최컴특 중간 (1-6)

01-1 행동유형의 이해와 생활 에티켓

행동 패턴, 행동 스타일

- 자기 나름대로의 독특한 동기요인에 의해 일정한 방식으로 하는 행동
- 4가지 유형으로 나뉨 - DISC
  - **주도형 Dominance**
  - **사교형 Influence**
  - **안정형 Steadiness**
  - **신중형 Conscientiousness**

주도형

- **목표의식, 결과 지향**
- 공격적, 성급함
- 통제력 상실에 두려움
- 스트레스 -> **타인 감정, 생각 무시**

사교형

- 우호적, 호의적인
- **관계 지향**
- 낙천적, 비현실적
- **칭찬 인정** -> 동기부여
- 스트레스 -> **충동적**

안정형

- **사람, 그룹** 지향적
- 일관성, 꼼꼼
- 전문적 기술 개발
- 스트레스 -> **소심, 양보**

신중형

- 일 중심, 과업지향
- **정확한 것** -> 동기부여
- 중요한 지시, 기준에 관심
- 스트레스 -> **까다롭고 비판적**

	주도형	사교형	안정형	신중형
특성	강한 자의식	낙천적	일관성	분석적
목표	목표/결과 성취	사람/관계 유지	조화/평화	과업/완벽함
동기요인	도전/지시	사회적 인정	현상유지	정확성/원칙
두려움	통제력 상실	사회적 거부감	변화	타인의 비판
약점	몰인정/비경청	비체계적	지나친 양보	비판적
자주 쓰는 말	할 수 있어	나 어때?	내가 도와줄께	그게 맞는 말이야?
	내가 책임질께	너무 재밌지?	잘 지내보자	생각할 시간을 줘
	애같이 굴지마	복잡한 거 질색이야	나에게 강요하지마	원칙을 지켜야지

- 종합적, 정적 분석 -> **환경, 상황 따라 극복 가능**
- 육하원칙(5W1H) : 누가, 무엇을, 언제, 어떻게, 어디서, 왜
- ATTITUDE : 100%

01-2 인공지능 시대의 임베디드 시스템 연구

- AI 기술의 시스템화 - 세 가지 이유
  1. Privacy
  2. Latency
  3. Cost
- Embedding ResNet @ Server
- Language 모델을 위한 딥러닝
  - 모델 크기는 기하급수적 증가중

과업

1. **정확도 유지하며 모델 크기 압축**
   - 업로드 시간, 용량, 추론 시간
2. **제한된 전력에서 속도 개선**
   - 10 Watt 이하
   - HW-SW 최적화
3. **온디바이스 AI 가속**
   - 새로운 NN 가속기
   - 병렬화
   - 동적 최적화

- **경량 딥러닝 모델**
  - GoogleNet, ResNet 등등 ~Net들
- **딥러닝 모델 가속 : 최적화, 병렬화**
  - DeepX, Cappuccino, DeepMon, New NN Accelerator

- **딥러닝 모델 압축 : 양자화, 프루닝**
  - Quantization, Prunning
- 추론 최적화 위한 NVIDIA **TensorRT**
- 이미지 분류에서 딥러닝 가속
- **Transformer** 기반 번역 모델 압축
- **DNN and MAC operation - 곱+합연산**
  - 딥 뉴럴 네트워크와 곱연산
- 프로세서 에서의 경량화/가속기술
- BEV - 위에서 본 화면

