

# ERC, 메모리반도체, 기계학습

이광근

서울대학교  
컴퓨터공학부

2/9/2022 @ SW재난연구센터 워크샵, 제주도

# 와글와글 ERC

운영철학: 탁월한 결과만 목표로 × 내실

- 최소로: 껌데기 행사/껌데기 논문/껌데기 일거리
- 숫자 세지 않기: 모든 정량지표는 근본적으로 가짜
- 뭘 선택하든(연구/교육/봉사) 탁월한 결과를 내는데 집중하자

운영철학: 탁월한 결과만 목표로 × 내실

- 최소로: 껌데기 행사/껌데기 논문/껌데기 일거리
- 숫자 세지 않기: 모든 정량지표는 근본적으로 가짜
- 뭘 선택하든(연구/교육/봉사) 탁월한 결과를 내는데 집중하자
- 이것 고집하다가, 3년후 1차 평가에서 B-C받음
  - 운영철학 수정안함
  - 센터의 문제 아니고 평가의 문제



연구개발 성과 지표 가이드라인  
Guideline for Research & Development Milestones

소프트웨어 무결점 연구센터

ROSAEC Center

KOSEF ERC

2009

센터 지원을 통한 연구성과를 아래 기준에 맞추어 작성해 주십시오.

진행중인 연구와 기대되는 성과도 포함해 주십시오.

교수님들이 개별적으로 작성해서 제출해 주시기 바랍니다.

우리의 성과 지표들은 다음의 방향으로 놓여있습니다.

- “Excellence”를 위한 노력을 존중하고 장려한다.
- 긴 안목에서 진행되는 연구/개발/교육을 존중하고 장려한다.
- 우리 주변의 문제를 연구/개발/교육을 통해 해결하는 노력을 존중하고 장려한다.
- 연구개발성고가 국내외에서 깊이 뿌리내리도록 하는 교육 자료 개발을 존중하고 장려한다.
- 연구/개발/교육/봉사 중 각자 선택/집중하는 항목을 존중하고 장려한다.



# SW무결점연구센터 워크숍: 센터활동의 꽃



# SW무결점연구센터 워크숍: 센터활동의 꽃

매년 여름×겨울 3박4일

- 성과발표(milestone talk): 30분-1시간
- 번개발표(lightning talk): 3분-5분
- 연구장터(poster): 저녁. 음식+술+음악
- 초청발표(invited talk)
- 기술전수(tutorial)
- 소그룹 난상토론(brainstorming)
- 워크숍공간의 생김새/분위기가 중요
  - 원하는 워크숍문화를 응원해줄 공간이어야

# SW무결점연구센터, 선도?

예가 되자

- 우리가 워샵에서 발표하던 것들이
  - 나중에 최고학회 논문이 되는: “그때 들은게 이렇게 크는구나”
  - 우리끼리 와-아! 한 것들  $\Rightarrow$  글로벌에서도 와-아!
  - 세계적인 연구  $\Leftarrow$  우리주변 문제를 제대로 푸는 연구

## 예가 되자

- 일류연구 비밀병기/즐거움 = 쉬운말로 소통하기
  - 쉬운 모국어의 은근 가공할 힘
  - 몸이 즉각 이해하게 표현해보라: 이해도↑
  - 워십 발표 제목만이라도
  - 뉴스레터만이라도
  - 애써 안 전문지식, 많이 쉽게 알게해야
    - 소수에만 맴돌면 ⇒ 쪼그라들어 망한다
    - 누구나 공헌할 선순환 필요: 집단지력 ↑
    - 쉬운모국어 × 영어

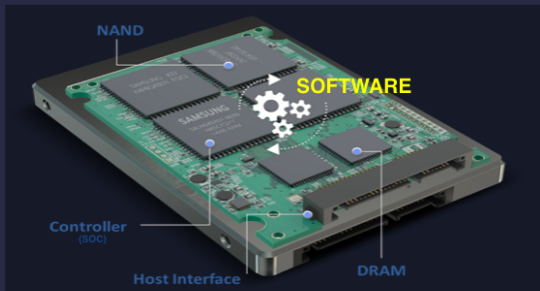
# 절체절명 메모리 반도체

# 삼성 플래시 메모리 SW 문제

- SSD: 51조 시장, 구글+아마존+MS+Meta+pc

**Consists of DRAM + Flash + Controller (CPUs + Accelerators) + Software**

Software integrates and orchestrates many components, and makes differentiations





# 삼성 플래시 메모리 SW 문제

## SSD 소프트웨어 품질/비용/성능 문제

- Visual Studio, C, C++, ARM Compiler
- 300만줄(2021)
- 개발 및 테스트 환경: in-house tools, Crucible, Coverity, SonarQube, Jira, Confluence
- 비용: SW개발팀 70% (100s defects) + QA 30% (10s defects)
- 비용증가율: 각각10배. SW개발팀 → QA → 고객
- memory leak, buffer overflow, race, deadlock, uninitialized var, memory overwrite, misconfigured build, uncaught exceptions

# 삼성 플래시 메모리 SW 문제

위기: 닥쳐온 먹구름

- 악화일로인 SSD SW 품질/비용/성능 문제
- 점점 복잡 어려워지고 있는 이유
  - 새로운 성능요구
  - 새로운 산업기준
  - 무너진 가정: HW결함 가능성

# 삼성 플래시 메모리 SW 문제

우리 모두 힘을 합쳐 능력을 보일때 (우리 존재 이유)

- 특수: 삼성 SSD SW에 특화시킨 기술로
- 할 수 있는 일만: 우리가 해낼 수 있는 문제에 집중

# 삼성 플래시 메모리 SW 문제

우리 모두 힘을 합쳐 능력을 보일때 (우리 존재 이유)

