R., Chou, W., Wang, K., Wu, M., Hartley, T., & Van Gool, L. (2018). Ai benchmark: Running deep neural networks on android smartphones. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops (pp. 0-0). 9. Li, Mingzhen, et al. "The deep learning compiler: A comprehensive survey." IEEE Transactions on Parallel and Distributed Systems 32.3 (2020): 708-727. 10. Zhao, R., Ng, H. C., Luk, W., & Niu, X. (2018, August). Towards efficient convolutional neural network for domain-specific applications on FPGA. In 2018 28th International Conference on Field Programmable Logic and Applications (FPL) (pp. 147-1477). IEEE. 11. Chen, T., Li, M., Li, Y., Lin, M., Wang, N., Wang, M., ... & Zhang, Z. (2015). Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. arXiv preprint arXiv:1512.01274. 12. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861. 13. Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2017, February). Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 31, No. 1). 14. Li, M., Liu, Y., Liu, X., Sun, Q., You, X., Yang, H., ... & Qian, D. (2020). The deep learning compiler: A comprehensive survey. IEEE Transactions on Parallel and Distributed Systems, 32(3), 708-727. 15. Shawahna, A., Sait, S. M., & El-Maleh, A. (2018). FPGA-based accelerators of deep learning networks for learning and classification: A review. IEEE Access, 7, 7823-7859. | | | | | |
|  | | | | | |

|  |
| --- |
| **指导教师意见（课题难度是否适中、工作量是否饱满、进度安排是否合理、工作条件是否具备、是否同意开题等）：**  指导教师签名：  年 月 日 |
| **学院（系）意见：**  审查结果： □ 同 意 □ 不 同 意  学院（系）负责人签名：  年 月 日 |