HW2 Handwritten Assignment Solution

Lecturor: Pei-Yuan Wu
TAs: Po-Yang Hsieh, Le-Rong Hsu

October 2024

Problem 1: (Automatic Differentiation) (0.5%)

Let $f = f_4 \circ f_3 \circ f_2 \circ f_1$ be a differentiable function defined as

$$\begin{aligned} f : \mathbb{R}^n &\to \mathbb{R} \\ \mathbf{x} &\mapsto f_4(f_3(f_2(f_1(\mathbf{x})))) \end{aligned}$$

where $f_1 : \mathbb{R}^n \to \mathbb{R}^{n_3}$, $f_2 : \mathbb{R}^{n_3} \to \mathbb{R}^{n_2}$, $f_3 : \mathbb{R}^{n_2} \to \mathbb{R}^{n_1}$, and $f_4 : \mathbb{R}^{n_1} \to \mathbb{R}$. We assume $n >> 1$ and $n > n_3 > n_2 > n_1 > 1$.

1. (0.2%) Write down the derivative of f with respect to $\mathbf{x}$ using **chain rule**. Next, express **forward-mode** and **reverse-mode** auto-differentiation respectively with the formula you just wrote down. The reverse-mode auto-differentiation is also called **backpropagation**. In this problem, please "assign variables" when using chain rule. For example, suppose $y = g(f(\mathbf{x}))$. Let $w_1 = f(\mathbf{x})$ and $y = w_2 = g(w_1)$. Then the partial derivative of y with respect to $\mathbf{x}$ is given by

$$\frac{\partial y}{\partial \mathbf{x}} = \frac{\partial y}{\partial w_2}\frac{\partial w_2}{\partial w_1}\frac{\partial w_1}{\partial \mathbf{x}} = 1 \cdot \frac{\partial w_2}{\partial w_1}\frac{\partial w_1}{\partial \mathbf{x}} = \frac{\partial w_2}{\partial w_1}\frac{\partial w_1}{\partial \mathbf{x}} \left(= \frac{\partial y}{\partial w_1}\frac{\partial w_1}{\partial \mathbf{x}}\right).$$

2. (0.3%) Which mode is better in terms of efficiency? Why? For the second question, you need to prove your answer. Note that if you only answer the first question, you will only get 0.01%. Hint: you may write down the size of Jacobian matrices and see how many scalar multiplications ($a \times b$, $a, b \in \mathbb{R}$) are performed in the forward-mode and reverse-mode auto-differentiation respectively.

Through this problem, we want students to tell the difference between the forward mode and the reverse mode, and to understand why the reverse mode is usually preferred for optimization over the forward mode.

Solution: 1. Let $w_1 = f_1(\mathbf{x})$, $w_2 = f_2(w_1)$, $w_3 = f_3(w_2)$, and $y = w_4 = f_4(w_3)$. By chain rule, we have

$$\frac{\partial y}{\partial \mathbf{x}} = \frac{\partial w_4}{\partial w_3}\frac{\partial w_3}{\partial w_2}\frac{\partial w_2}{\partial w_1}\frac{\partial w_1}{\partial \mathbf{x}}.$$

The forward-mode autodifferentiation is given by

$$\begin{aligned}\frac{\partial y}{\partial \mathbf{x}} &= \frac{\partial y}{\partial w_3}\frac{\partial w_3}{\partial \mathbf{x}} \\ &= \frac{\partial y}{\partial w_3}\left(\frac{\partial w_3}{\partial w_2}\frac{\partial w_2}{\partial w_1}\right) \\ &= \frac{\partial y}{\partial w_3}\left(\frac{\partial w_3}{\partial w_2}\left(\frac{\partial w_2}{\partial w_1}\frac{\partial w_1}{\partial \mathbf{x}}\right)\right),\end{aligned}$$

and the reverse-mode autodifferentiation is given by

$$\begin{aligned}\frac{\partial y}{\partial \mathbf{x}} &= \frac{\partial y}{\partial w_1}\frac{\partial w_1}{\partial \mathbf{x}} \\ &= \left(\frac{\partial y}{\partial w_2}\frac{\partial w_2}{\partial w_1}\right)\frac{\partial w_1}{\partial \mathbf{x}} \\ &= \left(\left(\frac{\partial y}{\partial w_3}\frac{\partial w_3}{\partial w_2}\right)\frac{\partial w_2}{\partial w_1}\right)\frac{\partial w_1}{\partial \mathbf{x}}\end{aligned}$$

2. Reverse-mode auto-differentiation is more efficient because it performs vector-matrix products, while the forward mode performs matrix-matrix products. The calculation is trivial and left as an exercise.

