

# The Essence of Deep Learning

<https://product.kyobobook.co.kr/detail/S000212175486>

데이터 구조와 연산 : 벡터와 행렬, 텐서 개념 이해

행렬 연산 : 덧셈, 뺄셈, 곱셈, 전치, 역행렬 연산

벡터 연산 : 크기, 방향, 내적, 외적, 노름 계산

행렬-벡터 곱셈 : 선형 변환, 데이터 표현, 연산 효율

기본 공간 : 벡터 공간, 차원, 기저, 선형 독립

열공간 : 행렬 곱으로 표현 가능한 모든 벡터 집합

영공간 : 행렬 곱에서 0이 되는 모든 벡터 집합

고유벡터와 고유값 : 선형 변환 고유 성질, 차원 축소, PCA 활용

사건과 확률 : 특정 결과 발생 가능성, 집합과 확률 척도

조건부 확률 : 사건 A가 일어난 경우 사건 B 확률,  $P(B|A)$

확률 변수 : 사건 결과 수치화, 이산/연속 확률 변수 구분

기댓값 : 확률분포 평균,  $E[X] = \sum x \cdot P(x)$  또는  $\int x \cdot f(x) dx$

분산 : 값의 흩어짐,  $Var(X) = E[(X - E[X])^2]$

베이즈 정리 : 사후 확률 계산,  $P(A|B) = P(B|A) \cdot P(A) / P(B)$

엔트로피 : 정보량 척도, 불확실성 정도,  $H(X) = -\sum P(x) \log P(x)$

교차 엔트로피 : 두 확률분포 차이,  $H(p,q) = -\sum p(x) \log q(x)$

KL 발산 : 분포 차이 측정,  $D_{KL}(p||q) = \sum p(x) \log(p(x)/q(x))$

연속 확률 분포 : PDF, CDF, 정규분포, 균등분포, 지수분포

확률 밀도 함수 : 연속 변수 확률 계산, 적분으로 확률 산출

누적 분포 함수 :  $X \leq x$  확률,  $F(x) = \int_{-\infty}^x f(t) dt$

확률 기대와 분산 : 연속 변수에서도 동일 개념 적용, 적분 계산

독립 사건 :  $P(A \cap B) = P(A) \cdot P(B)$ , 조건부 확률 단순화

조건부 독립 : 두 사건, 특정 조건 하에서 독립성 확인

지능형 머신 구축 : 데이터 기반 예측, 패턴 학습, 일반화  
전통적인 컴퓨터 프로그램 한계 : 규칙 기반, 복잡한 문제 처리 어려움  
머신러닝 동작 원리 : 입력 → 모델 → 출력, 손실함수로 학습  
뉴런 : 생물학적 뉴런 모사, 입력 가중치 합 → 활성화 함수  
선형 퍼셉트론 : 단순 선형 분류, 한계 존재, XOR 문제 처리 불가  
순방향 신경망 : 입력층 → 은닉층 → 출력층, 정보 단방향 전달  
선형 뉴런 한계 : 비선형 문제 처리 불가, 복잡 패턴 학습 어려움  
시그모이드, Tanh, ReLU 뉴런 : 비선형 활성화 함수, 학습 효율 개선  
소프트맥스 출력 레이어 : 다중 클래스 확률 출력, 합이 1  
요약 : 머신러닝 원리, 뉴런 모델, 활성화 함수와 한계 이해, 순방향 신경망 구조 파악

패스트푸드 문제 : 단순 예제, 입력과 출력 매핑, 학습 과정 이해  
경사하강법 : 손실함수 최소화, 파라미터 업데이트, 기울기 계산  
델타 규칙과 학습률 :  $\Delta w = -\eta \cdot \nabla L$ , 학습 속도 조절, 과소·과대 학습 방지  
시그모이드 뉴런 경사하강법 : 비선형 활성화, 연쇄법칙으로 미분, 가중치 업데이트  
역전파 알고리즘 : 출력층부터 입력층으로 오차 전파, 기울기 계산 자동화  
확률적 경사하강법, 미니배치 : 전체 데이터 대신 일부 샘플로 학습, 속도·안정성 개선  
테스트셋, 검증셋, 과적합 : 모델 일반화 평가, 과적합 방지, 하이퍼파라미터 조정  
심층 신경망 과적합 방지 : 정규화, 드롭아웃, 데이터 증강, 배치 정규화 활용

