

Dot product의 값은 90도에서 가장 작음

$\cos\theta$ 에서  $\theta$ 가 90이면  $\cos\theta$ 는 0.

한 complex wave에서 어떤 frequency 성분이 많은 지 구하는 것이 중요

각 freq마다 그래프를 뽑아내서 얼마나 들어있는 지 구해야 한다.

Simple wave들이 얼마나 들어있는 지 알기 위해서는 dot product를 해야한다.

똑 같은 크기의 웨이브들을 inner product 해야함.

값이 클수록 많이 들어가 있음.

Simple wave를 쓰는 이유는 너무 phasor의 변화에 민감하기 때문. 같은 frequency를 가지고 있더라도 조금만 이동하면 0이 나올 수 있기 때문에 complex phasor를 사용함.

Complex phasor는 벡터이긴 하지만 허수도 포함하는 값을 가짐. 하지만 inner product를 해서 나오는 값은 complex number. (1x1)

만들어진 complex number는 허수부분도 있기 때문에 plotting이 불가능함.

Plotting을 하려면 값들에 절대값을 씌워야함.  $|a+bi|$

x축을 a, y축을 b라고 했을 때 원점부터 (a,b)까지의 거리가  $|a+bi|$ .

```
nSamp = len(s)
```

벡터의 사이즈를 결정해주는 것. Inner product를 하기 위해 target의 vector와 sample의 개수를 맞춰보는 것.

```
z = np.exp(omega*1j) ** (np.arange(0, nSamp))
```

```
= ewi*[0.....100]
```

```
omega = 2*np.pi*n/nFFT
```

```
w = 2π*0/100
```

```
w = 2π*1/100 .....
```

➔  $e^{[2\pi \cdot \frac{1}{100} * [0 \dots 100]]} = e^{[2\pi * 1 * [0 \dots 1]]} = e^{[0 \dots 2\pi]}$  100개의 샘플이 한바퀴 도는 그래프 완성

3번째 ➔  $[0 \dots 4\pi]$  두바퀴

nSamp 바퀴까지 돌아감.

```
amp.append(np.abs(np.dot(s, z)))
```

위에서 구한 z의 값을 target인 s와 dot product해서 절댓값을 씌움 ➔ plotting 가능한 값 나옴.

Append ➔ 비어있는 `amp = []`에 값 넣음

For loop가 sample의 개수만큼 돌아가기 때문에 `len(amp)`는 100.

또한 amp에는 허수가 들어갈 수 없음. 절대값을 씌우기 때문.

```
freq = np.arange(1, nFFT+1) * sr / nFFT;
```

```
freq = np.arange(1,nFFT+1)*sr/nFFT;
[1....100] * 10000/100 = [100....10000]
```

완성된 그래프는 절반의 frequency만 의미가 있음. 그 절반은 spectrum이라고 함. 각 에너지 성분들이 inner product해서 고집어 오는 에너지 값이다. 그것의 절댓값.

예를 들어 2000hz정도의 성분 값은 그래프의 amplitude만큼 있는 것이다. 그 점에서 가장 많은 값은 500hz임.

A = 국어, b = 영어라고 했을 때

A = [ 10 20 30 40 ..... 100], B = [0 10 20 30 40.....90]이면

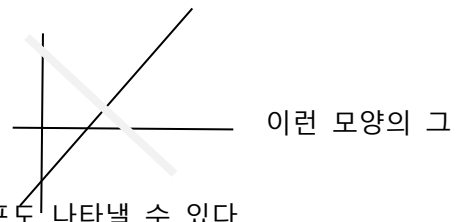
그래프도 나올 수 있고,

점수를 받은 학생의 수를 기준으로 분류하여 10차원의 그래프도 나타낼 수 있다.

10차원상의 두 점을 찍으면, 그 두 점은 원점에 대하여 각도를 가지고 그 각도는 두 벡터 사이의 similarity를 나타낸다.

만약 2차원상에서의 그래프에서 두 벡터가 일치하여 한 직선을 이루는 경우 correlation 값이 1이므로 10차원 상에서도 한 직선을 이룬다. 한 직선을 이루기 때문에  $\cos\theta = 1$ 이다.

R = -1인 경우 위 그래프에서 흰색 부분의 그래프가 해당하고, 10차원 상으로 옮기면 국어와 영어가 원점을 기준으로 반대방향으로 나아가는 그래프가 그려진다. 각도는 180이기 때문에  $\cos 180$ 은 -1이다.



그래프들의 similarity를 측정할 때 진폭은 의미가 없다.

스펙트럼에서 절반의 값만 이유가 있는 것은 그 이후부터는 이미 표현된 부분이기 때문에 의미가 없음.

```
win_size = 0.008
```

스펙트로그램을 만드는 것은 스펙트럼을 한장한장 이어서 만드는 것인데, 그것을 얼마만큼의 벡터들을 가지고 만들 것인지 정하는 것.

```
win_step = 0.001
```

위와 같은 과정을 0.001초를 움직여서 (0.001초의 간격으로) 실행하겠다는 것.

```
powspec = 1/nfft * (magspec**2)
```

스펙트로그램에 제곱을 한 것 → 진한 부분은 더 커지고 옅은 부분은 더 연해짐

이러한 제곱된 스펙트로그램을 파워 스펙트로그램이라고 한다.

이 과정을 거치는 이유는 로그를 취하기 위함.

```
logspec = 10 * np.log10(magspec)  
plot_spectrogram(logspec);
```

로그를 씌우는 것인데, 아주 작은 것이나 아주 큰 것이 적당히 크고 적당히 작은 값으로 나오게 됨. 우리가 다룰 수 있는 범위로 숫자들을 가져오는 것이 목적.

원래 스펙트로그램에 파워를 하고 로그를 씌우는 것이 순서.