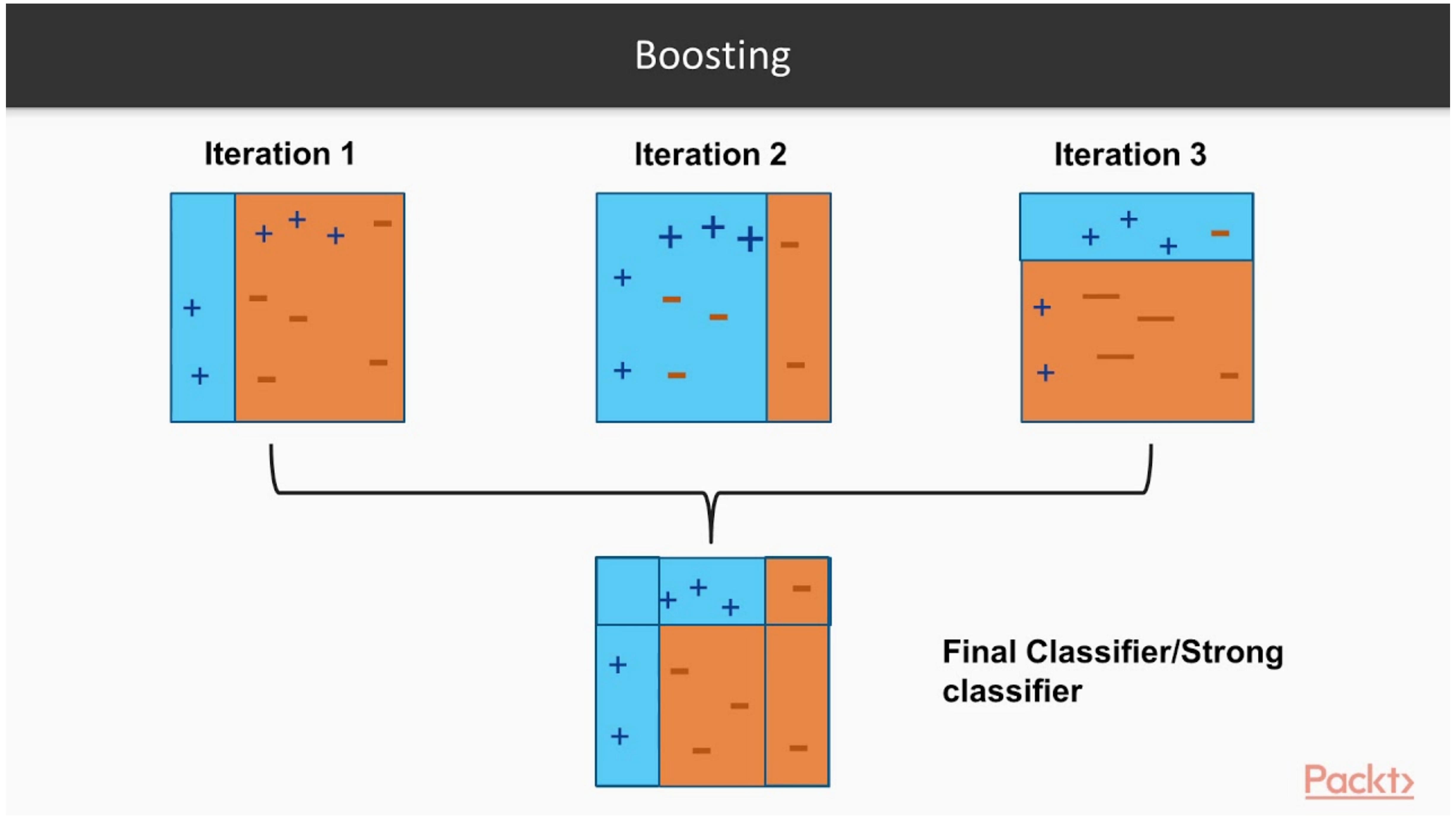
AdaBoost

데이터 분석 입문 ML1



배깅과 부스팅의 차이에서 부스팅의 특징
- 잘못된 예측에 대한 가중치
- 순차적 학습

약한 학습기 (실수를 하는 학습기들) 순차적으로 학습-예측 잘못 예측한 부분이 분명 발생
잘못 예측한 데이터에 가중치 부여를 통해 오류를 개선해 나가는 학습 방식

