7장 — 경사하강법

하나의 어려운 계산을 반복적인 쉬운 계산으로 — 학습률에서 Adam까지

조현태

2026

아주대학교 수학과

경고

수업자료에는 저작권법 제25조 제2항에 의해 수업 목적으로 이용한 저작물이 포함되어 있으므로 본 수업자료를 외부에 공개, 게시하는 것을 금지하며 이를 위반하는 경우 저작권 침해로서 관련법에 따라 처벌될 수 있습니다.

또한, 수업 내용 중 일부를 무단 캡처하여 배포, 전송할 경우 저작권 침해 또는 초상권 침해에 해당할 수 있으니 주의 바랍니다.

차례

하나의 어려운 계산 문제를 반복적인 쉬운 계산 문제로

학습률 선택의 중요성과 경사하강법의 한계

다변량으로의 확장

선형회귀에 경사하강법 적용하기

full-batch 그래디언트의 한계와 미니배치 SGD

PyTorch로 구현하는 경사하강법

적응적 최적화 — 모멘텀에서 Adam까지

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

학습목표

1. 경사하강법의 3단계 과정을 서술하고, 이를 기하적·해석적 관점에서 이해
2. 학습률에 따른 진동·발산과, 비볼록 손실에서 극소점으로 수렴할 가능성을 설명
3. 선형회귀에 경사하강법을 적용하여 파라미터 갱신 공식을 유도
4. 미니배치 확률적 경사하강법의 동작 방식과 에폭, 반복, 배치 크기의 관계를 이해
5. PyTorch에서 경사하강법 기반의 기본적인 최적화 코드를 작성
6. 모멘텀·RMSProp·Adam이 각각 어떤 기록을 쌓아 갱신량을 정하는지 비교

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

하나의 어려운 계산 문제를 반복적인 쉬운 계산 문제로

어려운 방정식 대신 쉬운 값 계산

- 선형모델의 최적해 $(\boldsymbol{w}^*, b^*)$는 총손실의 그래디언트가 0인 점
- Remark: 그래디언트가 0이 되는 점을 **임계점(critical point)**이라 부름

$$\nabla_{\boldsymbol{w},b}\,\mathcal{L}(\boldsymbol{w}^*, b^*) = \begin{pmatrix} \nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{w}^*, b^*) \\ \partial\mathcal{L}/\partial b\,(\boldsymbol{w}^*, b^*) \end{pmatrix} = \mathbf{0}$$

- 이 등식을 푸는 것보다, 임의의 점 $(\boldsymbol{w}, b)$에서 $\nabla_{\boldsymbol{w},b}\mathcal{L}(\boldsymbol{w}, b)$의 값을 계산하는 일이 훨씬 간단!(역전파)
- 경사하강법의 핵심: 쉽게 계산할 수 있는 값으로부터 임계점에 도달하기 위한 반복작업
- 표기: 모든 파라미터를 벡터 $\boldsymbol{\theta}$(예: $\boldsymbol{\theta} = (\boldsymbol{w}^\top, b)^\top$)로 표기. 우선 파라미터 하나($\theta \in \mathbb{R}$)로 시작하고 추후 벡터로 확장

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

정의 7.1 — 경사하강법과 학습률

미분가능한 손실함수 $\mathcal{L}(\theta)$와 초기값 $\theta_0 \in \mathbb{R}$이 주어졌을 때, 갱신 규칙

$$\theta_{k+1} = \theta_k - \eta \, \frac{d\mathcal{L}}{d\theta}(\theta_k), \qquad k = 0, 1, 2, \ldots$$

을 반복하여 손실을 줄여 나가려는 최적화 알고리즘을 **경사하강법(gradient descent, GD)**이라 한다.

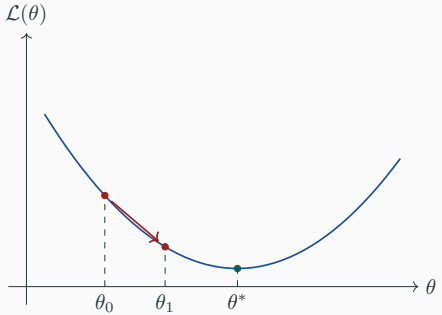
- $\eta > 0$은 **학습률(learning rate)**이라 하며, 한 번의 갱신에서 얼마나 멀리 이동할지를 조절[1].
- 아이디어의 기원: 19세기 중반 코시(Cauchy)

[1] 수치해석·최적화에서는 같은 양을 스텝 사이즈(step size)라고도 함

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

경사하강법의 기하학적 해석

- 파라미터가 하나면 그래디언트는 도함수 $\frac{d\mathcal{L}}{d\theta}$, “감소하는 방향”은 $\frac{d\mathcal{L}}{d\theta}$ 부호
- 최소점이 하나인 아래로 볼록한 $\mathcal{L}(\theta)$, 최적해 θ^*, 현재 위치 θ_0이 θ^*의 왼쪽



θ_0에서 그래프는 오른쪽으로 내려가는 중 $\Rightarrow \frac{d\mathcal{L}}{d\theta}(\theta_0) < 0$. 붉은 화살표가 한 번의 갱신 $\theta_0 \to \theta_1$

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

부호가 방향을 정한다

$\frac{d\mathcal{L}}{d\theta}(\theta_0) < 0$의 양변에 $-\eta\,(\eta > 0)$를 곱하면 부등호가 뒤집히고, θ_0을 더하면

$$-\eta \frac{d\mathcal{L}}{d\theta}(\theta_0) > 0 \quad \Longrightarrow \quad \theta_0 - \eta \frac{d\mathcal{L}}{d\theta}(\theta_0) > \theta_0$$

- 왼쪽 항을 갱신된 파라미터 $\theta_1 := \theta_0 - \eta \frac{d\mathcal{L}}{d\theta}(\theta_0)$라 하면 $\theta_1 > \theta_0$
  - θ^*를 향해 오른쪽으로 이동
- θ_0이 θ^*의 오른쪽에 있으면 도함수 부호가 바뀌어 $\theta_1 < \theta_0 \rightarrow$ 왼쪽으로 이동
- 두 경우 모두 이동 방향은 도함수 부호의 반대
- 도함수 부호가 $\mathcal{L}$의 감소 방향을 방향결정해주는것이며, 실제로 θ^*에 가까워지는지는 이동 폭 η에 의해 결정

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

알고리즘 — 경사하강법의 3단계

1. 초기화: 초기값 θ_0을 (보통 임의의 확률분포에서) 임의로 선택
2. 갱신($k = 0, 1, \ldots$): 현재 지점의 도함수 값 $\frac{d\mathcal{L}}{d\theta}(\theta_k)$를 계산하고 $\theta_{k+1} = \theta_k - \eta \frac{d\mathcal{L}}{d\theta}(\theta_k)$로 갱신. 학습률 η는 사용자가 지정
3. 반복: 정지 기준(stopping criterion)을 만족할 때까지 2단계를 반복. 대표적으로 $\left|\frac{d\mathcal{L}}{d\theta}(\theta_k)\right| \approx 0$이면 종료[2]하고, 그때의 파라미터를 최적해의 근삿값으로 채택. 슬라이드 뒷장에서 다시

- 이 기준만으로 최적해에 도달했다는 보장은 없음

[2] 딥러닝에서는 데이터셋을 기준으로 학습을 멈추기도 함. 대표적인 정지 기준은 뒷부분에서

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

해석적 확인 — 수렴값은 임계점

- 경사하강법을 통해 얻은 $\{\theta_k\}_{k=1}^{\infty}$가 일반적으로 수렴하는지 분석 자체는 쉽지 않음.
- 다만, $\{\theta_k\}$가 $\hat{\theta}$라는 특정 값을 수렴한다고 가정하면 이 값은 임계점!

짚고가기

주어진 $\eta > 0$에 대하여, 갱신 규칙

$$\theta_{k+1} = \theta_k - \eta \frac{d\mathcal{L}}{d\theta}(\theta_k)$$

에 k에 대한 극한 $\lim_{k\to\infty} \theta_k = \hat{\theta}$이 존재한다고 하자.

이 때, $\frac{d\mathcal{L}}{d\theta}$가 연속이면, 양변에 $k \to \infty$ 적용:

$$\hat{\theta} = \hat{\theta} - \eta \frac{d\mathcal{L}}{d\theta}(\hat{\theta}) \quad \Longrightarrow \quad \frac{d\mathcal{L}}{d\theta}(\hat{\theta}) = 0$$

어려운 방정식을 직접 풀지 않았는데도, 쉬운 계산의 반복이 그 방정식의 해에 도달!

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

정지 기준 — 언제 GD 반복을 멈추는가

실제로 쓰이는 대표적인 세 기준($\varepsilon > 0$은 사용자가 정하는 작은 허용치(tolerance))

1. 도함수의 크기가 작다: $\left|\frac{d\mathcal{L}}{d\theta}(\theta_k)\right| \le \varepsilon$
2. 파라미터 갱신량이 작다: $|\theta_{k+1} - \theta_k| \le \varepsilon$
3. 손실이 더 이상 줄지 않는다: $|\mathcal{L}(\theta_k) - \mathcal{L}(\theta_{k+1})| \le \varepsilon$

- 어느 기준도 만족되지 않을 때를 대비해 정해진 반복 횟수에 도달하면 멈추는 상한을 함께 둠

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

세 기준의 관계 — 사실 다 비슷한 의미

[기준 2] $\Longleftrightarrow$ [기준 1]: $\theta_{k+1} - \theta_k = -\eta \frac{d\mathcal{L}}{d\theta}(\theta_k)$ 이므로 갱신량 차이 $=$ 도함수 크기 $\times\, \eta$.