Problem 2 (Batch Normalization)(1%)

Batch normalization [?] is a popular trick for training deep networks nowadays, which aims to preserve the distribution within hidden layers and avoids vanishing gradient issue. The alogrithm can be written as below:

Algorithm 1 Batch Normalization

Input Feature from data points over a mini-batch $B = (x_i)_{i=1}^m$
Output $y_i = BN_{\gamma,\beta}(x_i)$

procedure BatchNormalize(B, γ, β)
  $\mu_B \leftarrow \frac{1}{m}\sum_{i=1}^m x_i$ ▷ mini-batch mean
  $\sigma_B^2 \leftarrow \frac{1}{m}\sum_{i=1}^m (x_i - \mu_B)^2$ ▷ mini-batch variance
  for $i \leftarrow 1$ to m **do**
    $\hat{x}_i \leftarrow \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$ ▷ normalize
    $y_i \leftarrow \gamma \hat{x}_i + \beta$ ▷ scale and shift
  end for
  return
end procedure

During training we need to backpropagate the gradient of loss ℓ through this transformation, as well as compute the gradients with respect to the parameters γ, β. Towards this end, please write down the close form expressions for $\frac{\partial \ell}{\partial x_i}$, $\frac{\partial \ell}{\partial \gamma}$, $\frac{\partial \ell}{\partial \beta}$ in terms of x_i, μ_B, σ_B^2, $\hat{x}_i$, y_i (given by the forward pass) and $\frac{\partial \ell}{\partial y_i}$ (given by the backward pass).

- Hint: You may first write down the close form expressions of $\frac{\partial \ell}{\partial \hat{x}_i}$, $\frac{\partial \ell}{\partial \sigma_B^2}$, $\frac{\partial \ell}{\partial \mu_B}$, and then use them to compute $\frac{\partial \ell}{\partial x_i}$, $\frac{\partial \ell}{\partial \gamma}$, $\frac{\partial \ell}{\partial \beta}$.

Solution: We use the gradient descent to update γ and β:

$$\gamma \leftarrow \gamma - \eta \frac{\partial l}{\partial \gamma}$$
$$\beta \leftarrow \beta - \eta \frac{\partial l}{\partial \beta}$$

where η is a learning rate and

$$\frac{\partial l}{\partial \gamma} = \sum_{i=1}^m \left(\frac{\partial l}{\partial y_i} \frac{\partial y_i}{\partial \gamma} \right) = \sum_{i=1}^m \frac{\partial l}{\partial y_i} \hat{x_i}$$
$$\frac{\partial l}{\partial \beta} = \sum_{i=1}^m \left(\frac{\partial l}{\partial y_i} \frac{\partial y_i}{\partial \beta} \right) = \sum_{i=1}^m \frac{\partial l}{\partial y_i}$$

Note that we sum from 1 to m because we are working with mini-batches. Now,

we derive some important terms by chain rule:

$$\frac{\partial l}{\partial \hat{x_i}} = \frac{\partial l}{\partial y_i}\frac{\partial y_i}{\partial \hat{x_i}} = \frac{\partial l}{\partial y_i}\gamma$$
$$\frac{\partial l}{\partial \sigma_B^2} = \frac{\partial l}{\partial \hat{x_i}}\frac{\partial \hat{x_i}}{\partial \sigma_B^2} = -\frac{1}{2}\sum_{i=1}^{m}\frac{\partial l}{\partial \hat{x_i}}(x_i-\mu_B)(\sigma_B^2+\epsilon)^{-\frac{3}{2}}$$

$$\begin{aligned}\frac{\partial l}{\partial \mu_B} &= \frac{\partial l}{\partial \hat{x_i}}\frac{\partial \hat{x_i}}{\partial \mu_B} \\ &= \sum_{i=1}^{m}\frac{\partial l}{\partial \hat{x_i}}\frac{\partial}{\partial \mu_B}(x_i-\mu_B)(\sigma_B^2+\epsilon)^{-\frac{1}{2}} \\ &= \sum_{i=1}^{m}\frac{\partial l}{\partial \hat{x_i}}\left[\frac{\partial (x_i-\mu_B)}{\partial \mu_B}(\sigma_B^2+\epsilon)^{-\frac{1}{2}} + (x_i-\mu_B)\frac{\partial(\sigma_B^2+\epsilon)^{-\frac{1}{2}}}{\partial \mu_B}\right] \\ &= \sum_{i=1}^{m}\frac{\partial l}{\partial \hat{x_i}}\left[\frac{-1}{\sqrt{\sigma_B^2+\epsilon}} + (x_i-\mu_B)\frac{\partial(\sigma_B^2+\epsilon)^{-\frac{1}{2}}}{\partial \mu_B}\right]\end{aligned}$$

