Machine Learning and Neural Networks (MATH3431) Epiphany term, 2025

Lecture notes 4: Gradient descent

Lecturer & author: Georgios P. Karagiannis georgios.karagiannis@durham.ac.uk

Aim. To introduce gradient descent, its motivation, description, practical tricks, analysis in the convex scenario, and implementation.

Reading list & references:

(1) Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding machine learning: From theory to algorithms. Cambridge university press.
- Ch. 14.1 Gradient Descent

1. Motivations

Problem 1. Consider a learning problem $(\mathcal{H}, \mathcal{Z}, \ell)$. Learning may involve the computation of the minimizer $h^* \in \mathcal{H}$, where $\mathcal{H}$ is a class of hypotheses, of the empirical risk function (ERF) $\hat{R}(h) = \frac{1}{n}\sum_{i=1}^{n} \ell(h, z_i)$ given a finite sample $\{z_i; i = 1, ..., n\}$ generated from the data generating model $g(\cdot)$ and using loss $\ell(\cdot)$; that is

$$h^* = \arg\min_h \left(\hat{R}(h)\right) = \arg\min_h \left(\frac{1}{n}\sum_{i=1}^{n} \ell(h, z_i)\right) \tag{1.1}$$

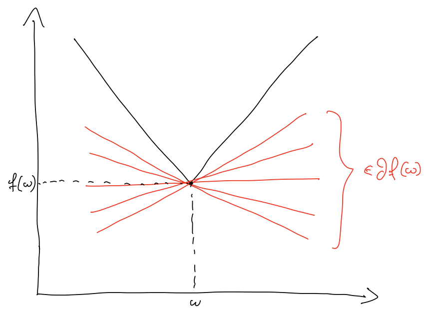
or its corresponding regularized version. If analytical minimization of (1.1) is impossible or impractical, numerical procedures can be applied; eg Gradient Descent (GD) algorithms. Such approaches introduce numerical errors in the solution.

2. subgradient

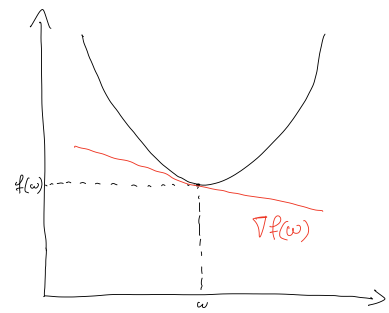
Definition 2. Vector v is called subgradient of a function $f : S \to \mathbb{R}$ at $w \in S$ if

$$\forall u \in S, \quad f(u) \geq f(w) + \langle u - w, v \rangle \tag{2.1}$$

Note 3. There may be more than one subgradients of a function at a specific point. As seen by (2.1), subgradients are the slopes of all the lines passing through the point $(w, f(w))$ and been under the function $f(\cdot)$.



(A) subgradients satisfying (2.1) in the non-differentiable case



(B) gradient satisfying the equality in (2.1) in the differentiable case

Definition 4. The set of subgradients of function $f : S \to \mathbb{R}$ at $w \in S$ is denoted by $\partial f(w)$.

Fact 5. *Properties of subgradient sets useful for the subgradient construction*

(1) *If function $f : S \to \mathbb{R}$ is differentiable at w then the only subgradient of f at w is the gradient $\nabla f(w)$, and (2.1) is equality; i.e. $\partial f(w) = \{\nabla f(w)\}$.*

(2) *for constants α, β and convex function $f(\cdot)$, it is*

$$\partial(\alpha f(w) + \beta) = \alpha(\partial f(w)) = \{\alpha v : v \in \partial f(w),\}$$

(3) *for convex functions $f(\cdot)$ and $g(\cdot)$, it is*

$$\partial(f(w) + g(w)) = \partial f(w) + \partial g(w) = \{v + u : v \in \partial f(w),\ and\, u \in \partial g(w)\}$$

Example 6. Consider the function $f : \mathbb{R} \to \mathbb{R}_+$ with $f(w) = |w| = \begin{cases} w & w \geq 0 \\ -w & w < 0 \end{cases}$. Find the set of subgradients $\partial f(w)$ for each $w \in \mathbb{R}$.

