

영어음성학 기말고사 필기

19/10/29

1. 데이터 - 소리/이미지/텍스트/숫자(주식, 온도, 날씨 등)의 4가지로 나뉘어짐.

- 여기서 숫자화 되지 않은 데이터들(소리, 이미지, 텍스트 데이터)이 어떻게 숫자화가 되는가?

- 행렬

- (영상이나 사진에서) 숫자들이 직사각형의 형태로 행(가로)과 열(세로) 속에 쳐 놓은 것이다.
- 예를들어 흑백 사진에서는 검정색이 0, 흰색이 10으로 포함된다면 회색은 1~9의 숫자값으로 직사각형의 형태로 배열되어 있는 것이다. - 2차원
- 컬러 사진일 경우에는 직사각형의 행렬이 여러 장이 있는 것이다. (예를 들어 RGB라면 한 직사각형 판은 얼마나 빨간 정도를 가지고 있는가, 다른 판은 초록, 다른 판은 파란 정도를 알려주는 것이다) - 3차원
- 여기에 시간까지 추가가 된다면 4차원이 되는 것이다.

- 벡터

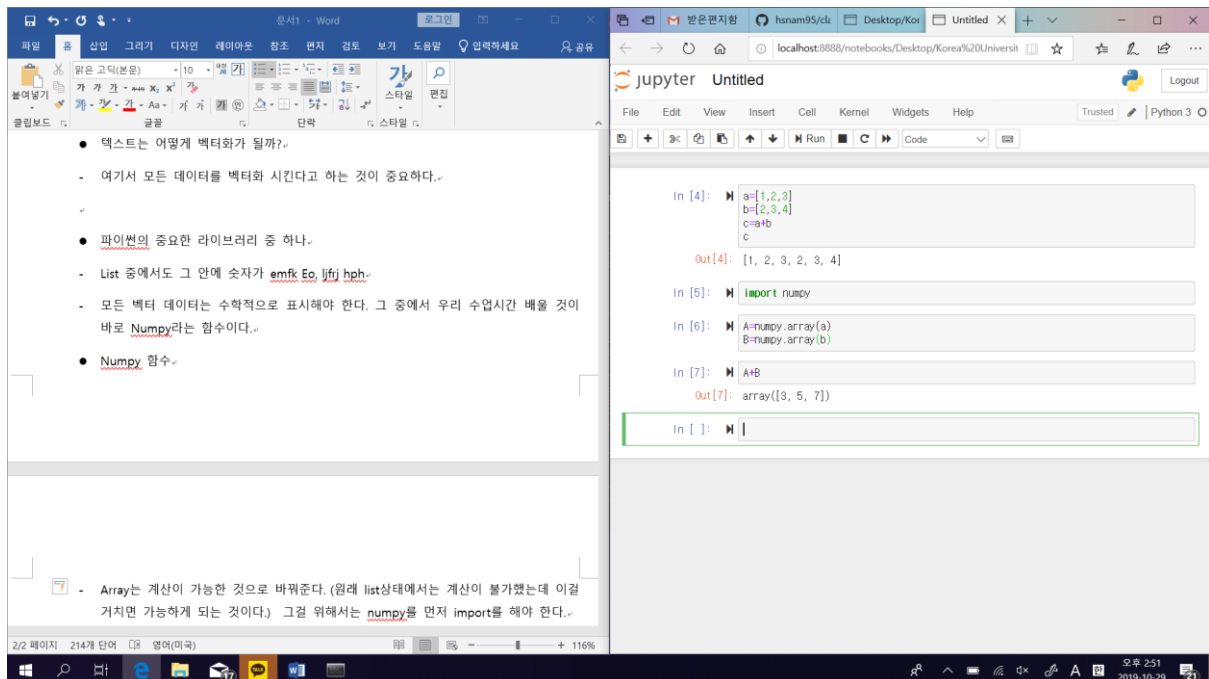
- 모든 데이터는 벡터의 형태로 되어야 한다. 행렬은 벡터는 아니다. 왜냐하면 직사각형 형태이기 때문이다.
- 이미지는 행렬이고, 이 것을 길게 쭉 늘어 놓은 것이 벡터라고 한다. (한 줄로)
- 모든 데이터는 벡터화 했을 때 다루기가 쉽다.
- 소리는 어떻게 영상과 비슷할까? (어떻게 벡터화가 되는 것일까?)
- 텍스트는 어떻게 벡터화가 될까?
- 여기서 모든 데이터를 벡터화 시킨다고 하는 것이 중요하다.

