

Statistic

1. 확률 모형과 확률 변수는 무엇일까요?

확률 분포를 보다 단순하게 묘사하기 위해 고안된 것이 **확률 모형 (probability model)** 이다.

확률 모형은 **분포 함수 (distribution function)** 또는 **밀도 함수 (density function)** 라고 불리는 미리 정해진 함수의 수식을 사용하여 분포의 모양을 정의 하는 방법이다. 이때, 분포의 모양을 결정하는 함수의 계수를 분포의 **모수 (parameter)** 라고 부른다.

확률 변수란?

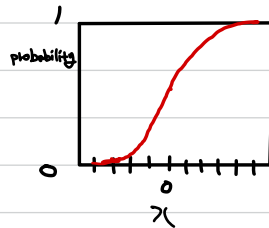
어떤 자료의 값이 분포가 특정한 확률 모형과 일치하는 경우 그 자료를 **확률 변수 (random variable)** 라고 하고 해당 확률 모형을 따른다고 말한다.

2. 누적분포 함수와 확률 밀도 함수는 무엇일까요? 수식과 함께 표현해주세요

누적 분포 함수 (cumulative distribution function) 란?

랜덤 변수 X 에 대하여 정의된 확률을 P_X 라고 한때 다음과 같이 정의되는 함수를 **누적 분포 함수 (cdf)** 라고 한다.

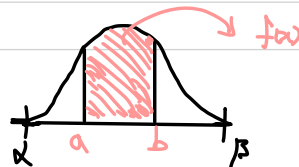
$$F_X(x) = P_X(X \leq x)$$



확률 밀도 함수란? (probability density function)

확률론에서 확률 밀도 함수는 확률 변수의 분포를 나타내는 함수로 확률 밀도 함수 $f(x)$ 와 구간 $[a, b]$ 에 대해서 확률 변수 X 가 구간에 포함될 확률 $P(a \leq X \leq b)$ 는

$$\int_a^b f(x) dx \text{ 가 된다.}$$



Deep Learning

! tensorflow와 pytorch의 특징과 차이?

구분	Tensorflow	PyTorch
패러다임	Define and Run	Define by Run
그래프 형태	Static graph(정적)	Dynamic graph(동적)
현재 사용자	많음	적음
자체 운영 포럼	없음	있음
한국 사용자 모임	Tensorflow Korea(TF-KR)	Pytorch Korea(Pytorch-KR)

Define and Run은 코드를 작성 한다는 환경인 세션으로 만들고 placeholder를 선언하고 이것으로 계산 그래프를 만들고 (Define), 코드를 실행하는 시점에 데이터를 넣어

실행하는 방식 (Run), 이는 계산그래프를 명확히 보여주면서 실행시점에 데이터만 바꿔줘도 되는 유연함은 갖지만, 그 자취로 비관적이다. 그래서 프레임워크층 난이도가 높은 편

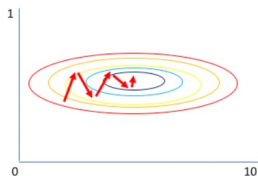
두 프레임 워크 모두 계산그래프를 정의하고 자동으로 gradient를 계산하는 기능이 있다. 하지만 TensorFlow의 계산그래프는 정적이고 pytorch는 동적이다.

즉, tensorflow에서는 계산그래프를 한 번 정의하고 나면 그래프에 들어가는 입력 데이터만 다르게 할 수 있을뿐, 같은 그래프만 실행할 수 있다. 하지만 pytorch는 각 순전파마다 새로운 계산그래프를 정의하여 이용한다.

