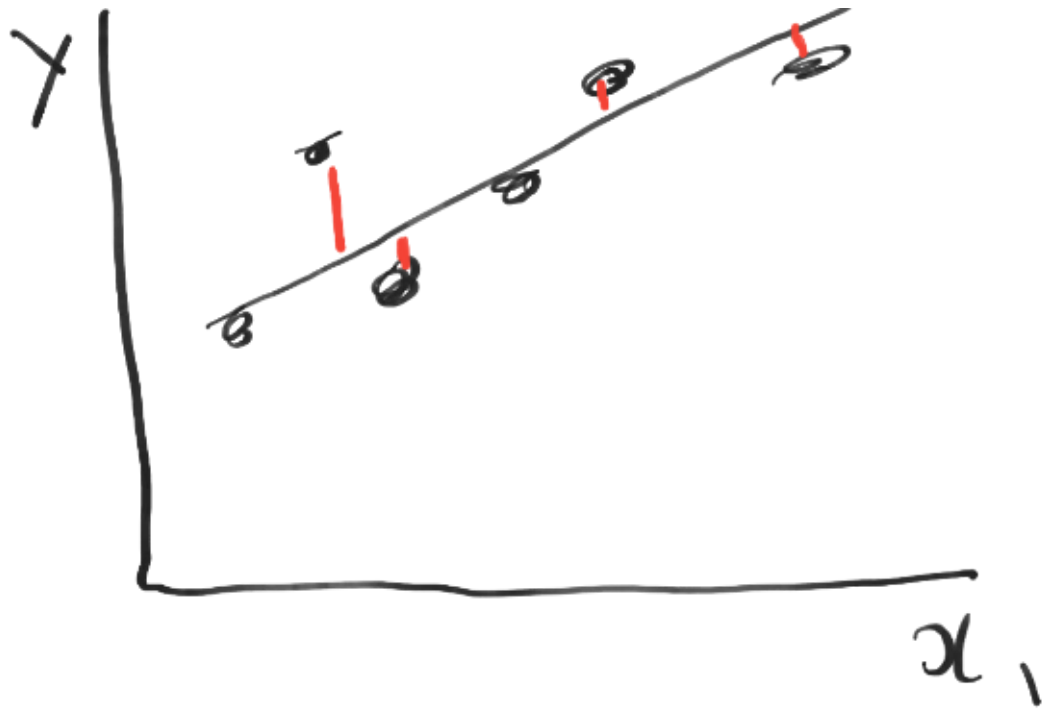
머신러닝



$y = w_1 x_1 + b$

손실 적을수록 정확함

loss

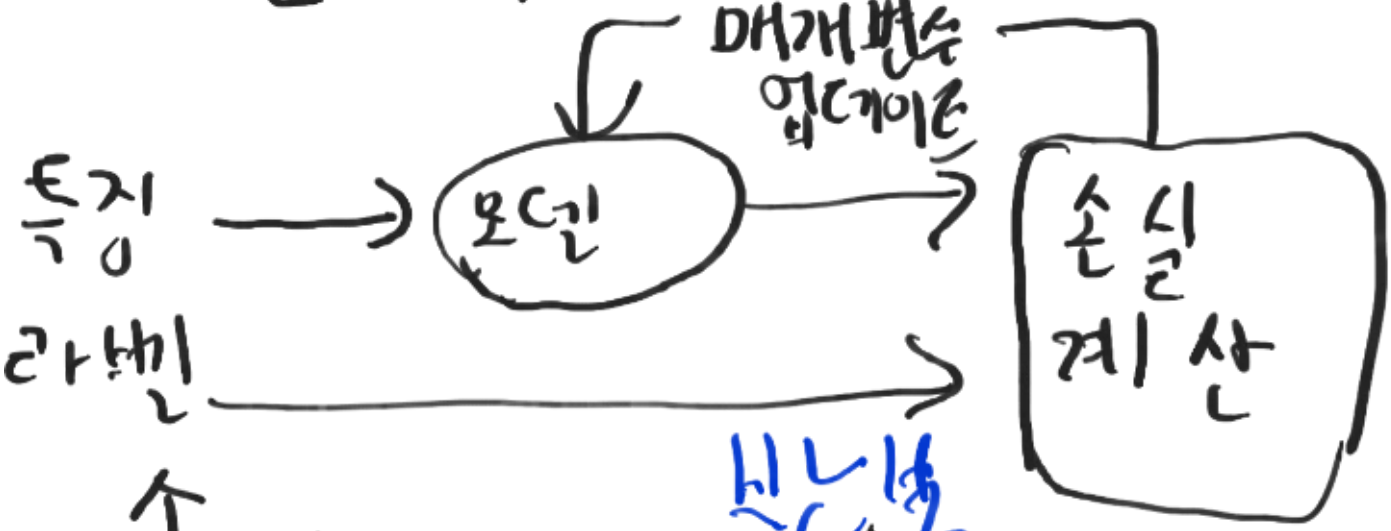
$= (y - y')^2$ → L_2 손실

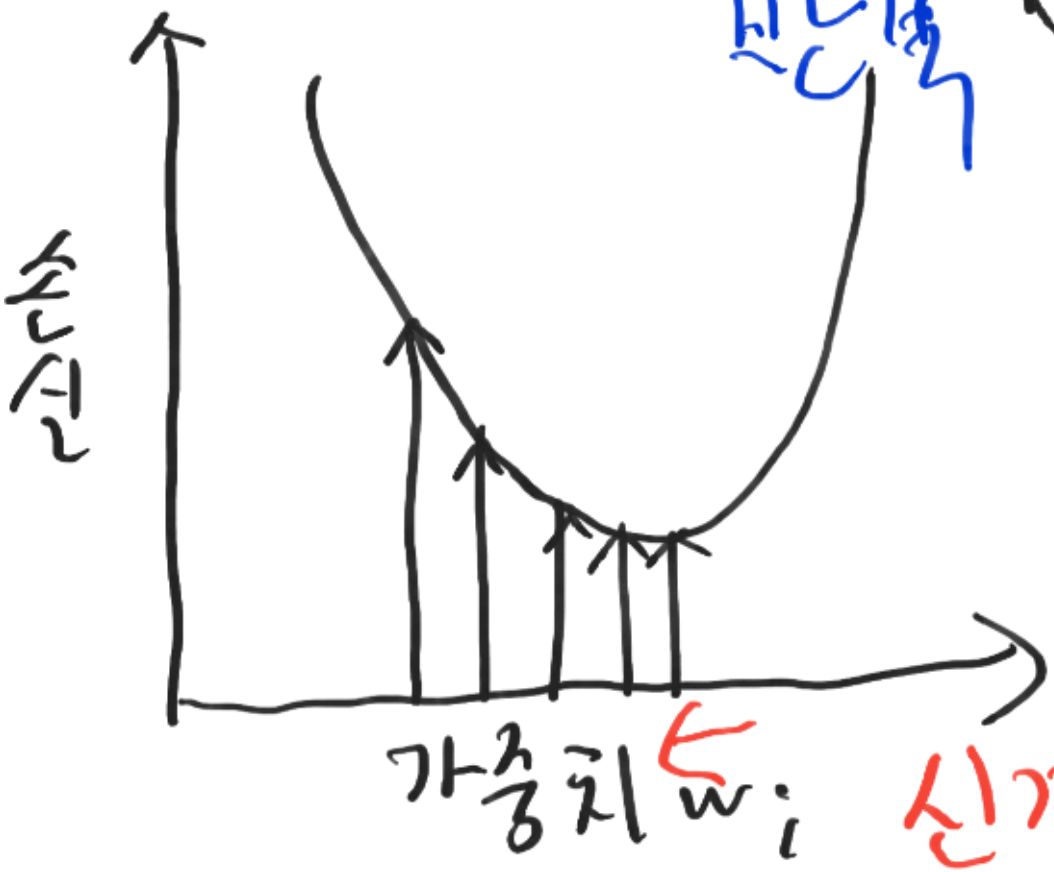
실제값 − 예측값

전체손실

$= (loss_1 + loss_2 \cdots\cdots loss_n)/n$

손실 줄이기 학습법





신경망

경사 하강법은

시작점 중요X

극소값 찾기 유용한 수로는 아니지만

음의 방향으로 감소시킨다.

(기울기의 반대방향으로)

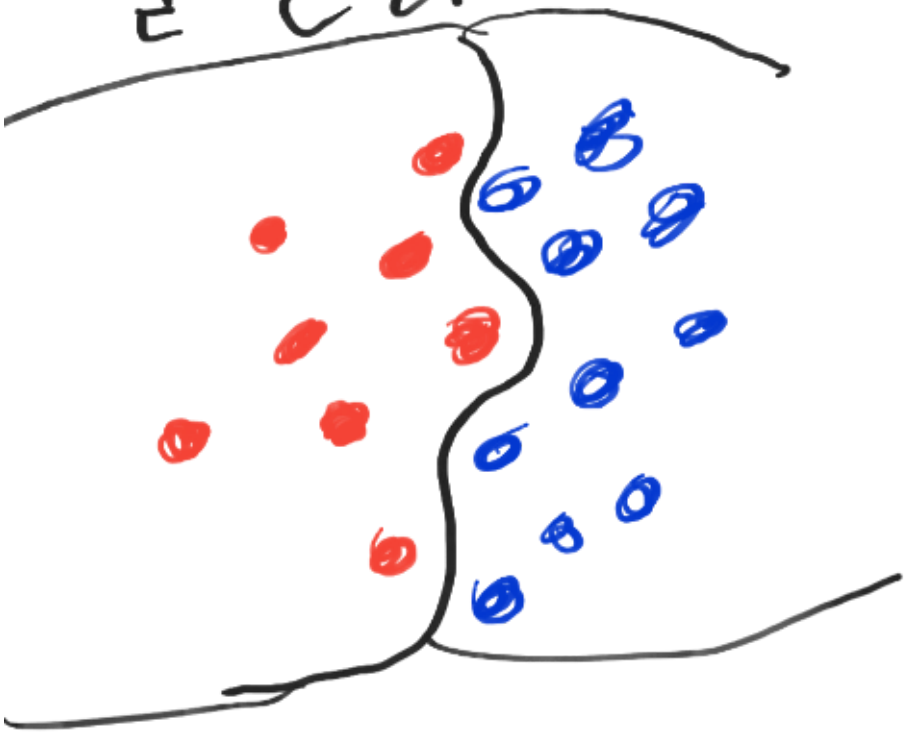
경사하강법은
기울기에 학습률을 곱하여
다음지점결정
학습률이작으면 속도↓
〃 크면 최저점을 넘어감

1. 확률적 경사 하강법
데이터가 많을수록
계산속도↑
랜덤으로 데이터하나를
뽑아서 사용 경험적으로
증명은 되었지만 논리는 모르다.
2. 미니 배치 확률적 경사 하강법

1~10~100배의

데이터를 랜덤으로 샘플

일반화



학습

실제데이터 샘플

i.i.d.

모델

전체 데이터

예측

새샘플

1.1.c

→ 경험적인 접근 방식

데이터 분포에서
새로운 데이터를 뽑아 실험
테스트 세트 선택

주의점

1. 독립적 개별적으로
데이터를 뽑아야 한다.

2. 정상성 시간이 지나도
변하지 않아야 한다.

3. 항상 동일한 데이터분포
사용 새로운 데이터
추가하면 안된다.

학습 테스트 세트를

오컴의 면도날 법칙

데이터는 단순할수록 좋다.

과적합의 발생

학습하는 동안 손실은
적지만 잘예측하지못한다.

↳ 데이터 적합도를
유지하되 최대한
단순하게바꾸어야한다.

학습세트

테스트세트 성능이 좋아야
한다.

1. 테스트세트가충분히크고
2. 같은세트를반복사용X

[illegible]

많이 적기

하나로 편중되지 않게하기

과적합 (여름, 겨울)

모든 학습 사례가 정확하게 같은

분류되는 모델이 있다.

