

비트맵 데이터 학습을 통한 딥러닝 기반의 메모리 수리 예측 기술 연구

강 유 진^{1†} · 오 준 석^{2‡} · 서 은 솔¹ · 박 석 민^{3,4} · 이 영 우^{5,6*}¹전남대학교 컴퓨터정보통신공학과 학사²전남대학교 지능전자컴퓨터공학과 석사과정³인하대학교 전기컴퓨터공학과 석사과정⁴인하대학교 반도체융합전공 석사과정⁵인하대학교 전기컴퓨터공학과 조교수⁶인하대학교 반도체융합전공 조교수

Study of a Deep Learning-Based Early Termination Detection Scheme Using Bitmap Data Learning

Yujin Kang^{1†} · Junseok Oh^{2‡} · Eunsol Seo¹ · Seokmin Park^{3,4} · Young-woo Lee^{5,6*}¹Bachelor's Degree, Department of Computer Engineering, Chonnam National University, Gwangju 61186, Korea²Master's Course, Department of Intelligent Electronics and Computer Engineering, Chonnam National University, Gwangju 61186, Korea³Master's Course, School of Electrical and Computer Engineering, Inha University, Incheon 22212, Korea⁴Master's Course, Program in Semiconductor Convergence, Inha University, Incheon 22212, Korea⁵Assistant Professor, School of Electrical and Computer Engineering, Inha University, Incheon 22212, Korea⁶Assistant Professor, Program in Semiconductor Convergence, Inha University, Incheon 22212, Korea

[요 약]

반도체 메모리 기술의 지속적인 발전은 메모리 칩의 저장 용량을 크게 증가시켰지만, 동시에 제조 및 테스트 비용 증가, 소형화 및 고집적화로 인한 결함 발생 확률 증가와 같은 여러 문제를 초래했다. 이러한 문제를 효과적으로 해결하기 위해, 전기 다이 분류(EDS) 공정에서는 불량 셀을 예비 셀로 대체하는 RA(Redundancy Analysis) 알고리즘을 사용한다. 그러나 기존의 C 언어 기반 RA 프로세스는 칩의 복잡성이 증가함에 따라 테스트 시간이 길어지고 비용이 증가하는 단점이 있다. 본 논문에서는 이러한 한계를 극복하고자, 과거 수리 데이터를 기반으로 합성곱 신경망(CNN)을 활용해 실패 비트맵을 학습하는 딥러닝 기반 수리 예측 모델을 제시하고, RA 알고리즘의 딥러닝 적용 가능성을 확인하였다.

[Abstract]

The continuous advancement of semiconductor memory technology has significantly increased memory chip capacity but has also introduced new challenges, including higher manufacturing and testing costs and a greater likelihood of defects due to miniaturization and high integration. To address these challenges, the Electrical Die Sorting (EDS) process employs a Redundancy Analysis (RA) algorithm to replace faulty cells with spare ones. However, as chip complexity increases, the traditional RA process, based on C language, becomes less efficient, resulting in longer testing times and higher costs. This paper proposes a deep learning-based repair prediction model that utilizes convolutional neural networks (CNN) to learn from historical repair data and failure bitmaps, thereby improving the efficiency of the RA process. The study confirms the applicability of deep learning to RA algorithms, demonstrating its potential to overcome the limitations of traditional methods.

색인어 : 반도체 메모리, RA 알고리즘, 딥러닝, 합성곱 신경망(CNN), 전기적 다이 분류**Keyword** : Memory Devices, RA Algorithm, Deep Learning, Convolutional Neural Networks, Electrical Die Sorting<http://dx.doi.org/10.9728/dcs.2024.25.10.3053>

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 17 August 2024; **Revised** 23 September 2024**Accepted** 02 October 2024‡ **These authors and contributed equally to this work*****Corresponding Author; Young-woo Lee****Tel:** +82-32-860-7351**E-mail:** yw.lee@inha.ac.kr

I. 서론

반도체 메모리의 지속적인 발전은 메모리 칩의 저장 용량을 꾸준히 증가시켜 왔으며, 메모리 칩의 용량은 대체로 18개월마다 두 배로 증가해 왔다[1]. 이러한 회로의 소형화는 더욱 정밀한 제조 공정을 요구하게 되며, 이에 따른 제조 비용 및 테스트 비용이 급격히 상승하게 된다. VLSI 기술의 혁신은 칩에 더 많은 저장 셀을 통합함으로써 반도체 메모리의 고집적화와 대용량화를 이루었지만, 불량 발생 확률도 상대적으로 증가시키는 결과를 초래했다[2]. 메모리 용량의 급격한 증가는 결함 발생 빈도를 높이고, 중요한 요소인 메모리 생산 수율을 낮추는 문제를 야기한다[3]. 많은 결함이 발생할 경우 어쩔 수 없지만, 적은 결함이 발생한 경우 해당 다이를 버리기보다 수리하여 사용하는 것이 더 효율적이며, 원가 절감의 장점도 있다. 이에 따라 메모리 수리는 반도체 공정에서 중요한 역할을 차지하게 되었다[4].

