AI 프로그래밍

- 1주차

컴퓨터 정보과

민 정혜

교수 소개

- 민정혜
- jhmin@inhatc.ac.kr

오늘 수업 순서

- **실습실 안전 교육**
- **교과목 개요**
- **인공 지능이란**
  - 정의
  - 적용 분야
  - 기술 분야
- **머신 러닝 프로그래밍**
- **구글 colab 실습**

교과목 개요

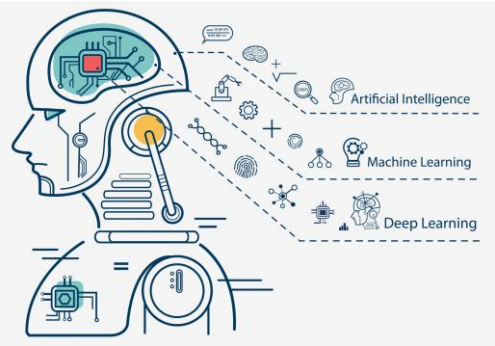
- **AI programming**
  - AI (Artificial Intelligence) 분야 개발에 필요한 programming 능력을 기른다
  - 관련 python library를 이용한 programming 실습

교과목 개요

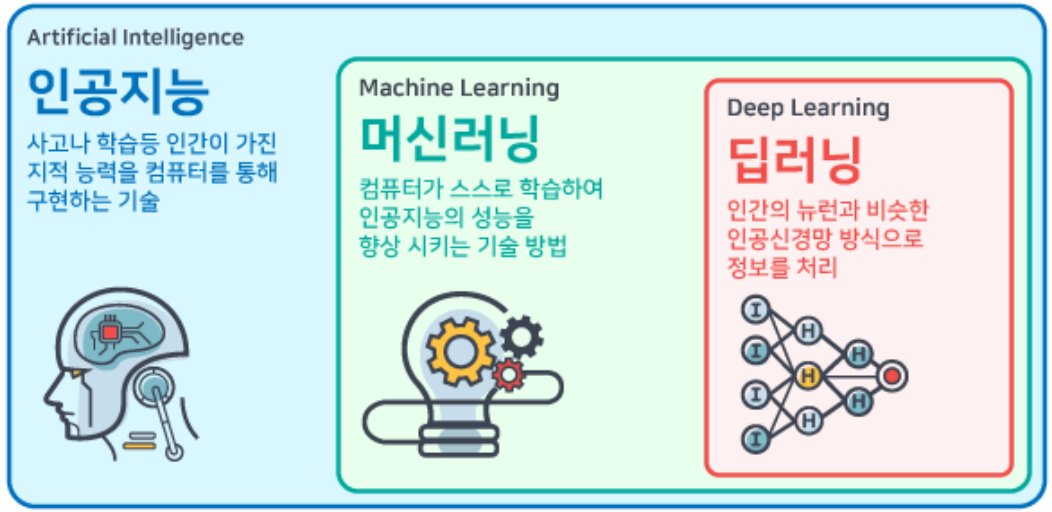
- **목표**
  - AI (인공 지능) 분야 개발자의 기본 역량을 기른다
- **이론강의/실습**
- **교재**
  - 모두의 딥러닝, 길벗 출판사, 2020년 출판
- **평가**
  - 중간 평가 40%, 기말 평가 40%, 출석 20%
- **실습 환경**
  - 구글 Colab (중간 평가 이전)
  - Anaconda (중간 평가 이후)

인공 지능 이란

- **인공 지능** (Artificial Intelligence)
  - **인간**이 가진 **지적 능력**을 **컴퓨터**를 통해 구현 하는 기술
- **머신 러닝** (Machine Learning)
  - 데이터와 **알고리즘**을 통해 컴퓨터를 **학습** 시켜 인공 지능의 성능을 향상 시키는 기술
- **딥 러닝** (Deep Learning)
  - **신경망 (Neural Network)** 을 이용한 머신 러닝 방법.
  - 여러 층으로 이루어진 신경망을 사용.



