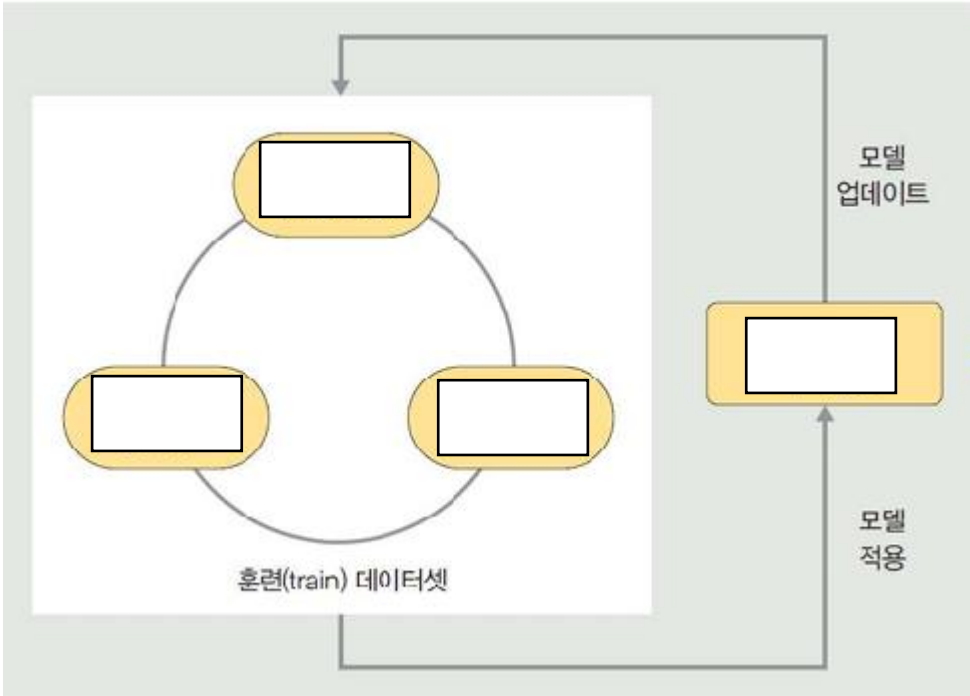


컴퓨터가 학습할 수 있게 하는 알고리즘과 기술을 개발하는 분야:

데이터별로 어떠한 특징을 갖고 있는지, 데이터를 벡터로 변환하는 작업: ( )

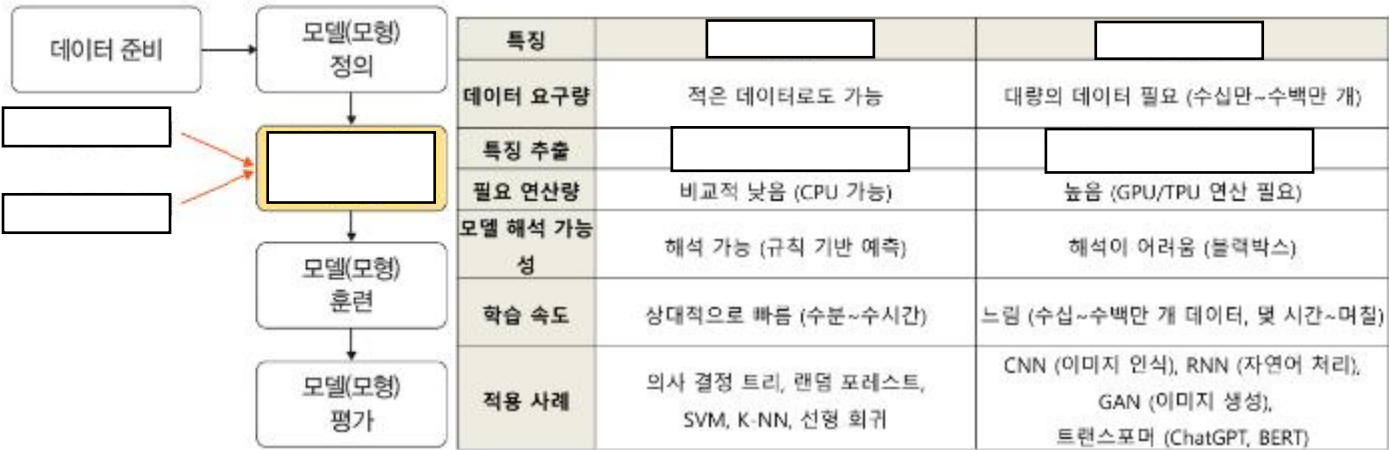
머신러닝의 학습 단계에서 얻은 최종 결과물로 가설이라고 하는 것은? ( )