We know $\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i-\mu_B)^2$, so the second term in bracket is

$$\begin{aligned}(x_i-\mu_B)\frac{\partial(\sigma_B^2+\epsilon)^{-\frac{1}{2}}}{\partial \mu_B} &= (x_i-\mu_B)\frac{-1}{2}(\sigma_B^2+\epsilon)^{-\frac{3}{2}}\frac{\partial(\sigma_B^2+\epsilon)}{\partial \mu_B} \\ &= \frac{-1}{2}(x_i-\mu_B)(\sigma_B^2+\epsilon)^{-\frac{3}{2}}\frac{\partial\left(\frac{1}{m}\sum_{i=1}^{m}(x_i-\mu_B)^2+\epsilon\right)}{\partial \mu_B} \\ &= \frac{-1}{2}(x_i-\mu_B)(\sigma_B^2+\epsilon)^{-\frac{3}{2}}\left(\frac{-2}{m}\sum_{i=1}^{m}(x_i-\mu_B)\right)\end{aligned}$$

Hence,

$$\begin{aligned}\frac{\partial l}{\partial \mu_B} &= \sum_{i=1}^{m}\frac{\partial l}{\partial \hat{x_i}}\frac{-1}{\sqrt{\sigma_B^2+\epsilon}} + \underbrace{\sum_{i=1}^{m}\frac{\partial l}{\partial \hat{x_i}}\frac{-1}{2}(x_i-\mu_B)(\sigma_B^2+\epsilon)^{-\frac{3}{2}}}_{\frac{\partial l}{\partial \sigma_B^2}}\left(\frac{-2}{m}\sum_{i=1}^{m}(x_i-\mu_B)\right) \\ &= \sum_{i=1}^{m}\frac{\partial l}{\partial \hat{x_i}}\frac{-1}{\sqrt{\sigma_B^2+\epsilon}} + \frac{\partial l}{\partial \sigma_B^2}\left(\frac{-2}{m}\sum_{i=1}^{m}(x_i-\mu_B)\right)\end{aligned}$$

To derive $\frac{\partial l}{\partial x_i}$, we use the chain rule $\frac{\partial l}{\partial x_i} = \frac{\partial l}{\partial \hat{x_i}}\frac{\partial \hat{x_i}}{\partial x_i} + \frac{\partial l}{\partial \sigma_B^2}\frac{\partial \sigma_B^2}{\partial x_i} + \frac{\partial l}{\partial \mu_B}\frac{\partial \mu_B}{\partial x_i}$. Now, calculate the remaining term:

$$\frac{\partial \hat{x_i}}{\partial x_i} = \frac{1}{\sqrt{\sigma_B^2+\epsilon}}$$
$$\frac{\partial \mu_B}{\partial x_i} = \frac{1}{m}$$
$$\frac{\partial \sigma_B^2}{\partial x_i} = \frac{2(x_i-\mu)}{m}$$

That is,

$$\frac{\partial l}{\partial x_i} = \frac{\partial l}{\partial \hat{x}_i}\frac{1}{\sqrt{\sigma_B^2 + \epsilon}} + \frac{\partial l}{\partial \sigma_B^2}\frac{2(x_i - \mu)}{m} + \frac{1}{m}\frac{\partial l}{\partial \mu_B}$$

Problem 3 (Multiclass AdaBoost)(1.5%)

Let $\mathcal{X}$ be the input space, $\mathcal{F}$ be a collection of multiclass classifiers that map from $\mathcal{X}$ to $[1, K]$, where K denotes the number of classes. Let $\{(x_i, \hat{y}_i)\}_{i=1}^m$ be the training data set, where $x_i \in \mathcal{X}$ and $\hat{y}_i \in [1, K]$. Given $T \in \mathbb{N}$, suppose we want to find functions

$$g_{T+1}^k(x) = \sum_{t=1}^{T} \alpha_t f_t^k(x), \quad k \in [1, K]$$

where $f_t \in \mathcal{F}$ and $\alpha_t \in \mathbb{R}$ for all $t \in [1, T]$. Here for $f \in \mathcal{F}$, we denote $f^k(x) = \mathbf{1}\{f(x) = k\}$, where $\mathbf{1}(\cdot)$ is an indicator function, as the k'th element in the one-hot representation of $f(x) \in [1, K]$. The aggregated classifier $h : \mathcal{X} \to [1, K]$ is defined as

