

2017 2학기 인공지능 강의노트

1주차

강사: 양일호

강의개요

- 강의

- 소개

- 강사 / 과목 / 교재 / 평가 방법 / 강의 계획 / 유의사항

- 인공지능의 응용분야

- 간단한 패턴인식: 전문가 시스템

- 실습

- Python 개발환경 준비

- WinPython / notepad++ (NppExec)

- 간단한 전문가 시스템 구현



강의

강사 소개 - 양일호

- 이력

- 2004년 3월~: 서울시립대학교 컴퓨터과학부 공학사
- 2008년 3월~: 서울시립대학교 컴퓨터통계학과 이학석사(화자 분할)
- 2010년 3월~: 서울시립대학교 컴퓨터과학과 공학박사(화자 인식)
- 2017년 9월~: 서울시립대학교 컴퓨터과학부 연구교수

- 근무지

- 장소: 정보기술관 323호 지능형로봇연구실

- 연락처

- 전화: 02-6390-5697 (010-9068-9582)
- 메일: heisco@hanmail.net



과목 소개

- 교과목명
 - 인공지능(Artificial Intelligence)
- 과목개요
 - 인공지능의 기본개념, 응용분야에 대한 지식 전달
 - 기계학습 / 패턴인식 관련 기술을 주로 다룸
 - 현재 널리 사용되고 있는 방법들 중 일부를 실습
 - 실습은 python을 통해 수행
 - 기본적인 python 문법 소개
 - 관련 라이브러리
 - Numpy, theano, keras



강의 교재

- 주교재

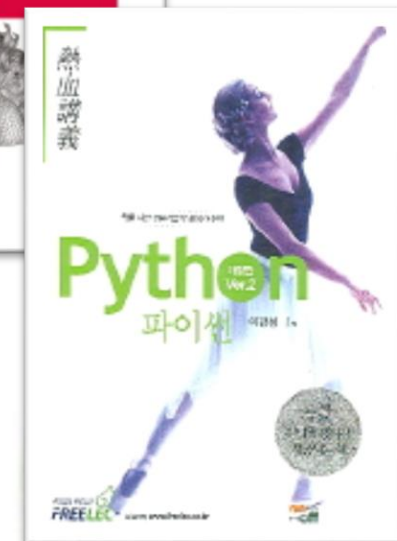
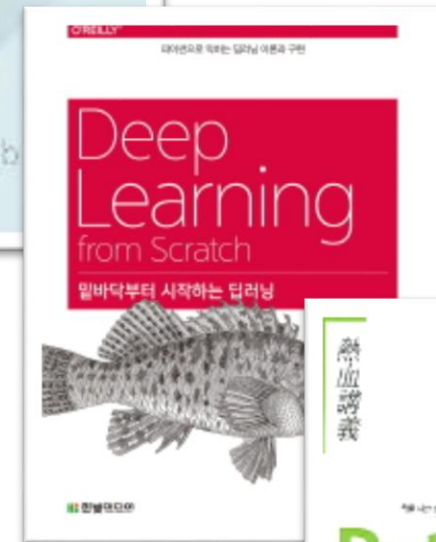
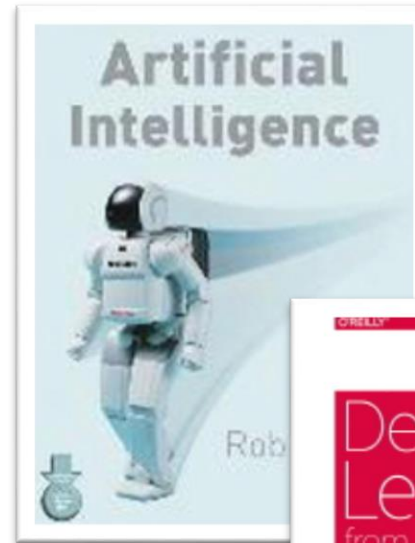
- 강의노트 (PPT)

- 매 주 목요일 에듀클래스 배포

- 부교재

- Artificial Intelligence

- (Rob Callan 저, Palgrave)



- 참고서적

- 밑바닥부터 시작하는 딥러닝

- (사이토 고키 저, 개앞맵시 역, 한빛미디어)

- 열혈강의 파이썬

- (이강성 저, 프리렉)



평가 방법

- 성적 평가 기준

- 출석(10%)
- 참여도(10%): 수업 참여도에 따라 마일리지 부여
- 수시과제(20%): 매 주 실습 과제 평가
- 기말과제(30%): 개인별 텀프로젝트(25%) + 보고서(설계 5%) 제출
- 중간시험(0%): 중간고사 없음
- 기말시험(30%)



강의 계획

- 화요일 8교시(16시~17시): 강의
 - 110호 강의실
 - 출석 체크
 - 해당 주차 주제 강의
- 목요일 6~7교시(14시~16시): 실습
 - 110호 강의실
 - 출석 체크
 - 실습 과제 설명 후 실습실로 이동
 - 326호 / 328호 실습실 (windows 10 / 내장 GPU)
 - 출석 체크 X
 - 늦게 온 경우 지각 처리는 가능
 - 자유롭게 이용
 - 개인 노트북 지참 / 집 데스크톱 실습 인정



강의 계획

주차	강의	실습
1	과목 소개 인공지능 및 python 소개 Python 프로그래밍 환경 조성 및 기본 문법	간단한 expert system 구현
2	Search (Blind Search, Heuristic Search)	DFS, BFS, A* 구현
3	Learning / Simulated Annealing	Simulated Annealing 구현
4	Genetic Algorithm	Genetic Algorithm 구현
5	보강 주간 (개천절 / 추석 연휴)	
6	Neural Network	Perceptron 구현
7	Neural Network	Single-Layer Perceptron 구현
8	Neural Network	Multi-Layer Perceptron 구현
9	Deep Neural Network	DNN 구현
10	Deep Neural Network	DNN 구현
11	DNN 심화 (혹은 Particle Swarm Optimization)	관련 내용 구현
12	DNN 심화 (혹은 Gaussian Mixture Model)	관련 내용 구현
13	DNN 심화 (혹은 Gaussian Mixture Model)	관련 내용 구현
14	DNN 심화 (혹은 Bayesian Networks)	관련 내용 구현
15	DNN 심화 (혹은 Decision Networks)	관련 내용 구현
16	기말고사	

강의 계획

주차	강의	실습
1	과목 소개 인공지능 및 python 소개 Python 프로그래밍 환경 조성 및 기본 문법	간단한 expert system 구현
2	Search (Blind Search, Heuristic Search)	DFS, BFS, A* 구현
3	Learning / Simulated Annealing	Simulated Annealing 구현
4	Genetic Algorithm	Genetic Algorithm 구현
5	보강 주간 (개천절 / 추석 연휴)	
6	Neural Network	Perceptron 구현
7	Neural Network	Single-Layer Perceptron 구현
8	Neural Network	Multi-Layer Perceptron 구현
9	Deep Neural Network	DNN 구현
10	Deep Neural Network	DNN 구현
11	DNN 심화 (혹은 Particle Swarm Optimization)	관련 내용 구현
12	DNN 심화 (혹은 Gaussian Mixture Model)	관련 내용 구현
13	DNN 심화 (혹은 Gaussian Mixture Model)	관련 내용 구현
14	DNN 심화 (혹은 Bayesian Networks)	관련 내용 구현
15	DNN 심화 (혹은 Decision Networks)	관련 내용 구현
16	기말고사	

Python
숙달을 위한
준비 기간

예비 기간
(대체 가능)

유의사항

- 지각 / 결석 / 조퇴
 - 피치 못할 사정이 있는 경우:
 - 사유서 및 증빙자료를 제출하면 출석 인정
- 강의노트 확인 및 과제 제출은 에듀클래스 이용
 - <http://club.uos.ac.kr>
 - UOS 포탈(portal.uos.ac.kr) 통해 접속
- 실습 과제
 - 동작하지 않는 코드는 제출할 필요 없음
 - 기능 구현이 된 경우 만점 / 제대로 안 되면 0점
 - (과제에 따라 부분 점수를 인정하는 경우는 제외)
 - 제출 마감: 다음 실습일 전날 밤 12시까지
 - 마감 기한을 넘겼을 경우 메일로 제출(시간 감점 있음)



유의사항

- 과제 복사 금지(소스코드 / 보고서 등)
 - 1회 적발시 제공자/수혜자 모두 해당 과제 점수 0점 처리
 - 2회 적발시 F학점 부여
 - -> 본인의 의도와 관계 없이 자신의 소스코드가 퍼져 나가지 않도록 수업 종료 후 실습 컴퓨터에서 소스코드 삭제 권장
- 실습실 컴퓨터에 관련 없는 프로그램 설치 금지
- 수업 중 잡담 / 전화 통화 / 게임 / 쇼핑 / 웹서핑 금지
 - 실습 중 옆 사람한테 물어보는 것은 허용(적극 권장)
 - 단, 소스코드를 복사해 주면 안됨
- 대리 출석 금지



인공지능

- 인공지능이란 무엇인가?

네이버 지식백과

인간의 학습능력과 추론능력, 지각능력, 자연언어의 이해능력 등을 컴퓨터 프로그램으로 실현한 기술.

위키피디아(2010)

AI is the **intelligence** of **machines** and the branch of computer science that aims to create it.

위키피디아(2017)

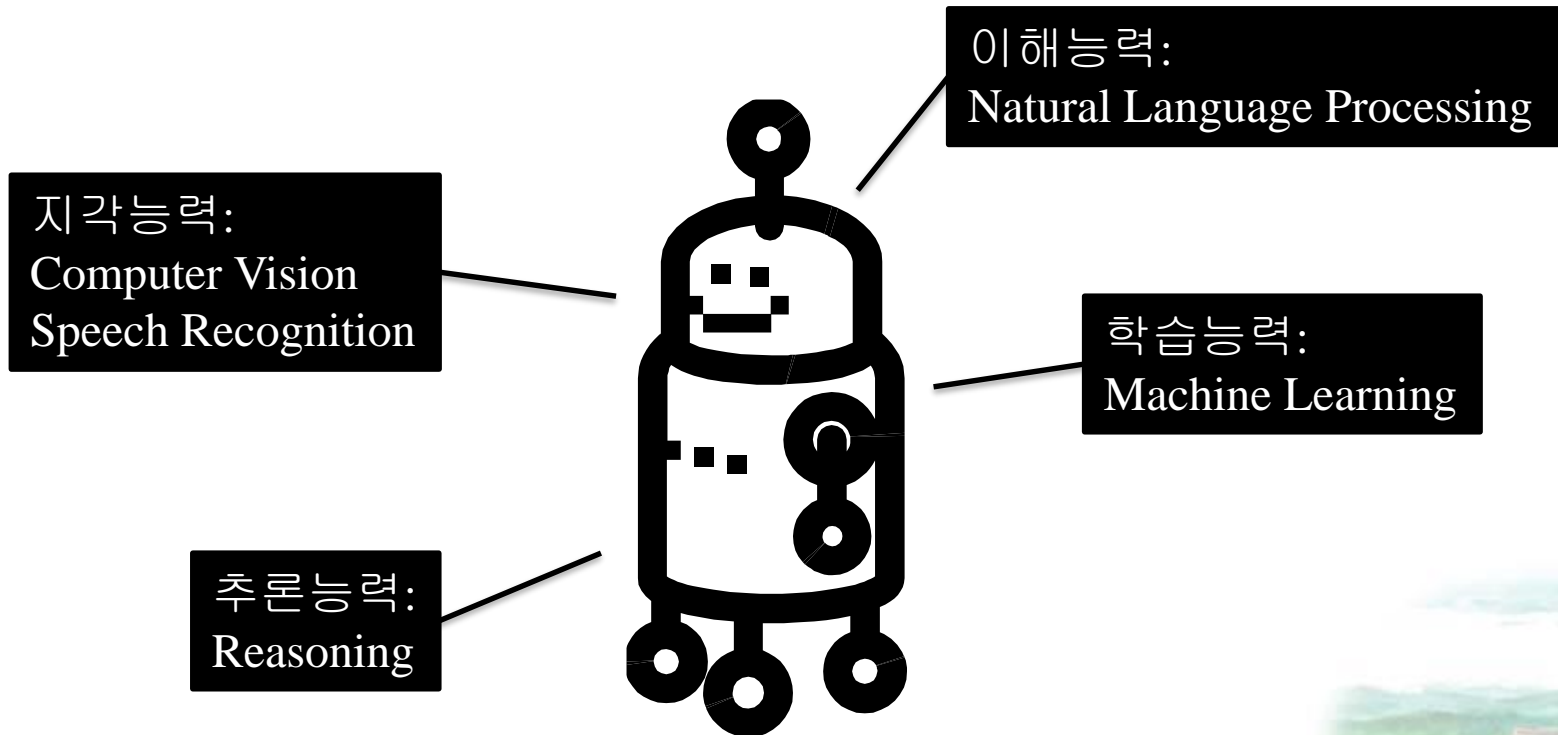
Artificial intelligence (AI, also **machine intelligence**, MI) is **intelligence** exhibited by **machines**, rather than humans or other animals (natural intelligence, NI).