- 파이썬의 중요한 라이브러리 중 하나

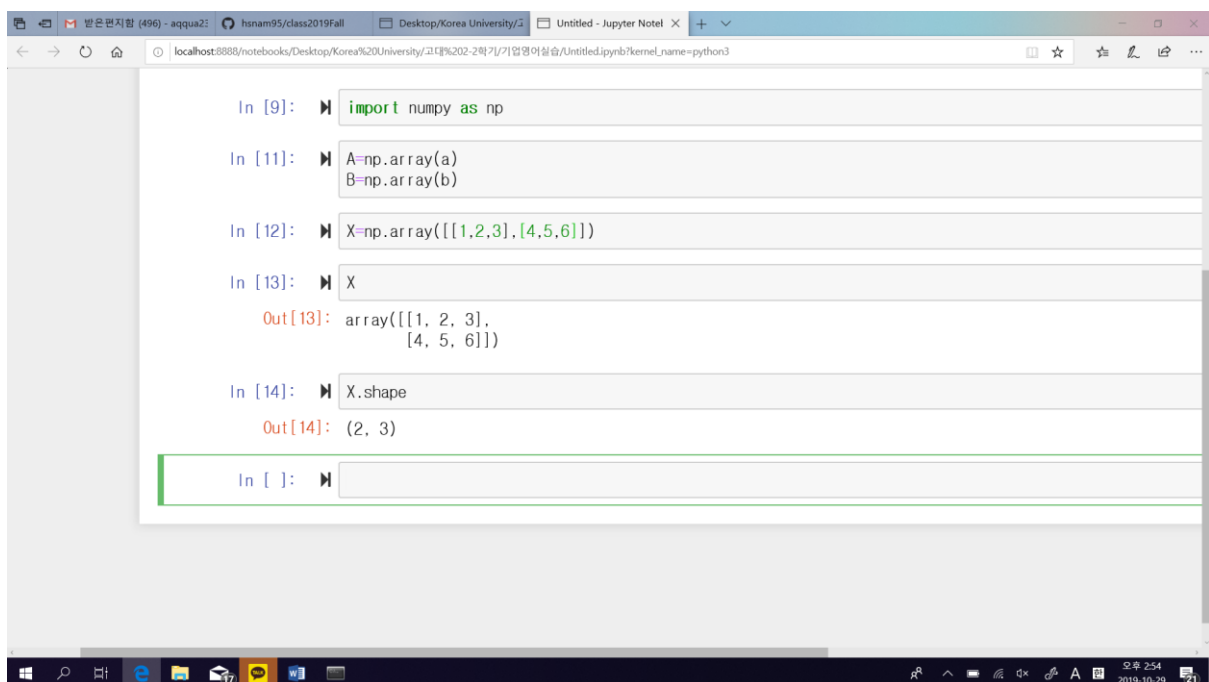
- List 중에서도 그 안에 숫자가 emfk Eo, ljfrj hph
- 모든 벡터 데이터는 수학적으로 표시해야 한다. 그 중에서 우리 수업시간 배울 것이 바로 Numpy라는 함수이다.

- Numpy 함수

- Array는 계산이 가능한 것으로 바뀌준다. (원래 list상태에서는 계산이 불가능했는데 이걸 거치면 가능하게 되는 것이다.) 그걸 위해서는 numpy를 먼저 import를 해야 한다.



- X.shpae은 차원을 말해주는 것이다.



- 교수님 github 계정 들어가서 미리 예습 해보자.

19/10/31

1. Numpy / Import / From

- Numpy가 가장 큰 집합을 의미하는 것이다.
- 줄여서 쓰고 싶을 때는 `import numpy as np`라고 해서 밑에서는 계속 `np`라고 쓸 수 있다.
- `""` 온점은 포함관계로 되어 있는 것을 의미한다. (집합 안에 집합을 지칭)
- Numpy는 list와 비슷한데, 쓰는 이유는 수학적인 계산이 용이하기 때문이다. (따라서 이제는 list 대신에 numpy를 더 많이 써야 한다.)
- 온점이 뒤에 적혀있다는 것은 데이터 타 | 입은



- Numby

`Np.linspace (0,10,6)` 0부터 10까지 6개로 나눠본다.