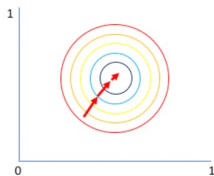
Pytorch 장점

1. 설치가 간편하다.
2. 이해와 디버깅이 쉬운 직관적이고 간결한 코드로 구성되었다.
3. Define by Run 방식을 기반으로 한 실시간 결과값을 시각화 한다.
4. 파이썬 라이브러리(Numpy, Scipy, Cython)와 높은 호환성을 가진다.
5. Winograd Convolution Algorithm 기본 적용을 통한 빠른 모델 훈련이 가능하다.
6. 모델 그래프를 만들 때 고정상태가 아니기 때문에 언제든지 데이터에 따라 조절이 가능하다(유연성).
7. Numpy스러운 Tensor연산이 GPU로도 가능하다.
8. 자동 미분 시스템을 이용해 쉽게 DDN(DataDirect Networks)을 짤 수 있다.
9. 학습 및 추론 속도가 빠르고 다루기 쉽다.

Data Normalization은 무엇이고 왜 필요한가?



Gradient of larger parameter dominates the update



Both parameters can be updated in equal proportions

왼쪽과 같이 Unnormalized 상태에서는 Learning rates를 매우 작게 해야 학습이 된다.

cost 그래프가 길쭉한 형태를 띄기 때문이다.

input의 range가 서로 다르면 Gradient Descent Algorithm은 작동하는 것이 매우 까다롭다.

하지만 Normalization을 적용하면 줄어 구의 형태를 띄게 되고 최적화 과정을 쉽고 빠르게 찾을 수 있다.

- • 정규화의 목적은 값 범위의 차이를 대폭 시키지 않고 data set을 공통 scale로 변경하는 것!

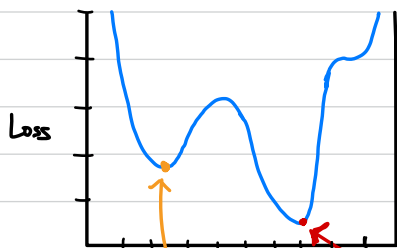
Machine Learning

1. Local minima와 Global Minima에 대해 설명해주세요!

Local minima

gradient descent를 사용하면서 마주치는 문제!

옆의 그림과 같이 학습을 진행할때 기저가 local minima에서 cost function이 최소값을 얻는 것으로 착각하여 학습을 끝내는 경우



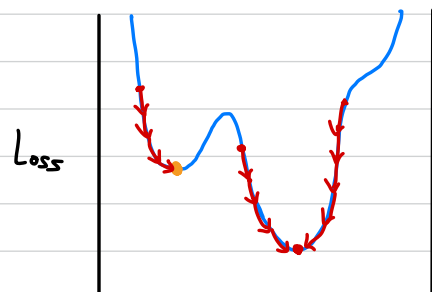
Global minimum

local minima

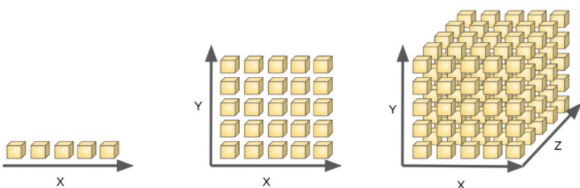
학습시 가중치 초기화를 반복하여 최적 해를 찾아가므로

local minima에 수렴하여 Loss 값이 0 가 가까이 떨어지지 못한다 한지라도 시작우리가 다른 가중치에서

Global minimum에 수렴하여 Loss 값이 0에 수렴할 수 있다.



2. 차원의 저주에 대해 설명해주세요!



1차원에 5개의 data 2차원의 25개의 data 3차원의 125개의 data

$$5^1 = 5$$

$$5^2 = 25$$

$$5^3 = 125$$

특정 문제를 설명하는 상황에서 차원이 귀찮아질수록 설명공간이 자주적으로 늘어남.

→ feature가 많아질수록 중요한 데이터를 설명하는 빈공간이 증가함.

→ 차원의 저주로 인해 알고리즘 모델링 과정에서 저장공간과 처리시간이 불필요하게 증가됨!

solution

PCA, LDA, LLE, MDS, t-SNE

Database

DBMS를 정의하세요!

