



AI 프로그래밍

컴퓨터 구조와 AI 프로그램

컴퓨터 구조와 AI 프로그램

01

컴퓨터 구조와 AI 프로그램

- Machine learning

- 컴퓨터가 **데이터**로부터 특정 작업에 필요한 **feature** (특징)를 식별할 수 있도록 학습시키는 작업
 - 예시
 - 통신 내용을 분석해 보이스 피싱인지 식별하는 시스템
 - 생체 신호를 분석해 질병 여부를 파악하는 시스템
 - 차량의 카메라 영상을 통해 안전한 운전을 돕는 시스템
- 20세기에는 컴퓨터의 처리 능력 및 데이터의 양이 부족
 - 사람이 데이터를 분석하여 **hand-crafted feature** (수제 특징)를 직접 추출
 - 수제 특징을 기반으로 컴퓨터가 효율적인 학습을 진행
- 21세기에 들어 컴퓨터의 **처리 능력**과 **데이터 양**이 폭발적으로 증가
 - 사람이 인지하지 못했던 feature까지 컴퓨터가 **스스로 추출**할 수 있게 됨
 - 기존 방식에 비해 문제 해결 능력이 **압도적**으로 높아짐

컴퓨터 구조와 AI 프로그램

- ANN (인공 신경망, Artificial Neural Network)
 - 현대 AI는 ANN을 활용
 - ANN이란 동물의 뇌를 소프트웨어로 구현한 모델
 - 인간이 인지하지 못했던 feature들을 식별하고 활용하기 위해 자연을 모방
 - 따라서 ANN은 매우 많은 수의 뉴런으로 구성
 - 다수의 뉴런을 동시에 작동시키기 위해선 소수의 CPU 코어로는 부족
 - 거대한 ANN을 실행하기 위해선 특수한 하드웨어 권장
 - 많은 ANN 시스템이 nVidia의 CUDA를 활용

- CUDA를 사용하는 이유

- CUDA (Compute Unified Device Architecture)란 GPU를 일반 작업에 동원하도록 하는 시스템

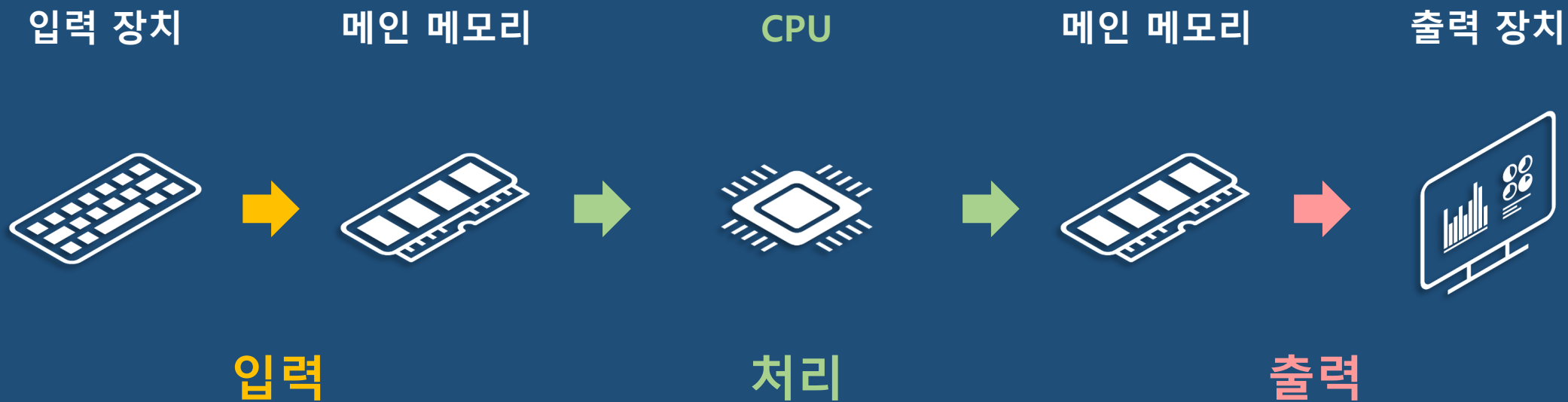
- GPU (Graphics Processing Unit)

