

파이썬 프로그래밍 기초3

목차

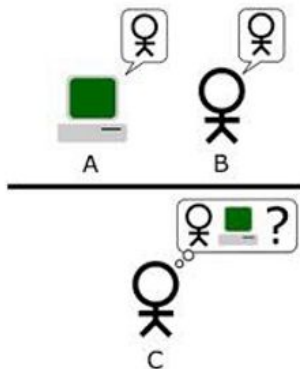
1. 머신러닝 개요
2. iris
3. mnist
4. 워드클라우드

머신러닝 개요#1

- 인공지능의 목표 (추론 능력)

튜링 테스트 (The Turing Test, 1950)

“기계가 생각할 수 있을까?”



Q: "벚꽃을 주제로 시를 지어줄래?"

A: "나한테 그런거 부탁하지마. 난 시를 잘 몰라"

Q: "341 더하기 701은 뭔지 계산 좀 해줄래?"

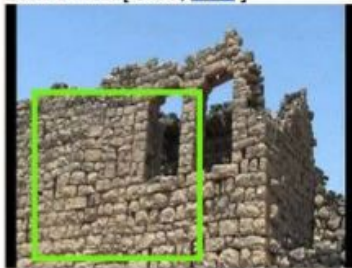
A: "음.. 그건1142이야"

머신러닝 개요#1

- 인공지능의 활용

영상인식 (Computer Vision)

stone wall [0.95, [web](#)]



dishwasher [0.91, [web](#)]



car show [0.99, [web](#)]



tractor [0.91, [web](#)]



tractor [0.91, [web](#)]



tractor [0.94, [web](#)]



머신러닝 개요#1

- 인공지능의 활용

감정 분석 (Sentiment Analysis)

"This movie should have NEVER been made. From the poorly done animation, to the beyond bad acting. I am not sure at what point the people behind this movie said "Ok, looks good! Lets do it!" I was in awe of how truly horrid this movie was."

긍정 혹은 부정 ? (Positive or Negative ?)

머신러닝 개요#1

- 인공지능의 활용

대화형 시스템 (Dialog System)

