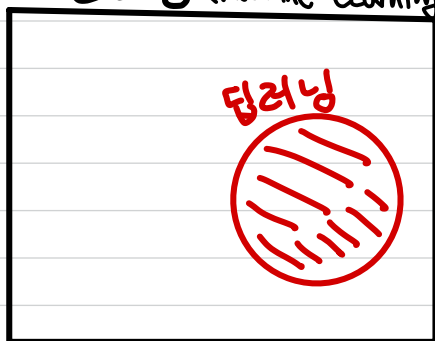


Deep Learning

• 딥러닝은 무엇인가요? 딥러닝과 머신러닝의 차이는?

딥러닝이란 인공지능영역을 사용하여 기계가 자동으로 대규모 데이터에서 중요한 패턴 및 규칙을 학습하고 이를 토대로 의사결정이나 예측 등을 수행하는 기술로 정의할 수 있습니다.

머신러닝 (machine learning)



머신러닝의 기본 정의

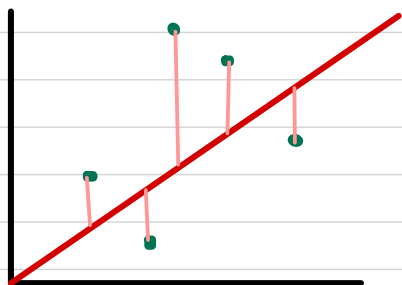
→ 데이터를 군 분석하고 해당 예시들을 통해 학습한 후 정보는 바탕으로 결정을 내리기 위해 학습한 내용을 적용하는 알고리즘.

기본 머신러닝 모델은 그 기능이 무엇이든 점진적으로 향상되는데, 약간의 인내가 필요하다. AI 알고리즘이 부정확한 예측을 반환하면 엔지니어가 개입하여 조정해야 합니다. 하지만 딥러닝 모델을 사용하면 알고리즘이 자체 신경망을 통해 예측의 정확성 여부를 스스로 판단해 학습을 진행합니다.

• Cost function과 Activation function은 무엇인가요?

• Cost (비용)

→ 가설이 얼마나 정확한지 판단하는 기준



• → 실제값

— → 예측값

— → 실제값과 예측값의 차이

비용함수 (cost function)

→ (예측값 - 실제값)²의 평균

$$\text{cost}(w, b) = \frac{1}{M} \sum_{i=1}^M (H(x^{(i)}) - y^{(i)})^2$$

∴ cost function으로 한 cost가 적을수록
실제와 맞은 가설은 선택했다고 할 수 있다!

How?

경사하강을 통해서!

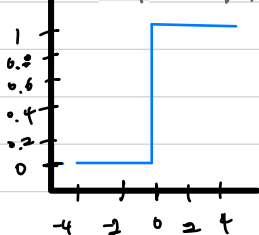
(gradient Descent)



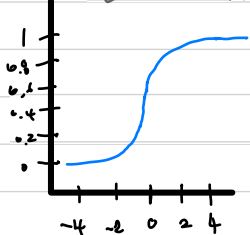
• 활성화 함수 (activation function)

→ 입력신호의 양상을 출력신호로 변환하는 함수
입력받은 신호를 얼마나 반복할지 결정하고
Network에 들어갈 값에 비선형성을 제공할 수 있게 해준다.

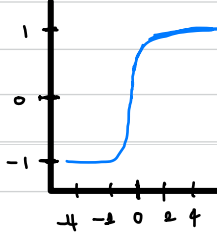
Step function



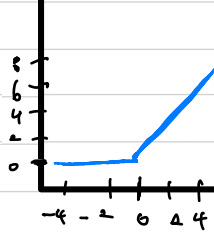
Sigmoid function



tanh function



ReLU function



Machine Learning

- 알고 있는 metric에 대해 설명해주세요! (ex: RMSE, MAE, recall, precision....)
↳ 평가지표!!

MSE (Mean Squared Error) : 평균 제곱 오차.

$$\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

오차의 제곱에 대한 평균을 취한 값으로 통계적 추정의 정확성에 대한 객적인 척도로 많이 이용됨.

RMSE (Root Mean squared Error)

$$\sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2}$$

MSE에서 오차를 값이 큰 것은 보강해준다.

MAE (Mean Absolute Error)

$$\frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|$$

실제값과 예측값의 차이의 절대값 합의 평균이다.
Outlier에게 robust하다는 장점이 있다.

