Week 1 2018130841 손윤지

There are a majority of English sounds. They can be classified as consonants and vowels, as large. In terms of sounds, we can have a study on sounds, and this one is named as phonetics. Phonetics is a study that figures out how speech is described. To be specific, there are three types of phonetics: articulatory phonetics, acoustic phonetics, and auditory phonetics. First of all, articulatory phonetics is the study about the production of speech. This is the most primitive step of speech. The very first step of articulation is to release air from lung to vocal tract. Vocal tract is composed of larynx, pharynx, nasal tract and vocal tract. To take a closer look, vocal tract is organized by upper side and lower side. Upper vocal tract includes upper lip, upper teeth, alveolar ridge, hard palate, soft palate(velum), uvula and pharynx wall, from the front. In addition, lower vocal tract constitutes lower lip, tongue, and epiglottis, also from the front.

Now again about the speech process, there are 5 speech organs, which are constrictors, also articulators, that make the actual sound during the speech. They are lips, tongue tip, tongue body, velum, and larynx. Phonation process occurs in larynx. Every sound appears through the larynx. Also, the sound from larynx can be divided by voiced sound and voiceless sound. Voiced sound is the sound that occurs by the vibration of vocal cords, and the examples are b, m, v and so on.

Oral-nasal process happens by the position of velum. When nasal sounds are made, velum goes downward so as to make the air go through the nose. Otherwise, velum goes downward, when it is oral sounds. Articulatory process is created in lips, tongue tip and tongue body.

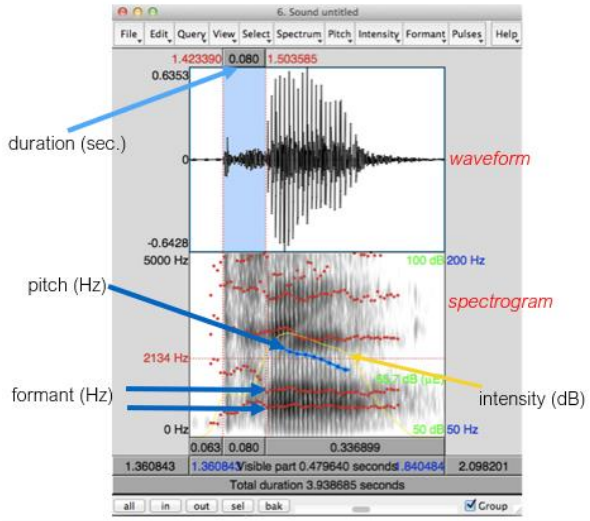
Moreover, each constrictor can be specified by constriction location and constriction degree. By the touch of lips, tongue body and tongue tip, the air flow is constricted at such location, and then it appears as different sounds. Constriction degree is about how much constrictions are made. The very much constriction made, we call it stops. The order goes as fricatives, approximants, vowels, from the most. Therefore, phonemes are produced by specifying constrictors, constriction location, and constriction degree.

Phonemes are individual sounds that form words. Also, phonemes can be defined as a combination of speech organs' actions. At lips, p, b, m, f, v, and w sounds are made. By tongue tip, ɵ, ð, ʒ, t, d, s, z, ʃ, l, and r sounds are made. By tongue body, k, g, j, ŋ and vowels are made. By velum, m, and ŋ sounds are made. By larynx, ɵ, ʃ, p, f, t, s, k and h sounds are made.

We can figure out acoustic process in Praat. We can check duration, pitch, formant and intensity in Praat. To verify voewl acoustics, we can measure pitch using Praat. By the number of occurrences of a reapeating event per second, we can find out the Hz. Then the repeating event shows the vibration of vocal folds, and the repeating parts show us a sine wave.

Week 2 2018130841 손윤지

Acoustic process in English. We can find out acoustic process by using Praat program.