Machine Learning: A Primer to Laboratory Applications (thermofisher.com)



https://m.blog.naver.com/pwj6971/221614497987

인공 지능 적용 분야

자율주행 자동차

테슬라,현대 자동차

AI 비서

AI 스피커, 가상 비서

챗 봇

AS 상담 챗봇, 호텔 예약 챗봇

로봇

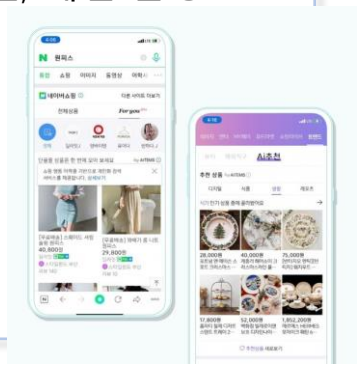
청소로봇, 교육용 로봇

영상 인식

얼굴 인식, 번호판 인식,

개인화 추천

영화 추천, 광고 추천,
상품 추천, 게임 운영



기계 번역

구글 번역, 네이버, 파파고

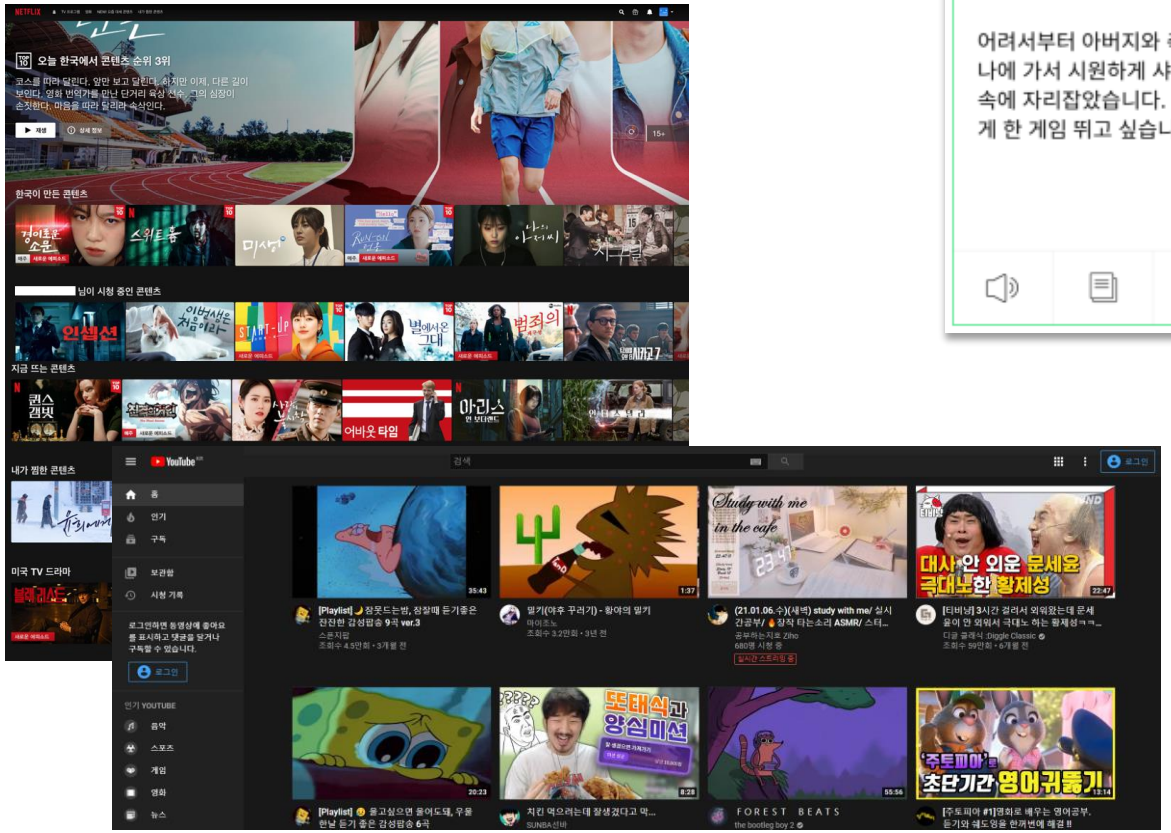
papago