Marvin Minsky(MIT)

AI is the science of making **machines** do things that would require **intelligence** if done by **men**.

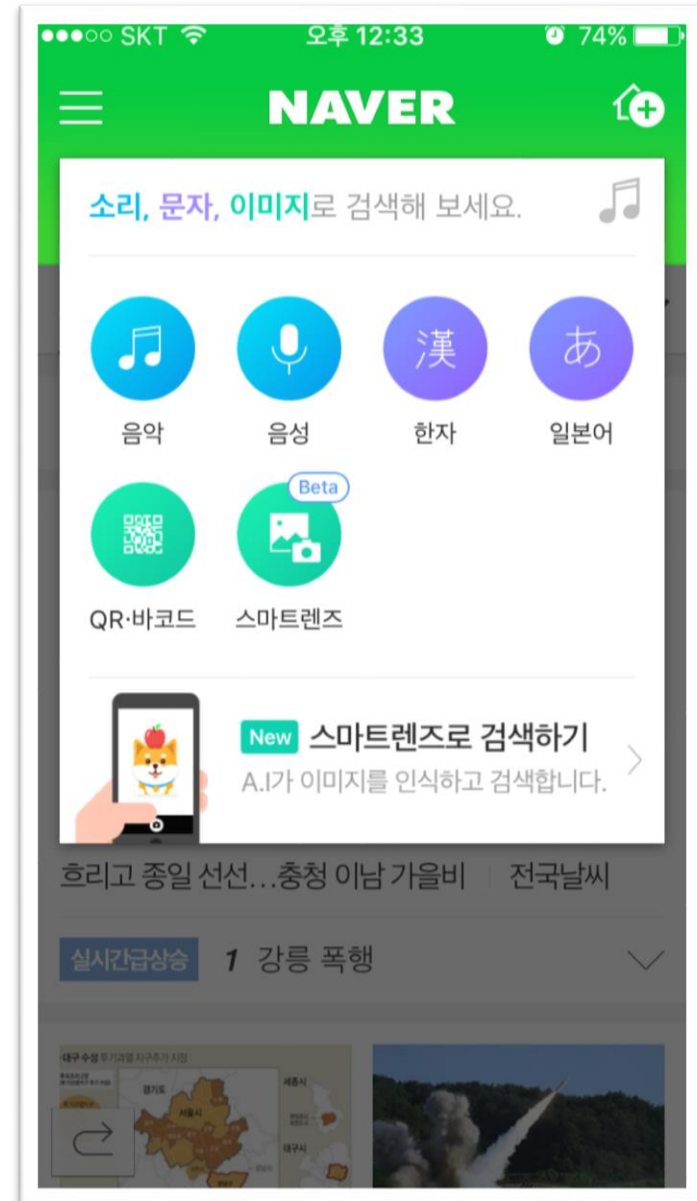
인공지능

- 인공지능이란 무엇인가?



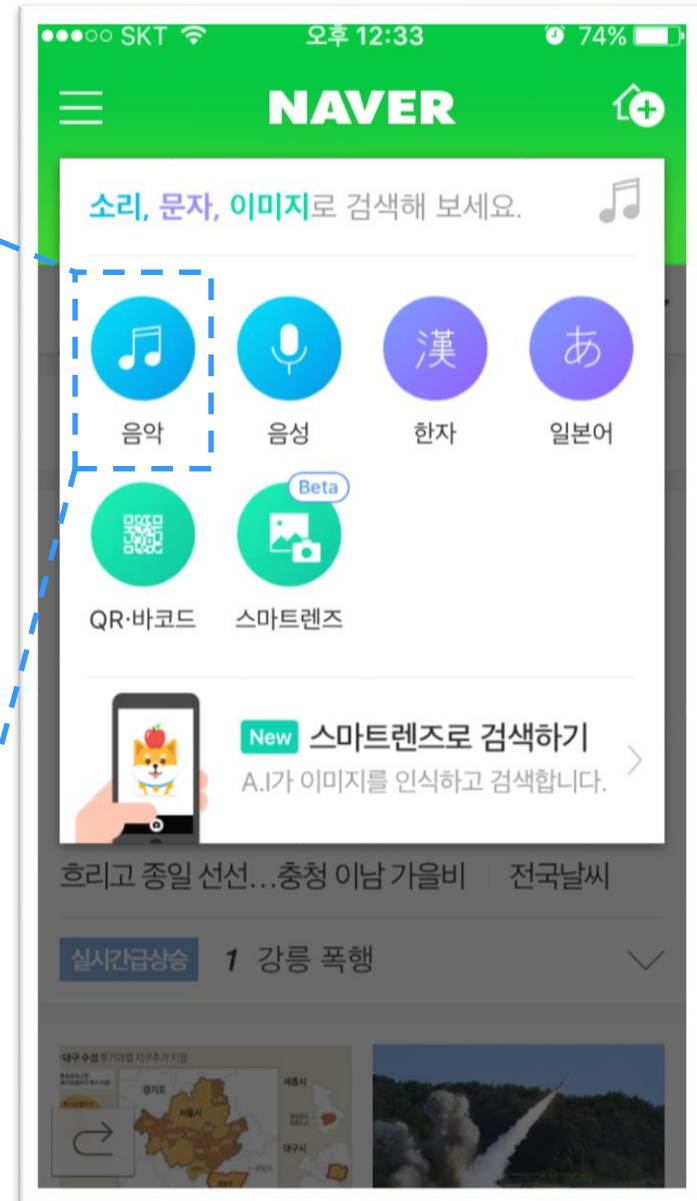
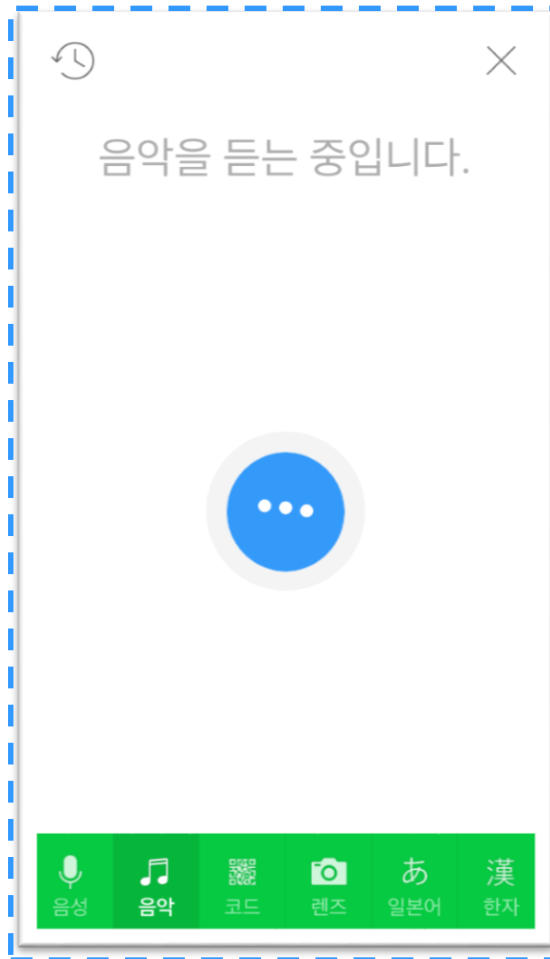
인공지능

- 인공지능의 응용분야
 - 인식



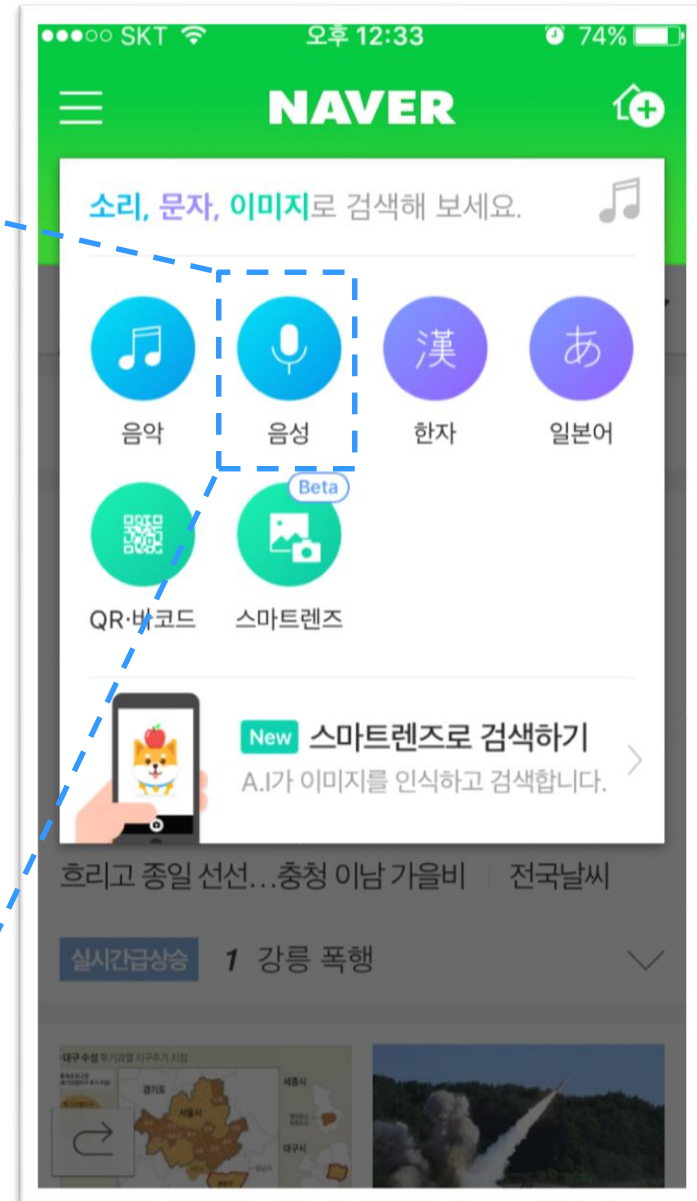
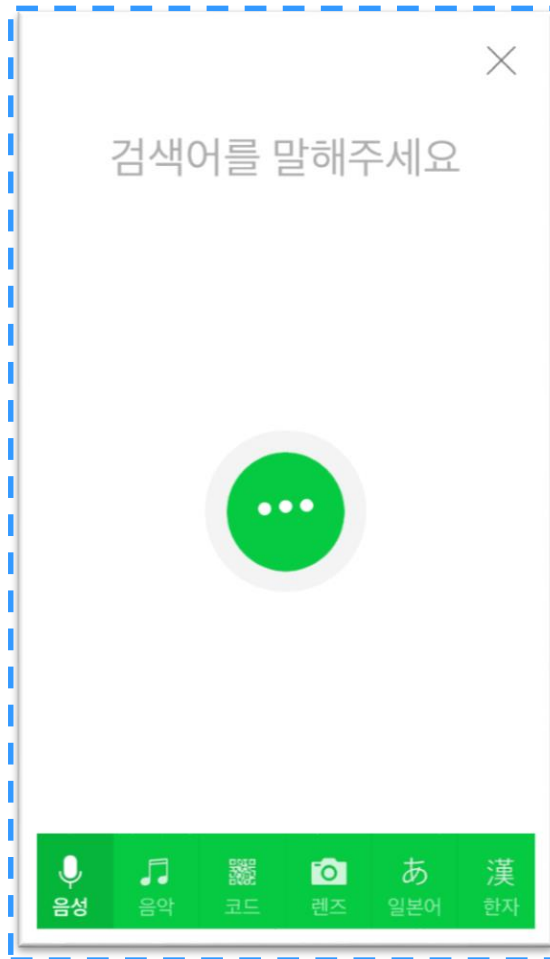
인공지능

