1. 개요

자연어 처리 변화 추세

- **변화 추세:** 자연어 처리 분야에서의 최근 시스템 성능 향상은 기계학습 방식의 진전에 의해 이루어졌다고 할 수 있음
- 도서 자료에서도 기계학습 기술들을 많이 다루고 있음
- **개발 언어:** 연구와 교육에서는 파이썬 언어를 주로 사용하고 있음
- **개발 환경:** Linux, 윈도우, MacOS 등에서 사용할 수 있음

강의 교재

- **교재 1:** *딥 러닝을 이용한 자연어 처리 입문*, 유원준, 온라인 문서
  - https://wikidocs.net/book/2155
  - 이 책은 Keras(Tensorflow)를 이용한 language processing 프로그램을 예제로 제시하고 있음
  - 이론적인 부분은 부족하지만 실질적인 알고리즘과 프로그램을 포함하고 있음

- **교재 2:** *Deep Learning with Python*, Francois Chollet, 2018.

  (*케라스 창시자에게 배우는 딥러닝*, 박해선 옮김, 길벗, 2018)
  - Keras를 이용한 deep learning 프로그래밍 기법을 다루고 있음
  - 영문 pdf version을 구할 수 있음

강의 계획

- 이론적인 내용보다 실질적인 알고리즘과 프로그래밍 환경 구축을 강조할 예정
- 기계학습 부분은 기초 개념부터 다룰 것임
- **자연어처리 기본 이론:** 텍스트 전처리, 언어 모델, 카운트 기반 단어 표현, 토픽 모델링 등
- **머신 러닝 개념:** 선형 회귀, 소프트맥스 회귀, 인공 신경망, gradient descent, 역전파, 기울기 소실, 케라스 사용 방법
- **자연어처리에서의 머신 러닝**: 순환 신경망, 워드 임베딩, 텍스트 분류
- **심화 과정:** RNN을 이용한 기계 번역, Attention, Transformer, BERT 등

자연어 처리 주요 도서

- **주요 도서:**

  - *자연어처리 바이블*, 임희석, 휴먼싸이언스, 2019.

  - *자연어 처리 딥러닝 캠프*, 김기현, 한빛미디어, 2019.
    - PyTorch(Facebook에서 개발한 deep learning 개발 환경)를 기반으로 하고 있음

  - *뉴럴 모델을 이용한 자연어 처리*, 이상근, 메이킹북스, 2020.
    - 신경망 기반 자연어 처리 기술을 설명

  - *Speech and Language Processing*, 3rd ed. 진행중, Jurafsky & Martin
    - https://web.stanford.edu/~jurafsky/slp3/
    - pdf 파일은 강의 폴더에 있음

자연어 처리 참고 자료

- 기계학습과 자연어처리에 대한 강의/참고 자료는 Stanford 대학에서 찾을 수 있음
- CS224n: Natural Language Processing with Deep Learning (http://cs224n.Stanford.edu/)
  - 이 사이트에 강의 slides, 문서, 동영상 등이 포함되어 있음

프로그래밍 환경 구성

- 이 강의에서는 **Windows** (OS) 환경에서 **Python** 언어와 **Anaconda** (Jupyter notebook) 프로그래밍 환경, **Konlpy** 한글 처리 도구, 그리고 **Tensorflow** (**Keras** 포함) 환경을 사용
- 다음에 설명되는 프로그래밍 환경을 구축하는 것이 **매우 중요함**

프로그래밍 환경 구축

- 교재의 1장 참고
- 교재에서는 윈도우 기반으로 사용하는 환경이며 파이썬 언어를 이용
- **Anaconda 설치**
  - 파이썬 언어와 주요 패키지들을 포함하고 있음
  - 포함 패키지: numpy, pandas, Jupyter notebook, ipython, scikit-learn, matplotlib, nltk 등
  - 링크: https://www.anaconda.com/distribution/
  - 파이썬 3.x 64 비트 버전으로 설치