Google Translate

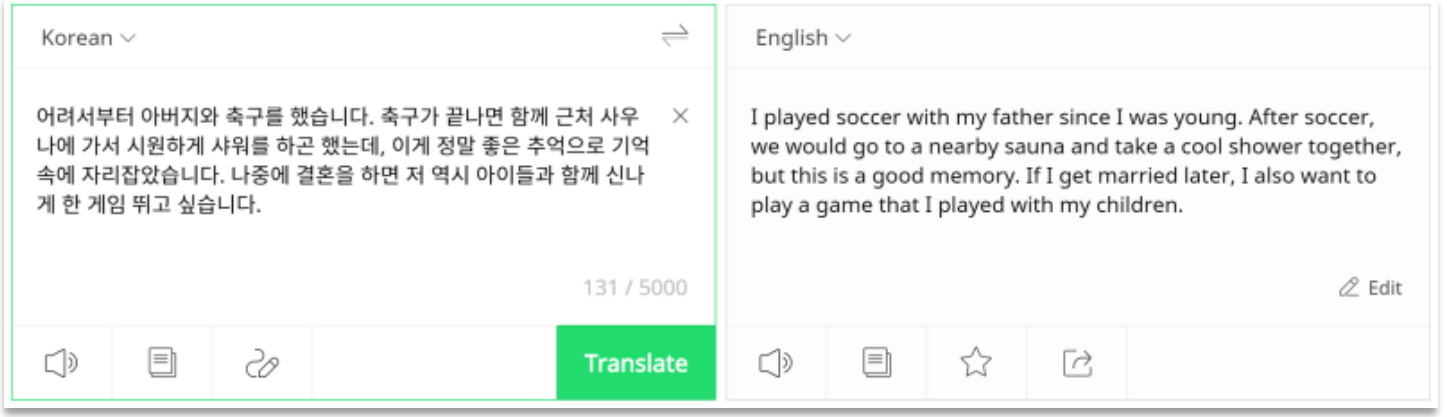
의료

질병 진단

인공 지능 적용 분야



넷플릭스 , 유튜브 콘텐츠 추천

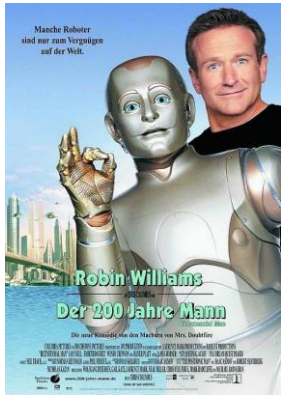


파파고 번역

얼굴 인식 열화상 카메라

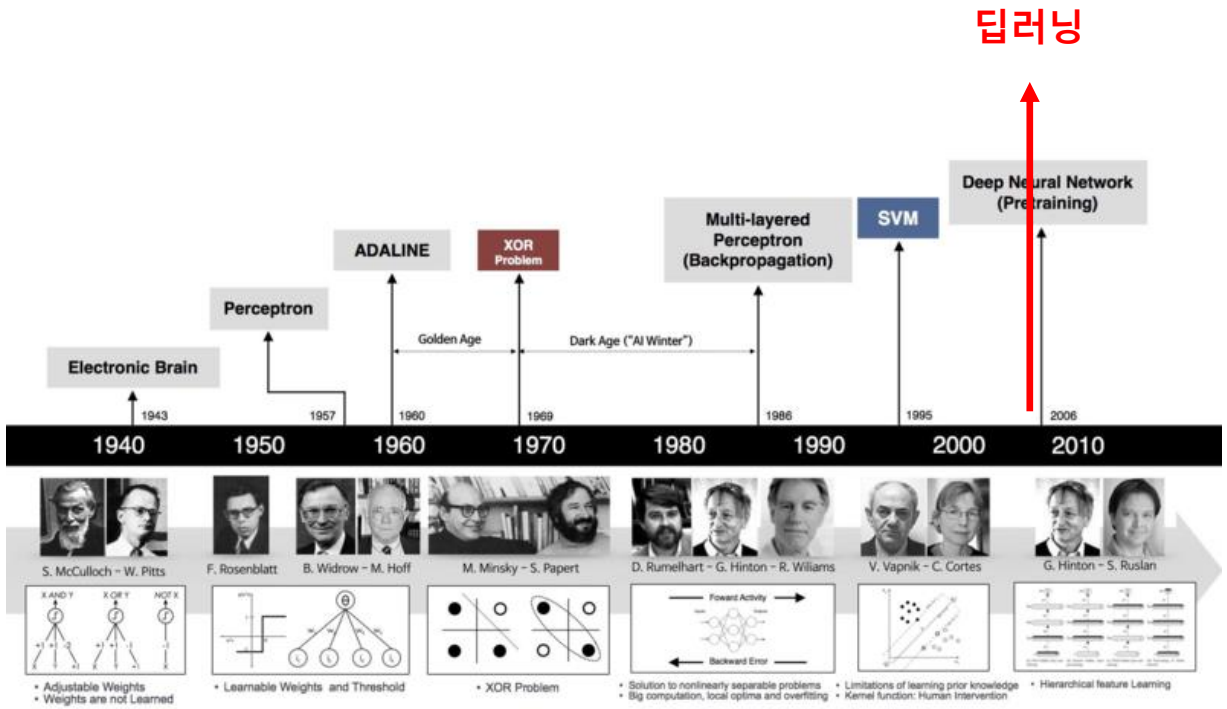
인공 지능 관련 영화

- **하이 젝시**
  - 새 폰을 구입하면서 만나게 된 인공지능 도우미 '젝시 '
- **프리 가이**
  - 게임의 캐릭터 들이 스스로 학습해 나가며 인공 지능 영역에 도달함.
- **바이센테니얼 맨**
  - 가사로봇 '앤드류'의 신경계에 문제가 생기면서 호기심과 지능을 가지게 됨.



인공지능의 역사

