Chapter 1

딥러닝을 위한 필수 기초 수학

혁펜하임의 AI DEEP DIVE

함수

- 한 개 입력 (x) -> 한 개 출력 $(y = f(x) = x^2)$
  - 그래프 그려보기
- 두 개 입력 (x, y) -> 한 개 출력 $(z = f(x, y) = yx^2)$
  - 그래프 그려보기
- 한 개 입력 (x) -> 두 개 출력 (벡터 한 개 출력) $(\mathbf{y} = \mathbf{f}(x) = \begin{bmatrix} x \\ x^2 \end{bmatrix})$
  - 그래프..?
- 두 개 입력 (x, y) -> 두 개 출력 $(\mathbf{z} = \mathbf{f}(x, y) = \begin{bmatrix} xy^2 \\ x + y \end{bmatrix})$

로그 함수

- **$\log_{□}△$ => □ (밑)를 몇 승 해야 △ (진수) 냐? (그럼 $\log_2 4$ 는?)**
- **i)** $\log_{10} 100$ **ii)** $\log_e e^3$
- $\log_{10} x$ **vs** $\log_e x$ **그래프 그리기**
- **중요한 성질들**

1. $\log_a xy = \log_a x + \log_a y$

2. $\log_a x^n = n \log_a x$

3. $\log_{a^m} x = \frac{1}{m} \log_a x$

4. $\log_a b = \frac{\log_c b}{\log_c a}$

5. $\log_a b = \frac{1}{\log_b a}$

6. $a^{\log_a x} = x$

7. $a^{\log_b c} = c^{\log_b a}$

참고: 앞으로 그냥 $\log$ 라고 써있으면 밑이 e 인 로그 입니다

벡터와 행렬

- 열 벡터와 행 벡터 그리고 행렬 (왜 배울까?)
- 연립 일차 방정식을 행렬과 벡터로 나타내보기 $\begin{cases} x + 2y = 4 \\ 2x + 5y = 9 \end{cases}$
- 벡터와 행렬에 대해서 공부하는 학문! 그것이 바로 선형대수학

벡터와 행렬

- 행렬과 벡터의 곱 (원래 식으로 돌아갈 수 있어야 한다)
- 행렬과 행렬의 곱 $\begin{cases} x_1 + 2y_1 = 4 \\ 2x_1 + 5y_1 = 9 \end{cases}$ & $\begin{cases} x_2 + 2y_2 = 3 \\ 2x_2 + 5y_2 = 7 \end{cases}$ 를 하나의 식으로!
- 2x2 (라고 쓰고 투 바이 투 라고 읽음) 행렬과 2x3 행렬을 곱할 수 있을까요?
- 2x2 행렬과 3x2 행렬은 왜 곱할 수 없을까요?
- 주요 성질: 1) 짝짜꿍 2) 결과 사이즈는 맨 앞 x 맨 뒤 3) 교환법칙 X
- 연습! 4x3 행렬 곱하기 3x5 행렬 곱하기 5x2 결과 사이즈는?

벡터와 행렬

- 벡터의 방향과 크기: $[2,1]$ 이 행 벡터를 좌표평면에 나타내보자
  - 원점을 시점, $[2,1]$ 좌표를 종점으로 놓으면 벡터를 화살표로 나타낼 수 있다!
- 크기와 방향이 같으면 같은 벡터라서 시점이 달라도 같은 벡터 일 수 있다
- 화살표의 길이로 벡터의 크기를 나타낼 수 있다 $\sqrt{1^2 + 2^2}$ <= 피타고라스!
- l2-norm: $\sqrt{x^2 + y^2}$, l1-norm: $|x| + |y|$

전치와 내적

- **뒤에서 계속 쓰일 것: 전치 (Transpose)**
- $\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ **면** $\mathbf{A}^{\mathrm{T}} = \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{bmatrix}$ **로 자리 바꾸기!**
- $[\mathbf{A}^{\mathrm{T}}]_{ij} = [\mathbf{A}]_{ji}$ **:** i **행** j **열에** j **행** i **열의 성분을 놓는다**
- **연습:** $\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$ **양변을 transpose 하면?**

전치와 내적

- **내적의 정의:** $\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \cdot \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = a_1 b_1 + a_2 b_2 = \mathbf{a}^{\mathrm{T}}\mathbf{b}$
- **내적은 닮은 정도를 나타낸다!** $\mathbf{a}^{\mathrm{T}}\mathbf{b} = \|\mathbf{a}\|\|\mathbf{b}\|\cos\theta$ **(제 2 cos 법칙으로 증명 가능)**
  - $\|.\|$는 2-norm을 나타냄
  - θ 는 두 벡터의 사잇각을 나타냄
  - $\mathbf{b}$ 의 크기를 1로 고정해놓고 방향을 돌려가며 내적 결과를 생각해보면.. 가장 닮았을 때 최대다!!
  - 수직할 때가 가장 안닮은 것

극한

- $\lim_{x \to a} f(x)$: x 가 a 와 무진장 가까운 값일 때, $f(x)$ 는 뭐랑 무진장 가깝냐? (극한값)
- 다가간다든가, 움직이는 것 아님!
- $\lim_{x \to a} f(x) = L$ 을 만족하는 것을 그래프로 봅시다 (극한 값이 없는 상황은 무엇일까요?)
- $\lim_{x \to a} f(x) = L$: L 주변 갭으로 어떤 양수 ε 을 잡더라도 요 갭 안으로 싹다 보내버릴 수 있는 a 주변 갭 δ 가 존재하면 a 에서의 극한 값은 L 이다!
  - 이것이 바로 $\varepsilon - \delta$ 논법!

미분

- 순간 변화율이다! (그래프에선 순간 기울기)
- $y = 2x$ 는 기울기 2 (1 만큼 변하면 2만큼 올라간다)
  $y = -2x$ 는 기울기 -2 (1 만큼 변하면 -2만큼 내려간다)
- 순간 기울기? (0.1 만큼 변하면..)
- 왜 필요할까? $y = x^2$ 는 기울기가 계속 변한다!
- 순간이란..? -> 극한이 필요!
- $x = 1$ 에서의 순간 기울기 (미분 값) = $\lim_{\Delta x \to 0} \frac{f(1 + \Delta x) - f(1)}{\Delta x}$
- 그럼 2 에서의 미분 값? 3 에서의 미분 값? -1 에서의 미분 값?