임베디드 AI 하드웨어

- CPU 보단 SW 싸움 -> 어플리케이션 최적화
- **Qualcomm Hexagon DSP**
  - 퀄컴 : SOC 강자
  - 저전력 성능
  - 고속 MAC 기반
- **Apple Neural Engine**
  - 뉴럴넷 가속기 + MMA 기반 병렬처리 GPU
- **Nvidia Jetson**
  - 구조가 다름 processing element 여러개 -> 뉴럴 닮음
  - PE 기반 MMA 가속형 병렬처리 GPU
  - 양자화, 프루닝 HW 지원
  - 최적화 SW 스택 지원
  - 384로 ai 돌림
- **Tensor Processing Unit - TPU**
  - Google사 개발, 뉴럴 프로세서
  - 검색엔진, 알파고 사용
- **FPGA 가속기**
  - 도메인과 응용에 특화
- **Microcontrollers (MCU)**
  - 제어용, 연산과 가속 X
  - 모델경량, 압축으로 최적화
- CPU는 이미 다다름 -> 캐시, 메모리 활용

랩실 작업

- TF, PyTorch XXX
- 오직 C언어
- Nvidia의 **TensorRT** 사용
- 실시간 온보드 AI 컴퓨팅 - 영상획득과 동시에 실시간 영상처리

02-1 Edge컴퓨팅 환경 인공지능 응용 연구

- 드론 이미지에서 물체 검출
  - drone 혹은 GCS에서
- **on Drone**
  - 크기, 전력, 무기
  - 비디오 전송 필요 X -> 고성능 이미지 사용, 바로 분석
  - 즉시 제어 가능
- **on GCS**
  - 드론에 ai 탑재 불필요
  - 더 강력한 ai 사용
  - 유저가 확인 가능

엣지 AI 컴퓨팅 장치

- **NVIDIA Jetson**
  - Nano series
  - NX series
  - AGX series
- **Google Coral**
  - USB accelerator
  - Coral dev board
  - Coral module (chip)
- **Hailo**
  - Hailo Module (M.2)

장점

- 낮은 레이턴시
- 연결 필요 X
- 넷 트래픽 ↓
- 확장성
- 보안성

Drone Defense

- 이슈들
  - 실시간 속도
  - 음영 구역
  - 확장성- 1cam & 1GPU?
  - 드론, 카메라 크기 문제

비포장 도로

- 경계가 따로 없음
- 계절별 변화
- Bottlenecked 아닌 **asymmetric non-local block (ANB)** 사용
- **SOTA 알고리즘** - 이미지 분석에 좋음
- **Lidar**로 free-space 검출
- 두 개의 다른 CNN-기반 알고리즘 적용
  - **TAN-Net**
  - **SalsaNet**
- **Point Cloud + Image segmentaion -> BEV map**
- **Alpha-shape** 알고리즘
  - point-wise -> **area-wise**
  - 시계열 축적을 넣으면 성능 up

부분 가려진 객체 감지

- 소형 저전력 AI 장비로 실시간 검출
- 대체 증강(Substitution augmentation)
  - 이미지 짜깁기

낙상 감지

- 가상 데이터 <-> 리얼 데이터 : 차이 큼
  - **둘 다 Skeleton으로 만들면 차이 ↓**
- 가상 데이터 -> 특징 추출 -> 실제 훈련 DNN에 넣어줌

기타

- 도메인들 -> 점점 확장
  - Edge 컴퓨터
  - Sensors/platforms
  - problem domain
- NN 구조 + 러닝 메소드
  - 자기지도 사전학습
    - pretext learning
    - masked image modeling
  - Neural network 구조
    - NN 구조
    - Attention Module
    - Non-local attention
  - 러닝 메소드

        - 멀티 태스크 러닝
        - One-shot learning
- 데이터 증강, 생성
    - 3D 엔진

02-2 컴퓨터 이론 연구

이론 전산학 (TCS)

- 수학을 이용해 전산학 문제들 정의, 해결
- 2021 **아벨상(=필즈상급** 최고 권위)은 전산학
- 결과를 수학적으로 증명이 중요!!
    - 실험 결과는 의미 X
    - Sorting network -> O(log n) depth sorting network의 존재가 증명됨
        - hidden 상수가 너무 커서 공학적으론 의미 X

정수 seq 표현

- **가능한 한 작은 공간, 임의 접근 가능**
    1. 알파벳 마다 **4bit -> O(1)**
    2. 10^n 개의 seq에 순서대로 숫자부여 **nlog10 bits -> O(n)**
    - 최적화 -> **(nlog10) bits, O(1) 접근 시간**