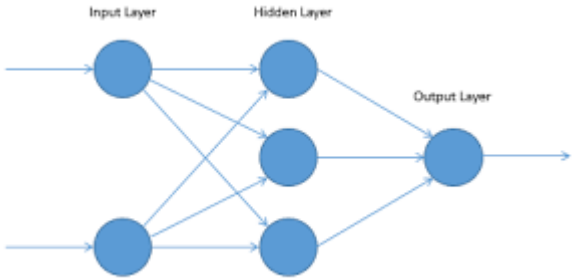
- 1950년대에 인공 지능의 개념과 신경망의 기본 개념인 퍼셉트론 개념이 성립됨.
- 2000년까지 신경망을 사용하지 않은 머신 러닝 사용됨.
- 2006 년부터 신경망을 이용한 딥 러닝 (Deep Learning)이 급속히 발전함.



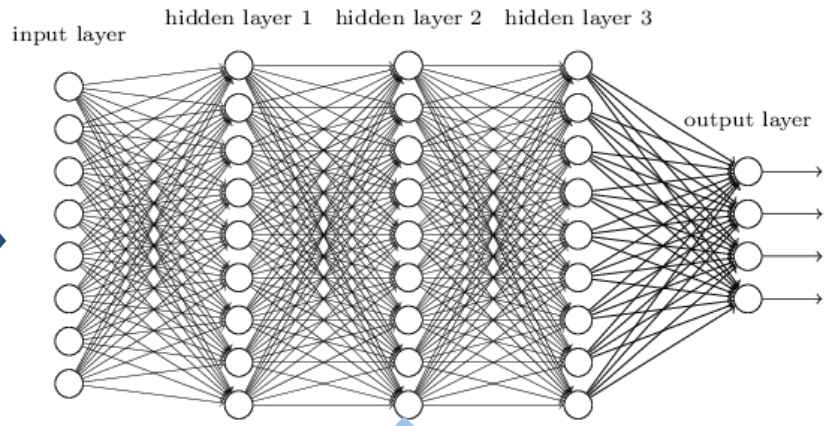
딥 러닝 (Deep Learning) 발전 이유

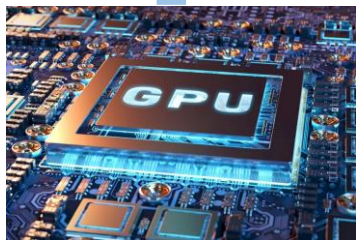
- **하드웨어의 발전**. 강력한 **GPU**는 딥러닝에서 복잡한 연산에 소요되는 시간을 크게 단축
- **빅 데이터**. 대량으로 쏟아져 나오는 데이터들을 학습에 활용 가능
- 신경망 이론의 단점 해결

