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

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

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

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

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

1. $w_1 = f_1(x)$，其中 $w_1 \in \mathbb{R}^{n_1}$
$w_2 = f_2(w_1)$，其中 $w_2 \in \mathbb{R}^{n_2}$
$w_3 = f_3(w_2)$，其中 $w_3 \in \mathbb{R}^{n_3}$
$w_4 = f_4(w_3) = f(x)$，其中 $w_4 \in \mathbb{R}$
chain rule: $\frac{\partial f}{\partial x} = \frac{\partial w_4}{\partial x} = \frac{\partial w_4}{\partial w_3}\frac{\partial w_3}{\partial w_2}\frac{\partial w_2}{\partial w_1}\frac{\partial w_1}{\partial x}$ #
$\frac{\partial w_4}{\partial w_3}: 1 \times n_3$, $\frac{\partial w_3}{\partial w_2} = n_3 \times n_2$, $\frac{\partial w_2}{\partial w_1} = n_2 \times n_1$, $\frac{\partial w_1}{\partial x} = n_1 \times n$

Forward Mode
$w_1 = f_1(x)$, $w_1 \in \mathbb{R}^{n_1}$
方向導數：$\dot{w}_1 = \frac{\partial w_1}{\partial x}\dot{x}$
$w_2 = f_2(w_1)$, $w_2 \in \mathbb{R}^{n_2}$
方向導數：$\dot{w}_2 = \frac{\partial w_2}{\partial w_1}\dot{w}_1$
$w_3 = f_3(w_2)$, $w_3 \in \mathbb{R}^{n_3}$
方向導數：$\dot{w}_3 = \frac{\partial w_3}{\partial w_2}\dot{w}_2$
$w_4 = f_4(w_3)$, $w_4 \in \mathbb{R}$
方向導數：$\dot{w}_4 = \frac{\partial w_4}{\partial w_3}\dot{w}_3$
⇒ $\dot{w}_4$ 是 f 對 x 的方向導數：$\dot{w}_4 = \frac{\partial f}{\partial x}\dot{x}$

Reverse Mode
令 $\bar{w}_4 = \frac{\partial w_4}{\partial w_4} = 1$ ⇒ output 對自身的梯度
$\bar{w}_3 = \left(\frac{\partial w_4}{\partial w_3}\right)^T \bar{w}_4$
$\bar{w}_2 = \left(\frac{\partial w_3}{\partial w_2}\right)^T \bar{w}_3$
$\bar{w}_1 = \left(\frac{\partial w_2}{\partial w_1}\right)^T \bar{w}_2$
$\bar{x} = \left(\frac{\partial w_1}{\partial x}\right)^T \bar{w}_1$
⇒ $\bar{x}$ 是 f 對 x 的梯度 $\bar{x} = \left(\frac{\partial f}{\partial x}\right)^T$

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. Some useful references:

- Automatic Differentiation in Machine Learning: a Survey
- CSC321 Lecture 10: Automatic Differentiation
- Wikipedia

2.

pf:

Forward Mode:

$\frac{\partial w_1}{\partial x}\dot{x}$: $\frac{\partial w_1}{\partial x}$: $n_1 \times n$ 矩陣

complexity : $O(n_1 n)$

$\dot{w}_2 = \frac{\partial w_2}{\partial w_1}\dot{w}_1$: $\frac{\partial w_2}{\partial w_1}$: $n_2 \times n_1$ 矩陣

complexity : $O(n_2 n_1)$

$\dot{w}_3 = \frac{\partial w_3}{\partial w_2}\dot{w}_2$: $\frac{\partial w_3}{\partial w_2}$: $n_3 \times n_2$ 矩陣

complexity : $O(n_3 n_2)$

$\dot{w}_4 = \frac{\partial w_4}{\partial w_3}\dot{w}_3$: $\frac{\partial w_4}{\partial w_3}$: $1 \times n_3$ 的行向量

complexity : $O(n_3)$

$O(n_1 n) + O(n_2 n_1) + O(n_3 n_2) + O(n_3)$

Reverse Mode

$\bar{w}_3 = \left(\frac{\partial w_4}{\partial w_3}\right)^T \bar{w}_4 \Rightarrow \frac{\partial w_4}{\partial w_3}$ 是 $1 \times n_3$ 向量 $\Rightarrow O(n_3)$

