11.19 화 영어음성학

행렬은 직사각형으로 생겼다.
벡터는 행렬중에서도 한줄짜리로 길게 되있던지,
인공지능

데이터 - 기계 - 데이터
벡터   벡터
숫자열이 꼭 같을 필요는 없다.
음성이 벡터의 형태로 들어가서 text형태의 벡터로 나오면, 음성인식이라는 인공지능이라고 한다.
text가 들어가서 뒤에서 음성이 나오면 음성의 벡터가 나오면, 음성합성이라고 부른다.
texxt(일본어) 들어가서 한국어 text가 나오면 기계번역
알파고 같은 경우에, 현재 바둑상태가 입력으로 들어가서 그렇다면 몇 번째 수를 두어야 하는가가 입력과 출력이다.

입력 형태의 벡터가 함수를 통해서 출력의 형태가 나오는 것이다.
기계를 인공지능이라고 부르기도 하고, 함수, 함수의 행렬이라고 불러도 됨.
인공지능에서 기계학습이야기를 하는데,
기계가 행렬이다.

데이터 부분은 벡터로 되어 있어야함

선형대수가 지금 AI 시대에 가장 중요한 게 선형대수이다.

5 3 0 1   -1 0 2
                 0 1 3
                  3 -5 7
                 2 3 4

5 3 0 1 은 이미지도 될 수 있다. 이게 중간의 행렬형태를 거치면, -- 누구의 얼굴! 표현되서 나온다. -- 표현되서 나오는 값도 벡터값이다.
어떻게 중간에 있는 행렬이 벡터를 바꾸는지?????
행렬의 젤 왼쪽의 칼럼이
5 3 0 1 이 -1 0 3 2 와 곱해지고, 0 1 -5 3 , 2 3 7 4 가 곱해져서 결과값이 나온다.
-3 6

인공지능에서 기계학습이 있는데 , 기계를 많은 데이터로부터 학습을 시켜서 저거 하나를 딱 얻어내는 것이다. 이게 인공지능이다.

인공지능은 행렬의 곱이다.
입력 벡터를 출력 벡터로 바꾸어주는 함수역할을 한다.
음성, text, 이미지 될 수있다.(입력, 출력 마찬가지)
모든 데이터는 벡터로 되어져야 하는 이유는 곱셈을 하기 위해서 그렇다.
이런 것들을 하는 것이 선형대수이다.
선형대수는 행렬이라고 생각하면 된다.

이것보다 더 심오하고 재밌는 선형대수의 세계
모든 인공지능이 선형대수인데, 모르면 안되기 때문이다.

Linear algebra

m행n열
m by n 행렬이다. m x n 행렬이다.

vector는 일종의 행렬이다.

길게 생긴 벡터를 column vector라고 한다. (열 백터)

v,w는 차원이 똑같은 길이가 같은 벡터이다. c는 단순한 숫자 – 곱하고 더하고 하면 linear combination이다. 이게 아닌 것은 나누고 몇 승하고 이런거..// 하지말고 단순히 더하고 곱한게 linear combination이다.

vector space - 여러 벡터들이 만들어내는 공간이다. (여러 벡터라는 것이 중요하다)
2차원공간에서 그 벡터를 표현가능하다.
(3,4)같은 똑같은 차원의 벡터를 무한개를 만들어보자. (-1, ...) 무한히 많이 만들었을 때 그 vector space가 차지하는 공간은 그 2차원을 싹 다 덮는다. 2차원의 공간을 다 덮으면 vector space이다. 1사분면만 vector space가 될까? 안된다.linear combination을 통해서 ,,,, 1차원에 있지만 그걸 곱하기 더하기 하면 어디로 튈지 모른다. 즉, 일부분들은 vector space가 될 수 없다. 2차원, 3차원 모든 공간이어야 된다.

실수의 1차원 공간 R의 1승)), 실수의 2차원 공간 R의 제곱)) -- 모두 벡터 스페이스이다.

R의 n차원의 공간은 n개의 component – 3차원의 vector space는 3개의 component의 모든 벡터들이 모두 채워져있는 공간을 n차원의 벡터스페이스라고 한다.