small neural network

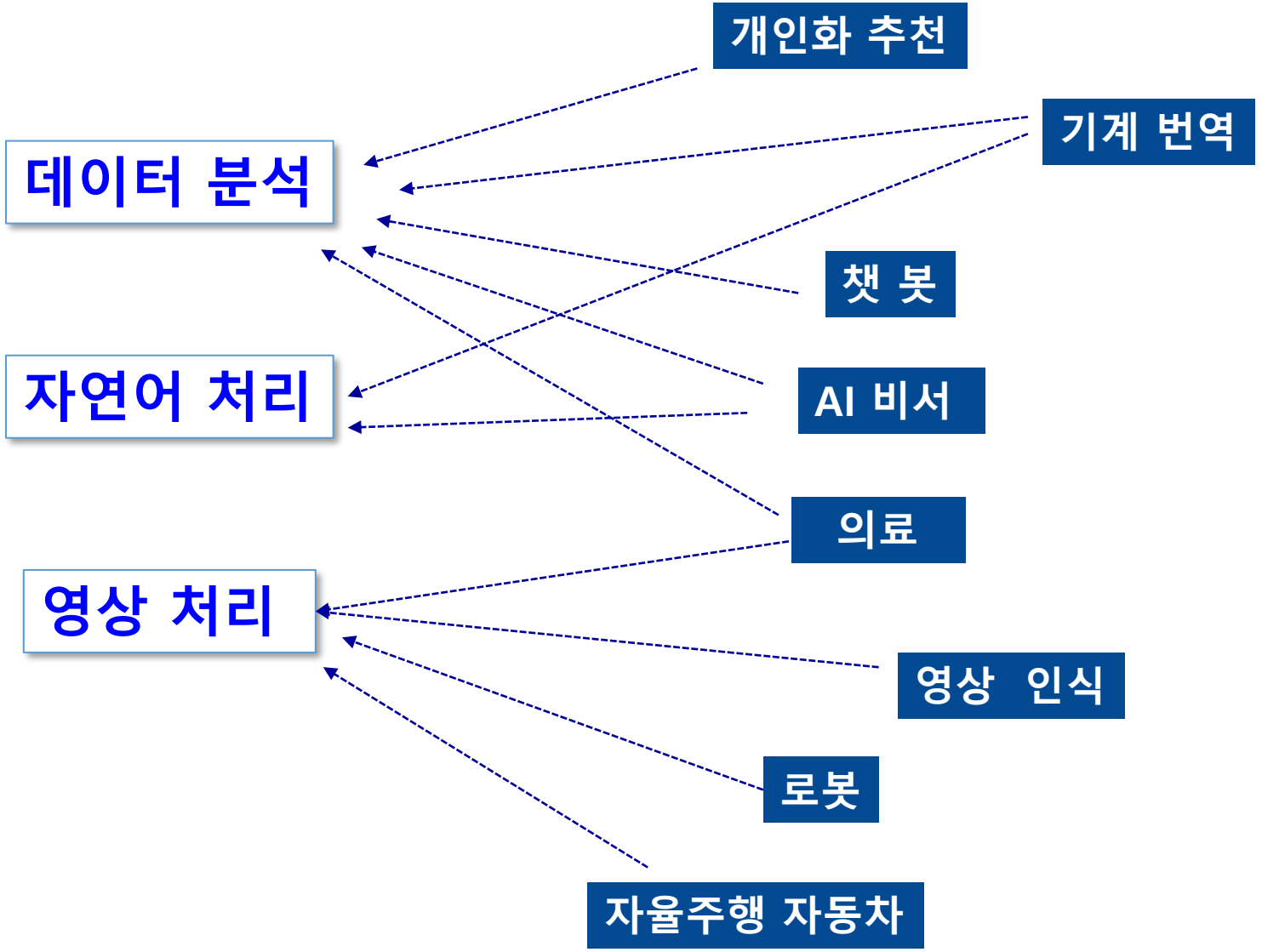


Deep Learning





인공 지능 기술 분야



인공 지능(머신 러닝) 분야 프로그래밍 환경

2010 년 이전

- 기본 수식 구현 부터 대부분의 기능을 c로 구현
- 프로그래밍 환경 설정이 쉬움.