2016 ten breakthrough technology from MIT technology review

Chat bot

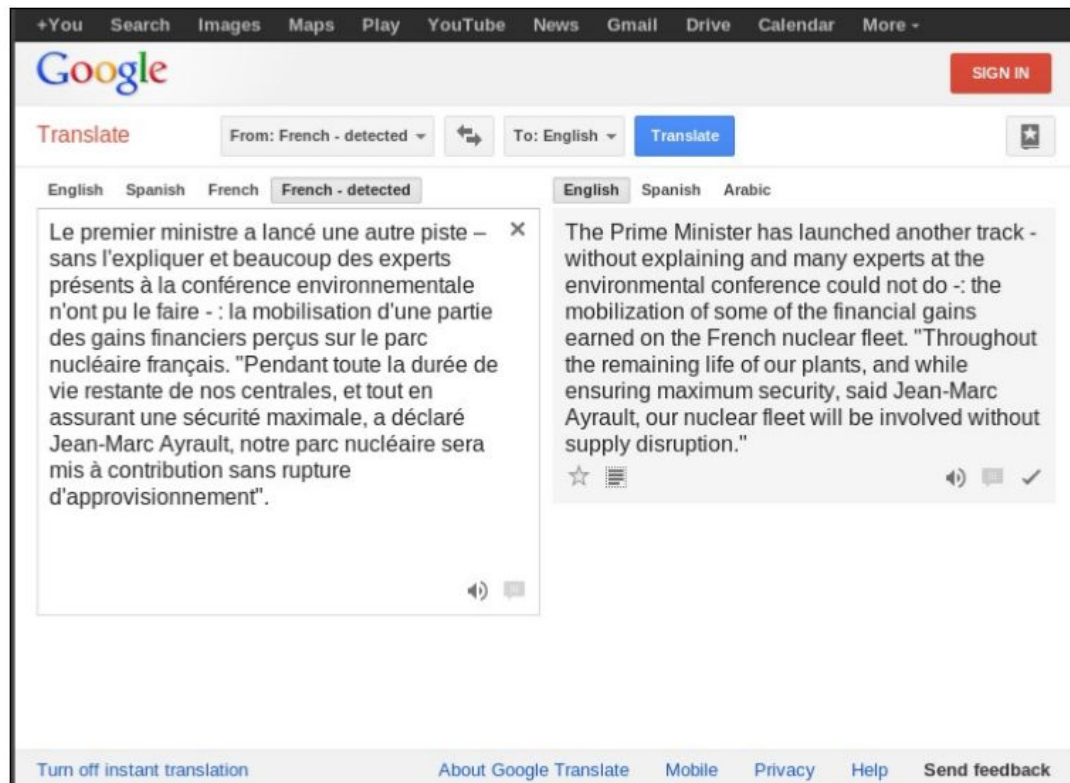
Tutor robot

...

머신러닝 개요#1

- 인공지능의 활용

기계 번역 (Machine Translations)



머신러닝 개요#1

- 인공지능의 활용

인간 대 기계 (Humans versus Machines)



1997: Deep Blue (chess)



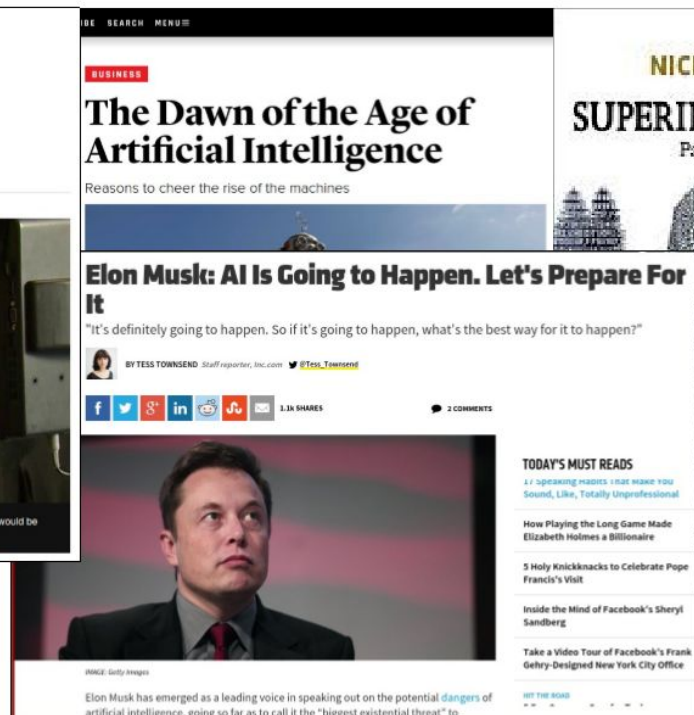
2011: IBM Watson (Jeopardy!)



2016: AlphaGo

머신러닝 개요#2

- 인공지능의 위험



머신러닝 개요#2

- 인공지능의 위험

자동화 위험이 높은 상위 20대 직업			자동화 위험이 낮은 하위 20대 직업		
분류코드	직업명	대체확률	분류코드	직업명	대체확률
5302	통신서비스 판매원	0.990	2440	영양사	0.004
5303	텔레마케터	0.990	2411	전문 의사	0.004
5304	인터넷 판매원	0.990	2591	장학관 · 연구관 및 교육 관련 전문가	0.004
8922	사진인화 및 현상기 조작원	0.990	1312	교육 관리자	0.007
2714	관세사	0.985	1331	보건의료관련 관리자	0.007
3125	무역 사무원	0.985	2521	중고등학교 교사	0.008
3142	전산 자료 입력원 및 사무 보조원	0.980	2545	학습지 및 방문 교사	0.009
3132	경리 사무원	0.970	2221	컴퓨터시스템 설계 및 분석가	0.011
5220	상품 대여원	0.970	2523	특수교육 교사	0.012
8212	표백 및 염색 관련 조작원	0.970	2420	약사 및 한약사	0.012
8222	신발제조기 조작원 및 조립원	0.970	1390	기타 전문서비스 관리자	0.014
8324	고무 및 플라스틱 제품 조립원	0.970	2542	컴퓨터 강사	0.014
8912	가구조립원	0.970	2489	기타 종교관련 종사자	0.017
8919	기타 목재 및 종이 관련 기계조작원	0.970	2481	성직자	0.017
9991	구두 미화원	0.970	2321	화학공학 기술자 및 연구원	0.017
3201	출납창구 사무원	0.965	2392	섬유공학 기술자 및 연구원	0.017
3126	운송 사무원	0.960	2393	가스에너지 기술자 및 연구원	0.017
8211	섬유제조 기계조작원	0.960	1311	연구 관리자	0.018
2712	회계사	0.957	2311	건축가 및 건축공학 기술자	0.018
2713	세무사	0.957	2341	환경공학 기술자 및 연구원	0.018