그래서 그 후의 값이 나오는 것이다.

Linear space는 공차(d-difference space)들이 같기 때문에 본다.

- Type 이야기를 했고.
- Type을 바꾸고싶으면 X.

2. Plotting

3. Hist = histogram. 바구니를 말하는 것이다. 히스토그램.

4. 여기까지가 중요했다. 그다음은 3번 Numpy I/O가 있는데 그게 또 중요하다. 알아서 니가 해 라고 할 때 -1이 된다. In [38]번에서! 괄호 맨 처음에 있는 숫자가 -1이든 4이든 값은 4로 같게 나온다.

5. 3번 Numpy I/O 바로 위에 있는 In [16]인가 그거는 할 필요 없다.

6. `savez`라는 함수를 쓰면 실제로 file로 저장이 되는 것이다.

7. In [42]번 ! 내 파이썬 사이트로 k서

- 이번줄 말에 복습해서 메모해서 정리하는 거 한번 해봐야 한다. 그리고 혼자서 정리 및설명을 할 수 있을 정도로 해라.
- 그리고 교수님 gitup 나중에 가서 수업자료 한번다운받아서 열어보기. 데이터를 서로 서로 전달을 할 때 데이터 숫자들을 파일로 전환해서 주는 거도.

8. 4번 Inspecting의 소제목이 등장!

9. type으로는 `numby.np.array`를 의미한다. 제일 위예가 ~이다.

10. Arrange 하나만 나오면 index가 나온다. 2개가 나오면 ~ 3개가 나오면 ~ (9부터 5까지 9,8,7,6으로 4개가 도출되는 것이다)

- In[9], [65], [13]은 다음에 알려주겠다고 했다.
- 5.2 Comparison은 수업 진행한다.

11. In [67], [68], [69]

- In [67] – np.arange랑 reshape.
- A=b 같으냐? 라고 묻는 것은 등호 2개 ==를 쓴다고 함. 질문형인 것임.
- 5.3번 Aggregate 는 매우 매우 중요한 것이다. 그래서 엄청 잘 봐야 한다.
- In [49]는 a / 70번에 a는 이미 처리 완료.

12. 함수면 반드시 괄호를 쳐야 한다. 자기 자신을 sum 하라고 하니까 일단 그렇게 한다. 뭘 sum을 해야 하는가 하면은 ... 이 것 쓰는 방법을 정확하게 이해해야 한다.

13. A가 3곱하기 3인데 이게 첫번째 차원에 해당되는게 가로행을 의미하는 것이고 두번째 차원은 세로열을 의미하는 것이다. 그래서 그 관점에서 실행을 하라고 명령을 하는 것이다.

- Broadcasting

- We-shape가 broadcasting의 일종이다.
- [52번에서] A의 차원이 5 이다
- 여기까지가 numpy고 다음시간에는 sound를 본격적으로 다룰 것이다. 엄청나게 import를 많이 한다. 많이 알아야 한다.
- 하나만 언급하고 넘어가겠다. sound라는 것이 continuous한 것인데 컴퓨터에 담으려면 하나의 값으로 담겨야 한다. 그래서 1초동안 얼마나 많은 값들이 뻑뻑하게 혹은 음성음성 있는 정도가 sampling rate라고 한다.
- 만약 10000이라고 rate를 해놓았다면 1초에 10000개의 값이 있는 것이다. Sampling rate 는 4만4천100 정도는 옛날의 CD같은 건데 사람이 구분할 수 있는 가장 높은... 것이다.
- 1초를 표현하는 데 소리를 표현할 수 있는 숫자를 의미한다. 그래서 이 개념을 사용하면서 다음 시간부터 적용할 것이다. Sampling rate를 잘 이해하고 오세요.