- 파라미터 값 자체가 중요하면 [기준 2]
- 임계점 자체가 중요하면 [기준 1]

[기준 1] $\Longrightarrow$ [기준 3] : 3장 선형근사 $\mathcal{L}(\theta_{k+1}) \approx \mathcal{L}(\theta_k) + \frac{d\mathcal{L}}{d\theta}(\theta_k)(\theta_{k+1} - \theta_k)$ 에 의해

$$\mathcal{L}(\theta_k) - \mathcal{L}(\theta_{k+1}) \approx \eta \Big(\frac{d\mathcal{L}}{d\theta}(\theta_k) \Big)^2$$

으로 근사 가능

- 오른쪽 미분값이 작아지면 파라미터 차이도 작아짐 ([기준 1] $\Longrightarrow$ [기준 2])

실전에서의 선택, 그리고 기준의 한계

문제마다 정하기 나름 $\rightarrow$ 절대적인 기준은 아님.

1. 파라미터 수가 적고 값 자체가 중요할 때(파라미터 추정): [기준 2]
   - 파라미터가 많아지면 [기준 2]는 벡터의 norm으로 '작다'를 결정하는 ε을 설정하기 애매함
   - 파라미터가 하나로 정해지지 않으면 (비식별가능성) 수렴 안함
2. 모델과 실제값 출력차이의 대략적 가늠: [기준 3]
   - 딥러닝(파라미터 수백만 개) $\rightarrow$ 개별 파라미터값 하나에 의의를 두지 않음
   - 선형회귀의 총손실 $=$ 개별 제곱오차의 평균이라 수치적 분석 가능
   - 값 자체만으로는 과적합등의 문제가 발생
3. 출력값 자체가 정량적이지 않을 때[3]에는 [기준 1]도 활용.

[3] 출력값의 숫자 자체가 큰 의미가 없는 경우

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

예제 7.1 — $\mathcal{L}(\theta) = \theta^2$에서의 경사하강법

주의 — 기준 만족 $\neq$ 임계점 수렴

- 앞의 세 기준은 유한 번의 반복에서 언제 멈출지 정하는 지표일 뿐.
- 예: $\mathcal{L}(\theta) = \theta^2$, $\eta = 1$, $\theta_0 = 1$이면 $\theta_k = (-1)^k$, $\mathcal{L}(\theta_k) = 1$
- 이 경우 [기준 3]은 만족되지만 도함수 크기는 항상 2이고 수열은 수렴하지 않음.
- 이 예시가 주는 추가 정보: [기준 3]이 만족한다고 일반적으로 [기준 1]이 만족하지는 않음

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

일반적인 η — 공비가 기준만족 (수렴)을 결정하는 요소 중 하나

- 앞과 동일한 예시: $\mathcal{L}(\theta) = \theta^2$은 $\theta = 0$에서 최솟값 (모른다고 가정).
- $\theta_0 = 1, \eta > 0$로 경사하강법 적용. $\frac{d\mathcal{L}}{d\theta} = 2\theta$이므로

$$\theta_{k+1} = \theta_k - \eta \cdot 2\theta_k = (1 - 2\eta)\,\theta_k$$

- 각 항이 직전 항의 일정한 배수(공비)인 수열 $=$ 등비수열.
- 이 경우 공비 $r := 1 - 2\eta$인 등비수열이므로 일반항 $\theta_k = r^k\theta_0$
- $|r| < 1$이면 $\theta_k \to 0$으로 수렴, $|r| \geq 1$이면 수렴하지 않음 (앞의 $\eta = 1$은 $r = -1$인 경우)
- 초기값 $\theta_0 = 0$이면 학습률과 관계없이 모든 항이 0

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

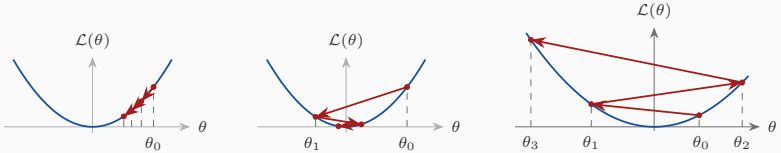
학습률 선택의 중요성과 경사하강법의 한계

주의 1

학습률의 선택이 수렴 여부와 속도를 결정

$\mathcal{L}(\theta) = \theta^2$, $\theta_0 = 1$ 예시에서 공비 $r = 1 - 2\eta$일 때,

- $\eta = 0.1$: $r = 0.8$, $\theta_k = 0.8^k$ — 단조 수렴 (그림 (a))
- $\eta = 0.75$: $r = -0.5$ — 부호를 바꾸며 진동하지만 수렴 (그림 (b))
- $\eta = 1.2$: $r = -1.4$, $|r| > 1$ — 진동하며 발산 (그림 (c))
- $\eta = 10^{-10}$: $r \approx 1$ — 수렴은 하지만 극단적으로 느림. 갱신이 되고 있는지 확인조차 어려움



(a) $\eta = 0.1$ ($r = 0.8$)
단조 수렴

(b) $\eta = 0.75$ ($r = -0.5$)
진동하며 수렴

(c) $\eta = 1.2$ ($r = -1.4$)
진동하며 발산

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

적당한 η를 미리 알 수 있는가

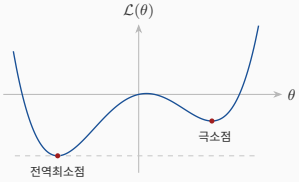
학습률 η처럼 데이터로부터 학습되지 않고 학습 절차를 제어하기 위해 따로 정해 주는 값:
하이퍼파라미터(hyperparameter)

- 손실이 아래로 볼록(예: 이차 함수)인 함수인 경우 곡률로 어느 정도 조절/분석 가능
  - 중요: 신경망의 손실함수는 일반적으로 여러 극소점을 갖는 비볼록(nonconvex) 함수
- 신경망의 손실은 곡률을 미리 알 수 없음 $\Rightarrow$ 문제에 맞는 η를 계산해 주는 일반적인 공식은 없음
- 실제 방향: 매 스텝의 η를 최적화로 구하는 고전적 방법(직선 탐색), 좌표별로 자동 조절하는 적응적 방법,
- 그 외 학습 도중 η를 점차 줄이는 스케줄 러닝도 있음.

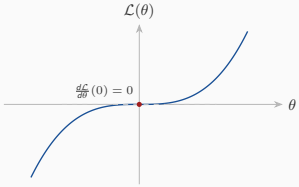
© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

주의 2 — 임계점이 곧 전역최소점은 아니다

그래디언트가 0이라는 조건 자체가 정의역 전체의 최소점인 **전역최소점(global minimum point)**만의 성질이 아님 (극소점, 안장점에서도 $\frac{d\mathcal{L}}{d\theta} = 0$ (3장))



(a) $\mathcal{L}(\theta) = \theta^4 - 2\theta^2 + 0.4\theta$: 오른쪽 골짜기 근처에서 시작해 수렴하면 극소점에 도착할 뿐 전역최소점을 놓침



(b) $\mathcal{L}(\theta) = \theta^3$: $\theta = 0$은 최소도 최대도 아님

경사하강법은 근본적으로 국소 탐색(local search) 알고리즘!

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

다변량으로의 확장

다변량으로의 확장

파라미터가 p개인 벡터 $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_p)^\top$: 도함수를 그래디언트로 바꾸어 그대로 확장

$$\boldsymbol{\theta}_{k+1} := \boldsymbol{\theta}_k - \eta \, \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_k), \qquad \nabla_{\boldsymbol{\theta}} \mathcal{L} = \begin{pmatrix} \partial \mathcal{L}/\partial \theta_1 \\ \vdots \\ \partial \mathcal{L}/\partial \theta_p \end{pmatrix}$$

- 기하적 근거: 미적분(3장) 내용대로 $-\nabla_{\boldsymbol{\theta}} \mathcal{L}$는 손실이 가장 빠르게 감소하는 최급강하 방향
- 해석적 확인: $\boldsymbol{\theta}_k$가 $\hat{\boldsymbol{\theta}}$로 수렴한다면, 마찬가지로 $\nabla_{\boldsymbol{\theta}} \mathcal{L}(\hat{\boldsymbol{\theta}}) = \mathbf{0}$를 만족
- 미분 변수가 $\boldsymbol{\theta}$만 있는 경우, 편의상 아래첨자를 생략하여 $\nabla \mathcal{L}$로 표기

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

직선 탐색: 위치에 따라 학습률을 결정하기

- "η 선택을 위한 일반적인 공식은 없음"이 "수학적으로 아무것도 할 수 없다"는 뜻은 아님
- η를 미리 정하지 않고 미지수로 남기면 다음 위치 $\boldsymbol{\theta}_{k+1}$는 η의 함수

$$\boldsymbol{\theta}_{k+1}(\eta) = \boldsymbol{\theta}_k - \eta\,\nabla\mathcal{L}(\boldsymbol{\theta}_k), \qquad F(\eta) := \mathcal{L}\big(\boldsymbol{\theta}_{k+1}(\eta)\big)$$