1차원 - 모든게 채워져 있는게 1차원의 벡터스페이스
2차원 -
3차원 -

다 마찬가지이다.
n차원은 n개의 component가 그 공간을 다 채우면 된다.
column space

행렬을 대문자, 행렬 중에서 벡터를 소문자로 쓴다. 인공지능하고 기계말했는데, 입력벡터 출력벡터 – x가 입력벡터, b는 출력벡터 - 두 차원은 달라질 수 있다.
A는 행렬인데, 이게 몇바이 몇이냐에 따라 해줄 수 있다.

기계는 행렬함수라고도 하고, linear transformation라고 부른다. 왜linear라고 할까? 곱하고 더했으니가 linear이다. transformation이란, 4개짜리를 3개짜리로 바꿨는데, 차원도 바꾸고 숫자도 바꿨으니까 transformation x를 b로 바꾸어주는 transformation matrix는 뭔가? A이다.

determinent = 행렬식이 0 이되면 역함수가 없다. dependent한걸 다시 돌리려면 안됨.
다이아몬드 같은 면적이 determinent와 동일하고 이 면적이 0 이 된다는 말은 dependent하다. 같은 선상에 열 벡터가 2개 있으니까 그 면적이 0 이된다.
행렬식 그리고 invertibility를 말해주는 행렬식이 0이 됨과 동일하고 왜 inverse가 존재하지 않는지 설명해준다. determinent는 기하적으로 다이아몬드 같은 면적과 같다. transfomration의 열 벡터 그 면적과 똑같다.

11.21 목
행렬
{ 1 3 } [6 2 0 [15 29 3
5 1 x 3 9 1] = 33 19 1
6 -1 33 3 -1]
2 x 3
3x2 2x 뭐 3 x 3
2 부분이 똑같아야 한다.
인접한 두 수가 똑같아야 곱해진다. 똑같지 않으면 곱해지지 않는다.
A
[1 3 [6 [15
5 1 x 3] = 33
6 -1] 33]
3 x 2 2x1 3x1
기계함수 입력 출력

입력과 출력은 모두 벡터화해서 표현해야 한다. 입력이 들어가서 출력이 나온다.
[6
3] 을 A 앞에 두면 곱해지지 않는다. 곱하려면 어떻게 해야하는가?1 b y 2 하면 된다. [6,3]

[6,3] x [1 5 6   [15 33 33]
3 1 -1] =

Ax=b,   xAt=b   x A b 다 transpose된다.

A
[1 2
-1 0
3 5]

3 x 2

칼럼쪽을 생각해보면 3쪽이다.   점 이 세 개니까 1 -1 3 이고 2 0 5 면 3차원상에서 찍어야 한다.칼럼 차원에서 3이면 whole space이다. column vector가 represent 되는 세상은 3차원의 whole space 이다   3차원상에 점을 찍고 칠해보면, 세 점을 찍으면 plain위에 있는 삼각형이 되는 것이다. 이것을 더 밀가루 밀 듯이 밀면 전첼ㄹ 커버하게 된다. 이것을 spanning이라고 한다. 2차원의 plane이 나온다. plane = column space //whole space는 3차원인데, /// plane = column space 와 whole space는 안똑같다. whole space가 더 크다. 다른 한 차원은 null 인데 left null이다. left null space이다.
플레인과 수직인 것은 하나가 있다. 한 선을 생각할 수 있다. 이 선은 1차원이다. 이것이 null space이다. column vector가 만들어내는 space와 수직인게 null space이다. whole space에서 column space를 뺀 선 하나를 생각할 수 있다. 이렇게 해서 전체를 커버한다. 그래서 총 3차원이 된다.
spanning이라는 것은 linear combination으로 표현가능한 모든 것. 기하적으로는... column vector두 개를 linear combinatino을 한다. 모든 가능한 Bs 가 column space가 되구나. spanning한 plane을 넘어서지 않는다.
[1 2
-1 0
3 5]
a b
+
B