Summary: the equations of backpropagation

$$\delta^L = \nabla_a C \odot \sigma'(z^L) \qquad \text{(BP1)}$$

$$\delta^l = ((w^{l+1})^T \delta^{l+}$$

$$\frac{\partial C}{\partial b^l_j} = \delta^l_j$$

$$\frac{\partial C}{\partial w^l_{jk}} = a^{l-1}_k \delta^l_j$$

```
main <- function() {
    header <- c("sepal_length", "sepal_width", "petal_length", "petal_width", "class")
    iris <- read.csv("https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data")
    names(iris) <- header
    sample <- sample.int(n = nrow(iris), size = floor(.75*nrow(iris)), replace = F)
    train <- iris[sample, ]
    test  <- iris[-sample, ]

    cl <- makePSOCKcluster(3)
    registerDoParallel(cl)
    lr <- multinom(class ~ ., data = train)
    lda_model <- lda(class ~ ., data = train)

    man_grid <- expand.grid(k = c(1:10))
    ctrl <- trainControl(method="cv", number = 5)
    knn <- train(class ~ ., data = train, method = "knn", trControl = ctrl, tuneGrid = man_grid)

    man_grid <- expand.grid(cost = c(1:10))
    ctrl <- trainControl(method="cv", number = 5)
    svm <- train(class ~ ., data = train, method = "svmLinear2", trControl = ctrl, tuneGrid = man_grid)

    stopCluster(cl)

    # Evaluate
    lr_accuracy <- sum(predict(lr, test) == test$class) / nrow(test)
    lda_accuracy <- sum(predict(lda_model, test)$class == test$class) / nrow(test)
    knn_accuracy <- sum(predict(knn, test) == test$class) / nrow(test)
    svm_accuracy <- sum(predict(svm, test) == test$class) / nrow(test)
    print(paste(lr_accuracy, ",", lda_accuracy, ",", knn_accuracy, ",", svm_accuracy))
}
```

2010 년 이후

- 파이썬 기반으로 다양한 라이브러리 사용 가능
- 대부분의 기능이 라이브러리에 함수로 구현되어 있음.
- 목적에 따라서 라이브러리를 선택 하여 구현
- 프로그래밍 환경 설정이 복잡함