- 출력할 화면의 실제 픽셀 값을 계산하는 처리 장치
 - 조명과 그림자, 흐림, 반사 및 굴절 등을 고해상도로 실시간 계산하여 출력해야 함
 - 따라서 GPU는 동시에 많은 연산을 수행해야 함
- CPU 코어에 비해 개별 성능은 떨어지지만 그 수가 압도적으로 많음
 - CUDA를 통해 GPU 코어를 일반 연산에 활용, 인공지능망의 뉴런 작업을 동시에 처리하게 함
 - 단, GPU가 활용하는 메모리는 별도로 존재(VRAM)하며, 메인 메모리에 비해 용량이 매우 적음

Slightly Mad Studios의 Project C.A.R.S. 캡처 화면

컴퓨터 구조와 AI 프로그램

- 컴퓨터 구조와 일반 프로그램



컴퓨터 구조와 AI 프로그램

- 컴퓨터 구조와 일반 프로그램

- 1. **입력**: 외부로부터 데이터를 받음

- 외부: 키보드/마우스/마이크 등의 유저 입력, 보조 기억 장치(HDD, SSD), 네트워크 등
 - 외부의 데이터를 메인 메모리(RAM)에 저장

- 2. **처리**: 입력 받은 데이터 가공

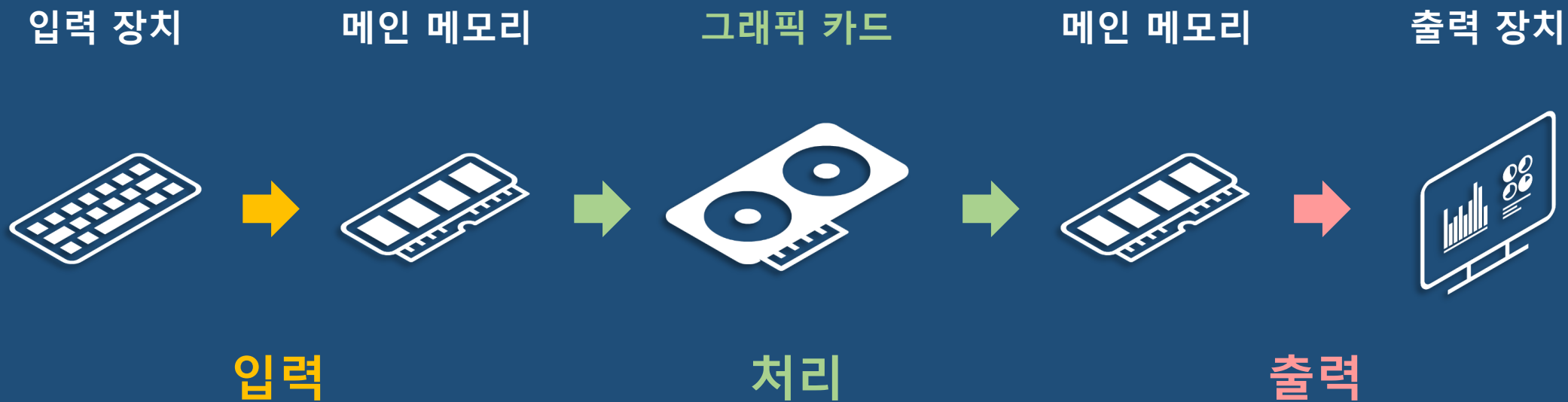
- 메인 메모리(RAM)의 데이터를 CPU 코어가 연산 처리
 - 처리한 데이터를 다시 메인 메모리(RAM)에 저장