- 학습률 선택 $=$ 일변수 함수 F의 최소화. 답은 현재 탐색 방향에서 손실을 최소화하는 η
- 매 스텝 이렇게 η를 최적화하는 전략 $=$ **직선 탐색(line search)** (정확히는 F의 최소점을 구하는 정확 직선 탐색; 실무에서는 손실이 충분히 줄어드는 η를 찾는 비정확 직선 탐색도 널리 쓰임)
- $\eta > 0$에서 최솟값이 존재하면 $F'(\eta) = 0$
  - 변수가 η 하나뿐이라 개형을 그리거나 값을 조사해서 $F(\eta)$의 최솟값을 찾아볼 수 직접 찾기 가능

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

직선 탐색의 기하

연속된 그래디언트는 수직

연쇄법칙으로 F를 미분. 안쪽 함수의 미분은 $\frac{d\boldsymbol{\theta}_{k+1}}{d\eta} = -\nabla\mathcal{L}(\boldsymbol{\theta}_k)$이므로

$$F'(\eta) = \nabla\mathcal{L}\big(\boldsymbol{\theta}_{k+1}(\eta)\big)^\top \frac{d\boldsymbol{\theta}_{k+1}}{d\eta} = -\nabla\mathcal{L}\big(\boldsymbol{\theta}_{k+1}(\eta)\big)^\top \nabla\mathcal{L}(\boldsymbol{\theta}_k)$$

- F가 $\eta^\star > 0$에서 최소점을 가지면 $\longrightarrow F'(\eta^\star) = 0$
- 동일하게, $\nabla\mathcal{L}\big(\boldsymbol{\theta}_{k+1}(\eta^\star)\big)^\top \nabla\mathcal{L}(\boldsymbol{\theta}_k) = 0$
- 최적의 η에서는 다음 지점의 그래디언트가 현재 지점의 그래디언트와 수직(직교)

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

예제 7.2 — 직선 탐색 직접 계산 (1) 최적 학습률

$\mathcal{L}(x,y) = x^2 + 3y^2$, 시작점 $(x_0, y_0) = (3, 2)$. $\nabla\mathcal{L} = (2x, 6y)^\top$ 이므로

$$x_1(\eta) = (1-2\eta)x_0, \quad y_1(\eta) = (1-6\eta)y_0, \qquad F(\eta) = (1-2\eta)^2 x_0^2 + 3(1-6\eta)^2 y_0^2$$

$F'(\eta) = -4(1-2\eta)x_0^2 - 36(1-6\eta)y_0^2 = 0$을 풀면

$$\eta^\star = \frac{x_0^2 + 9y_0^2}{2x_0^2 + 54y_0^2} = \frac{9+36}{18+216} = \frac{5}{26} \approx 0.19, \qquad (x_1, y_1) = \left(\tfrac{24}{13}, -\tfrac{4}{13}\right) \approx (1.85, -0.31)$$

예제 7.2 — 직선 탐색 직접 계산 (2) 직교성 확인

$\nabla \mathcal{L}(x_0, y_0) = (6, 12)^\top$, $\nabla \mathcal{L}(x_1, y_1) = \left(\frac{48}{13}, -\frac{24}{13}\right)^\top$ 이므로

$$(6, 12) \cdot \left(\frac{48}{13}, -\frac{24}{13}\right) = \frac{288 - 288}{13} = 0$$

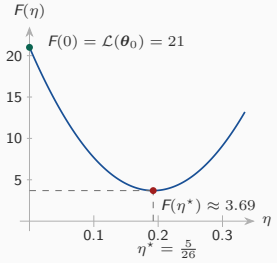
정확히 수직!

- y 좌표가 최적점(0)을 지나쳐 부호가 바뀜: 가파른 방향은 지나치고 완만한 방향은 덜 감
- 이 불균형은 방향마다 곡률이 다른 문제 — 뒤에 나올 적응적 방법에서 다시 다룸

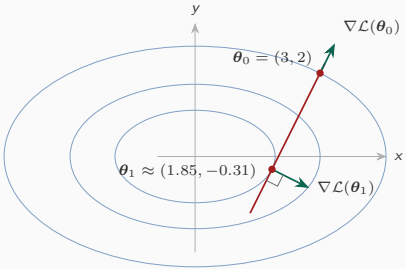
© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

예제 7.2 — 시각화

- 왼쪽: 학습률을 고르는 일이 일변수 함수 F의 최소화 문제
- 오른쪽: 두 그래디언트가 수직이며 탐색 직선은 도착점의 등고선에 접함



(a) 학습률에 따른 손실 $F(\eta)$



(b) 등고선 $x^2 + 3y^2 = c$ 위의 한 스텝

직선 탐색이 만능인가?

- 고전 최적화에서 학습률을 고르는 방법
- $F'(\eta) = 0$을 풀 수 있을때 유용
- 그러나 신경망의 손실은 $F'(\eta) = 0$을 만족하는 η의 공식이 없음 $\rightarrow$ 여러 후보 η를 계산해 보아야 함
- 후보 하나마다 $F(\eta) =$ 데이터 전체에 대한 순전파 한 번($+$ 만약 $F'(\eta)$까지 쓰면 역전파 한 번 더)

주의 1의 재확인

안정적인 새 위치를 얻을 수 있어 갱신 횟수 자체는 줄어듦

하지만 한 번의 갱신마다 $F'(\eta) = 0$을 만족하는 η를 찾기 위한 추가 평가 비용이 증가

- 파라미터가 수백만 수천만 개인 모델에서는 한 스텝마다 η를 정교하게 다듬기보다는, 적당히 고정한 η로 여러 번 반복하는 편이 효율적
- 실무적으로는 고정 학습률 · 스케줄러 · 적응적 방법(7절)으로 우회

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

선형회귀에 경사하강법 적용하기

선형회귀 문제의 설정

6장의 선형회귀로 복귀. 데이터 $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, 모델 $\hat{y} = \boldsymbol{w}^\top \boldsymbol{x} + b$, 특성 d개($\boldsymbol{x} \in \mathbb{R}^d$, $\boldsymbol{w} \in \mathbb{R}^d$, $b \in \mathbb{R}$), 파라미터 $\boldsymbol{\theta} = (\boldsymbol{w}^\top, b)^\top$

$$\mathcal{L}(\boldsymbol{w}, b) = \frac{1}{n} \sum_{i=1}^{n} \left(\boldsymbol{w}^\top \boldsymbol{x}_i + b - y_i \right)^2$$

- 목표: $\mathcal{L}$를 최소화하는 $\boldsymbol{\theta}^* = ((\boldsymbol{w}^*)^\top, b^*)^\top$
- 경사하강법을 적용하려면 $\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})$의 계산이 필요

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

그래디언트 유도

w_j에 대한 편미분을 직접 전개

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial w_j}(\boldsymbol{w}, b) &= \frac{1}{n}\sum_{i=1}^{n} \frac{\partial}{\partial w_j}\left(\boldsymbol{w}^\top \boldsymbol{x}_i + b - y_i\right)^2 \\ &= \frac{2}{n}\sum_{i=1}^{n}\left(\boldsymbol{w}^\top \boldsymbol{x}_i + b - y_i\right) x_{ij}.\end{aligned}$$

- $j = 1, \ldots, d$를 벡터로 모으면 $\nabla_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, b) = \frac{2}{n}\mathbf{X}^\top(\mathbf{X}\boldsymbol{w} + b\mathbf{1} - \boldsymbol{y})$
- 같은 방식으로 b에 대해

$$\frac{\partial \mathcal{L}}{\partial b}(\boldsymbol{w}, b) = \frac{2}{n}\sum_{i=1}^{n}\left(\boldsymbol{w}^\top \boldsymbol{x}_i + b - y_i\right) = \frac{2}{n}\mathbf{1}^\top\left(\mathbf{X}\boldsymbol{w} + b\mathbf{1} - \boldsymbol{y}\right)$$

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

정리 7.1 — 선형회귀 총손실의 그래디언트

다음을 가정하자.

1. 표본 수는 $n \geq 1$이다.
2. 설계행렬은 $\mathbf{X} \in \mathbb{R}^{n \times d}$이고 목표값은 $\boldsymbol{y} \in \mathbb{R}^{n}$이다.

모델 $\hat{\boldsymbol{y}} = \mathbf{X}\boldsymbol{w} + b\mathbf{1}$에 대한 총손실 $\mathcal{L}(\boldsymbol{w}, b) = \frac{1}{n}\|\mathbf{X}\boldsymbol{w} + b\mathbf{1} - \boldsymbol{y}\|_2^2$의 그래디언트는

$$\nabla_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, b) = \frac{2}{n}\mathbf{X}^{\top}\big(\mathbf{X}\boldsymbol{w} + b\mathbf{1} - \boldsymbol{y}\big), \qquad \frac{\partial \mathcal{L}}{\partial b}(\boldsymbol{w}, b) = \frac{2}{n}\mathbf{1}^{\top}\big(\mathbf{X}\boldsymbol{w} + b\mathbf{1} - \boldsymbol{y}\big)$$

이다.

- 구조: 잔차 $\mathbf{X}\boldsymbol{w} + b\mathbf{1} - \boldsymbol{y}$를 먼저 구하고 $\mathbf{X}^{\top}$를 곱하는 두 단계 $\Rightarrow$ 계산량 $\approx nd$
- 정규방정식 ($\approx nd^2 + d^3$) 대비 빠름.

자동미분(4장), 신경망(9장)과의 연계

왜 그래디언트 공식을 직접 유도하는가?