PyTorch 소개 : Python 기반 딥러닝 라이브러리, 자동 미분, GPU 지원  
PyTorch 설치 : pip, conda, CUDA 버전 확인, 환경 구성  
PyTorch 텐서 : 다차원 배열, NumPy 유사, 연산 효율적  
텐서 초기화 : zeros, ones, rand, arange 등으로 생성  
텐서 속성 : shape, dtype, device 확인, GPU 이동 가능  
텐서 연산 : 덧셈, 곱셈, 브로드캐스팅, 행렬 연산 지원  
PyTorch에서의 경사 : autograd, backward()로 자동 미분  
nn 모듈 : 레이어, 손실함수, 활성화 함수, 신경망 구성 도구 제공

데이터셋과 데이터로더 : Dataset 클래스, DataLoader로 배치 처리, 셔플, 병렬화  
MNIST 분류기 구축 : 입력→합성곱→활성화→풀링→출력, 학습, 검증, 평가

경사하강법 도전 과제 : 심층 신경망의 비선형 오차 표면, 극소점, 학습 난이도

심층 신경망 오차 표면 : 다수 극소점과 안장점, 최적화 어려움

모델 식별성 : 매개변수 대칭성, 동일 출력 여러 해 존재

가짜 극소점 영향 : 학습 정체, 수렴 지연

오차 표면 평평한 영역 : 기울기 작음, 학습 속도 느려짐

경사 방향 오류 : 최적화 비효율, 모멘텀 활용으로 개선 가능

모멘텀 기반 최적화 : 이전 기울기 활용, 학습 안정화, 진동 감소

이차 근사법 : Hessian 기반, 빠른 수렴, 계산 비용 높음

학습률 적응 : AdaGrad, RMSProp, Adam 등, 경사 변화에 따라 학습률 조정

옵티마이저 선택 철학 : 문제 특성, 모델 크기, 계산 자원, 수렴 속도 고려

인간 시각에서의 뉴런 : 시각 자극 반응, 특정 패턴 감지

피처 선택 한계 : 수작업 특징 추출 어려움, 복잡 이미지 표현 제한

기본 심층 신경망 확장 한계 : 완전연결층만으로 고차원 이미지 처리 한계

필터와 피처 맵 : 합성곱 필터로 특징 추출, 활성화 맵 생성

합성곱 레이어 상세 : 커널, 스트라이드, 패딩, 출력 크기 계산

맥스 풀링 : 다운샘플링, 불변성 제공, 계산량 감소

합성곱 신경망 아키텍처 : Conv→ReLU→Pooling 반복, FC→Softmax 출력

MNIST 문제 해결 : 합성곱 신경망 적용, 정확도 향상

이미지 전처리 파이프라인 : 정규화, 증강, 강건한 학습 지원

배치 정규화 : 내부 공변량 이동 감소, 학습 가속, 안정성 향상

그룹 정규화 : 메모리 제약 상황에서 배치 정규화 대안

CIFAR-10 합성곱 신경망 구축 : 다층 Conv, ReLU, Pooling, FC, 학습

학습 시각화 : 필터, 활성화 맵 확인, 모델 이해 및 디버깅

잔차 학습과 스킵 연결 : 깊은 네트워크 학습 용이, 그래디언트 소실 방지

잔차 신경망 구축 : ResNet, 인간 수준 이상 이미지 인식

합성곱 필터 예술 스타일 재현 : 이미지 변환, 특징 추출, 스타일 적용  
다른 문제 도메인 적용 : 합성곱 필터 활용, 영상, 오디오 등 확장