R-squared (R^2)

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

R^2 은 결정 계수라고도 불리며 반응 변수의 변동량(분산)에서 현재 적용 모델이 설명할 수 있는 부분의 비율을 뜻하고 예측의 정확도를 0과 1사이의 값으로 계산하고 1에 가까울수록 설명력이 높다고 말한다.

분류 성능 평가 지표

		실제 정답	
		True	False
분류 결과	True	True positive	False positive
	False	False negative	True negative

precision: True라고 분류한 것 중 실제로 True 인 것.

$$\rightarrow \frac{TP}{TP+FP}$$

Recall: 실제 True인 것에서 모델이 True라고 예측한 것.

$$\rightarrow \frac{TP}{TP+FN}$$

Accuracy:

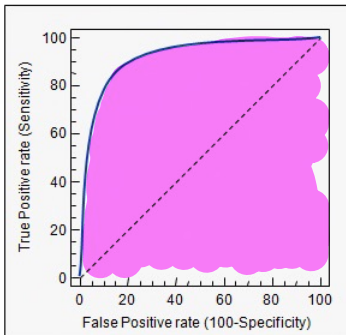
$$\frac{TP+TN}{TP+FN+FP+TN}$$

F1-score: precision과 recall의 조화평균

$$\Rightarrow 2 \times \frac{\text{precision} \times \text{Recall}}{\text{precision} + \text{Recall}}$$



ROC curve



Curve가 좌상단에 가까울수록 좋은 모델

● → AUC

① 정규화를 왜 해야 할까요? ② 정규화의 방법은 무엇이 있나요?

① 머신러닝 모델은 데이터가 가진 feature을 추출해서 학습한다.

이때 feature가 scaling 되지 않고 모델에 입력 된다면 모델이 데이터를 잘못 학습하여 성능에 지대한 영향을 준 수 있다.

따라서 이러한 문제가 발생하지 않도록 모든 데이터가 같은 정도의 scale로 반영되도록 해주는 것이 정규화의 목표이다.

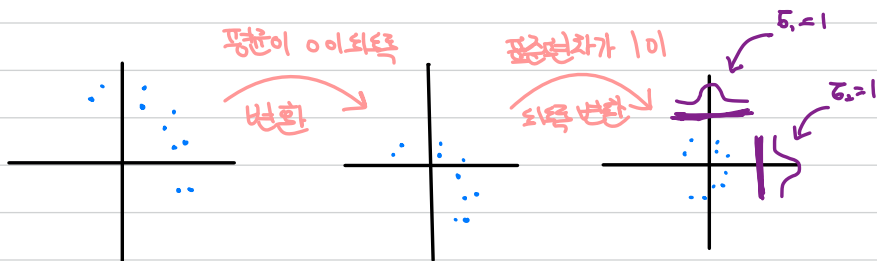
② min-max Normalization (최소 최대 정규화)

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

모델에 투입되는 모든 데이터 중에서 가장 작은 값은 0, 가장 큰 값을 1로 두고 나머지 값들을 비례로 맞춰서 0-1 사이로 scaling 해주는 것이다.

→ outlier에 취약

Z-score Normalization (Z 점수 정규화)



데이터를 표준 정규분포 (가우시안 분포)에 해당하도록 값을 바꾸어 준다.

Python

What is the difference between list and tuple in python?

list은 값을 수정할 수 있지만 → mutable

tuple은 값을 수정할 수 없다 → immutable

tuple은 list에 비해 더 작은 메모리를 필요로 하고, 속도가 빠르다.

What are the key feature of python? → 파이썬의 주요 특징?

1. python은 interpreted Language 이다.

↳ 실행 전에 기계어로의 컴파일 과정은 거치지 않으며 소스코드가 인터프리터에 의해 직접 해석되어 실행 되도록 구현된 언어

2. Dynamically-Typed Language

Python은 java처럼 정적형식이 아니다. 변수는 정의하는 동안 데이터 형식을 선언할 필요가 없다.

3. 파이썬은 객체지향적이지만 기능 및 객체지향 프로그래밍은 모두 지원합니다.
파이썬의 모든 것은 객체입니다.

Statistics / Math

• 고유값 (eigen value) 과 고유벡터 (eigen vector) 에 대해 설명해주세요! 그리고 왜 중요한가요?

①

②

① 행렬 A 를 선형변환으로 봤을때, 선형변환 A 에 의한 변환 결과가 자기 자신의 상수배가 되는 0이 아닌 벡터를 **고유벡터**라 하고 이 상수배 값을 **고유값** 이라고 한다.