$$x \mapsto \underset{1 \le k \le K}{\operatorname{argmax}}\ g_{T+1}^k(x)$$

Please apply gradient boosting to show how the functions f_t and coefficients α_t are computed with an aim to minimize the following loss function

$$L((g_{T+1}^1, \cdots, g_{T+1}^K) = \sum_{i=1}^{m} \exp\left(\frac{1}{K-1} \sum_{k \neq \hat{y}_i} g_{T+1}^k(x_i) - g_{T+1}^{\hat{y}_i}(x_i)\right)$$

Solution: Given $\mathbf{g}_{t-1} = \left\{g_{t-1}^k\right\}_{k=1}^K$, we update $\mathbf{g}_t = \left\{g_{t-1}^k + \alpha_t f_t^k\right\}_{k=1}^K = \mathbf{g}_{t-1} +$

$\alpha_t \mathrm{f}_t$ as follows:

$$
\begin{aligned}
&\mathbf{f}_t \in \underset{f \in \mathcal{F}}{\operatorname{argmin}} \frac{\partial}{\partial \alpha} L\left(\mathbf{g}_{t-1}+\alpha \mathbf{f}\right)\Big|_{\alpha=0} \\
&= \underset{f \in \mathcal{F}}{\operatorname{argmin}} \frac{\partial}{\partial \alpha} \sum_{i=1}^{n} \exp\left(\left(\left(\frac{1}{K-1} \sum_{k \neq \hat{y}_i} g_{t-1}^{k}\left(x_i\right)-g_{t-1}^{\hat{y}_i}\left(x_i\right)\right)+\alpha\left(\frac{1}{K-1} \sum_{k \neq \hat{y}_i} f^{k}\left(x_i\right)-f^{\hat{y}_i}\left(x_i\right)\right)\right)\right)\Big|_{\alpha=0} \\
&= \underset{f \in \mathcal{F}}{\operatorname{argmin}} \sum_{i=1}^{n} \exp\left(\frac{1}{K-1} \sum_{k \neq \hat{y}_i} g_{t-1}^{k}\left(x_i\right)-g_{t-1}^{\hat{y}_i}\left(x_i\right)\right)\left(\frac{1}{K-1} \sum_{k \neq \hat{y}_i} f^{k}\left(x_i\right)-f^{\hat{y}_i}\left(x_i\right)\right) \\
&= \underset{f \in \mathcal{F}}{\operatorname{argmin}} Z_t \mathbb{E}_{i \sim D_t}\left[\frac{1}{K-1} \sum_{k \neq \hat{y}_i} f^{k}\left(x_i\right)-f^{\hat{y}_i}\left(x_i\right)\right] \\
&= \underset{f \in \mathcal{F}}{\operatorname{argmin}} Z_t \mathbb{E}_{i \sim D_t}\left[\frac{1}{K-1} \cdot 1\left\{f\left(x_i\right) \neq \hat{y}_i\right\}-1\left\{f\left(x_i\right)=\hat{y}_i\right\}\right] \\
&= \underset{f \in \mathcal{F}}{\operatorname{argmin}} Z_t\left(\frac{K}{K-1} \mathbb{P}_{i \sim D_t}\left[f\left(x_i\right) \neq \hat{y}_i\right]-1\right)=\underset{f \in \mathcal{F}}{\operatorname{argmin}} \mathbb{P}_{i \sim D_t}\left[f\left(x_i\right) \neq \hat{y}_i\right] \\
&\alpha_t \in \underset{\alpha \in \mathbb{R}}{\operatorname{argmin}} L\left(\mathbf{g}_{t-1}+\alpha \mathbf{f}_t\right) \\
&= \underset{\alpha \in \mathbb{R}}{\operatorname{argmin}} \sum_{i=1}^{n} \exp\left(\left(\frac{1}{K-1} \sum_{k \neq \hat{y}_i} g_{t-1}^{k}\left(x_i\right)-g_{t-1}^{\hat{y}_i}\left(x_i\right)\right)+\alpha\left(\frac{1}{K-1} \sum_{k \neq \hat{y}_i} f_t^{k}\left(x_i\right)-f_t^{\hat{y}_i}\left(x_i\right)\right)\right) \\
&= \underset{\alpha \in \mathbb{R}}{\operatorname{argmin}} Z_t \mathbb{E}_{i \sim D_t}\left[e^{\alpha\left(\frac{1}{K-1} \sum_{k \neq \hat{y}_i} f_t^{k}\left(x_i\right)-f_t^{\hat{y}_i}\left(x_i\right)\right)}\right] \\
&= \underset{\alpha \in \mathbb{R}}{\operatorname{argmin}} Z_t \mathbb{E}_{i \sim D_t}\left[e^{\frac{\alpha}{K-1}} \cdot 1\left\{f_t\left(x_i\right) \neq \hat{y}_i\right\}+e^{-\alpha} \cdot 1\left\{f_t\left(x_i\right)=\hat{y}_i\right\}\right] \\
&= \underset{\alpha \in \mathbb{R}}{\operatorname{argmin}} Z_t\left(\epsilon_t e^{\frac{\alpha}{K-1}}+e^{-\alpha}\left(1-\epsilon_t\right)\right)=\left\{\frac{K-1}{K} \log \frac{(K-1)\left(1-\epsilon_t\right)}{\epsilon_t}\right\}
\end{aligned}
$$