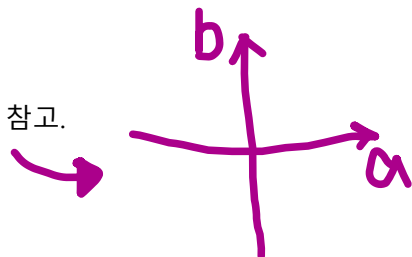
19/11/05

1. 몇 도: 0' ('는 작은 동그라미가 위에 있는 것으로 친다)

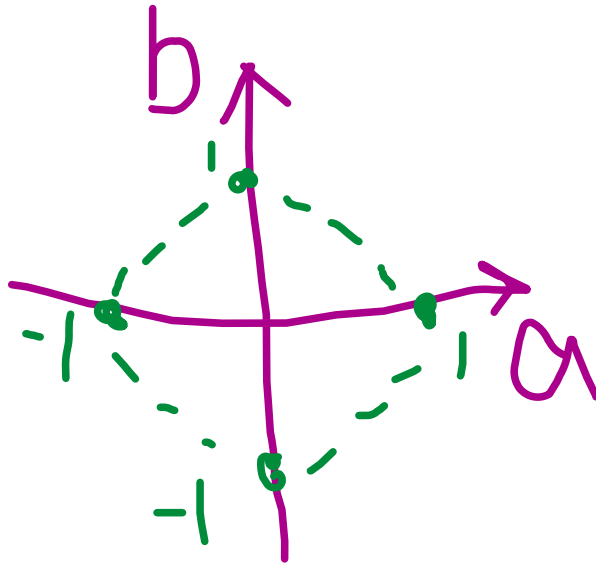
- 1) sinusoidal - sin과 cos 형태의 그래프를 만들어 내는 것
 - 2) phasor - 그냥 곡선 형태의 그래프를 만들어 내는 것
2. degrees: $0^\circ \sim 180^\circ \sim 360^\circ$
- radian: $0 \sim \pi \sim 2\pi$
- cos graph (인강 보고 그리기)
 - sin graph
- 0부터 100파이까지 반복되면 2파이가 50번 반복된다. Rhythmic 한 게 50번 반복되는 것이다. (?) – 인강 다시 확인. 잘 못 들었음.
 - 어차피 집 가서 정리는 인강 듣고 해야 할 것 같으니까 수업 시간에 한 번 듣고 이해하는 쪽으로 하기
 - 만약에 더 깊이 공부하고 싶으면 Euler 오일러 공식에 대해 따로 조사해봐라.
3. "i"는 imaginary를 뜻하므로 허수를 뜻한다. 루트 -1이 되는 값임. 제곱을 하면 -1이 되는 값.
 4. 복소수는 모든 수의 카테고리를 포괄하는 관계이다. 실수와 허수 모두를 포함하기 때문이다. 복소수의 수식 정의는 " $e=a+bi$ "인 것이다. 모든 수는 이 수식으로 표현이 가능하다. 따라서 e도 저렇게 복소수로 표현이 된다.
 5. Θ (theta)가 0일 때~

$\Theta = 0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}, 2\pi$

 - ➔ 차례대로 1, i, -1, -i, 1
 - ➔ [1, i, -1, -i] 이렇게 4개가 반복되는 것이다
 - ➔ Euler function에서는 어떻게 이를 표현하게 될 것인지 지금부터 배울 것이다.
 - ➔ $A+bi$ 라는 복소수를 PLOT하는 방법
 - ➔ 복소평면을 complex plane이라고 한다. 그래프 참고.



→ 복소평면에서 저 위의 $a+bi$ 를 그리게 되면! 아래와 같이 된다. 그 각도 값을 만들기



- 벡터 값으로 표현이 될 수 있다.

- Projection – flashlight을 비출 때 어떻게 나오는지 보는 것임. (투시, 투영)

- Projection의 2개의 방향

- 1) 허수를 신경 쓰지 않고 실수만 보고싶다 (실수만 본다면 \cos)

→ 방법은 complex plane을 머리 위에서 혹은 우측에서 보면 된다.

- Cos graph는 0에서 1부터 시작한다는 사실을 기억하기.

- 2) 실수를 신경 쓰지 않고 허수만 보고 싶다 (허수만 본다면 \sin)

→ 방법은 complex plane을 옆에서 대각선으로 보면 된다.

- Sin graph는 0에서 0부터 시작한다는 사실을 기억하기.

- Euler Phasor 은 \cos 와 \sin 의 성질을 동시에 볼 수 있는 훨씬 더 advanced된 phasor이다.