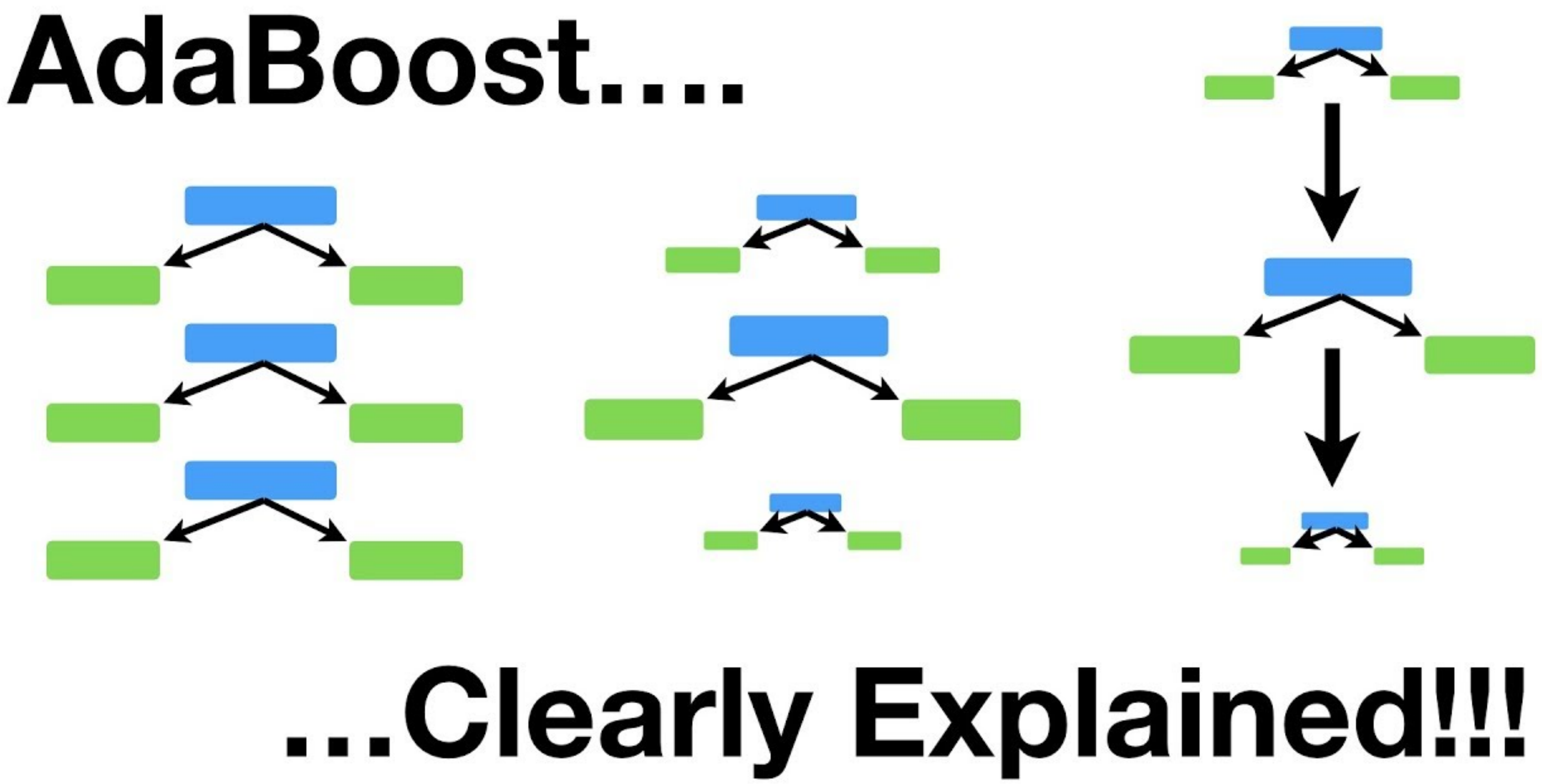
에이다 부스트는 오류 데이터에 가중치를 부여하면서 부스팅을 수행하는 대표적인 알고리즘