도함수

- 그냥 x 에 대해서 미분 값을 구하자! 그것이 도함수 = $\lim_{\Delta x \to 0} \frac{f(x+\Delta x) - f(x)}{\Delta x}$
- $f'(x)$ 와 같이 표기하기도 하고 $\frac{dy}{dx}$ 로 표기하기도 한다. (하지만, 분수는 아님!)
- 특정 위치 $x = 1$ 에서의 순간 기울기 (미분 값)은 $f'(1)$ 혹은 $\left.\frac{dy}{dx}\right|_{x=1}$ 로 표기
- $y = x^2$ 의 도함수를 구해봅시다.

도함수

- **딥러닝 공부하면서 자주 보이는 도함수들!**

1. $x^n \rightarrow nx^{n-1}$

2. $e^x \rightarrow e^x$

3. $\ln x \rightarrow \frac{1}{x}$ **&** $\log_2 x \rightarrow \frac{1}{\ln 2}\frac{1}{x}$

4. $f(x) + g(x) \rightarrow f'(x) + g'(x)$

5. $af(x) \rightarrow af'(x)$

6. $f(x)g(x) \rightarrow f'(x)g(x) + f(x)g'(x)$

연쇄법칙

- $(x^2+1)^2$ 를 미분해봅시다.
- $x \to x^2 \to x^2+1 \to (x^2+1)^2$
- $$\frac{d(x^2+1)^2}{dx} = \frac{d(x^2+1)^2}{d(x^2+1)}\frac{d(x^2+1)}{dx^2}\frac{dx^2}{dx}$$ 로 쪼개기 가능!
- 과정을 뒤로 뒤로 미분하고 곱하는 것으로 생각 가능
- $x \to y \to z$, $y = f(x)$ & $z = g(y) = g(f(x))$

$$\frac{dz}{dx} = \lim_{\Delta x \to 0} \frac{g(f(x+\Delta x)) - g(f(x))}{\Delta x} = \lim_{\Delta x \to 0} \frac{g(f(x+\Delta x)) - g(f(x))}{f(x+\Delta x) - f(x)} \frac{f(x+\Delta x) - f(x)}{\Delta x} = \frac{dz}{dy}\frac{dy}{dx}$$

편미분과 그라디언트

- $f(x, y) = yx^2$ 와 같이 여러개 변수로 이루어진 함수를 미분할 때
  각각에 대해서 미분하는 것!
- x에 대한 변화율만 보면 x에 대한 편미분,
  y에 대한 변화율만 보면 y에 대한 편미분
- 기호는 $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}$ 이고 다른 변수들은 전부 상수로 취급하고 미분하면 된다!
- $f(x, y) = yx^2$ 에 대해서 $\frac{\partial f}{\partial x} = 2yx$, $\frac{\partial f}{\partial y} = x^2$
- 정의는 $\frac{\partial f}{\partial x} = \lim_{\Delta x \to 0} \frac{f(x + \Delta x, y) - f(x, y)}{\Delta x}$ & $\frac{\partial f}{\partial y} = \lim_{\Delta y \to 0} \frac{f(x, y + \Delta y) - f(x, y)}{\Delta y}$

그라디언트

- **그라디언트는?** $\begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$ **그냥 벡터로 묶은 것!**
  - $f(x, y) = yx^2$ **의 그라디언트 =** $\begin{bmatrix} 2yx \\ x^2 \end{bmatrix}$
- **그라디언트 값은** $\left.\begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}\right|_{x=1,y=1} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$

(LEVEL 2)

테일러 급수

- 어떤 임의의 함수를 다항함수로 나타내는 것
  - **다항함수**란? 다항식으로 표현되는 함수
  - **다항식**이란? 변수와 상수의 곱과 합으로 이루어진 식
  - **항**? 곱으로 이루어진 것 $3x + xy$ 은 두 개의 항
- $\cos(x)$ 와 같이 다항식이 아닌 함수를 $x, x^2, x^3, x^4, \ldots$ 이런 것들을 잘 조합해서 나타내보자! (우선 이게 가능은 한건지 그래프로 확인해봅시다)
  - 위에서 "조합"한다는 것은 주루룩 더하는 것을 의미, 이를 급수라고 해요.
- 왜 필요한가? 다항함수는 전구간 미분 가능, 미분도 간단해서 다루기 쉬운 함수
- $\cos(x) = c_0 + c_1x + c_2x^2 + c_3x^3 + c_4x^4 + \ldots$ 에서 계수 c_n 만 알아내면 끝!
  - 어떻게 구할 수 있을까..? 같이 하나씩 구해봅시다.

(LEVEL 2)

테일러 급수

- 이쯤에서 들어야 하는 의문
  - 다른 조합은 없을까..? 독립!
  - $x = 0$ 에서만 잘 맞는다..!
- 사실 지금까지는 Maclaurin 급수 였고 진짜 테일러 급수는..
- $f(x) = c_0 + c_1 x + c_2 x^2 + c_3 x^3 + c_4 x^4 + \dots$
  - $x = 0$ 근처에서 잘한다 $c_n = \dfrac{f^n(0)}{n!}$
- $f(x) = c_0 + c_1 (x-a) + c_2 (x-a)^2 + c_3 (x-a)^3 + c_4 (x-a)^4 + \dots$
  - $x = a$ 근처에서 잘한다 $c_n = \dfrac{f^n(a)}{n!}$

(LEVEL 2)

테일러 급수

- $e^x = 1 + x + \frac{1}{2!}x^2 + \frac{1}{3!}x^3 + \frac{1}{4!}x^4 + \frac{1}{5!}x^5 + \dots$
- $\ln(x) = c_0 + c_1(x-1) + c_2(x-1)^2 + c_3(x-1)^3 + c_4(x-1)^4$
- **얘는 이상하게 $x > 2$ 범위에서 수렴을 못한다..!**
- $(\ln(x))' = \frac{1}{x}$, $(\ln(x))'' = -\frac{1}{x^2}$, $(\ln(x))''' = \frac{2}{x^3}$, $(\ln(x))'''' = -\frac{6}{x^4}$

