

大模型静默数据损坏的统计检验方法

——基于均值波动统计量

张琪¹, 张娜², 严晓东³

1. 山东大学, 山东 济南 250000

2. 阿尔伯塔大学, 加拿大 阿尔伯塔 T6G 2R3

3. 西安交通大学, 陕西 西安 710000

摘 要 : 本文探讨了深度学习技术在大模型训练中遇到的静默数据损坏 (SDC) 问题, SDC 由硬件故障引起, 难以被处理器内部错误报告机制捕获, 可能导致非预期训练结果。为了探索 SDC 检测方法, 本文选取了 Adam 优化器中的历史梯度值作为观测对象, 并运用三种方法: 大数定律阈值检验、概率 Cusum 检验和均值波动统计量检验进行对比分析。实验结果表明, 大数定律阈值检验法主要适用于数据错误较大的情况, 对微小变化不敏感; Cusum 检验法虽然检验精度明显提升, 但较均值波动统计量仍有差距, 且对数据结构有特定要求; 均值波动统计量通过累加历史统计信息, 既能精准捕捉微小错误, 同时也适用于不同数据类型, 检验效果好且通用性强。

关 键 词 : 硬件故障; 静默数据损坏; 变点检测; 双臂老虎机

Exploration of Silent Data Corruption Detection Methods in Large Models —— Based on Mean Fluctuation Statistics

Zhang Qi¹, Zhang Na², Yan Xiaodong³

1. Shandong University, Jinan, Shandong 250000

2. University of Alberta, Alberta, Canada T6G 2R3

3. Xi'an Jiaotong University, Xi'an, Shanxi 710000

Abstract : This paper explores the Silent Data Corruption (SDC) issue encountered in the training of large models with deep learning techniques. SDC is caused by hardware failures and is difficult to be captured by the internal error reporting mechanisms of the processor, potentially leading to unexpected training results. To explore the detection methods for SDC, this paper selects the historical gradient values in the Adam optimizer as the observation objects and conducts a comparative analysis using three methods: the law of large numbers threshold test, the probabilistic Cusum test, and the mean fluctuation statistic test. The experimental results indicate that the law of large numbers threshold test method is mainly applicable to situations where data errors are relatively large and is insensitive to minor changes. Although the probabilistic Cusum test method shows a significant improvement in test accuracy, there is still a gap compared with the mean fluctuation statistic test, and it also has specific requirements for the data structure. The mean fluctuation statistic test method can accurately capture tiny errors by skillfully accumulating historical statistical information. Meanwhile, it is suitable for different data types, achieving good test results and possessing strong versatility.

Keywords : hardware failure; silent data corruption; change point detection; two-armed bandit

一、背景

最近, 深度学习技术在自动驾驶和航空电子设备等安全关键领域的应用日益增多。相关市场对模型设定了极为严苛的安全性标准, 如自动驾驶汽车的 ISO 26262 功能安全标准要求系统能够抵御 99% 的故障。在非关键性应用中, 随着大模型时代的到来, 模型的训练和推理都需要消耗大量的时间和计算资源, 系统的鲁棒性和数据的准确性对削减训练成本至关重要。

然而, 近几年由 Google^[1]、Facebook^[2] 等大型数据中心报告的硬件故障数量不断上升, 对神经网络的鲁棒性和安全性带来了

严峻的挑战。硬件故障的类型多种多样, 包括瞬态故障、间歇故障、延迟故障、永久性故障等。这些故障的产生可能源于多种因素, 诸如制造环节的缺陷、电路的自然退化、电压的波动变化、外部环境条件的影响、软错误的出现以及动态变化等复杂成因。

硬件故障中的静默数据损坏 (Silent Data Corruption, 记作 SDC) 是在大规模运行的应用程序中切实存在的现象, 它并不罕见无法忽视。静默数据损坏具有较强的隐匿性, 无法被处理器 (CPU 或 GPU 等) 内部的错误报告机制捕获, 难以在第一时间被发现, 随着系统持续运行, 其影响可以在多个服务之间传播, 直至应用程序调用相关数据时才会使问题得以暴露。这种故障会