Solution. Using Fact 5, it is $\partial f(w) = 1$ for $w > 0$ and $\partial f(w) = -1$ for $w < 0$ as f is differentiable for $x \neq 0$. At $x = 0$, f is not differentiable; hence from condition (2.1) it is

$$\forall u \in \mathbb{R}, \ \ |u| \geq |0| + (u - 0)\, v$$

which is satisfied for $v \in [-1, 1]$. Hence,

$$\partial f(w) = \begin{cases} \{-1\} & , w < 0 \\ [-1, 1] & , w = 0 \\ \{1\} & , w > 0 \end{cases}$$

3. Description of Gradient Descent

Problem 7. For the sake of notation simplicity and generalization, we will present Gradient Descent (GD) in the following minimization problem

$$w^{*}=\arg\min_{w\in\mathcal{H}}\left(f\left(w\right)\right) \tag{3.1}$$

where $f:\mathcal{H}\rightarrow\mathbb{R}$, and $w\in\mathcal{H}\subseteq\mathbb{R}^{d}$; $f\left(\cdot\right)$ is the function to be minimized, e.g., $f\left(\cdot\right)$ can be an empirical risk function $\hat{R}\left(\cdot\right)$.

Algorithm 8. *Gradient Descent (GD) algorithm with learning rate $\eta_t>0$ for the solution of the minimization problem (3.1)*

For $t=1,2,3,...$ iterate:

(1) *compute*

$$w^{(t+1)}=w^{(t)}-\eta_{t}v_{t};\ \ \text{where}\ \ v_{t}\in\partial f\left(w^{(t)}\right) \tag{3.2}$$

(2) *terminate if a termination criterion is satisfied, e.g.*

$$\text{If}\ t\geq T_{\max}\ \ \text{then}\ \ \text{STOP}$$

Note 9. (Intuition) Assume f is differentiable. GD produces a chain $\left\{w^{(t)}\right\}$ that drifts towards a minimum w^{*}. The chain is directed towards the opposite direction of that of the gradient $\nabla f\left(\cdot\right)$ and at a rate controlled by the learning rate η_t.

Note 10. (More intuition) Assume f is differentiable. Consider the (1st order) Taylor polynomial for the approximation of $f\left(w\right)$ in a small area around u (i.e. $\|v-u\|=\text{small}$)

$$f\left(u\right)\approx P\left(u\right)=f\left(w\right)+\langle u-w,\nabla f\left(w\right)\rangle$$

Assuming convexity for f, it is

$$f\left(u\right)\geq\underbrace{f\left(w\right)+\langle u-w,\nabla f\left(w\right)\rangle}_{=P\left(u;w\right)} \tag{3.3}$$

meaning that P lower bounds f. Hence we could design an updating mechanism producing $w^{(t+1)}$ which is nearby $w^{(t)}$ (small steps) and which minimize the linear approximation $P\left(w\right)$ of $f\left(w\right)$ at $w^{(t)}$

$$P\left(w;w^{(t)}\right)=f\left(w^{(t)}\right)+\langle w-w^{(t)},\nabla f\left(w^{(t)}\right)\rangle. \tag{3.4}$$

while hoping that this mechanism would push the produced chain $\left\{w^{(t)}\right\}$ towards the minimum because of (3.3). Hence we could recursively minimize the linear approximation (3.4) and the distance between the current state $w^{(t)}$ and the next w value to produce $w^{(t+1)}$; namely

$$
\begin{aligned}
(3.5)\qquad w^{(t+1)} =& \arg\min_{\forall w}\left(\frac{1}{2}\left\|w-w^{(t)}\right\|^{2}+\eta P\left(w;w^{(t)}\right)\right)\\
=& \arg\min_{\forall w}\left(\frac{1}{2}\left\|w-w^{(t)}\right\|^{2}+\eta\left(f\left(w^{(t)}\right)+\langle w-w^{(t)},\nabla f\left(w^{(t+1)}\right)\rangle\right)\right)\\
=& w^{(t)}-\eta\nabla f\left(w^{(t)}\right)
\end{aligned}
$$

where parameter $\eta>0$ controls the trade off in (3.5).