  $c_0 = 0,\ c_1 = 1,\ c_2 = -\frac{1}{2},\ c_3 = \frac{1}{3},\ c_0 = -\frac{1}{4}$
- $x = 3$ **대입해보니.. 2 -> 0 -> 8/3 -> -4/3 -> … 발산한다..!**
- $\lim_{n\to\infty} \left| \frac{p_{n+1}}{p_n} \right| < 1$ **만족 하는 x 영역에서만 수렴!** (LEVEL 3)
  - **radius of convergence 는 1 이다!**

(LEVEL 2)

스칼라를 벡터로 미분

- 스칼라는 숫자 하나, 벡터는 숫자 여러 개! $f(x_1, x_2) = x_1 x_2^2$ 을 $\mathbf{x} = [x_1 \quad x_2]$로 미분
  (벡터 입력 스칼라 출력) (여기부터는 행 벡터를 x로 표기 하겠습니다)
- 이미 했던 것! 그라디언트다 -> $\begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} \end{bmatrix}$
- 그냥 편미분하면 되지만.. $y = f(x_1, x_2) = f([x_1 \quad x_2]) = f(\mathbf{x}) = \mathbf{x}\mathbf{x}^{\mathrm{T}}$
  미분하라고 하면..? 다 풀어제껴야 함..
  (지금이야 변수 달랑 두개라서 쉽지만 x 의 길이가 100이고 더 복잡한 수식이라면..?)
- 순간 변화량을 한번 생각해보면..

$$
\begin{aligned}
df &= \lim_{\Delta x_1 \to 0, \Delta x_2 \to 0} f(x_1 + \Delta x_1, x_2 + \Delta x_2) - f(x_1, x_2) \\
&= \lim_{\Delta x_1 \to 0, \Delta x_2 \to 0} f(x_1 + \Delta x_1, x_2 + \Delta x_2) - f(x_1, x_2) + f(x_1, x_2 + \Delta x_2) - f(x_1, x_2 + \Delta x_2) \\
&= \lim_{\Delta x_1 \to 0, \Delta x_2 \to 0} \frac{f(x_1 + \Delta x_1, x_2 + \Delta x_2) - f(x_1, x_2 + \Delta x_2)}{\Delta x_1} \Delta x_1 + \frac{f(x_1, x_2 + \Delta x_2) - f(x_1, x_2)}{\Delta x_2} \Delta x_2 \\
&= \frac{\partial f}{\partial x_1} dx_1 + \frac{\partial f}{\partial x_2} dx_2
\end{aligned}
$$

(LEVEL 2)

스칼라를 벡터로 미분

- $df = \frac{\partial f}{\partial x_1}dx_1 + \frac{\partial f}{\partial x_2}dx_2 = \begin{bmatrix} dx_1 & dx_2 \end{bmatrix} \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix} = d\mathbf{x}\frac{\partial f}{\partial \mathbf{x}^{\mathrm{T}}}$ 로 전개 가능!!
- 즉, df 를 구하고 $d\mathbf{x}$ 뒤에 곱해진 것이 바로 구하고 싶은 미분 결과!
- $f(\mathbf{x}) = \mathbf{x}\mathbf{x}^{\mathrm{T}}$ 에 대해 연습해보면..
- $df = f(\mathbf{x} + d\mathbf{x}) - f(\mathbf{x}) = (\mathbf{x} + d\mathbf{x})(\mathbf{x} + d\mathbf{x})^{\mathrm{T}} - \mathbf{x}\mathbf{x}^{\mathrm{T}} = 2d\mathbf{x}\mathbf{x}^{\mathrm{T}} + d\mathbf{x}d\mathbf{x}^{\mathrm{T}}$ 에서 $d\mathbf{x}d\mathbf{x}^{\mathrm{T}}$ 는 생략 가능
  (극한을 취했을 때 무조건 0으로 수렴되는 항)
- 따라서 $d\mathbf{x}$ 뒤에있는 $2\mathbf{x}^{\mathrm{T}}$ 가 바로 미분 결과인 $\frac{\partial f}{\partial \mathbf{x}^{\mathrm{T}}}$ 이다! (전개 후 편미분해서 쌓는 것으로도 확인)
- 알고 있는 미분 공식을 변화량으로 적용하면 그대로 적용 가능 ($\mathbf{x}\mathbf{x}^{\mathrm{T}}$에 곱하기의 미분 공식 적용)

(LEVEL 2)

왜 그라디언트는 가장 가파른 방향을 향하는가

- loss 함수 $L(\mathbf{w})$를 $\mathbf{w} = \mathbf{w}_k$ 에서 1차까지만 Taylor series 전개 하면

$$L(\mathbf{w}) \simeq L(\mathbf{w}_k) + (\mathbf{w} - \mathbf{w}_k)\frac{\partial L}{\partial \mathbf{w}^{\mathrm{T}}}\bigg|_{\mathbf{w}=\mathbf{w}_k}$$

- 윗 식에 $\mathbf{w}_k$ 에서 조금 update한 $\mathbf{w}_{k+1} = \mathbf{w}_k + \Delta$ 를 대입한다면

$$L(\mathbf{w}_{k+1}) \simeq L(\mathbf{w}_k) + \Delta \frac{\partial L}{\partial \mathbf{w}^{\mathrm{T}}}\bigg|_{\mathbf{w}=\mathbf{w}_k}$$

- 따라서, $L(\mathbf{w}_{k+1}) - L(\mathbf{w}_k) \simeq \Delta \frac{\partial L}{\partial \mathbf{w}^{\mathrm{T}}}\bigg|_{\mathbf{w}=\mathbf{w}_k}$ 이고 우변은 Δ와 그라디언트의 내적이다!

- 방향만 보기 위해 Δ의 크기를 1로 고정한다면 $L(\mathbf{w}_{k+1}) - L(\mathbf{w}_k)$가 최대한 양수로 크려면 Δ의 방향은 그라디언트 방향과 일치해야 한다! 다시 말해, Δ만큼 업데이트 할 때, 그라디언트 방향으로 업데이트 하는 게 가장 L 을 키울 수 있는 방향이다.