导致数据丢失或错误等非预期训练结果，并且需要数月的调试时间。若能及时检测出系统中静默数据损坏的情况，既能降低系统应用过程中出现数据错误的概率，提高训练精度，还能有效避免在调试系统问题时所产生的额外成本与资源开销。

本文聚焦于致使非预期训练结果出现的静默数据损坏现象，选取 Adam 优化器中的历史梯度值作为观测对象，运用三种方法展开对比分析，以统计方法为依托，对静默数据损坏（SDC）检测问题的解决进行初步的探索与尝试。后续章节安排如下：第二章阐述现有相关工作；第三章设计实验并引入方法予以运行；第四章给出实验结论；第五章提出对未来工作思路及研究方向。

二、相关工作

（一）静默数据损坏检测相关工作

1. 硬件层面

错误校验码（ECC）和奇偶校验（parity）是两种常用于检测和纠正数据传输或存储中错误的方法。在设计触发器和小型 SRAM 缓存时，如果广泛地使用 ECC 或奇偶校验，虽然可以提高数据的可靠性，但这样做会消耗更多的芯片面积，并且 ECC/奇偶校验对发生在计算单元等非存储原件的故障仍然无法检测。

此外，防止应用程序级故障的一个更好的方法是采用硬件和软件级冗余技术，并在多个检查点上定期验证计算数据的准确性。此方法在空间研究、航空航天及汽车工程领域已历经实践检验。然而，该方法会增加面积与功耗成本，对于资源受限的系统而言，成本过高。

2. 算法层面

Siva^[3]等人以 CNN 网络为例提出了 ABED 技术，该技术使用校验和来检测卷积网络中的数据错误，并提供了三种检验方法：FC、IC、FIC，分别代表单独对过滤器使用校验和、单独对输入特征图使用校验和以及同时对过滤器和输入特征图使用校验和，其中 IC、FIC 方法还可以在模型推理过程中开展在线检测。尽管 ABED 技术具有高精度和强覆盖性，但是无法化解所有由硬件故障引发的意外训练结果，ABED 技术只能针对线性运算进行检验，非线性运算不受算法保护，例如激活和池化。此外，ABED 技术对每个卷积的过滤器/输入和输出都要生成校验和，这会带来高昂的开销。

Yi He^[1]等学者通过对硬件失效机理进行分析，将硬件错误抽象成寄存器错误（RIL），具体表现为模型训练中的神经元误差和权值误差。例如，优化器中较大的绝对梯度历史值或归一化层中较大的绝对移动方差值，是硬件故障产生意外训练结果的必要条件。文章中根据 DNN 的特性进行数学推导，借助大数定律确定了绝对历史梯度和绝对移动方差两个统计量的边界值。表 1 为 Adam 优化器的计算方法，在第三章大数定律阈值检验中，复现了文章中的方法对 Adam 优化器中的绝对梯度历史值 m_t 进行了检验，结果显示该方法只能检测出统计量出现较大误差的情形，对于微小误差难以奏效，存在检测盲区。

表 1 Adam 优化器计算方法

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t, \quad v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

$$u_t = \eta \frac{\frac{m_t}{1 - \beta_1^t}}{\sqrt{\frac{v_t}{1 - \beta_2^t} + \epsilon}}, \quad w_t = w_{t-1} - u_t$$

g_t : 迭代中的梯度值; β_1, β_2 : 衰减因子; m_t : 梯度历史值

v_t : 梯度平方的历史值; u_t : 用于更新权重的值

w_t : 权重值; η : 学习率; ϵ : 为了数值稳定性给定的较小数

（二）变点检测相关工作

变点检测旨在识别序列数据内某一统计特性（诸如分布类型、分布参数等）因系统性因素所引发的突变，并对这些变点的位置予以估计。作为一种在线或离线的信号处理工具，其在过程控制、脑电图分析、DNA 分割、计量经济学以及疾病人口统计学等诸多应用领域均展现出显著的有效性。