자료 : Frey & Osborne(2013), LG경제연구원

머신러닝 개요#3

- 인공지능에 필요한 두가지

Computation (time/memory)

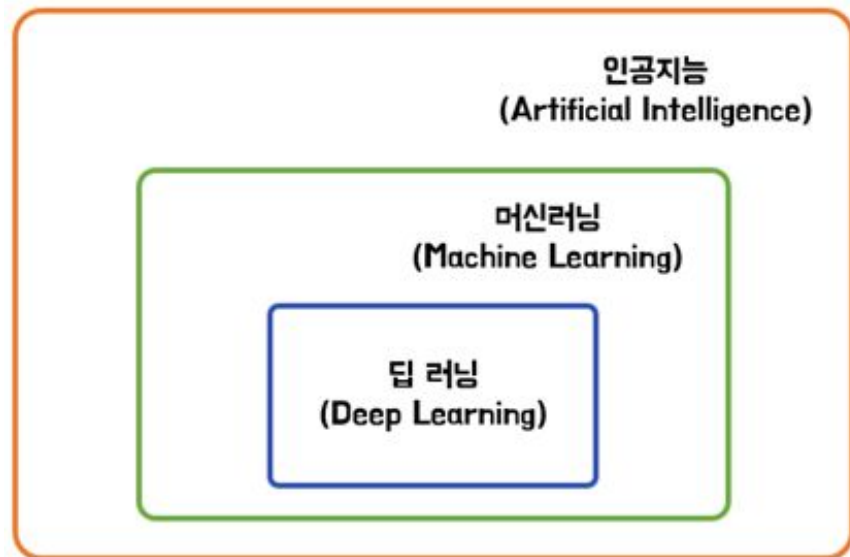


Information (data)