DBMS - Database Management System

다수의 사용자가 데이터베이스 내에 데이터에게 접근, 사용할 수 있도록 해주는 소프트웨어를 말한다.

DBMS의 장점

1. 데이터 베이스 내 중복을 최소화할 수 있다.
(완전히 허용하지 않는 것은 아님)
2. 같은 내용의 데이터를 여러 가지 구조로 자원해
줄 수 있는 DBMS의 정교한 기법 덕분에
데이터베이스의 공유가 가능하다.
3. 데이터 베이스가 접근 처리 될때마다
제어 계층을 통해 그 유효성을 검사하기
때문에 데이터 무결성을 유지할 수 있다.
4. 철저한 보안 유지

DBMS의 단점

1. 컴퓨팅 시스템 자원의 소모량이 높음
2. 뚜렷한 목적을 가지는 응용 시스템은 설계 시간이
길어지고 보아 전문적인 인력이 필요
3. DBMS는 통합된 시스템이기 때문에 인력 강애는
임의키면 전체 시스템은 정지시켜 시스템 신뢰성과
가용성을 저해할 수 있다.

데이터 중복 최소화, 데이터 공유, 일관성 무결성 보일성유지
최신의 데이터유지, 데이터 표준화 가능, 데이터의 논리적 물리적
독립성
효율한 데이터 접근, 데이터 저장공간 절약

DBMS의 종류

ORACLE[®]
DATABASE

MySQL[®]

Microsoft[®]
SQL Server[®]

RDBMS를 정의하고 장점에 대해 설명해주세요!

RDBMS는 관계형 데이터 베이스를 생성하고 수정하고 관리하는 소프트웨어라고 정의할 수 있다.

RDBMS의 장점

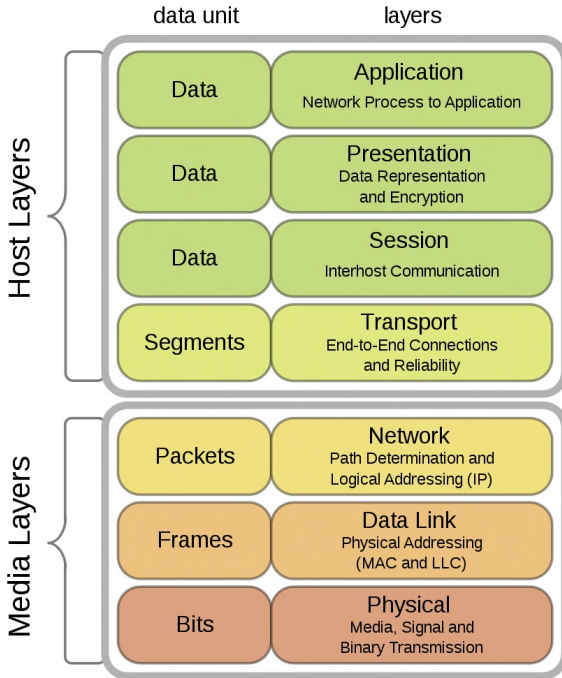
1. DB 수준에서의 ACID 트랜잭션을 사용한 쉬운 개발 방식 사용
2. view를 사용한 column과 row에 대한 세밀한 보안 설정은 안아받지 않은 사용자로 부터의 조회나 변경을 막음.
3. 대부분의 SQL 표준은 오픈소스로 포함한 다른 SQL 데이터 베이스로 포팅이 가능
4. 다양한 쿼리어나 제약등을 사용해서 데이터 베이스에 정보를 추가하기 전에 유효성 검사를 하여 데이터 품질을 향상함.