↳ 새로운 사례가 들어오면

결과도 별로고 잘 들어맞지

않을수 있다. (모델이 너무 복잡)

→ 과적합의 발생

Tensorflow Estimators ← 높은 수준의 개체지향

tf.layers, losses, metrics 일반모델 구성요소를 재사용 가능레이어로 제공

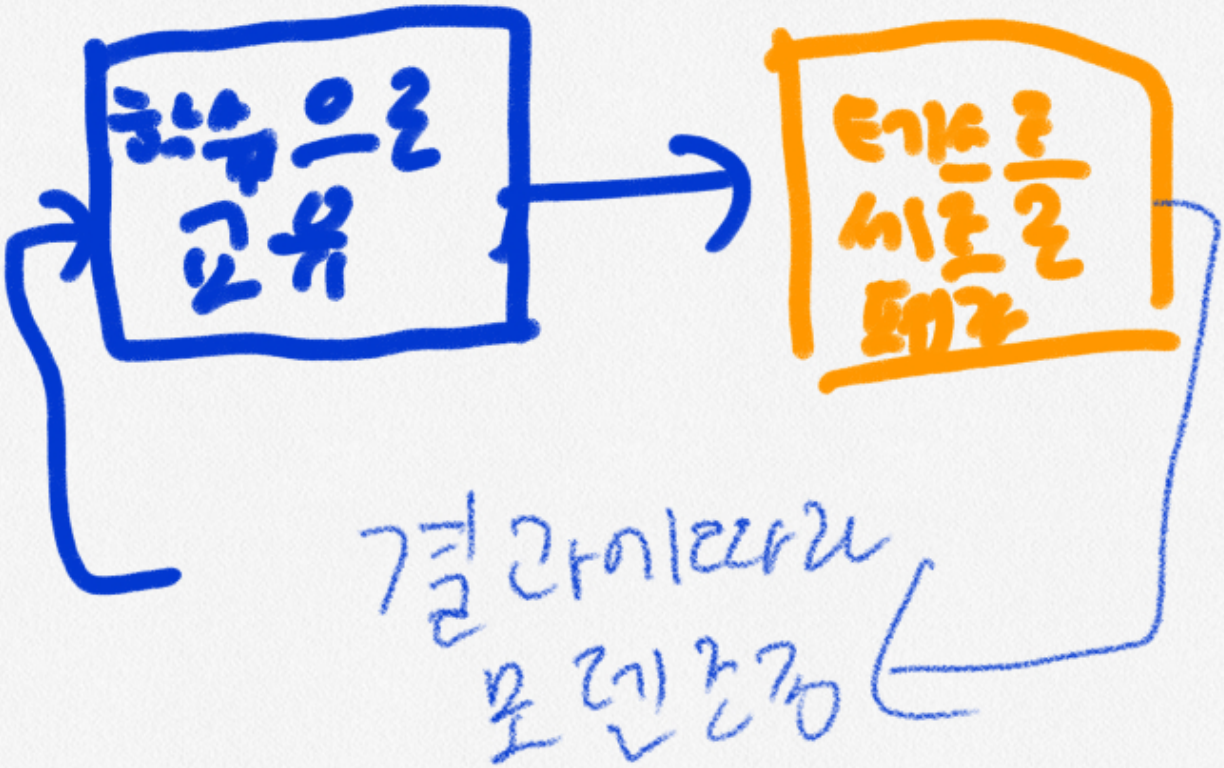
tf python C++커널을 래핑하는 오퍼레이션 제공

tf C++

CPU GPU TPU 커널은 하드웨어별의 플랫폼에서 작동

TensorFlow는
그래픽 프로토콜 버퍼
분산된 그래프를 실행하는 런타임
JVM과 비슷하게
모든 CPU GPU 등에서
작동 가능해야 한다.
최근 수 년의 API 추상화
사용되는 유연한
취급은
최근의 API 추상화
나중에 수준 낮출 것이다

검증



↓

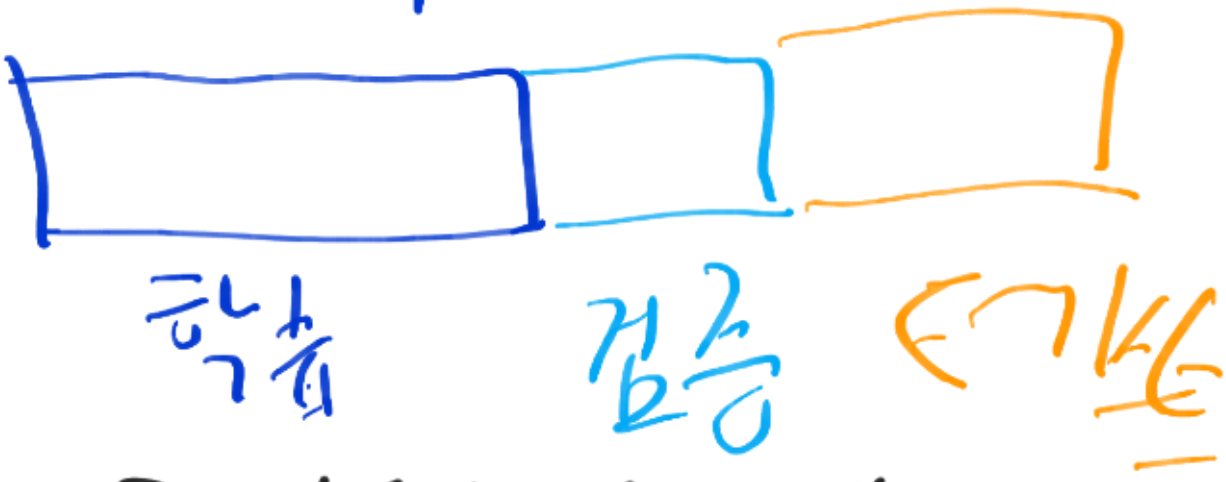
가장 우수한 모델을 선택

반복 후 → 학습률 조정

테스트세트를 통해 최적 모델 찾음

과적합 모델이나온 우로인나

해결책



테스트데이터 X

검증데이터를 통해 모델스러운
결과가 나올때까지 하이퍼파라미터
등을 바꾸가며 검증
검증데이터를 통해확인 후
테스트데이터를 통해 재확인

→ 학습 → 검증

결과조정

표현
특성추출

숫자 ⊗ → 문자 (유한 한계로 → 더 빠른 결과)

↓

숫자 [0,0,0,1,0,0,1,0...0] 원핫인코딩 숫자

좋은 특성

1. 0이아닌 데이터가 연속적으로 나와야한다.

