

머신러닝

머신러닝 - 입력을 결합하여 이전에 본 적이 없는 데이터를 적절히 예측하는 방법!

라벨 => 예측하는 실제 항목 . 기본 선형 회귀의 y 변수 (스팸인지 아닌지 정도)
(라벨은 예측하는 항목입니다 (단순 선형 회귀의 y 변수) . 밀의 향후 가격, 사진에 표시되는 동물의 종류, 오디오 클립의 의미 등 두

특성-> 데이터를 설명하는 입력 변수 (방식) (이메일의 포함된 정도, 헤더 정보, 이메일에서 찾을 수 있는 정보) (특성은 입력 변수

예) 데이터의 특정 인스턴스임!

라벨이 있는 예 (특성, 라벨) (x, y) 이 포함됨 (모델을 학습시키는 데 사용!)

라벨이 없는 예 (특성, ?) $(x, ?)$ 이 포함되며! (새 데이터를 예측하는데 사용)

모델은 예를 예측된 라벨 (y') 에 매핑함 (학습되는 내부 매개변수에 의해 정의되!)

학습은 모델을 만들거나 배우는 것을 의미합니다. 즉 라벨이 있는 예를 모델에 보여 주고, 모델이 특성과 라벨의 관계를 점차적으로

추론은 학습된 모델을 라벨이 없는 예에 적용하는 것을 의미합니다. 즉 학습된 모델을 사용하여 유용한 예측 (y') 을 해냅니다. 예를

회귀 모델은 연속적인 값을 예측합니다. 예를 들어 회귀 모델은 다음과 같은 질문에 대한 답을 예측합니다.

캘리포니아의 주택 가격이 얼마인가요?

사용자가 이 광고를 클릭할 확률이 얼마인가요?

분류 모델은 불연속적인 값을 예측합니다. 예를 들어 분류 모델은 다음과 같은 질문에 대한 답을 예측합니다.

주어진 이메일 메시지가 스팸인가요, 스팸이 아닌가요?

이 이미지가 강아지, 고양이 또는 햄스터의 이미지인가요?

$$y = w(\text{가중치의 벡터 } w) * x() + b(\text{편향})$$

그래프로 손실을 구할 수 있음!

$$L2 \text{ 손실 (제곱오차)} / \text{예측과 라벨간의 차이의 제곱} / (\text{관측}-\text{예측})^2 / (y - y')^2$$

ex) 귀뚜라미가 우는걸 데이터로 이용해 선형회귀로 나타내기!

1분당 우는 횟수 및 섭씨온도

$$y = wx + b$$

(y는 섭씨온도(예측값) / w은 선의 기울기 / x는 1분당 우는 횟수(입력 특성 값) / b는 y절편)

$$y' = b + w_1x_1 \text{ (y'는 예측된 라벨(언고자하는 출력) / b는 편향임, } w_0 \text{이라고도 함} / w_1 \text{은 특성 1의 가중치 (위에서 m으로}$$

새로운 분당 우는 횟수 x1에서 온도 y'를 추론(예측)하려면 x1 값을 이 모델에 삽입하기만 하면 됨!

아리 첨자(예:w1, x1)은 여러 특성에 의존하는 좀 더 정교한 모델을 예시! (세 가지 특성에 의존하는 모델은 다음과 같은 방정식을

모델을 학습시킨다는 것을 단순히 말하면 라벨이 있는 데이터로부터 올바른 가중치와 편향값을 학습(결정)하는 것! / 지도 학습에서

손실은 잘못된 예측에 대한 벌점! (손실은 한가지 예에서 모델의 예측이 얼마나 잘못되었는지를 나타내는 수! / 모델의 예측이 완벽하

$$\begin{aligned} &= \text{the square of the difference between the label and the prediction} \\ &= (\text{observation} - \text{prediction}(x))^2 \\ &= (y - y')^2 \end{aligned}$$

평균 제곱 오차(MSE)는 예시당 평균 제곱 손실입니다. MSE를 계산하려면 개별 예의 모든 제곱 손실을 합한 다음 예의 수로 나눕니다

$$MSE = 1/N ((x,y \Rightarrow D), (y - \text{prediction}(x))^2)$$

(x,y)는 모델이 예측하는 데 사용하는 특성 집합(온도,나이 짝짓기 성공률), y는 예의 라벨(분당 우는 소리!) , prediction(x)

손실 줄이는 방법 (가중치와 편향에 대한 도함수 $(y - y')^2$ 는 주어진 예제의 손실 변화정도를 보여줌!!)

손실을 최소화하는 방향 (작은보폭을 반복하여 취함-> 기울기 보폭 (음의 기울기 보폭) / 경사하강법이라고함!)

경사하강법의 블록 다이어그램!

특성들 -> 모델(예측함수) -> 추측 예측 -> 손실계산 -> 매개변수 업데이트 계산 -> 모델 -> 추론 반복 (라벨의 경우 바로 손실

학습률이 아주 작으면 작은 보폭으로 여러 번 이동해서 계산을 여러번 해야 극솟값에 도달함

학습률이 크면 더 큰 보폭으로 이동하게 되고 극솟값보다 더 멀리 갈 수도 있음!

(높은 차원의 모델에서는 발산이 발생할 수 있음 -> 이러면 학습률 자릿수를 줄여야함!)

블록 문제에서는 가중치가 임의의 값을 가질 수 있음 (모두 0) => 블록:그림 모양을 생각 (최소값은 단하나!)

예고 (포어셰도링): 신경망에서는 해당없음-> 블록하지않음, 계란판모양, 최소값이 둘이상 있음, 초기값에 따라 크게 달라짐!

SGD/ 미니 배치경사하강법

적은 양의 데이터 샘플에서 기울기 계산이 잘 작동함 (모든 보폭에서 새로운 무작위 샘플을 얻음)

확률적 경사하강법 (한 번에 하나의 예!) / 미니 배치 경사하강법: 10~1000개의 예로 구성된 배치 (손실과 기울기는 배치 전반에 걸쳐

반복 방식의 모델 학습

$y' = b + w_1x_1$ (b 와 w_1 의 초기값은 중요치 않음)

y' : 특성 x 에 대한 모델의 예측 값입니다.

y : 특성 x 에 대한 올바른 라벨입니다.

경사하강법 알고리즘은 시작점에서 손실 곡선의 기울기를 계산 / 편미분의 벡터 (어느 방향이 더 정확한지, 부정확한지 알려줌! /

경사하강법 알고리즘은 기울기에 학습률 또는 보폭이라 불리는 스칼라를 곱하여 다음 지점을 결정합니다. 예를 들어 기울기가 2.5이

초매개변수는 프로그래머가 머신러닝 알고리즘에서 조정하는 값입니다. 대부분의 머신러닝 프로그래머는 학습률을 미세 조정하는 데 상

반대로 학습률을 너무 크게 설정하면 양자역학 실험을 잘못된 것처럼 다음 지점이 곡선의 최저점을 무질서하게 이탈할 우려가 있습니다

모든 회귀 문제에는 골디락스 학습률이 있습니다. 골디락스 값은 손실 함수가 얼마나 평탄한지 여부와 관련 있습니다. 손실 함수의 7

경사하강법에서 배치는 단일 반복에서 기울기를 계산하는 데 사용하는 예의 총 개수

(배치가 너무 커지면 단일 반복으로도 계산하는 데 오랜 시간이 걸릴 수 있습니다)

적당한 중복성은 노이즈가 있는 기울기를 평활화하는 데 유용할 수 있지만, 배치가 거대해지면 예측성이 훨씬 높은 값이 대용량 배치

데이터 세트에서 예를 무작위로 선택하면 (노이즈는 있겠지만) 훨씬 적은 데이터 세트로 중요한 평균값을 추정할 수 있습니다. 확률적

미니 배치 확률적 경사하강법 (미니 배치 SGD)는 전체 배치 반복과 SGD 간의 절충안입니다. 미니 배치는 일반적으로 무작위로 선택한

(소규모 배치 혹은 예가 하나뿐인 배치 (SGD))

놀랍게도 소규모 배치 또는 예가 하나뿐인 배치에서 경사하강법을 수행하는 것이 일반적으로 전체 배치보다 훨씬 효율적입니다. 예시