즉 $n \times n$ 정방행렬 (**고유값, 고유벡터는 정방행렬에 대해서만 정의된다**) A 에 대해 $Av = \lambda v$ 를 만족하는 0이 아닌 원벡터 v 를 고유벡터 상수 λ 를 고유값이라 한다.

② 특이값 분해 (SVD), 주성분 분석 (PCA), 유사역행렬 (Pseudo-inverse) 등의 응용이 고유값, 고유벡터를 그 바탕으로 갖고 있기 때문에.

샘플링과 리샘플링에 대해 설명해주세요. 리샘플링은 무슨 장점이 있을까요?

샘플링은 모집단에서 임의의 sampling을 뽑아내는 것으로 표본추출을 의미한다.

모집단 전체에 대한 조사가 불가능하기 때문에 sampling을 통하여 모집단에 대한 추론을 하게 되는 것이다.

리샘플링은 내가 가지고 있는 샘플에서 다시 샘플 ~~추출~~을 뽑아서 통계량의 변동성을 확인하는 것은 이야기 같다. 즉 같은 샘플을 여러번 사용해서 성능 측정하는 방식이다.

→ k-fold 교차 검증, 부트스트래핑

Operating System

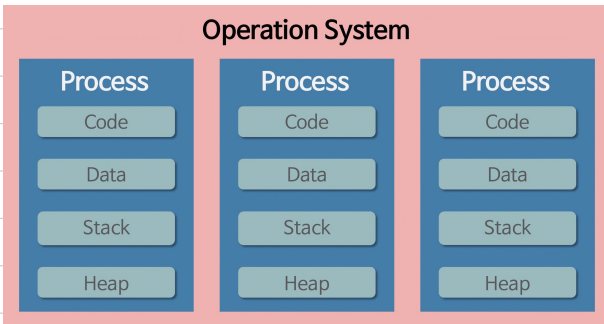
- 프로세스와 스레드의 차이를 알려주세요!

프로세스란?

• 사전적 의미

- 컴퓨터에서 연속적으로 실행되고 있는 컴퓨터 프로그램
- 메모리에 올라와 실행되고 있는 프로그램의 인스턴스 (동작 중인 개체)
- 운영체제로부터 시스템 자원을 할당 받는 작업의 단위
- 즉, 동작 중인 개념이라는 실행된 프로그램을 의미한다.

• 특징



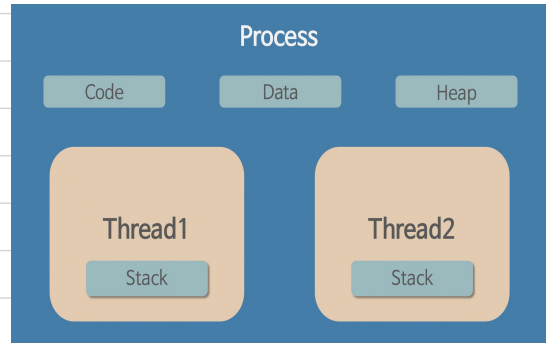
- 프로세스는 각각 독립된 메모리 영역 (Code, Data, Stack, Heap의 구조)을 할당받는다.
- 기본적으로 프로세스당 최소 1개의 스레드를 가진다.
- 각 프로세스는 별도의 주소 공간에서 실행되며, 한 프로세스는 다른 프로세스의 변수나 자료구조에 접근할 수 없다.
- 한 프로세스가 다른 프로세스의 자원에 접근하려면 IPC를 사용해야 한다.

• 스레드란?

• 사전적 의미

- 프로세스 내에서 실행되는 여러 흐름의 단위
- 프로세스의 특정한 수행 경로
- 프로세스가 할당 받은 자원을 이용하는 실행의 단위

• 특징



- 스레드는 프로세스 내에서 각각 Stack만 따로 할당 받고 Code, Data, Heap 영역은 공유한다.
- 스레드는 한 프로세스 내에서 동작하는 여러 실행의 흐름으로, 프로세스 내의 주소 공간이나 자원을 같은 프로세스 내에 스레드까지 공유하면서 실행된다.

- 멀티 프로세스 대신 멀티 스레드를 사용하는 이유는 설명해주세요

멀티 프로세스

→ 멀티 프로세싱이란 하나의 응용프로그램을 여러 개의 프로세스로 구성하여 각 프로세스가 하나의 작업을 처리하도록 하는 것이다.