Cusum 方法是一种经典的变点检测算法，该算法以观察序列作为输入数据，对输入数据的偏差进行累积，借助平均运行长度（ARL）来确定阈值，一旦累积和超出阈值便发出警报。由此可见，整个过程呈现为一种“在线”算法特性，能够应用于实时数据流的处理。然而，如何合理地设置变化点阈值颇具挑战性。若阈值设置过于宽松，则容易遗漏变点；反之，若阈值设置过窄，则极易引发频繁的误报现象。针对这一问题，在第三章引入了 Cusum 的概率版本，旨在借助统计分布特性精准判别警报触发条件，进而达成对静默数据损坏情形的实时监测。

三、静默数据损坏分析与检验

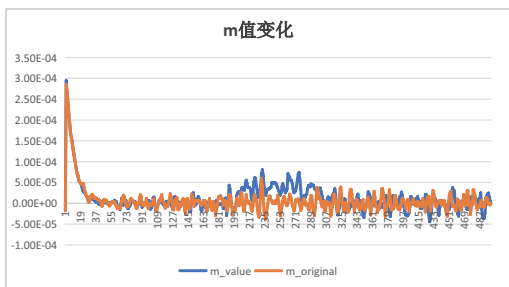
为了能够及时且有效地检测出静默数据损坏，并精准监测数据在生成过程中是否存在显著的变化，本章选取 Adam 优化器中的梯度历史值 m_t 作为检验目标，采用了三种不同的方法展开深入的实验对比研究，旨在从统计方向上对解决静默数据损坏（SDC）问题进行初步的探索与尝试。

（一）实验设计

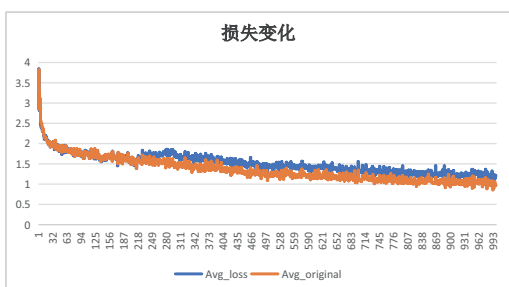
深度学习模型的运行主要涵盖两个关键阶段，即首先利用迭代学习过程开展训练的训练阶段，随后依据已训练完毕的模型将其部署至实际应用场景的推理阶段。在训练阶段，其迭代特性使其内置了验证步骤，能够借助测试数据集周期性地对模型的准确性予以校验。相比之下，在推理阶段，通常不会针对输出执行验证操作。这一差异直接导致在推理过程中，一旦系统发生故障，其负面影响会即刻在输出结果中得以直观呈现，便于观测与察觉。然而，在训练过程中若系统出现故障，只要给予足够长的训练时长，除非产生无穷大（INFs）或非数字（NaNs）等特殊情况，训练过程往往具备从故障影响中自行恢复的能力，这使得故障的发生及其影响难以被轻易察觉。

本研究聚焦于因硬件故障而引发潜在非预期结果出现的特定情形，即错误发生时不易察觉但是对系统结果产生影响的情况。依据 Yi He^[1]等学者观察可知，非预期结果产生的必要前提是故障的影响能够在多次训练迭代过程中持续存在；否则，训练过程

极有可能会恢复。基于此，与文献保持一致，本实验将检验目标锁定为 Adam 优化器中的梯度历史值 m_t ，它能够在不同的训练迭代之间有效承载故障所产生的影响。为了模拟故障持续影响的情况，实验选择将错误值持续注射在梯度值上，使其影响在多次训练迭代中持续存在，并且会传播至检验目标梯度历史值 m_t 。