- 따라서, 그라디언트는 해당 지점에서 항상 가장 가파른 방향을 향하는 벡터임을 알 수 있고 learning rate가 존재해야 하는 이유도 알 수 있다! why?

(LEVEL 2)

벡터를 벡터로 미분

- **스칼라를 벡터로 미분:** $df = \frac{\partial f}{\partial x_1} dx_1 + \frac{\partial f}{\partial x_2} dx_2 = \begin{bmatrix} dx_1 & dx_2 \end{bmatrix} \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix} = d\mathbf{x} \frac{\partial f}{\partial \mathbf{x}^{\mathrm{T}}}$

- **벡터 입력 벡터 출력?** $\mathbf{f}([x_1 \quad x_2]) = \begin{bmatrix} x_1 x_2^2 & x_1 + x_2 \end{bmatrix}$ **여러 개 들어가서 여러 개 나오는 것!**

- **벡터를 벡터로 미분:** $d\mathbf{f} = \begin{bmatrix} df_1 & df_2 \end{bmatrix} = \begin{bmatrix} dx_1 & dx_2 \end{bmatrix} \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_2}{\partial x_1} \\ \frac{\partial f_1}{\partial x_2} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} = d\mathbf{x} \frac{\partial \mathbf{f}}{\partial \mathbf{x}^{\mathrm{T}}}$

- $\mathbf{y} = \mathbf{f}(\mathbf{x}) = \mathbf{x}\mathbf{A}$ **를** $\mathbf{x}$ **로 미분 =>** $d\mathbf{f} = d\mathbf{x}\mathbf{A}$ **이므로** $\frac{\partial \mathbf{f}}{\partial \mathbf{x}^{\mathrm{T}}} = \mathbf{A}$ **로 바로 나온다!**

  (열심히 다 전개한 다음 하나하나 편미분해서 쌓은 것과 비교해 봅시다)

(LEVEL 2)

벡터를 벡터로 미분 – 연쇄법칙

- $\mathbf{y} = \mathbf{x}\mathbf{A}, \mathbf{z} = \mathbf{y}\mathbf{B}$ 일 때 $\mathbf{z}$ 를 $\mathbf{x}$ 로 미분? => chain rule!
- $\mathbf{x} \rightarrow \mathbf{y} \rightarrow \mathbf{z}$ 로 이어질 때 과정을 뒤로 뒤로 미분하고 곱하는 것으로 연쇄 법칙 똑같이 적용 가능!
- $\mathbf{z} = \mathbf{y}\mathbf{B} = \mathbf{x}\mathbf{A}\mathbf{B}$ 에서 $\frac{\partial \mathbf{z}}{\partial \mathbf{x}^{\mathrm{T}}} = \mathbf{A}\mathbf{B}$ 를 아는 상태에서 연쇄법칙을 통해 끌어내 보자
- 앞서, $\mathbf{x} \rightarrow \mathbf{y}$ 인 상황에서 $d\mathbf{y} = d\mathbf{x}\frac{\partial \mathbf{y}}{\partial \mathbf{x}^{\mathrm{T}}}$ 임을 알게 됐다.
- 즉, $\mathbf{y} \rightarrow \mathbf{z}$ 만 보면 $d\mathbf{z} = d\mathbf{y}\frac{\partial \mathbf{z}}{\partial \mathbf{y}^{\mathrm{T}}}$ 임을 알 수 있고, 자연스럽게 $d\mathbf{z} = d\mathbf{x}\frac{\partial \mathbf{y}}{\partial \mathbf{x}^{\mathrm{T}}}\frac{\partial \mathbf{z}}{\partial \mathbf{y}^{\mathrm{T}}}$ 가 됨을 알 수 있다!
  - 구하고 보니, 여기서는 뒤로 뒤로는 아니고 과정 순서대로 앞으로 앞으로 미분하고 곱하는 것으로 생각 가능

(LEVEL 2)

스칼라를 행렬로 미분

- 행렬이 입력, 스칼라가 출력되는 함수에 적용 가능 (ex) $f(\mathbf{X}) = \mathrm{tr}(\mathbf{XA})$)
- 앞서 나온 방법과 똑같이, $df = \frac{\partial f}{\partial x_{11}} dx_{11} + \frac{\partial f}{\partial x_{12}} dx_{12} + \frac{\partial f}{\partial x_{21}} dx_{21} + \frac{\partial f}{\partial x_{22}} dx_{22}$
- 이를 $d\mathbf{X} = \begin{bmatrix} dx_{11} & dx_{12} \\ dx_{21} & dx_{22} \end{bmatrix}$와 $\frac{\partial f}{\partial \mathbf{X}^{\mathrm{T}}} = \begin{bmatrix} \frac{\partial f}{\partial x_{11}} & \frac{\partial f}{\partial x_{21}} \\ \frac{\partial f}{\partial x_{12}} & \frac{\partial f}{\partial x_{22}} \end{bmatrix}$ 로 잘 표현해야 할 것
- 결론적으로, $df = \mathrm{tr}\left(d\mathbf{X} \frac{\partial f}{\partial \mathbf{X}^{\mathrm{T}}} \right)$와 같이 표현됨을 알 수 있다! (전에 배운 것들과 비교해 봅시다)
- 예시로, $f(\mathbf{X}) = \mathrm{tr}(\mathbf{XA})$의 미분을 구해보면 $df = \mathrm{tr}(\mathbf{dXA})$ 이므로 $\frac{\partial f}{\partial \mathbf{X}^{\mathrm{T}}} = \mathbf{A}$ 이다!