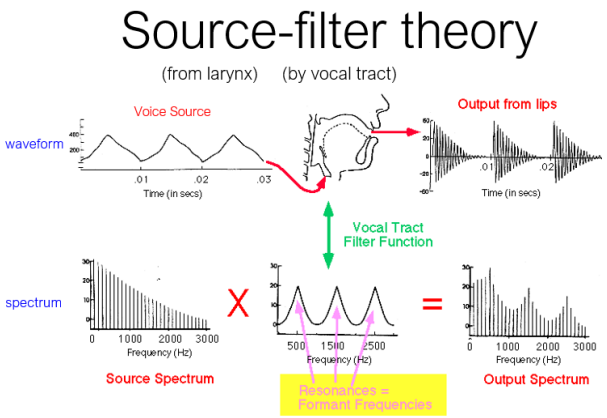
By Praat, we can figure out the waveform, spectrogram and other more information. To be specific, intensity shows the decibel, pitch shows the hertz. Formant is the darkest line of spectrogram. Also, spectrogram is the one that turns spectrum in to time line. Now, having a closer look at the vowel acoustics, all of us have different pitch, so the sine wave(pure tone) of our sound looks different.

If we listen to a sentence, we can notice that it is not a pure or simplex tone. Thus, the sound we perceive through the mouth is complex tone. In addition, how sound like differs depending on our vocal tract. Listening to the source of the sound, there are just sound and pitch. According to sign wave, x-axis describes time line and y-axis describes the value. Moreover, you can turn the sine wave into the spectrum. Spectrum shows frequency by x-axis, and amplitude by y-axis. Simple tone of 100hz is a fundamental frequency, which we will call it as F0.. By synthesizing all simple tones, harmonics are made, and we call it complex tone. In terms of complex tone, the frequency of complex tone is same as that of fundamental frequency.

Human voice source consists of harmonics. Harmonics are composed of harmonic overtone. As usual, female has less harmonic overtone than male. After that, if filtered by vocal tract, the organization of harmonic overtone is same but magnitude differs from that of simplex tone. In other words, peaks and valleys appear in the complex tone.

In conclusion, the lowest pure tone is fundamental frequency, the rate of vibration of the larynx and the number of opening-closing cycles of the larynx per second. Amplitude of pure tones gradually decreases. On the other hand, if the sound is filtered by the vocal tract, spectrogram shows formants which is the peaks that frequencies vocal tract likes.

By means of Praat, we can synthesize source ourselves. As we create one lowest pure tone and make its harmonic overtone, we can make harmonics. In spectrum, there is a mountain from the beginning and we call it F1 and F2 in order. In spectrogram it appears as formants. Each vowel has distinguishable formants and that why we can classify each vowel. Also, if we put F1 as y-axis and F2 as x-axis, we can watch vowel space.



Week 3 2018130841 손윤지

Computer program also has a language. We call it coding as usual. There are four important aspects of coding. First, there are variables which are same as the words of language. We need a function that makes system to work on, which is called 'if conditioning'. Next, for loops are needed for the repetitive act of program. Lastly, only number and characters could be the information in computer program.

By using 'anaconda prompt' we can figure out the way to test the function and programming. First of all, turn on the 'anaconda prompt', type 'jupyter notebook' on the black monitor. Then the program leads us to the jupyter notebook program. Then, now we are ready to practice to use the program.

The thing located on right is the information, and the one on the left is the variable. In the case of making a=1, 1 belongs to information and a to the variable. Then, put the function of 'print ()', which make the variable print out its information, the result would be 1. If we make new command such as 'a=2', the result of 'print' would be '2'. Thus, what we have to focus on is the number between the square brackets. After clicking 'in' at the left side, we can make new line below with tapping 'b' on the keyboard, make new line up above with tapping 'a' on the keyboard, and delete the line with tapping 'x'. Also, you can make the result by clicking 'run' box, or by tapping shift+enter.

Every character just used in the program are perceived as a variable, so to make it as a literal word of English, we need to put quotation marks such as 'love'. In addition, if you put any variable twice, especially for the last variable, when just typing variable without function, it makes same function as print function.

In the case of matching more than one information to one variable, you can put those information inside '[]'. After having type function, the result is 'list'. When there is just one information which belongs to one variable, the result of type function is 'int'. On the other hand, information of prime number and the word make the result of 'float'. When there are the word that

meets the variable, the result is the ‘dictionary’.