where

$$Z_t=\sum_{i=1}^{n} \exp\left(\frac{1}{K-1} \sum_{k \neq \hat{y}_i} g_{t-1}^{k}\left(x_i\right)-g_{t-1}^{\hat{y}_i}\left(x_i\right)\right)$$

and that D_t is a probability distribution for $t=1, \cdots, n$ given by

$$D_t(i)=\frac{1}{Z_t} \exp\left(\frac{1}{K-1} \sum_{k \neq \hat{y}_i} g_{t-1}^{k}\left(x_i\right)-g_{t-1}^{\hat{y}_i}\left(x_i\right)\right)$$

and that $\epsilon_t=\mathbb{P}_{i \sim D_t}\left[f_t\left(x_i\right) \neq \hat{y}_i\right]$ is the error of f_t on training sample weighted by the distribution D_t.

Problem 4 (AdaBoosting)(1%)

1. Consider training an Adaboosting classifier using decision stumps on the data set illustrated in Figure 3:

Figure 1: AdaBoost Data set

   (a) Which examples will have their weights increased at the end of the first iteration? Circle them.

   (b) How many iterations will it take to achieve zero training error? Justify your answers and show each iteration step and weight.

2. Suppose AdaBoost is run on N training examples, and suppose on each round that the weighted training error ϵ_t of the t'th weak hypothesis is at most $1/2-\gamma$, for some number $0 < \gamma < 1/2$. After how many iterations, T, will the combined hypothesis be consistent with the N training examples, i.e., achieves zero training error? Your answer should only be expressed in terms of N and γ. (Hint: Recall that exponential loss is an upper bound for 0-1 loss. What is the training error when 1 example is misclassified?)

Solution:

1. (a) We execute Adaboost algorithm on the data set. First we note that if the classifier classifies the symbol as circle, $f(x) = 1$. On the other hand, if the classfier classifies the symbol as cross, $f(x) = -1$. We choose our first decision stump as a horizontal line to the right of the plane, which is the red line in the following figure. We consider all points as the cross symbol. $\epsilon_1 = 0.2, \alpha_1 = \ln\left(\sqrt{\frac{0.8}{0.2}}\right) = 0.693$. For the cross symbol, the weights becomes to 0.5 , and the weight of the circle symbol becomes to 2 . Hence, the weight of the circle symbol increases at the end of the first iteration.

(b) We choose the first decision stump as we mentioned above. Next, we choose the blue line in the following figure as the second classifier. The classifier classifies the symbol as the circle if it is on the left to the line, and classifies as the cross if it is on the right to the line. At last, the third decision stump is the green line, which classifies the right hand side symbol as circle and the left hand side symbol as cross. After executing, $\alpha_1 = 0.693, \alpha_2 = 0.549, \alpha_3 = 0.822$. $H(x) = \text{sign}\left(\alpha_1 f_1(x) + \alpha_2 f_2(x) + \alpha_3 f_3(x)\right)$. For the circle symbol, $H(x) = \text{sign}(0.678) = 1$. For the left crosses, $H(x) = \text{sign}(-0.966) = -1$. For the right crosses, $H(x) = \text{sign}(-0.42) = -1$. From the above result, we achieve zero training error by these decision stumps.