Note 11. GD output (e.g. in Note 14) converges to a local minimum, $w_{\text{GD}}^{(T)}\rightarrow w_{*}$ (in some sense), under different sets of regularity conditions (some are weaker other stronger). Section 4 has a brief analysis.

Note 12. The parameter η_t is called learning rate (or step size, gain). It determines the size of the steps GD takes to reach a (local) minimum. $\{\eta_t\}$ is a non-negative sequence and it is chosen by the practitioner. In principle, regularity conditions (Note 11) often imply restrictions on the decay of $\{\eta_t\}$ which guide the practitioner to parametrize it properly. Some popular choices of learning rate η_t are:

(1) <u>constant</u>; $\eta_t=\eta$, for where $\eta>0$ is a small value. The rationale is that GD chain $\{w_t\}$ performs constant small steps towards the (local) minimum w_* and then oscillate around it.
(2) <u>decreasing and converging to zero</u>; $\eta_t\searrow$ with $\lim_{t\rightarrow\infty}\eta_t=0$. E.g. $\eta_t=\left(\frac{C}{t}\right)^{\varsigma}$ where $\varsigma\in[0.5,1]$ and $C>0$. The rationale is that GD algorithm starts by performing larger steps (controlled by C) at the beginning to explore the area for discovering possible minima. Also it reduces the size of those steps with the iterations (controled by ς) such that eventually when the chain $\{w_t\}$ is close to a possible minimum w_* value to converge and do not overshoot.
(3) <u>decreasing and converging to a tiny value</u> τ_*; $\eta_t\searrow$ with $\lim_{t\rightarrow\infty}\eta_t=\tau_*$ E.g. $\eta_t=\left(\frac{C}{t}\right)^{\varsigma}+\tau_*$ with $\varsigma\in(0.5,1]$, $C>0$, and $\tau_*\approx 0$. Same as previously, but the algorithm aims at oscillating around the detected local minimum.
(4) <u>constant until an iteration T_0 and then decreasing</u>; Eg $\eta_t=\left(\frac{C}{\max(t,T_0)}\right)^{\varsigma}$ with $\varsigma\in[0.5,1]$ and $C>0$, and $T_0<T$. The rationale is that at the first stage of the iterations (when $t\leq T_0$) the algorithm may need a constant large steps for a significant number of iterations T_0 in order to explore the domain; and hence in order for the chain $\{w_t\}$ to reach the area around the (local) minimum w_*. In the second stage, hoping that the chain $\{w_t\}$ may be in close proximity to the (local) minimum w_* the algorithm progressively performs smaller steps to converge towards the minimum w_*. The first stage ($t\leq T_0$) is called burn-in; the values $\{w_t\}$ produced during the burn-in ($t\leq T_0$) are are often discarded/ignored from the output of the GD algorithm.

- Parameters C,ς,τ_*,T_0 may be chosen based on pilot runs against a small fraction of the training data set.

Note 13. There are several practical termination criteria that can be used in GD Algorithm 8(step 2). They aim to terminate the recursion in practice. Some popular termination criteria are

(1) terminate when the gradient is sufficiently close to zero; i.e. if $\left\|\nabla f\left(w^{(t)}\right)\right\| \leq \epsilon$, for some pre-specified tiny $\epsilon > 0$ then STOP
(2) terminate when the chain $w^{(t)}$ does not change; i.e. if $\left\|w^{(t+1)} - w^{(t)}\right\| \leq \epsilon \left\|w^{(t)}\right\|$ for some pre-specified tiny $\epsilon > 0$ then STOP
(3) terminate when a pre-specified number of iterations T is performed; i.e. if $t \geq T$ then STOP

Here (1) may be deceive if the chain is in a flat area, (2) may be deceived if the learning rate become too small, (3) is obviously a last resort.