There could be both number and letters in the 'list'. Dictionary need more than two pairs of variable and information. Also, it is allowed only to use {}, while using dictionary. Int appears when it's natural number, and float appears when decimal. When the letter information is included, it is called as a string. Tuple is the one similar to list, but the difference is that tuple uses [] can be used when access to the information inside, such as for order. So, when it is written as 'a[2]', it means the third order of the letter. It's because the thing that comes first is regulated as '0'.

```
a = '123'  a = list(a); print(type(a)); print(a); print(a[2])
```

```
<class 'list'>
['1', '2', '3']
3
```

If it becomes as a formation of dictionary, the index becomes the variable itself, instead of using the order. Function of string is similar to that of list. In the case of order written in minus, you can think of it as a vertical line. Thus, [-1] always means the letter of the last index, and [0] always means the first. Moreover ':' shows the range of the information.

```
s = 'abcdef'
print(s[0], s[5], s[-1], s[-6]); print(s[1:3], s[1:], s[:3], s[:])
```

```
a f f a
bc bcdef abc abcdef
```

Function 'len()' shows the length of the information. You can compound the results by using plus or *. Using period after the variable and writing upper makes the information becomes upper case.

```
s = ' this is a house built this year.\n'
s
```

```
' this is a house built this year. ₩n'
```

```
result = s.find('house')
result
```

```
11
```

```
result = s. rindex('this')
result
```

```
23
```

*rindex means to find the information at the right side.

s = s.strip () s
'this is a house built this year.'

*strip: wipe out all the inaccurate factors of the sentence.

Tokens = s. split (' ') tokens
'this', 'is', 'a', 'house', 'built', 'this', 'year'

*split: split the sentence by following order in the '()'

```
s = ' '.join(tokens)
s
```

'this is a house built this year.'

*join: put up the words into a sentence using the factor between '()'

s = s. replace('this', 'that') s
'that is a house built that year.'

One of the important processes of programming is 'for loop' process. The way to make for loop process is writing down 'for_ in _ :' in 뒤에 있는 것을 하나씩 돌려서 in 앞에 있는 것이 받아서 계속 동작을 반복하라는 것이다.

```
a = [1, 2, 3, 4]
for i in a:
    print(i)
```

1
2
3
4

Range 함수는 range 뒤에 어떤 숫자가 나오면 몇 개의 index를 만들라는 것.

```
a = [1, 2, 3, 4] #index 를 4개를 i 에 넣어라.
 for i in range(4):
       print(a[i]) #print (i) 의 결과는 0 1 2 3임.
```

1
2
3
4

Len () 함수는 길이를 의미. Len (a) = 4

```
a = ["red", "green", "blue", "purple"]
b = [0.2, 0.3, 0.1, 0.4]
```

```
for i, s in enumerate(a): #i가 번호고(0, 1, 2, 3) s는 자기 자신.
    Print (a[i])
```

Red

Green

Blue

Purple

Enumerate: 번호를 추가로 매겨라.

```
a = ["red", "green", "blue", "purple"]
b = [0.2, 0.3, 0.1, 0.4]
for i, s in enumerate(a): #zip 은 페어로 4 개가 됨.
   print("{}: {}%".format(s, b[i]*100)) #왼쪽과 같은 형태로 적고 싶을 때.
Format 뒤에 두개가 왼쪽과 같이 적힌다는 것.
```

```
red: 20.0%
green: 30.0%
blue: 10.0%
purple: 40.0%
```

```
a = 0
if a == 0:
   print(a)#두개 써야 진짜 equal 사인
else:
   print(a+1) #if a != 0 > !가 0 이 아니면 이라는 뜻.
```