• 장점

여러 개의 자식 프로세스 중 하나에 문제가 발생하면 그 자식 프로세스만 죽는 것 이상으로 다른 영향이 확산되지 않는다.

• 단점

Context Switching에서의 오버헤드

Context switching 과정에서 캐시 메모리 최적화 등 무거운 작업이 진행되고 많은 시간이 소요되는 등의 오버헤드가 발생하게 된다.

- 프로세스는 각각의 독립된 메모리 영역을 할당받았기 때문에 프로세스 사이에서 공유하는 메모리가 없어, Context Switching가 발생하면 캐시에 있는 모든 데이터를 모두 리셋하고 다시 캐시 정보를 불러와야 한다.
- 프로세스 사이의 어렵고 복잡한 통신 기법(IPC)
 - 프로세스는 각각의 독립된 메모리 영역을 할당받았기 때문에 하나의 프로그램에 속하는 프로세스들 사이의 변수를 공유할 수 없다.
- 참고 Context Switching란?
 - CPU에서 여러 프로세스를 돌아가면서 작업을 처리하는 데 이 과정을 Context Switching라 한다.
 - 구체적으로, 동작 중인 프로세스가 대기하면서 해당 프로세스의 상태(Context)를 보관하고, 대기하고 있던 다음 순서의 프로세스가 동작하면서 이전에 보관했던 프로세스의 상태를 복구하는 작업을 말한다.

멀티 스레드

- 멀티 스레딩이란
 - 하나의 응용프로그램을 여러 개의 스레드로 구성하고 각 스레드로 하여금 하나의 작업을 처리하도록 하는 것이다.
 - 윈도우, 리눅스 등 많은 운영체제들이 멀티 프로세싱을 지원하고 있지만 멀티 스레딩을 기본으로 하고 있다.
 - 웹 서버는 대표적인 멀티 스레드 응용 프로그램이다.
- 장점
 - 시스템 자원 소모 감소 (자원의 효율성 증대)
 - 프로세스를 생성하여 자원을 할당하는 시스템 콜이 줄어들어 자원을 효율적으로 관리할 수 있다.
 - 시스템 처리량 증가 (처리 비용 감소)
 - 스레드 간 데이터를 주고 받는 것이 간단해지고 시스템 자원 소모가 줄어들게 된다.
 - 스레드 사이의 작업량이 작아 Context Switching이 빠르다.
 - 간단한 통신 방법으로 인한 프로그램 응답 시간 단축
 - 스레드는 프로세스 내의 Stack 영역을 제외한 모든 메모리를 공유하기 때문에 통신의 부담이 적다.
- 단점
 - 주의 깊은 설계가 필요하다.
 - 디버깅이 까다롭다.
 - 단일 프로세스 시스템의 경우 효과를 기대하기 어렵다.
 - 다른 프로세스에서 스레드를 제어할 수 없다. (즉, 프로세스 밖에서 스레드 각각을 제어할 수 없다.)
 - 멀티 스레드의 경우 자원 공유의 문제가 발생한다. (동기화 문제)
 - 하나의 스레드에 문제가 발생하면 전체 프로세스가 영향을 받는다.

멀티 프로세스 대신 멀티 스레드를 사용하는 이유?

- 멀티 프로세스 대신 멀티 스레드를 사용하는 것의 의미?
 - 쉽게 설명하면, 프로그램을 여러 개 키는 것보다 하나의 프로그램 안에서 여러 작업을 해결하는 것이다.



- 여러 프로세스(멀티 프로세스)로 할 수 있는 작업들을 하나의 프로세스에서 여러 스레드로 나눠가면서 하는 이유?
 - 자원의 효율성 증대
 - 멀티 프로세스로 실행되는 작업을 멀티 스레드로 실행할 경우, 프로세스를 생성하여 자원을 할당하는 시스템 콜이 줄어들어 자원을 효율적으로 관리할 수 있다.
 - > 프로세스 간의 Context Switching시 단순히 CPU 레지스터 교체 뿐만 아니라 RAM과 CPU 사이의 캐시 메모리에 대한 데이터까지 초기화되므로 오버헤드가 크기 때문
 - 스레드는 프로세스 내의 메모리를 공유하기 때문에 독립적인 프로세스와 달리 스레드 간 데이터를 주고 받는 것이 간단해지고 시스템 자원 소모가 줄어들게 된다.
 - 처리 비용 감소 및 응답 시간 단축
 - 또한 프로세스 간의 통신(IPC)보다 스레드 간의 통신의 비용이 적으므로 작업들 간의 통신의 부담이 줄어든다.
 - > 스레드는 Stack 영역을 제외한 모든 메모리를 공유하기 때문
 - 프로세스 간의 전환 속도보다 스레드 간의 전환 속도가 빠르다.
 - > Context Switching시 스레드는 Stack 영역만 처리하기 때문
- 주의할 점!**
 - 동기화 문제
 - 스레드 간의 자원 공유는 전역 변수(데이터 세그먼트)를 이용하므로 함께 상용할 때 충돌이 발생할 수 있다.