Note 14. Given T iterations of GD algorithm, the output of GD can be (but not a exclusively),

(1) the average (after discarding the first few iterations of $w^{(t)}$ for stability reasons)

$$w_{\mathrm{GD}}^{(T)} = \frac{1}{T}\sum_{t=1}^{T} w^{(t)} \tag{3.6}$$

(2) or the best value discovered

$$w_{\mathrm{GD}}^{(T)} = \arg\min_{\forall w_t}\left(f\left(w^{(t)}\right)\right)$$

(3) or the last value discovered

$$w_{\mathrm{GD}}^{(T)} = w^{(T)}$$

4. Analysis of Gradient descent (Algorithm 8)

Note 15. Recall we address the minimization Problem 7 under Assumptions 16.

Assumption 16. *For the sake of the analysis of the GD, let us consider:*

(1) *The function $f(\cdot)$ is convex and Lipschitz*
(2) *GD has a constant learning rate $\eta_t = \eta$,*
(3) *GD output $w_{GD}^{(T)} = \frac{1}{T}\sum_{t=1}^{T} w^{(t)}$*

Lemma 17. *Let $\{v_t; t = 1, ..., T\}$ be a sequence of vectors. Any algorithm with $w^{(1)} = 0$ and $w^{(t+1)} = w^{(t)} - \eta v_t$ for $t = 1, ..., T$ satisfies* No need to memorize

$$\sum_{t=1}^{T}\langle w^{(t)} - w^{*}, v_t\rangle \leq \frac{\|w^{*}\|^2}{2\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\|v_t\|^2 \tag{4.1}$$

Proof. Omitted; see the Exercise 13 in the Exercise sheet 1. □

Note 18. To find an upper bound of the GD error, we try to bound the error $f\left(w_{\mathrm{GD}}^{(T)}\right) - f(w^{*})$ with purpose to use Lemma 17.

Note 19. Consider the minimization problem (3.1). Given Assumptions 16, the error can be bounded as

$$f\left(w_{\mathrm{GD}}^{(T)}\right) - f(w^{*}) \leq \frac{\|w^{*}\|^2}{2\eta T} + \frac{\eta}{2}\frac{1}{T}\sum_{t=1}^{T}\|v_t\|^2 \tag{4.2}$$

where $v_t \in \partial f\left(w^{(t)}\right)$. If $f(\cdot)$ is differentiable then $v_t = \nabla f\left(w^{(t)}\right)$

Proof. It is[1]

$$
\begin{aligned}
f\left(w_{\mathrm{GD}}^{(T)}\right)-f\left(w^*\right)=&f\left(\frac{1}{T}\sum_{t=1}^{T}w_t\right)-f\left(w^*\right)\\
\le&\frac{1}{T}\sum_{t=1}^{T}\left(f\left(w_t\right)-f\left(w^*\right)\right) && \text{(by Jensen's inequality)} \quad (4.3)\\
\le&\frac{1}{T}\sum_{t=1}^{T}\langle w^{(t)}-w^*,v_t\rangle && \text{(by convexity of } f\left(\cdot\right)\text{)}\\
\le&\frac{\left\|w^*\right\|^2}{2\eta T}+\frac{\eta}{2}\frac{1}{T}\sum_{t=1}^{T}\left\|v_t\right\|^2 && \text{(by roceeding Lemma)} \quad (4.4)
\end{aligned}
$$

□

Note 20. Note 19 shows that it is important to bound the (sub-)gradient in (4.2) in a meaningful manner. Lemma 21 shows that if we assume Lipschitzness as well, the (sub-)gradient has the so-called self-bounded behavior.

Lemma 21. [2] *$f:S\to\mathbb{R}$ is ρ-Lipschitz over an open convex set S if and only if for all $w\in S$ and $v\in\partial f\left(w\right)$ it is $\left\|v\right\|\le\rho$.*

Proof. $\Longrightarrow$ Let $f:S\to\mathbb{R}$ be ρ-Lipschitz over convex set S, $w\in S$ and $v\in\partial f\left(w\right)$.

- Since S is open we get that there exist $\epsilon>0$ such as $u:=w+\epsilon\frac{v}{\|v\|}$ where $u\in S$. So $\langle u-w,v\rangle=\epsilon\left\|v\right\|$ and $\left\|u-w\right\|=\epsilon$.
- From the subgradient definition we get

$$f\left(u\right)-f\left(w\right)\ge\langle u-w,v\rangle=\epsilon\left\|v\right\|$$

- From the Lipschitzness of $f\left(\cdot\right)$ we get

$$f\left(u\right)-f\left(w\right)\le\rho\left\|u-w\right\|=\rho\epsilon$$

Therefore $\left\|v\right\|\le\rho$.