저차원 표현 학습 : 고차원 데이터 압축, 핵심 정보 유지  
주성분 분석 : 데이터 분산 최대 방향으로 축 설정, 차원 축소  
오토인코더 아키텍처 필요성 : 입력→잠재 공간→출력, 데이터 압축 및 특징 학습  
PyTorch 오토인코더 구현 : 인코더, 디코더 구성, 재구성 손실 최소화  
디노이징 : 입력 노이즈 제거, 강건한 표현 학습  
희소성 : 잠재 벡터 희소화, 중요한 특징 강조  
컨텍스트 정보 활용 : 입력보다 주변 정보에서 더 많은 표현 학습 가능  
Word2Vec 프레임워크 : 단어 임베딩, 단어 의미 벡터로 표현  
Skip-Gram 아키텍처 : 중심 단어로 주변 단어 예측, 단어 관계 학습

가변 길이 입력 분석 : 시퀀스 길이 다름, 패딩·마스킹 필요  
신경망 N-Gram : 고정 길이 입력 기반, 단어 연속 패턴 학습  
품사 태거 구현 : 단어에 품사 레이블, 문법 구조 이해  
의존성 파싱과 SyntaxNet : 문장 구조 트리, 단어 간 의존 관계 파악  
빔 서치와 전역 정규화 : 최적 경로 탐색, 출력 품질 향상  
스테이트풀 딥러닝 모델 : 이전 상태 기억, 시퀀스 학습 강화  
순환 신경망(RNN) : 시퀀스 데이터 처리, 내부 상태 업데이트  
경사 소실 문제 : 긴 시퀀스 학습 어려움, 장기 의존성 학습 제한  
장단기 메모리 유닛(LSTM) : 장기 의존성 학습, 게이트 구조로 정보 선택  
RNN 모델 PyTorch 기본 요소 : nn.RNN, nn.LSTM, nn.GRU, 텐서 입력/출력 처리  
감정 분석 모델 구현 : 텍스트 시퀀스 입력, 긍정/부정 예측  
순환 신경망 seq2seq : 입력 시퀀스를 다른 시퀀스로 변환, 번역 등 활용  
어텐션 증강 : 중요한 시점 집중, 성능 개선, 장기 의존성 해결  
번역 신경망 분석 : 인코더-디코더 구조, 어텐션으로 품질 향상  
셀프 어텐션과 트랜스포머 : 병렬 연산 가능, 장기 의존성 처리, NLP 표준 모델

생성적 적대 신경망(GAN) : 생성자와 판별자 경쟁 학습, 현실적 데이터 생성  
변이형 오토인코더(VAE) : 잠재 공간 샘플링, 확률적 데이터 생성  
VAE 구현 : 인코더→잠재 변수 샘플링→디코더, 재구성 손실과 KL 발산 최소화  
점수 기반 생성 모델 : 데이터 분포의 점수 함수 학습, 샘플링 활용  
디노이징 오토인코더와 점수 매칭 : 노이즈 제거 학습, 데이터 분포 근사

결정 트리와 트리 기반 알고리즘 : 직관적 규칙 기반, 피쳐 중요도 확인 용이  
선형 회귀 : 계수 해석, 변수 영향력 파악  
피쳐 중요도 평가 : 모델에서 입력 변수 영향 측정, 순열 피쳐 중요도, 부분 의존도 그래프  
추출적 합리화 : 입력 특징 기반 모델 판단 설명, 신뢰성 향상  
LIME : 로컬 선형 모델로 개별 예측 설명, 블랙박스 해석 가능  
SHAP : Shapley 값 기반, 공정한 기여도 계산, 피쳐 중요도 시각화