6. 이제 Jupyter Notebook으로 Phasor을 setting하는 법을 코딩하기 시작하려고 넘어감.

- 그 전에 설명: 소리라는 개념의 실체는 시간의 개념이 들어 있어야 한다.

- Phasor/ In [160]:부터 시작하면 된다.

- Amp=amplitude

- Sr=Sampling rate는 음질의 해상도를 의미한다.

- Frequency=sine frequency를 의미함.

7. [In [161]과 [In [162]가 가장 중요한 개념 중에 하나들이다.

```

19/11/05

sinusoidal - sin과 cosin의 형태의 그래프를 만들어 내는 것
phasor - 그냥 곡선 형태의 그래프를 만들어 내는 것

In [5]: # parametersetting
        samp=1 # range [0.0, 1.0]
        sr=10000 # sampling rate, Hz
        dur=0.5 # in seconds
        freq=100.0 # sine frequency, Hz

t=> 0.0001 0.0002 0.0003 ... 0.5000 이런 식으로 제일 먼저 시간을 설정해야 한다. 그래서 이 시간을 만들기 위해서 아래와 같은 수식을 사용한다.

np.arange(1, sr*dur+1) 이렇게 duration을 급하게 되면 10,000이 아니라 5,000까지만 설정할 수 있게 된다. 여기서 sr로 나누어 주면 time이 되는 것이다. 그러
면 만분의 1초부터 0.5초까지 만들어진다. 따라서 최종적인 수식은 > t=np.arange(1, sr * dur+1)/sr

In [10]: t=np.arange(1, sr*dur+1)/sr

NameError                                Traceback (most recent call last)
<ipython-input-10-a2cd44a70768> in <module>
----> 1 t=np.arange(1, sr*dur+1)/sr

NameError: name 'np' is not defined

```

- Frequency는 몇 바퀴를 돌아라는 개념을 설정할 수 있는 것이다.

- Plotting을 하는 방법 – In [164]를 참고하기!

Fig=plt.figure()

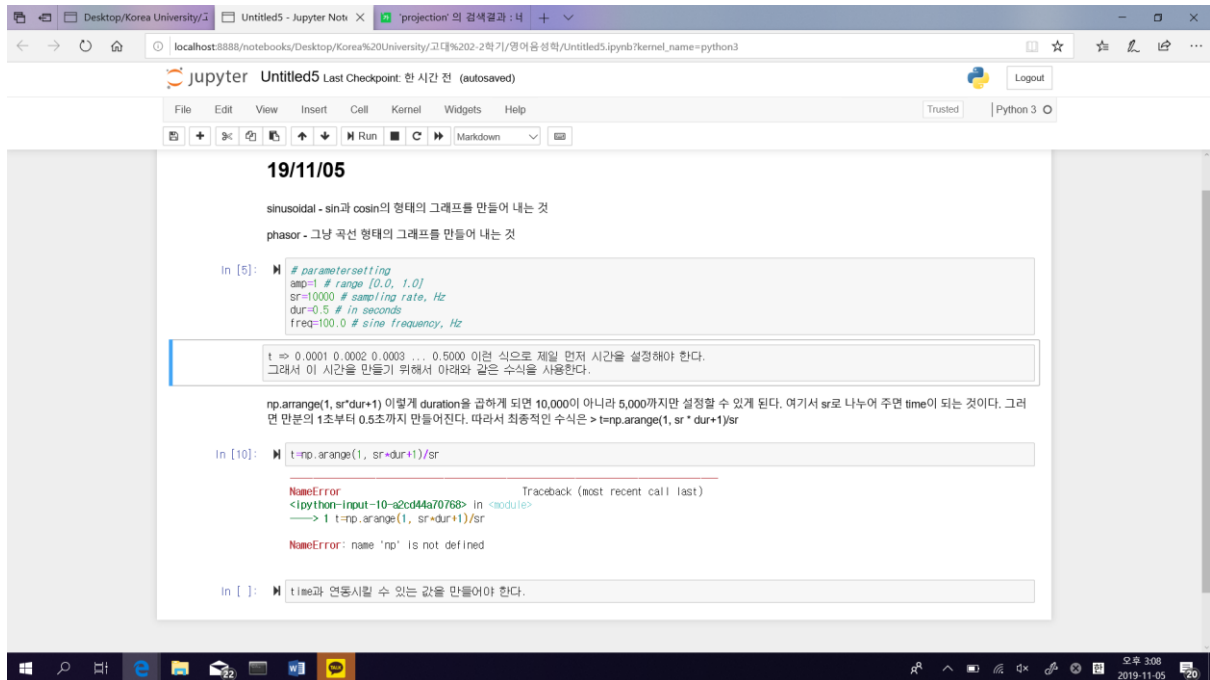
Ax=fig.add_subplot(111)