For $\Longleftarrow$ see Exercise 4 in the Exercise sheet. □

Note 22. The following summaries Note 19 and Lemma 21 with respect to the GD algorithm satisfying Assumption 16.

Note 23. Let $f\left(\cdot\right)$ be a convex and ρ-Lipschitz function. If we run GD algorithm of f with learning rate $\eta>0$ for T steps the output $w_{\mathrm{GD}}^{(T)}=\frac{1}{T}\sum_{t=1}^{T}w^{(t)}$ satisfies

$$f\left(w_{\mathrm{GD}}^{(T)}\right)-f\left(w^*\right)\le\frac{\left\|w^*\right\|^2}{2\eta T}+\frac{\eta}{2}\frac{1}{T}\sum_{t=1}^{T}\left\|v_t\right\|^2$$

[1] Jensen's inequality for convex $f\left(.\right)$ is $f\left(\mathrm{E}\left(x\right)\right)\le\mathrm{E}\left(f\left(x\right)\right)$

[2] If this was a Homework there would be a Hint:

- If S is open there exist $\epsilon>0$ such as $u=w+\epsilon\frac{v}{\|v\|}$ such as $u\in S$

where $\|v_t\| \leq \rho$, $v_t \in \partial f\left(w^{(t)}\right)$. If $f(\cdot)$ is differentiable then $v_t = \nabla f\left(w^{(t)}\right)$.

Proof. Straightforward from Lemma 17 and Note 19. □

Note 24. The following shows that a given learning rate depending on the iteration t, we can reduce the upper bound of the error as well as find the number of required iterations to achieve convergence.

Note 25. (Cont Prop. 23) Let $f(\cdot)$ be a convex and ρ-Lipschitz function, and let $\mathcal{H} = \{w \in \mathbb{R} : \|w\| \leq B\}$. Assume we run GD algorithm of $f(\cdot)$ with learning rate $\eta_t = \sqrt{\frac{B^2}{\rho^2 T}}$ for T steps, and output $w_{\text{GD}}^{(T)} = \frac{1}{T}\sum_{t=1}^{T} w^{(t)}$. Then

(1) upper bound on the sub-optimality is

$$f\left(w_{\text{GD}}^{(T)}\right) - f(w^*) \leq \frac{B\rho}{\sqrt{T}} \tag{4.5}$$

(2) a given level off accuracy ε such that $f\left(w_{\text{GD}}^{(T)}\right) - f(w^*) \leq \varepsilon$ can be achieved after T iterations

$$T \geq \frac{B^2\rho^2}{\varepsilon^2}.$$

Proof. Part 1 is a simple substitution from Proposition 23, and part 2 is implied from part 1. □

The result on Note 25 heavily relies on setting suitable values for B and ρ which is rather a difficult task to be done in very complicated learning problems (e.g., learning a neural network).

5. Examples [3]

Example 26. Consider the simple Normal linear regression problem where the dataset $\{z_i = (y_i, x_i)\}_{i=1}^{n} \in \mathcal{S}$ is generated from a Normal data generating model

$$\begin{pmatrix} y_i \\ x_i \end{pmatrix} \overset{\text{iid}}{\sim} \text{N}\left(\begin{pmatrix} \mu_y \\ \mu_x \end{pmatrix}, \begin{bmatrix} \sigma_y^2 & \rho\sqrt{\sigma_y^2\sigma_x^2} \\ \rho\sqrt{\sigma_y^2\sigma_x^2} & \sigma_x^2 \end{bmatrix}\right) \tag{5.1}$$

for $i = 1, ..., n$. Consider a hypothesis space $\mathcal{H}$ of linear functions $h : \mathbb{R}^2 \to \mathbb{R}$ with $h(w) = w_1 + w_2 x$. The exact solution (which we pretend we do not know) is given as

$$\begin{pmatrix} w_1^* \\ w_2^* \end{pmatrix} = \begin{pmatrix} \mu_y - \rho\frac{\sigma_y}{\sigma_x}\mu_x \\ \rho\frac{\sigma_y}{\sigma_x} \end{pmatrix}. \tag{5.2}$$