(LEVEL 3)

행렬을 행렬로 미분

- 벡터를 벡터로 미분을 할 줄 알면 행렬을 행렬로도 미분가능하다! ($\mathbf{Y} = \mathbf{F}(\mathbf{X}) = \mathbf{XA}$)
- 하지만 앞선 증명 방식으로는 한계가 있다. 따라서, 행렬을 벡터로 바꾼 다음,
  벡터를 벡터로 미분을 그대로 적용하여 구한다. (vectorize)
- $\text{vec}\left(\begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix}\right) = [x_{11} \quad x_{12} \quad x_{21} \quad x_{22}]$, $\text{vec}\left(\begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{bmatrix}\right) = [y_{11} \quad y_{12} \quad y_{21} \quad y_{22}]$
- $d\text{vec}(\mathbf{F}) = d\text{vec}(\mathbf{X})\dfrac{\partial \text{vec}(\mathbf{F})}{\partial \text{vec}^{\mathrm{T}}(\mathbf{X})}$ (벡터를 벡터로 미분한 $d\mathbf{f} = d\mathbf{x}\dfrac{\partial \mathbf{f}}{\partial \mathbf{x}^{\mathrm{T}}}$ 와 비교해보세요)

(LEVEL 3)

벡터를 행렬로 미분

- **벡터를 행렬로 미분도 마찬가지 방식으로 접근 가능! 예를 들면, $\mathbf{y} = \mathbf{x}\mathbf{W}$ 를 $\mathbf{W}$로 미분한다면?**
- **위에서** $\mathbf{y} = [y_1 \quad y_2]$, $\mathbf{x} = [x_1 \quad x_2]$, $\mathbf{W} = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix}$
- **Let** $\mathbf{w} = \mathrm{vec}(\mathbf{W}) = [w_{11} \quad w_{12} \quad w_{21} \quad w_{22}]$
- $[y_1 \quad y_2] = [x_1 \quad x_2] \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} = [x_1 w_{11} + x_2 w_{21} \quad x_1 w_{12} + x_2 w_{22}]$
- $\dfrac{\partial \mathbf{y}}{\partial \mathbf{w}^{\mathrm{T}}} = \begin{bmatrix} \dfrac{\partial y_1}{\partial w_{11}} & \dfrac{\partial y_2}{\partial w_{11}} \\ \dfrac{\partial y_1}{\partial w_{12}} & \dfrac{\partial y_2}{\partial w_{12}} \\ \dfrac{\partial y_1}{\partial w_{21}} & \dfrac{\partial y_2}{\partial w_{21}} \\ \dfrac{\partial y_1}{\partial w_{22}} & \dfrac{\partial y_2}{\partial w_{22}} \end{bmatrix} = \begin{bmatrix} x_1 & 0 \\ 0 & x_1 \\ x_2 & 0 \\ 0 & x_2 \end{bmatrix} = \mathbf{x}^{\mathrm{T}} \otimes \mathbf{I}_2$ **(Kronecker product)**

랜덤 변수와 확률 분포

- 랜덤 변수는 함수다!
- 사건이 입력, 실수의 값을 출력해주는 함수 (앞면: 1 뒷면: 0, 사건을 숫자에 대응!)
- 랜덤 변수는 대문자로, 실수 값으로 변환 되고 난 값은 소문자로 표현
- 이 실수 값을 확률 값으로 바꿔 주는 것이 확률 함수 (앞면: 1 -> 1/2 뒷면: 0 -> 1/2)
- 확률 질량 함수와 확률 밀도 함수가 있어요 (동전의 면과 키 값)
- 이 확률 함수로 랜덤 변수의 확률 분포를 나타냅니다.

랜덤 변수와 확률 분포

- 확률 질량 함수 (PMF) 예) 1. 동전, 2. 주사위
  - 동전 던지기: $p_X(X=0)=\frac{1}{2},\ p_X(X=1)=\frac{1}{2}$

    주사위 던지기: $p_X(X=5)=\frac{1}{6}$ 이런 식으로 표현 (그냥 편하게 $p(x)$로 표기하기도..)
  - 각각이 양수이면서 0과 1 사이 값을 가진다.
  - 합이 1 이다.
- 확률 밀도 함수 (PDF) 예) 키
  - 양수는 맞지만 0과 1 사이는 아니다.
  - 적분이 1 이다.
- 질량은 밀도 곱하기 부피. 즉, 밀도 함수를 x 값의 범위를 줘서 (범위가 곧 부피 같은 것)
  정적분 해줌으로써 확률 질량을 구하는 것이다. (키가 160~170 사이일 확률)
  적분을 통해 확률을 구하기 때문에 딱 165 일 확률은 0 이다! (165~165 까지 적분하면?)

평균과 분산

- 확률 분포를 설명하는 두 가지 대푯값이다.
- 평균
  - 1, 2, 3 의 평균은 2 -> 이것은 mean (수학적 단어) 또는 average (일상적 단어)를 구한 것!
  - mean의 종류
    - 1. 산술 평균 (위에서 구한 것이 바로 산술 평균!) 2. 기하 평균 3. 조화 평균
  - 수업에서 다룰 평균의 정체는 기댓값 (Expectation)!
    - 주사위를 다섯번 던져서 나온 값들 2, 2, 1, 6, 4 에 대해 산술 평균을 구하면 3
    - 위의 시행을 무한번 하고 산술 평균을 구하면 기댓값과 같아짐
    - 기댓값의 정의 $\mathrm{E}[X] = \sum_i x_i p_i$ (주사위에 대해 해보면 3.5가 나와요)
    - 분포를 잘 대표하는 지 살펴봅시다.
    - 연속 랜덤 변수에 대해서는? $\mathrm{E}[X] = \int_{-\infty}^{\infty} xp(x)\,dx$, $\mathrm{E}[X] = \mu$ 라고 보통 표기해요.