Network

TCP/IP 의 각 계층에 대해 설명해주세요!

OSI 7 Layer

L7	응용 계층 (Application Layer)
L6	표현 계층 (Presentation Layer)
L5	세션 계층 (Session Layer)
L4	전송 계층 (Transport Layer)
L3	네트워크 계층 (Network Layer)
L2	데이터 링크 계층 (Data Link Layer)
L1	물리 계층 (Physical Layer)

TCP/IP 4 Layer

L4	응용 계층 (Application Layer)
L3	전송 계층 (Transport Layer)
L2	인터넷 계층 (Internet Layer)
L1	네트워크 액세스 (Network Access Layer)

TCP/IP 4계층	역할	데이터 단위	전송주소	예시	장비
응용 계층 (Application)	응용프로그램 간의 데이터 송수신	Data/Message	-	파일전송, 이메일, FTP, HTTP, SSH, DNS 등	-
전송 계층 (transport)	호스트 간의 자료 송수신	Segment	port	TCP, UDP, RTP, RTCP 등	게이트웨이
인터넷 계층 (Internet)	데이터 전송을 위한 논리적 주소 지정 및 경로 지정	packet	IP	IP, ARP, ICMP, RARP, OSPF	라우터
네트워크 연결 계층 (Network Access)	실제 데이터인 프레임의 송수신	Frame	MAC	Ethernet, PPP, Token Ring	브리지, 스위치

OSI 7 계층과 TCP/IP 계층의 차이

설명해주세요

1 계층 - 물리 계층(Physical Layer)

물리 계층은 전기적, 기계적, 기능적인 특성을 이용해서 통신 장비로 데이터를 전송하게 된다.

물리 계층에서 사용되는 통신 단위는 Bit로 1과 0으로 나타낸다. 즉 전기적으로 On, Off 상태라고 생각하면 된다.

물리 계층에서는 단지 데이터를 전달만 한다. 전송할 때(또는 받을 때) 데이터가 무엇인지, 어떤 의미가 있는지 등에는 전혀 신경 쓰지 않는다. 정말 단순히 데이터를 전기적인 신호로 변환해서 주고받는 기능만 할 뿐이다.

결국 물리 계층은 어떤 의미가 있는지 전혀 관여하지 않는다.

- PDU : 비트(Bit)
- 프로토콜 : Modem, Cable, Fiber, RS-232C
- 대표장비 : 허브, 리피터

프로토콜은 데이터 단위(PDU, Protocol Data Unit)는 데이터 통신에서 상위 계층이 전달한 데이터에 붙이는 제어정보를 뜻한다.

2 계층 - 링크 계층(Link Layer)

링크 계층은 네트워크 기기들 사이의 데이터를 전송하는 역할을 한다. 물리 계층을 통해 송수신되는 정보의 오류와 흐름을 관리하여 안전한 정보의 전달을 수행할 수 있도록 도와주는 역할을 한다.

링크 계층에서는 프레임에 주소장을 물리적으로 할당받는데 이는 맥(MAC; Media Access Control) 주소라고 불린다.

결국 링크 계층은 에러검출 / 재전송 / 흐름제어 역할을 한다.

- PDU : 프레임(Frame)
- 프로토콜 : 이더넷, MAC, PPP, ATM, LAN, Wifi
- 대표장비 : 브릿지, 스위치

3 계층 - 네트워크 계층(Network Layer)

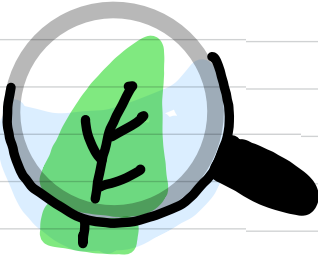
네트워크 계층은 네트워크에서 아주 중요하다.