메모리 테스트 공정에서는 정확한 분석을 위해 RA (Redundancy Analysis) 알고리즘을 사용한다[5]. RA 알고리즘은 불량 셀을 예비 메모리 셀로 대체하기 위해 필요한 정보를 얻는 데 사용되는 알고리즘이다[6]-[11]. EDS (Electrical Die Sort) 공정에서 테스트 시간과 RA 프로세스 시간은 제조 비용과 직결되므로, 테스트 시간을 단축하는 것이 제조 비용 절감에 매우 중요한 역할을 한다. 기존의 C언어 기반 RA 프로세스는 메모리 테스트 데이터 입력 시 순차적으로 불량을 파악하고 수리 방법을 제시하는데, 반도체의 크기가 커지고 알고리즘이 복잡해질수록 테스트 시간이 늘어난다는 단점이 있다. 또한, 과거의 수리 데이터 이력을 참고하지 않고 매번 새로운 데이터에 전체 수리 알고리즘을 적용해야 한다.

이에 반해, 딥러닝 기반 수리 예측 모델은 과거의 수리 데이터 이력을 학습하여 수리 방법을 제시하기 때문에 매번 알고리즘을 적용할 필요가 없어 RA 프로세스 시간을 단축할 수 있다. 반도체 산업의 막대한 비용을 고려할 때, 개발 기간, 양산 준비 기간, 출시 기간, 수익 창출 기간 등 시간과 관련된 사항은 매우 중요하다. 그리고 가장 중요한 변수는 수율 달성 시간이다. 새로운 공정의 연구개발부터 초기 생산, 대량 제조까지 소요되는 시간을 조금만 단축해도 반도체 제조사는 수십억 달러에 상응하는 가치를 창출할 수 있다. 본 논문은 메모리 테스트에서 발생하는 failure bitmap 데이터를 학습하여 수리 방법을 제시하는 딥러닝 기반의 메모리 수리 예측 기술을 제안함으로써 RA에 소요되는 시간을 절감하고 원가 경쟁력을 강화하고자 한다.

II. 기존 연구

2-1 EDS 공정

반도체 제조공정에서 웨이퍼 제조공정은 반도체 IC 칩을 생산하는 과정으로, 가장 핵심적이며 복잡한 공정이다. IC 칩

생산 공정 중에서 웨이퍼 제조(FAB)는 포토, 확산, 이온 주입, 박막, 식각, 연마 등의 공정들이 반복적으로 이루어진다. EDS는 반도체 제조공정 중 하나로, FAB 공정 진행 후 웨이퍼에 완성된 각각의 칩들이 설계한 대로 전기적 특성을 가지고 동작하는지 검사한다. 이를 바탕으로 각각의 다이를 Good/Bad로 분류하고, 수리가 가능한 칩에 대해서는 수리를 진행하는 공정이다.

그림 1은 EDS 공정을 통한 양품 선별의 예시를 나타낸 그림으로서, EDS 공정은 Electrical Test, Pre-laser Test, Laser Repair, Post-laser Test와 같은 순서로 진행된다. 그 중 Pre-laser Test는 Laser Repair 단계에서 사용될 Redundancy Cell Repair 정보를 생성하기 위한 단계이다. 한 번의 FAB 공정으로 메모리 셀들을 제대로 만들어내기 어렵고, 소수의 셀이 고장났다고 해도 한 다이를 모두 버리는 것은 효율적이지 않다. 이를 위해 칩 내에 고장난 셀을 대체할 여분의 셀을 설계한다. Pre-laser Test에서 각 셀의 고장 여부를 확인하고 그에 따라 사용될 여분 셀의 주소를 저장하면, Laser Repair 공정에서 레이저 빔을 이용해 수리를 진행한다. Post-laser Test에서는 전 단계에서 수리된 주소에 해당하는 셀이 정상적으로 동작하는지 확인하기 위한 테스트를 진행한다.