머신러닝 개요#3

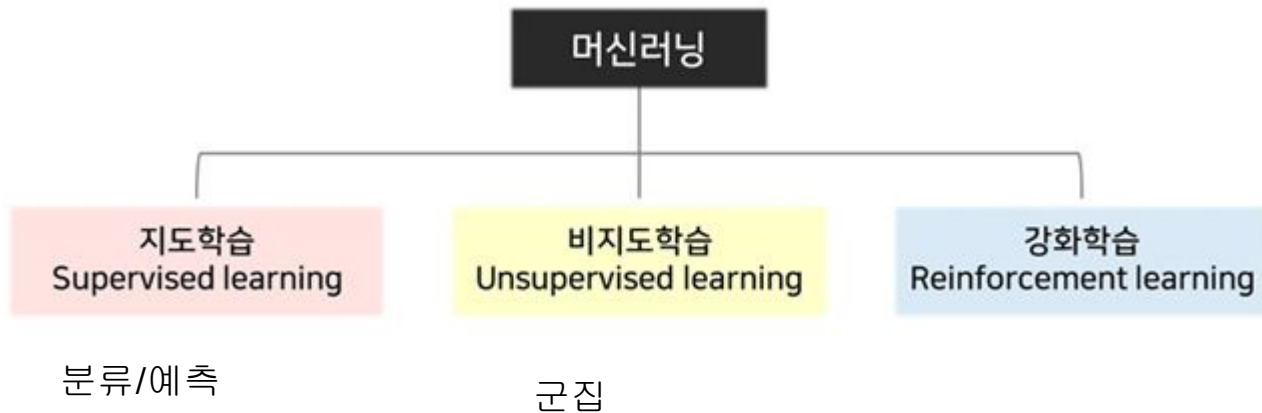
- 머신러닝과 딥러닝



<머신러닝과 딥러닝의 관계 도식>

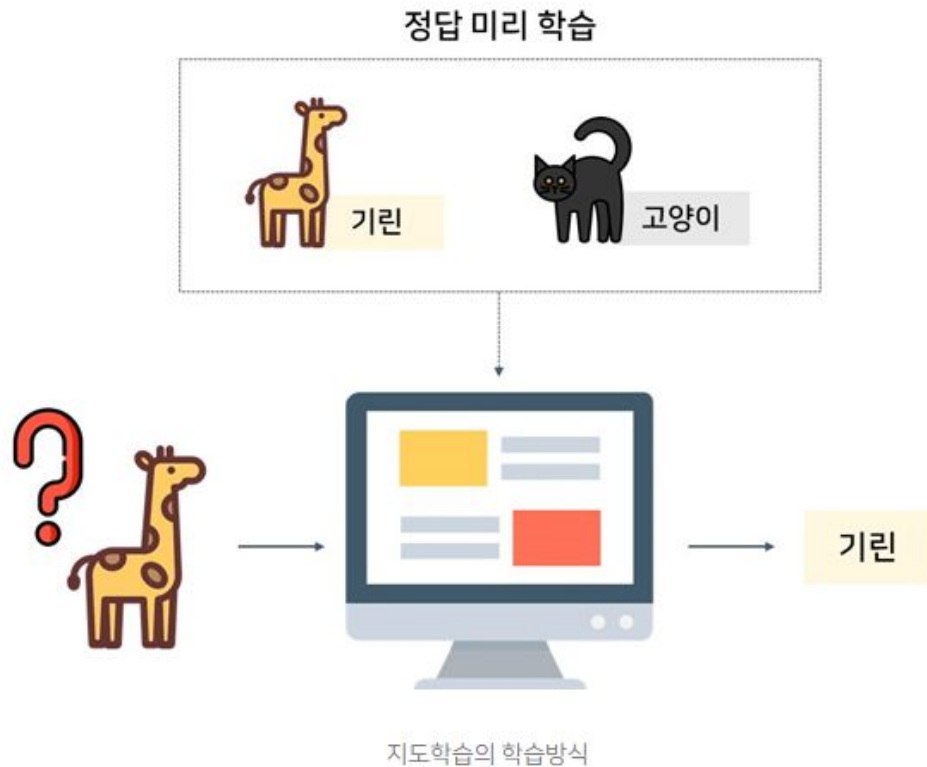
머신러닝 개요#3

- 머신러닝 종류
- 지도학습
- 비지도학습
- 강화학습



머신러닝 개요#3

- 머신러닝 종류
- 지도학습
- 비지도학습
- 강화학습



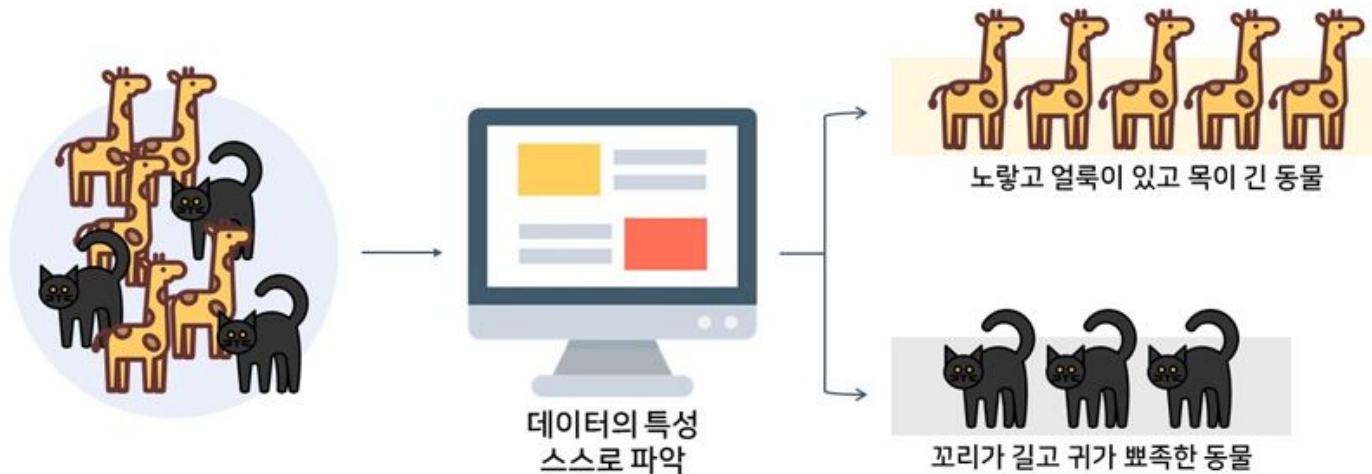
머신러닝 개요#3

- 머신러닝 종류
- **지도학습**
- 비지도학습
- 강화학습



머신러닝 개요#3

- 머신러닝 종류
- 지도학습
- **비지도학습**
- 강화학습



비지도학습의 학습방식

