선형 회귀와 분류

I. 선형 적합화와 경사 하강

II. 다변수 선형 회귀

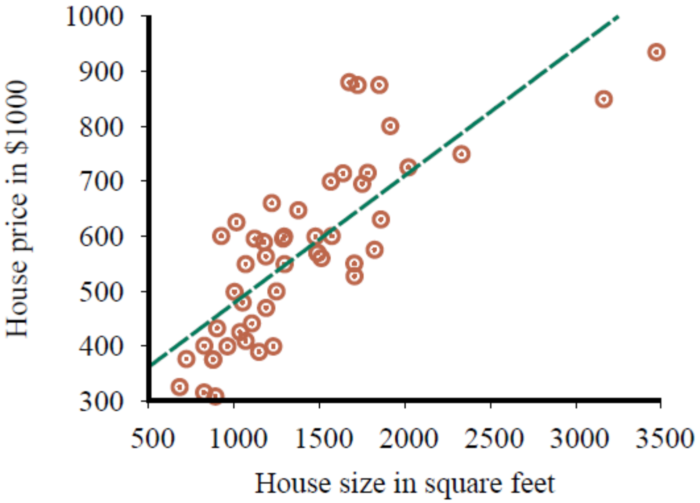
III. 선형 분류기

IV. 로지스틱 회귀

I. 선형 회귀

데이터 포인트: $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$

입력 출력



Houses for sale in Berkeley, CA

가설 공간: 일변량 선형 함수

$$h_{\mathbf{w}}(x) \equiv w_1 x + w_0$$

(w_0, w_1)

선형 회귀: 데이터에 최적 적합화되는 $h_{\mathbf{w}}$ 찾기

직선 적합화

경험적 손실을 최소화하는 가중치 (w_0, w_1) 를 탐색

모든 점에 대해 합산된 제곱 오차 손실식 $L_2(y, h_{\mathbf{w}}) = (y - h_{\mathbf{w}})^2$을 사용

$$Loss(h_{\mathbf{w}}) = \sum_{j=1}^{N} L_2(y_j, h_{\mathbf{w}}(x_j))$$

$$= \sum_{j=1}^{N} \left(y_j - h_{\mathbf{w}}(x_j)\right)^2$$

$$= \sum_{j=1}^{N} \left(y_j - (w_1 x_j + w_0)\right)^2$$

$$\mathbf{w}^* = \underset{\mathbf{w}^*}{\operatorname{argmin}}\, Loss(h_{\mathbf{w}})$$

편미분의 소실점

w의 최소점에서 $Loss(h_w)$의 기울기는 소실(=0)

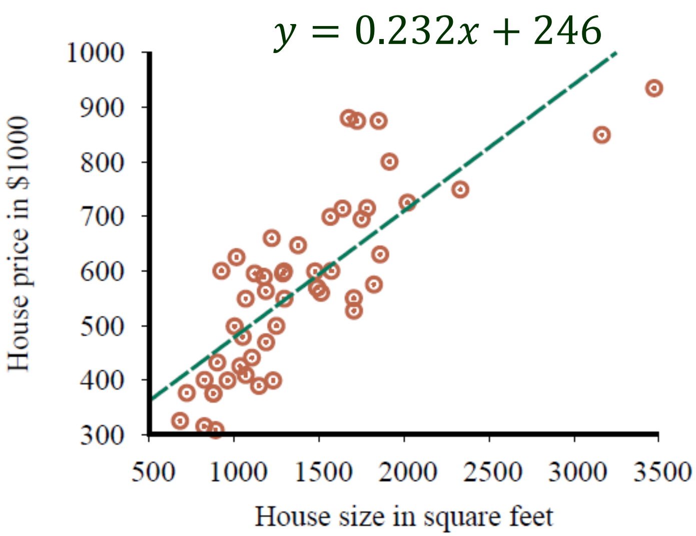
$$\nabla Loss(h_w) = \left(\frac{\partial Loss}{\partial w_0}, \frac{\partial Loss}{\partial w_1}\right) = 0$$

$$\frac{\partial Loss}{\partial w_0} = \frac{\partial}{\partial w_0}\sum_{j=1}^{N}\left(y_j - (w_1 x_j + w_0)\right)^2 = 0$$

$$\frac{\partial Loss}{\partial w_1} = \frac{\partial}{\partial w_1}\sum_{j=1}^{N}\left(y_j - (w_1 x_j + w_0)\right)^2 = 0$$

$$w_1 = \frac{N\sum_{j=1}^{N} x_j y_j - \left(\sum_{j=1}^{N} x_j\right)\cdot\left(\sum_{j=1}^{N} y_j\right)}{N\left(\sum_{j=1}^{N} x_j^2\right) - \left(\sum_{j=1}^{N} x_j\right)^2}$$