$\mathcal{L}(\boldsymbol{w}, b)$에 자동미분을 쓰면 공식 없이도 $\nabla_{\boldsymbol{\theta}}\mathcal{L}$ 값을 얻을 수 있음. 그럼에도 직접 유도하면 이점이 있음

1. 공식으로 계산의 구조를 파악 — 잔차 계산 후 $\mathbf{X}^\top$ 곱, 계산량 nd 분석 가능
2. 순전파 스텝으로 얻은 변수만으로 그래디언트 계산이 가능
   - 사실상 그래디언트 계산에 직접 미분이 필요한 태스크가 없거나 줄어듦

신경망에서도 마찬가지

모델이 복잡해지면 직접 유도는 현실적이지 않음 $\Rightarrow$ 자동미분에 맡김.

다만 신경망의 경우 선형회귀와 패턴이 유사 (9장)

- 역전파 공식(9장)에서 실제로 "미분"하는 것은 활성화함수의 도함수 σ' 하나
- 나머지는 이미 구한 변수들의 사칙연산

즉, 신경망에서도 손실 그래디언트 계산이 거의 사칙연산 수준으로 내려감

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

반복 갱신 — 1개의 파라미터 경우와 동일

초기값 $\boldsymbol{\theta}_0 = (\boldsymbol{w}_0^\top, b_0)^\top$을 무작위로 선택, 학습률 η(예: 10^{-3})를 정하면

$$(\boldsymbol{w}_{k+1}^\top, b_{k+1}) = (\boldsymbol{w}_k^\top, b_k) - \eta \nabla_{\boldsymbol{w},b} \mathcal{L}(\boldsymbol{w}_k^\top, b_k)$$

- 학습률이 적절히 작을 때 $(\boldsymbol{w}_k, b_k)$는 점차 정규방정식의 해 $(\boldsymbol{w}^*, b^*)$에 근접
- 정규방정식의 해가 유일하지 않으면 초기값에 따라 그중 다른 해에 이를 수 있음

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

full-batch 그래디언트의 한계와 미니배치 SGD

경사하강법의 한계

경사하강법: full-batch 그래디언트

매 갱신마다 n개 표본 모두를 사용

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^{n} \nabla_{\boldsymbol{\theta}} l_i(\boldsymbol{\theta})$$

여기서, $l_i = \left(\boldsymbol{w}^\top \boldsymbol{x}_i + b - y_i\right)^2$ 와 같이 i번째 표본 $(\boldsymbol{x}_i, y_i)$에 대응하는 손실 함수.

- 표본 수 n이 매우 크면(수백만 이상) 매 반복마다 전체 그래디언트를 계산하는 비용이 매우 큼
- 주어진 파라미터에서 갱신 방향을 하나로 고정 $\Rightarrow$ 손실 지형에 여러 극소점이 있을 때 그중 하나에 도달하면 그곳을 벗어나지 못함
- 해결: 매 반복마다 무작위로 뽑은 일부 표본만으로 그래디언트를 근사

정의 7.2 — 미니배치 확률적 경사하강법

- 미니배치(minibatch): 매 반복마다 무작위로 뽑은 일부 표본의 묶음
- 배치 크기(batch size): 미니배치에 포함된 표본 수
- 미니배치 확률적 경사하강법(minibatch stochastic gradient descent, SGD): 미니배치만으로 그래디언트를 근사하여 파라미터를 갱신하는 경사하강법
- 반복(iteration): 하나의 미니배치로 파라미터를 갱신
- 에폭(epoch): 모든 미니배치를 한 번씩 써서 전체 데이터를 사용하는 단위.
  - 1 에폭 동안 파라미터는 미니배치 수만큼 갱신됨. 에폭 수 $=$ 전체 데이터를 훑은 횟수

기타

- 배치 크기가 1인 경우를 좁은 뜻의 **SGD**
- 배치 크기가 표본 수 $n=$ 미니배치 1개 $=$ 앞서 다룬 **full-batch 경사하강법**

예제 7.4 — 6개 표본, 배치 크기 2인 경우

무작위로 섞고 번호를 다시 붙인 $n = 6$개 표본 $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_6, y_6)$, 배치 크기 2. 전체 표본을 3개의 미니배치로

$$\mathcal{B}_1 = \big\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2)\big\}, \quad \mathcal{B}_2 = \big\{(\boldsymbol{x}_3, y_3), (\boldsymbol{x}_4, y_4)\big\}, \quad \mathcal{B}_3 = \big\{(\boldsymbol{x}_5, y_5), (\boldsymbol{x}_6, y_6)\big\}$$

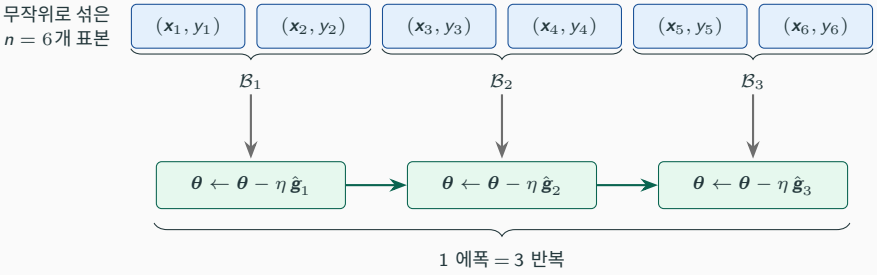
각 미니배치 $\mathcal{B}_k$에 대해 $\hat{\boldsymbol{g}}_k = \frac{1}{2} \sum_{(\boldsymbol{x}_i, y_i) \in \mathcal{B}_k} \nabla_{\boldsymbol{\theta}} l_i(\boldsymbol{\theta})$라 하면

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta\, \hat{\boldsymbol{g}}_k$$

전체 6개 표본을 한 번씩 사용하는 1 에폭 $= 3$번의 반복 $\leftarrow$: 오른쪽 값을 계산해 왼쪽 변수에 덮어쓴다는 뜻(알고리즘 서술의 표기)

예제 7.4 — 미니배치와 에폭의 구성

$\hat{\boldsymbol{g}}_k$는 배치 $\mathcal{B}_k$에서 계산한 평균 그래디언트



© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

알고리즘 — 미니배치 SGD

n개 표본의 순서를 무작위로 섞은 뒤 크기 B($B < n$)인 그룹으로 나눔(약 n/B개; 나누어떨어지지 않으면 마지막 그룹은 B보다 작음). 각 그룹 $\mathcal{B}$에 대해

1. 그 그룹의 표본만으로 미니배치 그래디언트 $\hat{\boldsymbol{g}} = \frac{1}{|\mathcal{B}|} \sum_{(\boldsymbol{x}_i, y_i) \in \mathcal{B}} \nabla_{\boldsymbol{\theta}} l_i(\boldsymbol{\theta})$를 계산
2. 파라미터 갱신 $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta\, \hat{\boldsymbol{g}}$ (1·2단계를 합쳐 1 반복)

모든 미니배치를 한 차례 순회하면(약 n/B번의 반복) 1 에폭 완료. 에폭마다 순서를 다시 섞음.

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

예제 7.5 — 용어 정리: 100개의 표본

$n = 100$개의 표본

- 배치 크기 5: $100/5 = 20$개의 그룹 $\Rightarrow$ 1 에폭 동안 20번의 반복
- 배치 크기 10: $100/10 = 10$개의 그룹 $\Rightarrow$ 1 에폭 동안 10번의 반복

작은 배치일수록 각 갱신이 더 적은 표본에 기반 $\Rightarrow$ 그래디언트 값이 **덜 정확**.

덜 정확: 무엇에 비해, 어떤 뜻으로?

우선 용어 정리부터... 표본 손실, 총손실, 실제 손실

관측 전 표본을 확률변수 $(X_i, Y_i) \overset{\text{i.i.d.}}{\sim} p$로 보면

- **표본 손실** $l(\boldsymbol{\theta}; \boldsymbol{x}, y)$: 표본 하나에서 잰 손실. $l_i = l(\boldsymbol{\theta}; \boldsymbol{x}_i, y_i)$
  - 예: 제곱손실 $\big(y - f(\boldsymbol{x}; \boldsymbol{\theta})\big)^2$
- **총손실** $\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n} l_i$: 표본 손실의 표본평균(경험 손실).
  - 지금까지 경사하강법이 최소화해 온 대상
- **실제 손실** $\mathcal{L}_{\text{true}}(\boldsymbol{\theta}) = \mathbb{E}_{(X,Y)\sim p}\big[l(\boldsymbol{\theta}; X, Y)\big]$: 표본이 나온 분포 p에 대한 표본 손실의 기댓값.
  - 실제 손실을 줄이는게 이상적이지만, p를 모르므로 계산 불가

총손실은 실제 손실의 불편추정량! $\mathbb{E}[\mathcal{L}(\boldsymbol{\theta})] = \mathcal{L}_{\text{true}}(\boldsymbol{\theta})$

그래디언트도 같은 관계 — 다음 슬라이드

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

“덜 정확”의 뜻 (1) — 추정 대상은 실제 손실의 그래디언트

결과적으로, 실제 손실 $\mathcal{L}_{\text{true}}(\theta)$ 대신 총손실 $\mathcal{L}(\theta)$를 최소화하는 파라미터를 찾고 있었던것임.
그래디언트도 마찬가지.