- 인공지능의 응용분야
 - 음악 인식



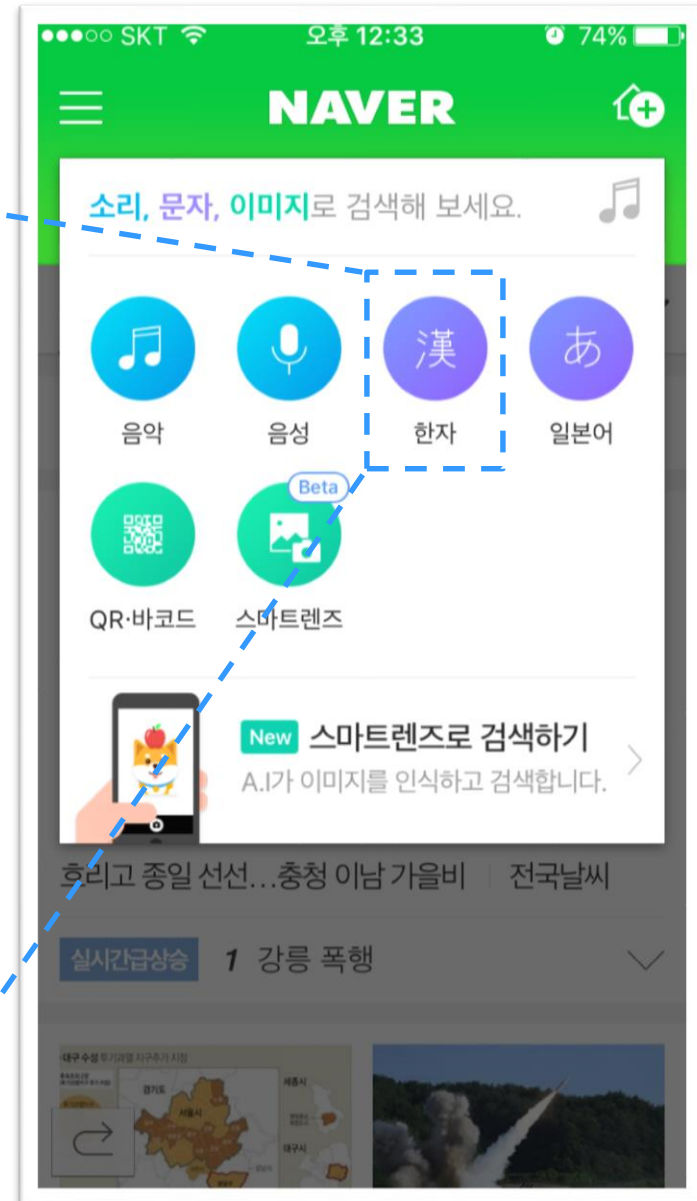
인공지능

- 인공지능의 응용분야
 - 음성 인식



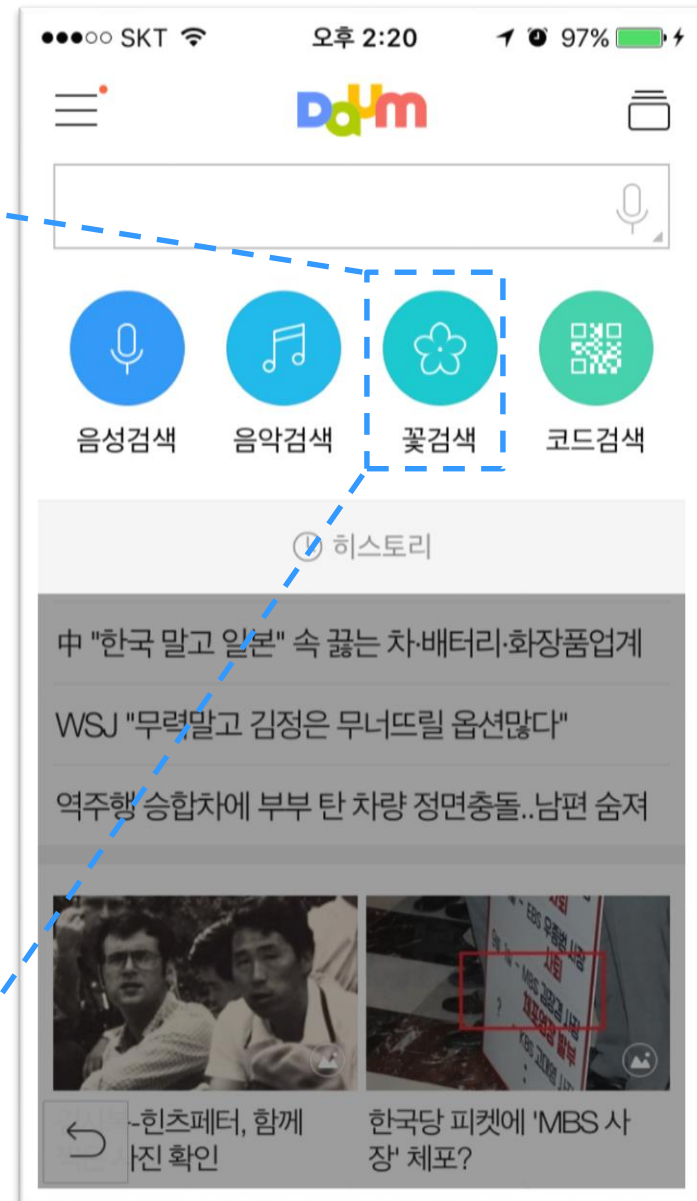
인공지능

- 인공지능의 응용분야
 - 필기 인식



인공지능

- 인공지능의 응용분야
 - 꽃 인식



인공지능

- 인공지능의 응용분야

- 뇌파 인식(Emotiv)

- <https://youtu.be/YxMux4uEkLI?t=63>



인공지능

- 인공지능의 응용분야

- 음성 비서

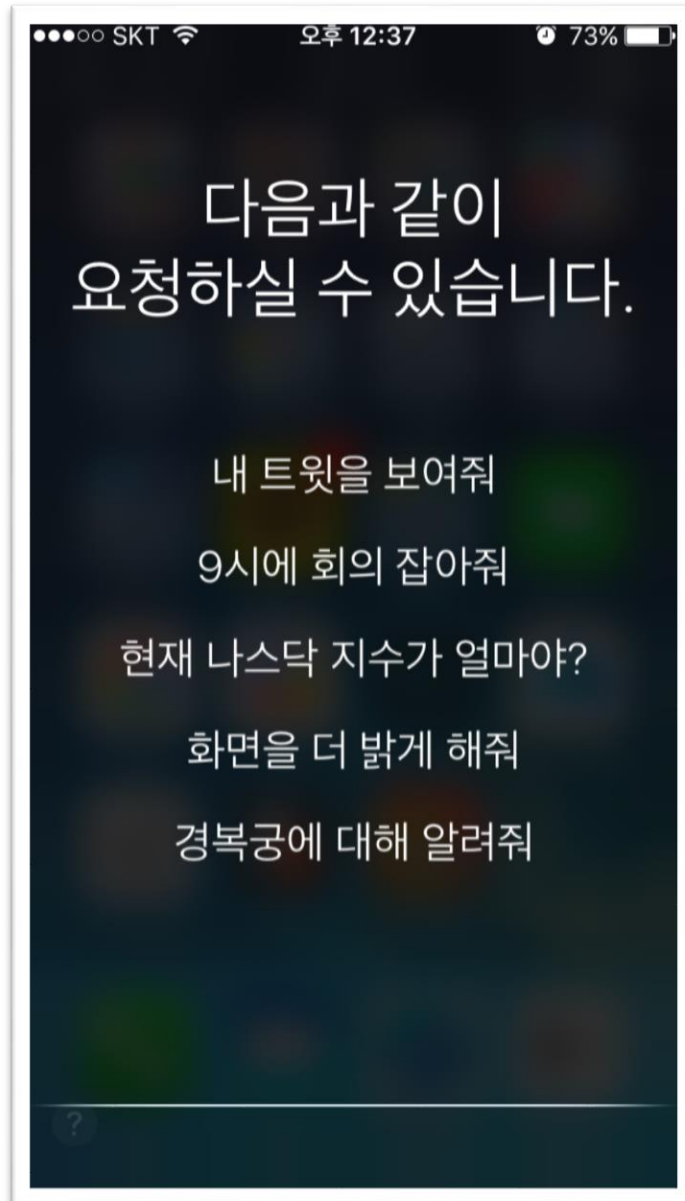
- 음성 인식

- Speech-to-text

- 자연어 처리

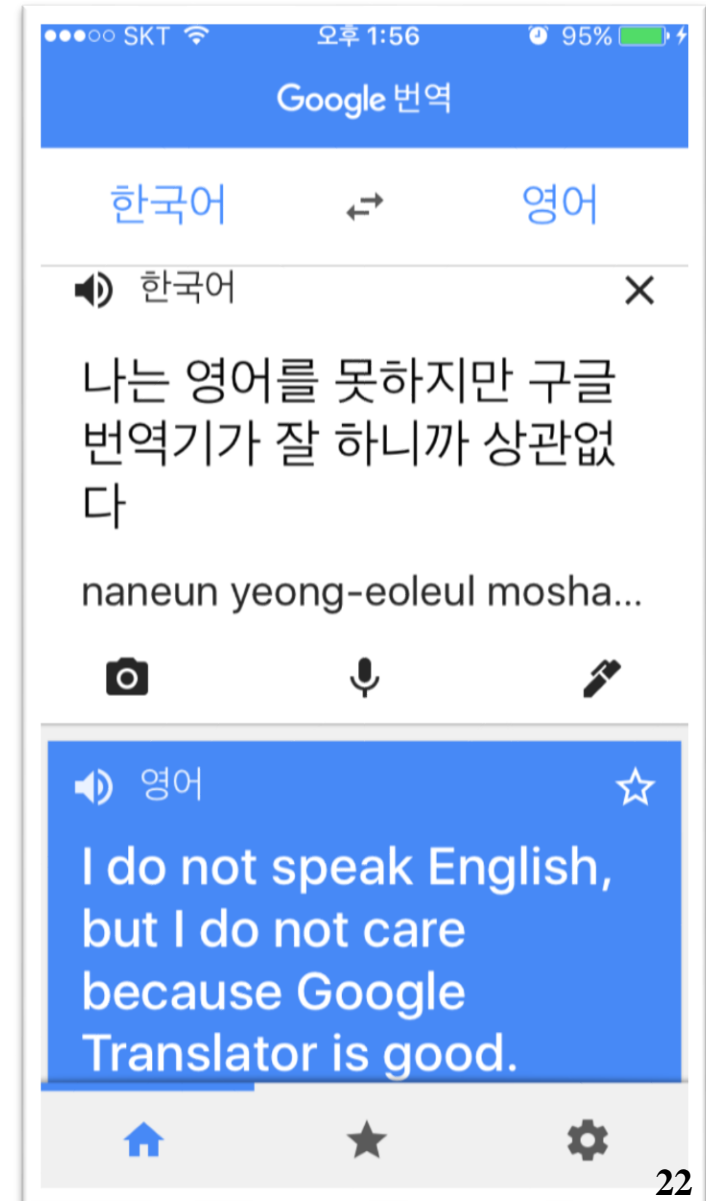
- Natural language processing

- (+ 답변 생성 / 요청 처리)