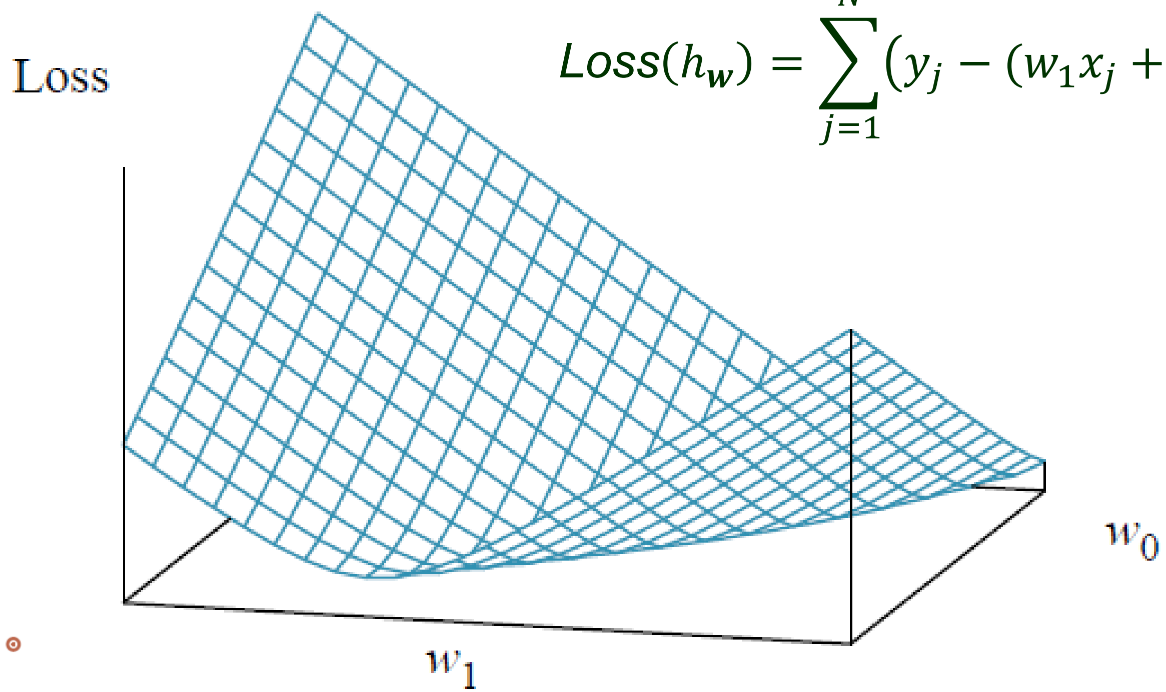
$$w_0 = \frac{1}{N}\left(\sum_{j=1}^{N} y_j - w_1\sum_{j=1}^{N} x_j\right)$$

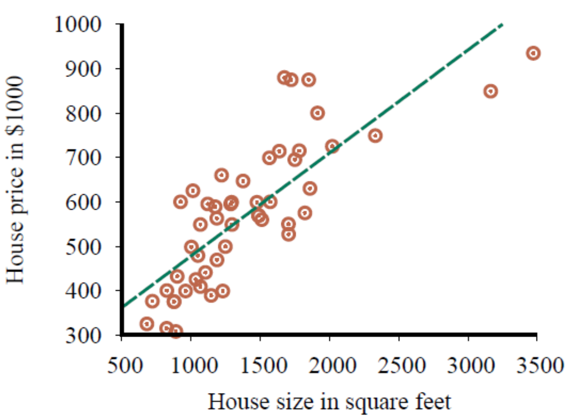


참고: 최적 적합화 선은 데이터 점들로부터 선까지의 거리 제곱의 합을 최소화하지 못함. 모델 $h_w(x) \equiv w_1 x + w_0$ 은 이미지에서 직선 모서리를 추출하는 목적으로 사용되는 일반적인 직선 방정식 $ax+by+c=0$보다 성능이 뒤떨어짐. 이는 $h_w(x) \equiv w_1 x + w_0$ 가 수직 선을 표현할 수 없기 때문이며, 기울기가 매우 커질 경우 바람직하지 않은 결과가 산출됨