2. The training error after T iterations is upper bounded by

$$\prod_{t=1}^{T}\left(2\sqrt{\epsilon_t\left(1-\epsilon_t\right)}\right) \leq \prod_{t=1}^{T}(2\sqrt{(1/2-\gamma)(1/2+\gamma)}) = \left(1-4\gamma^2\right)^{T/2}$$

Zero training error can be achieved when $\left(1-4\gamma^2\right)^{T/2} < 1/N$, namely

$$T > \frac{2\log N}{\log\left(1/\left(1-4\gamma^2\right)\right)} \geq \frac{\log N}{2\gamma^2}$$

Here the second equality indicates a (looser but simpler) lower bound for T.

Figure 2: First decision stump f_1

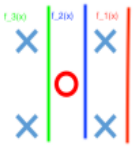

Figure 3: An example of to achieve zero training error by three decision stumps

Problem 5 (Gradient Descent Convergence) (2%)

Suppose the function $f : \mathbb{R}^n \to \mathbb{R}$ is differentiable. Also, f is β-smoothness and α-strongly convex.

$$\beta - smoothness : \beta > 0, \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n, \|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|_2 \leq \beta \|\boldsymbol{x} - \boldsymbol{y}\|_2$$

$$\alpha - strongly\ convex : \alpha > 0, \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n, f(\boldsymbol{x}) - f(\boldsymbol{y}) - \nabla f(\boldsymbol{y})^T(\boldsymbol{x} - \boldsymbol{y}) \geq \frac{\alpha}{2}\|\boldsymbol{x} - \boldsymbol{y}\|_2^2$$

Then we propose a gradient descent algorithm

1. Find a initial $\boldsymbol{\theta}^0$.
2. Let $\boldsymbol{\theta}^{n+1} = \boldsymbol{\theta}^n - \eta \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^n)\ \forall n \geq 0$, where $\eta = \frac{1}{\beta}$.

The following problems lead you to prove the gradient descent convergence.

1. Prove the property of β-smoothness function

$$\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n, f(\boldsymbol{x}) - f(\boldsymbol{y}) - \nabla f(\boldsymbol{y})^T(\boldsymbol{x} - \boldsymbol{y}) \leq \frac{\beta}{2}\|\boldsymbol{x} - \boldsymbol{y}\|_2^2$$

   (a) Define $g : \mathbb{R} \to \mathbb{R}, g(t) = f(\boldsymbol{y} + t(\boldsymbol{x} - \boldsymbol{y}))$. Show that $f(\boldsymbol{x}) - f(\boldsymbol{y}) = \int_0^1 g'(t)\, dt$.
   Solution: $f(\boldsymbol{x} - \boldsymbol{y}) = g(1) - g(0) = \int_0^1 g'(t) dt$.

   (b) Show that $g'(t) = \nabla f(\boldsymbol{y} + t(\boldsymbol{x} - \boldsymbol{y}))^T(\boldsymbol{x} - \boldsymbol{y})$.
   Solution: Let $\boldsymbol{z} = \boldsymbol{y} + t(\boldsymbol{x} - \boldsymbol{y})$.
   $g'(t) = \frac{dg(t)}{dt} = \frac{df(\boldsymbol{z})}{dt} = \sum_{i=1}^n \frac{df(\boldsymbol{z})}{\boldsymbol{z}_i}\frac{d\boldsymbol{z}_i}{dt} = \sum_{i=1}^n \frac{df(\boldsymbol{z})}{\boldsymbol{z}_i}\frac{d(\boldsymbol{y}_i + t(\boldsymbol{x}_i - \boldsymbol{y}_i))}{dt} = \sum_{i=1}^n \frac{df(\boldsymbol{z})}{\boldsymbol{z}_i}(\boldsymbol{x}_i - \boldsymbol{y}_i) = \nabla f(\boldsymbol{z})^T(\boldsymbol{x} - \boldsymbol{y}) = \nabla f(\boldsymbol{y} + t(\boldsymbol{x} - \boldsymbol{y}))^T(\boldsymbol{x} - \boldsymbol{y})$

   (c) Show that $|f(\boldsymbol{x}) - f(\boldsymbol{y}) - \nabla f(\boldsymbol{y})^T(\boldsymbol{x} - \boldsymbol{y})| \leq \int_0^1 |(\nabla f(\boldsymbol{y} + t(\boldsymbol{x} - \boldsymbol{y})) - \nabla f(\boldsymbol{y}))^T(\boldsymbol{x} - \boldsymbol{y})|\, dt$.
   Solution:

$$|f(\boldsymbol{x}) - f(\boldsymbol{y}) - \nabla f(\boldsymbol{y})^T(\boldsymbol{x} - \boldsymbol{y})| = |\int_0^1 g'(t)dt - \nabla f(\boldsymbol{y})^T(\boldsymbol{x} - \boldsymbol{y})|$$