결국 부스팅을 수행한다는 것이 약한 학습기들이 개별적으로 성능이 뛰어나진 않지만
여러 개를 결합하면 성능이 향상될 수 있다.

Weak learner

AdaBoost 학습 단계에서 모델이 잘못 예측한 샘플에 더 큰 가중치를 부여
그래서 다음 학습 때 더 잘 학습하도록 유도

초기 시작점에는 모두 다 동일한 가중치! -> 단계가 진행될수록 잘못 예측한 샘플의 가중치를 점진적으로 증가
이전 학습기들의 잘못 예측한 부분을 보완하도록 유도한다.



Input: Dataset $D=\{(x^1,y^1),\cdots,(x^N,y^N)\}$, $y^{(i)}\in\{-1,1\}$ for $i=1,\cdots,N$

Step0: 모든 $i=1,\cdots,N$에 대해, 초기 가중치는 $w_1(i)=\frac{1}{N}$로 설정한다.

Step1: $t=1,\cdots,T$에 대해 아래 과정을 반복한다.

- 가중치 $w_t(i)$, $i=1,\cdots,N$ 에 따라 N개의 데이터를 훈련 집합 D_t로 설정한다. (*)

1. 초기화
2.반복과정
 - 가중치로 학습기를 학습시키고
- 학습기 학습은 어떤식으로 진행되는가?
- 오류율은 어떤식으로 계산되는가?
- 학습기의 가중치가 계산이 어떤식으로 진행되는가?
- 샘플가중치의 업데이트는 어떤식으로 진행되는가?
3. 학습기의 가중치 계산
4.샘플의 가중치 업데이트 (데이터의 행)
5.최종 모델 계산 진행
미리 반복된 횟수가 진행하거나, 오류율이 거의 0에 가까워 지면 학습을 중단

- D_t로 분류기 h_t를 학습한 후, D에 대한 훈련 오류율 ε_t를 계산한다.

$$\varepsilon_t=\sum_{i:h_t(x^{(i)})\neq y^{(i)}} w_t(i)$$

- 만약, $\varepsilon_t>\frac{1}{2}$이면, 루프를 중단한다.

- $\alpha_t=\frac{1}{2}\log\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$를 계산한다.

- 잘못 분류된 예제의 가중치는 증가시키고 잘 분류된 예제의 가중치는 감소시킨다.

이때, Z_t는 가중치를 정규화하기 위한 변수이다. $Z_t=\sum_{i=1}^{N} w_t(i)\exp(-\alpha_t y^{(i)}h_t(x^{(i)}))$

$$w_{t+1}(i)=\frac{w_t(i)}{Z_t}\times\begin{cases}e^{-\alpha_t}, & \text{if } y^{(i)}=h_t(x^{(i)})\\ e^{\alpha_t}, & \text{if } y^{(i)}\neq h_t(x^{(i)})\end{cases}=\frac{w_t(i)}{Z_t}\exp(-\alpha_t y^{(i)}h_t(x^{(i)}))$$

Step3: $g(x)=\sum_{i=1}^{N}\alpha_t h_t(x)$로 정의하며, 최종 분류기 $H(x)=sgn(g(x))$이다.

하나씩 이해해 보자!