평균과 분산

- 평균만 가지고는 분포를 설명하기에 부족하다! (100점 0점 2명 vs 50점 2명)
- 분산
  - 퍼진 정도 (어디로부터? 평균으로부터 얼마나 퍼져있는가)
  - 평균과의 차이 (편차라고 해요), 그런데 양수로 만들어줘야 => 제곱!
    - 절댓값은 왜 안쓸까? (-7, -1, 1, 7 vs -4, -4, 4, 4)
    - 그렇게 구한 그냥 싹 더해버리면 값이 많을수록 점점 커진다 (-4, -4, -4, 4, 4, 4 면?) => 평균 내자!
  - 따라서, 평균과의 차이의 제곱 (편차의 제곱)의 평균을 내주자!
- **discrete:** $\mathrm{V}[X] = \sum_i (x_i - \mu)^2 p_i$ **& continuous:** $\mathrm{V}[X] = \int_{-\infty}^{\infty} (x-\mu)^2 p(x)\,dx$
- $$\begin{aligned}\mathrm{V}[X] &= \mathrm{E}\left[(X-\mu)^2\right] \\ &= \mathrm{E}\left[X^2 - 2X\mu + \mu^2\right] \\ &= \mathrm{E}\left[X^2\right] - 2\mathrm{E}\left[X\right]\mu + \mu^2 \\ &= \mathrm{E}\left[X^2\right] - \mu^2\end{aligned}$$, 참고로 $\mathrm{E}\left[aX + bY + c\right] = a\mathrm{E}\left[X\right] + b\mathrm{E}\left[Y\right] + c$
- 표준편차(standard deviation) σ 는 분산의 양의 제곱근 ($\sqrt{V[X]}$)! (왜 만들었을까? => 단위를 맞춰주기 위함!)

균등 분포와 정규 분포

- **균등 분포 (Uniform distribution)**
  - **생김새: 평평하다! (주사위, continuous도 물론 가능)**
  - **식:** $p(x) = \begin{cases} \frac{1}{b-a}, & \text{for } a \le x \le b \\ 0 & \text{otherwise} \end{cases}$, $X \sim U(a, b)$ **: 랜덤 변수** X **가 균등 분포를 따른다.**
  - **평균:** $\frac{1}{2}(a+b)$
  - **분산:** $\frac{1}{12}(b-a)^2$
- **정규 분포 (Normal distribution or Gaussian distribution)**
  - **생김새: 종모양 (키)**
  - **식:** $p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$, $X \sim N(\mu, \sigma^2)$ **: 랜덤 변수** X **가 정규 분포를 따른다.**
  - **평균:** μ
  - **분산:** σ^2

(LEVEL 2)

최대 우도 추정: MLE

Maximum Likelihood Estimation

- 먼저, 조건부확률 vs likelihood 비교를 통해 likelihood가 뭔지 알아봅시다. (주머니 예시로!)
  - likelihood: 조건부 확률 값인건 맞는데 확률 분포는 아님!
  - 앞의 것은 고정하고, 뒤의 것의 함수로 봤기 때문에 적분이 1이 아니기 때문!
- MLE는 measurement z를 보고 그 속에 숨어있는 x (알아내고 싶은 것)를 찾고자 함 (| 는 마치 벽)
- 예시 문제: $z_1 = x + n_1$, $z_2 = x + n_2$, $n \sim N(0,\sigma^2)$ 에 대해서 likelihood는

  $p(z_1, z_2 \mid x) = p(z_1 \mid x)p(z_2 \mid x)$ (독립 시행 가정), $p(z_1 \mid x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(z_1 - x)^2}{2\sigma^2}}$

- x 가 대체 뭐였기래 measurement가 이렇게 나왔을까? 에 대한 답을 하고 싶은 것

  likelihood는 x 에 대해서 measurement가 z_1, z_2 로 값이 나올 그 확률 밀도 값이기 때문에

  likelihood가 최대가 되도록 하는 x 가 바로 내가 찾고자 하는 x 다! (확률론적 관점에서 추정한 것)

- 구해보면.. $\hat{x} = \arg\max_x p(z_1 \mid x)p(z_2 \mid x) = \frac{z_1 + z_2}{2}$ 즉, 들어온 measurement 들을 평균 내라!
- $-\log$ 를 취하면 MSE와 같다는 사실도 알 수 있다.
- $\mathbf{z} = A\mathbf{x} + \mathbf{n}, \mathbf{n} \sim N(\mathbf{0}, \mathbf{I})$ 에 대해서 MLE로 $\hat{x}$ 구하면 Least squares solution (최소자승법)과 일치한다.

  (여기만 잠깐 열벡터 표현으로.. 죄송)

(LEVEL 3)

MAP

- likelihood 뿐만 아니라 prior distribution 까지 고려한 posterior를 maximize 하자는 것! (MAP: Maximum A Posteriori)
- 우선 알아야할 것: Bayesian rule => $p(x|z) = \frac{p(x,z)}{p(z)} = \frac{p(z|x)p(x)}{p(z)}$
- 여기서 posterior distribution = $p(x|z)$ 를 말함.
  즉, $p(z|x)$ (measurement가 이렇게 나올 조건 x를 바꿔가며 확률 밀도 값을 보자) 말고
  $p(x|z)$ (measurement가 이렇게 주어져 있을 때의 x에 대한 확률 밀도 값을 보자) 를 사용하자는 것!
- $\hat{x} = \arg\max_x p(x|z) = \arg\max_x \frac{p(z|x)p(x)}{p(z)} = \arg\max_x p(z|x)p(x)$
- MLE: $\hat{x} = \arg\max_x p(z|x)$ 와 비교하면 $p(x)$가 추가된 것을 알 수 있고, $p(x)$ 를 안다는 것은 x 의 분포를 사전에 알고 있다는 의미 => 사전 정보를 제공해 주는 것이므로 prior distribution 이라고 함!
  (물론, 잘못된 사전 정보는 오히려 추정 성능에 악영향!)
- $x \sim N(0, \sigma_x^2)$를 가정하고 $z_1 = x + n_1$, $z_2 = x + n_2$ 예시 문제를 여기에 적용하면..
- 이것이 바로 l2-regularization 의 loss 함수다! (solution은 $\hat{x} = \frac{z_1 + z_2}{2 + \sigma^2/\sigma_x^2}$)

정보 이론 기초 (찍먹)