- $\boldsymbol{g}_{\text{true}} := \nabla_\theta \mathcal{L}_{\text{true}}(\theta) = \mathbb{E}_{(X,Y)\sim p}\left[\nabla_\theta l(\theta; X, Y)\right]$
  - 파라미터 갱신에 쓰고 싶은 이상적인 방향. 그러나 p를 모르므로 계산 불가능
- 표본 하나의 그래디언트: $\boldsymbol{g}_i := \nabla_\theta l(\theta; X_i, Y_i)$
  - $\nabla_\theta l(\theta; X, Y)$의 i번째 관측. 같은 분포에서 왔으므로 $\mathbb{E}[\boldsymbol{g}_i] = \boldsymbol{g}_{\text{true}}$
- 총손실의 그래디언트(full-batch): $\bar{\boldsymbol{g}}_n := \nabla_\theta \mathcal{L}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{g}_i$
  - $\boldsymbol{g}_1, \ldots, \boldsymbol{g}_n$의 표본평균 $\Rightarrow$ 통계량(표본으로 계산 가능)
- $\mathbb{E}[\bar{\boldsymbol{g}}_n] = \frac{1}{n}\sum_i \mathbb{E}[\boldsymbol{g}_i] = \boldsymbol{g}_{\text{true}} \Rightarrow$ 총손실 그래디언트는 $\boldsymbol{g}_{\text{true}}$의 **불편추정량**

모집단의 양 $\boldsymbol{g}_{\text{true}}$(모름) $\longleftarrow$ 추정량 $\bar{\boldsymbol{g}}_n$(표본으로 계산).

실제로 계산하는 full-batch 그래디언트는 추정값!

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

“덜 정확”의 뜻 (2) — 미니배치는 더 작은 표본평균

미니배치 $\mathcal{B}$: n개 표본 중 B개의 번호를 값과 무관하게 무작위로 뽑음

$\Longrightarrow$ 미니배치 그래디언트는 같은 표본평균을 더 작은 표본으로 만든 것

- 미니배치 그래디언트: $\boldsymbol{g}_{\mathcal{B}} := \frac{1}{B}\sum_{i\in\mathcal{B}} \boldsymbol{g}_i$
- 뽑힌 B개도 p의 i.i.d. 표본이므로 $\mathbb{E}[\boldsymbol{g}_{\mathcal{B}}] = \frac{1}{B}\sum_{i\in\mathcal{B}} \mathbb{E}[\boldsymbol{g}_i] = \boldsymbol{g}_{\text{true}}$
  - $\bar{\boldsymbol{g}}_n$과 마찬가지로 $\boldsymbol{g}_{\text{true}}$의 불편추정량
- 차이는 분산: $\mathrm{Var}(\nabla_{\boldsymbol{\theta}} l(\boldsymbol{\theta}; X, Y)) = \sigma_g^2$이라 하면

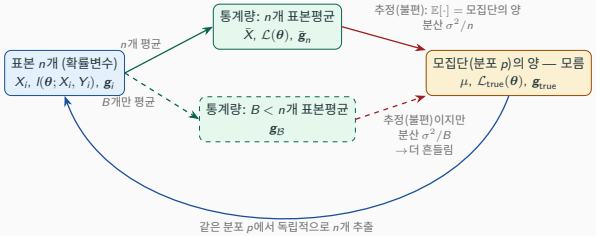
$$\mathrm{Var}(\bar{\boldsymbol{g}}_n) = \frac{\sigma_g^2}{n}, \qquad \mathrm{Var}(\boldsymbol{g}_{\mathcal{B}}) = \frac{\sigma_g^2}{B}$$

- $B < n \Rightarrow \mathrm{Var}(\boldsymbol{g}_{\mathcal{B}}) > \mathrm{Var}(\bar{\boldsymbol{g}}_n)$
  - “덜 정확” = 미니배치 쪽이 더 흔들림.
  - “확률적” = 어떤 B개가 뽑히느냐에 따라 추정값이 달라짐

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

“덜 정확”의 뜻 (3) — 5장의 표본평균과 같은 구조

5장 (모집단과 표본)	손실	그래디언트
확률변수 X_i	표본 손실 $l(\boldsymbol{\theta}; X_i, Y_i)$	$\boldsymbol{g}_i = \nabla_{\boldsymbol{\theta}} l(\boldsymbol{\theta}; X_i, Y_i)$
표본평균 $\bar{X}$ (통계량)	총손실 $\mathcal{L}(\boldsymbol{\theta})$	$\bar{\boldsymbol{g}}_n = \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})$, 미니배치 $\boldsymbol{g}_{\mathcal{B}}$
모평균 μ (모집단의 양)	실제 손실 $\mathcal{L}_{\text{true}}(\boldsymbol{\theta})$	$\boldsymbol{g}_{\text{true}} = \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{true}}(\boldsymbol{\theta})$
$\mathbb{E}[\bar{X}] = \mu$, $\mathrm{Var}(\bar{X}) = \sigma^2/n$	$\mathbb{E}[\mathcal{L}] = \mathcal{L}_{\text{true}}$	$\mathbb{E}[\bar{\boldsymbol{g}}_n] = \mathbb{E}[\boldsymbol{g}_{\mathcal{B}}] = \boldsymbol{g}_{\text{true}}$, Var: σ_g^2/n vs σ_g^2/B



full-batch든 미니배치든 모두 $\boldsymbol{g}_{\text{true}}$의 추정량!

“확률적”의 실체와 이점

엄밀한 해석 — 앞의 분산 계산이 뜻하는 것

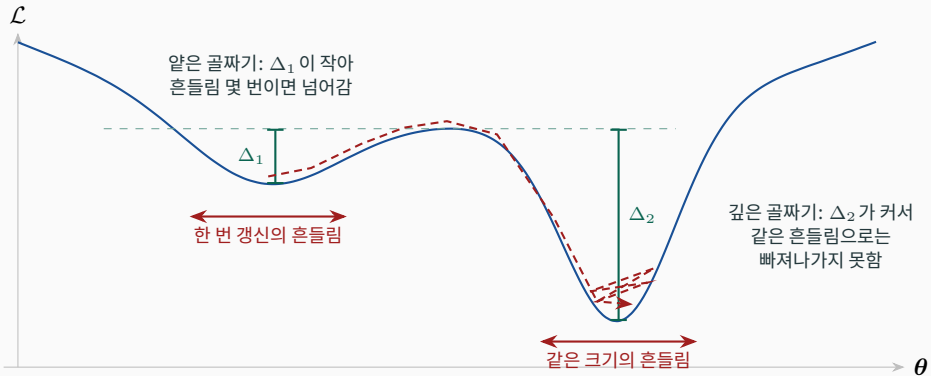
- 앞 계산은 배치 하나를 p에서 B개 복원추출했을 때의 해석
- 실제 알고리즘은 에폭마다 n개 표본을 섞은 뒤 비복원으로 나눔
  - 한 에폭 안의 배치들은 서로 겹치지 않아 독립이 아님
- 표본마다 표본 손실이 다르므로 한 에폭 내 반복시 갱신 방향도 조금씩 달라짐
  - 이것이 “확률적”의 의미

이로 인한 이점도 명확함:

- 확률적 이점: 얕은 구조의 극소점에 갇히더라도 다음 반복의 다른 방향으로 빠져나올 가능성이 생김
- 메모리 절약: full-batch를 한 번에 계산하면 n개 표본과 그 중간 계산 결과를 모두 저장해야 하지만, 미니배치는 B개분만 있으면 됨
- 폭넓은 탐색: 한 에폭에 약 n/B번 갱신 — 그럴듯한 방향으로 여러 번 탐색

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

얕은 극소점과 깊은 극소점 — 같은 크기의 흔들림, 다른 결과



- “얕다/깊다”는 극소점을 벗어나는 데 필요한 손실 증가분 Δ가 갱신의 흔들림에 비해 작은가 큰가를 뜻하는 비공식 표현
- Full-batch 경사하강법은 흔들림이 없으므로 Δ_1 이 아무리 작아도 왼쪽 골짜기에 그대로 머묾

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

PyTorch로 구현하는 경사하강법

PyTorch — 데이터와 파라미터 준비

셀 1은 데이터를 만들고, 셀 2는 파라미터와 옵티마이저를 초기화한다.

```
import torch

X = torch.randn(200, 3)
y_true = X @ torch.tensor([1.5, -2.0, 0.5]) + 3.0 + 0.1*torch.randn(200)
```

```
w = torch.randn(3, requires_grad=True)
b = torch.zeros(1, requires_grad=True)
optimizer = torch.optim.SGD([w, b], lr=1e-2)
```

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

PyTorch — 갱신과 최종 결과

셀 3은 full-batch 갱신을 반복하고, 셀 4는 마지막 파라미터의 손실을 확인한다.

```
for step in range(200):
    optimizer.zero_grad()                # 이전 반복의 그래디언트 초기화
    y_hat = X @ w + b                    # 순전파
    L = torch.mean((y_hat - y_true)**2)  # 손실 계산
    L.backward()                         # 역전파로 그래디언트 계산
    optimizer.step()                     # 갱신 규칙 실행
```

```
print("w =", w.detach().numpy(), " b =", b.item())
with torch.no_grad():
    L_final = torch.mean((X @ w + b - y_true)**2)
print("최종 손실 =", L_final.item())
```

backward와 step — 공식 없이 그래디언트를

- 4장의 자동미분을 이용하면 그래디언트 계산 자체를 공식 없이 구현 가능
- `L.backward()`: $\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_k)$를 계산(연쇄법칙을 자동으로 적용)
- `optimizer.step()`: 갱신 규칙 $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_k)$를 실행
- `optimizer.zero_grad()`: 이전 반복의 그래디언트를 지움
- 이 루프는 매 반복마다 전체 `X`를 쓰므로 미니배치가 아니라 full-batch 경사하강법

직접 유도한 그래디언트 공식 $\Longleftrightarrow$ PyTorch가 계산그래프로 자동 계산

적응적 최적화 — 모멘텀에서 Adam까지

하나의 학습률로는 부족한 두 상황

지금까지의 기본 경사하강법은 한 스텝 안에서 모든 좌표에 같은 η를 사용. 직선 탐색도 마찬가지.