$\bar{w}_2 = \left(\frac{\partial w_3}{\partial w_2}\right)^T \bar{w}_3 \Rightarrow \frac{\partial w_3}{\partial w_2}$ 是 $n_3 \times n_2$ 矩陣 $\Rightarrow O(n_2 n_3)$

$\bar{w}_1 = \left(\frac{\partial w_2}{\partial w_1}\right)^T \bar{w}_2 \Rightarrow \frac{\partial w_2}{\partial w_1}$ 是 $n_2 \times n_1$ 矩陣 $\Rightarrow O(n_1 n_2)$

$\bar{x} = \left(\frac{\partial w_1}{\partial x}\right)^T \bar{w}_1 \Rightarrow \frac{\partial w_1}{\partial x}$ 是 $n_1 \times n$ 矩陣 $\Rightarrow O(n n_1)$

$\Rightarrow O(n_3) + O(n_2 n_3) + O(n_1 n_2) + O(n n_1)$

當函數輸入維度遠大於輸出維度 ($n >> 1$)，reverse mode 效率好

1. forward mode 需要對每個輸入維度獨立計算導數，共要 n 次傳播

2. reverse mode 只需一次反向傳播即可得所有輸入維度的導數

由上向的推導，forward mode 需逐層進行矩陣向量乘法，reverse mode 進行 矩陣向量的轉置乘法，由於輸出是 scalar，reverse mode 的計算複雜度不隨輸入維度線性增長

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 algorithm can be written as below (see Algorithm 1):

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 $\frac{\partial \ell}{\partial x_i}$, $\frac{\partial \ell}{\partial \gamma}$, $\frac{\partial \ell}{\partial \beta}$.

$\frac{\partial \ell}{\partial y_i}$ (已知)

根據 $y_i = \gamma \hat{x}_i + \beta$

$\frac{\partial \ell}{\partial \gamma} = \sum_{i=1}^m \frac{\partial \ell}{\partial y_i} \cdot \frac{\partial y_i}{\partial \gamma} = \sum_{i=1}^m \frac{\partial \ell}{\partial y_i} \cdot \hat{x}_i$ #

$\frac{\partial \ell}{\partial \beta} = \sum_{i=1}^m \frac{\partial \ell}{\partial y_i} \cdot \frac{\partial y_i}{\partial \beta} = \sum_{i=1}^m \frac{\partial \ell}{\partial y_i}$ #

$\boxed{\frac{\partial \ell}{\partial \hat{x}_i}}$: $\frac{\partial \ell}{\partial \hat{x}_i} = \frac{\partial \ell}{\partial y_i} \cdot \frac{\partial y_i}{\partial \hat{x}_i} = \frac{\partial \ell}{\partial y_i} \cdot \gamma$

$\frac{\partial \hat{x}_i}{\partial \sigma_B^2}$:

$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} = (x_i - \mu_B) \cdot (\sigma_B^2 + \epsilon)^{-1/2}$

$\Rightarrow \frac{\partial \hat{x}_i}{\partial \sigma_B^2} = (x_i - \mu_B) \cdot \left(-\frac{1}{2}(\sigma_B^2 + \epsilon)^{-3/2}\right)$

$= -\frac{1}{2}(x_i - \mu_B)(\sigma_B^2 + \epsilon)^{-3/2}$

$\frac{\partial \hat{x}_i}{\partial \mu_B}$: $\frac{\partial \hat{x}_i}{\partial \mu_B} = \frac{\partial}{\partial \mu_B}\left(\frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}\right) = -\frac{1}{\sqrt{\sigma_B^2 + \epsilon}}$

$\boxed{\frac{\partial \ell}{\partial \sigma_B^2}}$: $\frac{\partial \ell}{\partial \sigma_B^2} = \sum_{i=1}^m \frac{\partial \ell}{\partial \hat{x}_i} \cdot \frac{\partial \hat{x}_i}{\partial \sigma_B^2} = -\frac{1}{2}\sum_{i=1}^m \frac{\partial \ell}{\partial \hat{x}_i}(x_i - \mu_B)(\sigma_B^2 + \epsilon)^{-3/2}$

$\boxed{\frac{\partial \ell}{\partial \mu_B}}$: $\frac{\partial \ell}{\partial \mu_B} = \sum_{i=1}^m \frac{\partial \ell}{\partial \hat{x}_i} \cdot \frac{\partial \hat{x}_i}{\partial \mu_B} = -\frac{1}{\sqrt{\sigma_B^2 + \epsilon}}\sum_{i=1}^m \frac{\partial \ell}{\partial \hat{x}_i}$