손실 함수의 그래프

$$Loss(h_{\mathbf{w}}) = \sum_{j=1}^{N} \left(y_j - (w_1 x_j + w_0)\right)^2$$





♦ 로컬 최소값이 없는 컨벡스 함수

경사 하강법

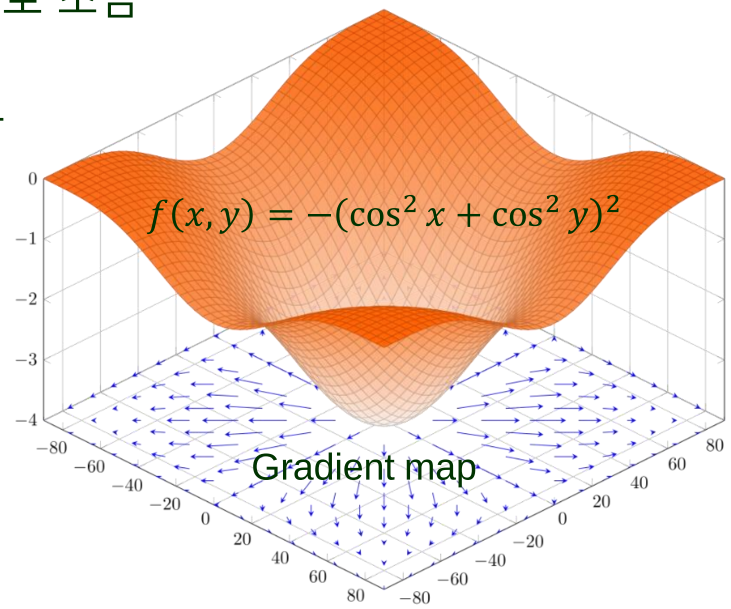
♠ 복잡한 손실 함수에서 기울기가 소실점에 다다르면 종종 닫힌 형태의 해결책이 없는 $\boldsymbol{w}$에 대한 비선형 연립 방정식 형태가 됨 // closed-form solution: // 근의 공식(?)

♦ 대신, 경사 하강법을 사용하여 해결:

- 가중치 공간의 한 점 $\boldsymbol{w}$에서 시작
- 손실 함수의 기울기 추정치 계산
- 음의 기울기 방향, 즉 가장 가파른 내리막 방향으로 조금 이동
- 손실이 (로컬) 최소인 지점에 수렴할 때까지 반복

$\boldsymbol{w} \leftarrow$ any point in the parameter space
while not converged do
    for each w_i in $\boldsymbol{w}$ do
        $$w_i \leftarrow w_i - \alpha \frac{\partial}{\partial w_i} Loss(\boldsymbol{w})$$

step size or *learning rate*



다변수 선형 회귀

- 예) n-vector $\boldsymbol{x}_j = (x_{j,1}, \ldots, x_{j,n})$
- 가설 공간:

$$h_{\boldsymbol{w}}(\boldsymbol{x}) = w_0 + w_1 x_1 + \cdots + w_n x_n = w_0 + \sum_{i=1}^{n} w_i x_i$$

편의를 위해 $\boldsymbol{x}$에 $x_0 = 1$ 을 추가하여
$\boldsymbol{x}$=(1,x_1,…,x_n)로 확장

$$h_{\boldsymbol{w}}(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x}$$

- 최적 가중치 벡터:

$$\boldsymbol{w}^* = \operatorname*{argmin}_{\boldsymbol{w}} \sum_j L_2(y_j, \boldsymbol{w} \cdot \boldsymbol{x}_j)$$

최적 가중치

- 열 벡터 $\boldsymbol{w}$, 즉, $\boldsymbol{w} = (w_0, w_1, \dots, w_n)^{\boldsymbol{T}}$
- m개의 출력으로 구성된 벡터 $\boldsymbol{y} = (y_1, y_2, \dots, y_m)^{\boldsymbol{T}}$.
- $(m \times n)$크기의 *데이터 행렬* $\boldsymbol{X} = \begin{pmatrix} \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_m \end{pmatrix}$
- 예측 출력: $\hat{\boldsymbol{y}} = \boldsymbol{X}\boldsymbol{w}$
- 전체 훈련 데이터에 대한 손실 : $L(\boldsymbol{w}) = \| \hat{\boldsymbol{y}} - \boldsymbol{y} \|^2 = \| \boldsymbol{X}\boldsymbol{w} - \boldsymbol{y} \|^2$

$$0 = \nabla L(\boldsymbol{w}) = \nabla\Big((\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})^T (\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})\Big)$$

$$\Downarrow$$

$$\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}\boldsymbol{w} - \boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{y} = 0$$

$\Downarrow$ $m \gg n$ 이기 때문에 $\boldsymbol{X}$는 거의 항상 풀랭크

$$\boldsymbol{w}^* = \boldsymbol{w} = \underbrace{\left(\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T}_{\text{pseudoinverse of } \boldsymbol{X}}\boldsymbol{y}$$

// m은 학습 데이터 레코드 수, n은 한 레코드의 필드 수. 풀랭크는 해당 행렬의 크기가 선형독립으로 구성할 수 있는 최대 수라는 의미. 역행렬 존재

정규화(Regularization)

// Normalization을 정규화로, Regularization은 정치화로 구분하는 경우도 있음

다변수 선형 함수에서 과적합을 피하기 위해 자주 사용

$$Cost(h_{\mathbf{w}}) = EmpLoss(h_{\mathbf{w}}) + \lambda Complexity(h_{\mathbf{w}})$$

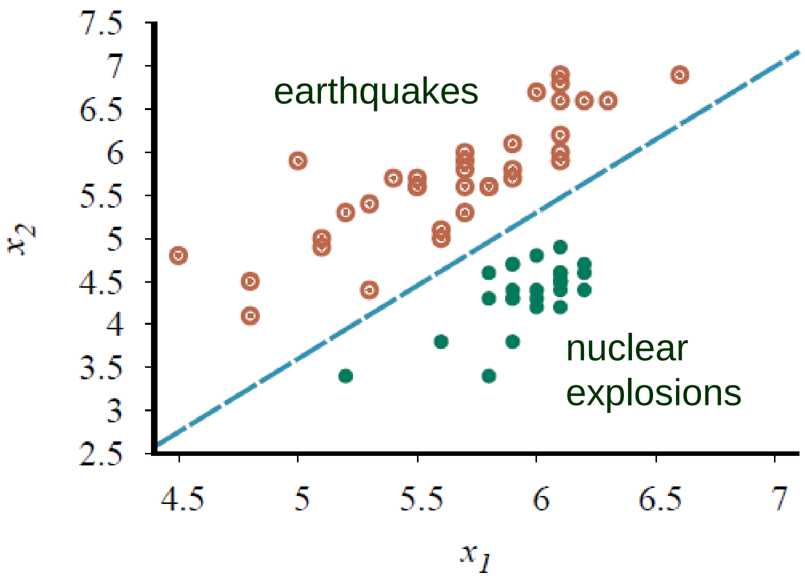
여기서,

$$Complexity(h_{\mathbf{w}}) = L_q(\mathbf{w}) = \sum_{i=1}^{n} |w_i|^q$$

♦ L_1 (q=1) 정규화는 $w_0, w_1, \ldots, w_n$ 축으로 몰아가는 경향 때문에 많은 가중치가 0으로 설정됨. 이에 따라 희소 모델을 생성하는 경향이 있음

♠ L_2 (q=2) 정규화는 차원 축에 몰리는 경향이 없음

III. 선형 분류기



지진과 핵 폭발에 의한 지진 데이터:
x_1 및 x_2는 지진 신호에서 계산된 실체파와 표면파의 크기

동일 도메인에 더 많은 데이터 포인트 추가

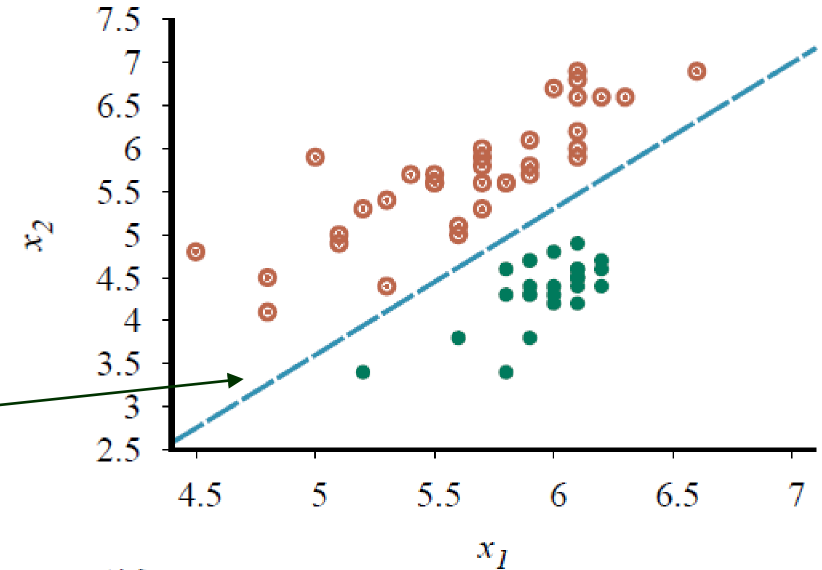
Task 새로운 (x_1, x_2) 점들을 받아들여 지진에는 0을 폭발에는 1을 반환해 줄 수 있는 가설을 학습

선형 분리자

결정 경계: 두 클래스를 분리하는 선

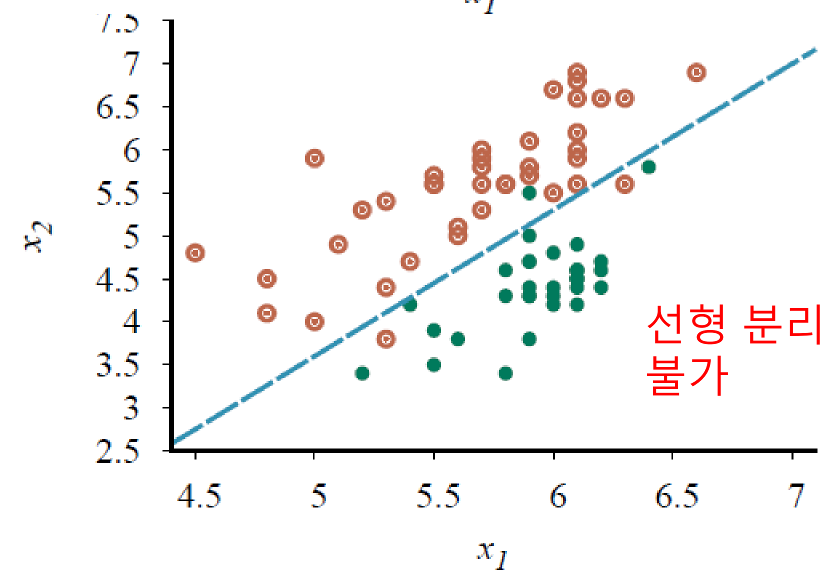
선형 분리자: 선형 결정 경계

e.g., $-4.9 + 1.7x_1 - x_2 = 0$



- $\boldsymbol{w} = (w_0, w_1, \dots, w_n)^{\boldsymbol{T}}$
- $\boldsymbol{x} = (x_0, x_1, \dots, x_n)$, $x_0 = 1$
- *분류 가설*:

$$h_{\boldsymbol{w}}(\boldsymbol{x}) = \begin{cases} 1 \text{ if } \boldsymbol{xw} \geq 0 \\ 0 \text{ if } \boldsymbol{xw} < 0 \end{cases}$$



학습 규칙

♠ 우측의 식과 같은 가설의 경우
- 기울기 $\nabla h_{\boldsymbol{w}}$ 는 0으로 소실되거나(평평한 구간)
- 미분 불가능하여 정의되지 않음(불연속 구간)

$$h_{\boldsymbol{w}}(\boldsymbol{x}) = \begin{cases} 1 \text{ if } \boldsymbol{xw} \geq 0 \\ 0 \text{ if } \boldsymbol{xw} < 0 \end{cases}$$

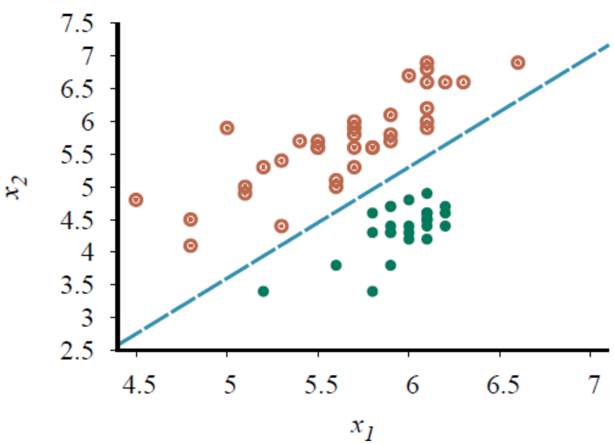
♦ *퍼셉트론 학습 규칙*(본질적으로 경사 하강법에서 차용)을 사용

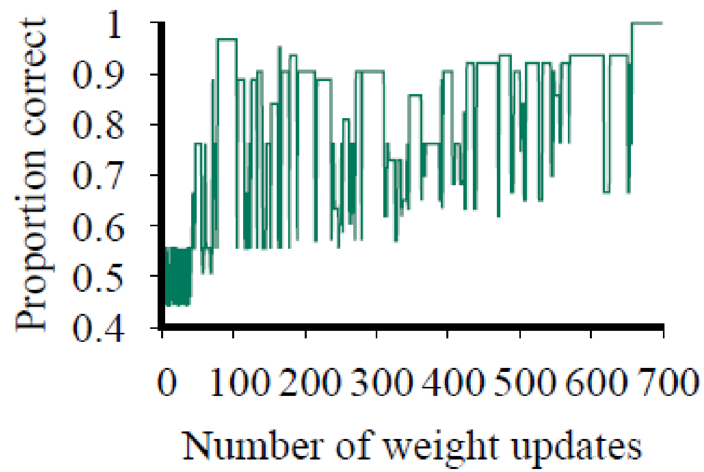
$$w_i \leftarrow w_i + \alpha(y_j - h_{\boldsymbol{w}}(\boldsymbol{x}_j))x_{j,i}$$ 단일 샘플 $(\boldsymbol{x}_j, y)$에 대하여

- $y_j = h_{\boldsymbol{w}}(\boldsymbol{x}_j)$: 정답을 출력했으므로 가중치에 변화 없음
- $y_j = 1$ 이고 $h_{\boldsymbol{w}}(\boldsymbol{x}_j) = 0$: $x_{j,i} > 0$ 일 경우 w_i 는 증가, $x_{j,i} < 0$ 일 경우 w_i 는 감소. 두 경우 모두 $\boldsymbol{xw}$는 1을 출력하려는 목적 때문에 증가함
- $y_j = 0$ 이고 $h_{\boldsymbol{w}}(\boldsymbol{x}_j) = 1$: $x_{j,i} > 0$ 일 경우 w_i 감소, $x_{j,i} < 0$ 일 경우 w_i 증가. 두 경우 모두 $\boldsymbol{xw}$는 0을 출력하려는 의도로 감소

퍼셉트론 학습을 위한 훈련 곡선

- 학습 규칙은 한 번에 하나의 샘플씩 적용
- *학습 곡선*은 주어진 학습 데이터 세트을 학습하는 과정에서 분류기의 성능이 어떻게 변화하는지를 측정하여 표시. 한번에 한 샘플씩 학습

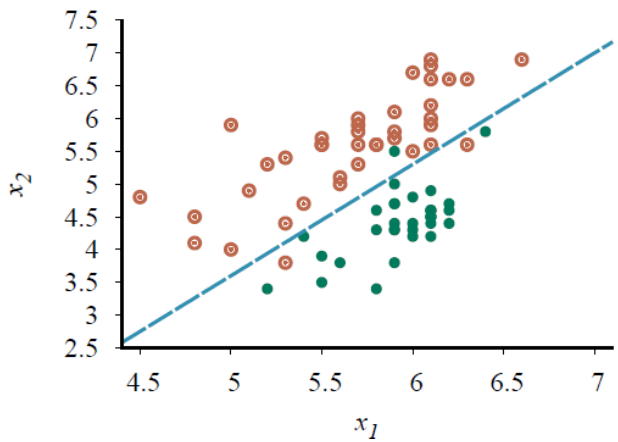




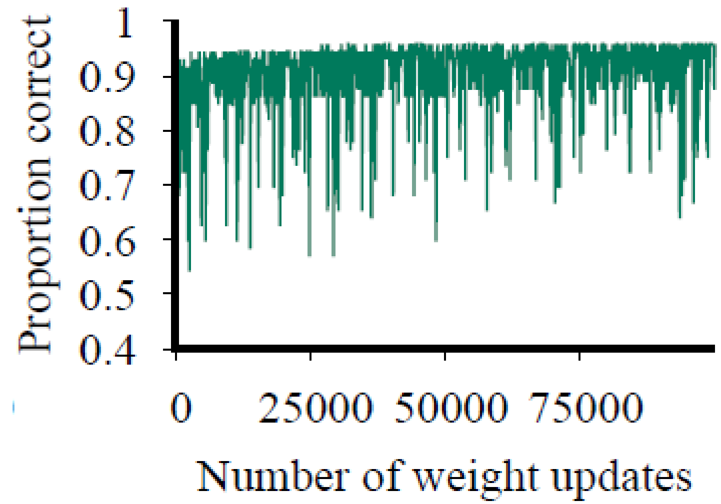
$\alpha = 1$

- 657 단계 진행 후 수렴
- 63 개의 데이터 샘플, 각 샘플은 평균 10회씩 사용됨

학습 커브 (cont'd)