> 图 1 注入错误后统计量变化



> 图 2 注入错误后损失变化

本实验构建了 Resnet-18 网络架构，针对 Cifar-10 数据集执行分类任务，选取 softmax 交叉熵作为损失函数，并运用 Adam 优化器进行模型训练。在第 200 至 300 轮迭代期间，持续向梯度引入微小偏差（具体数值为 0.00005），实验结果如图 1 和图 2，训练损失的下降态势出现一定程度的放缓，就单次训练过程而言，该变化较为隐匿，不易被察觉；然而，与无错误的训练进程相比，该错误会在后续的迭代进程中持续对模型的收敛速率施加影响，致使收敛速度减慢。与此同时，在无错误的训练过程中，统计量在 0 附近波动；而引入错误后，统计量的取值有所上升，并且随着错误的持续注入，其波动均值逐渐趋近于错误值附近，呈现出明显的偏离正常波动范围的趋势。

接下来本研究将采用三种不同的方法对该情景进行检测，目标在于能够精准且敏锐地捕捉数据在训练过程中所发生的异常变化，为后续深入研究静默数据损坏问题奠定基础。

（二）大数定律阈值检验

首先对 Yi He^[1]的方法进行了复现，原文是以 DNN 网络作为研究对象，并对网络做出了如下假设：

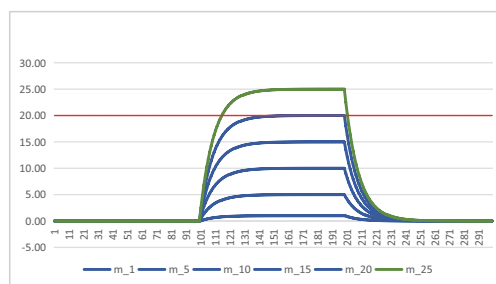
- 假设 DNN 各层输入与输出（激活函数前）均值为零，且所有层方差大致相等。
- 将所有输入数据集标准化为零均值、单位方差。
- 选用 softmax 交叉熵作为损失函数，Adam 作为优化器。
- 设定权重梯度值服从正态分布。

在以上假设下推导出统计量的分布为： $m_t \sim N(0, \frac{n_t}{m})$ ，根据切比雪夫大数定律 $P\left(\frac{1}{n} \sum_{i=1}^n X_i - E(X_i)\right) \geq \varepsilon \leq \frac{Var(X_i)}{n\varepsilon^2}$ ，获得统计量

m_t 的阈值为：

$$P(|m_t| > 20 \times \sqrt{\frac{n_t}{m^2}}) < 3 \times 10^{-89}$$

如图 3 所示，当向梯度注入数值大小为 1 的错误时（以橙色线条呈现），可以明显观察到统计量发生了显著变化。然而，即便如此，统计量仍未超出阈值范围。为了进一步探索该方法的检验效果，实验注入了不同大小的错误值，分别为 5、10、15、20、25。可以清晰地看出，只有当注入的错误值超过 20 时，该方法才能够成功检测出错误。这一现象充分表明，该方法在捕捉数据中出现的微小变化方面存在一定的局限性，只有在数据中出现较大错误时，该方法才能发挥作用。此外，通过大数定律确定的阈值 $20 \times \sqrt{\frac{n}{m^2}}$ 和 ε 的选择密切相关，阈值的大小决定着检验效果的好坏。