row space의 whole space는 2차원
A
[1   2   이것들이 스페닝 하는 공간은 몇차원인가? 2차원을 넘어가지 않는다.
-1 0
3   5]
-- 좌표평면에 점 찍고, 원점을 포함하여 삼각형 그리기
independent한 것이 몇 개 인가가 중요 - column space에서 independent한 건 2개 :

column space의 spanning된 space와 똑같다. = rank

row vector는 3개이다. independent linear combination으로 만들어지면 안됨.
c는 a, b 에 대해 dependent 하다. independent한 것이 2개 밖에 없다. = rank

column 이든 row든 independent한 숫자는 같다.
3 x 2
3은 columnized whole space, 2는 rowrized whole space
2 = 2 (같아야 하고 이를 rank라고 부른다.)
column space , row space
1 0
null space null space
null space는 다를 수 있음.

[1 2
-1 0
3 5]
xA= 0 A에 다가 뭐를 곱해도 0되는 무엇 =x

xA = 0 x는 반드시 왼쪽에 있어야 한다.
x = 1x3, A= 3x2

한 벡터(숫자의 열, sequence of numbers) 는 차원에서의 점이다.
0.5[6
8] 0.5는 1x1 matrix 이고, 스케일러이다.

[6 [-1
-2 + 0
4] 2]

ws 는 3차원이다.
linear combination해서 만들어지는 점은 두 개를 합해도 여전히 plane위에 있다.
이 두 개가 만들어내는 linear combination 중 하나일 경우,

linear combination하면 scaling해서 더한다.
R3 = 3차원 속에 있는 모든 공간은 3차원으로 표현하는 모든 벡터들

column space는 행렬로 설명,

1. 어떤 matrix있다면, column에 해당하는 벡터들이 spanning 된 것이다.
2. column vector들이 linear combination으로 만들어내는 모든 공간

whole space의 포함관게는 subspace이다.
m차원이 columnized whole space이고 column space는 그걸 넘어갈 수 없다는 점에서 subspace라고 불린다.

1차원은 선, 0 차원은 원ㅇ점, 3차원은 공간, 2차원은 plane

row로 바꾸면 row space의 정의가 된다.
n이 rowrized whole space보다 같거나 작은게 row space이다.

2차원의 평면상에 직선으로 그려지면, ws는 2차원이고, cs는 1차원이다.

row, column space차원은 같아야 한다. column space dimension 3, row space dimension 3 = rank = independent

[1 2 -2 [1+1=2 [1+1x2=3
2 4 0 2+1=3 2+1x2=4
4 8 5] 4+1=5] 4+1x2=6]
WS : R3
CS : P

[1 2 4
2 4 8
4 8 16]
WS : R3
CS : L
null space : P

given that matrix
Ax = [1 2 [0
2 4 x [x1 = 0
3 6] x2] 0]
여기는 무조건 2차원이 와야 한다. (right null space)

xA = [x1, x2, x3] x [1 2
3차원, 2 4 = [0,0]
(left null space) 3 6]

Ax= b
A는 matrix이고, x와 b는 matrix중 vector이다. x와 b의 차원은 같을 필요는 없다. A에 의해 b바뀜

ex
[1 0
 0 1]을 미리 만들어 둔다.

역함수의 경우
 determinent가 면적과 동일하다. 즉, 면적이 0인 것.
1x0,125 – 0,5-0,25=0
column vector 2개가 일직선 상에 있는 것과 일치
AD-BC=0, inverse matrix가 존재 하지 않는다.

[1.25   0.25   [1
0.25   1]   1]
transformation   [1
                       1]은 transformation하면 원점과 일직선 상에 있지 않으므로,
[1.25   0.25
0.25   1] 의 eigenvector가 뭐냐? eigenvector하면, 어떤 행렬의 egien vector가 무엇이냐는 질문이다.
 [1
 1]은 eigenvector가 아니다. transformation하면 원점과 일직선 상에 있지 않으므로,
다른 벡터는 transformation해도 원점과 일직선상에 있다. -- 이것이 eigenvector이다.
eigenvector하면, 어떤 행렬의 egien vector가 무엇이냐는 질문이다.