AdaBoost 핵심 개념 정리

1. 오류율 계산
- 예측이 틀렸을 때는 1을 반환, 맞으면 0을 반환한다.
- wt(i) i번째 샘플의 가중치
- ht(xi) i번째 샘플에 대한 t번째 학습기의 예측값
- y(i) i번째 샘플의 실제 라벨

- 예시 x1, x2, x3
- 실제 예측값은 y1= 1, y2 =-1, y3= 1
- 초기 가중치 w1(1) = 1/3, w1(2) = 1/3,w1(3) = 1/3
- 첫 번째 학습기가 예측 h1(x1) = 1 , h1(x2) = 1, h1(x3) =-1
- 가중치 계산 : 0 + 1*1/3 + 1*1/3 = 2/3

- 오류율 계산 오류율이. 낮을수록 가중치는 커진다.
- 2/3 가중치 = 0.6667

1/2ln((1-0.6667)/0.6667) = 가중치 계산 업데이트 = 1/2ln(0.5) = -0.34 음수값이 나왔다.

가중치를 정규화 하기
a1 = -0.34
첫 번째 학습기 예측 h1(x1) = 1 , h1(x2) = 1, h1(x3) =-1
초기 가중치 w1(1) = 1/3, w1(2) = 1/3,w1(3) = 1/3

가중치 업데이트
= w2(1) = 1/3(exp(-0.34*0)=1/3 , w2(2) 1/3*(exp(-0.34*1) = w2(3) 1/3*(exp(-0.34*1) =0.23
정규화를 하기 위한 확인해 보면 => 0.333 + 0.23 + 0.23 -= 0.8
정규화를
w2(1) = 0.3/0.8= 0.41
w2(2) = 0.23/0.8= 0.3
w3(2) = 0.23/0.8= 0.3
업데이트된 가중치를 다음 학습기 사용한다.

H(x) = sign(-0.34 * h1(x))
= 최종 예측 값이 나오게 되는 것 값에 대한 예측이 결과 바뀌게 된다.

하나씩 이해해 보자!

선형 결과를 통해서 음수가 나와서 최종 결과 바뀐다.
h1(x) 선형결합

H(x) = sign(-0.34 * h1(x))
H(x) = sign(at * ht(x))

ht(x) = 예측에 대한 결과값 -1 또는 1 가진다.
at = t번째의 학습긱의 가중치 -> 이 값은 성능에 따라 달라진다. 성능이 좋은 학습기는 큰 가중치를 가질 수 있다.

H(x) = sign(-0.34 * h1(x))
-0.34 첫 번째 학습기에 대한 가중치 -> 이 가중치가 음수 h1(x) 예측 결과가 최종적으로 예측에 반대로 적용

—
가중치의 부호 : 가중치가 양수일 때 해당 학습기의 예측 결과는 긍정적인 영향을 미친다는 의미, 음수는 반대의 영향을 미친다.

최종예측 : 여러학습기의 가중치의 적용된 결과를 합산하여 -> 그 합이 양수냐 음수냐에 따라 최종 예측이 결정된다.

h(x) = 1을 예측한다면
H(x) = sign(-0.34 *1)= sign(-0.34)= -1
h(x)= -1 예측하면
H(x) = sign(-0.34*-1 = sing(0.34) = 1

하나씩 이해해 보자!

at = 1/2ln((1-et)/et)

et 약한 학습기의 ht의 오차율 -> 틀린 예측의 비율
무작위로 둘 중 하나 찍었는데 1/2 무작위 추측보다는 더 나쁜 성능을 보이면 안된다는 의미

가중치가 왜 음수로 설정될까?
학습기의 오류율이 50% 초과하나는 경우
가중치가 음수로 설정되는 이유 -> 무작위 추측보다 더 나쁜 성능을 보이게 되는 것이니 -> 해당 학습기를 반대로 고려하여서 최종 결정을 내리려고 한다.
개별 학습기가 잘못된 방향으로 학습하더라도 -> 전체적으로는 올바른 방향으로 예측할려고 이렇게 만들어 진 것이다.