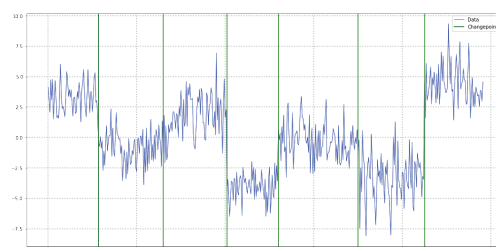
> 图 3 大数定律阈值检验结果

（三）概率 Cusum 检验

在序列分析中，传统的 Cusum 算法基于计算两个跟踪统计量 S_t^+, S_t^- ，累计序列远离目标均值 μ_0 的偏移，偏移通常表示为 $\delta\sigma$ ，其中 δ 是指定的常数。并确定上下阈值为： H^+, H^- ，当向上偏移很重要，则算法会在 $S_t^+ > H^+$ 时发出警报，当向下偏移很重要的时候，则算法会在 $S_t^- > H^-$ 时发出警报。

$$S_t^+ = \max\{0, S_{t-1}^+ + m_t - (\mu_0 + \delta\sigma)\}, S_0^+ = 0$$

$$S_t^- = \max\{0, S_{t-1}^- + (\mu_0 - \delta\sigma) - m_t\}, S_0^- = 0$$



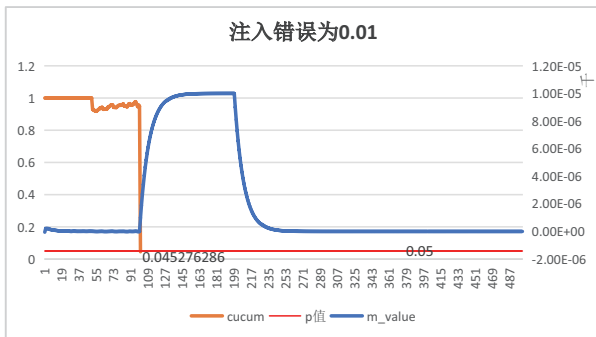
> 图 4 Cusum 算法示例数据

Cusum 算法适用于检测序列从一个平稳分布转变为另一个平稳分布的情况，图 4 就是一个典型的例子，序列在每对变化点之间是静止的。由此可见，本文中错误注射后的序列适用于该算法。然而，合理确定阈值颇具挑战，为此本文将传统的 Cusum 的方法进行轻微修改，使其可以直接进行概率比较。

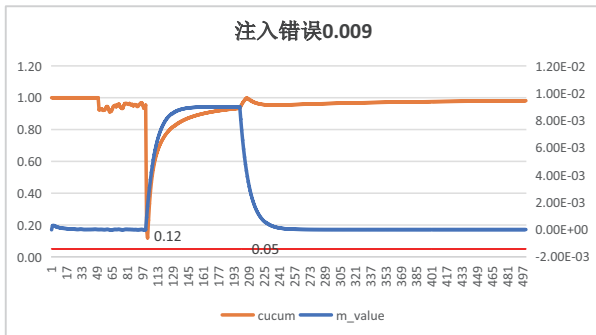
首先，将时间序列观测值进行标准化 $Z_t = \frac{m_t - \hat{\mu}_m}{\hat{\sigma}_m}$ ，序列的平均值和标准差使用以前的实现作为估计值。根据中心极限定理，标准化后的观测值累计和服从正态分布： $S_T = \sum_{t=1}^T Z_t \sim N(0, T) \Rightarrow$

$\tilde{S}_T = \frac{1}{\sqrt{T}} S_T \sim N(0,1)$ ，由标准正态分布的分布函数可知： $\Phi(\tilde{s}_T) \approx P(\tilde{S}_T \leq \tilde{s}_T)$ 等同于经典单边假设检验中 p 值，可以直接用于统计判断。

按照上述方法，计算统计量 m_t 前 50 个时刻的均值和方差作为样本估计量，自第 51 个时刻起计算 S_T 及其对应的 p 值进行检验，结果如图 5、6 所示。当注入错误率为 0.01 时，在注入错误后的首次迭代即第 101 次迭代时，检测出错误，此时 p 值小于 0.05 显著性水平。而当注入错误率降至 0.009 时，检验全程 p 值均大于 0.05，未能检测出错误。可见，当前方法虽有捕捉小错误的潜力，但检测存在盲区，对小于 0.01 的错误识别能力欠佳。此外，Cusum 算法对序列数据有特定要求，后续仍需深入探究该算法在不同情形下的应用效果。