인공지능

- 인공지능의 응용분야
 - 번역(텍스트)
 - Ex) 한글 → 영문
 - 번역(음성)
 - 음성 인식(한국어 → 한글)
 - Speech-to-text
 - 텍스트 번역(한글 → 영문)
 - Text-to-text
 - 음성 합성(영문 → 영어)
 - Text-to-speech(TTS)



인공지능

- 인공지능의 응용분야
 - 번역(이미지)
 - 영상 인식 + 번역



인공지능

- 인공지능의 응용분야
 - 게임
 - 게임 내 캐릭터들의 행동 결정



심즈 3



엘더스크롤 IV

인공지능

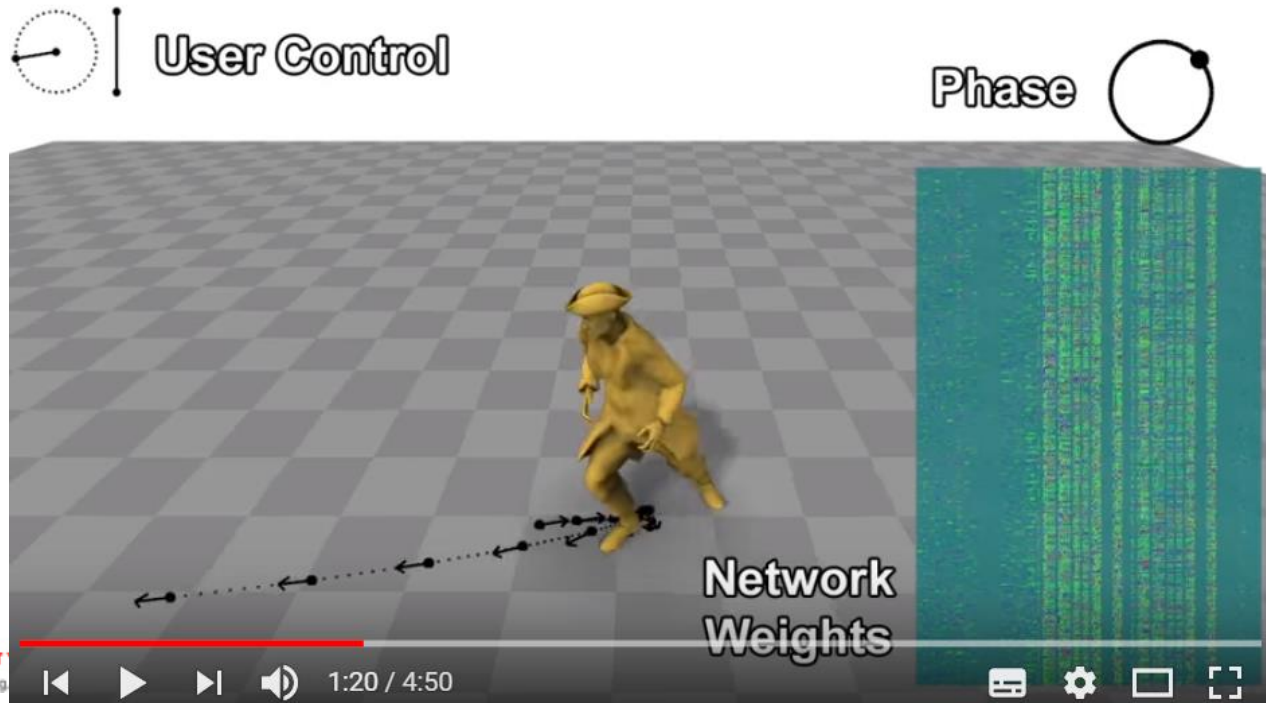
- 인공지능의 응용분야

- 게임/애니메이션

- 캐릭터의 모션 제어

- Holden, D., Komura, T., & Saito, J. (2017). Phase-functioned neural networks for **character control**. ACM Transactions on Graphics (TOG), 36(4), 42.

- <https://youtu.be/Ul0Gilv5wvY?list=PLXYxupSNekFhJip-OEPdVoO6CCPzCB6eE&t=80>



인공지능

- 인공지능의 응용분야
 - 게임
 - 사람처럼 게임을 플레이



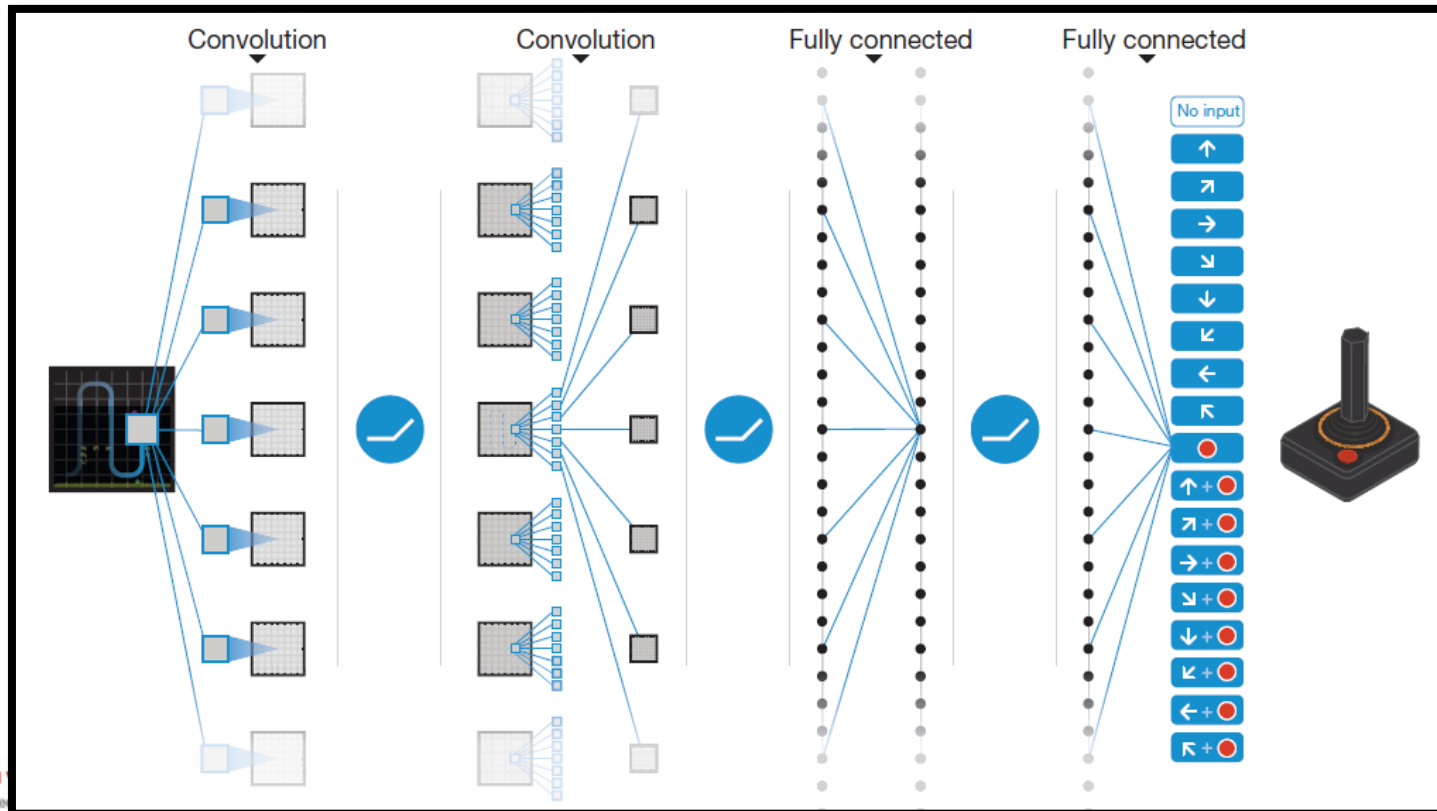
인공지능

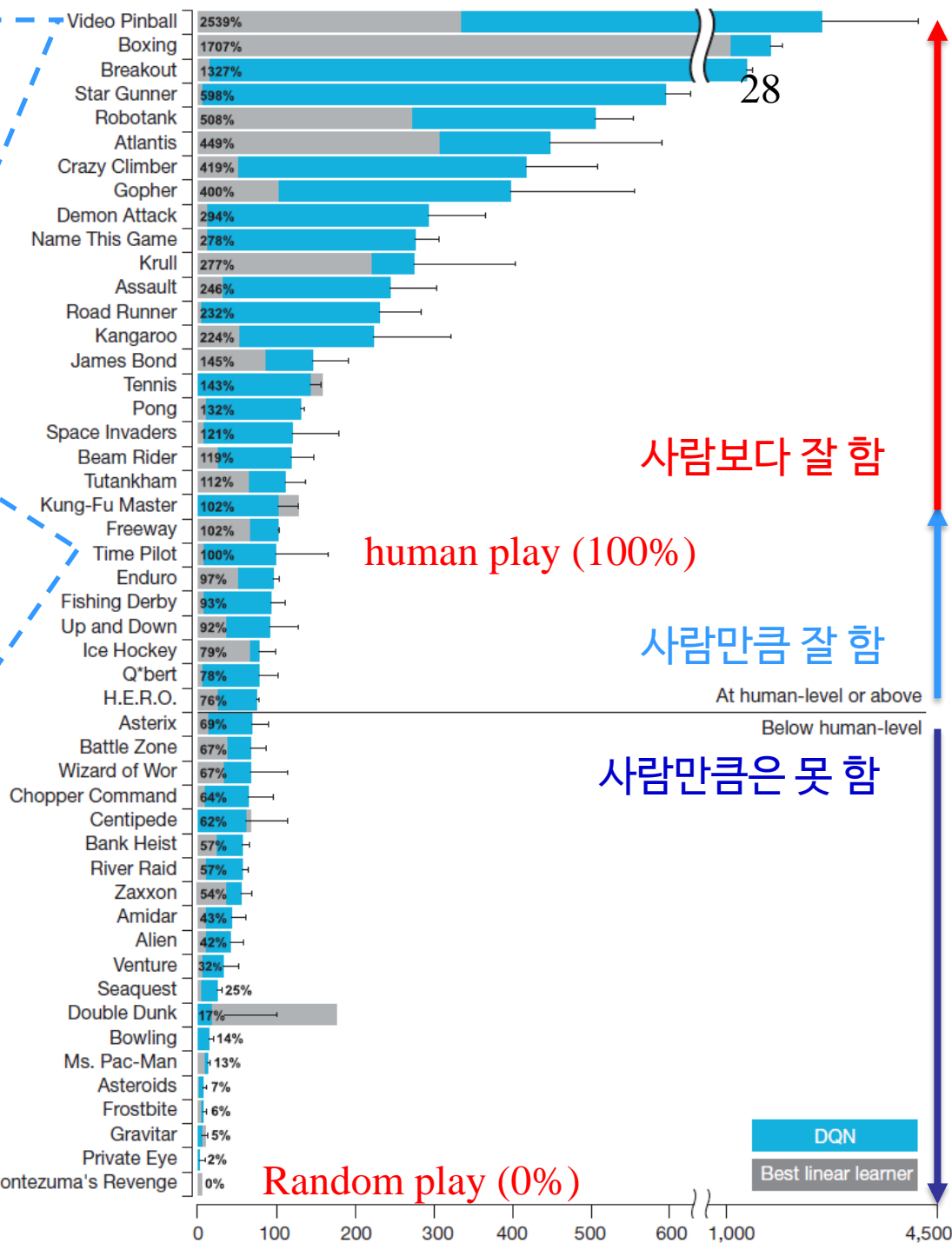
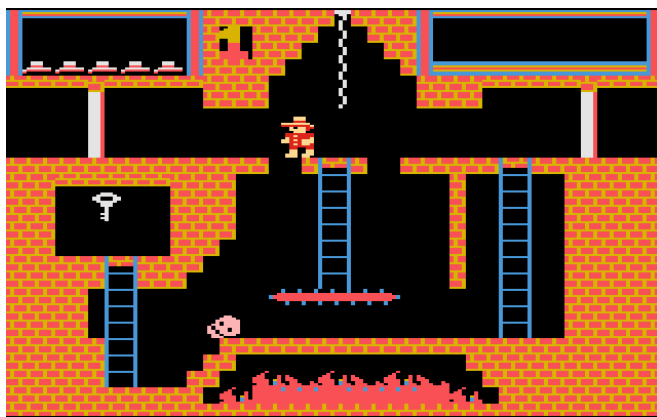
- 인공지능의 응용분야

- 게임

- 사람처럼 게임을 플레이

- Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Petersen, S. (2015). Human-level control through **deep reinforcement learning**. Nature, 518(7540), 529-533.