- 특수: 삼성 SSD SW에 특화시킨 기술로
- 할 수 있는 일만: 우리가 해낼 수 있는 문제에 집중
- 1단계 1년: 현장상황 상세히 파악
  - 해결할 수 있는 문제 리스트 발굴

# 삼성 플래시 메모리 SW 문제

우리 모두 힘을 합쳐 능력을 보일때 (우리 존재 이유)

- 특수: 삼성 SSD SW에 특화시킨 기술로
- 할 수 있는 일만: 우리가 해낼 수 있는 문제에 집중
- 1단계 1년: 현장상황 상세히 파악
  - 해결할 수 있는 문제 리스트 발굴
- 2단계 N년: 문제마다 팀결성 및 해결
  - 제대로 해결  $\Rightarrow$  보람 + 세계적인 성과
  - 최고 논문은 자연스레 따라나오고

# 요놈봐라 기계학습

# 데이터의 중력: 새 프로그래밍 중력

- 기계 학습 *machine learning* (인덕션 *induction*)

# 데이터의 중력: 새 프로그래밍 중력

- 기계 학습 *machine learning* (인덕션 *induction*)
  - 입력: 관찰한 데이터  $\{(a_0, b_0), \dots, (a_k, b_k)\}$
  - 출력: 짐작하는 함수



# 데이터의 중력: 새 프로그래밍 중력

- 기계 학습 *machine learning* (인덕션 *induction*)
  - 입력: 관찰한 데이터  $\{(a_0, b_0), \dots, (a_k, b_k)\}$
  - 출력: 짐작하는 함수
  - 예:  $\{(-1, 0), (2, 3)\}$ 에서?

# 지금까지의 프로그래밍

## 기둥 A: 기계학습 이전

- 언어로 논리적인 문제풀이법을 표현
- 계산방법을 꼼꼼히 상위의 언어로 손수작성
  - 현실적인 비용의 계산과정이어야

*프로그래밍* *programming*, *알고리즘* *algorithm*

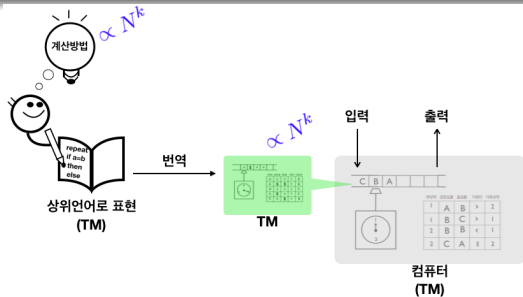
## 기둥 Z: 기계학습

- 어떨하지? 언어로 논리적으로 표현못하겠다
- 정답예시를 주입식으로
  - “보고 따라하길”
- 기계학습: “서당개 5만년” 훈련
  - 수많은 정답예시 주입해서 SW 자동생성
  - 기계학습 과정과 결과 SW 모두
  - 현실적인 비용의 계산과정이어야
- 결과 SW - 뉴럴넷<sub>neural net</sub> 을 컴퓨터에 심는다

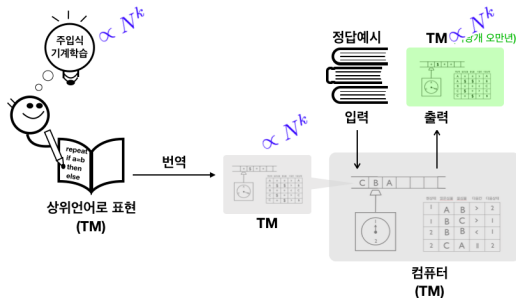
기계학습<sub>machine learning</sub>, 에코리즘<sub>ecorithm</sub>

# 프로그래밍의 두 기동

기동 A



기동 Z



# 기계학습이 만드는 프로그램을 분석/검산/검증

- 기계학습으로 만든 SW의 특성

- “분자단위”로 표현된 SW (예, 뉴럴넷)
- 틀릴 수 있는 아마도의 세계
- 얼추거의맞는 *probably approximately correct* SW

틀릴비율  $\epsilon$ 과 확률  $\delta$ 가 있어서, 학습결과 SW  $F$ 는 다음을 만족시킨다:

$$\text{Prob}[\text{Wrong}(F) \leq \epsilon] \geq \delta.$$

# 기계학습이 만드는 프로그램을 분석/검산/검증

- 기계학습으로 만든 SW의 특성

- “분자단위”로 표현된 SW (예, 뉴럴넷)
- 틀릴 수 있는 아마도의 세계
- 얼추거의맞는 *probably approximately correct* SW

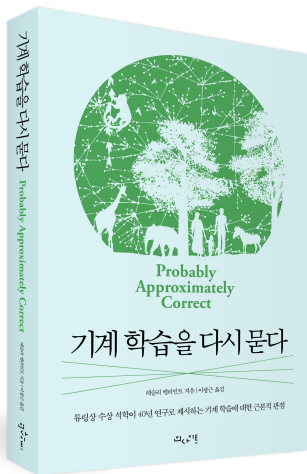
틀릴비율  $\epsilon$ 과 확률  $\delta$ 가 있어서, 학습결과 SW  $F$ 는 다음을 만족시킨다:

$$\text{Prob}[\text{Wrong}(F) \leq \epsilon] \geq \delta.$$

- 새 문제: 이런 인공지능SW를 분석/검산/검증하기

- 문제 정의의 어려움:  $F \models \mathcal{P}$  iff ?
- $\mathcal{P}$  정의의 어려움: 예) 공평함? 잘깨지지않음?
- 확률/통계적으로만 가능
- 새로운 정적분석/검증 기술을 열어야

# 기계학습의 근본정의



감사합니다