> 图 5 Cusum 检验法 (1)



> 图 6 Cusum 检验法 (2)

(四) 均值波动统计量检验

本文引入 Chen^[4]等人提出的基于双臂老虎机过程的均值波动统计量，用于对变化的历史梯度值 m_t 进行在线检测。该统计量能够累计历史统计数据，从而有效捕捉观测数据的微小变化。同时，通过在更大的概率空间中对其渐近性质进行研究，提出了策略中心极限定理，确定了该统计量的极限分布，为假设检验构建了一个坚实的框架。

1. 提出原假设

假设在系统中观测到了一组梯度历史值： $m_0, m_1, m_2, m_3, \dots, m_t, \dots$ ， t 为迭代次数， μ_m 为总体期望。考虑原假设和备择假设如下：

$$H_0: \mu_m = 0; H_1: \mu_m > 0$$

2. 均值波动统计量

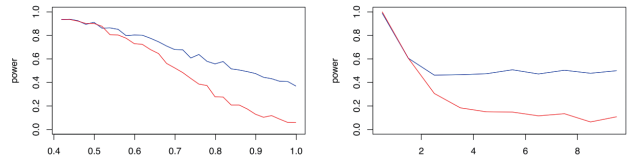
均值波动统计量 T_t 定义如下：

$$\hat{\sigma}_t T_t = \frac{1}{t} \sum_{i=1}^t m_i \text{sgn}(T_{i-1}) + \frac{1}{\sqrt{t}} \sum_{i=1}^t \frac{m_i \text{sgn}(T_{i-1})}{\hat{\sigma}_i}; T_0 = 0$$

其中， t 为迭代次数， $\text{sgn}(\cdot)$ 为符号函数， m_t 为观测值样本标

准差。

在线检验中，当前时刻 t 的统计量与此前的所有时刻相关，该统计量通过累积历史统计信息，为当前检验提供辅助，从而更易于捕捉序列中的微小错误，使得检验系统能够对异常变化产生更高的敏感度，进而有效提升了检验功率。此外，根据 Chen^[4]等人的研究成果可知（图 7），相较于经典正态检验统计量，该统计量具备更强的功效，能够取得更优的检验效果。



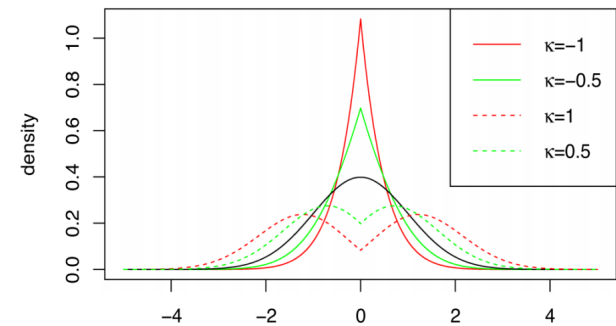
> 图 7 本文检验统计量（蓝线）和经典检验统计量（红线）的功率比较图

3. 检验统计量的渐进分布

在零假设和备选假设下， T_t 的极限分布具有一个定义明确的概率密度函数，其定义如下：

$$f^\kappa(y) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(y+\kappa)^2}{2} - \kappa e^{2\kappa|y|} \Phi(-|y|-\kappa)}$$

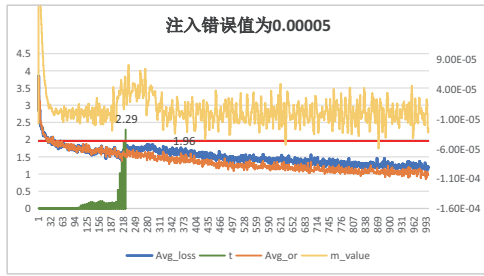
其中 κ 为参数，图 8 展示了不同 κ 值下的分布密度。从图中可以看出，在 $\kappa < 0$ 下，密度函数比经典的标准正态分布更集中，因此被称为“尖峰分布”； $\kappa > 0$ 时，密度函数呈现双峰分布，较标准正态分布更分散； $\kappa = 0$ 表示标准正态分布。均值波动统计量 T_t 的渐进分布在原假设下为正态分布，在备择假设下为双峰分布。