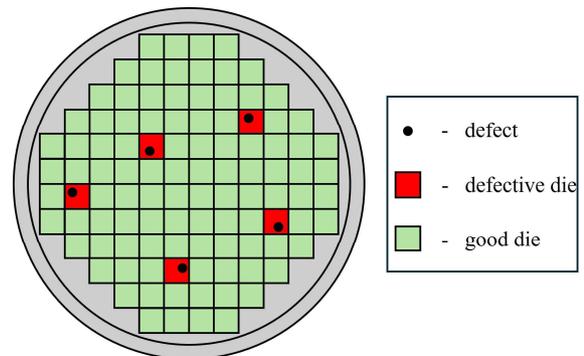


그림 1. EDS 공정을 통한 양품 선별의 예시
 Fig. 1. An example of selecting good products through the EDS process

이때 Pre-laser 테스트 완료 후 한 개의 메모리에서 고장 셀들의 위치를 한눈에 알아볼 수 있도록 나타낸 것을 비트맵(bitmap)이라고 부른다. 비트맵은 고장 셀의 위치를 논리적 주소로 표현한 논리 비트맵(logical bitmap)과 실제 메모리 구조 내 고장 셀의 위치를 표현한 물리 비트맵(physical bitmap)으로 분류되는데, 웨이퍼 테스트에서는 통상적으로 물리 비트맵을 이용한다. Pre-laser 테스트가 종료된 후 물리 비트맵을 분석하여 고장 셀들을 주어진 여분 셀로 대체 가능 여부를 확인하여야 한다. 이러한 분석 작업을 redundant cell analysis라 하며, 통상적으로 웨이퍼 테스트 장비에서는 제공되는 redundancy analysis algorithm을 이용하여 분석이 이루어진다.

2-2 메모리 수리

메모리 수리는 메모리 셀에 결함이 발생한 경우, 미리 내장 해놓은 여분의 메모리 셀과 교체하는 방법론이다[12]. RA(Repair Analysis)는 주어진 여분을 사용하여 메모리 수리 솔루션을 찾는 과정으로, 두 단계로 구성된다. 첫 번째 단계는 강제 수리 분석(forced-repair analysis)으로, 동일한 차원의 여분 셀로 교체해야 하는 특정 행 또는 열을 결정한다. 두 번째 단계는 희소 수리 분석(sparse-repair analysis)으로, 강제 수리 단계 이후 남은 결함에 대해 수리 솔루션을 결정하며, 이전에 강제 수리 이후 남은 여분 셀을 사용한다. RA를 통한 메모리 수리 덕분에 메모리 수율을 유지할 수 있지만, 결함 발생 확률이 증가함에 따라 수리 비용도 함께 증가했다. 일반적으로 수리 비용은 수리율과 수리 시간과 관련이 있으며, 수리율은 테스트된 전체 메모리 중 수리된 메모리의 비율로 정의할 수 있다. 수리 시간은 메모리 수리 솔루션을 찾는 데 필요한 총 시간으로 정의된다. 수리율과 수리 시간 사이에는 상충 관계가 있기 때문에 높은 수리율을 위해서는 많은 수리 시간이 소요된다. 이러한 이유 때문에 메모리 수리에 많은 비용이 발생하게 되었다. 또한, 반도체 제조 기술의 발전으로 인해 테스트 및 수리가 필요한 메모리의 수가 증가함에 따라 메모리 수리 비용이 전체 제조 비용 중 많은 비중을 차지하게 되었다. 기존의 RA 알고리즘들은 순차적으로 결함을 수리하는 방식으로, 메모리 집적도가 증가할수록 처리 시간이 늘어나며, 이는 제조 비용의 증가로 이어진다. 또한, 기존 방식은 과거 데이터를 활용하지 않고 매번 동일한 분석 과정을 거치기 때문에 효율성 면에서 한계가 존재한다. 본 연구에서는 디러닝 기법을 적용하여 기존 방식의 한계를 극복하고자 하였으며, 이를 통해 수리 시간을 단축하고 정확도를 향상시킬 수 있음을 확인하였다.

2-3 Repair-most algorithm

Repair-most 알고리즘은 대표적인 RA 알고리즘 중 하나로, 먼저 각 행과 열의 카운터에 결함의 개수를 저장한다. Repair-most 알고리즘은 그림 2와 같은 순서로 진행된다. 가장 먼저 행과 열의 결함 데이터를 초기화한다. ARC(available redundant columns)와 ARR(available redundant rows)는 여유 셀의 개수를 표시하는 카운터이며, repair-most 알고리즘은 가장 카운터 값이 높은 행과 열을 must-repair로 설정하여 우선적으로 수리하는 것이 특징이다. 따라서, 남아 있는 예비 열이나 예비 행이 있다면 가장 카운터가 높은 행 또는 열을 우선 수리하고, 이 과정을 여유 셀을 다 사용하거나 결함이 모두 수리될 때까지 반복한다. 만약 남아 있는 예비 행이나 열보다 결함의 수가 더 많다면, 수리할 수 없다고 판단한다. 이 알고리즘을 사용하여 수리를 진행할 시, physical bitmap을 읽어 들이는 단계부터 결함 수리 완료 단계까지 각각의 결함 데이터가 순차적으로 진행된다.