RDBMS의 단점

1. 객체-관계형 매핑 레이어가 복잡해 질수있음
2. ER 모델링이 테스트 전에 완료되어 있어야하며, 이는 개발은 더디게 한다.
3. RDBMS 시스템은 조인이 필요한 경우, 확장성을 제약함.
4. 여러서버를 사용한 sharding 기술이 가능하지만, 분포의 어포리케이션 코드가 필요하여 비효율적
5. 테이블에 다양한 가변성이 있는 데이터는 저장하기 어렵

Network

1. Frame, packet, Segment, Datagram을 비교하세요!



Keyword: 사용자 - Data

TCP - Segment

IP - Packet

데이터 링크 - Frame

Segment: 상위 계층에서 데이터를 전달받은 전송계층에서는
이러한 계층을 추가해 구성화한다. 이때부터 데이터는
Segment라 불린다.

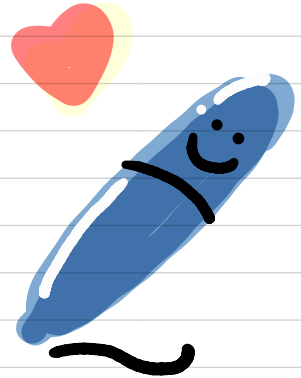
packet: 전송 계층으로 부터 전달받은 Segment는 네트워크
계층의 정보를 포함해 packet이라 불린다.

Datagram: 데이터그램은 순수한 사용자의 message를
다르게 부르는 말이다.

Frame: Data Link 계층에 전송되는 단위로 전송 프레임에는 상위 계층에서 보낸 전송 데이터에 오류 확인을
위한 체크섬, 송수신 호의 주소, 기타 프로토콜에서 사용하는 제어 코드 등의 정보가 포함된다.

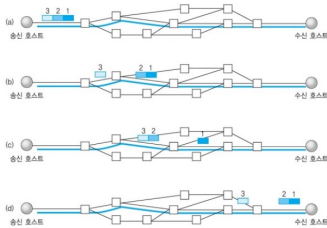
TCP와 UDP의 차이를 만해주세요!

프로토콜 종류	TCP	UDP
연결 방식	연결형 서비스	비연결형 서비스
패킷 교환 방식	가상 회선 방식	데이터그램 방식
전송 순서	전송 순서 보장	전송 순서 바뀔 수 있음
신뢰 여부 확인	O	X
통신 방식	1:1 통신	1:1 or 1:N or N:N
신뢰성	높다	낮다
속도	느리다	빠르다



TCP (Transmission Control Protocol)

인터넷상에서 데이터를 메시지 형태로 보내기 위해 IP와 함께 사용하는 프로토콜



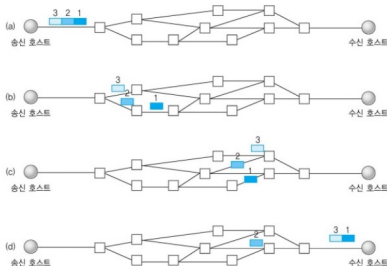
< 가상회선 패킷 교환 방식 >

TCP의 특징

- ① 연결형 서비스로 가상회선 방식 제공
- ② 3-way handshaking 과정을 통해 연결을 설정하고 4-way handshaking을 통해 해제한다.
- ③ 흐름 제어 및 혼잡 제어
- ④ 높은 신뢰도 보장
- ⑤ 전이중(Full-Duplex), 점대점(point-to-point) 방식

UDP (User Datagram Protocol)

데이터를 데이터그램 단위로 처리하는 프로토콜



< 데이터그램 패킷 교환 방식 >

UDP의 특징

- ① 비연결형 서비스로 데이터그램 방식을 제공
- ② 정보 교환에서 신뢰성 X
- ③ UDP 헤더에서 checksum 필드를 통한 최소한의 오류만 검출
- ④ 신뢰성 ↓, TCP 보다 빠름

Operating System

캐시의 지역성에 대해 설명해주세요

캐시메모리는 속도가 빠른 장치와 느린 장치간의 속도차의 따른 병목현상을 줄이기 위한 비용 메모리이다. 이러한 역할은 수행하기 위해서는 CPU가 어떤 데이터를 읽을 것인가를 어느 정도 예측할 수 있어야 한다. 캐시의 성능은 작은 용량의 캐시 메모리에 CPU가 나중에 참조할 정보가 얼마나 들어있느냐에 따라 좌우되기 때문이다.