11.26 화
실용적인 측면에서 null space가 왜 필요한가

A   x   b
[1 5   3   [=[
2   6   -1]   ]   ]
   2x3   3x1   2x1
3차원이 whole space가 되는 것은 rowrized 이다.
row vector 2개가 있는게 그게 나타낼 수 있는 whole space는 3차원이다.
이게 spanning 해낼 수 있는 총 공간 row space는 2차원 (같은 라인 선상에 있지 않다.) 서로서로 independent하다. 3차원은 절대 될 수 없다. 자생적으로 이 공간에서는 3차원으로 갈 수가 없다. 갖고 있는 것이 두 점이기 때문이다.   null space는 1차원(2차원과 수직인 선 하나)
null space를 정의할 때 Ax=0 라고 정의한다. x가 뭐가 오던지, given A에 대해서 0이 되는 x를 찾아라.

Ax=[0
0] 한 직선상에 해당하는 모든 점들이 x이다.
어떤 입력이 들어오든 출력에 영향을 미치지 않는,, 그 효과가 0이다. 일상생활과 인공지능에서 중요한 의미가 있다. skilled action이 되면 필요없는, 부가적인 것들을 많이 한다. 어떤 입력이 들어갔을 때 출력에 별로 영향 안 미치는 것이 null space이다. 실제로 그 task를 위해서는 별로 중요하지는 않지만, 점점점 더 그 공간을 확대시켜서 output에는 별로 영향 안 미친다.

A x b
[1 5 3 [=[
2 6 -1]]]
2x3 3x1 2x1
A(기계이고 학습이 가능하다,) 인 기계만 있으면 null space와 그렇지 않은 공간으로 쪼갤 수 있다.
x인 입력이 조금 바뀌더라도 영향안주는 것과 바뀌는 것
1. null space의 수학적 해석 Ax=[0
0]
2. 기하적 해석
3. 실용적, 응용적 해석

어떤 이미지나 그림이 들어가면 뭐다 이렇게 인식을 한다.
강아지 사진을 x에 주면 나온다. 입력값을 다른 강아지로 해도 출력에 영향을 미치지 않는다. x가 null space를 따라서 움직인다.

선형대수와 인공지능과의 관계를 null space차원에서 말하는 것

eigenvectors
given A가 있을 때
v는 입력값
A=[2.14 0.49
1.92 2.35]
[1 0
0 1] -은 아무런 영향도 안마친다. 입력이 어떤게 들어가도 그대로 있다.
정사각형 꼭젓점 있던 것은 평행사변형 꼭짓점으로 이동한다.
transformation matrix와 기하적인 visualization을 눈으로 확인한 것이다

eigenvector
원점과 입출력이 평행한 현상이 생긴다.

벡터라는 것을 정의할 때 matrix의 한 형태이다. 숫자열이다. 사실은 조금 더 물리학적 개념으로 벡터는 방향이다. 아까도 null space를 말할 때 방향이다. (null space의 방향으로 평행하게 , 방향이 중요한 것이지 그 선을 반드시 따라갈 필요는 없다는 것)
eigenvector - 2by2 행렬에 eigenvector는 무엇인가? given 해열이 있을 때 eigenvector는 뭐냐?? 어떤 값만 딱 존재한다. x 엄밀히 말하면 eigenvector는 여러 가능한 eigenvector의 집합이니까 eigenvector space라고 부른다.
vector는 한 값이라기 보다는 방향이 더 중요하다.