머신러닝의 하위 분야로 인간의 뇌 신경망을 모방하여 데이터를 학습하는 기술:

인간의 뇌는 수많은 ( ) 과 ( ) 로 구성되어 있다

epoch가 10, batch가 20, testData가 1000일 때 가중치 업데이트 식과 횟수: / \*



데이터를 한번에 학습하는 양을 뭐라고:

입력데이터와 정답데이터가 함께 제공되는 학습:

- 주요 기법 2개: ,

정답 데이터 없이 입력데이터 많으로 학습하는 방식:

- 주요 기법 2개: ,

에이전트가 환경과 상호작용하며 보상을 최대화하는 전략 학습 방식:

- 주요 개념 5개: , , , ,

다른 변수에 의해 영향을 받는 변수:

- 위 변수에 영향을 주는 변수:

데이터가 평균을 기준으로 얼마나 퍼져 있는지:

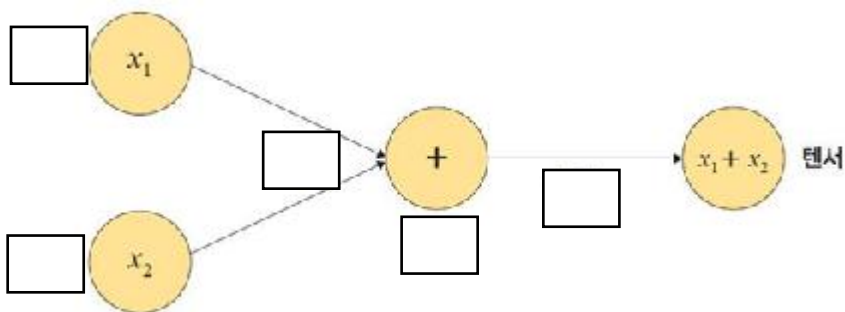


- 3번을 다른 용어로:

1차원 배열 형태:

2차원 배열 형태:

3차원 이상의 배열 형태:



- 수학적 연산을 처리하는 것:
- 사이의 관계를 표현하는 것:
- 이동되는것:

모델 컴파일의 주요 파라미터:

- 훈련하는동안 출력과 실제 값 사이의 오차 측정:
- 훈련과 검증단계를 모니터링하여 모델의 성능 측정:
- 데이터와 손실 함수를 바탕으로 업데이트 방법을 결정:

모델 훈련 코드:

모델 활용 예측 코드:

데이터셋 내부 관계를 일반화하여 처음 보는 데이터 결과 예측하는 학습자:

비슷한 레코드를 검색해서 예측하는 학습자:

- 대표적 알고리즘:

관측 값이 어느 특정한확률분포를 따른다고 전제할 수 없거나 모집단에 대해 아무런 정보가 없을 때 활용하는 방법:

관측값이 특정한 확률분포를 따른다고 전제, 분포의 모수에 대한 검정 실시하는 방법: 모수적 방법 (                      method)

두 레코드 사이의 유사도를 수량화 하는 것:

- 거리:                      ,                      ,
- 등등:                      ,                      ,                      ,

개별 속성 사이의 차이의 합을 갖고 구하는 것은:

데이터 셋에 있는 모든 속성들의 사이의 차이 중 최댓값:

유클리드 거리와 맨해튼 거리를 일반화한 거리:

수치형 척도시 사용 방법:

이진형 척도시 사용 방법:                      ,

수치형으로 변환하면 정보가 많아진다: O, X

k-nn에서 k개의 최근접 이웃이 결정되면 k개중 (                      )클래스가 (                      ) 클래스

knn 가중치 두가지 조건: 이웃들로부터 주어진 데이터 포인트까지 거리에                      . 모든 가중치 합

이진형 속성을 갖고 얼마나 두 데이터가 유사한지 계수

- $x = (01001011)$ ,  $y = (11101100)$ 일 때:

텍스트 문서 유사도:

- (                      ) 횟수를 제외하고 (                      ) 횟수만 유사도 측정

벡터의 방향이 얼마나 유사한가:

서포트 경계 머신은 학습 후에는 모든 학습용 데이터가 불필요하다: O, X

- 경계를 (                      ) 활용하는 (                      )가 필요

서포트 벡터 머신의 경계 명칭:                      (                      )

속성이 2개인 경우 데이터포인트 차원:

속성이 3개인 경우 데이터포인트 차원:

좋은 경계 설정 방법: 두 구역 사이의 (                      ,                      )이 최대가 되도록 하는 경계

경계의 구성에 영향을 주는 데이터 포인트:

데이터를 선형 분리가 가능한 형태로 변환해주는 함수:

결과 모델이 트리 구조인 모델:

- 장점:                      ,
- 단점:

- 결정 트리의 성능과 구조를 결정하는 핵심 기준:

확률 변수의 불확실성을 수치로 환산한거 두 종류:

- 0~1:
- 0~0.5:

알고리즘 성능 평가에 사용되는 행렬: 혼동 행렬

		예측 값	
		Positive	Negative
실제 값	Positive		
	Negative		

- 그리기:

여러 개의 결정 트리를 무작위로 만들어 결과를 종합해 최종 예측하는 학습 방법:

- 이 학습 방법은 어떠한 학습 방법인가:

분류 문제 위한 회귀:

연속적인 숫자값 예측을 위한 회귀:

- 독립변수가 하나일 때:
- 독립변수가 여러 개일 때:

데이터를 입력받아 K개의 군집으로 묶는 알고리즘:

- 과정: - - -

중심점으로부터 각 군집 내의 데이터포인트들의 거리의 제곱을 모두 더한 값 용어: ,

데이터 밀도를 기준으로 클러스터를 자동으로 찾아주는 알고리즘:

- 뜻:
- 이 알고리즘은 노이즈에 영향을 덜 받는다: O, X
- 밀도 범위:

데이터의 차원을 줄이면서 핵심 정보를 최대한 보존하는 기법:

- 제1 주성분은 데이터 ( ) 이 가장 ( ) 방향
- 제2 주성분은 ( ) 이랑 ( ) 하며, 두번째로 ( ) 방향

PCA 과정: - - - -

- 각 변수의 평균을 0으로 맞추는 과정:

PCA는 학습이다: O, X

입력 값이 연산 결과에 미치는 영향력을 조절하는 요소:

각 노드에서 들어오는 신호\*가중치 모두 더한 합계: ( )

전달 함수에서 받은 값을 출력시 일정 기준에 따라 출력 값 변화시키는 비선형 함수:

선형 함수의 결과를 0~1 사이에서 비선형 형태로 변형해 주는 함수:

- 대표적 문제:

위 함수를 평균을 0으로 만드는 함수:

음수면 0, 양수는  $x$ :

- 대표적 문제:

음수일 경우 매우 작은 수, 양수는  $x$ :

입력 값을 0~1 사이에 출력되도록 정규화 하여 총합이 항상 1이 되도록 하는 함수:

오차를 구할 때 사용하는 함수:

손실 함수의 순간 기울기를 활용해 가중치 업데이트 하는 방법:

실제 값과 예측 값의 차이를 제공하여 평균을 낸 것:  $\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$

분류 시 원 핫 인코딩과 같이 사용하는 손실함수:  $\sum_{i=1}^n \sum_{j=1}^K -y_{ij} \log(\hat{y}_{ij})$

훈련 데이터가 들어올 때:

위 과정을 통해 도출된 오차로 가중치 수정할 때:

은닉층이 많을수록 생기는 문제 3개:  $\rightarrow$  과적합,  $\rightarrow$  기울기 소멸,  $\rightarrow$  기울기 폭발

과적합 해결 방안:

기울기 소멸 문제 해결한 활성화 함수:

- 음수값 학습 안됨 해결:

경사 하강법의 데이터가 너무 크고, 지역 최소점 탈출 어려운 문제:

- 해결법  $\rightarrow$  미니배치 경사 하강법
- 모든 데이터별 한번씩 수정하는 것:
- 데이터를  $n$ 개로 쪼개서 평균을 합해서 가중치 수정:
- 오차가 상대적으로 더 들쭉날쭉한 하강법:

현실적으로 전역 최소점 찾기 어려운 이유:

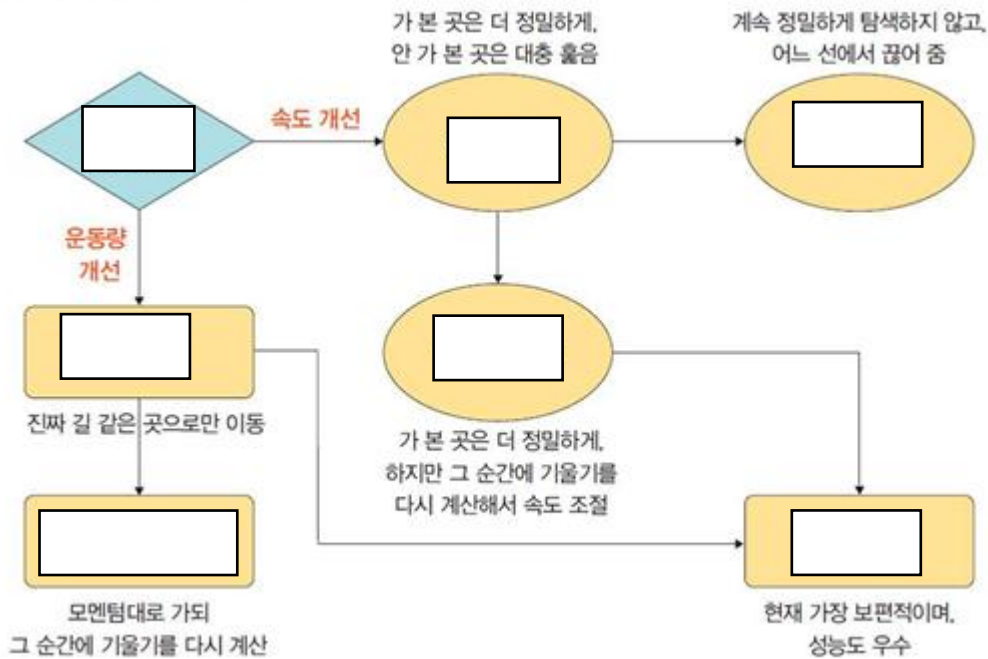
SGD일때 1000개 데이터를 활용할 때 epoch당 가중치 업데이트 횟수:  $n$  번

Batch GD일 때 10000개 데이터 활용 시 epoch당 가중치 업데이트 횟수:  $\frac{n}{10000}$  번

Mini-Batch GD일때 100000개 데이터 학습 중 10개씩 나누어 할 때 가중치 업데이트 횟수:  $\frac{n}{10}$  번

가중치를 얼마나 어떤 방식으로 업데이트 할 지 결정하는 알고리즘:

- 종류:  $\rightarrow$  경사 하강법,  $\rightarrow$  확률적 경사 하강법,  $\rightarrow$  미니배치 경사 하강법



가중치 업데이트 횟수에 따라 학습률 조정하는 방법:

- 문제:

$$w(i+1) = w(i) - \frac{\eta}{\sqrt{G(i)+\epsilon}} \nabla E(w(i))$$

$$G(i) = G(i-1) + (\nabla E(w(i)))^2$$

$w(i)$ :

$\eta$ :

$G(i)$ : 지금까지의

$\epsilon$ : 0으로 나누는 걸 방지하기 위한 아주 작은 값

$\nabla E(w(i))$ : 현재 단계의 기울기

최근의 기울기 들 반영:

$$w(i+1) = w(i) - \frac{\sqrt{D(i-1)+\epsilon}}{\sqrt{G(i)+\epsilon}} \nabla E(w(i))$$

$$G(i) = \gamma G(i-1) + (1-\gamma)(\nabla E(w(i)))^2$$

$$D(i) = \gamma D(i-1) + (1-\gamma)(\Delta(w(i)))^2$$

분모 G:

분자 D:

- 학습률 따로 정해야 된다: O, X

AdaGrad의 G(i)값이 무한히 커지는 것 방지하고자 한 최적화 함수:

- 사용자가  $\gamma$ 를 통하여 G값 조절 가능: O, X

- ( )을 이용하여 최근 데이터에 더 많은 가중치를 주는 경향

- $\gamma$ 은 ( )

관성을 준 SGD:

- 지그제그 현상이 줄어들지는 않는다: O, X
- 문제:

모멘텀+기울기:

- 해결 문제:

이전 기울기 방향 참고하고 RMSProp처럼 크기 변화도 자동 조절 옵티마이저:

- 영문 뜻:

합성곱 신경망 층들: , , , , ,

- 입력 데이터 형식:
- 특성 추출하기 위해 사용:
- 이동 간격:

특성맵 차원 다운샘플링 방법:

- 주요 2개: ,

3차원 벡터를 1차원으로 펼쳐지는 층:

출력층 활성화 함수:

합성곱 종류: , ,

- 좌우로:
- 상하좌우:
- 상하좌우 앞뒤:
- 3D합성곱 깊이와 입력의 깊이가 같으면 결과는:

특성맵 크기 축소 방지:

사전 훈련된 모델 사용:

마지막 완전연결층 부분만 새로 만드는 것:

학습 여부 설정 파라미터: =

flatten 이전 마지막 합성곱 층 적용 풀림:

사전 훈련 모델 사이트:

사전 훈련 모델과 유사성 작을 경우:

데이터 셋을 증가시키는 클래스:

더욱 데이터에 맞도록 가중치 업데이트 하는 기법:

데이터셋이 적고 자료 유사도가 적으면 수정하는 부분은 합성곱의 일부이다: O, X

데이터 셋이 작고 자료 유사도가 높으면 완전 연결층만 수정한다: O, X

미세 조정을 너무 많이 하면 과적합 발생 가능성이 낮아진다: O, X

모델 결과의 신뢰 향상을 위해 중간 처리과정을 시각화 한 CNN:

입력이 그래프인 합성곱: 그래프 합성곱 네트워크 ;

상호 연결 된 항목을 ( )로, 미 연결된 항목을 ( ) 으로 작성하여 ( ) 로 변환  
- 위 행렬: 인접행렬

인접행렬을 ( ) 로 변환한다.

과정: -> ->  
- 2, 3 사이에 : ( ) 주입

얀 르쿤이 개발한 초창기 CNN 모델:

	현 CNN	LeNET5
활성화 함수		
파라미터 수	수백만 ~ 수 억개	파라미터 적음
정규화/최적화 함수		정규화 및 최적화 O, X
활용범위	수많은 분야	제한적

중간 부분만 잘라서 활용:

학습 내역을 불러주는 것: