

**本科生毕业设计（论文）开题报告**

题　　目：面向推荐系统训练的纠删码高效容错方案研究

|  |  |
| --- | --- |
| 院 系 | 计算机科学与技术 |
| 专业班级 | CS2103 |
| 姓 名 | 余正浩 |
| 学 号 | U202115404 |
| 指导教师 | 胡燏翀 |

2025年2月

**开题报告填写要求**

1. 开题报告主要内容：
   * + 1. 课题来源、目的及意义；
       2. 国内外研究现况及发展趋势；
       3. 课题研究的内容和技术方案；
       4. 可行性与风险分析；
       5. 课题研究进度安排；
       6. 主要参考文献。
2. 报告内容用小四号宋体字编辑，采用A4号纸双面打印，封面与封底采用浅蓝色封面纸（卡纸）打印。要求内容明确，语句通顺。
3. 指导教师评语、教研室（系、所）或开题报告答辩小组审核意见用蓝、黑钢笔手写或小四号宋体字编辑，签名必须手写。
4. 理、工、医类要求字数在3000字左右，文、管类要求字数在2000 字左右。
5. 开题报告应在第八学期第二周之前完成。

# 课题来源、目的及意义

## 课题来源

随着互联网技术的飞速发展和互联网业务的快速扩张，推荐系统已成为众多平台提供个性化服务的关键技术，其核心深度学习推荐模型（DLRM）训练规模不断扩大[6][7]。在实际生产环境中，DLRM 训练需要大量计算资源，且易受节点故障影响，严重影响了训练效率和模型准确性。传统的检查点机制在处理 DLRM 训练容错时存在诸多不足，而纠删码技术[5]为解决这些问题提供了新的思路。本课题来源于实际应用场景中对推荐模型分布式训练的高效性和可靠性的需求，旨在基于现有的纠删码检查点机制，对纠删码的编码进行改进，以实现推荐系统训练的高效容错。

## 课题目的及意义

DLRM 被广泛用于需要提供个性化内容服务的各种场景中，其训练效率和准确性直接影响其的性能。由于 DLRM 训练规模庞大，常分布在多个节点上[2][8]，节点故障发生频繁。在该场景下，传统的检查点机制[4][9]在容错时存在恢复速度慢、所需冗余空间多等问题，难以满足生产环境中 DLRM 频繁训练和部署的需求 [2]。

本课题旨在通过改进纠删码的编码方式实现推荐模型训练高效容错的方案，复现并改进现有的面向推荐系统的纠删码容错机制。其意义在于通过降低训练时间开销，提高故障恢复速度，减少内存和网络资源的消耗，为推荐系统在大规模生产环境下的稳定、高效运行提供有力支持，助力互联网企业提升推荐服务质量，增强用户体验和市场竞争力 。

另一方面，通过深入研究纠删码应用在DLRM 训练时的容错机制的方式方法，有助于我丰富和拓展相关理论知识，为今后的研究打下基础。

# 国内外研究现况及发展趋势

## DLRM 系统研究现状

基于深度学习的推荐系统（DLRM）近年来受到广泛关注。与DLRM训练相关的工作包括来自大型机构的系统[3][19]以及模型 - 系统协同设计[20][21]。这些系统在不同方面优化了 DLRM 的训练，但在容错方面仍关注较少，大多仍采用传统的检查点机制[4]。

## 检查点技术研究现状

检查点技术长期以来用于计算机系统容错[22][23][24]。在神经网络训练领域，近期的工作对神经网络训练中的检查点技术进行了优化[25][26]，但专住于 DLRM 训练的工作较少。部分工作开发了基于近似技术的方法以减少检查点开销，然而这些方法可能会降低 DLRM 训练的准确性。部分工作将纠删码应用于DLRM的检查点技术，但缺乏深入的研究。

## 纠删码技术研究现状

纠删码在各个领域被广泛用于容错、负载均衡和缓解性能下降[12][13][14][15][16][17][18]。近期的工作也将编码理论的思想应用于神经网络推理以及某些类别的模型训练中。但针对 DLRM 训练这一特定场景，纠删码的应用还面临诸多挑战，目前相关研究仍处于探索阶段，有待更深入的研究。

## 发展趋势

随着用户规模的增加， DLRM 规模的不断增大[10][11]，未来对其训练系统的容错能力要求将更高。研究将朝着在保证模型精度的前提下，进一步降低容错开销、提高恢复速度以更好地适应大规模模型训练的方向发展。将纠删码技术结合进入DLRM训练的容错机制，探索更高效的容错方案，以及深入探索纠删码技术在不同规模和结构的 DLRM 训练容错中的潜力，将成为未来的重要研究方向。

# 研究内容与技术方案

## 课题研究内容

深入研究现有的纠删码检查点机制在 DLRM 训练中的应用现状，包括其实现方式、优缺点以及与 DLRM 训练特性的交互影响。研究 DLRM 训练的独特属性，如嵌入表和神经网络参数在内存占用、更新频率、网络带宽消耗等方面的特点，分析这些特性对纠删码编码的影响。基于上述研究，对纠删码的编码方式进行改进。探索更合理的编码参数设置，如优化 k、r 值以平衡内存开销和容错能力；研究新的编码算法，提高编码和解码效率，减少对网络带宽和计算资源的消耗。设计并实现基于改进纠删码的 DLRM 训练容错系统，、确保系统在高效容错的同时，不影响 DLRM 训练的准确性。对改进后的系统进行全面性能评估，与传统检查点机制以及现有基于纠删码的系统（如 ECRec[1]）进行对比，分析在训练时间开销、恢复速度、吞吐量、内存使用等方面的性能提升情况。

## 课题预期目标

改进纠删码的编码方式，使其更适配 DLRM 训练的特点，降低训练时间开销。显著提高故障恢复速度，故障恢复速度比传统检查点机制快至少 8 倍，比现有基于纠删码的系统（如 ECRec[1]）在恢复速度上提升 10% - 20%。在恢复过程中，保持较高的训练吞吐量，吞吐量下降控制在 10% 以内。确保改进后的系统在提供高效容错能力的同时，不降低 DLRM 训练的准确性，与底层训练系统保持相同的精度。

## 技术路线与方案

### 关键技术

（1）纠删码优化技术：深入研究纠删码原理，通过实验和理论分析，确定适合 DLRM 训练的编码，使用新的编码算法，提高编码和解码的速度与效率。

（2）系统性能评估技术：采用多种评估指标，如训练时间开销、恢复速度、吞吐量等，对改进后的系统进行全面性能评估。通过与传统检查点机制和现有基于纠删码的系统对比，验证改进方案的有效性。

### 总体方案

基于现有的 DLRM 训练系统架构，融入改进后的纠删码编码方案。构建一个包含参数管理模块、冗余更新模块、故障恢复模块和一致性保障模块的容错系统。参数管理模块负责对 DLRM 的嵌入表和神经网络参数进行管理，根据优化后的混合冗余策略进行存储和调度。冗余更新模块利用改进的差异传播技术，及时准确地更新冗余参数。故障恢复模块在服务器故障时，通过优化后的纠删码解码算法进行快速恢复，同时利用粒度锁机制确保恢复过程与新的训练迭代互不干扰。一致性保障模块采用优化的两阶段提交协议，保证参数更新的一致性，避免在恢复过程中出现数据不一致问题。在实现过程中，充分考虑系统的可扩展性和兼容性，确保改进后的系统能够适应不同规模和结构的 DLRM 训练需求。

# 可行性与风险分析

## 制约因素与风险

技术实现难度方面，纠删码编码的改进涉及复杂的算法设计和系统适配，需要深入理解 DLRM 训练原理和纠删码技术，可能面临算法实现困难、系统兼容性问题等挑战。性能优化难度较大，在保证容错能力的同时，降低训练时间开销和提高吞吐量需要精细的系统调优，可能无法达到预期的性能提升目标。此外，系统的稳定性和可靠性也是需要关注的问题，改进后的系统可能引入新的故障点或不稳定因素。从外部环境影响来看，系统的运行可能对数据中心的资源分配产生影响，增加服务器负载，若系统出现故障可能影响推荐系统的正常运行，对用户体验和企业业务造成不利影响。

## 成本估算与可行性

采用 COCOMO 模型进行软件开发成本估算。选择该模型是因为它综合考虑了软件开发项目的类型、规模等多种因素，适用于本课题的开发场景。本课题的 DLRM 训练容错系统开发可视为半独立型项目，规模和复杂度属于中等。假设源指令条数（DSI）预估为 4000 条（根据系统功能和实现复杂度大致估算）。根据 COCOMO 模型中半独立型项目的估算公式：工作量 MM = 3.0\*(KDSI)^1.12 ，开发进度 TDEV = 10.5 (MM)^0.35 。将 KDSI = 4（4000 条指令换算为 KDSI）代入工作量公式可得：MM = 3.0\*(4)^1.12 ≈ 3.0 \* 4.72 ≈ 14.17（人月）。再将 MM = 14.17 代入进度公式可得：TDEV = 10.5\*(14.17)^0.35 ≈ 10.5 \* 2.53 ≈ 26.56（月）。虽然开发成本较高，但考虑到本课题研究成果对提升推荐系统训练效率和可靠性的重要意义，以及未来在互联网企业中的广泛应用前景，项目整体具有可行性。针对可能面临的风险，可以通过加强技术研究、进行充分的实验和测试、优化系统设计等措施来降低风险。

