

## Statistic/Math

1. 확률 모형과 확률 변수는 무엇일까요?
2. 누적 분포 함수와 확률 밀도 함수는 무엇일까요? 수식과 함께 표현해주세요.

## Machine Learning

3. dimension reduction 기법으로 보통 어떤 것들이 있나요?
4. PCA는 차원 축소 기법이면서, 데이터 압축 기법이기도 하고, 노이즈 제거 기법이기도 합니다. 왜 그런지 설명해주실 수 있나요?

## Deep Learning

5. 알고있는 Activation Function에 대해 알려주세요. (Sigmoid, ReLU, LeakyReLU, Tanh 등)
6. 오버피팅일 경우 어떻게 대처해야 할까요?
7. 하이퍼 파라미터는 무엇인가요?
8. Weight Initialization 방법에 대해 말해주세요. 그리고 무엇을 많이 사용하나요?

## Python

9. What type of language is python? Programming or scripting?

## Algorithm

10. 다음 코드에서 print\_all\_prime\_numbers 함수의 파라미터 N에 대한 시간 복잡도는 무엇일까요?

```
double sqrt(double n){  
    /** n의 제곱근을 출력합니다 **/  
}
```

$O(1)$

```
bool is_prime_number(int n){  
    for(int i = 2; i <= sqrt(n); i++){  
        if(n % i == 0){  
            return false;  
        }  
    }  
    return true;  
}
```

$O(\sqrt{N})$

```
void print_all_prime_numbers(int N){  
    for(int i = 1; i <= N; i++){  
        if(is_prime_number(i)){  
            printf('%d is prime number\n', i);  
        }  
    }  
}
```

$O(N)$

$O(N\sqrt{N})$

1. 확률 모형: 데이터와 확률간의 관계를 이용하여 데이터분포를 묘사하는 방법

확률 변수: 표본 공간의 각 단위 사건의 결과값이 확률적으로 정해지는 함수

2. 누적 분포 함수: 임의의 변수가 특정 값보다 작거나 같을 확률을 나타내는 함수

$$F(a) = P(X \leq a) = \int_{-\infty}^a f(x) dx$$

확률 밀도 함수: 확률 변수의 분포를 나타내는 함수 (누적 분포 함수와 미적분 관계)

$$\int_{-\infty}^{\infty} f(x) dx = 1$$

3. 차원 축소 기법?

· 피처 추출 (Feature Extraction): 피처를 차원의 중요 피처로 압축하여 추출 - PCA, LDA, NMF

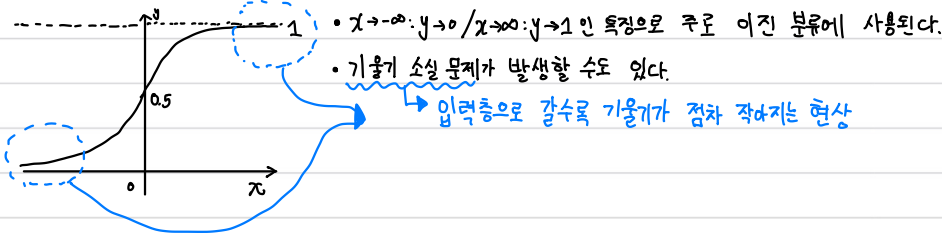
· 피처 선택 (Feature Selection): 모든 피처 중에서 필요한 것들만 선택 - Filtering, Wrapper, Embedded

4. PCA: 여러 변수 간 존재하는 상관관계를 이용해 이를 대표하는 주성분을 추출해 차원을 축소하는 기법

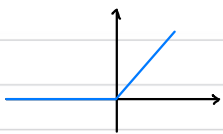
입력 데이터의 공분산 행렬을 기반으로 고유벡터를 생성하고 이렇게 구한 고유벡터에 입력 데이터를 선형 변환하여 차원을 축소한다.

차원은 곧 입력 데이터의 피처를 뜻하므로 데이터 압축 기법으로 볼 수 있다. 또한, PCA는 고유값이 가장 큰, 즉 데이터의 분산이 가장 큰 순으로 주성분 벡터를 추출하는데, 가장 나중에 뽑힌 벡터보다 가장 많이 뽑힌 벡터가 데이터를 더 잘 설명할 수 있기 때문에 노이즈 제거 기법이기도 하다.

$$5. \text{Sigmoid} = \frac{1}{1+e^{-x}}$$



$$\text{ReLU} = \max(0, x)$$

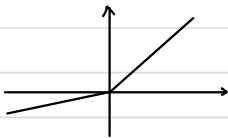


· Sigmoid, tanh보다 학습이 빠르고, 연산 비용이 적고, 구현이 매우 간단하다.

·  $x \leq 0$ 에서 0을 출력하여 가중치 업데이트가 안되는 현상이 발생한다.  $\Rightarrow$  Dying ReLU

· 0에서 미분불가능

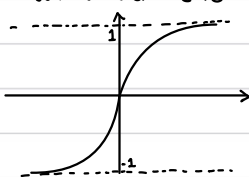
$$\text{Leaky ReLU} = \max(\alpha x, x)$$



·  $x < 0$ 에서 미미한 값 ( $\alpha=0.01$ )을 곱하여 Dying ReLU 방지

· But, 일부 사례에서 Sigmoid나 tanh에 비해 성능이 낮음

$$\tanh = \frac{\sinh x}{\cosh x} = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$



· Sigmoid의 단점 보완

· -1에서 1사이의 값을 출력하며, 중앙값으로 0을 가진다.

· Sigmoid에 비해 기울기 소실이 적은 편

## 6. 오버피팅에 대처하는 경우

- 규제 (Dropout,  $L_1$ ,  $L_2$  ...)
- 피쳐 수 ↓
- Data ↑
- Parameter 개수 ↓
- Data scaling

## 7. 하이퍼 파라미터: 모델이 학습할 수 없고 사람이 직접 지정해야 하는 파라미터

↳ 학습률, Batch size, loss function, activation function...

## 8. 가중치 초기화 방법

- Zero 초기화: 모든 가중치를 0으로 초기화 → 모든 가중치가 같다면 역전파 과정에서도 같아지므로 의미 없음
- Xavier 초기화: 이전 은닉층의 노드가  $n$ 개이고 현재 은닉층의 개수가  $m$ 개일 때,  $\frac{2}{n+m}$ 을 표준편차로 하는 정규분포로 초기화 (시그모이드가 쓰일 때 사용됨)
- He 초기화: ReLU 함수를 활성화 함수를 사용할 때 추천되는 방법으로 이전 은닉층의 개수가  $n$ 일 때,  $\sqrt{\frac{2}{n}}$ 을 표준편차로 하는 정규분포 초기화

## 9. Python은 프로그래밍 언어이자 스크립팅 언어이다.

↳ 특별한 컴파일 과정 X