- 방향마다 곡률이 크게 다르면 하나의 η로 모든 방향을 만족시킬 수 없음
- SGD에서는 그래디언트 자체가 지나치게 흔들릴 수 있음

추가로 다룰 세 알고리즘

모멘텀 방법과 RMSProp(Root Mean Squared Propagation), 그리고 둘을 결합한 Adam — 현대 딥러닝의 기본 최적화 알고리즘. 갱신식과 그 효과만 소개하며 수렴 조건은 이 장의 범위 밖

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

아이디어 — 현재 그래디언트만 쓸 이유가 있나?

- t번째 갱신에 쓰는 그래디언트를 $\boldsymbol{g}_t := \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_{t-1})$로 쓰면 경사하강법은 $\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \eta\, \boldsymbol{g}_t$: 현재 파라미터에서의 (총)손실 그래디언트 $\boldsymbol{g}_t$만 사용.
- k 스텝 전 파라미터에서의 그래디언트: $\boldsymbol{g}_{t-k}$.

아이디어: 원래 $\boldsymbol{g}_t$ 자리에 이전 갱신에서의 그래디언트 $\boldsymbol{g}_{t-1}, \boldsymbol{g}_{t-2}, \dots$ 정보를 추가하자.

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \eta \left(\boldsymbol{g}_t + \beta\, \boldsymbol{g}_{t-1} + \beta^2 \boldsymbol{g}_{t-2} + \beta^3 \boldsymbol{g}_{t-3} + \cdots + \beta^{t-1} \boldsymbol{g}_1 \right), \qquad 0 \le \beta < 1$$

- 현재 기준 k 스텝 전의 그래디언트 $\boldsymbol{g}_{t-k}$에는 β^k만큼의 가중치
- $\beta = 0$: 기존의 경사하강법
- $0 < \beta < 1$: 먼 과거의 그래디언트일수록 상대적으로 비중을 줄임.

다만 이대로 코드를 만들면 지나온 그래디언트를 모두 저장해 두어야 하니 비효율적임.

$\Rightarrow$ 점화식 형태로 바꾸자!

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

모멘텀 — 그래디언트의 관성

- 괄호 안의 가중합을 $\boldsymbol{v}_t$라 두면 $\boldsymbol{v}_t = \boldsymbol{g}_t + \beta(\boldsymbol{g}_{t-1} + \beta\boldsymbol{g}_{t-2} + \cdots) = \boldsymbol{g}_t + \beta\,\boldsymbol{v}_{t-1}$
- $\boldsymbol{v}_0 = \boldsymbol{0}$라 하면, 모든 그래디언트를 저장할 필요 없이 $\boldsymbol{v}_{t-1}$과 $\boldsymbol{g}_t$ 두 개만 저장하면 끝:

$$\boldsymbol{v}_t = \beta\,\boldsymbol{v}_{t-1} + \boldsymbol{g}_t, \qquad \boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \eta\,\boldsymbol{v}_t,\ 0 \le \beta < 1$$

- $\boldsymbol{v}_t$는 과거 그래디언트가 β배씩 감쇠하며 누적 — 파라미터가 예전에 움직였던 방향을 크게 벗어나지 않게 함.
- 관성: 물리에서 움직이는 물체가 기존 운동을 유지하려는 성질이 관성
  - 그 양이 운동량(momentum) $\Rightarrow$ **모멘텀 방법(momentum method)**이라고도 함
- $\beta = 0$이면 기존의 경사하강법
- 경사하강법 대비, 상대적으로 그래디언트 부호(방향)가 유지됨 (다음 페이지)

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

예제 7.6 — 진동 방향에서의 모멘텀

$\mathcal{L}(\theta) = 10\theta^2$, $\eta = 0.09$, $\theta_0 = 1$. 보통의 경사하강법은 공비 $1 - 20\eta = -0.8$의 등비수열

$$\theta_0 = 1,\ \theta_1 = -0.8,\ \theta_2 = 0.64,\ \theta_3 = -0.512,\ \theta_4 = 0.4096,\ \ldots$$

$\beta = 0.5$ 모멘텀: $\frac{d\mathcal{L}}{d\theta} = 20\theta$이므로 $v_t = 0.5\, v_{t-1} + 20\theta_{t-1}$, $\theta_t = \theta_{t-1} - 0.09\, v_t$, $v_0 = 0$

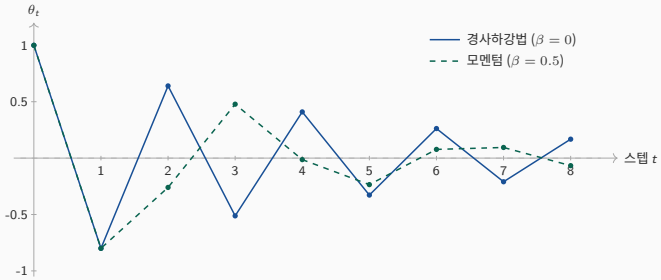
$$\begin{aligned} v_1 &= 20, & \theta_1 &= 1 - 0.09 \cdot 20 = -0.8, \\ v_2 &= 10 - 16 = -6, & \theta_2 &= -0.8 + 0.54 = -0.26, \\ v_3 &= -3 - 5.2 = -8.2, & \theta_3 &= -0.26 + 0.738 = 0.478, \\ v_4 &= -4.1 + 9.56 = 5.46, & \theta_4 &= 0.478 - 0.4914 = -0.0134 \end{aligned}$$

네 스텝 만에 크기가 0.013까지 — 이 시점의 크기는 경사하강법의 $|\theta_4| = 0.4096$보다 작음

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

예제 7.6 — 궤적 비교와 오버슈트

$t = 8$까지 늘려보면?



- v_2에서 새 그래디언트 -16이 직전 속도의 절반 10과 부호가 엇갈려 상쇄 $\Rightarrow$ θ_2의 진동 폭이 크게 감소
- 관성이 무조건 효율적이지는 않음: $v_5 = 2.462$, $\theta_5 \approx -0.235$로 크기가 다시 커짐.
  - β가 지나치게 크면 $g_5 \approx 0$이어도 오버슈트(overshoot)가 발생함.

원인 — 좌표마다 한 번 갱신에서 움직이는 정도가 다르다

모멘텀은 기존 진행 방향을 반영해 움직이지만, 얼마나 움직일지는 제어하지 않는다. 대표적인 원인은 각 좌표가 움직이는 정도를 하나의 학습률 η로만 제어하기 때문이다. 이것이 왜 문제일까?

예제

$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{2}(\theta_1^2 + 25\theta_2^2)$의 그래디언트는

$$\nabla_{\boldsymbol{\theta}}\mathcal{L}(\theta_1, \theta_2) = (\theta_1,\ 25\theta_2)$$

경사하강법 $\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \eta\,\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_{t-1}) = \boldsymbol{\theta}_{t-1} - \eta\,\boldsymbol{g}_t$을 성분별로 쓰면($\theta_{j,t}$: t번째 갱신 후의 j번째 성분)

$$\theta_{1,t} = (1-\eta)\,\theta_{1,t-1}, \qquad \theta_{2,t} = (1-25\eta)\,\theta_{2,t-1}$$

즉 두 성분은 각각 공비 $1-\eta$, $1-25\eta$의 등비수열이다. 두 수열이 수렴하려면,

$$|1-\eta| < 1 \iff 0 < \eta < 2, \qquad |1-25\eta| < 1 \iff 0 < \eta < \tfrac{2}{25} = 0.08$$

상대적으로 완만한 방향(θ_1)은 η가 커도 상관없지만, 가파른 방향(θ_2)은 η가 훨씬 작아야 한다. 하나의 η로 둘을 동시에 맞출 수 없는 이유는 그래디언트 성분의 크기 $|\theta_1|$, $|25\theta_2|$가 좌표마다 크게 다르기 때문이다.