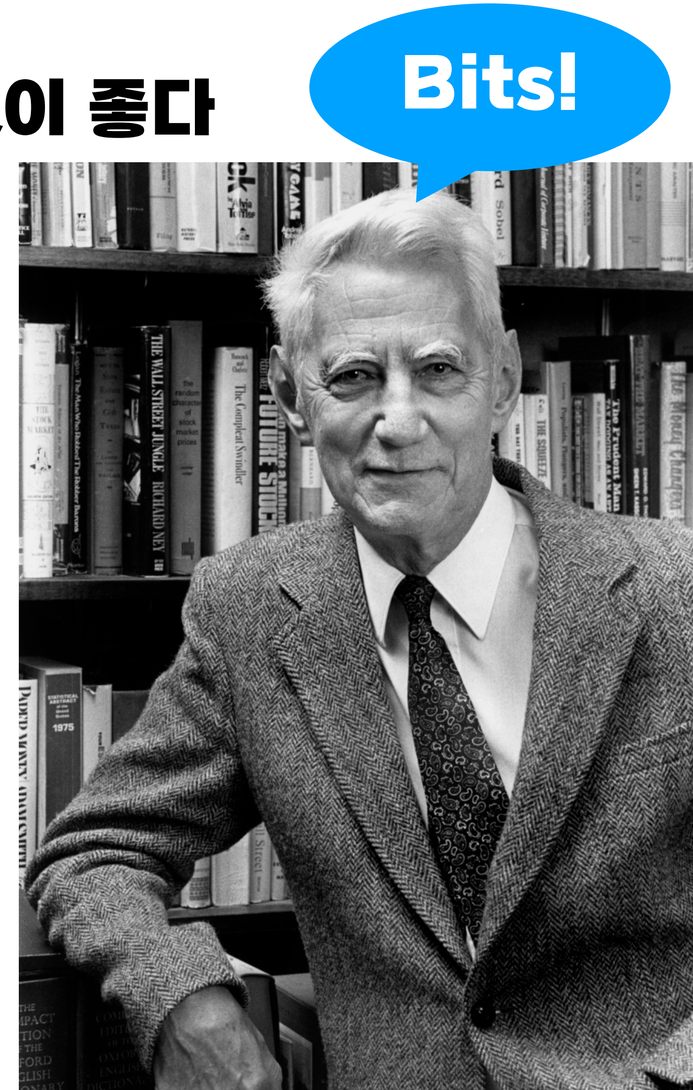
- ‘정보’를 하나의 언어(만국 공통어)로 표현하자

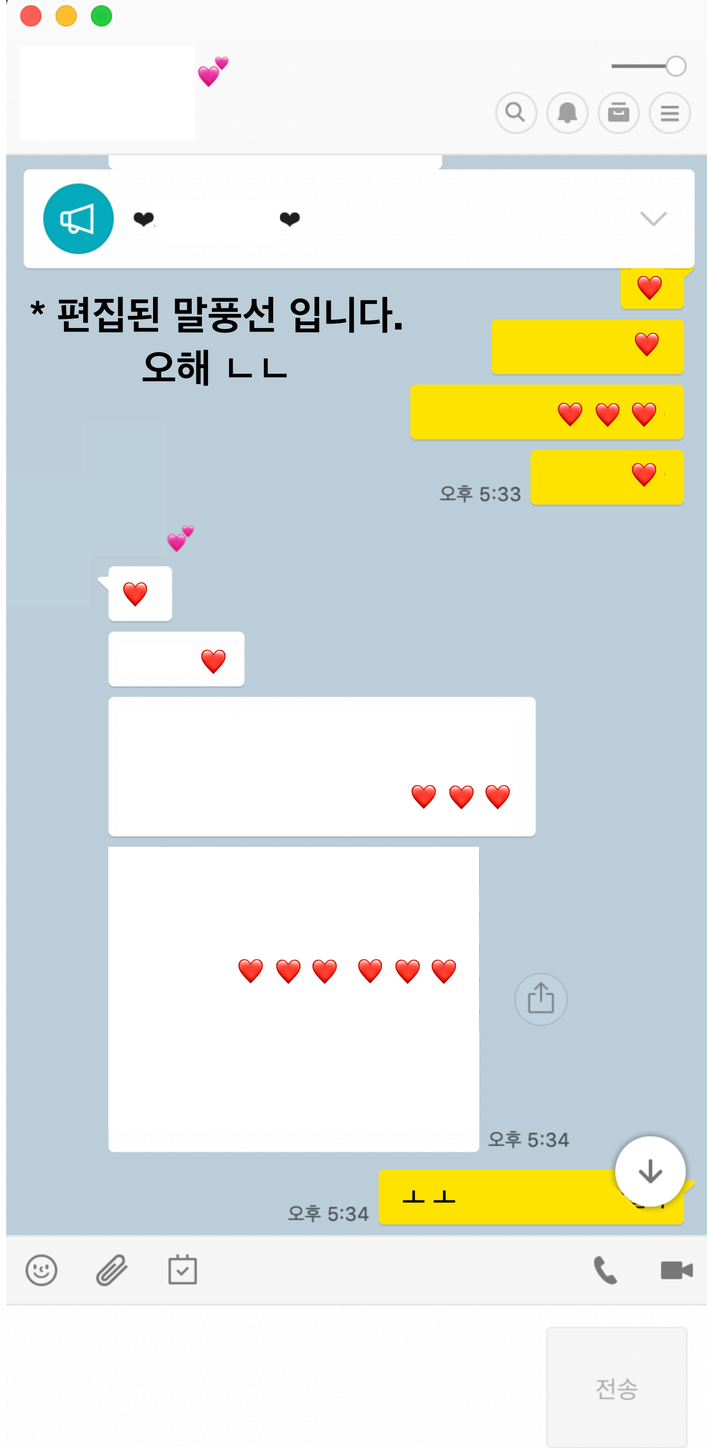
  => Bits! (이진수, 모스부호)

- 정보를 이진수로 나타낼 때 최대한 효율적으로 표현 하는 것이 좋다 (Source coding)
- 그런데, 정보란 것은 랜덤 하다..! (카톡)
- 그렇다면

  높은 확률로 나오는 글자는 짧게,

  낮은 확률로 나오는 글자는 길게 코딩하는 것이 좋을 것!