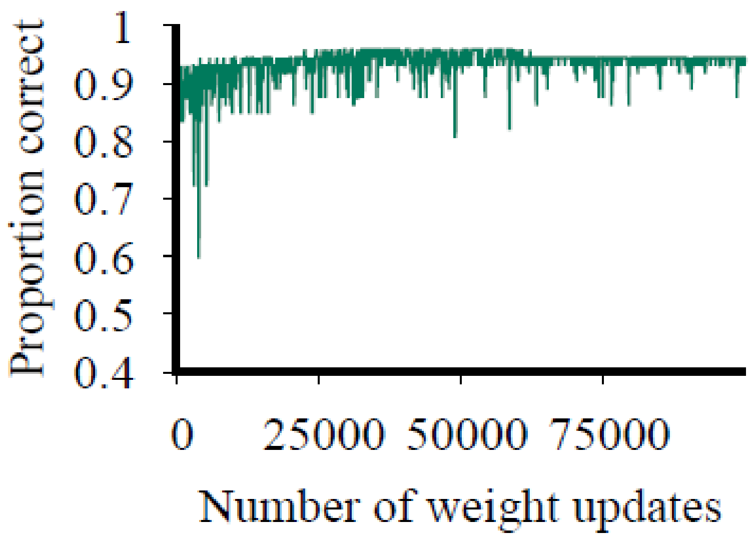
완벽한 선형 분리는 불가능



- 1만 번 단계 수행 후에도 수렴하지 않음
- α를 $O(1/t)$의 비율로 감소하도록 조치. 여기서 t는 반복 횟수

$$w_i \leftarrow w_i + \alpha(y_j - h_{\mathbf{w}}(\mathbf{x}_j))x_{j,i}$$

e.g., $\alpha(t) = 1000/(1000 + t)$



IV. 로지스틱 함수

- 앞의 사례에서 가설 함수는 연속적이지 않음 ==> 미분 불가능
- 퍼셉트론 규칙에 따른 학습 방식은 예측 가능성이 매우 낮음
- 몇몇 샘플은 불명확한 경계 상황으로 분류하는 것이 더 유리함

- 임계값을 부드럽게 하는 연속적이고 미분 가능한 함수를 사용하여 개선

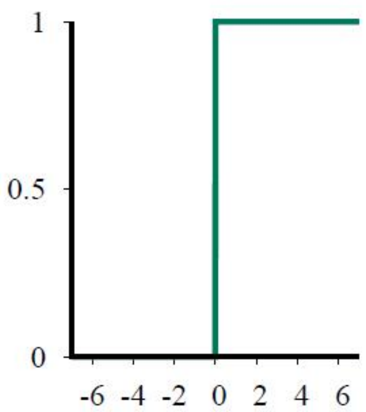
로지스틱 함수:

$$Logistics(z) = g(z) = \frac{1}{1+e^{-z}}$$

가설 함수:

$$h_{\boldsymbol{w}}(\boldsymbol{x}) = Logistics(\boldsymbol{xw}) = \frac{1}{1+e^{-\boldsymbol{xw}}}$$

$$h_{\boldsymbol{w}}(\boldsymbol{x}) = \begin{cases} 1 \text{ if } \boldsymbol{xw} \geq 0 \\ 0 \text{ if } \boldsymbol{xw} < 0 \end{cases}$$



로지스틱 회귀

주어진 데이터 세트에 대해 손실이 최소화되도록
모델 $h_{\mathbf{w}}(\mathbf{x}) = Logistics(\mathbf{xw})$ 적합화

경사 하강법 동일하게 적용

$$Logistics(z) = g(z) = \frac{1}{1+e^{-z}}$$

$$\frac{\partial}{\partial w_i} Loss(\mathbf{w}) = \frac{\partial}{\partial w_i}\left(y - h_{\mathbf{w}}(\mathbf{x})\right)^2$$

$$\vdots$$

$$= -2(y - h_{\mathbf{w}}(\mathbf{x})) \cdot g'(\mathbf{xw}) \cdot x_i$$

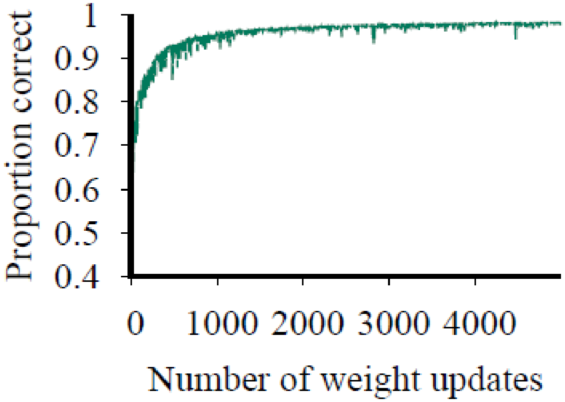
$$g'(\mathbf{xw}) = g(\mathbf{xw})(1 - g(\mathbf{xw}))$$
$$= h_{\mathbf{w}}(\mathbf{x})(1 - h_{\mathbf{w}}(\mathbf{x}))$$