신경망 튜링 머신(NTM) : 외부 메모리와 신경망 결합, 읽기·쓰기 연산 학습  
어텐션 기반 메모리 접근 : 메모리 위치 선택, 정보 읽기/쓰기 집중  
NTM 메모리 주소 지정 : 연속·이산 주소, 가중치 기반 접근  
미분 가능 신경망 컴퓨터(DNC) : NTM 확장, 복잡한 데이터 구조 학습 가능  
DNC 간섭 없는 쓰기 : 기존 메모리 손상 방지, 효율적 업데이트  
DNC 메모리 재사용 : 사용 빈도 기반, 메모리 최적화  
DNC 쓰기의 시간적 연결 : 순차적 정보 기록, 시퀀스 처리 향상  
DNC 읽기 헤드 : 메모리 조회, 정보 추출  
DNC 컨트롤러 신경망 : 읽기/쓰기 제어, 학습 가능한 정책  
DNC 시각화 : 동작 중 메모리 접근, 학습 과정 이해  
PyTorch DNC 구현 : 컨트롤러, 메모리, 읽기/쓰기 헤드, 학습 루프  
DNC 독해 학습 : 시퀀스 데이터 처리, 질문-응답, 복잡 구조 이해

Atari 게임 심층 강화 학습 : 신경망 기반 에이전트, 화면 입력 → 행동 출력, 보상 학습  
강화 학습 소개 : 에이전트, 환경, 상태, 행동, 보상, 정책  
마르코프 결정 과정(MDP) : 상태 전이, 보상, 정책, 미래 보상 할인

정책 : 상태에서 행동 선택 전략,  $\pi(a|s)$   
미래 보상 : 장기 보상 합, 기대값 계산  
할인된 미래 보상 :  $\gamma$  사용, 먼 미래 보상 감소  
탐색과 활용 : 새로운 행동 시도 vs 최적 행동 선택  
 $\epsilon$ -그리디 : 확률  $\epsilon$ 로 무작위 행동, 탐색 균형  
어닐링된  $\epsilon$ -그리디 : 학습 진행 시  $\epsilon$  감소, 초기 탐색 후 활용 강화  
정책 학습과 가치 학습 비교 : 정책 직접 학습 vs 상태 가치 함수 학습  
정책 경사 폴 카트 : 연속 상태·행동, 정책 직접 최적화, OpenAI Gym 환경  
OpenAI Gym : 강화 학습 환경, 시뮬레이션, 표준화된 인터페이스  
에이전트 만들기 : 상태 입력, 행동 선택, 학습 루프 구축  
모델 및 최적화기 구축 : 신경망, 손실함수, 옵티마이저 정의  
샘플링 액션 : 정책 따라 행동 선택, 환경과 상호작용  
이력 추적 : 보상, 상태, 행동 기록, 학습 피드백  
정책 경사 main 함수 : 학습 루프, 손실 계산, 파라미터 업데이트  
폴 카트 PGAgent 성능 : 학습 후 균형 유지, 보상 최대화  
신뢰 영역 정책 최적화(TRPO) : 정책 변화 제한, 안정적 학습  
근접 정책 최적화(PPO) : 간단한 TRPO 대안, 효율적 업데이트  
Q러닝과 DQN : 가치 기반 학습, 행동-가치 함수 최적화  
벨만 방정식 : 상태 가치 함수 재귀적 정의  
가치 이터레이션 문제 : 계산 비용, 수렴 속도  
Q함수 근사화 : 신경망으로  $Q(s,a)$  근사  
DQN : 경험 리플레이, 타깃 Q 네트워크, 안정적 학습  
DQN 훈련 : 샘플 배치 학습, 손실 최소화  
학습 안정성 : 타깃 네트워크, 경험 리플레이, 과적합 방지  
프레임 스택킹 : 연속 프레임 입력, 상태 표현 강화  
DQNAgent 결과 : Breakout 등 게임에서 성능 평가  
DQN 개선 : 심층 순환 Q, 비동기 우위 액터-크리틱, 보조 학습, 비지도 강화