0

```
for i in range(1, 3):
     for j in range(3, 5):
          print(i*j)
```

3

4

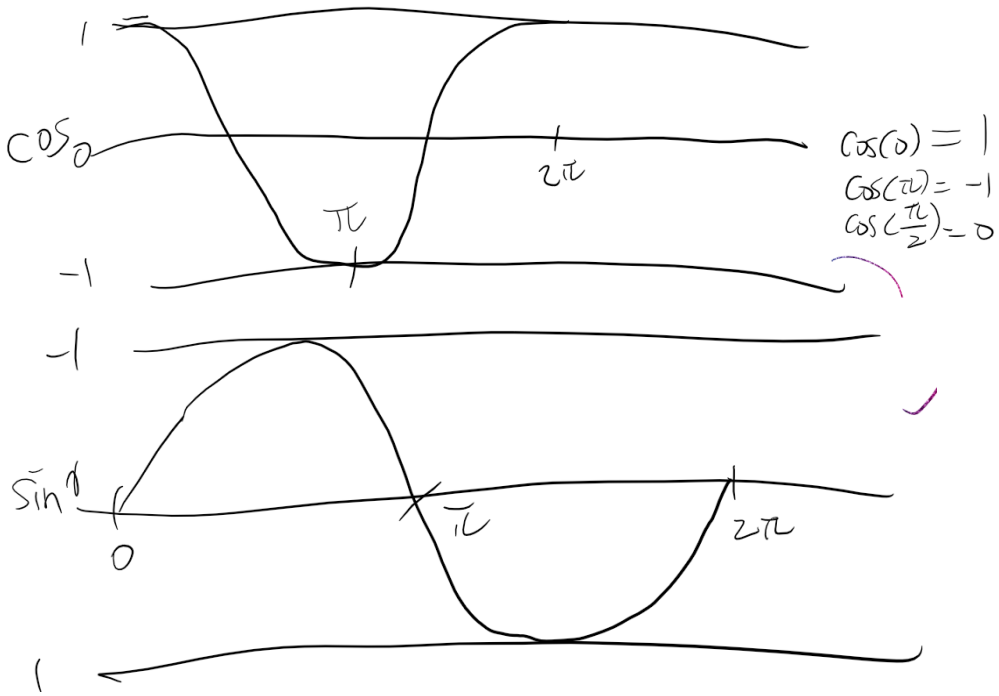
6

8

모든 데이터는 행을 늘어놓은 벡터의 상태로 이루어진다. 모든 이미지는 행렬로 이루어져 있다. 흑백 이미지가 3층이 되면 색을 나타낸다. 동영상은 그 이미지들이 시간 순으로 놓인 것이다. 이를 차원의 개념으로 생각해 볼 수 있다. 흑백 이미지는 2차원이고 색 이미지는 3차원, 동영상은 4차원이다. Wave form 과 텍스트 모두 벡터로 데이터화 할 수 있다.

Numpy는 list와 비슷한데 list와 달리 수학적인 데이터로 쓸 수 있게 하는 것이다. Import는 누가 만들어 놓은 library를 부르는 작업이다. 이를 직접 사용할 때에는 import numpy 로 사용하면 된다. 그리고 '.'은 그 안에 있는 subpackage를 빼낼 때 사용한다. 예를 들어, import numpy. A. D. function() 이라면, numpy라는 library 안에 있는 subpackage A를 사용한다는 것이다. 이는 From numpy import A. D 와 같은 것이다.

Python을 이용한 코딩으로도 소리를 분석하고 만들 수 있다. 이를 시행하기 위해서 미리 알아야하는 것이 싸인, 코싸인 함수이다. Radians 는 각도 값으로 이를 pi 값으로 표현할 수 있다. 따라서 0도는 0으로 표현이 가능하고 2*pi는 360도가 된다. 이를 그대로 sin, cos 값으로 함수를 그릴 수 있다.



함수를 그리면 위와 같은 형태로 나타난다. 또한 이 두 함수를 phasor로 볼 수 있다. Cos(θ), sin(θ)의 값으로 표현하여 숫자 값으로 나타낼 수 있기 때문이다. 데이터는 숫자로 볼 수 있으며 이를 벡터라고 할 수 있다. 따라서 오일러 공식이 필요하다. 오일러 공식은 다음과 같이 나타낼 수 있다.

$$e^{ix} = \cos x + i \sin x$$

θ값 중 주요 값들을 이 공식에 넣어서 복소평면(complex plain)으로 보면 원이 나타난다. 이 원에서 보면 project하면, cos 값이나 sin 값만 볼 수 있다. 이렇게 만드는 것은 단순한 벡터 값이고, 소리를 만들고자 하면 시간 값을 넣어줘야 한다.

Sampling rate는 1초에 몇 번으로 나누는 것이고 이는 곧 음질의 정도를 이야기한다. 만약 sampling rate가 10000이라고 하면, 1초에 10000개의 지표를 만드는 것이다. 소리에는 이에 더해 frequency도 필요하다. 그렇다면 sampling rate이 100이고 frequency가 1hz라고 해보자. 이 경우, sine wave를 한 번 그려주면 된다. 같은 sampling rate에 frequency가 100hz라고 해보자. 이 경우는 불가능하다. 표현할 숫자가 너무 적기 때문이다. 그렇다면 최대 몇 개까지 표현할 수 있을까? Sampling rate의 반까지만 가능하다. 이를 Nyquist frequency라고 한다. 즉, Nyquist frequency는 최대로 표현할 수 있는 frequency이고 이는 'sr/2'이다. 만들어 놓은 sound 값에 amplitude를 곱하면 사람의 소리와 가까워지는 소리가 나타난다.