머신러닝 개요#3

- 머신러닝 종류
- 지도학습
- 비지도학습
- 강화학습

-정답이 있는 데이터는 개수가 제한적이기 때문에 비지도 학습이 앞으로 많아질것

-정답을 알려주지 않고 군집(비슷한 것끼리 모음)을 통해 미래예측



머신러닝 개요#3

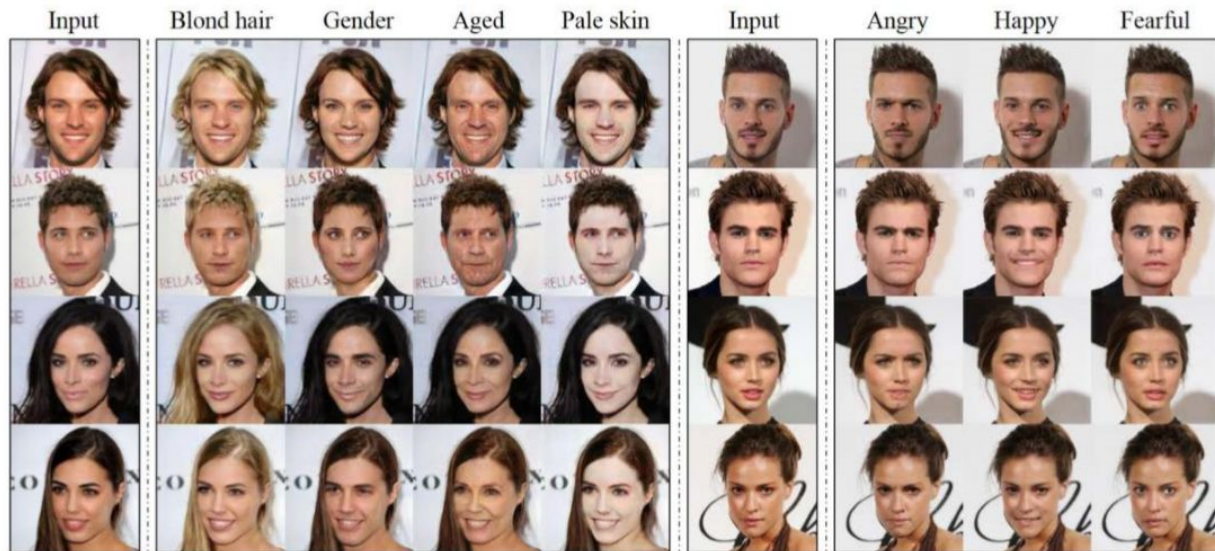
- 머신러닝 종류
- 지도학습
- **비지도학습**
- 강화학습

정답이 있는 데이터는 개수가 제한적이기 때문에
비지도 학습이 앞으로 많아질 것

<http://research.sualab.com/introduction/practice/2019/05/08/generative-adversarial-network.html>