python™

Keras

Caffe

TensorFlow

PyTorch

머신 러닝 프로그래밍

- 필요한 라이브러리 선택 후 이미 구현된 함수를 사용

라이브러리

머신 러닝

머신 러닝 프로그래밍 단계

1단계 : 짜여져 있는 구조(코드)를 목적에 맞게 사용할 수 있음

2단계: 기존 구조(코드)를 기반으로 최소한의 변경 하여 머신 러닝의 성능을 높일 수 있음
- 신경망 구조 변경
- 학습 시 파라미터 변경

3단계: 성능을 높일 수 있는 새로운 구조를 설계 할 수 있음

4단계: 성능을 높일 수 있는 새로운 함수를 개발 할 수 있음

머신 러닝 라이브러리

머신러닝

TensorFlow

Keras

PYTORCH
pytorch

scikit learn
Scikit learn

데이터 분석

pandas

seaborn

Matplotlib

자연어 처리

spaCy

Transformers
Hugging face

영상 처리

OpenCV

pillow

머신 러닝 라이브러리

기본 머신 러닝

Summary: the equations of backpropagation

$$\delta^L = \nabla_a C \odot \sigma'(z^L) \qquad \text{(BP1)}$$

$$\delta^l = ((w^{l+1})^T \delta^{l+}$$

$$\frac{\partial C}{\partial b^l_j} = \delta^l_j$$

$$\frac{\partial C}{\partial w^l_{jk}} = a^{l-1}_k \delta^l_j$$

```
main <- function() {
    header <- c("sepal_length", "sepal_width", "petal_length", "petal_width", "class")
    iris <- read.csv("https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data")
    names(iris) <- header
    sample <- sample.int(n = nrow(iris), size = floor(.75*nrow(iris)), replace = F)
    train <- iris[sample, ]
    test  <- iris[-sample, ]

    cl <- makePSOCKcluster(3)
    registerDoParallel(cl)
    lr <- multinom(class ~ ., data = train)
    lda_model <- lda(class ~ ., data = train)

    man_grid <- expand.grid(k = c(1:10))
    ctrl <- trainControl(method="cv", number = 5)
    knn <- train(class ~ ., data = train, method = "knn", trControl = ctrl, tuneGrid = man_grid)

    man_grid <- expand.grid(cost = c(1:10))
    ctrl <- trainControl(method="cv", number = 5)
    svm <- train(class ~ ., data = train, method = "svmLinear2", trControl = ctrl, tuneGrid = man_grid)

    stopCluster(cl)

    # Evaluate
    lr_accuracy <- sum(predict(lr, test) == test$class) / nrow(test)
    lda_accuracy <- sum(predict(lda_model, test)$class == test$class) / nrow(test)
    knn_accuracy <- sum(predict(knn, test) == test$class) / nrow(test)
    svm_accuracy <- sum(predict(svm, test) == test$class) / nrow(test)
    print(paste(lr_accuracy, ",", lda_accuracy, ",", knn_accuracy, ",", svm_accuracy))
}
```

2010 년 이후

- 파이썬 기반으로 다양한 라이브러리 사용 가능
- 대부분의 기능이 라이브러리에 함수로 구현되어 있음.
- 목적에 따라서 라이브러리를 선택 하여 구현
- 프로그래밍 환경 설정이 복잡함

python™

Keras

Caffe

TensorFlow

PyTorch

머신 러닝 개발을 위한 필요 역량

- 환경 설정
- 데이터 셋 선정 관리
- 많은 training/ test 경험

인공 지능 과목에서는 ….

- 데이터 분석
- 영상 처리
- Python, tensorflow, keras, pandas, numpy

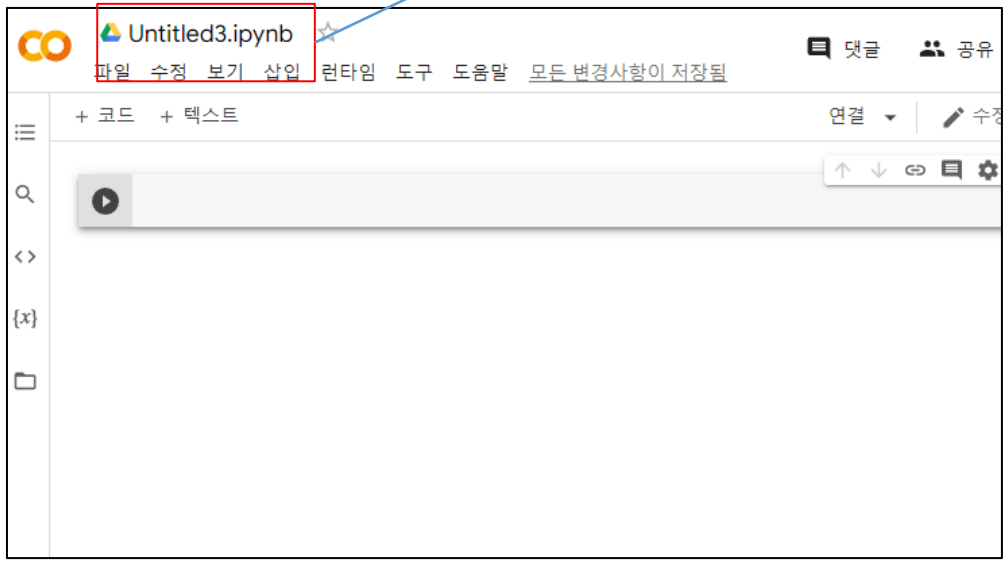
구글 Colab 실습

교재 350p

https://colab.research.google.com/ 구글 계정이 없으면 계정 생성

파일->새노트

수정: 코랩-테스트.ipynb



+코드 버튼 을 누른 후 다음 내용 입력

```
Import tensorflow as tf
print(tf.__version__)
```

재생 버튼 클릭 후 결과 확인

구글 Colab 실습 350p

이 러닝 시스템->.자료 공유 -> colab 실습 자료 download

- **파일-> 노트 업로드**
  - 파일 선택 -> My_First_Deeplearning.ipynb 선택

재생 버튼 클릭

파일 업로드 버튼 -> ThoracicSurgery.exel 선택

실행 결과 확인

수고 하셨습니다

jhmin@inhatc.ac.kr