- Anaconda prompt, Jupyter notebook 등이 설치됨

nltk 패키지

- nltk(Natural language toolkit)
  - 미국에서 개발된 자연어 처리 기능을 가진 toolkit
  - Anaconda를 설치하면 같이 설치됨: 별도로 설치할 필요가 없음
- nltk 확인
  - Anaconda prompt에서 ipython을 실행

```
> ipython
...
In [1]: import nltk
In [2]: nltk.__version__          # underscore(_)가 2개씩임
Out [2]: '3.5'
```

- nltk 기능을 제대로 사용하려면 여러 데이터를 추가적으로 설치

```
In [3]: nltk.download()
```

koNLPy 패키지

- **koNLPy(Korean natural language processing in Python)**
  - 한국어 형태소 분석기 패키지
  - Anaconda 설치 이후 추가로 설치
  - 소개 자료: https://konlpy.org/en/latest/

- Anaconda prompt에서 설치

  > pip install konlpy

  > ipython

  …

  In [1]: import konlpy

  In [2]: konlpy.__version__

  Out [2]: '0.5.2'

koNLPy 에러가 발생할 때

- koNLPy는 Java로 되어있기 때문에 JDK 1.7 이상 버전과 JPype가 설치되어 있어야 함: 교재 1.3절 참조
- 교재를 참고하여 다음을 수행

    1) JDK 설치

    2) JDK 환경 변수에서 JAVA_HOME을 설정

    3) JPype 설치

Deep learning 패키지 설치: Tensorflow와 keras

- Anaconda prompt 도구를 이용하여 설치

  > pip install tensorflow

- 설치 확인: Anaconda prompt에서 ipython을 실행

  > ipython

  ...

  In [1]: **import** tensorflow **as** tf

  In [2]: tf.__version__

  Out[2]: '2.3.0'

- keras도 마찬가지로 설치

  > pip install keras

설치된 패키지 확인: pandas, numpy, matplotlib

- 데이터 분석을 위한 패키지들인데, Anaconda를 설치하면 자동으로 설치됨
- **판다스:** 데이터 처리 패키지
- **넘파이:** 수치 데이터 처리 패키지
- **matplotlib:** 데이터 시각화 패키지

판다스

- 데이터 처리를 위한 패키지
- 지원되는 데이터 구조: series (1차원 배열), data frame(2차원 리스트), panel 등
- Series: 1차원 배열값에 index를 부여할 수 있음

```
import pandas as pd
sr = pd.Series([17000, 18000, 1000, 5000],
       index=['피자', '치킨', '콜라', '맥주'])
print(sr)
```

```
피자     17000
치킨     18000
콜라      1000
맥주      5000
dtype: int64
```

print(sr.values)

```
[17000 18000 1000 5000]
```

print(sr.index)

```
Index(['피자', '치킨', '콜라', '맥주 '],  dtype='object')
```

판다스: Data frame

- 2차원 데이터 처리를 위한 패키지로 행방향(index)과 열방향(column) 인덱스가 있음

```
values = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
index = ['one', 'two', 'three']
columns = ['A', 'B', 'C']

df = pd.DataFrame(values, index=index, columns=columns)
print(df)
```

```
       A  B  C
one    1  2  3
two    4  5  6
three  7  8  9
```

판다스 외부 데이터 읽기

- 외부 데이터 읽기: 판다스는 csv, 텍스트, Excel, SQL, HTML, JSON 등 다양한 포맷의 데이터를 읽을 수 있음

```
df = pd.read_csv('example.csv')
print(df)
```

	student id	name	score
0	1000	Steve	90.72
1	1001	James	78.09
2	1002	Doyeon	98.43
3	1003	Jane	64.19
4	1004	Pilwoong	81.30