eigen vector의 방향은 두 개다. two by two에서 egienvector는두개가 있다. 3 by 3는 세개가 있다.
eigen vector는 two by two matrix에서는 2개가 있다. 그런데 각각의 eigenvector에 대해서 eigen value가 있다. 얼마큼 확대를 시키는 느낌? 확대시키는 비율 - eigenvalue는 2.35배로 확장. 1.49정도만 확장시키는 것.
null space가 왜 필요하냐? 1. 우리가 살면서도 늘 쓴다. 2. 더 skilled하게 보이는게 목적이 아니라, 진화를 해오면서 null space를 확보해왔다. 늘, 사냥도 하고 위험도 받고 있다. 어떤 task를 할 때, ax=b 라고 할 때, 출력의 부분이 우리가 하련ㄴ 일이다. 아무런 장애가 없으면,,, 장애가 존재할 수 있다. 피해서 돌아가야한다. task를 이루는 데에는 지장이 없는... 왜 null space의 확장시켜야 하는 이유이다. b라는 task를 하는데 방해받을 수 있는 방해요인을 피해갈 수 있는 것이 null space이다. 3. machine learning에서 어떤 이미지를 인식한다. 인식을 하는데, 인식되는 결과가 그림이 많이 변해도(입력이 많이 변해도) -- null space를 많이 이용하고 있구나.
하지만 eigen은 왜 배우는 것인가?
선형대수를 하면 eigen이 늘 필수적으로 나온다. 왜 배우는 건가??? 어떤 걸로 eigen이 나오는지?
하나의 matrix가 있다. A1이라는 column vector A2라는 column vector 두 개의 벡터를 또 다른 두 개의 벡터로 바꾼 것이다. 2by2 matrix - 또 다른 두 개의 eigenvector로 갖고 있게 되는 것이다. 왜 하냐? 우리말로 고유벡터라고 한다. 훨씬 더 고유한게 뭐냐? unique하다는 의미. 어떤 것을 분석할 때 딱 쪼개고 싶은 경우가 많다. 서로서로가 영향을 주지 않는 correlation 하지 않게 고유의 능력으로 쪼개고 싶다. 그냥 주어진 matrix는 그냥 섞여있는,,, eigen anaylsis를 하면 더 고유한 것으로 바꾸어준다. 통계에서는 Principle component analysis-- PCA 도 이 방법을 쓴다.

국어 점수 올라가면 영어점수 올라가고 같이 가는 느낌이 상관관계라고 한다. (correlation) 매우 중요한 개념이다. machine learning에서도 늘 나오는 이야기이다.

상관관계를 r이라는 값으로 표현할 수 있다. -1보다 크거나 같고 1보다 작거나 같다. 상관관계가 0 일 때 상관관계가 제일 낮다. -1은 음으로 높은 것이다. 그래프 상으로는
음의 상관관계를 보이는 예 x가 커피 y는 수면시간이 들어갈 수 있다. 음의 상관관계든 양의 산과관계는 -1, 1이 정확하게 나올 수는 없으나, -1,1 이라고 정확하게 나오는 순간은 언제인가. 그래프가 더 뾰족해지면 된다 완전한 선 상에 있으면 -1 또는 1이면 된다. 기울기가 어떻든 간에 선 상이면 된다. 얼마나 더 선 상에 가까운가가 절대값이라고 생각하면 된다.

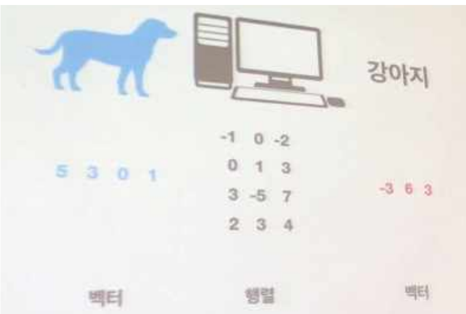
완전 원, 동그랗게 됬으면 어디로 갔는지 알 수 없으므로 r=0이라고 할 수 있다. 근데 완벽히 동그란경우는 잘 없으므로 r=0에 가까운 경우가 있을 것이다.

85차원에서 한 차원은 첫 번 째 학생, 그다음은 두 번째 학생 등등
국어 vector하나만들고, 영어 vector하나 만들고 .... 다른 차원을 지우면 삼각형이 된다. 그냥 점 세 개다. 3차원 상에서 두 점이 있다고 생각하면 된다. 3차원이라 할지라도 원점이랑 하면 삼각형이 된다. 그럼 간단한 문제가 된다.

각도 값 어떻게 재냐??? inner product(dot) - spectrogram을 직접 만들기 위해서
a가 (1,2,3), b가 (4,5,6) 일 때, 기하학적 계산법은 lal x cos쎄타 x lbl 이다.
inner product [1,2,3]
x x x
[4,5,6]
= 32

inner product 하면, 서로 유사하면, inner product가 높은 값 - 이 원리 이용하여 spectrogram만든다.