# 课题研究进度安排

由于本课题是一个研究性的课题，构思和实验的进度难以确定。大致的进度安排如表1所示。

表1 课题研究进度安排表

|  |  |
| --- | --- |
| **月份** | **工作任务** |
| 2025年1月  ~2025年2月 | 接受课题，搜索并阅读相关文献 |
| 准备开发测试环境 |
| 2025年2月  ~2025年4月 | 完成翻译和开题报告 |
| 复现论文算法，阅读代码和结果并从中获得启发 |
| 2025年5月 | 编写代码反复实验验证效果，撰写论文 |
| 2025年6月 | 毕业答辩 |

# 主要参考文献

1. Tianyu Zhang, Kaige Liu, Jack Kosaian, Juncheng Yang, and Rashmi Vinayak. 2023. Efficient Fault Tolerance for Recommendation Model Training via Erasure Coding. Proc. VLDB Endow. 16, 11 (July 2023), 3137–3150. https://doi.org/10.14778/3611479.3611514
2. Bilge Acun, Matthew Murphy, Xiaodong Wang, Jade Nie, Carole-Jean Wu, and Kim Hazelwood. Understanding Training Efficiency of Deep Learning Recommendation Models at Scale. In 2021 IEEE International Symposium on High Performance Computer Architecture (HPCA 21), 2021.
3. Udit Gupta, Carole-Jean Wu, Xiaodong Wang, Maxim Naumov, Brandon Reagen, David Brooks, Bradford Cottel, Kim Hazelwood, Mark Hempstead, Bill Jia, et al. The Architectural Implications of Facebook’s DNN-based Personalized Recommendation. In 2020 IEEE International Symposium on High Performance Computer Architecture (HPCA 20), 2020.
4. Assaf Eisenman, Kiran Kumar Matam, Steven Ingram, Dheevatsa Mudigere, Raghuraman Krishnamoorthi, Krishnakumar Nair, Misha Smelyanskiy, and Murali Annavaram. Check-N-Run: A Checkpointing System for Training Deep Learning Recommendation Models. In 19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22), 2022.
5. 黄建忠,曹强,秦啸,谢长生.纠删码存储集群系统设计与优化[M].北京:科学出版社,2016.
6. Paul Covington, Jay Adams, and Emre Sargin. Deep Neural Networks for YouTube Recommendations. In Proceedings of the 10th ACM Conference on Recommender Systems, 2016.
7. Maxim Naumov, Dheevatsa Mudigere, Hao-Jun Michael Shi, Jianyu Huang, Narayanan Sundaraman, Jongsoo Park, Xiaodong Wang, Udit Gupta, Carole-Jean Wu, Alisson G Azzolini, et al. Deep Learning Recommendation Model for Personalization and Recommendation Systems. arXiv preprint arXiv:1906.00091, 2019.
8. Biye Jiang, Chao Deng, Huimin Yi, Zelin Hu, Guorui Zhou, Yang Zheng, Sui Huang, Xinyang Guo, Dongyue Wang, Yue Song, et al. XDL: An Industrial Deep Learning Framework for High-Dimensional Sparse Data. In Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data, 2019.
9. Kiwan Maeng, Shivam Bharuka, Isabel Gao, Mark C Jeffrey, Vikram Saraph, Bor-Yiing Su, Caroline Trippel, Jiyan Yang, Mike Rabbat, Brandon Lucia, et al. CPR: Understanding and Improving Failure Tolerant Training for Deep Learning Recommendation with Partial Recovery. In The Fourth Conference on Systems and Machine Learning (MLSys 21), 2021.
10. Rashish Tandon, Qi Lei, Alexandros G Dimakis, and Nikos Karampatziakis. Gradient Coding: Avoiding Stragglers in Distributed Learning. In International Conference on Machine Learning (ICML 17), 2017.
11. Juncheng Yang, Anirudh Sabnis, Daniel S. Berger, K. V. Rashmi, and Ramesh K. Sitaraman. C2DN: How to harness erasure codes at the edge for efficient content delivery. In 19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22), pages 1159–1177, Renton, WA, April 2022. USENIX Association.
12. Saurabh Kadekodi, Francisco Maturana, Suhas Jayaram Subramanya, Juncheng Yang, K. V. Rashmi, and Gregory R. Ganger. Pacemaker: Avoiding heart attacks in storage clusters with disk-adaptive redundancy. In Proceedings of the 14th USENIX Conference on Operating Systems Design and Implementation, OSDI’20, USA, 2020. USENIX Association.