판다스 프로파일링

- 데이터 상태를 분석하는 패키지를 다음과 같이 별도로 설치

  > pip install –U pandas-profiling

- **실습:** 교재 1.5절에 설명되어 있는 “spam.csv” 파일을 download 하여 교재에 설명된 순서대로 분석을 수행

넘파이(Numpy)

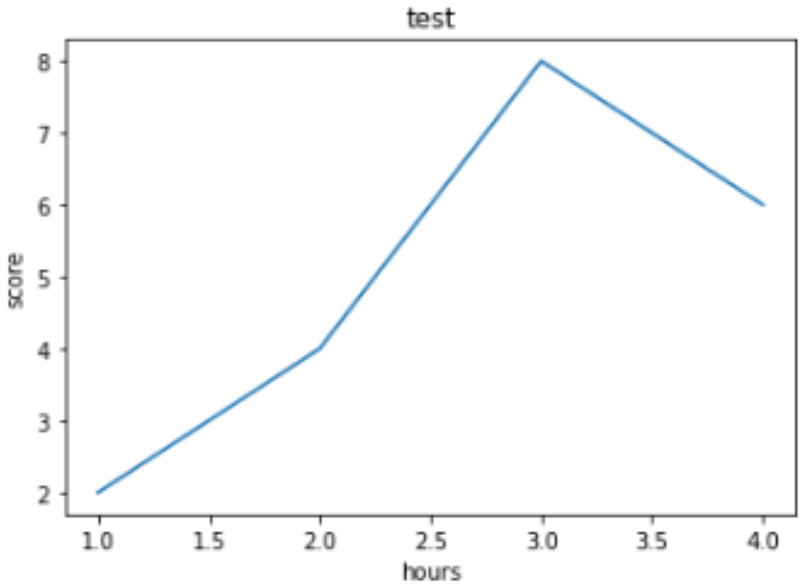
- 수치 데이터 처리를 위한 패키지로서 n차원 배열인 ndarray를 통해 벡터와 행렬을 계산
- 사용 사례

```
x = np.array([1,2,3])
y = np.array([4,5,6])

b = x + y
b = b * x
b = b / x
```

matplotlib

- 데이터를 차트나 플롯으로 시각화함
- 사용 사례

```
import matplotlib.pyplot as plt

plt.title('test')
plt.plot([1,2,3,4],[2,4,8,6])
plt.xlabel('hours')
plt.ylabel('score')
plt.show()
```



nltk 기능 확인

- nltk를 이용하여 영어 문장에서 단어의 품사를 확인함(tagging)
- Jupyter notebook을 실행시킨 다음, 새로운 python 파일을 생성하고 다음 문장들을 입력

```
import nltk

sentence = "Seoul is the capital of Korea."
wordsInSent = nltk.word_tokenize(sentence)
print(wordsInSent)
partOfSent = nltk.pos_tag(wordsInSent)
print(partOfSent)
```

- 이 프로그램을 수행시키면 다음의 결과를 얻음

```
['Seoul', 'is', 'the', 'capital', 'of', 'Korea', '.']
[('Seoul', 'NNP'), ('is', 'VBZ'), ('the', 'DT'), ('capital', 'NN'), ('of',
'IN'), ('Korea', 'NNP'), ('.', '.')]
```

koNLPy 기능 확인

- konlpy를 이용하여 한글 문장에서 단어의 품사를 확인함
- 새로운 python 파일을 생성하고 다음 문장들을 입력

```
from konlpy.tag import Kkma

kkma = Kkma()

sentence = "서울은 한국의 수도라고 하던데."
print(kkma.nouns(sentence))
print(kkma.pos(sentence))
```

- 이 프로그램을 수행시키면 다음의 결과를 얻음

['서울', '한국', '수도']

[('서울', 'NNG'), ('은', 'JX'), ('한국', 'NNG'), ('의', 'JKG'), ('수도', 'NNG'), ('라고', 'JX'), ('하', 'VV'), ('던데', 'ECD'), ('.', 'SF')]