$$= |\int_0^1 \nabla f(\boldsymbol{y} + t(\boldsymbol{x} - \boldsymbol{y}))^T(\boldsymbol{x} - \boldsymbol{y})dt - \int_0^1 \nabla f(\boldsymbol{y})^T(\boldsymbol{x} - \boldsymbol{y})dt|$$

$$= |\int_0^1 (\nabla f(\boldsymbol{y} + t(\boldsymbol{x} - \boldsymbol{y}))^T - \nabla f(\boldsymbol{y})^T)(\boldsymbol{x} - \boldsymbol{y})dt|$$

(d) By Cauchy-Schwarz inequality and the definition of β-smoothness, show that $|f(\boldsymbol{x}) - f(\boldsymbol{y}) - \nabla f(\boldsymbol{y})^T(\boldsymbol{x}-\boldsymbol{y})| \leq \frac{\beta}{2}\|\boldsymbol{x}-\boldsymbol{y}\|_2^2$, hence we get

$$f(\boldsymbol{x}) - f(\boldsymbol{y}) - \nabla f(\boldsymbol{y})^T(\boldsymbol{x}-\boldsymbol{y}) \leq \frac{\beta}{2}\|\boldsymbol{x}-\boldsymbol{y}\|_2^2$$

Solution: By Cauchy-Schwarz inequality,

$$\begin{aligned}
&|\int_0^1 (\nabla f(\boldsymbol{y}+t(\boldsymbol{x}-\boldsymbol{y}))^T - \nabla f(\boldsymbol{y})^T)(\boldsymbol{x}-\boldsymbol{y})dt| \\
&\leq \int_0^1 |(\nabla f(\boldsymbol{y}+t(\boldsymbol{x}-\boldsymbol{y}))^T - \nabla f(\boldsymbol{y})^T)(\boldsymbol{x}-\boldsymbol{y})|dt \\
&\leq \int_0^1 |(\nabla f(\boldsymbol{y}+t(\boldsymbol{x}-\boldsymbol{y})) - \nabla f(\boldsymbol{y}))^T(\boldsymbol{x}-\boldsymbol{y})|dt \\
&\leq \int_0^1 ||(\nabla f(\boldsymbol{y}+t(\boldsymbol{x}-\boldsymbol{y}))-\nabla f(\boldsymbol{y}))||_2||(\boldsymbol{x}-\boldsymbol{y})||_2 dt \quad \text{By Cauchy-Schwarz inequality} \\
&\leq \int_0^1 \beta||t(\boldsymbol{x}-\boldsymbol{y})||_2(\boldsymbol{x}-\boldsymbol{y})||_2 dt \leq \int_0^1 t\beta||(\boldsymbol{x}-\boldsymbol{y})||_2^2 dt = \beta||(\boldsymbol{x}-\boldsymbol{y})||_2^2 \int_0^1 tdt = \frac{\beta}{2}||(\boldsymbol{x}-\boldsymbol{y})||_2^2
\end{aligned}$$

2. Let $\boldsymbol{y} = \boldsymbol{x} - \frac{1}{\beta}\nabla f(\boldsymbol{x})$ and use 1., prove that

$$f(\boldsymbol{x} - \frac{1}{\beta}\nabla f(\boldsymbol{x})) - f(\boldsymbol{x}) \leq -\frac{1}{2\beta}\|\nabla f(\boldsymbol{x})\|_2^2$$

and

$$f(\boldsymbol{x}^*) - f(\boldsymbol{x}) \leq -\frac{1}{2\beta}\|\nabla f(\boldsymbol{x})\|_2^2,$$

where $\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} f(\boldsymbol{x})$.

Solution: By the fact that $f(\boldsymbol{y}) - f(\boldsymbol{x}) - \nabla f(\boldsymbol{x})^T(\boldsymbol{y}-\boldsymbol{x}) \leq \frac{\beta}{2}\|\boldsymbol{x}-\boldsymbol{y}\|_2^2$,

$$f(\boldsymbol{x}) - f(\boldsymbol{x} - \frac{1}{\beta}\nabla f(\boldsymbol{x})) - \nabla f(\boldsymbol{x})^T\frac{1}{\beta}\nabla f(\boldsymbol{x}) \leq \frac{1}{2\beta}\|\nabla f(\boldsymbol{x})\|_2^2.$$