11.28 목
인공지능에 핵심에 해당하는 게 수학이다.
보이는 수학, 이야기 수학, 필요한 수학 – 기하학적으로 아는 것이 중요(이야기에 의해 습득하는 수학)
안녕하세요라고 말하면 기계 통해서 안녕하세요라는 글이 나오는 인공지능 등등
영상 소리 텍스트라는 데이터를 받아서 또 다른 형태의 데이터로.
데이터는 숫자열이다.
(영상데이터 - 픽셀은 다 숫자로 이루어져 있다. 하나씩 전체 그림에서 숫자를 따오면,
소리데이터 - 말하면, 소리 음파가 나오는데, 이것을 확대해서 끊어서, 숫자로 표현할 수 있다.
텍스트 데이터 - 1번 단어는 10000000000000000.... 2번째 단어는 010000000000....... 총 500000개)



eigen analysis
어떤 A라는 행렬이 있을 때,
A[1 2
5 6] x
v나 x나 같은데,
given A에 어떤 벡터를 곱했을 때,(given matrix에 어떤 벡터를 곱했을 때), Av는 transformation의 결과이다. 이것은 transformation되지 않은 v의 상수배이다.
Av=람다v (람다는 상수이고, Av는 transformation된 것이 원래 v의 상수배이다.) - 결국 한 라인 상에 있다. -- 이 식이 eigen analysis에 대한 식이다.
b가 eigen vector라고 할 때 ratio - 얼만큼 확장이 되는가? eigen value가 람다이다. v는 eigen vector이다.

inner product
1. 기하로 이해
삼각형이 만들어지면,
a b a=[1,2,3] 1x3
[1 [2 b=[2,4,7] 1x3
2 x 4 axb
3] 7] 1x3 1x3
a x b b를 3x1로 바꾸면, a x bt가 된다.

b=[2,4,7] 1x3
axb
1x3 1x3
b를 3x1로 바꾸면, a x bt가 된다.

1. a x bt
3 x 1
[1 2 3] [2 = 31
4
7]
2. at x b
3x1 1x3
[1 [2 4 7] = 3 x 3
2
3]

= 1번을 inner product , 2번을 outer product라고 부른다.
inner product는 a x bt 인데, a.b가 무조건 1x1 가 나와야 한다.

cos 쎄타 각도를 어떻게 측정하는가?
a.b/ lal lbl = cos쎄타 즉, a=[1,2,3] b=2,4,7] 이것만 있으면 다 구할 수 있다.
cos90=0
project 시킨 게 0 이면, inner product가 0 이 될 수 밖에 없다.

cosine similarity - a와 b가 얼마나 비슷한가? 두 벡터가 얼마나 비슷한지 말해주는.

inner product
phasor
100Hz   =10 (성분이 거의 없다.)
200Hz   - 1000 (크게 있다. 즉, 스펙트로그램에서 진하다)
.
.
.
5000Hz

inner product하면, 똑같은게 들어있으면 높은 innerproduct 값 나오고, 다르면 낮게 나온다.
(a가 올라갈 때 b도 올라가고, 완전 correlate - a.b는 상당히 크다.
a가 올라갈 때 c는 조금 떨어짐. 불일치 존재 - a.c는 a.b보다 조금 작다.)

phasor 차원에서의 각도와 벡터 차원에서의 각도가 같다.
inner product는 wave내에서 어떤 주파수가 많은가??
그러나, 같음을 이용한 idea는 좋았으나, phase shift하면 일치하지 않는다는 점에서 phase shift에 대한 민감도가 안 좋음. 그래서 sine cosine phasor말고 complex phasor를 왜 쓸까? sine cosine은 phase shift에 대한 민감도가 안좋음. wave는 real에서 존재하는 데, 근데 inner product – 여러 개 wave를 만들어서 이걸 inner product 할 것을 complex로 -complex value도 한 값이니까 9개를 만들어서 inner product로 그러면 phase에 대한 민감도 해결 가능