2. 분명하고 명확해야한다.

3. 특성은 특수값을 포함해선 안된다

4. 특성의 정의는 시간이 지나도 변하지 않아야합니다.

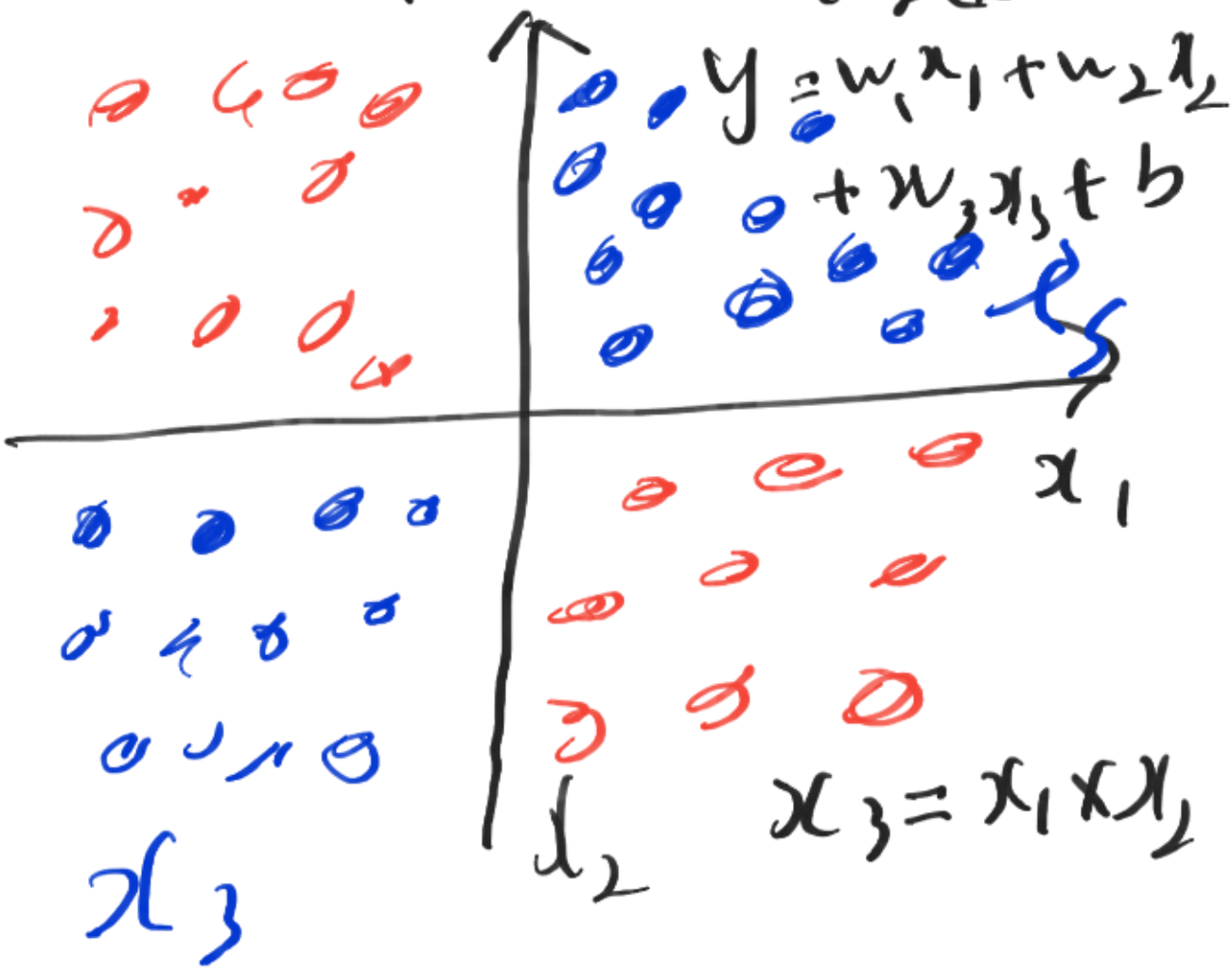
5. 분단에 극단적인 이상점이 없어야한다.

비니지니

특정지역의 빈을 나타내는
1 지역 나머지는 0으로 가능

슈 관들
시각화 출연빈도 → 히스토그램
디버그 잘 못된 기에 터가 있는 데
모니터링
특성 범위 및 시간의 다른
예 매일 애일 상태가 변환
특성 교차 다.
1. 양식 A×B의 특징들 값을
정의합니다.
2. 더 복잡한 수도 있다.
3. A와 B가 빈 가 같을 경우