인공지능

- 인공지능의 응용분야

- 게임

- 사람처럼 게임을 플레이

- Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Petersen, S. (2015). Human-level control through **deep reinforcement learning**. Nature, 518(7540), 529-533.

» 시연영상: <https://youtu.be/iqXKQf2BOSE?t=83>



인공지능

- 인공지능의 응용분야

- 예술

- 이미지 모방

- <https://youtu.be/iV-hah6xs2A?list=PLXYxupSNekFhJip-OEPdVoO6CCPzCB6eE&t=29>



인공지능

- 인공지능의 응용분야

- 예술

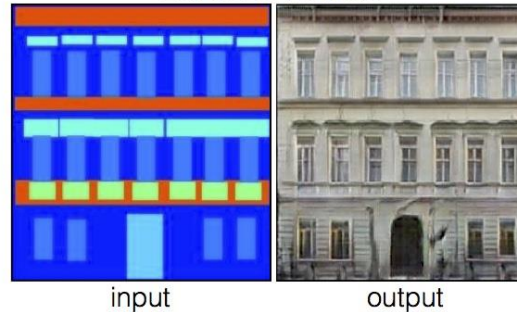
- 이미지 자동완성/생성

- Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2016). **Image-to-image translation** with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*.

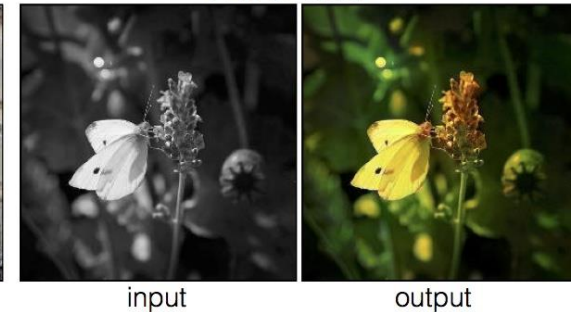
Labels to Street Scene



Labels to Facade



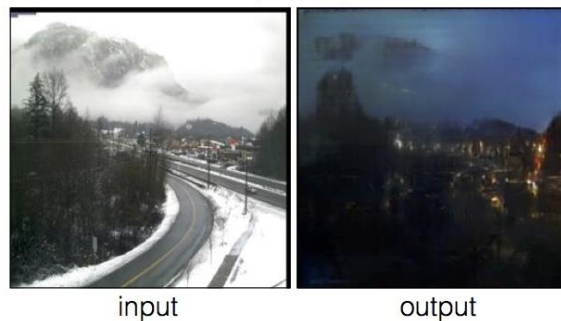
BW to Color



Aerial to Map



Day to Night



Edges to Photo



인공지능

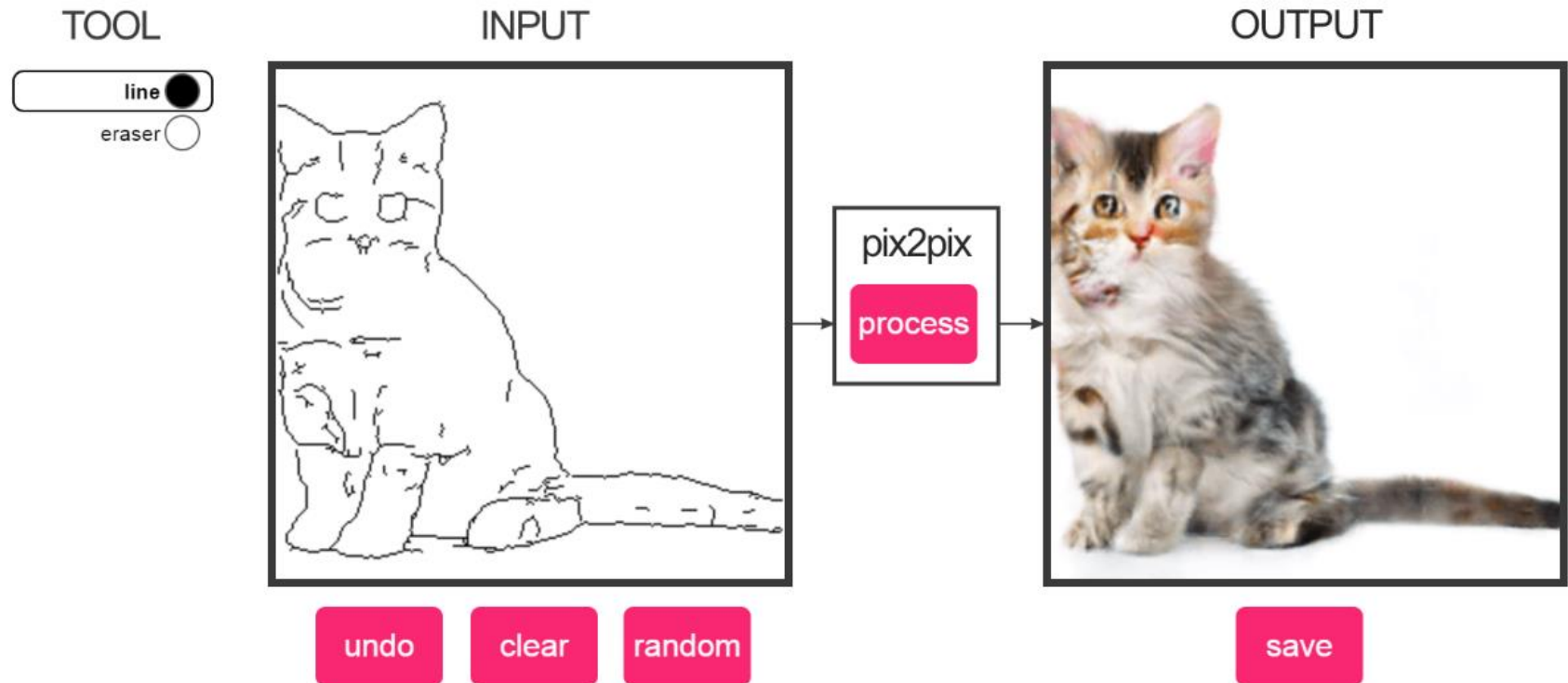
- 인공지능의 응용분야

- 예술

- 이미지 자동완성/생성

- <https://affinelayer.com/pixsrv/>

edges2cats



인공지능

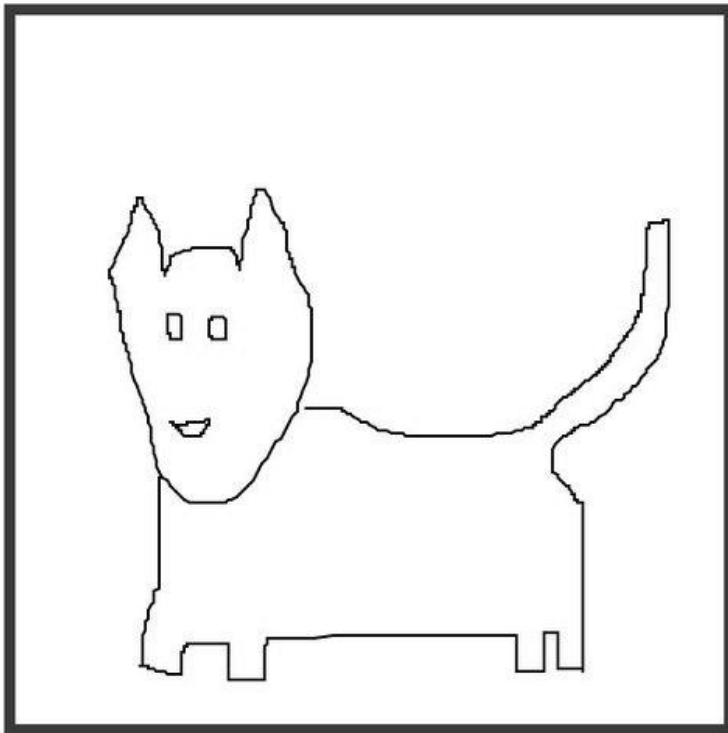
- 인공지능의 응용분야

- 예술...?

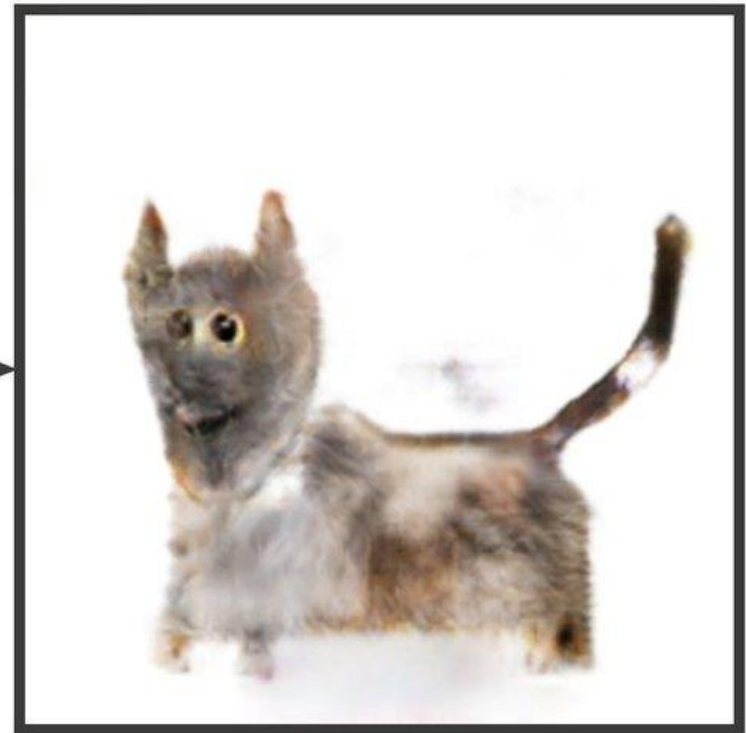
- 이미지 자동완성/생성

- <https://affinelayer.com/pixsrv/>

INPUT



OUTPUT



인공지능

- 인공지능의 응용분야

- 자율주행 자동차

- <https://youtu.be/KSX2psajYrg?list=PLXYxupSNekFhJip-OEPdVoO6CCPzCB6eE&t=1116>



기말과제(2010년 예시)

- 다음의 과제 중 한 가지를 택일
 - 미로 찾기
 - 최적화 알고리즘 시뮬레이터
 - 문자인식기
 - 음성인식기
 - 먹이 찾기 시뮬레이터
 - (자유 과제)
- 각 과제별로 구현 난이도에 따른 점수 차등화



기말과제(2010년 예시)

- 음성인식기

- 0, 1, 2, 3, 4, 5, 6, 7, 8, 9의 숫자 발성을 인식하도록 MLP 학습
 - 학습에 사용한 발성 수에 따른 성능 평가 실험

- 문자인식기

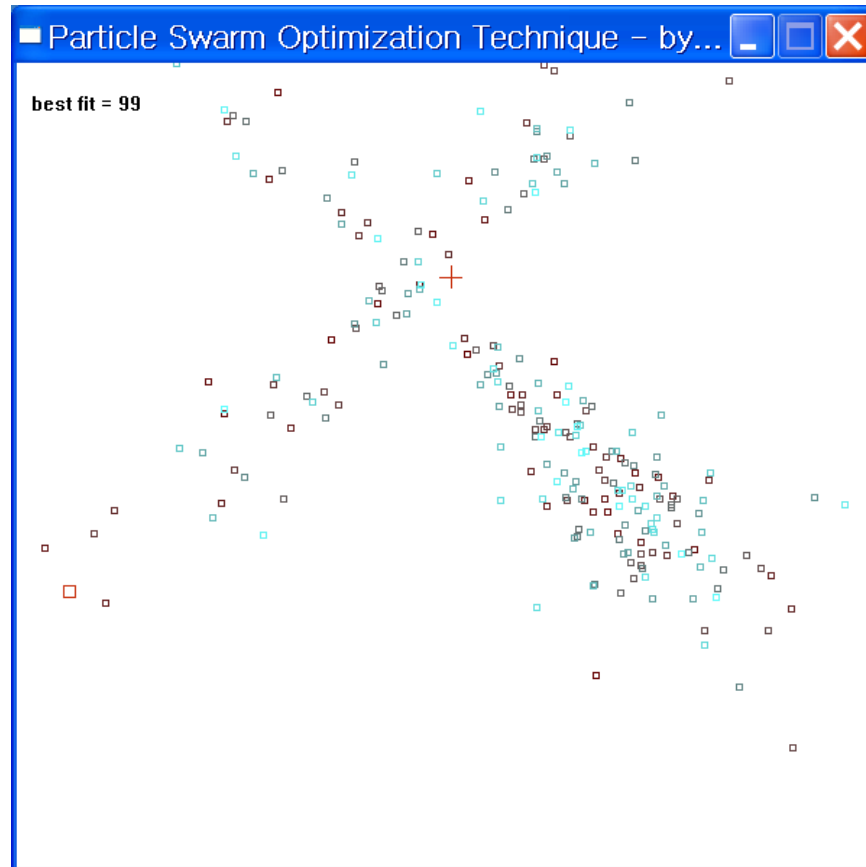
- 0, 1, 2, 3, 4, 5, 6, 7, 8, 9의 숫자 이미지를 인식하도록 MLP 학습
 - 학습에 사용한 이미지 수에 따른 성능 평가 실험



기말과제(2010년 예시)

- 최적화 알고리즘 시뮬레이터

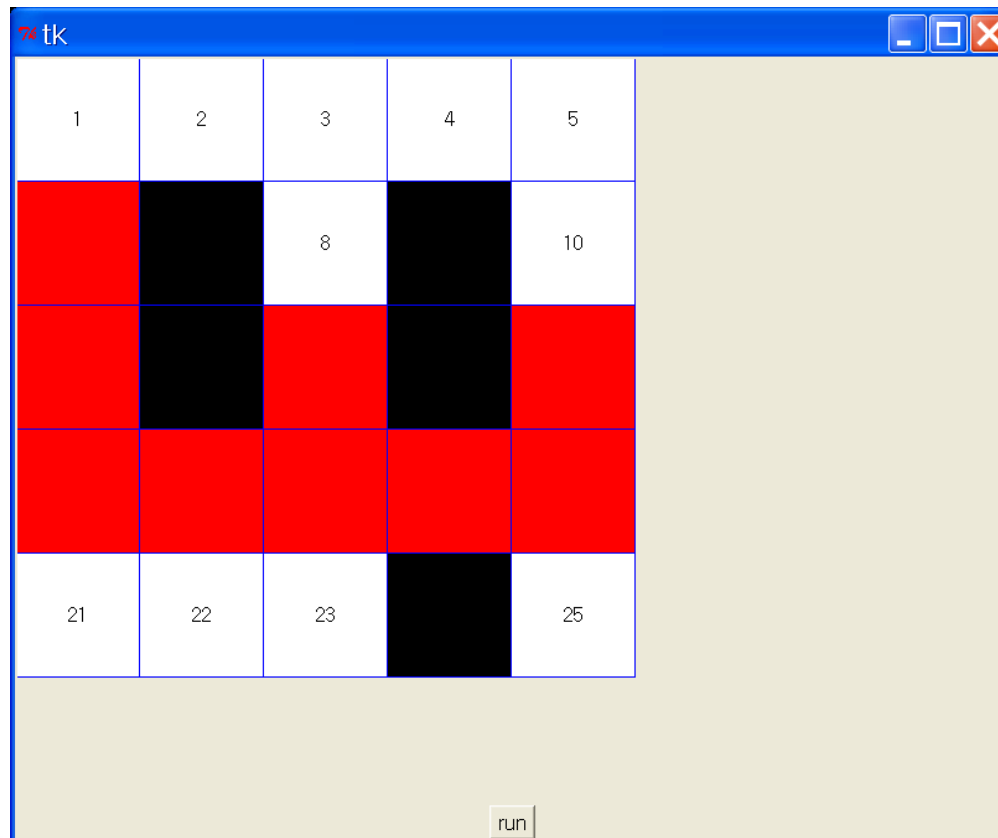
- 2차원 공간상의 좌표를 파라미터로 갖는 개체들이 지정한 목적지로 이동하도록 PSO / GA를 이용하여 탐색
- PSO / GA 성능 비교



기말과제(2010년 예시)

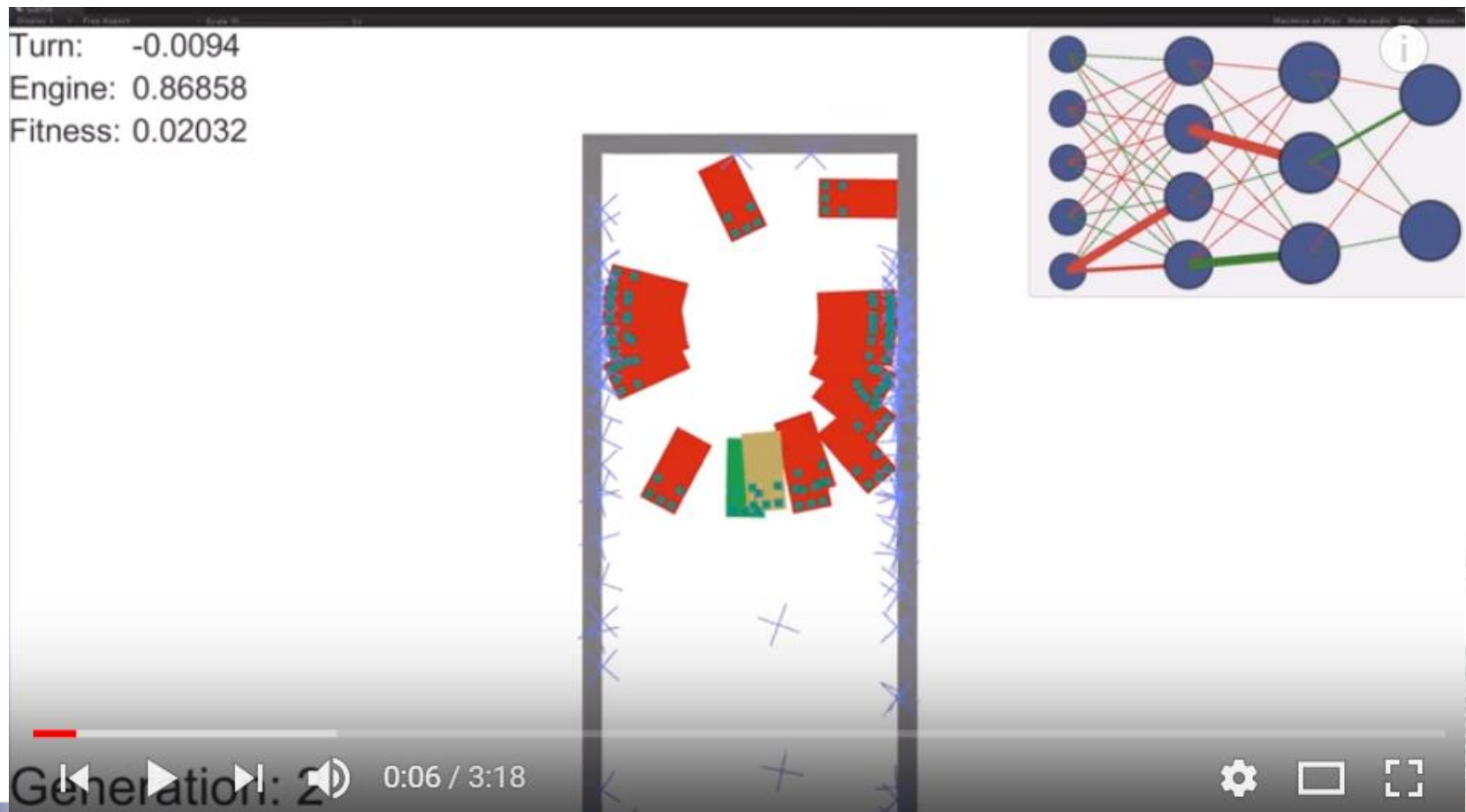
• 미로 찾기

- BFS, DFS, A*로 길찾기 프로그램 구현
- 각각 탐색 속도, 메모리 사용량 비교
- 다양한 형태의 미로에서 시작점과 도착점을 변경해가며 실험



기말과제(예시)

- Genetic algorithm + neural network
 - <https://youtu.be/Aut32pR5PQA?list=PLXYxupSNekFhJip-OEPdVoO6CCPzCB6eE&t=6>



강의 계획

주차	강의	실습
1	과목 소개 인공지능 및 python 소개 Python 프로그래밍 환경 조성 및 기본 문법	간단한 expert system 구현
2	Search (Blind Search, Heuristic Search)	DFS, BFS, A* 구현
3	Learning / Simulated Annealing	Simulated Annealing 구현
4	Genetic Algorithm	Genetic Algorithm 구현
5	보강 주간 (개천절 / 추석 연휴)	
6	Neural Network	Perceptron 구현
7	Neural Network	Single-Layer Perceptron 구현
8	Neural Network	Multi-Layer Perceptron 구현
9	Deep Neural Network	DNN 구현
10	Deep Neural Network	DNN 구현
11	DNN 심화 (혹은 Particle Swarm Optimization)	관련 내용 구현
12	DNN 심화 (혹은 Gaussian Mixture Model)	관련 내용 구현
13	DNN 심화 (혹은 Gaussian Mixture Model)	관련 내용 구현
14	DNN 심화 (혹은 Bayesian Networks)	관련 내용 구현
15	DNN 심화 (혹은 Decision Networks)	관련 내용 구현
16	기말고사	

전문가 시스템(expert system)

- 간단한 인공지능 예시
 - 물고기 종류 구분

어떤 가공식품 공장에서 식재료인 연어와 농어를 한꺼번에 납품 받는다.

어부들이 보내온 납품 박스 안에는 연어와 농어가 섞여있는데,
공정에 들어가기 전에 이것을 종류별로 분류해야 한다.

지금까지는 사람의 손으로 일일이 분류했지만 시간이 너무 많이 걸리므로
이제는 컴퓨터가 사람 대신 분류할 수 있도록 자동화하고 싶다.