이는 100개의 결함 데이터 분석을 진행할 때, 100번의 repair-most 알고리즘이 실행된다는 뜻이다. Repair-most 알고리즘은 효율적이고 널리 사용되는 방식으로, 기존 메모리 수리에서 성능이 검증되어 디러닝 기법과의 비교를 위한 기준 알고리즘으로 선정하였다. 이를 통해 디러닝 방식의 개선 효과를 명확하게 평가할 수 있었다.

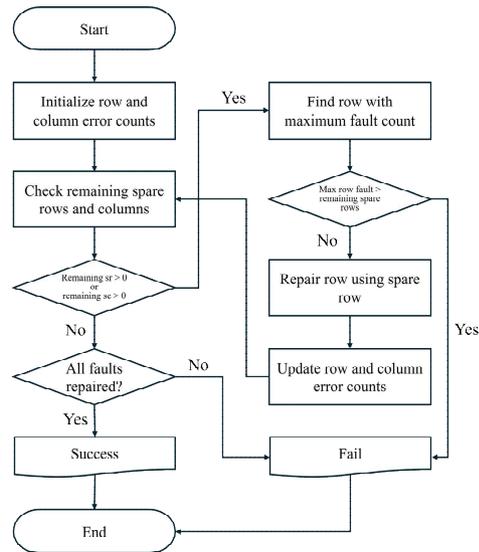


그림 2. Repair-most 알고리즘의 순서도
Fig. 2. Flowchart for the repair-most Algorithm

2-4 조기 종료(Early Termination)

메모리 수리에서 수리 속도는 수리 비용과 직결되므로 알고리즘을 구현할 때 평균 분석 시간을 최소화하는 것이 중요하다. 따라서, 수리 초기에 수리 가능성을 판단하는 것이 매우 중요하다. 조기 종료(early termination)는 메모리 수리 과정에서 수리할 수 없는 메모리를 조기에 식별하고, 더 이상의 자원과 시간을 낭비하지 않도록 수리 프로세스를 중단하는 방법론이다. 예를 들어, 특정 장치가 각각 여분 행과 여분 열을 가진 네 개의 영역으로 구성되어 있을 때, 각 영역에서 수리 전 발생할 수 있는 총 결함 수는 장치의 총 행(Total Rows, TR)과 총 열(Total Columns, TC)에 여분 행(Spare Rows, SR)과 여분 열(Spare Columns, SC)을 곱한 후, 중복된 결함 수를 제거하는 방식으로 계산된다. 이는 다음 식 (1)과 같다[13].

$$[(TR \times SR) + (TC \times SC)] - (SR \times SC) \quad (1)$$

강제 수리 분석 단계 후, 사용되지 않은 여분 행의 수를 “USR”, 사용되지 않은 여분 열의 수를 “USC”라고 하면, 강제 수리 이후 남은 여분 행과 여분 열이 처리할 수 있는 최대 결함 수는 다음 식 (2)와 같이 표현된다.

$$Max\ Defect = 2 \times USR \times USC \quad (2)$$

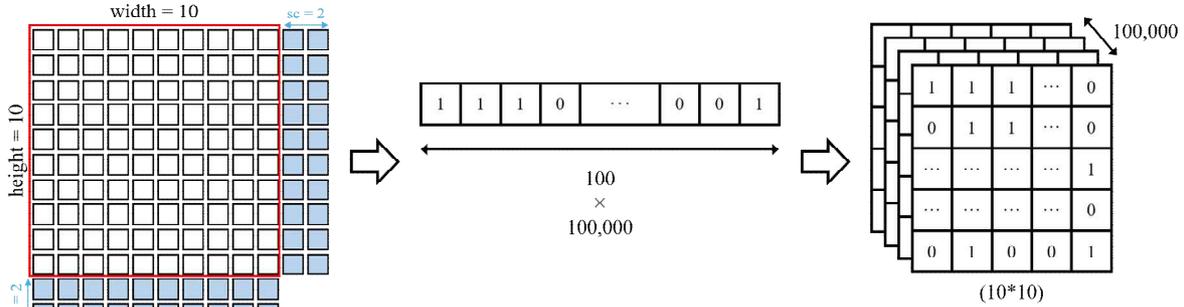


그림 3. 메모리 테스트 데이터 전처리 과정
 Fig. 3. Memory test data preprocessing procedure

이때, 2는 남은 여분 행과 여분 열이 결합하여 처리할 수 있는 두 가지 방향(행과 열)을 고려한 값이다. 따라서, 남은 오류 총 개수가 식 (2)의 값보다 작거나 같다면 수리가 가능하고, 더 크다면 수리가 불가능하다고 판별하게 된다. 본 연구에서는 남은 여분 행과 여분 열의 수보다 많은 수리가 필요할 경우 수리가 불가능하다고 판단하였다.

III. 본 론

3-1 Repair-most algorithm 구현

Repair-most 알고리즘 구현을 위해 반도체의 결함 정보를 나타내는 physical bitmap 데이터가 입력된다. 실제 반도체 회사의 양산 데이터를 사용할 수 없으므로, 실험 조건에 알맞은 failure bitmap을 생성하는 Random fault generator를 개발하여 사용하였다. 실제 반도체 양산 환경에서 결함이 특정 패턴을 띄며 나타나지 않는다는 점을 바탕으로 내장 함수를 이용하여 결함의 위치를 불특정하게 지정하였다. 결함이 없는 셀의 값을 0으로 두고, 내장 함수를 통해 임의로 설정된 주수의 셀 값을 1로 변경하여 결함이 있음을 표현했다. RA algorithm은 결함의 발생 횟수, 행과 열의 주소, 예비 자원 수 등과 같은 데이터를 바탕으로 메모리 셀이 가진 예비 자원을 할당하게 된다. 1차원 배열 두 개를 만들어 각각 행과 열의 결함 수를 저장하는 카운터 역할을 하게 만들었고, 이를 바탕으로 예비 자원을 할당하는 RA algorithm을 C언어로 구현하였다.

3-2 메모리 반도체의 셀 구조

학습 및 실험에 사용하는 반도체는 10개의 행과 열을 갖고, 여분 수리 행과 열을 각각 2개씩 총 4개를 갖도록 구성하였다. 결함 발생률은 실제 양산 환경의 수율과 비슷하게 설정하기 위해 0.05(5%)로 지정하여, 무작위의 위치에 결함이 발생하도록 하였다. 이를 표현하기 위해 10×10 크기의 행렬을

1차원 배열 형태로 변환하였고, 이를 100,000번 반복하여 CSV 파일로 저장하였다.

3-3 데이터 전처리

C언어의 Fault-Map Generator로 생성된 반도체 칩의 결함 데이터는 그림 3에서 표기된 것처럼 1차원 배열의 형태로 펼쳐져 CSV 파일의 하나의 행으로 저장된다. 2차원 데이터의 공간적·지역적 특성을 고려하는 CNN의 장점을 살리기 위해 Python에서는 이를 2차원 배열로 재구성하여 독립 변수로 사용하였다. 수리 방법은 정수 값을 가지지만, 연속된 값으로 나타나지 않으며, 수리에 있어 정수 순서를 따르지 않으므로 범주형 특성을 가진다고 볼 수 있다. 범주형 변수를 표현하는 방식에는 대표적으로 라벨 인코딩(Label Encoding)과 원-핫 인코딩(One-Hot Encoding)이 있다. 원-핫 인코딩은 각 정수 값에 해당하는 인덱스만 1로 설정하여 범주형 변수를 표현하는 방식으로, 출력값을 이산화함으로써 분류 모델, 특히 딥러닝 기술에 적용할 수 있는 이점을 제공한다. 따라서 본 논문에서는 해당 방식을 사용하여 범주형 변수를 표현하였다. 그림 4는 실제 수리 방법 데이터가 원-핫 인코딩 과정을 거치는 것을 도식화한 것이다.