습의 결과 범위가 매우
희소하게 나타낼 수 있다

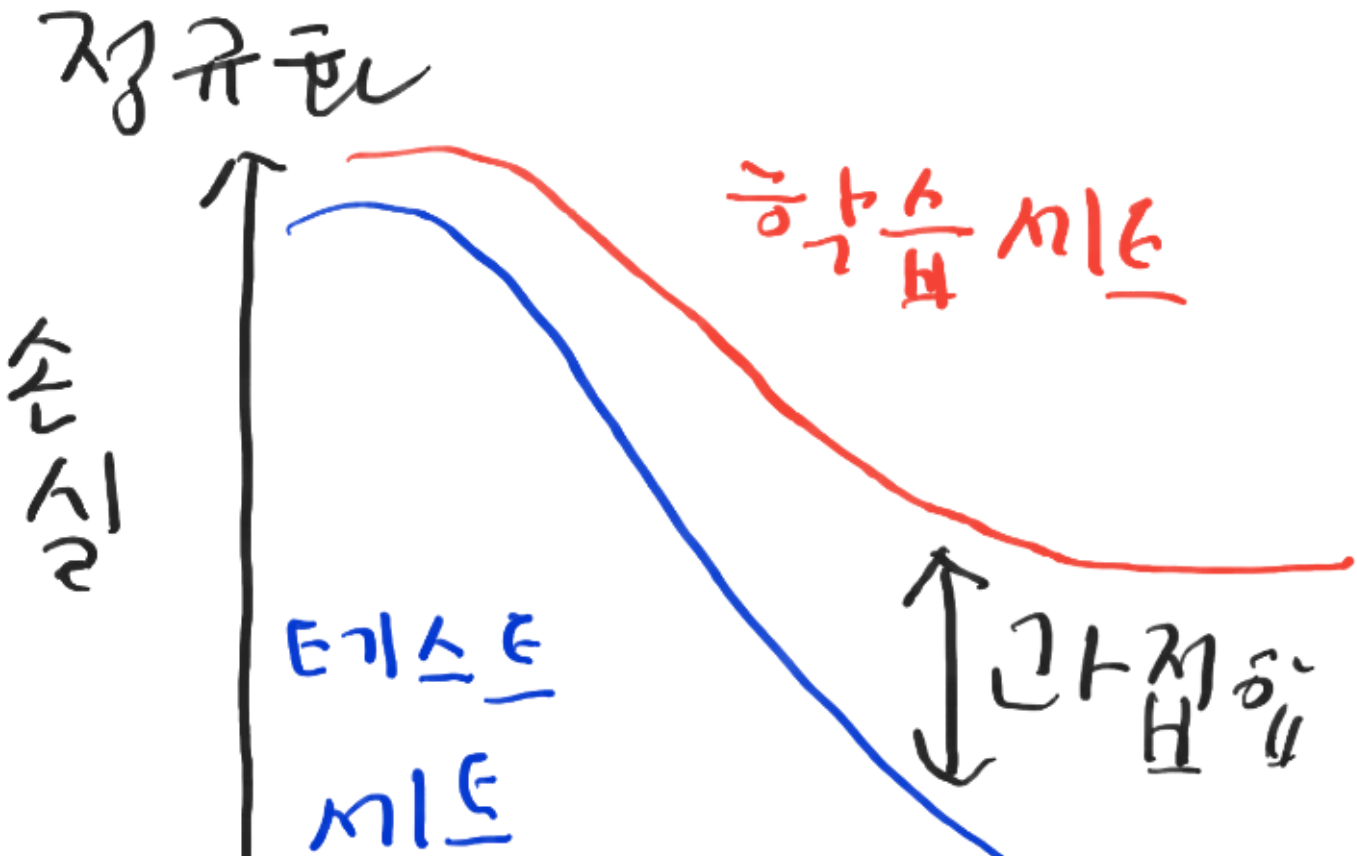


↳ 선형학습자만
비선형학습자를
합하여 표현가능하다.

선형 → 데량의데이터
에맞다.

특정 오차가 없으면
훈련이 불가능하다

↳ 특정 오차와 대량의 데이터를 사용하면 복잡한 모델은 훈련 불가능하다.



학습 반복

과적합을 피하기위한 방법

정규화

1. 조기종료

학습시트가 상승하기전에

멈춘다.

2. 모델복잡도에 패널티주기

모델복잡성에 페널티를 준다

최소화 → 손실 + 복잡도

모델 정의하는 방법

1. 더작은 가중치 선호

학습사례를 축출하면서 매개변

수작기만든다.

L2 정규화

복잡도 = 가중치의 제곱 합

아주 큰 가중치에 대한 페널티를 부여

선형 모델에서는 더 평평한

기울기선을

데이터만 사전 확률

가중치는 0을 중심으로

가중치는 정규분포되어야

함

수식 모델복잡도

$L(w, D) + \lambda \|w\|_2$

주제, 점수

등에 나누어진다

더 단순한

L 학습오류 낮추는것이 목표

λ 가중치의 균형을 유지하는

방법을 제어하는 스칼라값

w 복잡도를 낮출수 있다

$\frac{2}{2}$: 다음의 제곱 : L_2 가중치

C_2 정규화

$\|w\|_2^2 = w_1^2 + w_2^2 + \cdots$

w_n^2

가중치 높으면 모델 단순

데이터 라도 적합

↳ 중복한 학습 X

↓ 모델 복잡 라면

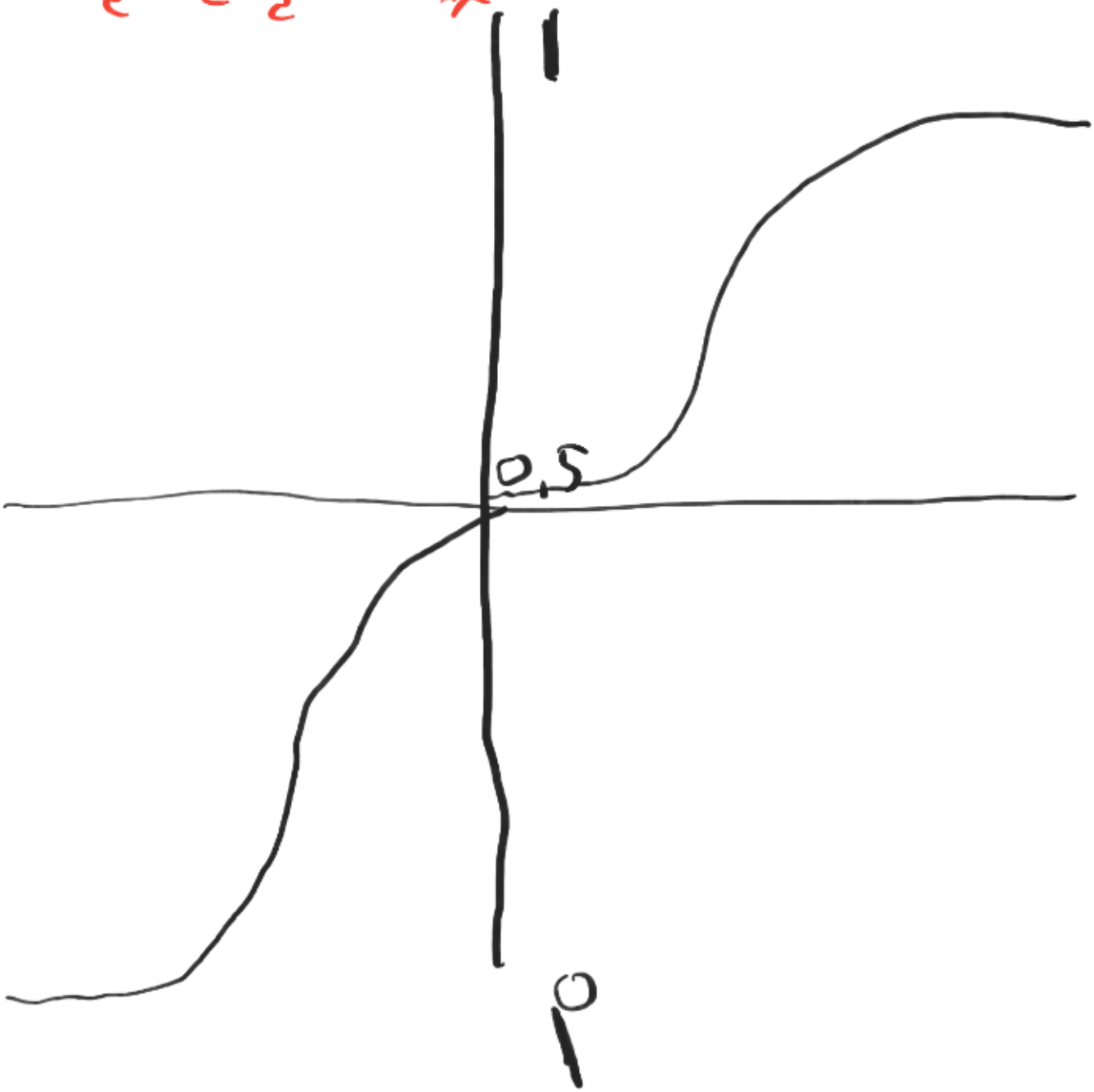
특수성을 너무 많이 학습

하게되고 새로운 데이터 인식 X

로지스틱 회귀

0 ~ 1 까지 예측값이 벗어나면 안된다.

0 ~ 1 사이의 잘 보정된 확률을 얻을 수 있다.



$$y' = \frac{\quad}{(w^T x + b)}$$

$$y = \frac{1}{1+e^{-(wx+b)}}$$

$$f(x) = \frac{1}{1+e^{-x}}$$

cost

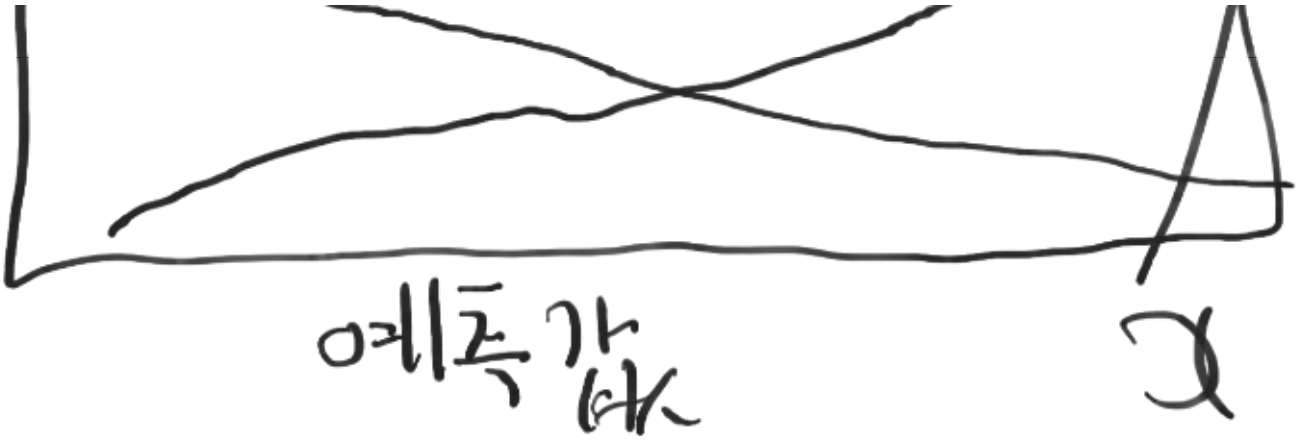
1에서 멀어질수록 cost↑

1이면 cost 0

0에서 멀어질수록 cost↑

0이면 cost도 0

손실



$$\text{Log Loss} = \sum_{(x,y)} -y \log y' - (1-y) \log(1-y')$$

1. 베타로 학습 및 예측 정확하다.

비선형구조 고차에이터 추가하면 된다.

정규화 ⊗⊗

하지만 다변 고차원에서

계속 업데이트 할 것이다.

L2 정규화

조기중단

나

분류

정확성

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

평가법 : 올바른 결과로 모두 계산후

1. 정확성 다음 존재하는 모든 결과로 올바른 예측 비율 나눈다.

단 : 양성이나 음성이 거의 없는 클래스 불균형 등이 있다.

양성/음성 다른 클래스를 예측하는 방법을 세분화해야 한다.

TP

늑대가 나타났다고
올바르게 외침

FP

늑대가 나타났다
고 거짓말

FN

늑대가 있지만
늑대 발견 X
닭 모두잡힘

TN

늑대도 없고
경고도 없다.

정밀도 재현율 (둘다 양성)

외침과 동시에 얼마나
늑대등장율 만은 늑대가 정확한가

양성중 실제양성 실제양성중 정확히양성

정밀도와 재현율 둘다 중요!!