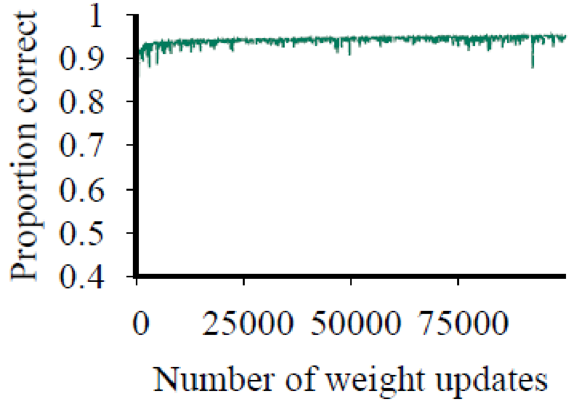
$$= -2(y - h_{\mathbf{w}}(\mathbf{x})) \cdot h_{\mathbf{w}}(\mathbf{x})(1 - h_{\mathbf{w}}(\mathbf{x})) \cdot x_i$$

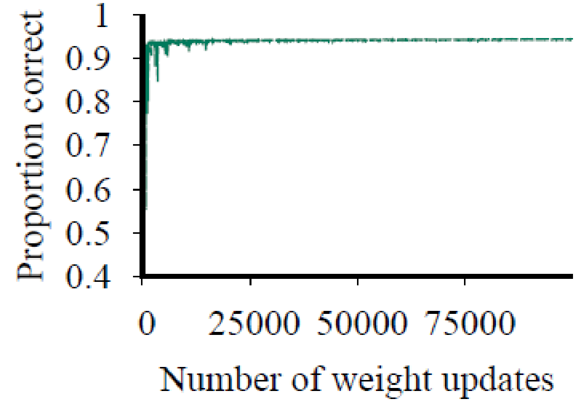
가중치 갱신:

$$w_i \leftarrow w_i + \alpha(y - h_{\mathbf{w}}(\mathbf{x})) \cdot h_{\mathbf{w}}(\mathbf{x})(1 - h_{\mathbf{w}}(\mathbf{x})) \cdot x_i$$

학습 결과 개선

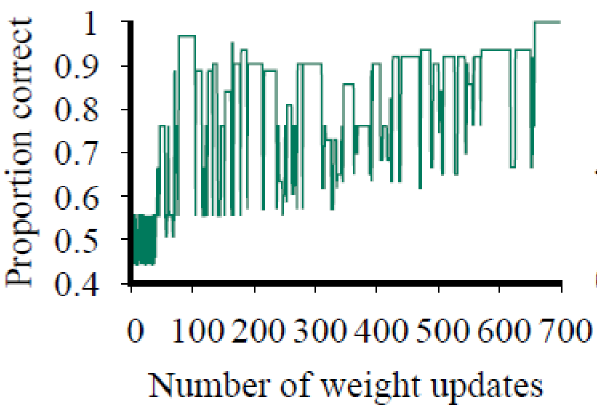




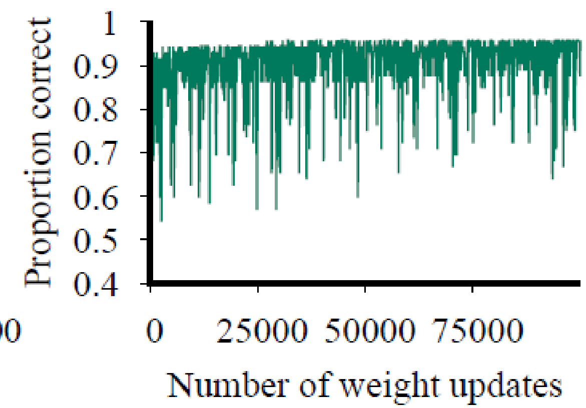


$\alpha(t) = 1000/(1000 + t)$

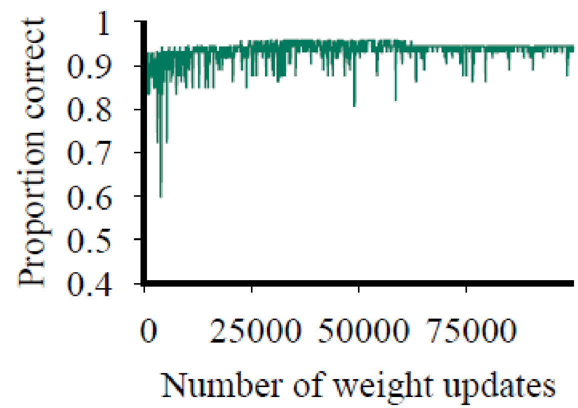
로지스틱 회귀는 훨씬 빠르고
신뢰할 수 있게 수렴



$\alpha = 1$



$\alpha = 1$



$\alpha(t) = 1000/(1000 + t)$