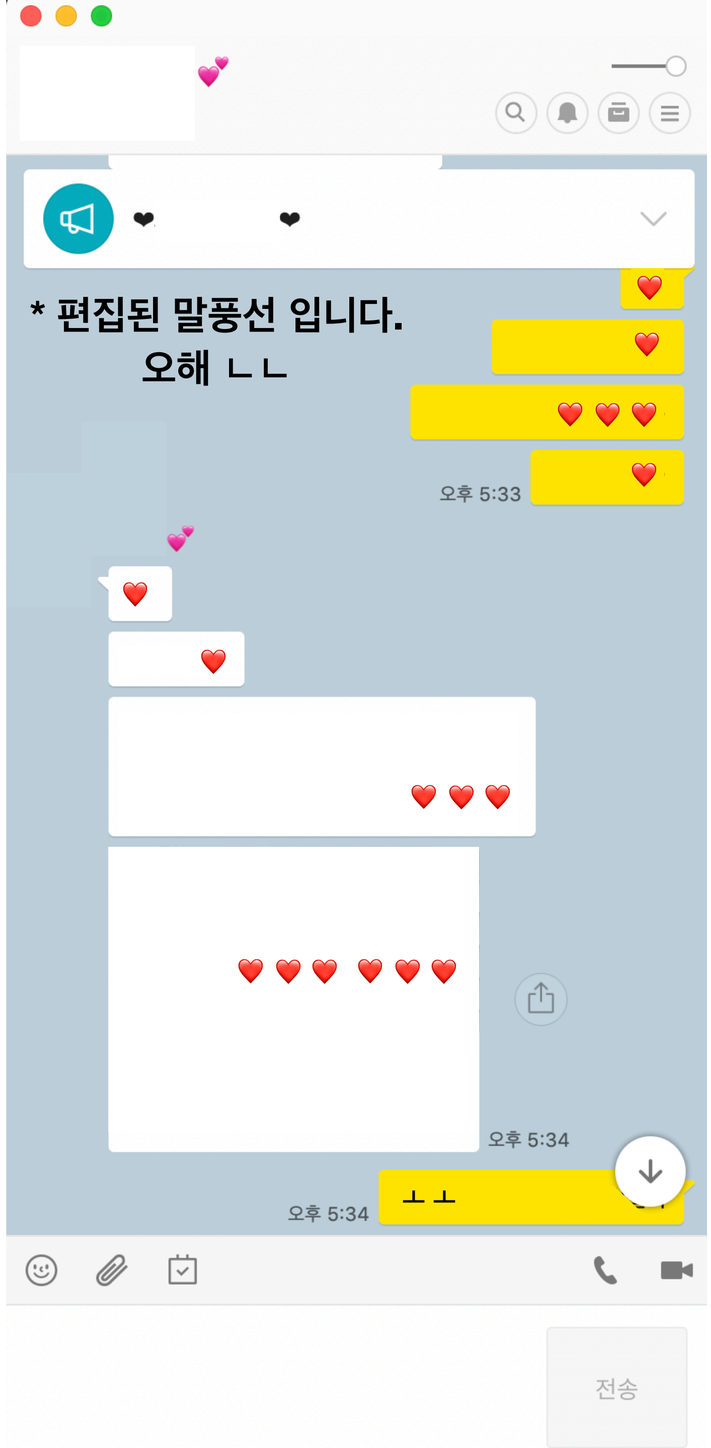
$$P(❤) = 0.5$$

$$\vdots$$

$$P(ㅗ) = 0.001$$

Entropy

- 하트를 111, 빠큐를 0 으로 코딩한다면? 반대로 한다면?
- 즉, 평균 길이 $\sum_i p_i l_i$ 를 최소화 하는 것이 좋을텐데.. 섀넌이 밝혔다!
  바로 Entropy가 이 평균 코드 길이의 최소임을!
- Entropy (불확실성)
  - 주사위 눈금 -> 굉장히 불확실함, 조작해서 3만 나오는 주사위를 만든다면? 엔트로피 0 (확실하게 3)
  - 식: $\sum_i -p_i \log_2 p_i$ 즉, i 번째 글자에 대해 코드 길이를 p_i 에 맞춰 $-\log_2 p_i$ 로 하면 된다는 뜻!
    (그래프로도 확인해 봅시다)
  - 즉, $\sum_i -p_i \log_2 p_i \leq \sum_i p_i l_i$ 와 같이 lower bound의 역할을 한다!
  - 동전 던지기의 entropy는 1, 조작된 동전이라 앞면만 나온다면? 0
  - 균등 분포일 때가 최악의 상황임을 알 수 있다!
    확률 분포가 좀 불균등해야 그나마 source coding을 통해 이득을 볼 수 있다는 의미



$$Q(❤) = 0.01$$

$$\vdots$$

$$Q(\perp) = 0.01$$

Cross-entropy & KL-divergence & Mutual information

- **Cross-entropy**
  - 실제로는 p_i 를 따르지만 q_i 에 맞춰 길이를 정한다
    (실제 p_i 를 몰라서도 그럴 수 있고 $-\log_2 p_i$ 가 정수가 아닐 수 있기 때문)
  - 식: $\sum_i -p_i \log_2 q_i$ <= (당연하게도) 항상 Entropy 보다 크다
    딥러닝에서는 이 q_i 가 신경망 출력으로, 최대한 p_i 와 비슷하게 만드려고 노력한다.
- **KL-divergence** (LEVEL 2) (KL 은 제안한 두 사람의 앞글자를 딴 것)
  - 식: $\sum_i -p_i \log_2 q_i - \sum_i -p_i \log_2 p_i = \sum_i -p_i \log_2 q_i + p_i \log_2 p_i = \sum_i -p_i \log_2 \frac{p_i}{q_i}$
  - 양수가 나올 것 (제대로 못 짠 평균 코드 길이 - 이론적으로 최대한 잘 짠 평균 코드 길이 이므로!)
  - p 와 q 의 분포 차이(거리)로 해석 가능! (만약 q_i 만 update 할 수 있다면 CE를 최소화 하는 해와 같다)
- **Mutual(상호간의) information** (LEVEL 2)
  - 식: $\sum_i \sum_j -p(x_i, y_j) \log_2 \frac{p(x_i, y_j)}{p(x_i)p(y_j)}$ => 독립이면 0, 즉, 독립적이지 않은 정도!