또한, sin과 cos값은 90도, 즉 2*pi만 이동하면 똑같은 함수를 나타낸다. 그렇기에 cos과 sin 함수가 바뀐다고 해서 소리가 바뀌지는 않는다. 이는 우리의 귀가 각도(phase)의 변화를 인식하지 못한다는 것을 의미하기도 한다. 그러나 우리의 귀는 frequency의 차이는 인식한다.

pulse train 역시도 programming을 통해 만들 수 있다. 음성의 기본 원리와 같이 최소의 harmonics와 frequency를 잡는다. F0를 정하고 sine wave 를 만들고 배음을 만드는 것이다. F0을 정하고 최대가 Nyquist frequency인 sr/2이므로 Fend 값을 sr/2 로 만들어 준다. 그리고 s를 계속해서 더하고 더해나가는 것으로 pulse train을 만들어간다.

Spectrum 역시 만들 수 있는데, spectrum은 타임을 나열하는 것이 아닌 한 타임 안에 있는 주파수를 보여주는 것이다. Def 로 만들고 싶은 function을 정의할 수 있다. 또한, return은 출력을 의미하는 것이다. 이후 기본 source에 vocal tract가 지나면 사람이 발음하는 소리처럼 나타난다. 이는 RG와 BWG를 통해서 만들 수 있다.

행렬은 직사각형 형태의 행과 열이 있는 것이고 벡터는 그 행렬 중에서도 긴 것이다. 최근의 인공지능의 경우 기계에 벡터로 된 데이터가 들어가 출력 값을 내는 형태로 나타난다. 그 기계. 및 인공지능의 자리는 행렬화 되어있다. 이렇게 벡터와 행렬로 출력 값을 만들어내는 것은 벡터와 행렬을 곱하는 과정으로 나타난다. 이 때 곱하는 방법은 벡터의 각 요소에 행렬의 각 열을 곱해주는 것이다. 인공지능은 여러가지 방법을 통해 학습을 해서 필요한 행렬을 만들어 낸 것이고 행렬의 곲으로 이루어지는 것이다.

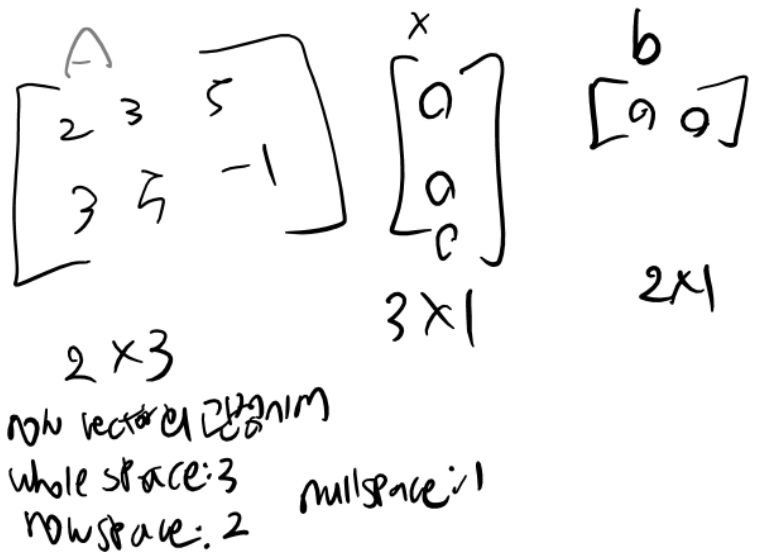
세로로 길게 생긴 벡터를 column vector라고 할 수 있다. 이를 x축과 y축으로 옮겨 표현할 수 있다. 행이 두 개 일 경우 2차원에서 한 개의 점으로 나타낼 수 있고, 행이 세 개일 경우 3차원에서 한 점으로 나타낼 수 있다. 그리고 이 점은 원점과 이어 1차 함수로 그려줄 수 있다. 또한, 벡터 앞에 상수를 두어 벡터 값에 곱해 변화를 줄 수 있다. 그리고 여러 개의 벡터를 각 요소별로 더해 줄 수도 있다. 이를 역시 함수로 그려줄 수도 있다.