편집 거리

- 최소의 삽입, 삭제, 추가로 같게 만들기
- O(n^(2-e))가 하한값
    - SETH가 거짓이 아닌 경우
    - SETH : 큰 k에 대해 k-SAT 해결에는 대략 2^n 시간 필요

Optimal online binary search tree

- Total search cost를 최소로 하는 BST 설계
- Splay tree 는 online case에서도 O(1)-competitive 일까?
    - 온라인에서 최악의 상한

03-1 컴퓨터비전연구

- PC가 사람의 인지능력 얻기
- 인공지능 크게 세 가지
    - **Classification** : 분류
    - **Detection** : 위치 검출 (bounding box)
    - **Segmentation** : 객체 검출 (픽셀 단위)

- Segmentation
  - **Semantic : 카테고리 별**
  - **Instance : 객체 별**
  - **Panoptic : 카테고리 + 객체**
- 얼굴 인식
  - **Yaw rotation**
  - **Pitch rotatiton**
  - **Roll rotation**
  - **Partial**
- 이미지처리
  - 화질 개선
  - 디-웨더링
  - 디헤이징 - 안개
  - 이미지 색칠
  - Super-화질개선
  - 가린 부분 복원
  - style transfer
  - img-to-img 번역
    - 라벨, 낮/밤, 색깔, 테두리 등등
- 이미지 생성
- 이미지 분류
  - **AlexNet - first winner CNN**
- CNN
  - Convolutional layer : 필터로 맵 생성
  - Receptive field : 이미지에서 정보 수집
  - Pooling : 맵 크기 줄이고 과적합 방지
- ZFNet : 향상된 하이퍼파라미터
- VGG, GoogleNet : 더 깊은 network
- ResNet, SENet : 깊이의 혁명
- ConvNet인 BiT가 transformer인 ViT보다 우세
  - **데이터/파라미터 효율성, 특징 계층 구조, 미세 조정 때문** !
- Image enhancement
  - Multi exposure correction
- CVPR 2023 - 컨퍼런스
  - 3d from multi view and sensor
  - 3d object pose & shape reconstruction

03-2 PL Research

1. **정적 분석** : 실행을 안하고 결론을 냄

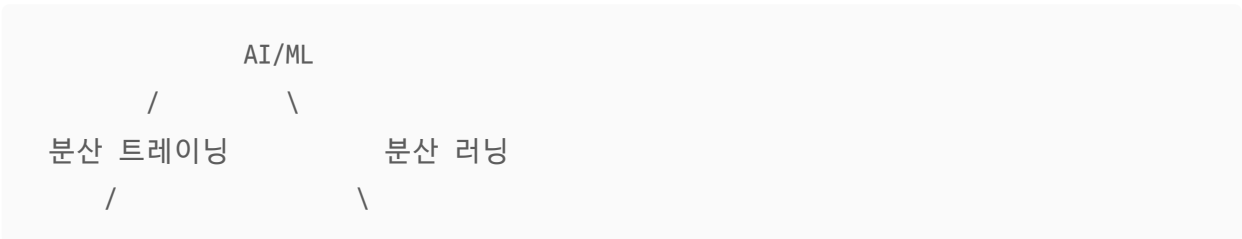
2. **동적 분석** : 실행시키면서 동작을 유추, 결과를 분석

- 기준 -> 실행!!
  - static은 실행은 없다(실행 이전)
  - dynamic은 실행중(runtime)
- 수식을 통해 동작 법칙, 규칙 정의 -> semantics
- Malicious behavior : Activity Injection
- Security vulerabilities in Ads platform SDKs
  - CallGraphBuilder -> Vulnerable Pattern ditector
- Task Migration : 기기들끼리 일을 나눠서 함
- 프로그래밍 언어 이론이란 **프로그램의 실행 동작을 이해**하기 위한 노력
  - (때때로) 잘 정제된 수식을 사용하여 프로그램의 실행 동작을 유추
  - 프로그램의 실행 없이도 프로그램의 형태에 기반한 분석을 수행
    - 분석도구는 자동으로 결함, 보안취약점, 악성행동을 탐지
  - 탐지 뿐 아니라 프로그램을 자유롭게 가지고 놀 수 있는 방법론
    - 프로그램 합성, 변형 등을 수행하기 위해 기본이 되는 이론

