3.1 선형 회귀[1]

회귀는 독립 변수와 종속 변수 간의 관계를 모델화하는 방법을 지칭한다. 과학에서 회귀의 목적은 대체로 입력값과 산출값 간의 관계를 특징짓는 것이다. 반면에 머신 러닝에서 회귀는 대체로 예측과 관련된다.

3.1.1

선형 회귀는 회귀의 가장 표준적인 도구이다. 이것에는 몇 가지 가정이 있다.

(1) 독립변수 x와 종속변수 y는 선형적이다: y는 x 내의 원소들의 가중합으로 표현된다.
(2) any noise is well-behaved (following a Gaussian distribution)

사례.

- 면적, 준공년도를 기반으로 집값을 추정하기

요구되는 것: 각 집들의 가격, 면적, 준공년도

=> 트레이닝 셋 내의 각 가로줄(하나의 판매에 해당하는 데이터)은 example이라고 불린다.
=> 예측하고자 하는 것(가격)은 label(target)이라고 불린다.
=> 이 예측이 기반하는 독립 변수(준공 시기, 면적)은 feature (covariate) 이라고 불린다.

3.1.1.1 선형 모델

$$\text{price} = w_{\text{area}} \cdot \text{area} + w_{\text{age}} \cdot \text{age} + b.$$

w는 가중치, area와 age는 특징이다. 가중치는 **각 특징에 대하여** 우리의 예측에 끼치는 영향을 결정하고, 편향은 모든 feature 이 0일 때 예측 값(y)이 어떤 값을 갖는지만을 말할 뿐이다. 비록 우리가 0 평의 면적을 갖거나, 준공된 지 정확히 0 년 밖에 되지 않은 집을 보지 못하더라도, 우리는 여전히 bias가 필요한데, 그 이유는 그렇지 않으면 우리는 해당 모델의 표현을 제약할 것이기 때문이다. 엄격히 말하자면, 위 식은 Input features의 아핀 변환 affine transformation이라고 여겨지는데, 이것은 <가중 편향bias에 의한 traslation(shift)>이 결합된 <가중합에 의한 features>의 선형 변환이라고 특징지어진다.

우리의 목표는 가중치 w와 bias인 b를 찾는 것이다. 선형 모델은 산출값인 예측은 입력값인 feature의 아핀 변환에 의해 결정되는 모델인데, 여기서 아핀 변환은 선택된 가중치와 바이어스에 의해 명시된다.

3.1.1.2 손실 함수

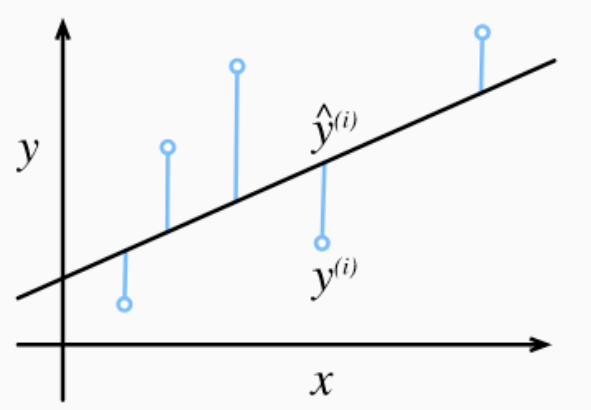
어떻게 모델을 조정fit하는지에 대해 생각하기 이전에, 우리는 fitness의 측정을 결정할 필요가 있다. 손실 함수는 target의 실제 값과 예측 값 간의 거리를 양화해준다. 손실은 보통 더 작은 값일수록 좋고, 완전한 예측은 0을 나타내는 음이 아닌 수이다. 회귀 문제에서 가장 인기 있는 손실 함수는 squared error이다.

---

[1] https://d2l.ai/chapter_linear-networks/linear-regression.html#basic-elements-of-linear-regression

$$l^{(i)}(\mathbf{w}, b) = \frac{1}{2}\left(\hat{y}^{(i)} - y^{(i)}\right)^2.$$

여기서 계수 1/2는 실제적인 차이를 만들지는 않지만 손실을 미분할 때 2* (0.5)가 계산되어 계수가 커지는 것을 상쇄해줌으로써 표기를 더 편하게 해준다.



1차원 경우에 대한 회귀 문제를 보자. 추정값 y_hat과 관찰값 y_i 간의 큰 차이는 손실에 큰 영향을 끼친다. N개의 examples의 전체 데이터 셋에서 모델의 우수성을 측정하기 위해 우리는 트레이닝 셋에서의 손실의 평균을 구하기만 하면 된다.

$$L(\mathbf{w}, b) = \frac{1}{n}\sum_{i=1}^{n} l^{(i)}(\mathbf{w}, b) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{2}\left(\mathbf{w}^\top \mathbf{x}^{(i)} + b - y^{(i)}\right)^2$$

모델을 학습시킬 때 우리는 모든 트레이닝 examples에 대한 총 손실을 최소화 해주는 매개변수 w*, b*를 찾고자 한다:

$$\mathbf{w}^*, b^* = \underset{\mathbf{w}, b}{\operatorname{argmin}}\ L(\mathbf{w}, b).$$

3.1.1.4 minibatch 스토캐스틱 경사 하강법

우리가 모델을 분석적으로 해결할 수 없을 때 조차도 우리는 최적화를 통해 실제에서 효과적으로 모델을 학습시킬 수 있다. 딥러닝 모델을 최적화하는 주요한 기법은 손실 함수가 계속 낮아지는 방향으로 모수를 업데이트 해줌으로써 오차를 반복적으로 줄여주는 것이다. 이 알고리즘을 경사 하강법이라고 부른다.

경사 하강법의 가장 나이브한 적용은 손실 함수의 미분을 취하는 것인데, 이는 데이터 셋 내의 모든 각각의 example에서 계산된 손실의 평균이다. 하지만 실제에서 이것은 매우 느리다. 우리는 단일 업데이트를 하기에 앞서, 전체 데이터셋 중 일부만을 골라내야 한다. 따라서 우리는 우리가 업데이트를 하고자 할 때 마다 examples의 랜덤 minibatch를 샘플링할 것이다. 이것이 minibatch stochastic gradient descent이다.

매 iteration 마다, 우리는 먼저 고정된 개수의 트레이닝 examples로 구성된 미니배치 B를 임의적으로 샘플링한다. 그리고 나서 우리는 모델의 매개변수에 대하여 미니배치의 평균 손실의 미분을 계산한다. 마지막으로, 우리는 이 그래디언트에 미리 결정된 양의 상수 n을 곱해주고, 현재의 매개변수 값에서 그 결과 항을 빼준다:

$$\mathbf{w} \leftarrow \mathbf{w} - \frac{\eta}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \partial_{\mathbf{w}} l^{(i)}(\mathbf{w}, b) = \mathbf{w} - \frac{\eta}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathbf{x}^{(i)} \left(\mathbf{w}^\top \mathbf{x}^{(i)} + b - y^{(i)}\right),$$
$$b \leftarrow b - \frac{\eta}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \partial_b l^{(i)}(\mathbf{w}, b) = b - \frac{\eta}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \left(\mathbf{w}^\top \mathbf{x}^{(i)} + b - y^{(i)}\right).$$

3.1.1.5 학습된 모델로 예측하기

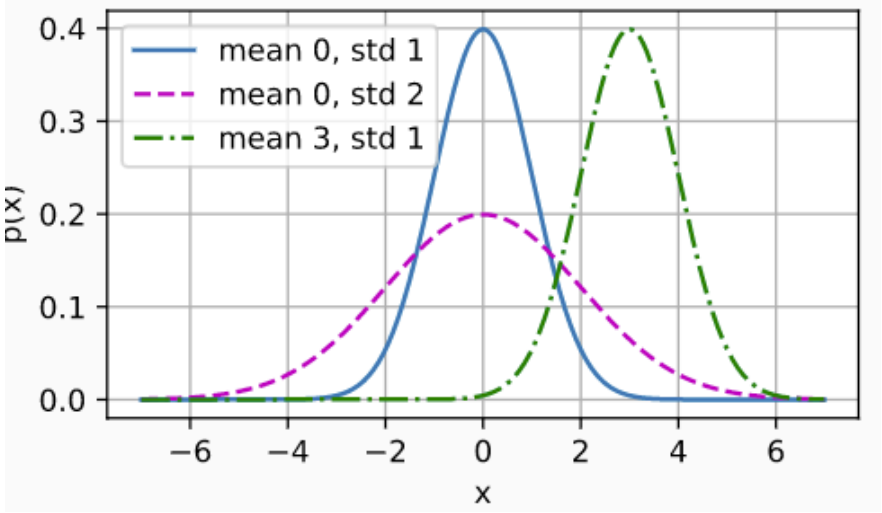
학습된 선형 회귀 모델 $\hat{w}^T x + \hat{b}$를 고려할 때, 우리는 면적 x1, 준공 시기 x2를 고려하여 (트레이닝 셋에 포함되지 않은) 새로운 집값을 추정할 수 있다. 특징을 고려하여 타겟을 추정하는 것은 소위 예측 또는 추론이라고 불린다.

3.1.3 정규 분포와 squared loss

정규 분포와 선형 회귀는 매우 밀접한 관련이 있다. 이전에 배웠던 것을 상기해보자. 정규 분포의 확률 밀도는 다음과 같았다: (u는 평균, o는 표준 편차)

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right).$$

또한, 이 정규 분포는 평균을 바꾸면 그래프가 shift 되고, 표준 편차를 바꾸면 그래프의 높이가 높아지거나 낮아진다.



4.1.1 다층 퍼셉트론 (FC-Net의 도입 배경)

우리는 앞에서 아핀 변환은 편향이 추가된 선형 변환임을 배웠다. 이 모델은 단일 아핀 변환으로 입력값을 직접 산출값에 사상해준다. 만약 우리의 라벨이 정말로 아핀 변환에 의해 입력값과 옳게 연결된다면, 이 접근은 충분하다. (하지만 실상은 그렇지 않다.) 아핀 변환에서의 선형성은 **강한 전제**일 뿐이다.

사실 선형성은 단조성에 대하여 **더 약한 전제**를 함축한다. 즉, 어떤 선형 함수에서, 특징feature의 증가는 항상 (1) (상응하는 가중치가 양수라면) 해당 모델의 산출값을 증가시키거나, (2) (상응하는 가중치가 음수라면) 그것을 감소시킨다.

예를 들자면, 어떤 개인이 대출금을 상환하는 상황이 있다고 가정해보자. 우리는 보통 수입이 더 많은 사람이 더 적은 사람 보다 대출금을 상환할 가능성이 더 높다는 것을 쉽게 받아들일 수 있다. 하지만 수입과 상환 확률이 서로 단조적인monotonic 동안에도, 이들은 항상 선형적으로 연관되는 것은 아니다. 가령, 상환금이 300 만원인 상황에서, 개인의 수입이 월 0원에서 500 만원으로 증가하는 것은 월 1억에서 1억 500 만원으로 증가하는 것 보다 상환 확률을 더 크게 증가 시킨다.

더욱이, 우리는 단조성을 위배하는 예들을 쉽게 찾아볼 수 있다. 가령, 체온이 37 도 보다 높은 사람은 그것보다 더 높은 체온을 가질 때 사망률이 높아지는 반면에, 체온이 37 도 보다 낮은 사람은 그것보다 더 낮은 체온을 가질 때 사망률이 더 낮아진다. 우리는 전처리 과정을 통해 이 문제를 해결해야 할 것이다. 즉, 우리는 37 도로부터 떨어진 거리를 특징feature로 사용할 수 있다.

그러나 이 경우에 고양이와 강아지 이미지를 분류하는 것은 어떻게 처리할 것인가? 위치 (13, 17)에서 픽셀 세기intensity의 증가는 해당 이미지가 강아지를 묘사할 가능성을 높여주는가? 선형 모델에 대한 의존성은 — 고양이와 강아지 간의 차이를 만들어주는 데 요구되는 유일한 것은 개별 픽셀의 밝기를 평가하는 것이라는 — 함축된 전제에 상응한다. 하지만 이것은 "이미지 뒤집기" 같은 변칙 경우에도 카테고리(라벨)를 보존하고자 하는 작업을 제대로 수행해내지 못한다.

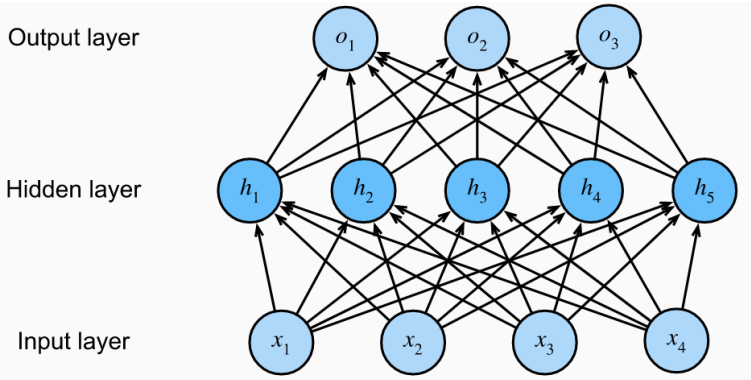


Fig. 4.1.1 An MLP with a hidden layer of 5 hidden units.

우리는 이런 선형 모델의 한계를 극복할 수 있고 더 많은 은닉층을 통합함으로써 더 일반적인 함수를 다룰 수 있다. 이것을 처리하는 가장 쉬운 방법은 **전연결 레이어**를 쌓는 것이다. 여기서 각각의 레이어는 산출값이 나올 때까지 그 위의 레이어에 연결된다feed into. 우리는 L - 1 개의 레이어를 표현representation으로 간주할 수 있고, 마지막 레이어를 선형 예측으로 간주할 수 있다. 이 아키텍처

는 소위 “다층 퍼셉트론multilayer perceptron MLP” (추가 설명을 하자면, 아래 그림처럼 FC-layer가 최소한 3 개 있어야 한다.[2])라고 불린다.

[2] https://www.quora.com/What-is-the-difference-between-an-MLP-and-a-fully-connected-layer