3-4 합성곱 신경망

합성곱 신경망(Convolutional Neural Networks, CNN)은 인간의 시신경이 시각피질로 신호를 전달하는 과정을 모방한 알고리즘이다. 대표적인 딥러닝 알고리즘 중 하나로, 그리드 형태의 위상(grid-like topology)을 가진 데이터를 처리하는 데 널리 사용된다. 그림 5는 CNN의 과정을 간단하게 도식화한 것이다. CNN 모델은 여러 은닉층을 쌓아 구현되며, 주로 합성곱 연산(Convolution)층과 풀링 연산(Pooling)층으로 구성된다. DNN(Deep Neural Network)은 주로 1차원 형태의 데이터를 사용하지만, 최근에는 이미지 데이터 등 2차원 배열 데이터의 급증으로 인해 기존 방법의 효율적 처리에 한계가 있다. 이미지와 같은 2차원 형태의 데이터를 1차원 형

태로 변형하여 사용할 경우 공간적·지역적 정보와 같은 고유의 특성이 사라져 큰 오류로 이어질 수 있다. 이러한 단점을 극복하기 위해 CNN이 도입되었으며, 본 연구에서도 데이터를 2차원 배열 형태로 재구성하여 CNN을 활용하였다 [14].

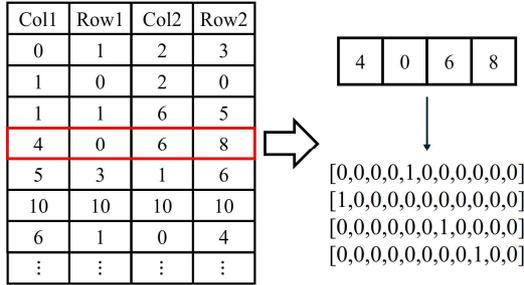


그림 4. 수리 방법 데이터 전처리 과정
Fig. 4. Data preprocessing for repair method

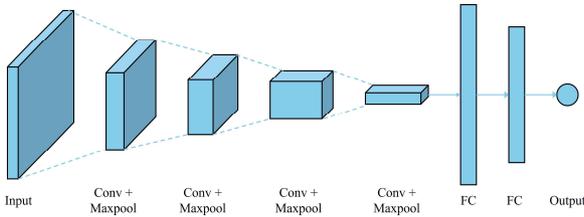


그림 5. CNN의 진행 과정
Fig. 5. The process of CNN

3-5 제안하는 모델 구조

제안하는 CNN 모델의 층 구조는 총 6개의 층으로, 합성곱 연산 층과 풀링 연산 층이 반복되도록 구성하였다. CNN에는 Filter(Kernel)가 존재하며, 합성곱 연산 층에서는 입력 데이터와 Filter 간의 내적 연산을 진행한다. 제안하는 모델의 입력 데이터는 100개의 칸 중 무작위로 선정된 5개의 칸만이 1의 값을 갖는 흑백 이미지이다. 따라서 Filter의 크기를 작게 설정하면 결함의 위치 특성을 파악하기 어려워질 수 있다. Filter 내에 결함의 특성이 담길 수 있도록 입력 데이터에 비해 크기를 크게 설정하여 모델을 구성하였다.

머신러닝이나 딥러닝 분야에서 모델 학습 시 과적합(Overfitting)을 고려해야 한다. 이는 모델이 training 데이터를 과하게 학습하여 해당 데이터에 좋은 성능을 나타내지만, 새로운 데이터에 대해서는 좋지 않은 성능을 나타내는 것을 말한다. 과적합 해결에는 여러 방법이 있지만, 제안하는 모델에서는 반복되는 층 사이에 드롭아웃(Dropout) 층을 두어 특정 데이터에 과적합 되는 문제를 방지하였다. 드롭아웃은 설정한 확률만큼의 다층 신경망의 숨겨진 뉴런 출력을 0으로 설정하여 학습하는 방법이다. 확률에 속한 뉴런은 전진 패스에 기여하지 않고, 역전파에도 참여하지 않는다. 제안하는 모델에서는 드롭아웃 층의 확률 인수를 0.3으로 설정하여, 약

30%의 뉴런을 탈락시킴으로써 과적합 현상을 해결하였다. 제안하는 모델은 수리 가능 여부를 예측하므로 0과 1 사이의 정수를 출력한다. 즉, 이진 분류 모델이므로 출력 노드의 활성화 함수로 softmax를 사용하였다. softmax 함수의 출력은 각 데이터 포인트가 특정 범주에 속할 확률을 나타내며, 가장 높은 확률을 가진 범주에 해당 데이터 포인트가 속한다고 예측한다.