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

아이디어 — 그래디언트 성분별 크기를 보정하기

- 학습률 η는 마찬가지로 고정.
- m개의 파라미터: $\boldsymbol{\theta} \in \mathbb{R}^m$
- t번째 갱신에서 손실함수 그래디언트 $\boldsymbol{g}_t$의 j번째 성분 $g_{t,j}$
- 경사하강법 1회 갱신 시 j번째 성분이 이동하는 거리 $d_{t,j} := \eta\,|g_{t,j}|$

아이디어: $d_{t,j}$를 t시점까지 계산한 그래디언트의 각 성분별 크기의 제곱 $g_{t,j}^2, g_{t-1,j}^2, \ldots$ 으로 보정하기

아이디어 — 각 성분을 현재까지의 그래디언트 성분별 크기로 보정한다

아이디어: $d_{t,j}$를 t시점까지 계산한 그래디언트의 각 성분별 크기의 제곱 $g_{t,j}^2, g_{t-1,j}^2, \dots$으로 보정하기

- 모멘텀과 비슷하게, 여태까지 그래디언트 값들에 가중치를 둔 합을 $s_{t,j}$로 두자
  - 현재 값 $g_{t,j}^2$에는 $1-\beta_2$를, 지금까지의 기록 $s_{t-1,j}$에는 β_2

$$s_{t,j} = \beta_2\, s_{t-1,j} + (1-\beta_2)\, g_{t,j}^2, \qquad s_{0,j} = 0, \quad 0 \le \beta_2 < 1$$

- $\sqrt{s_{t,j}}$는 현재까지 그래디언트 성분들의 제곱평균제곱근(root mean square, RMS)이다. 이 값으로 $g_{t,j}$을 나눈 **정규화한 그래디언트**로 갱신한다.

$$\theta_{j,t} = \theta_{j,t-1} - \eta\, \frac{g_{t,j}}{\sqrt{s_{t,j}}}$$

- 즉, 보정한 이동 거리 $d_{t,j} = \eta\, \frac{|g_{t,j}|}{\sqrt{s_{t,j}}}$

RMSProp — 정규화한 그래디언트로 갱신

좌표별 식을 벡터 연산으로: 성분별 제곱은 Hadamard 곱 $\boldsymbol{g}_t \odot \boldsymbol{g}_t$, 벡터의 제곱근과 나눗셈도 성분별 연산으로 정의하면

$$\boldsymbol{s}_t = \beta_2\, \boldsymbol{s}_{t-1} + (1-\beta_2)\, \boldsymbol{g}_t \odot \boldsymbol{g}_t, \qquad \boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \eta\, \frac{\boldsymbol{g}_t}{\sqrt{\boldsymbol{s}_t} + \epsilon}, \qquad \boldsymbol{s}_0 = \boldsymbol{0}$$

- 이름: **Root Mean Squared Propagation (RMSProp)** 방법
- $\epsilon > 0$(보통 10^{-8}): 0으로 나누는 것을 막는 상수
- 그래디언트가 꾸준히 컸던 성분(가파른 방향)은 작아지고, 작았던 성분(완만한 방향)은 커짐.
- RMSProp은 모멘텀과 별개의 갱신. 모멘텀은 부호(방향)를, RMSProp은 크기를 다룸 (둘을 합친 것이 Adam).

예제 7.7 — 두 파라미터에서의 정규화한 그래디언트

앞의 예와 같이 $\mathcal{L}(\theta_1, \theta_2) = \frac{1}{2}(\theta_1^2 + 25\theta_2^2)$, $\nabla_{\boldsymbol{\theta}}\mathcal{L} = (\theta_1,\ 25\theta_2)$로 주어져있다. $\boldsymbol{\theta}_0 = (-2.5,\ 0.5)$, $\eta = 0.07$, $\beta_2 = 0.9$, $s_{0,j} = 0$으로 RMSProp을 적용하면

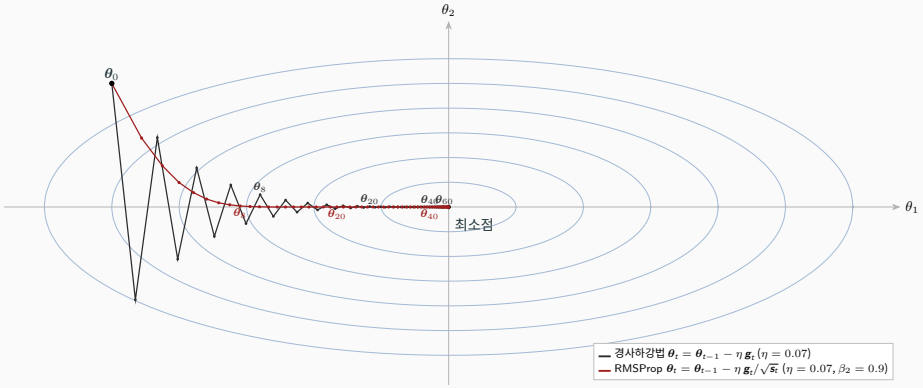
$$s_{t,j} = 0.9\, s_{t-1,j} + 0.1\, g_{t,j}^2, \qquad \theta_{j,t} = \theta_{j,t-1} - 0.07\, \frac{g_{t,j}}{\sqrt{s_{t,j}}} \qquad (j = 1, 2)$$

t	$\boldsymbol{g}_t = \nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_{t-1})$	$\boldsymbol{s}_t$	$\sqrt{\boldsymbol{s}_t}$	$\boldsymbol{g}_t / \sqrt{\boldsymbol{s}_t}$	$\boldsymbol{\theta}_t$
1	$(-2.50,\ 12.50)$	$(0.625,\ 15.625)$	$(0.79,\ 3.95)$	$(-3.16,\ 3.16)$	$(-2.279,\ 0.279)$
2	$(-2.28,\ 6.97)$	$(1.082,\ 18.915)$	$(1.04,\ 4.35)$	$(-2.19,\ 1.60)$	$(-2.125,\ 0.167)$

- 원래 그래디언트 $(-2.50,\ 12.50)$는 θ_2 성분이 5배 컸지만, 정규화한 뒤 $(-3.16,\ 3.16)$에는 두 성분의 크기가 비슷하다.

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

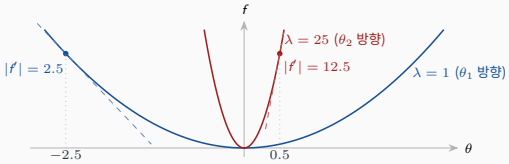
예제 7.7 — 경사하강법과 RMSProp



© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

RMSProp — 비고 (1) 곡률, 그리고 RMSProp

- 한 좌표만 보면 손실은 $f(\theta) = \frac{1}{2}\lambda\theta^2$ 형태
- 가파른 정도를 결정: 이계미분 $f''(\theta) = \lambda$
  - $= $ **곡률**
  - (예제 7.7) θ_1 방향 $\lambda = 1$, θ_2 방향 $\lambda = 25$.
- 그래디언트: $f'(\theta) = \lambda\theta$
- **그래디언트의 크기 $=$ 곡률 $\times$ 최소점까지의 거리**



좌표별 이상적인 학습률: $\eta_j \propto 1/\lambda_j$(곡률의 역수)

따라서, RMSProp은 그래디언트 크기로 나누므로 거리와 섞인 채로 추정한다.

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

RMSProp — 비고 (2) 1차 모멘트와 2차 모멘트, 그리고 $(1-\beta_2)$

그래디언트를 누적한 것을 **1차 모멘트**, 성분별 제곱을 누적한 것을 **2차 모멘트**라 부른다[4].

- 1차 모멘트(모멘텀): $\boldsymbol{v}_t = \beta\,\boldsymbol{v}_{t-1} + \boldsymbol{g}_t$.
  - 부호가 남아 있으므로 방향 정보를 포함 $\longrightarrow$ 기록들이 같은 방향이면 쌓이고 반대 방향이면 상쇄.
- 2차 모멘트(RMSProp): $\boldsymbol{s}_t = \beta_2\,\boldsymbol{s}_{t-1} + (1-\beta_2)\,\boldsymbol{g}_t \odot \boldsymbol{g}_t$.
  - 제곱이라 부호가 없고 크기 정보만 담는다.

질문: 왜 $\boldsymbol{s}_t$에는 $\boldsymbol{g}_t \odot \boldsymbol{g}_t$ 앞에 $(1-\beta_2)$를 곱하는가?

[4] 확률론의 모멘트 $\mathbb{E}[X]$, $\mathbb{E}[X^2]$와 흡사

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

비고 (2) 계속 — $(1-\beta_2)$가 없으면 어떻게 되는가

j번째 성분에서, $(1-\beta_2)$ 없이 그래디언트 제곱을 쌓는 기록을 $S_{t,j}$라 하자.

$$S_{t,j} = \beta_2\, S_{t-1,j} + g_{t,j}^2 \qquad s_{t,j} = \beta_2\, s_{t-1,j} + (1-\beta_2)\, g_{t,j}^2 \qquad S_{0,j} = s_{0,j} = 0$$

- 상황: 긴 골짜기 바닥처럼 이 좌표의 그래디언트 크기가 여러 스텝 동안 거의 같다고 하자.
- $g_{1,j}^2 = g_{2,j}^2 = \cdots = G \Longrightarrow S_{t,j}, s_{t,j}$ 모두 등비급수

$$S_{t,j} = G(1+\beta_2+\cdots+\beta_2^{t-1}) = G\,\frac{1-\beta_2^t}{1-\beta_2}, \qquad s_{t,j} = (1-\beta_2)\,G\,\frac{1-\beta_2^t}{1-\beta_2} = G(1-\beta_2^t)$$

- t가 충분히 커서 $\beta_2^t \approx 0$이 되면(정상 상태) $S_{t,j} \to G/(1-\beta_2)$, $s_{t,j} \to G$
  - 합은 실제 제곱값의 $1/(1-\beta_2)$배 증가, 평균은 실제 제곱값과 동일
- 각 값으로 파라미터를 갱신시($\theta_j \leftarrow \theta_j - \eta\, g_{t,j}/\sqrt{\cdot}$) 한 스텝의 이동 거리 ($|g_{t,j}| = \sqrt{G}$):

$$S_{t,j}: \eta\,\frac{\sqrt{G}}{\sqrt{G/(1-\beta_2)}} = \eta\sqrt{1-\beta_2}, \qquad s_{t,j}: \eta\,\frac{\sqrt{G}}{\sqrt{G}} = \eta$$

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

비고 (2) 계속 — 결론: η와 β_2의 역할을 분리한다

	$S_{t,j}$ ($(1-\beta_2)$ 없음)	$s_{t,j}$ ($(1-\beta_2)$ 있음)
정상 상태의 기록	$G/(1-\beta_2)$	G
한 스텝 이동 거리	$\eta\sqrt{1-\beta_2}$	η
$\beta_2 = 0.9$	$0.32\,\eta$	η
$\beta_2 = 0.999$	$0.03\,\eta$	η

- $S_{t,j}$로 설계시, β_2를 0.9에서 0.999로 바꾸는 순간 이동 거리가 10배 줄어든다. η와 β_2가 섞여 양쪽 다 조절해야한다.
- $(1-\beta_2)$를 곱하면 정상 상태의 이동 거리가 β_2와 무관하게 η다.
- 모멘텀의 v_t에는 왜 이 인자가 없는가? v_t도 정상 상태에서 $g/(1-\beta)$만큼 차이가 발생하지만, 이동 으로만 쓰이므로 그 배율은 학습률을 $\eta/(1-\beta)$로 보는 것과 같아 η로 조절 가능하다.

주의: 유한한 t에서는 $s_{t,j} = G(1-\beta_2^t)$라 이동 거리가 $\eta/\sqrt{1-\beta_2^t}$이다. $t=1$이면 $\eta/\sqrt{0.1} = 3.16\,\eta$

이 차이를 되돌리는 것이 편향 보정이다.

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

편향 보정 — 왜 필요한가

$\boldsymbol{s}_0 = \mathbf{0}$에서 시작한 기록은 가중치 합이 $1 - \beta_2^{\,t}$뿐이므로, t가 작을 때 실제 크기 G를 $s_{t,j} = G(1 - \beta_2^{\,t})$로 과소평가한다. 이 합으로 나누어 되돌리는 방법이 **편향 보정(bias correction)**이다.

$$\hat{\boldsymbol{s}}_t = \frac{\boldsymbol{s}_t}{1 - \beta_2^{\,t}}$$

- 그래디언트 크기가 거의 같은 구간에서는 $\hat{s}_{t,j} = \frac{G(1-\beta_2^{\,t})}{1-\beta_2^{\,t}} = G$로 정확히 복원
- t가 커지면 $\beta_2^{\,t} \to 0$이므로 $\hat{\boldsymbol{s}}_t \approx \boldsymbol{s}_t$가 되어 보정은 자연히 사라짐
  - $\beta_2 = 0.999$처럼 클수록 $1 - \beta_2^{\,t}$가 1에 가까워지는 데 오래 걸리므로($t = 100$에서도 0.095) 보정이 더 오래 필요
- (다음 주제) Adam은 1차 모멘트도 같은 형태 $\boldsymbol{v}_t = \beta_1 \boldsymbol{v}_{t-1} + (1 - \beta_1)\, \boldsymbol{g}_t$로 사용
  - 가중치 합이 $1 - \beta_1^{\,t}$이므로 2차 모멘트와 같은 이유로 $\hat{\boldsymbol{v}}_t = \boldsymbol{v}_t / (1 - \beta_1^{\,t})$로 보정

요약: 보정 전에는 초반 갱신에서 $1/\sqrt{1 - \beta_2^{\,t}}$배 왜곡 $\Longrightarrow$ 편향 보정 후에는 첫 스텝부터 이동 거리 $\approx \eta$

Adam — 모멘텀 + RMSProp

Adam(adaptive moment estimation)은 1차 모멘트(그래디언트의 평균)와 2차 모멘트(성분별 제곱의 평균)를 모두 사용하는 방법이다.

- 1차 모멘트 $\boldsymbol{v}_t$: 모멘텀처럼 방향을 누적하되, RMSProp과 같은 방식으로 현재 값에 $1-\beta_1$을 곱한 평균으로 정의.
- 2차 모멘트 $\boldsymbol{s}_t$: RMSProp의 $\boldsymbol{s}_t$와 같은 방식[5]

모멘텀이 다루는 방향(부호)과 RMSProp이 다루는 크기를 한 갱신식에 합친다.

[5](참고) 원 논문 Kingma and Ba [2015]은 1,2차 모멘트를 각각 $\boldsymbol{m}_t$, $\boldsymbol{v}_t$로 사용: Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations* (*ICLR*), 2015

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

알고리즘 — Adam의 모멘트 추정

알고리즘 — Adam (Kingma and Ba [2015][6])

조건 $\eta > 0, 0 \le \beta_1, \beta_2 < 1, \epsilon > 0$. 초기화 $\boldsymbol{v}_0 = \boldsymbol{s}_0 = \mathbf{0}$, 기본값 $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$. 매 스텝 $t = 1, 2, \dots$ 에서 $\boldsymbol{g}_t = \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_{t-1})$를 계산한 뒤

$$\boldsymbol{v}_t = \beta_1 \, \boldsymbol{v}_{t-1} + (1 - \beta_1) \, \boldsymbol{g}_t \qquad \text{(1차 모멘트)}$$

$$\boldsymbol{s}_t = \beta_2 \, \boldsymbol{s}_{t-1} + (1 - \beta_2) \, \boldsymbol{g}_t \odot \boldsymbol{g}_t \qquad \text{(2차 모멘트)}$$

두 기록은 $\mathbf{0}$에서 시작하므로 초반에는 실제 값보다 작게 잡힌다. 이를 편향 보정한 뒤 파라미터 갱신에 쓴다 (다음 페이지).

[6] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations* (*ICLR*), 2015

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

알고리즘 — Adam (계속)

앞 프레임에서 계산한 $\boldsymbol{v}_t, \boldsymbol{s}_t$를 가중치의 합 $1-\beta_1^t, 1-\beta_2^t$로 나누어 보정한다.

$$\hat{\boldsymbol{v}}_t = \frac{\boldsymbol{v}_t}{1-\beta_1^t}, \qquad \hat{\boldsymbol{s}}_t = \frac{\boldsymbol{s}_t}{1-\beta_2^t}$$

보정한 1차 모멘트가 이동 방향을, 2차 모멘트가 성분별 크기 조절을 담당한다.

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \eta \, \frac{\hat{\boldsymbol{v}}_t}{\sqrt{\hat{\boldsymbol{s}}_t} + \epsilon}$$

다음 스텝에서는 갱신한 파라미터 $\boldsymbol{\theta}_t$에서 그래디언트 $\boldsymbol{g}_{t+1}$을 다시 계산한다.

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

예제 7.8 — Adam 두 스텝 계산

$\mathcal{L}(\theta) = \theta^2$, $\theta_0 = 1$, $\eta = 0.1$ (ϵ은 무시).

- $g_t = \mathcal{L}'(\theta_{t-1}) = 2\theta_{t-1}$
- $t = 1$: $g_1 = 2$. $v_1 = 0.1 \times 2 = 0.2$, $s_1 = 0.001 \times 4 = 0.004$, $\hat{v}_1 = \dfrac{0.2}{1 - 0.9} = 2$, $\hat{s}_1 = \dfrac{0.004}{1 - 0.999} = 4$

$$\theta_1 = 1 - 0.1 \cdot \frac{2}{\sqrt{4}} = 1 - 0.1 = 0.9$$

- $t = 2$: $g_2 = 1.8$. $v_2 = 0.9 \times 0.2 + 0.1 \times 1.8 = 0.36$, $s_2 = 0.999 \times 0.004 + 0.001 \times 3.24 = 0.007236$, $\hat{v}_2 = \dfrac{0.36}{1 - 0.81} \approx 1.8947$, $\hat{s}_2 = \dfrac{0.007236}{1 - 0.998001} \approx 3.6198$

$$\theta_2 \approx 0.9 - 0.1 \cdot \frac{1.8947}{\sqrt{3.6198}} \approx 0.9 - 0.0996 = 0.8004$$

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

예제 7.8 — 이동 거리가 η로 표준화된다

- 두 스텝 각각의 이동 거리 $0.1000, 0.0996$이 거의 같다.
- $\hat{v}_t/\sqrt{\hat{s}_t}$는 분자·분모가 모두 그래디언트 크기에 비례하므로 크기가 상쇄된다. 그래디언트가 대체로 일정한 경우 Adam의 이동 거리는 η 안팎으로 매칭.
- 문제에 맞는 학습률 η 자체는 여전히 따로 정해야 한다.

세 알고리즘의 공통점

- 모두 경사하강법의 갱신식에 과거 그래디언트의 기록을 더한 변형
- 모멘텀은 과거 그래디언트를 누적해 방향을 정하고, RMSProp은 그래디언트 성분의 크기를 좌표별로 정규화
- Adam $=$ 모멘텀 $+$ RMSProp

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

주의 — Adam이 최고인가?

- Adam은 기본값만으로 넓은 문제에서 무난히 작동하여 널리 쓰이나, 만능은 아님
  - 이미지 분류 등 일부 태스크에서는 잘 조율된 모멘텀 SGD가 학습에 쓰지 않은 데이터에서의 예측 성능(일반화 성능; 10장)에서 더 좋은 결과에 이른다는 사례도 있음
- 8장 분류: 모델 설계 · 손실 정의만 바꾸고 최적화는 이 장의 갱신 규칙을 그대로 쓴다.
- 9장 이후 신경망 학습에서는 Adam을 전제로 최적화를 진행 예정.

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지

감사합니다

다음 장: 분류 — 로지스틱 회귀와 소프트맥스 회귀

© 2026 조현태 · 아주대학교 수학과 | 수강생 전용, 재배포·상업적 이용 금지