To learn the optimal $w^* = (w_1^*, w_2^*)^\top$, we consider a loss $\ell\left(w, z_i = (x_i, y_i)^\top\right) = (y_i - [w_1 + w_2 x_i])^2$, which leads to the minimization problem

$$w^* = \arg\min_w \left(\hat{R}_{\mathcal{S}}(w)\right) = \arg\min_w \left(\frac{1}{n}\sum_{i=1}^{n}(y_i - w_1 - w_2 x_i)^2\right)$$

The GD Algorithm 8 with learning rate η is

For $t = 1, 2, 3, ...$ iterate:

[3] Code is available in `https://github.com/georgios-stats/Machine_Learning_and_Neural_Networks_III_Epiphany_2025/tree/main/Lecture_notes/code/Gradient_descent_example_1.R`

(1) compute

$$w^{(t+1)} = w^{(t)} - \eta v_t, \qquad \text{where } v_t = \begin{pmatrix} 2w_1^{(t)} + 2w_2^{(t)}\bar{x} - 2\bar{y} \\ 2w_1^{(t)}\bar{x} + 2w_2^{(t)}\overline{x^2} - 2y^\top x \end{pmatrix} \tag{5.3}$$

(2) terminate if a termination criterion is satisfied, e.g.

$$\text{If } t \geq T \text{ then STOP}$$

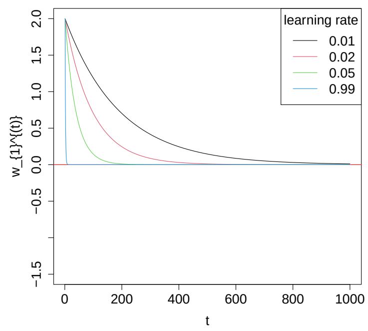
This is because $\hat{R}_{\mathcal{D}}(w)$ is differentiable in $\mathbb{R}^2$ so $\partial \hat{R}_{\mathcal{D}}(w) = \left\{ \nabla \hat{R}_{\mathcal{D}}(w) \right\}$ and because

$$\nabla \hat{R}_{\mathcal{D}}(w) = \begin{pmatrix} \frac{\mathrm{d}}{\mathrm{d}w_1} \hat{R}_{\mathcal{S}}(w) \\ \frac{\mathrm{d}}{\mathrm{d}w_2} \hat{R}_{\mathcal{S}}(w) \end{pmatrix} = ... = \begin{pmatrix} 2w_1^{(t)} + 2w_2^{(t)}\bar{x} - 2\bar{y} \\ 2w_1^{(t)}\bar{x} + 2w_2^{(t)}\overline{x^2} - 2y^\top x \end{pmatrix}$$

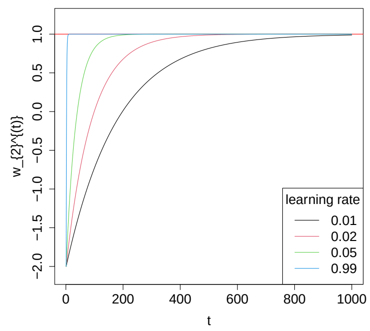
Consider data size $n = 100$, and parameters $\rho = 0.2$, $\sigma_y^2 = 1$ and $\sigma_x^2 = 1$. Then the real value (5.2) that I need to learn equals to $w^* = (0, 1)^\top$. Consider a GD seed $w_0 = (2, -2)$, and total number of iterations $T = 1000$.

Figures 5.1a, 5.1b, and 5.1c present trace plots of the chain $\left\{ \left(w^{(t)} \right) \right\}$ and error $\hat{R}_{\mathcal{S}}\left(w^{(t)} \right) - \hat{R}_{\mathcal{S}}(w^*)$ produced by running GD for $T = 1000$ total iterations and for different (each time) constant learning rates $\eta \in \{0.01, 0.02, 0.05, 0.99\}$. We observe that the larger learning rates under consideration were able to converge faster to the minimum w^*. This is because they perform larger steps and can learn faster -this is not a panacea.