Linear combination는 c*v+d*w 처럼 단순히 곱하고 더한 것이다. 이 때 v와 w는 벡터이고 c와 d는 어떤 값이다. Vector space는 여러 벡터들이 만들어내는 공간을 이야기한다. 이는 모든 벡터 값들을 생각해본 것이기 때문에 2차원을 덮는 모든 공간이라고 생각할 수 있다. R의 n승은 n차원을 의미한다. Column space는 column 벡터를 linear combination 해서 나타나는 모든 점을 의미하는 것으로 같은 차원의 space를 모두 덮는 것이 된다. 그리고 column space의 차원은 column vector의 차원을 넘지 못한다. Column vector의 값을 원점과 모두 이어 plane을 만들고 그것을 늘리는 것이 spanning이다. independent하다는 것은 원점과 이은 column vector의 값이 한 선에 있지 않다는 것이다. Dependent 한 경우 원점을 이어 아무리 확장시켜봤자 line이 늘어날 뿐이다. Whole space는 행의 개수를 의미하고 column space는 independent한 column의 개수이다.

행렬의 행과 열을 뒤집어줄 수도 있다. 그 경우 모양과 whole space는 달라질 수 있지만, column space는 변화하지 않는다. Column vector의 관점에서의 whole space, column space가 있듯이 row의 관점에서의 whole space와 row space가 존재한다. 그래서 행렬을 볼 때 두개의 whole space가 존재한다고 말할 수 있다. Whole space와 row/column space간의 gap을 null space라고 한다. Transformation을 하는 것을 기하적으로 생각해볼 수 있다. 또한, 역함수에 출력값을 하면 x값이 나타난다. 그러나 dependent한 경우는 불가능하다. 면적은 determinent와 같고 그 면적이 0이 된다는 것은 행렬이 dependent하다는 말이다.

어떤 행렬의 eigen vector는 transformation한 후의 결과값이 그 전의 값과 일직선상에 있는 것이다.

A에 x라는 행렬이 입력되어 b라는 값이 나오는 것이 인공지능의 작업 과정이라고 생각할 수 있다. 2x3 행렬에 3x1을 곱해서 2x1이 결과로 나오는 경우를 생각해보자.



Null space를 수학적으로 생각했을 때 나오는 식이 Ax에서 b에 해당하는 벡터 값이 0이 되게 하는 것이다. 즉, '어떤 입력이 들어오든 출력에 영향을 미치지 않는 것이다.' 현실에서 생각해봤을 때 소리 내는 데에는 영향을 미치지 않지만 동작을 화려하게 하는 등 skilled action을 보여주는 것과 유사한 경우이다.

Row space에 있는 점을 A에 곱하면 b 값에 영향을 미친다. 그러나 null space에 수평이 되는 그 어떤 값을 넣는다면 b 값에 영향을 미치지 않는다. 이것이 null space 의 정의.

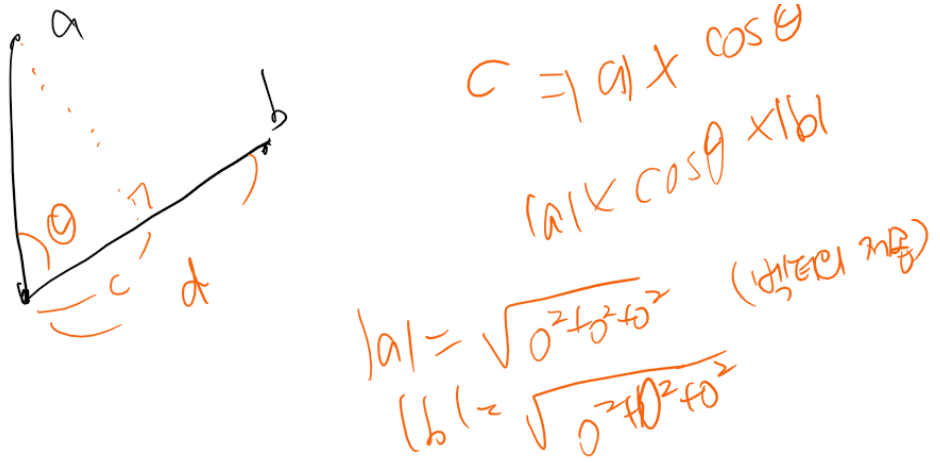
인공지능으로 생각해봤을 때, A에 matrix 값을 하나 받아 놓고 x 값의 방향을 따져 어느 방향으로 변하면 b 값에 영향을 주고 어느 방향으로 변하면 영향을 주지 않는 것으로 나뉜다. 이것이 인공지능과 선형대수와의 관계이다.

Given vector에 Eigenvector가 무엇이냐 물으면 그 space에 있는 eigenvector space라고 말할 수 있다. 2x2에는 eigenvector 2개. Eigen value는 각각의 eigen vector에 대해서 확대되는 비율을 의미. Eigen analysis를 하면 여러 matrix들을 쪼개서 훨씬 더 unique하고 고유한 해석을 가능하게 해준다.

선형대수를 통계와 연관지어 생각해보자. 상관관계는 r로 표현할 수 있고 '$-1 \le r \le 1$'이다. 상관관계를 가장 적게 보이는 경우는 r이 0인 경우이다. 같은 벡터 값에서 그것을 선형대수로 생각했을 때 두 값과 원점과의 각도는 r, 즉 correlation이 된다. 90도일 때 r 값은 0이 되고, 0도일 때 r 값은 1이된다.

Inner product=dot product인데, 원점과 두 벡터가 있을 때 각각 곱하고 더한 값이다. 이는 기하학

적으로 봤을 때 두 벡터가 있을 때 하나의 값에서 수직을 내리고 원점에서 그 점까지의 길이와 다른 벡터까지의 길이를 곱한 것이다.



이는 어떤 신호가 있을 때 어떤 주파수가 많은지를 보여주는 것이 spectrogram 인데 이를 만들기 위해서 알아낸 것이다. 이 때 여러 벡터(주파수)값들을 inner product해 그 값이 높게 나오면 correlation이 높은 것이다.

Eigen vector의 수식으로 보이려면 'Av=상수xv' 원래 벡터 값이 transformed 된 결과 값은 원래 벡터 값에 상수를 곱한 값이다. 여기서 상수가 사실상 eigen value.

Inner product는 차원을 줄여서 계산하는 방식. Outer product는 차원을 확대시키는 방법이다.

4차원 상에서 a벡터와 b 벡터 잡고 그에 대한 cos 각도 값을 cos correlation이라고 한다. 이는 a 값과 b 값이 얼마나 유사한지를 보여준다.

Sine wave가 있는데 이것은 다 점 하나하나의 벡터 값이다. 이것을 역시 inner product 할 수 있다. Phasor로 벡터 값은 똑같도록 여러 개를 만든다. Spectrogram으로 생각했을 때 진하면 많은 것이고 흐리면 적은 것이다. 이것이 시간 적으로 반복되며 이어짐.

같은 sine wave를 inner product 할 때보다 더 빠른 frequency의 sine wave를 곱하는 것이 작다. 왜냐하면 더 빠른 frequency 사이에서는 떨어지는 경우도 있기 때문이다. 즉, 똑같은 성분이면 값이 크게 나와서 반응을 보이고 다른 성분이면 다른 반응을 보인다. 그런데 똑같은 궤적의 sine wave라도 하나는 cos이고 하나는 sin인, 즉 90도의 차이를 보이는 두 sine wave의 경우 inner product하면 값이 다르게 나온다. 이는 그래프 상에 벡터를 나타냈을 때의 90도와 같다. 이렇게 phase의 이동에 대한 민감도가 좋지 않아서 complex phasor를 사용하여 inner product하면 된다.

벡터를 sine wave로 만들어주고 그것들을 inner product했을 때 correlation이 많으면 값이 높게 나오고 적으면 결과 값이 낮게 나온다. 이 것을 또 기하학적으로 생각해보았을 때 각도가 0에 가까울수록 결과 값이 커지고 90도가 될수록 작아진다. 이렇게 판단하는 것을 spectral analysis라고 한다. 이는 어떤 frequency가 얼마나 많이 들어있는 가를 inner product로 분석하는 것이다. 그러나 말했듯이 simple phasor로 계산할 경우 shift에 따라 결과에 영향을 많이 주기 때문에 그 영향이 적은 complex phasor로 계산을 한다.

Complex phasor로 inner product하면 이 자체가 complex phasor이기 때문에 inner product하면 결과값은 complex number로 나온다. 그렇기에 plotting이 불가능하다. 따라서 a+bi 값에 절대값을 씌워 실수 값으로 뺀다. 이 때의 절댓값은 x축이 a, y축이 b인 평면에서 (a, b)와 원점사이의 거리이다.