Also, $\nabla f(\boldsymbol{x})^T\frac{1}{\beta}\nabla f(\boldsymbol{x}) = \frac{1}{\beta}\|\nabla f(\boldsymbol{x})\|_2^2$. Hence,

$$f(\boldsymbol{x} - \frac{1}{\beta}\nabla f(\boldsymbol{x})) - f(\boldsymbol{x}) \leq -\frac{1}{2\beta}\|\nabla f(\boldsymbol{x})\|_2^2$$

Owing to the fact that $f(\boldsymbol{x}^*) \leq f(\boldsymbol{x}) - \frac{1}{\beta}$, we get

$$f(\boldsymbol{x}^*) - f(\boldsymbol{x}) \leq -\frac{1}{2\beta}\|\nabla f(\boldsymbol{x})\|_2^2.$$

3. Show that $\forall n \geq 0$,

$$\|\boldsymbol{\theta}^{n+1} - \boldsymbol{\theta}^*\|_2^2 = \|\boldsymbol{\theta}^n - \boldsymbol{\theta}^*\|_2^2 + \eta^2 \|\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^n)\|_2^2 - 2\eta \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^n)^T (\boldsymbol{\theta}^n - \boldsymbol{\theta}^*),$$

where $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} f(\boldsymbol{\theta})$.

Solution:

$$\|\boldsymbol{\theta}^{n+1} - \boldsymbol{\theta}^*\|_2^2 = \|\boldsymbol{\theta}^n - \eta \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^n) - \boldsymbol{\theta}^*\|_2^2$$

$$(\boldsymbol{\theta}^n - \eta \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^n) - \boldsymbol{\theta}^*)^T (\boldsymbol{\theta}^n - \eta \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^n) - \boldsymbol{\theta}^*) = \|\boldsymbol{\theta}^n - \boldsymbol{\theta}^*\|_2^2 + \eta^2 \|\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^n)\|_2^2 - 2\eta \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^n)^T (\boldsymbol{\theta}^n - \boldsymbol{\theta}^*)$$

4. Use 2. and the definition of α-strongly convex to prove $\forall n \geq 0$

$$\|\boldsymbol{\theta}^{n+1} - \boldsymbol{\theta}^*\|_2^2 \leq (1 - \frac{\alpha}{\beta}) \|\boldsymbol{\theta}^n - \boldsymbol{\theta}^*\|_2^2,$$

where $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} f(\boldsymbol{\theta})$.

Solution: By rearranging the inequality we get in 2., $\eta^2 \|\nabla f(\boldsymbol{\theta}^n)\|_2^2 \leq 2\beta\eta^2 (f(\boldsymbol{\theta}^*) - f(\boldsymbol{\theta}^n))$.
Also, by the definition of α-strongly convex, $-2\eta \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}^n)^T (\boldsymbol{\theta}^n - \boldsymbol{\theta}^*) \leq -\alpha\eta ||\boldsymbol{\theta}^n - \boldsymbol{\theta}^*||_2^2 - 2\eta (f(\boldsymbol{\theta}^*) - f(\boldsymbol{\theta}^n))$ Combined the above results and $\eta = \frac{1}{\beta}$, we get

$$\|\boldsymbol{\theta}^{n+1} - \boldsymbol{\theta}^*\|_2^2 \leq \|\boldsymbol{\theta}^n - \boldsymbol{\theta}^*\|_2^2 + \frac{2}{\beta}(f(\boldsymbol{\theta}^*) - f(\boldsymbol{\theta}^n)) - \frac{\alpha}{\beta} ||\boldsymbol{\theta}^n - \boldsymbol{\theta}^*||_2^2 - \frac{2}{\beta}(f(\boldsymbol{\theta}^*) - f(\boldsymbol{\theta}^n)) = (1 - \frac{\alpha}{\beta}) \|\boldsymbol{\theta}^n - \boldsymbol{\theta}^*\|_2^2.$$

5. Use the above inequality to show that $\boldsymbol{\theta}^n$ will converge to $\boldsymbol{\theta}^*$ when n goes to infinity.

Solution: Because $\frac{\alpha}{\beta} > 0$, also, by the property of β-smoothness function and the definition of α strongly convex function, we get $\alpha \leq \beta$, which derives $\frac{\alpha}{\beta} < 1$ and $0 < 1 - \frac{\alpha}{\beta} < 1$. When n goes to infinity, $||\boldsymbol{\theta^n} - \boldsymbol{\theta^*}|| = 0$. Hence, $\boldsymbol{\theta^n}$ will converge to $\boldsymbol{\theta^*}$

Version Description

1. First Edition: Finish Problem Solution 1 to 5