분류임계값 이 있어야

수정확률

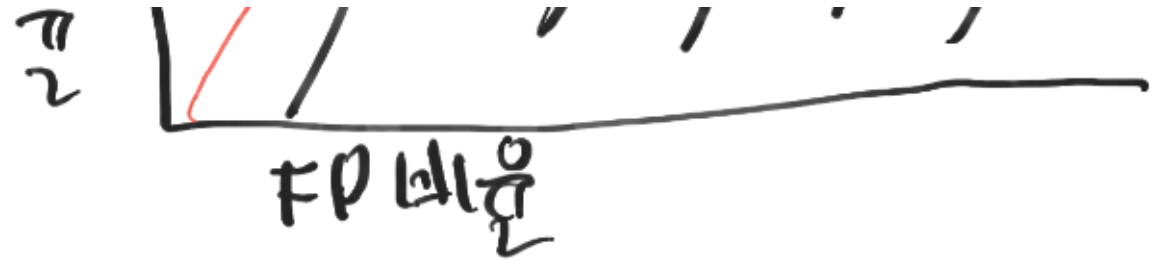
둘다 구할수 있다 상상

분류 임계값 구하기

1. 여러 분류 임계값에서 모델적용 평가

↳ ROC곡선, 수신자 조작특성 곡선

1. 모든 분류 임계값 평가후 TP율과 FP율을 확인



AUC ROC 곡선 아래 영역

이 0.9라면 쌍 비교가 정확할

확률도 0.9이다.

2. 예측 편향

모든 예측 값의 합 ⟷ 모든

관찰자의 합

비교

둘이 동일하면 편향이 아니다.

아무런 값이 없는 모델

→ 무편향 모델

무편향이 없으면 잘못된

모델이다.

→ 어떤 영역에서 몰려 있는지 체크

보정 모델

그룹을 배깅한 후 평균 예측값과 관찰값을 비교한다.

보정이 유의미하려면 다량의 데이터가 필요

특성 교차

문제점

희소 특성 교차 → 모델 매우 커짐

노이즈 계수, 런타임 증가

↳ 정규화 필요 (메모리를 나눈 마술 지팡이 0으로 방향

두개 값에 비교하여

0이 아닌 가중치의 페널티

↳ 볼록함수의 아니다

문제

↳ 절대값의 합의

페널티를 주다,

ㄴ 다양한 소멸 유형과

ㄴ2 와는 달리 희소성을

유지

↳ 지금의 대해 페널티

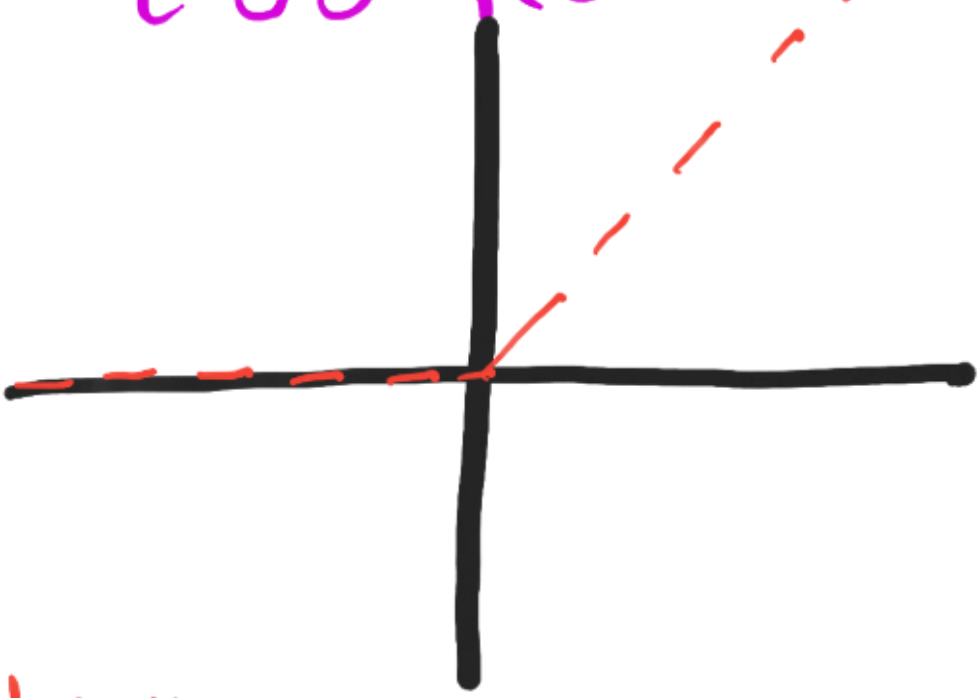
매우 큰, 작은 가중치 수

0은 아니지만 0에 가깝게 만들다

신경망 비선형데이터의 중요

비선형성을
우아해진다

비선형성 ReLU 예)



될 수 최적화를 사용할 수

있음으로 초기화중요

학습법

역전파

비 선형이란

$b + w_1 x_1 + w_2 x$

→ 정확이라면도
예측할 수 없다

역전파

학습할 사례를 변별해주는
경사가 매우중요하다.
미분가능한 함수가 필요하다.
경사가 사라지는 경우

너무 깊어져서
최소효과 깊이로 제한한다.
ReLU → 사용으로 해결
학습률이 너무 높아지면
발산해버리고 NaN값을

얼게된다.

→학습률 재조정

ReLU 실행 되지않는다.

0이하 작동X 모든값이 0이하로

나온다면 작동X

학습률 재조정

심층 네트워크 학습시에는

드롭아웃이라는 정규화방식을

기억해두면 좋다

확률 p에서 노드하나를

택하고 네트워크의 단일경사

스텝에서 이노드를 삭제합니다

다른경사스텝에서는 이과정

반복할수록 정확해진다

전체 → 단순 유사로

→ 중간의 양이 유용하다.

다중클래스 신경망

로지스틱회귀는 양자택일에 유용하다

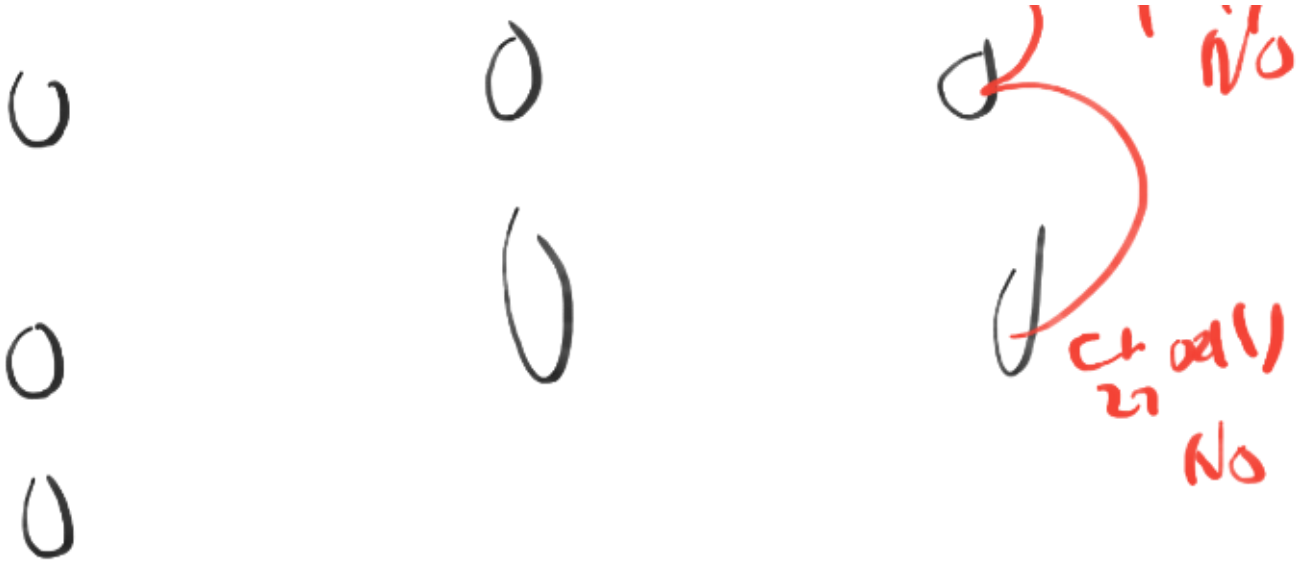
다중클래스

방법1

히든	히든	로지스
O	O	O
O	O	O

출력 NO

개에!



처음에는 여러가지 출력노드를 쌓
하고 이후 내부표현을 공유하면
합리적, 효율적 학습 가능하다.

특정 사례가 하나의 클래스에
속한다. 예)

사과
바나나 — 과일
배

이미지이다

→ 소프트맥스 사용
로지스틱 회귀를 여러클래스로 일반화
한다.

다중 클래스, 단일 라벨
하나의 예 → 한 클래스 입니다
클래스가 상호 배타적인
제약조건이 유용한 구조
손실이 서 인코딩 유용
모든 가능한 클래스에 하나의
소프트맥스 손실 사용

다중 클래스 다중 라벨
하나의 예가 두개 이상의 클래스
클래스 입력에 추가로 적용될
제약 없음

각각의 클래스에 하나의 로지스틱
회귀모델을 의미

소프트맥스
문제가 구조화되면 출력값을
잘 보정된 형태의 확률로
사용할 수 있다.

분리조건

하나의 사진의 여러개의 라벨중
의 고양이 또는 사람+개
가 섞인 사진을 계산되면
합이 1일 필요가 없다
1대다 분류전략을 사용
해야 한다

클래스의 수가 많을때

후보 샘플링의 사용

출력노드를 속한클래스의 맞게

훈련시키고 네거티브클래스의

샘플을 가져와서 출력노드의

샘플만 업데이트한다.

5마리의개를

4마리 X

1마리양성으로한다

소프트맥스 추가

각클래스의 소수점값을

확률해주며 합은 1이 되어야한다

$$P(y=j|x) = \frac{e^{(w_j^T x + b_j)}}{\sum e^{(w_k^T + b_k)}}$$

$\sum_{k \in K}$

전체 소프트맥스

모든 가능한 클래스의 확률을 계산하는

소프트맥스

확률 샘플링

임베딩 (훈련 워밍업

→ 특정 사용자의 관심 샘플

영화 추천 시스템 아키텍

2차원으로 표현하며

작업을 위해서 50 ~ 100차원도

가능

d차원 임베딩

d개의 특징이 있는 영화 수만큼 가드

데이터를 통해 학습 가능

심층 앙이에서 입베딩 학습

차원당 단위하나를 갖는 히든

레이어에서 변로의 학습과정

이해보 하기 방법

입력 값을

0 ~ 9999 로 입력 받는

예) 1. 주택가격

예상과 실제 판매가격 사이의

제곱 손실 최적화

입베딩 차원수를 늘리면

과적합 발생 가능

차원 $\approx \sqrt[4]{\text{possible value}}$

유사한 항목이 서로 근접하도록
실제벡터로 해싱
임베딩을 인접데이터로 적용가능

다양한데이터를 공동으로 임베딩
하면 서로간의 유사성이 정의

망손기
가중치가 커지 MxN의 가중치를
계산해야한다
데이터의 양수

계산량과 벡터간의 의미있는 관계 부족

임베딩

크기가 큰 희소벡터를 의 의존적 관계를 보존하는 저차원공간으로 변환하는 임베딩 사용

고차원 → 저차원

희소한 벡터에서 밀집벡터를 얻으려면 계산량이 임베딩 진행후 전반적으로 된다

어휘함 용량가 포함되었다면

임베딩이 항목수를 곱한 후
단을 이용 함께의 곱가 가능

임베딩 학습 알고리즘
Word2Vec
분포가설에 기반하여
의미론적으로 유사한 단어를
기하학적으로 가까운
임베딩 벡터로 매핑

함께 등장하는 단어목록과 문장위
그룹화된 단어를 구분하도록
신경망 학습시킨다.

대상 단어대신 무작위 노이즈
단어를 추가해 음의 예를 만들어
양의 예와 음의 예 둘중에
어떤 쌍이 함께 표시되는지
구분법을 학습시킵니다.

마지막 수정: 오후 5:03