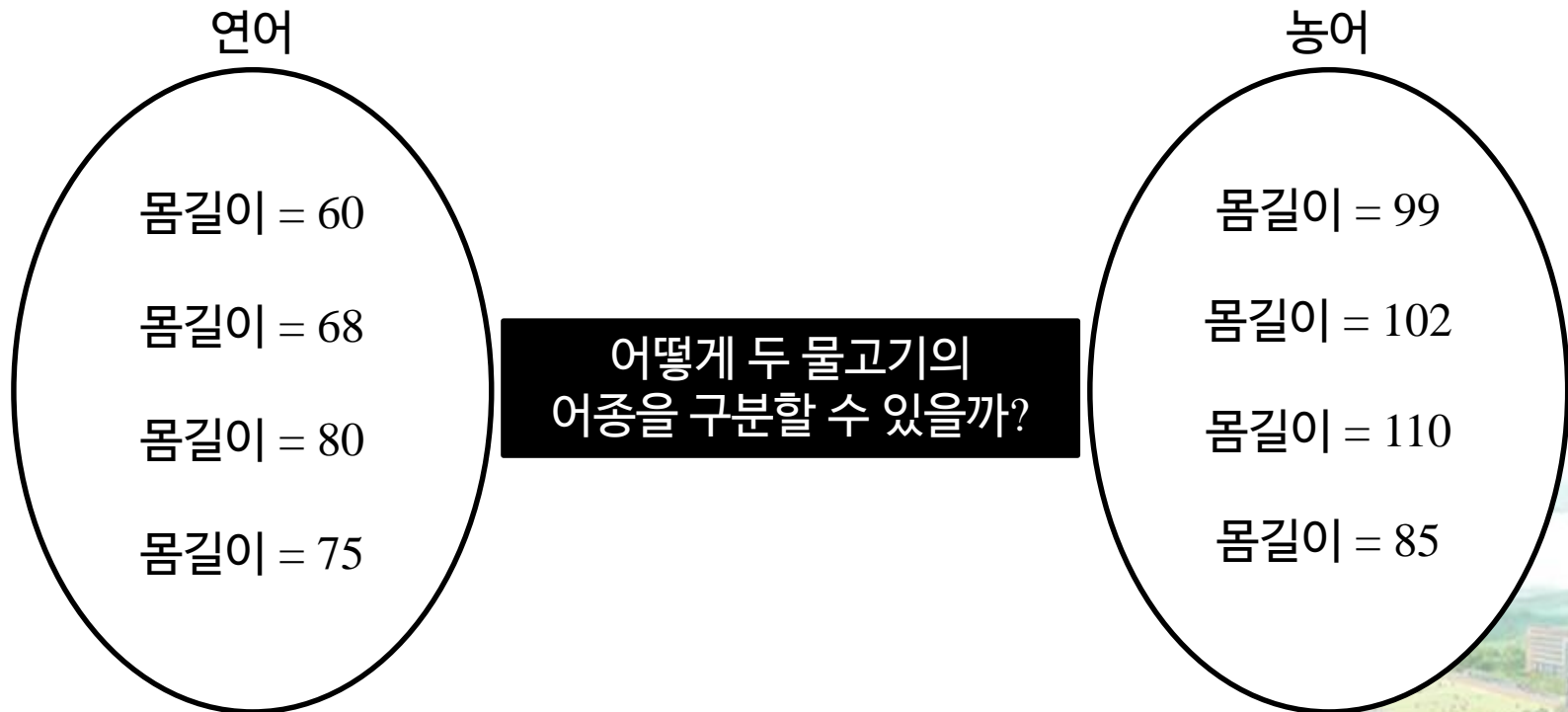
공장에서는 인공지능 어종 분류기를 제작하기 위한 프로젝트에
어류 전문가인 여러분을 초빙하였다.

여러분은 어떻게 두 종류의 물고기를 분류할 것인가?



전문가 시스템(expert system)

- 간단한 인공지능 예시
 - 물고기 종류 구분
 - 몸길이를 특징으로 사용하는 경우



전문가 시스템(expert system)

- 간단한 인공지능 예시
 - 물고기 종류 구분
 - 몸길이를 특징으로 사용하는 경우

연어

몸길이 = 60

몸길이 = 68

몸길이 = 80

몸길이 = 75

```
if 몸길이 < x:  
    return '연어'  
else if 몸길이 >= x:  
    return '농어'
```

농어

몸길이 = 99

몸길이 = 102

몸길이 = 110

몸길이 = 85

전문가 시스템(expert system)

- 간단한 인공지능 예시
 - 물고기 종류 구분
 - 몸길이만으로 구분하기 어려운 경우

연어

몸길이 = 60

몸길이 = 68

몸길이 = 80

몸길이 = 75

몸길이 = 84

```
if 몸길이 < x:  
    return '연어'  
else if 몸길이 >= x:  
    return '농어'
```

농어

몸길이 = 99

몸길이 = 102

몸길이 = 110

몸길이 = 85

몸길이 = 82

전문가 시스템(expert system)

- 간단한 인공지능 예시
 - 물고기 종류 구분
 - 또 다른 특징을 함께 고려
 - Ex) 꼬리 길이

연어

몸길이 = 60	꼬리길이 = 11
몸길이 = 68	꼬리길이 = 18
몸길이 = 80	꼬리길이 = 15
몸길이 = 75	꼬리길이 = 13
몸길이 = 84	꼬리길이 = 20

농어

몸길이 = 99	꼬리길이 = 12
몸길이 = 102	꼬리길이 = 9
몸길이 = 110	꼬리길이 = 5
몸길이 = 85	꼬리길이 = 8
몸길이 = 82	꼬리길이 = 7

전문가 시스템(expert system)

- 간단한 인공지능 예시

- 물고기 종류 구분

- 또 다른 특징을 함께 고려

- Ex) 꼬리 길이

연어

농어

몸길이 = 60 꼬리길이 = 11

몸길이 = 68 꼬리길이 = 18

몸길이 = 80 꼬리길이 = 15

몸길이 = 75 꼬리길이 = 13

몸길이 = 84 꼬리길이 = 20

몸길이 = 99 꼬리길이 = 12

몸길이 = 102 꼬리길이 = 9

몸길이 = 110 꼬리길이 = 5

몸길이 = 85 꼬리길이 = 8

몸길이 = 82 꼬리길이 = 7

if 몸길이 < x and 꼬리길이 > y:

return '연어'

else if 몸길이 >= x and 꼬리길이 <= y:

return '농어'



QnA

실습

Python 개발환경 준비

- 유의사항

- 326호 먼저 / 앞자리부터 빠짐 없이 채워 앉기
 - 다음 번 실습 때 설치 안 된 자리에 앉아서 또 설치해야 되는 상황을 줄입시다...
- Python 및 라이브러리 설치 경로 준수
 - 다음 번 실습 때 다른 자리에 앉더라도 동일한 환경이 될 수 있도록...
- Python 및 라이브러리 버전 준수
 - 버전별로 API 사용법 등이 달라질 수 있으므로...
 - 강사 컴퓨터에서 안 돌면 0점
- 강의에 사용할 버전(2017년 9월 6일 기준 최신)
 - Python 패키지: WinPython-64bit-2.7.13.1Zero
 - Numpy: numpy-1.13.1+mkl-cp27-cp27m-win_amd64
 - Scipy: scipy-0.19.1-cp27-cp27m-win_amd64
 - Theano: 0.9.0
 - Keras: 2.0.8



Python 개발환경 준비

- WinPython

- <http://winpython.sourceforge.net/>
 - “downloads page” 링크 클릭
 - “WinPython 2.7” 링크 클릭
 - “2.7.13.1” 링크 클릭
 - “WinPython-64bit-2.7.13.1Zero.exe” 다운로드 및 설치
 - 설치 경로 변경: C:\SciSoft\WinPython-64bit-2.7.13.1Zero
 - 환경변수 설정: 다음 두 경로를 환경변수(PATH)에 추가
 - C:\SciSoft\WinPython-64bit-2.7.13.1Zero\python-2.7.13.amd64
 - C:\SciSoft\WinPython-64bit-2.7.13.1Zero\python-2.7.13.amd64\Scripts



Python 개발환경 준비

- pip 업데이트
 - 콘솔 창(cmd) 열기
 - 다음 두 명령어 순차 입력
 - “python -m pip install -U pip”
 - “pip install --upgrade setuptools”



Python 개발환경 준비

- Numpy / scipy 설치
 - <http://www.lfd.uci.edu/~gohlke/pythonlibs/>
 - 다음 두 파일 다운로드
 - “numpy-1.13.1+mkl-cp27-cp27m-win_amd64.whl”
 - “scipy-0.19.1-cp27-cp27m-win_amd64.whl”
 - 콘솔 창(cmd) 열기
 - 다운로드 위치로 경로 이동
 - 다음 두 명령어 순차 입력
 - “pip install numpy-1.13.1+mkl-cp27-cp27m-win_amd64.whl”
 - “pip install scipy-0.19.1-cp27-cp27m-win_amd64.whl”



Python 개발환경 준비

- Theano / Keras 설치
 - 콘솔 창(cmd) 열기
 - 다음 두 명령어 순차 입력
 - “pip install theano”
 - “pip install keras”
- Keras backend engine 환경 설정
 - 다음 코드를 한번 실행(keras 한번 이상 실행해야 .keras 폴더가 생성됨)

```
import keras
```

- “시작” → “사용자 이름(개인 폴더)” 열기
 - Window 10의 경우: c:\Users\사용자 이름\keras
 - 간혹 C:\SciSoft\WinPython-64bit-2.7.13.1Zero\settings 아래에 생성됨
- “.keras/keras.json” 파일 메모장으로 열기
- “backend”를 “theano”로 변경 후 저장



Python 개발환경 준비

- TDM-GCC 설치
 - <http://tdm-gcc.tdragon.net/>
 - “tdm-gcc-5.1.0-3.exe” 다운로드 및 설치
 - “MinGW-w64/TDM64 (32-bit and 64-bit)” 선택
 - 설치 경로 변경: C:\SciSoft\TDM-GCC-64
 - “Components → gcc” 세부 옵션에서 “openmp” 활성화
 - 환경변수 설정: 다음 경로를 환경변수(PATH)에 추가
 - C:\SciSoft\TDM-GCC-64\bin



Python 개발환경 준비

- Notepad++ 설치

- <https://notepad-plus-plus.org/download/v7.5.1.html>
 - 32bit용 다운로드 및 설치
 - “? → 플러그인 받기” 선택
 - “NppExec” 플러그인 다운로드
 - NppExec_053_dll_Unicode.zip
 - 압축 해제
 - Notepad++의 “설정 → 가져오기 → 플러그인 가져오기...” 선택
 - “플러그인” 메뉴에 “NppExec” 항목이 생기지 않는 경우
 - Notepad++를 관리자 권한으로 실행하여 플러그인 가져오기
 - 압축 해제한 경로의 “NppExec.dll” 선택



Python 개발환경 준비

- Notepad++ / python 스크립트 실행 커맨드 설정
 - “설정 → 환경 설정...” 클릭
 - “언어 메뉴 → 탭 공백으로 바꾸기” 활성화
 - “플러그인 → NppExec → Follow \$(CURRENT_DIRECTORY)” 활성화
 - “F6” 눌러서 실행 커맨드창에 다음과 같이 입력

NPP_SAVE

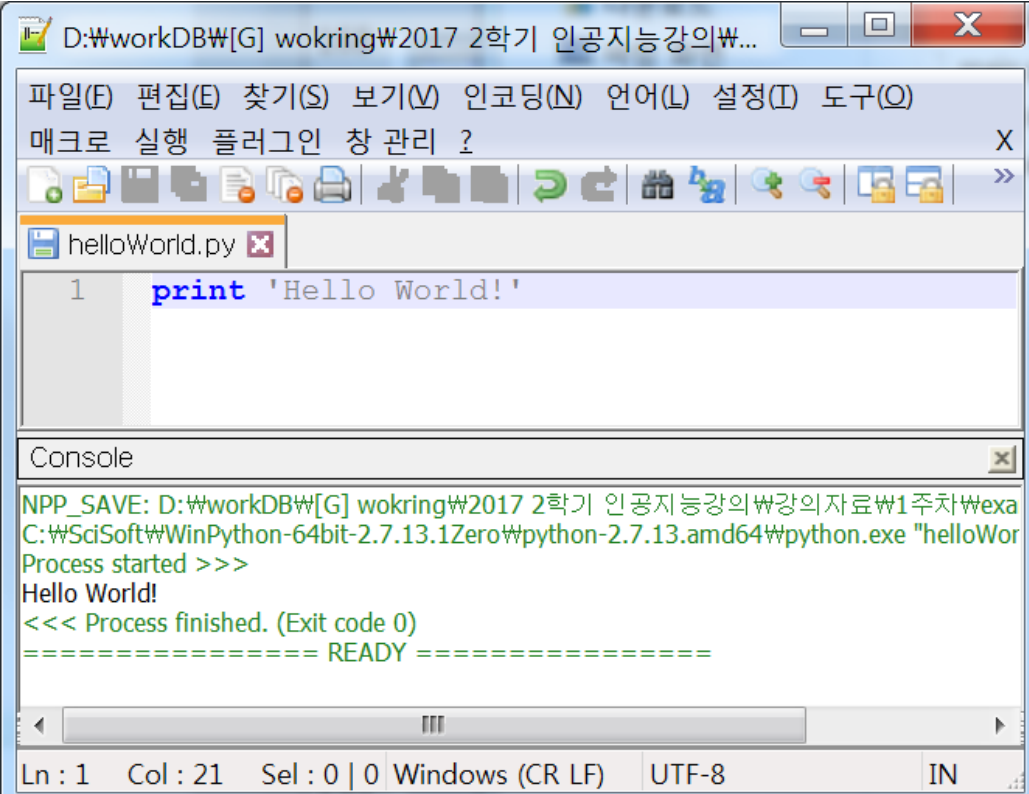
C:\SciSoft\WinPython-64bit-2.7.13.1Zero\python-2.7.13.amd64\python.exe "\$(FILE_NAME)"

- “Save” 클릭
- “run_py27”로 커맨드 저장



Python 개발환경 준비

- Python 스크립트 생성 및 실행 방법
 - 텍스트 파일 생성
 - *.py로 확장자 바꾸기
 - 우클릭 메뉴 → notepad++로 열기
 - “인코딩 → UTF-8로 변환”
 - 스크립트 작성
 - “F6”으로 실행



The screenshot shows a Notepad++ window titled "D:\workDB\G] wokring\2017 2학기 인공지능강의\...". The menu bar includes "파일(F)", "편집(E)", "찾기(S)", "보기(V)", "인코딩(N)", "언어(L)", "설정(I)", and "도구(O)". The toolbar contains various icons for file operations and editing. The editor shows a file named "helloWorld.py" with the following code:

```
1 print 'Hello World!'
```

Below the editor is a "Console" window showing the execution output:

```
NPP_SAVE: D:\workDB\G] wokring\2017 2학기 인공지능강의\...\자료\1주차\hexa
C:\SciSoft\WinPython-64bit-2.7.13.1Zero\python-2.7.13.amd64\python.exe "helloWor
Process started >>>
Hello World!
<<< Process finished. (Exit code 0)
===== READY =====
```

The status bar at the bottom indicates "Ln : 1 Col : 21 Sel : 0 | 0 Windows (CR LF) UTF-8 IN".

Python 개발환경 준비

- Python 에디터 및 실행 방법
 - 아무 텍스트 에디터로나 편집 → cmd 창 / batch 파일 이용하여 실행
 - “python [스크립트파일명]” 커맨드 입력
 - Notepad++ (NppExec)
 - 장점: 메모장보다는 편함
 - 단점: 실행 중 동적 입력이 발생하는 경우(ex) input() 등) 결과 출력이 지연됨
 - → 이런 코드의 경우 cmd 커맨드 / batch 파일을 이용하여 실행
 - Eclipse(pydev)
 - 장점: notepad++보다 편함, 런타임 디버깅 가능!
 - 단점: 무거움, workspace 지정, project 생성 등의 절차가 조금 귀찮음
 - PyCharm, ...
 - 어떤 에디터를 사용해도 상관 없음
 - 과제 검사할 때 실행만 잘 되게...



Python 기본문법

- 변수 선언 없음 (형이 가변적)
- 끝에 세미콜론(;)을 붙이지 않아도 됨
- 중괄호({}) 대신 탭(혹은 스페이스 몇 개)으로 구분
- if / for/ while 끝에 콜론(:) 사용
- ‘&&’대신 ‘and’ 사용
- ‘||’대신 ‘or’ 사용
- ‘null’대신 ‘None’ 사용
 - ‘None’은 가급적이면 사용하지 말 것
 - 일부 라이브러리에서 허용하지 않음
- ‘def 함수명(인자):’의 형태로 함수 정의
- 함수 반환값이 여러 개가 될 수 있음
 - Ex) a, b, c, d = func(c)



Python 실습 – 변수 선언

```
#include <stdio.h>
#include <string.h>
```

```
void main()
{
```

```
    int a;
    double b;
    char c[10];
```

```
    a = 10;
    b = 3.14;
    strcpy(c, "apple");
```

```
    printf("%d %f %s", a, b, c);
```

```
}
```

```
a = 10
```

```
b = 3.14
```

```
c = 'apple'
```

```
print a, b, c
```

C / C++

Python

print(a, b, c)도 가능 (python 3 형식)



Python 실습 – 자료구조

- 튜플 : ()
 - 상수형 리스트
- 리스트 : []
 - 배열, 연결리스트
- 사전 : {}
 - 인덱스가 지정된 리스트
 - Key와 value로 구성



Python 실습 – 반복문 (for)

```
a = [1, 3, 4, 56, 100]
```

```
print a
```

```
print a[0], a[1]
```

```
a[0] = 99
```

```
print a
```

```
a = {}
```

```
a['salmon'] = 85
```

```
a['sea bass'] = 100
```

```
print a
```

```
print a['salmon'], a['sea bass']
```

```
a['salmon'] = 99
```

```
print a
```

리스트

사전

Python 실습 – 반복문 (for)

```
#include <stdio.h>
```

```
void main()
```

```
{
```

```
    int sum = 0;
```

```
    for(int i = 1; i <= 100; i++)
```

```
    {
```

```
        sum += i;
```

```
    }
```

```
    printf("%d", sum);
```

```
}
```

C / C++

```
sum = 0
```

```
for i in range(1, 101):
```

```
    sum += i
```

```
print sum
```

Python

range(1, 101) → [1, 2, ..., 100]



Python 실습 – 파일 입출력

```
lines = ['line1', 'line2']  
  
fd = open('temp.txt', 'w')  
for line in lines:  
    fd.write(line + '\n')  
fd.close()
```

```
fd = open('temp.txt', 'r')  
lines = fd.readlines()  
fd.close()  
  
for line in lines:  
    print line
```

파일 읽기

파일 쓰기



Python 실습 – 문자열 구성

```
a = 10  
b = 3.14  
c = 'apple'  
out = '%d %f %s' % (a, b, c)
```

print out

```
a = 10  
b = 3.14  
c = 'apple'  
out = ('%d %f' % (a, b)) + c
```

print out

예시 1

예시 2



Python 실습 – 주식 달기

한 줄 주식

```
a = 10
```

```
b = 3.14
```

```
c = 'apple'
```

```
out = '%d %f %s' % (a, b, c)
```

```
print out
```

예시 1

“

여러 줄 주식

”

```
a = 10
```

```
b = 3.14
```

```
c = 'apple'
```

```
out = '%d %f %s' % (a, b, c)
```

```
print out
```

예시 2



Python 실습 – 기타

```
a = '1 2'  
b = a.split()  
print a  
print b
```

```
a = '1 2 '  
b = a.strip() # 양 끝 공백 제거  
c = b.split() # 공백 기준 분할  
print a  
print b  
print c
```

```
a = [1, 2]  
b, c = a
```

```
print a  
print b  
print c
```

문자열 strip() / split()

여러 인자 한꺼번에 받기



실습과제 평가기준

- 10점 만점

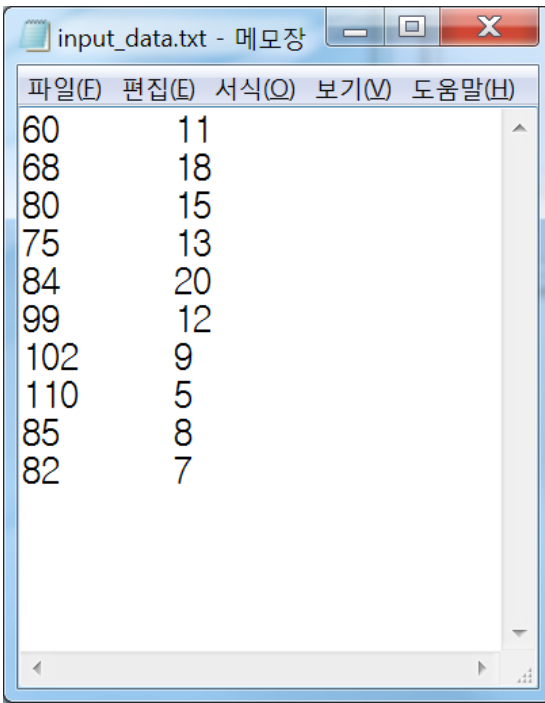
- 기능 구현이 되었으면 10점, 동작하지 않으면 0점
 - 일부 부분 점수가 명시되어 있는 과제의 경우: 최소한 1개 이상의 기능 구현 필요
- 감점요인: 최대 -5점
 - 제출 형식을 따르지 않음(-1점)
 - 과제 설명에서 제시한 제출 형식(ex) 파일명 등)을 따르지 않은 경우
 - 제출 기한 초과(-1점씩):
 - 1주일 넘어서 제출할 때마다 -1점씩
 - 주석 미흡(-1점):
 - 어떤 동작을 하는 프로그램인지 주석을 보고 쉽게 알 수 있도록
 - 프로그램 출력 미흡(-1점):
 - 어떤 동작을 하는 프로그램인지 출력(콘솔 혹은 GUI)을 보고 쉽게 알 수 있도록
 - 예기치 못한 오류 발생 등(-1점씩):
 - Ex) 대체로 잘 동작하나 특정 상황에서 예외 처리가 잘 되어 있지 않아 오류 발생



실습과제

- “[학번].py” 스크립트 작성
 - 다음 기능을 구현(10점)
 - 같은 디렉토리의 “input_data.txt” 파일 읽기
 - 각 행의 데이터에 대해 순차적으로 어종 분류
 - 분류 결과를 같은 디렉토리의 “output_result.txt”에 출력
 - 위 세 파일을 압축하여 에듀클래스로 제출

입력 텍스트 파일



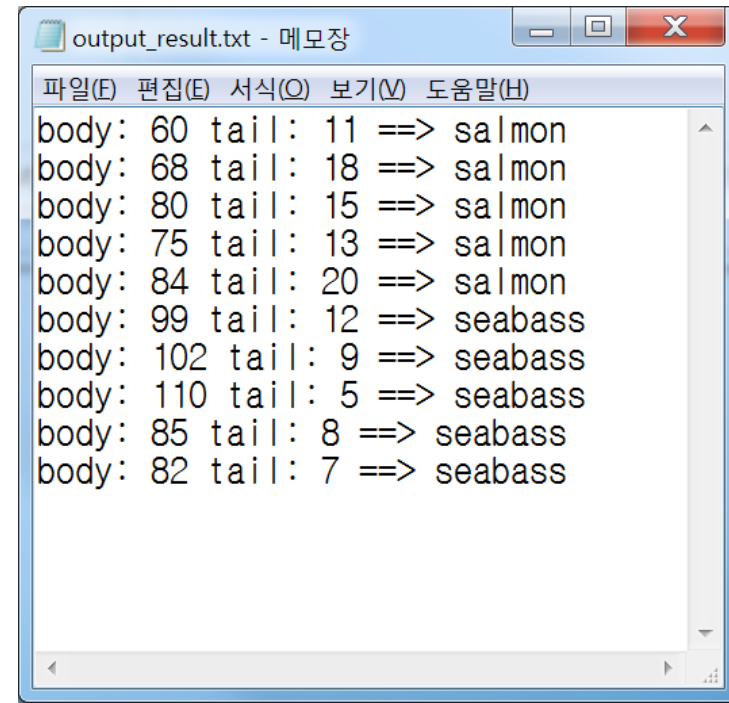
```
60      11
68      18
80      15
75      13
84      20
99      12
102     9
110     5
85      8
82      7
```



[학번].py



출력 텍스트 파일



```
body: 60 tail: 11 ==> salmon
body: 68 tail: 18 ==> salmon
body: 80 tail: 15 ==> salmon
body: 75 tail: 13 ==> salmon
body: 84 tail: 20 ==> salmon
body: 99 tail: 12 ==> seabass
body: 102 tail: 9 ==> seabass
body: 110 tail: 5 ==> seabass
body: 85 tail: 8 ==> seabass
body: 82 tail: 7 ==> seabass
```

QnA