04-1 분산 학습 시스템

- 통신의 단계
  1. 비트 전달
  2. 의미 전달
- 이미지 전달
  - 과거 : 이미지 -> 비트
  - 현재 : 이미지 -> 텍스트 의미 전달 -> 이미지 생성

—



분산 컴퓨팅 ----------(무선)통신

분산 트레이닝

- 모델 사이즈 up, 연산 부담도 up
- 하나의 GPU로 모델 학습 X
- 여러 GPU 활용

1. **Data Parallelism 데이터를 나눠서, 병렬화**
   - 병렬화 쉬움

    - 활용 어려움, 비용 ↑
2. **Model Parllelism 학습(순차적)을 나눠서**
    - 활용도 ↑
    - 연결 전달, 병렬화 어려움, 로드 밸런싱 문제
3. **Tensor Parallelsim 텐서(연산)를 쪼개서 뿌리자**

Data Parallelism

- **Parameter Server**
    - 높은 대역폭 요구 -> worker가 많을 경우 overload
        1. **중앙화 서버 없이 분산 학습**
        2. **기울기 압축**

분산 Learning

- 여러 기기로부터 얻은 정보
    - 중앙 서버로 -> Cloud
    - 기기에서 학습 -> On Device
- 여러 곳에 존재하는 데이터를 모으지 않고 학습
    - but, 개인의 데이터 충분 X, 자원도 X
- 종류
    - **연합 학습**
    - **분할 학습**
    - **탈중앙 학습**

연합 학습

- 서버가 있지만 데이터 전달은 X
- 학습한 결과만 수집해서 모델 업데이트
- 과제
    - 통신 문제
    - 평균 결과 -> 각각 모델 성능 저하
    - 결과도 데이터 침해 가능

분할 학습

- 전체 모델을 쪼개서 디바이스 - 서버로 학습
- 학습의 중간 결과를 서버로 전달
- 장점
    - 모델 privacy ↑
    - 클라 연산 부담 ↓
    - 서버-클라 통신 오버헤드 ↓

    - 빠른 모델 수렴
- 단점
    - Label 유출
    - 느린 학습
    - 서버-클라 잦은 통신

04-2 Using Trees in Machine Learning

Decision Tree

- 분류 Classification 모델
- 리프 노드에 도달 -> Label 결정
- 답이 알려진 데이터 (학습 데이터) => 트리 구성
- 직관적이지만 Unstable함
    - -> 앙상블 메소드 사용 -> **Random Forest**

Random Forest

- 랜덤하게 구성 -> 병렬적
    - 서로 보완하려면? -> XGBoost

XGBoost

- Extreme Gradient Boosting
- 이전 트리의 예측 오차를 기반으로 새로운 트리를 훈련시켜 더함

Outlier Detection

- Outlier (Anomaly, abnormality, or novelty)
- **Outlier 이상치** : majority에서 벗어나는 데이터 샘플
- 활용
    - 사기꾼, 고장, 질병 detection
- 많은 방법이 있음 -> Tree-based
    - Deep SVDD : 딥러닝 + SVM