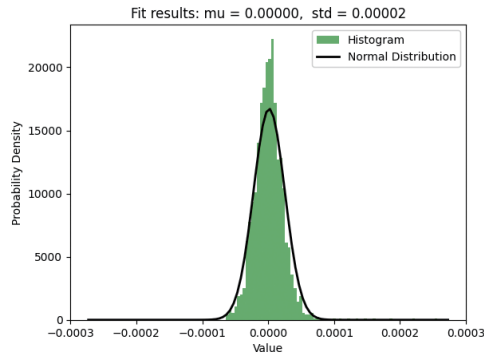
> 图 8 检验统计量 T_t 渐进分布密度图

4. 实验结果

采用上述 T 检验统计量，以每五次迭代为一个检验周期进行检验，直至首次错误出现时停止，实验结果如图 9。由于原假设下检验统计量服从正态分布，故将统计量同显著性水平 α 为 0.05 的检验值 $Z_{\alpha/2} = 1.96$ 进行比较。在模型训练过程中，于第 200 至 300 次迭代阶段持续引入微小错误，该错误的具体数值设定为 0.00005，使得模型的损失下降速率呈现出一定程度的减缓，同时使得统计量受到显著扰动而增大。由实验结果得知，在注入错误后的第 5 个检验节点处（即对应于第 225 次迭代时刻），成功检测出错误的存在，此时的检验统计量满足 $T > 1.96$ 。综上所述，T 检验统计量能够通过历史统计信息进行累积和分析，有效地实现对微小错误的精准检测。结合训练过程中 m_t 的分布直方图（图 10）可知，对于超出 3σ 范围的微小错误，均能够被 T 统计量敏锐地检测出来。此外，该方法在实际应用中对检验数据并未设定严苛的分布条件，展现出了较强的通用性和适应性。



> 图 9 T 统计量检测结果



> 图 10 m 值的分布直方图

四、结论

本文运用三种不同的检验方式，针对神经网络里数据出现持续错误的状况展开在线检验，以达成精确且迅速地检测出序列变化的目的，进而有效防范因硬件损坏而引发的静默数据损坏现象。

在研究的三种方法中，大数定律阈值检验法存在局限。它仅对数据较大错误敏感，难以捕捉微小序列变化，且阈值确定困难，合理性与科学性影响检验成效。变点检测的 Cusum 法较前者精度提

参考文献

[1] Ding and Mitigating Hardware Failures in Deep Learning Training Systems', in Proceedings of the 50th Annual International Symposium on Computer Architecture. ISCA' 23: 50th Annual International Symposium on Computer Architecture, Orlando FL USA: ACM, pp. 1 - 16.

[2] Dixit, H.D. et al. (2022) 'Detecting silent data corruptions in the wild' .

[3] Hari, S.K.S. et al. (2022) 'Making Convolutions Resilient Via Algorithm-Based Error Detection Techniques' , IEEE Transactions on Dependable and Secure Computing, 19(4), pp. 2546 - 2558.

[4] Chen, Z., Yan, X. and Zhang, G. (2023) 'Strategic two-sample test via the two-armed bandit process' , Journal of the Royal Statistical Society Series B: Statistical Methodology, 85(4), pp. 1271 - 1298.

[5] Chen, H. (2024) 'Single phase grounding fault location of low voltage distribution network based on artificial neural network' , in 2024 3rd International Conference on Energy, Power and Electrical Technology (ICEPET). 2024 3rd International Conference on Energy, Power and Electrical Technology (ICEPET), pp. 641 - 647.