https://news.chosun.com/site/data/html_dir/2018/07/04/2018070400201.html

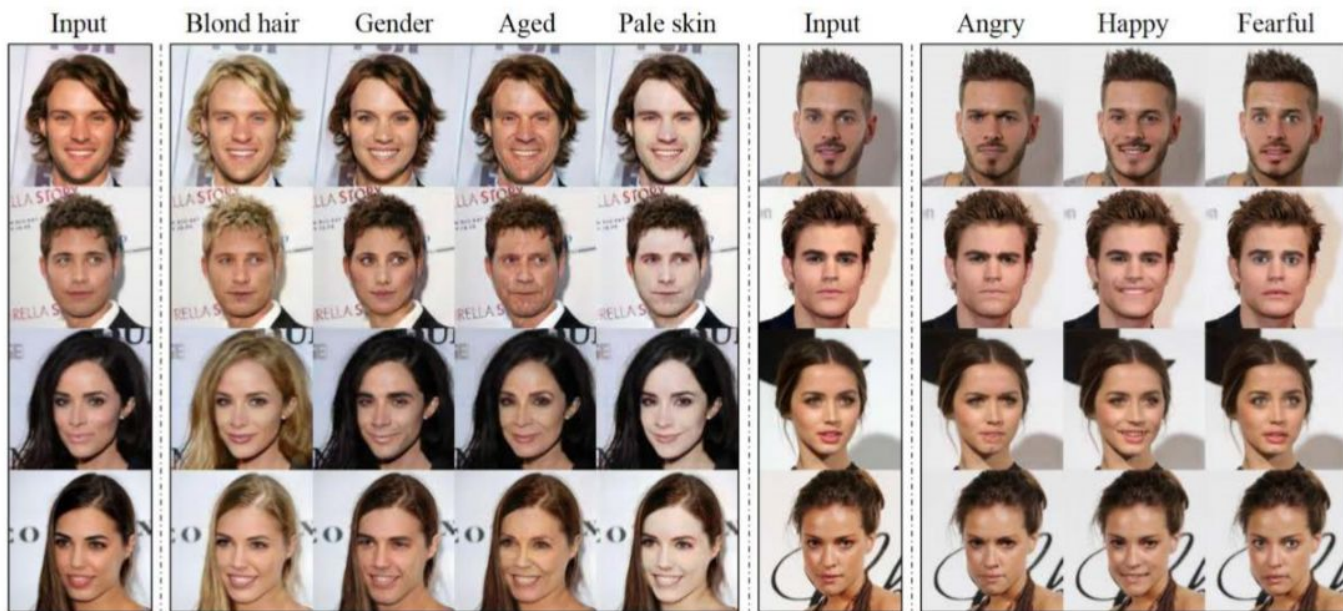
영상 조작



머신러닝 개요#3

- 머신러닝 종류
- 지도학습
- **비지도학습**
- 강화학습

영상 조작



머신러닝 개요#3

- 머신러닝 종류
- 지도학습
- 비지도학습
- 강화학습

지도 학습, 비지도 학습의 대표적인 알고리즘

지도 학습, 비지도 학습의 대표적인 알고리즘 목록입니다.
모든 알고리즘을 이해하기는 힘들지만 각각의 키워드를 이용하여 특징들을 살펴볼 수 있습니다.

지도학습(Supervised Learning),	Classification	kNN
		Naive Bayes
		Support Vector
		Machine Decision
	Regression	Linear Regression
		Locally Weighted Linear
		Ridge
		Lasso
비지도학습(Unsupervised Learning),		Clustering
		K Means
		Density Estimation
		Exception Maximization
		Pazen Window
		DBSCAN

머신러닝 개요#3

- 머신러닝 종류
- 지도학습
- 비지도학습
- **강화학습**

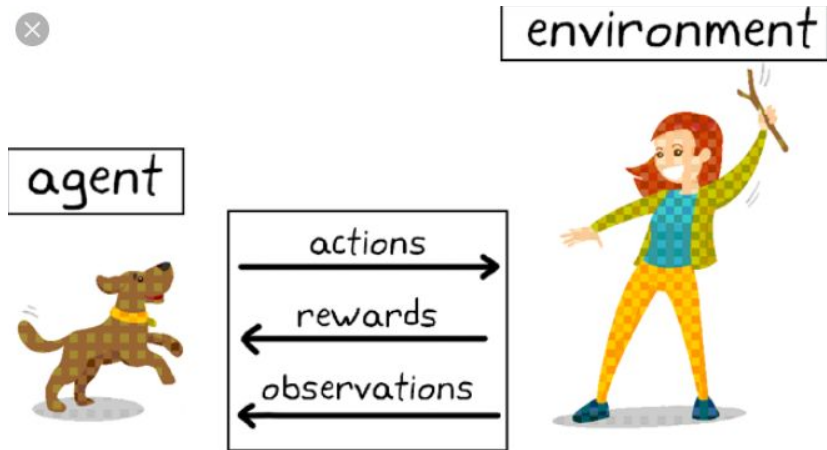
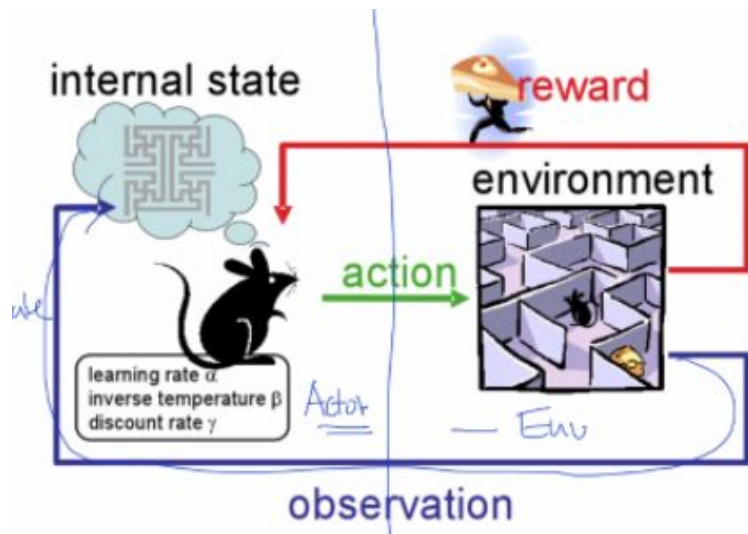
바둑의 경우의 수 **361**²⁰⁰