Isolation Forest

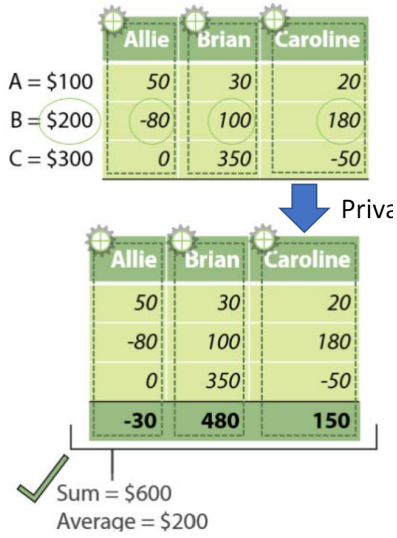
- iTree를 여러개 사용 -> 이상 데이터를 분리
- 이상 데이터는 루트 노드에 근접
- 아웃라이어 스코어 : 몇 번만에 리프 노드로 분리되는가?
    - iTree 들의 아웃라이어 스코어의 합을 계산
- 각 클라에서 로컬 데이터로 트리 앙상블 구성 -> 합침

06-1 Confidential Computing

- 신뢰 컴퓨팅의 세 가지 측면
  1. at **Rest** : 저장된 상태 : DB 암호화 등
  2. in **Transit** : 전송할 때 : HTTPS, ...
  3. in **Use** : 연산할 때
     - 이건 어떻게? -> 최신 기술

Protection in Use

- **Secure Multiparty computation**
  - 평균, 총합 쪼개기 -> 비밀 노출 없이 공유



- Homomorphic Encryption
  - 암호화된 채로 연산 가능

CPU 보안

- OS 단도 악성코드 피격 가능
- OS를 안전하게? -> 가상화 기술
  - 가상화 위에 OS -> 하지만, 가상화도 피격 가능
- 그러면 정답은? -> **하드웨어**
- Intel **SGX**
  - 격리된 공간에 프로세스 실행
  - 보호되는 특정 메모리 공간 : **Enclave**
- AMD **SEV**
- ARM **TrustZone**
  - 주로 모바일 폰
  - 메모리 공간을 나눔 : 응용/중요한
  - CPU 모드 노말/시큐어 모드
  - Secure Enclave Processor -> apple : 보안 전용 CPU 추가

- DRM : Digital Right Management
  - 지식재산권 컨텐츠 보호
  - HW에 구워두고, 유저가 쓸 때 사용
- 개인정보 유지 접촉 트래킹
- KNOX
  - 리얼타임 커널 프로텍션 : RKP
- TEE 넘어서 -> ARM CCA
- CPU 제조사를 믿어야 함 ;;

06-2 4차 산업혁명 시대의 인재상과 인공지능융합대학원

- 산업혁명
  - 1차 : 생산, 수송 기계
    - **육체 노동의 자동화**
  - 2차 : 전기, 대량생산
  - 3차 : 전자, 컴퓨터, 인터넷
  - 4차 : SW, 지능정보
    - **SW혁명 : 정신 지식노동의 자동화**
- 4차 산업혁명 요인
  - 디지털 기술의 보편화
    - **고성능 컴퓨터 칩**
    - **초고속 네트워크**
    - **SW, 인공지능**
  - 최고의 메타기술!!
- 4차 산업혁명의 진짜의미는?
  - 전통적 : 사람 입 -> CPU -> 사람 출
  - 스마트 : Sensing -> processing -> Control
  - Smart City
- 사회적 의미
  - **맞춤화 - 소비자 위주**
  - **개방 - 공유 - 분권화**
- **4차 -> 전문 서비스**
  - 1,2 차 -> 제품
  - 3차 -> 정보
- **가치 창출 대학**
  - **도전 정신, 기업가 정신**
- 디자인 씽킹
  - 사람에 대한 이해 -> 가치 창조
  - Empathize -> Define -> Ideate -> Protype -> Test
  - **EDIPT**

- 창업/ 기업가 정신
  - 혁신!
- 융복합
  - 융합은 **수단!!**
- 문제/프로젝트 중심
- 인재상 : **SEARCH**
  - Seek
  - Ensemble & Empathy
  - Application
  - Redefine
  - Communication & Collarboartion
  - Help
- ABC + D + E = V