13. Aurick Qiao, Bryon Aragam, Bingjing Zhang, and Eric Xing. Fault Tolerance in Iterative-Convergent Machine Learning. In International Conference on Machine Learning, pages 5220–5230, 2019.
14. K. V. Rashmi, Nihar B Shah, Dikang Gu, Hairong Kuang, Dhruba Borthakur, and Kannan Ramchandran. A Hitchhiker’s Guide to Fast and Efficient Data Reconstruction in Erasure-Coded Data Centers. In Proceedings of the 2014 ACM SIGCOMM Conference (SIGCOMM 14), 2014.
15. Mahesh Sathiamoorthy, Megasthenis Asteris, Dimitris Papailiopoulos, Alexandros G Dimakis, Ramkumar Vadali, Scott Chen, and Dhruba Borthakur. XORing Elephants: Novel Erasure Codes for Big Data. Proceedings of the VLDB Endowment, 6(5), 2013.
16. Mark Wilkening, Udit Gupta, Samuel Hsia, Caroline Trippel, Carole-Jean Wu, David Brooks, and Gu-Yeon Wei. RecSSD: Near Data Processing for Solid State Drive Based Recommendation Inference. In Proceedings of the Twenty-Sixth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 21), 2021.
17. Jie Amy Yang, Jianyu Huang, Jongsoo Park, Ping Tak Peter Tang, and Andrew Tulloch. Mixed-Precision Embedding Using a Cache. arXiv preprint arXiv:2010.11305, 2020.
18. Chunxing Yin, Bilge Acun, Xing Liu, and Carole-Jean Wu. TT-Rec: Tensor Train Compression for Deep Learning Recommendation Models. In The Fourth Conference on Systems and Machine Learning (MLSys 21), 2021.
19. Michael Lui, Yavuz Yetim, Özgür Özkan, Zhuoran Zhao, Shin-Yeh Tsai, Carole-Jean Wu, and Mark Hempstead. Understanding Capacity-Driven Scale-Out Neural Recommendation Inference. In 2021 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS 21), 2021.
20. Assaf Eisenman, Maxim Naumov, Darryl Gardner, Misha Smelyanskiy, Sergey Pupyrev, Kim Hazelwood, Asaf Cidon, and Sachin Katti. Bandana: Using Non-Volatile Memory for Storing Deep Learning Models. In The Second Conference on Systems and Machine Learning (SysML 19), 2019.
21. Udit Gupta, Samuel Hsia, Vikram Saraph, Xiaodong Wang, Brandon Reagen, GuYeon Wei, Hsien-Hsin S Lee, David Brooks, and Carole-Jean Wu. DeepRecSys: A System for Optimizing End-to-End At-Scale Neural Recommendation Inference. In 2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA 20), 2020.
22. John T Daly. A Higher Order Estimate of the Optimum Checkpoint Interval for Restart Dumps. Future Generation Computer Systems, 22(3):303–312, 2006.
23. Richard Koo and Sam Toueg. Checkpointing and Rollback-Recovery for Distributed Systems. IEEE Transactions on software engineering, (1):23–31, 1987.
24. Adam Moody, Greg Bronevetsky, Kathryn Mohror, and Bronis R De Supinski. Design, Modeling, and Evaluation of a Scalable Multi-level Checkpointing System. In Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis (SC 10), 2010.
25. Jayashree Mohan, Amar Phanishayee, and Vijay Chidambaram. CheckFreq: Frequent, Fine-Grained DNN Checkpointing. In 19th USENIX Conference on File and Storage Technologies (FAST 21), 2021.
26. David A. Patterson, Garth Gibson, and Randy H. Katz. A Case for Redundant Arrays of Inexpensive Disks (RAID). In Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD 88), 1988.

**华中科技大学本科生毕业设计（论文）开题报告评审表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **姓名** | 余正浩 | **学号** | U202115404 | **指导教师** | 胡燏翀 |
| **院（系）专业** | | 计算机科学与技术学院CS2103 | | | |
| **指导教师评语**   1. 学生前期表现情况。 2. 是否具备开始设计（论文）条件？是否同意开始设计（论文）？ 3. 不足及建议。 | | | | | |
| 评 分： 指导教师（签名）：  2025年2月 27 日 | | | | | |
| **教研室（系、所）或开题报告答辩小组审核意见** | | | | | |
| 教研室（系、所）或开题报告答辩小组负责人（签名）：    2025年2月27 日 | | | | | |