Ax.plot(t[0:1000], s[0:1000], '.') > 3번째 plotting값은 .을 쓰라고 했다.

Ax.set_xlabel('time(s)')

Ax.set_ylabel('real')

- 소리 만드는 법도 배웠다.
- lpd.Audio(s, rate=sr)라고 치면 소리가 나옴!



19/11/07

1. Sound- Jupyter Notebook/Phasor/In [19]: # parameter setting 부터 수업 시작.
2. 첫번째 function은 plotting 하기 위한 것이다.
3. Theta – 각도 값을 지정할 때 사용함.

숫자 값을 지정할 때는 np.arange()를 사용해서 괄호 안에 뭔가를 넣어서 각도의 벡터(0 부터 2π 까지)를 만드는 것이 우선적인 행위이다.

- 제일 첫 번째 값은 0, 제일 마지막 값은 2π 라고 하면 되기 때문에, > theta = np.arange(0, 2*np.pi,)라고 적고 실행시키면 된다.
- 그리고 같은 행 다음 줄에 theta라고 적으면, Out 값이 array([0., ~ 6.])까지 나온다.
- 다시 한 번 말하지만 theta는 지금 radian으로 정의한 것이다.
- 이 두개를 plus 하면 sine 곡선이 나온다.

4. In [9] 후에 나오는 Out[9]는 4개의 그래프가 나오는 상자가 나오는데 좌측 상단은 1, 우측 상단은 2 좌측 하단은 3 우측 하단은 4라고 부른다.

- y축은 sine 함수의 결과
- sin은 0부터 시작, cos는 -1? 부터 시작

5. x는 theta in radians / y는 value로 설정하기.

- X와 y의 관계가 line의 형태가 아닐 때에는 non-linear이라고 하면 된다.
- 시간은 소리가 반드시 있어야 하고 또한 소리는 시간이 반드시 있어야 한다.
- 시간을 만들 때 `t= np.arange(1, sr)`식으로 적으면 > 1초라면 time tick(?)의 개수는 sampling rate와 일치한다.

6. Time을 연동시키는 방법 => #generate phase

➤ `Theta=t * 2*np.pi * freq`

#generate signal by cosine-phasor

➤ `S=np.sin(~~)`

- Time, theta, s는 다른 것임

7. 실제로 plot을 할 때는 x축에 굳이 theta를 쓰지 않고(어차피 한 바퀴 도는 게 뻔하기 때문에) time을 적는다.

- In [24]에서 plot을 `t[0:1000]`, `s[0:1000]`까지 했을 때 나오는 그래프에서의 점의 개수는 1000개이다. 그리고 만약에 `t[0:1000]`, `s[1:1000]`일 경우에는 실행이 안된다. 그 이유는 ...(영상 확인)

8. In[13]: #generate signal by complex-phasor

`C= np.exp(theta*1j)` > 1j는 i라고 생각하면 된다. (복소수의 범위까지 확대가 되었다.)

Exp도 함수이다.

- 컴퓨터는 정보량을 정할 필요가 있다.
- Complex number은 복소수라고 한다.

9. In [19] 확인하면, 이제 figure을 만들 수 있다.

Prejection을 '3d'로 설정하면 그래프가 3d로 들어가게 된다. (시간이라는 개념 추가)

- 축 설정할 때는 `ax.set.z/y/x('imag','real','time(s)')`
- Complex phasor은 sin 도 갖고 있고 cos도 갖고 있어서, 두 가지의 얼굴을 가지고 있다고 볼 수 있다.

10. ipd.Audio라는 것을 사용할 수 있다.

➤ `lpd.Audio(s, rate=sr)`

➤ 다