[6] Amiri, H. (2023) 'Analysis and Comparison of Artificial Neural Network and Traveling Wave Algorithm for Distribution Networks Fault Location' , in 2023 5th International Conference on Optimizing Electrical Energy Consumption (OEEC). 2023 5th International Conference on Optimizing Electrical Energy Consumption (OEEC), pp. 1 - 5.

[7] Atoofian, E. (2024) 'Hardened-TC: A Low-cost Reliability Solution for CNNs Run by Modern GPUs' , in 2024 IEEE 37th International System-on-Chip Conference (SOCC). 2024 IEEE 37th International System-on-Chip Conference (SOCC), Dresden, Germany: IEEE, pp. 1 - 6.

[8] Dixit, H.D. et al. (2021) 'Silent Data Corruptions at Scale' .

[9] Gräfe, R. et al. (2023) 'Large-Scale Application of Fault Injection into PyTorch Models -an Extension to PyTorchFI for Validation Efficiency' , in 2023 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks - Supplemental Volume (DSN-S). 2023 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks - Supplemental Volume (DSN-S), pp. 56 - 62.

[10] He, Y., Balaprakash, P. and Li, Y. (2020) 'Fidelity: Efficient Resilience Analysis Framework for Deep Learning Accelerators' , in 2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO). 2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), pp. 270 - 281.

[11] He, Y. and Li, Y. (2023) 'Understanding Permanent Hardware Failures in Deep Learning Training Accelerator Systems' , in 2023 IEEE European Test Symposium (ETS). 2023 IEEE European Test Symposium (ETS), pp. 1 - 6.

[12] Huang, H. et al. (2024) 'MRFI: An Open-Source Multiresolution Fault Injection Framework for Neural Network Processing' , IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 32(7), pp. 1325 - 1335.

[13] Montes De Oca, V. et al. (2010) 'A cusum change-point detection algorithm for non-stationary sequences with application to data network surveillance' , Journal of Systems and Software, 83(7), pp. 1288 - 1297.

升，能更敏锐察觉数据波动，但与均值波动统计量检测相比仍有差距。同时，其对序列数据特性要求严苛，如数据分布、序列相关性等因素会干扰检测，限制了适用场景。本文引用的均值波动统计量检测法优势显著，具备检测微小错误的能力，在 3σ 范围外的错误均能检验出来，且无特殊使用条件限制，检验效果好且通用性强。

五、未来与展望

直接在硬件上进行错误捕捉实验会带来高昂的运行成本，为了推进 SDC 检测研究，应当深入研究硬件故障对深度学习训练的影响，通过模拟不同类型的硬件故障在软件层面的表现，理解硬件故障对系统数据的影响。Facebook 和 Google 等公司已对常见的硬件损伤进行了报告。根据 Yi He^[1] 等人研究，硬件故障在软件层面主要体现为三种错误类型：单周期 FF 位翻转、全局控制 FF 位翻转、数据路径 FF 位翻转。这些错误可能影响单个或多个周期内输入、输出、权重等不同位置的数据，进而导致数据产生 INFs/NaNs 以及不同的非预期结果。例如，单周期 FF 位翻转广泛用于研究动态变化、不稳定电路行为和软错误；全局控制 FF 位翻转影响多个数据路径寄存器，可能导致输出张量在多个周期内出现大量错误元素；数据路径 FF 位翻转中，代表高指数位的位翻转或与数据精度配置相关的控制 FF 位翻转可能产生 INFs/NaNs。

本文探索了硬件故障对数据产生的多周期影响情形，在后续研究中，可进一步拓展对单个周期数据错误情况的模拟，增加错误类型的多样性，以便更全面地涵盖实际可能出现的硬件故障情况。

此外，考虑到大型系统训练对算力和资源的巨大消耗，在进行相关实验和技术开发时，还需高度重视控制检验程序所产生的额外开销，确保解决方案在实际应用中的可行性和有效性。