이때 **적중률(hit rate)**은 높이기 위해 데이터 **지역성(Locality)**의 원리를 사용한다. 지역성의 전제조건으로 프로그램은 모든 코드나 데이터를 균등하게 Access 하지 않는다는 특성을 조건으로 한다. 즉 **Locality**란 기억 장치 내의 정보를 균일하게 Access 하는 것이 아닌 어느 한 순간에 특정부분을 집중적으로 참조하는 특성!

- 시간 지역성 : 최근에 참조된 주소의 내용은 곧 다음에 다시 참조되는 특성
- 공간 지역성 : 대부분의 실제 프로그램이 참조된 주소와 인접한 주소의 내용이 다시 참조되는 특성

Thread-safe란?

멀티스레드 프로그래밍에서 일반적으로 어떤 함수나 변수 혹은 객체가 여러 스레드에서 동시에 접근이 이루어져도 프로그램의 실행에 문제가 없음을 뜻한다.

Thread-safe 보장을 위한 방법.

critical section 정의에서 한번에 하나

점프 가능!



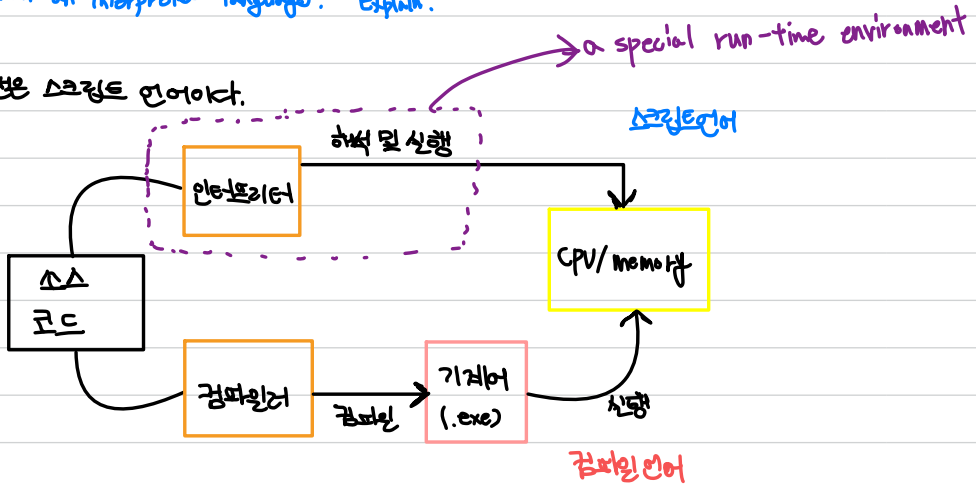
1. Mutual exclusion — 한 Thread가 자원에 접근할때 다른 Thread 막기
2. Re-entrancy — 어떤 함수가 실행중일때 다른 스레드가 호출하여도 결과가 각각 올바르게!
3. Thread-local storage — 각자 자신의 사본 공간만 제공하여 각각의 스레드에서만 접근 가능한 자원 사용.
⇒ 불변성 막음!
4. Atomic operation — 리소스에 접근할때 원자적인 이용. 현재적으로 정의된 방법 사용. → 상호 배제 구현!

Python

What type of language is python? programming or scripting?

Python is interpreted language. Explain.

파이썬은 스크립트 언어이다.



스크립트 언어

- special run-time environment 덕분에 단계가 많이 줄어듦.
- 수정이 빈번하게 발생할 경우에는 소스 코드를 Line by Line으로 읽어 실행하는 인터프리터 방식이 유리

→ 장점: 개발시간 단축, 소스 코드 수정이 간단.

단점: 중간코드를 만드는 것은 간단하지만 실제로 실행시키는 것은 많은 작업이 필요로 함. 실행시간이 오래 걸림.