208168199381979984699478633344862770286522453884
530548425639456820927419612738015378525648451698
519643907259916015628128546089888314427129715319
317557736620397247064840935가지입니다

- 이런경우 지도학습, 비지도학습이 잘 되지 않음

머신러닝 개요#3

- 머신러닝 종류
- 지도학습
- 비지도학습
- 강화학습



머신러닝 개요#3

- 머신러닝 종류
- 지도학습
- 비지도학습
- **강화학습**

드론의 수평을 어떻게 맞출까
어떤 상태가 최적일까

<https://www.youtube.com/watch?v=8YGpo2OckB4>

<https://www.youtube.com/watch?v=V1eYniJ0Rnk&t=2s>

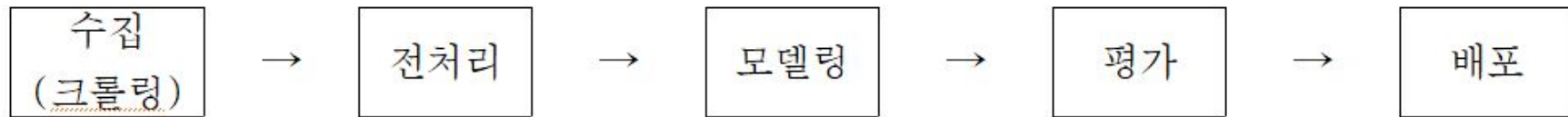
https://www.youtube.com/watch?v=iO_rVfOrKbE

<https://www.youtube.com/watch?v=TYvoZDPyzq0>

머신러닝 개요#4

- 머신러닝 프로세스

-



워드클라우드#1

- 토큰화 해보기 - 토큰화를 위해 우리가 준비해야 할 것 - **KoNLPy** 라이브러리를 이용할 것
- **KoNLPy**에서 지원하는 형태소 분석기들
 - Okt(Twitter) (open korean text)
 - Komoran
 - Kkma
 - Mecab
 - Hannanum

워드클라우드#1

- <http://colab.research.google.com>

실행 : Shift + Enter
Ctrl + Enter



워드클라우드#1

실행 : Shift + Enter
Ctrl + Enter

```
[1] !pip install konlpy
    from konlpy.tag import Okt
    okt = Okt()

    def tokenize_by_morphs(text):
        return okt.morphs(text)
    def tokenize_by_nouns(text):
        return okt.nouns(text)
    def tokenize_by_phrases(text):
        return okt.phrases(text)
    def tokenize_by_pos(text):
        return okt.pos(text)
```

워드클라우드#1

실행 : Shift + Enter
Ctrl + Enter

```
[2] text = "안녕하세요. 반갑습니다. 오늘은 자연어 처리에 대해서 배워 보겠습니다."  
print(tokenize_by_morphs(text)) #형태소  
print(tokenize_by_nouns(text)) #명사  
print(tokenize_by_phrases(text)) #구  
print(tokenize_by_pos(text)) #품사
```