표 1. 수리방법에 따른 수리 시간
Table 1. Repair Time By Method

number of analysis	Previous Method [7] (RM algorithm)		Proposed Method	
	time(ms)	accuracy(%)	time(ms)	accuracy(%)
100	17	100	256	100
1000	204	100	348	100
5000	512	100	481	100
10000	1368	100	682	100
20000	1966	100	1427	100

IV. 실험 결과

4-1 학습 결과

제안하는 모델은 원-핫 인코딩으로 전처리 된 종속변수를 사용하기 때문에 손실 함수로 categorical cross entropy를 사용하였다. 손실 함수는 신경망 성능의 저하 정도를 나타내며, 수치가 낮을수록 훈련 데이터를 잘 처리한 것이다. 평가 지표로는 accuracy를 사용하였고, 이는 전체 데이터 중 정확하게 예측한 데이터의 비율을 의미한다. 학습에는 epoch를 100, batch size를 32로 설정하여 진행하였다. 데이터의 20%는 검증을 위해 예약되어 모델의 성능을 측정하였다. 표 1은 기존의 RA 알고리즘 기반 수리 방법과 딥러닝 수리 예측 모델의 수리 시간을 분석 횟수에 따라서 비교한 것이다. 딥러닝 수리 예측 모델은 분석 횟수가 100개, 1,000개와 같이 적은 경우에는 기존 RA 알고리즘보다 낮은 성능을 보였지만, 분석 횟수가 늘어날수록 높은 정확성을 유지하면서도 RA 알고리즘에 비해 약 400 ~ 1800ms 정도 빠른 수리 속도를 보였다. 실제 양산 환경은 많은 데이터를 다루기 때문에, 이러한 결과는 딥러닝 수리 예측 모델이 수리 시간을 단축하는 데 유용할 수 있음을 시사한다.

V. 결론

본 논문의 실험은 두 가지 개발 언어를 사용하여 직렬 및 병렬로 결함을 수리하는 알고리즘의 수리 시간을 비교하는 방식으로 진행되었다. C언어를 통해 양산 데이터 산출 과정

과 결합 데이터를 순차적으로 수리하는 RA 알고리즘을 구현하였고, 해당 양산 데이터의 전처리 및 디버닝 수리 예측 모델은 Python으로 구현하였다. 디버닝 수리 예측 모델은 100,000개의 데이터를 학습하였으며, 총 100개, 1,000개, 10,000개의 데이터를 수리하는 데 걸리는 시간을 비교하였다. 실험 결과, 제안하는 디버닝 기반 수리 예측 모델은 높은 정확성을 유지하면서 기존의 RA 알고리즘보다 더 빠른 속도로 메모리를 수리할 수 있음을 확인하여, 디버닝 기반 수리 예측 모델이 RA 알고리즘에 적용 가능함을 확인하였다. 또한, 처리해야 하는 데이터가 많아질수록 더욱 빠른 처리 속도를 보이기 때문에, 실제 반도체 양산 환경에서 축적된 풍부한 raw 데이터를 학습하여 높은 정확도를 달성함으로써 반도체의 신뢰성을 향상시키고, 동시에 수리 시간을 단축하여 반도체 제조 비용 절감 효과를 기대할 수 있다. 본 연구에서는 repair-most 알고리즘을 기준으로 하였으나, 다른 RA 알고리즘에도 디버닝 기법을 적용하면 유사한 성능 향상이 있을 가능성이 있다. 이는 다양한 RA 알고리즘에 디버닝 기법이 적용될 가능성을 제시하며, 향후 연구를 통해 디버닝 기반 RA의 범용성을 더욱 강화할 수 있을 것이다.

감사의 글

본 연구는 산업통상자원부(20019363)와 KSRC 지원 사업인 미래반도체소자 원천기술개발사업의 연구결과로 수행되었음.

참고문헌

[1] G. E. Moore, "Progress in Digital Integrated Electronics [Technical Literature, Copyright 1975 IEEE. Reprinted, with Permission. Technical Digest. International Electron Devices Meeting, IEEE, 1975, pp. 11-13.]," *IEEE Solid-State Circuits Society Newsletter*, Vol. 11, No. 3, pp. 36-37, September 2006. <https://doi.org/10.1109/N-SSC.2006.4804410>

[2] H. Lee, S. Lee, and S. Kang, "RA-Aware Fail Data Collection Architecture for Cost Reduction," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 32, No. 6, pp. 1136-1149, June 2024. <https://doi.org/10.1109/TVLSI.2023.3343369>

[3] H. Lee, K. Cho, S. Kang, W. Kang, S. Lee, and W. Jeong, "Fail Memory Configuration Set for RA Estimation," in *Proceedings of 2020 IEEE International Test Conference (ITC)*, Washington, DC, pp. 1-9, November 2020. <https://doi.org/10.1109/ITC44778.2020.9325273>

[4] H. Lee, D. Han, S. Lee, and S. Kang, "Dynamic Built-In Redundancy Analysis for Memory Repair," *IEEE*

Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 27, No. 10, pp. 2365-2374, October 2019. <https://doi.org/10.1109/TVLSI.2019.2920999>

[5] H. Lee, Y. Yoo, S. H. Shin, and S. Kang, "Redundancy Analysis Simplification Scheme for High-Speed Memory Repair," in *Proceedings of the 20th International SoC Design Conference (ISOCC)*, Jeju, pp. 339-340, October 2023. <https://doi.org/10.1109/ISOCC59558.2023.10396258>

[6] C.-T. Huang, C.-F. Wu, J.-F. Li, and C.-W. Wu, "Built-In Redundancy Analysis for Memory Yield Improvement," *IEEE Transactions on Reliability*, Vol. 52, No. 4, pp. 386-399, December 2003. <https://doi.org/10.1109/TR.2003.821925>

[7] M. Tarr, D. Boudreau, and R. Murphy, "Defect Analysis System Speeds Test and Repair of Redundant Memories," *Electronics*, Vol. 57, No. 1, pp. 175-179, January 1984.

[8] S. Kuo and W. K. Fuchs, "Efficient Spare Allocation for Reconfigurable Arrays," *IEEE Design & Test of Computers*, Vol. 4, No. 1, pp. 24-31, February 1987. <https://doi.org/10.1109/MDT.1987.295111>

[9] H. Cho, W. Kang, and S. Kang, "A Fast Redundancy Analysis Algorithm in ATE for Repairing Faulty Memories," *ETRI Journal*, Vol. 34, No. 3, pp. 478-481, June 2012. <https://doi.org/10.4218/etrij.12.0211.0378>

[10] H. Cho, W. Kang, and S. Kang, "A Very Efficient Redundancy Analysis Method Using Fault Grouping," *ETRI Journal*, Vol. 35, No. 3, pp. 439-447, June 2013. <https://doi.org/10.4218/etrij.12.0112.0467>

[11] H. Lee, K. Cho, D. Kim, and S. Kang, "Fault Group Pattern Matching with Efficient Early Termination for High-Speed Redundancy Analysis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 37, No. 7, pp. 1473-1482, July 2018. <https://doi.org/10.1109/TCAD.2017.2760505>

[12] H. Lee, D. Han, H. Kim, and S. Kang, "W-ERA: One-Time Memory Repair with Wafer-Level Early Repair Analysis for Cost Reduction," in *Proceedings of 2020 IEEE International Test Conference in Asia (ITC-Asia)*, Taipei, Taiwan, pp. 94-99, September 2020. <https://doi.org/10.1109/ITC-Asia51099.2020.00028>

[13] J. R. Day, "A Fault-Driven, Comprehensive Redundancy Algorithm," *IEEE Design & Test of Computers*, Vol. 2, No. 3, pp. 35-44, June 1985. <https://doi.org/10.1109/MDT.1985.294737>

[14] L. O. Chua and T. Roska, "The CNN Paradigm," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, Vol. 40, No. 3, pp. 147-156, March 1993. <https://doi.org/10.1109/81.222795>



강유진(Yujin Kang)

2022년 : 전남대학교
컴퓨터정보통신공학과
(공학사)

2022년~현 재: 포스코 품질기술부
※관심분야 : 빅데이터 분석, 인공지능



이영우(Young-woo Lee)

2011년 : 인하대학교 전자공학
(공학사)
2015년 : 연세대학교 전기전자공학
(공학석사)
2020년 : 연세대학교 전기전자공학
(공학박사)

2011년~2015년: Teradyne SoC Test Division 선임
2020년~2021년: 삼성전자 Test & System Package 책임
2021년~2024년: 전남대학교 컴퓨터정보통신공학과 조교수
2024년~현 재: 인하대학교 반도체시스템공학과 조교수
※관심분야 : VLSI/SoC 설계 및 테스트, 메모리 테스트 기술,
보안 설계 방법론 등



오준석(Junseok Oh)

2023년 : 전남대학교
컴퓨터정보통신공학과
(공학사)

2023년~현 재: 전남대학교 지능전자컴퓨터공학과 석사과정
※관심분야 : SoC 시스템 설계 및 테스트, 반도체 보안 설계



서은솔(Eunsol Seo)

2023년 : 전남대학교
컴퓨터정보통신공학과
(공학사)

2023년~현 재: 삼성전자 메모리사업부
※관심분야 : 메모리 테스트



박석민(Seokmin Park)

2023년 : 전남대학교
컴퓨터정보통신공학과
(공학사)

2024년~현 재: 인하대학교 전기컴퓨터공학과 석사과정
※관심분야 : 메모리 테스트