- 3. **출력**: 가공한 데이터를 외부로 출력

- 외부: 모니터, 스피커, 프린터, 보조 기억 장치(HDD, SSD), 네트워크 등
 - 메인 메모리(RAM)의 데이터를 외부에 저장

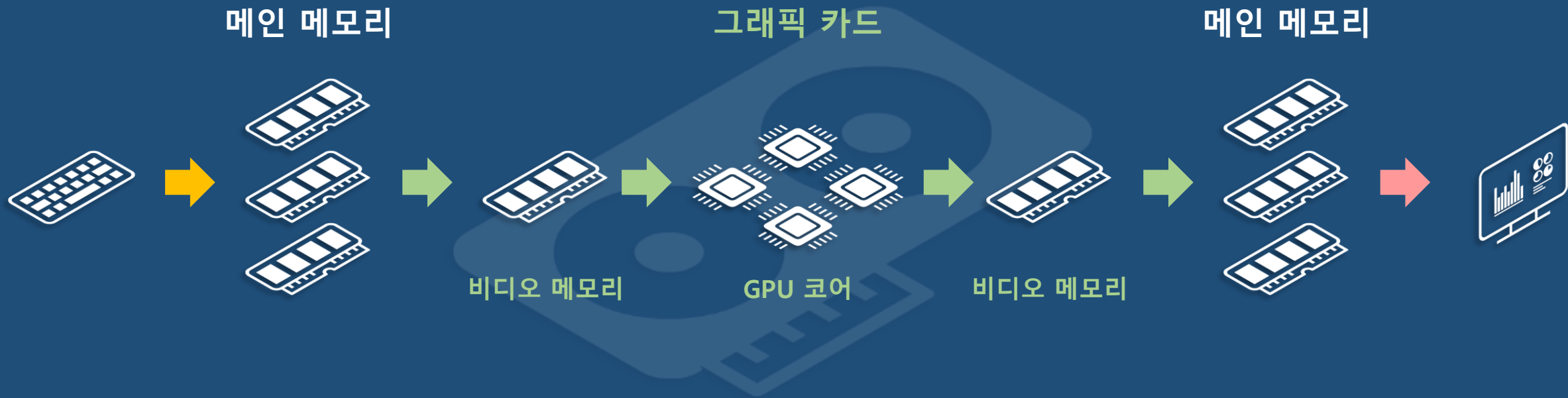
컴퓨터 구조와 AI 프로그램

- 컴퓨터 구조와 AI 프로그램



컴퓨터 구조와 AI 프로그램

- 컴퓨터 구조와 AI 프로그램



컴퓨터 구조와 AI 프로그램

- AI 프로그램의 처리 과정

1. 메인 메모리의 데이터를 그래픽 카드의 **비디오 메모리(VRAM)**로 복사
2. 비디오 메모리(VRAM)의 데이터를 다수의 **GPU 코어**가 나누어 **병렬 처리**
3. 각 GPU 코어가 처리한 결과를 **비디오 메모리(VRAM)**에 저장
4. 비디오 메모리(VRAM)의 데이터를 메인 메모리에 복사

컴퓨터 구조와 AI 프로그램

- AI 프로그램의 구조

딥러닝 라이브러리

: 인공신경망 **개발**, **학습**, **실행** 등을 쉽게 구현하기 위한 라이브러리

CUDNN

: CUDA를 **ANN**에 최적화시킨 라이브러리

CUDA

: nVidia GPU 코어를 **일반 연산**에 활용하기 위한 라이브러리

nVidia GPU

컴퓨터 구조와 AI 프로그램

• 딥러닝 라이브러리

- ANN 개발, 학습, 실행에 필요한 구체적인 내용들을 **간단한 API**로 활용할 수 있음
- 다양한 **ANN 프로토타입**을 빠르게 개발하고 실험하는데 매우 유용함
- TensorFlow, Keras, PyTorch 등 다양한 라이브러리들이 있음



- TensorRT와 같이 ANN 실행에 특화된 라이브러리도 있음



컴퓨터 구조와 AI 프로그램

- 딥러닝 라이브러리

- 본 강의에서는 **PyTorch** 사용법을 학습
 - PyTorch는 **Python**을 언어로 사용하는 딥러닝 라이브러리
 - Python 언어를 사용하기 위해 IDE(통합 개발 환경)으로 **Anaconda**를 활용하여 강의 진행
- PyTorch를 제대로 활용하기 위해선 **CUDA**를 지원하는 nVidia의 **그래픽 카드** 사용 권장
 - 그래픽 카드가 없는 경우 **CPU 전용 버전**으로 설치 가능
 - 또는 본 수업을 명분으로 컴퓨터를 새로 구매하는 것도?

개발 환경 구축 (Windows)

02

개발 환경 구축(Windows)

- 개발 환경 구축 순서
 1. Anaconda 설치
 2. CUDA 설치
 - PyTorch가 지원하는 버전의 CUDA 및 nVidia 드라이버 설치
 3. CUDNN 설치
 - 설치한 CUDA 버전에 맞는 버전의 CUDNN 설치
 4. Anaconda에 PyTorch 설치
- 자세한 방법은 실습시간에 학습

개발 환경 구축 (Ubuntu)

03

개발 환경 구축(Ubuntu)

- 개발 환경 구축 순서
 1. Anaconda 설치
 2. CUDA 설치
 - PyTorch가 지원하는 버전의 CUDA 및 nVidia 드라이버 설치
 3. CUDNN 설치
 - 설치한 CUDA 버전에 맞는 버전의 CUDNN 설치
 4. Anaconda에 PyTorch 설치
- 자세한 방법은 실습시간에 학습

감사합니다