중요한 기능 중 하나는 라우팅이다. 이는 데이터를 목적지까지 안전하고 빠르게 전달하는 기능을 말한다. 경로를 선택하고 주소를 정하고 경로에 따라 패킷을 전달해주는 것이 네트워크 계층의 역할이다.

네트워크 계층은 사용되는 프로토콜 종류도 다양하고 라우팅하는 기술도 다양하다. 또한 어느 컴퓨터에게 데이터를 전송할지 주소를 가지고 있어서 이를 바탕으로 통신한다. 우리가 아는 IP 주소가 바로 네트워크 계층의 헤더에 있다.

결국 네트워크 계층은 주소 부여(IP) / 경로 설정(Route) 역할을 한다.

- PDU : 패킷(Packet)
- 프로토콜 : IP, ICMP 등
- 대표장비 : 라우터, L3 스위치



OSI 7 Layer Model

7 Layer	Application Layer
6 Layer	Presentation Layer
5 Layer	Session Layer
4 Layer	Transport Layer
3 Layer	Network Layer
2 Layer	DataLink Layer
1 Layer	Physical Layer

TCP/IP Protocol

Application		
telnet	FTP	DHCP
HTTP	SMTP	DNS
		TFTP
		SNMP
TCP	Transport	UDP
Internet		
ICMP	ARP	RARP
		IP
Network Interface		

4 계층 - 전송 계층(Transport Layer)

전송 계층은 통신을 활성화하기 위한 계층이다. 양 끝단의 사용자들이 신뢰성있는 데이터를 주고 받게 해주는 역할을 한다.

보통 TCP 프로토콜을 이용하며, 포트를 열어서 응용프로그램이 전송을 할 수 있게 한다.

중요한 것은 데이터 전송을 위해서 Port 번호가 사용된다는 점이다. 대표적인 프로토콜로 TCP와 UDP가 있다. 이 계층에서 사용하는 데이터 단위는 세그먼트이다.

결국 전송 계층은 패킷 생성(Assembly/Sequencing/Deassembly/Error detection/Request repeat/Flow control) 및 전송 역할을 한다.

- PDU : 세그먼트(Segment)
- 프로토콜 : TCP, UDP , ARP, RTP
- 대표장비 : 게이트웨이, L4 스위치

5 계층 - 세션 계층(Session Layer)

통신 세션을 구성하는 계층으로, 포트(Port)번호를 기반으로 연결한다. 통신장치 간의 상호작용을 설정하고 유지하며 동기화한다.

세션이란 데이터가 통신하기 위한 논리적인 연결을 말한다. (통신을 하기 위한 문)

세션 계층은 TCP/IP 세션을 만들고 없애는 책임을 진다.

결국 세션 계층은 통신을 하기 위한 세션을 확립 / 유지 / 중단 역할을 한다. (운영체제가 해줄)

- PDU : 데이터(Data)
- 프로토콜 : NetBIOS, SSH, TLS

6 계층 - 표현 계층(Presentation Layer)

표현 계층(Presentation layer)은 코드 간의 번역을 담당하여 사용자 시스템에서 데이터의 형식상 차이를 다루는 부담을 응용 계층으로부터 덜어 준다. MIME 인코딩이나 암호화 등의 동작이 표현 계층에서 이루어지는 것이다.

예를 들면, EBCDIC로 인코딩된 문서 파일을 ASCII로 인코딩된 파일로 바꿔 주는 것, 해당 데이터가 TEXT인지, 그림인지, GIF인지 JPG인지의 구분 등이 표현 계층의 몫이다.

결국 표현 계층은 사용자의 명령어를 완성 및 결과 표현하며, 압축 / 암호화 역할을 한다.

- PDU : 데이터(Data)
- 프로토콜 : JPG, MPEG, SMB, AFP

7 계층 - 응용 계층(Application Layer)

응용 계층은 사용자와 바로 연결되어 있으며 응용 SW를 도와주는 계층이다. 사용자로부터 정보를 입력받아 하위 계층으로 전달하고 하위 계층에서 전송한 데이터를 사용자에게 전달한다.

파일 전송, DB, 메일 전송 등 여러가지 응용 서비스를 네트워크에 연결해주는 역할을 한다.

결국 응용 계층은 응용 프로세스와 직접 관계하여 일반적인 응용 서비스를 수행한다.

- PDU : 데이터(Data)
- 프로토콜 : DHCP, DNS, FTP, HTTP