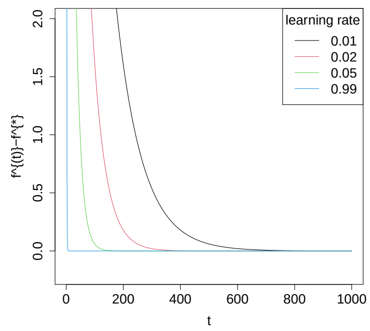
Figures 5.1d, 5.1e, and 5.1f present trace plots of the chain $\left\{ \left(w^{(t)} \right) \right\}$, and of the error $\hat{R}_{\mathcal{S}}\left(w^{(t)} \right) - \hat{R}_{\mathcal{S}}(w^*)$ produced by running GD for $T = 1000$ total iterations and for learning rate $\eta = 1.0$ (previously considered) and a very big learning rate $\eta = 3.0$. We observe that the very big learning rate $\eta = 3.0$ presents slower convergence to the minimum w^*. This is because it creates unreasonably big steps in (3.2) that the produced chain overshoots the global minimum; see the cartoon in Figures 5.1g and 5.1h.



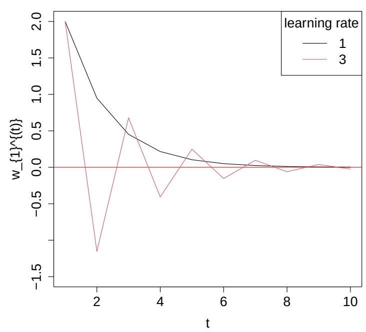
(A) $\left\{\left(w_1^{(t)}\right)\right\}$



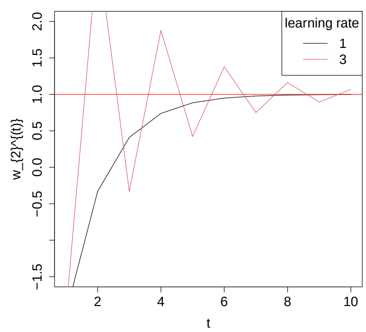
(B) $\left\{\left(w_2^{(t)}\right)\right\}$



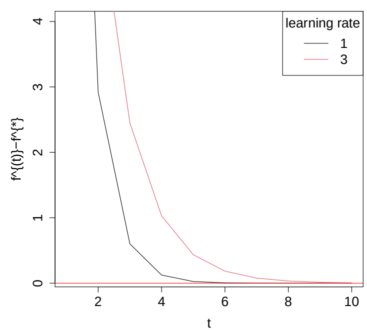
(C) $\hat{R}_{\mathcal{S}}\left(w^{(t)}\right) - \hat{R}_{\mathcal{S}}\left(w^{*}\right)$



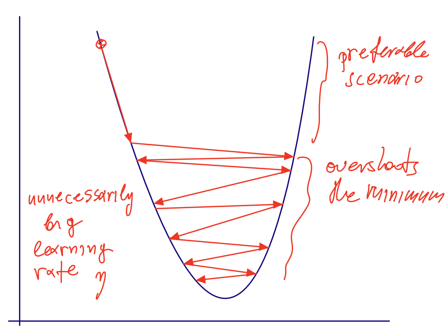
(D) $\left\{\left(w_1^{(t)}\right)\right\}$



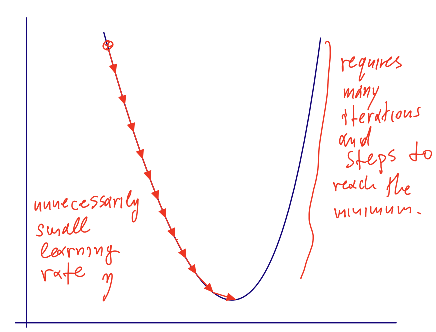
(E) $\left\{\left(w_2^{(t)}\right)\right\}$



(F) $\hat{R}_{\mathcal{S}}\left(w^{(t)}\right) - \hat{R}_{\mathcal{S}}\left(w^{*}\right)$



(G) Unnecessarily large learning rate



(H) Unnecessarily small learning rate

FIGURE 5.1