☞ ['안녕하세요', '.', '반갑습니다', '.', '오늘', '은', '자연어', '처리', '에', '대
['오늘', '자연어', '처리', '대해']
['오늘', '자연어', '자연어 처리', '대해', '처리']
[('안녕하세요', 'Adjective'), ('.', 'Punctuation'), ('반갑습니다', 'Adjective')]

워드클라우드#1

실행 : Shift + Enter
Ctrl + Enter

```
[3] import re
def clean_text(str):
    pattern = re.compile('[^가-힣ㄱ-ㅎa-zA-Z0-9₩,₩.₩ ]')
    str = re.sub(pattern, '', str)
    return str
```

```
[4] # findall연습 -> 리스트로 만들어줌
p = re.compile('[a-z]+')
result = p.findall("life is too short")
print(result)
```

☞ ['life', 'is', 'too', 'short']

```
[5] # sub연습
re.sub('apple|orange', 'fruit', 'apple box orange tree')
```

☞ 'fruit box fruit tree'

워드클라우드#1

실행 : Shift + Enter
Ctrl + Enter

```
[6] article_example = ""
```

아시아경제

市 "서울 집값 급등, 공급부족 문제 아냐...부동산 국민공유제 실행계획 곧 마련"

[아시아경제 김유리 기자] 서울시가 서울 집값 급등의 원인으로 지목된 '서울 시내 공
진희선 서울시 행정2부시장은 6일 서울시청에서 서울 주택 수급 등 주요 현안에 대한
서울시는 2008~2013년, 2014~2019년, 향후 6년(2020~2025년) 서울 주택 공급 현황을
""

```
print(clean_text(article_example))
```



아시아경제 서울 집값 급등, 공급부족 문제 아냐부동산 국민공유제 실행계획 곧 마련(

```
[7] clean_str = clean_text(article_example)
sentence_pattern = re.compile('[^ㄱ-ㅎ.] *다.')
print(sentence_pattern.findall(clean_str))
```

워드클라우드#1

실행 : Shift + Enter
Ctrl + Enter

```
[7] clean_str = clean_text(article_example)
    sentence_pattern = re.compile('[^ㄱ-ㅎ.] *다.')
    print(sentence_pattern.findall(clean_str))
```

☞ ['아시아경제 서울 집값 급등, 공급부족 문제 아나부동산 국민공유제 실행계획



```
[8] import collections
    counter_dict = collections.Counter()

    for sentence in sentence_pattern.findall(clean_str):
        counter_dict.update(collections.Counter(oka.nouns(sentence)))
    print(counter_dict)
```

☞ Counter({'공급': 13, '서울': 7, '가구': 7, '서울시': 4, '주택': 4, '부족'

워드클라우드#1

실행 : Shift + Enter
Ctrl + Enter

예제7

```
!apt-get install fonts-nanum*  
!apt-get install fonts-woowa-hanna  
  
from wordcloud import WordCloud  
import matplotlib.pyplot as plt  
  
wordcloud = WordCloud(font_path =  
"/usr/share/fonts/truetype/nanum/NanumGothic.ttf",background_color='white',  
width=800, height=800).generate_from_frequencies(counter_dict)  
plt.rcParams["figure.figsize"] = (10,10)  
plt.imshow(wordcloud)  
plt.axis('off')  
plt.show()
```


워드클라우드#1

- <https://regexr.com>

The screenshot shows the regexr.com website interface. The browser address bar displays `https://regexr.com/`. The page has a dark theme. On the left, there is a sidebar menu with options: Menu, Pattern Settings, My Patterns, Cheatsheet, RegEx Reference, Community Patterns, and Help. The main area is divided into two sections: 'Expression' and 'Text'. The 'Expression' section shows the pattern `/[\^\.]디./g`. The 'Text' section shows a block of Korean text with 8 matches highlighted. The matches are: '디', '디', '디', '디', '디', '디', '디', and '디'. Below the text, there is a 'Tools' section with buttons for 'Replace', 'List', 'Details', and 'Explain'. The 'Details' button is selected, showing a list of regex symbols and their meanings:

- `[^]` Negated set. Match any character that is not in the set.
- `\.` Escaped character. Matches a "." character (char code 46).
- `*` Quantifier. Match 0 or more of the preceding token.
- `디` Character. Matches a "디" character (char code 45796).
- `.` Dot. Matches any character except line breaks.

워드클라우드#1

- <https://regexr.com>