$\frac{\partial \ell}{\partial x_i} = \frac{\partial \ell}{\partial \hat{x}_i} \cdot \frac{\partial \hat{x}_i}{\partial x_i} + \frac{\partial \ell}{\partial \mu_B} \cdot \frac{\partial \mu_B}{\partial x_i} + \frac{\partial \ell}{\partial \sigma_B^2} \cdot \frac{\partial \sigma_B^2}{\partial x_i}$

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

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

$= \frac{\partial \ell}{\partial \hat{x}_i} \cdot \frac{1}{\sqrt{\sigma_B^2 + \epsilon}} - \left(\sum_{i=1}^m \frac{\partial \ell}{\partial \hat{x}_i} \cdot \frac{1}{\sqrt{\sigma_B^2 + \epsilon}}\right) \cdot \frac{1}{m} - \left(\frac{1}{2}\sum_{i=1}^m \frac{\partial \ell}{\partial \hat{x}_i}(x_i - \mu_B)(\sigma_B^2 + \epsilon)^{-3/2}\right) \cdot \frac{2(x_i - \mu_B)}{m}$

$= \frac{1}{\sqrt{\sigma_B^2 + \epsilon}}\left(\frac{\partial \ell}{\partial \hat{x}_i} - \frac{1}{m}\sum_{i=1}^m \frac{\partial \ell}{\partial \hat{x}_i} - \frac{(x_i - \mu_B)}{(\sigma_B^2 + \epsilon)} \cdot \frac{1}{m}\sum_{i=1}^m \frac{\partial \ell}{\partial \hat{x}_i}(x_i - \mu_B)\right)$

$= \frac{1}{\sqrt{\sigma_B^2 + \epsilon}}\left(\frac{\partial \ell}{\partial \hat{x}_i} - \frac{1}{m}\sum_{i=1}^m \frac{\partial \ell}{\partial \hat{x}_i} - \hat{x}_i \frac{1}{m}\sum_{i=1}^m \frac{\partial \ell}{\partial \hat{x}_i}\hat{x}_i\right)$ #

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

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

$g_1^k(x) = 0, \quad \forall k \in [1, k], \quad \forall x \in X$

negative gradient of the loss function : $r_{ik} = -\dfrac{\partial L}{\partial g_T^k(x_i)}$

$$\frac{\partial L}{\partial g_T^k(x_i)} = \exp\left(\frac{1}{k-1} S_i\right) \cdot \left(\frac{1}{k-1} \delta_{k \neq \hat{y}_i} - \frac{k-1}{k-1} \delta_{k = \hat{y}_i}\right)$$

$S_i = \sum_{l \neq \hat{y}_i} \left(g_T^l(x_i) - g_T^{\hat{y}_i}(x_i)\right)$, δ_p = 指示函数 $p = 1$ (true) $p = 0$ (False)

$k = \hat{y}_i$: $r_{i\hat{y}_i} = -\left(-\exp\left(\frac{1}{k-1} S_i\right)\right) = \exp\left(\frac{1}{k-1} S_i\right)$

$k \neq \hat{y}_i$: $r_{ik} = -\left(\frac{1}{k-1} \exp\left(\frac{1}{k-1} S_i\right)\right) = -\frac{1}{k-1} \exp\left(\frac{1}{k-1} S_i\right)$

$w_i = \exp\left(\frac{1}{k-1} S_i\right)$, 最小化加权错误率 $e_t = \frac{\sum_{i=1}^m w_i \cdot 1\{f_t(x_i) \neq \hat{y}_i\}}{\sum_{i=1}^m w_i}$

找最佳的 α_t , 使得沿着 f_t 的方向，损失函数最小

$g_{T+1}^k(x) = g_T^k(x) + \alpha_t f_t^k(x)$, $L(\alpha_t) = \sum_{i=1}^m \exp\left(\frac{1}{k-1}(S_i + \alpha_t \Delta_i)\right)$, $\Delta_i = \sum_{k \neq \hat{y}_i} \left(f_t^k(x_i) - f_t^{\hat{y}_i}(x_i)\right) = \begin{cases} -(k-1), & f_t(x_i) = \hat{y}_i \\ 1, & f_t(x_i) \neq \hat{y}_i \end{cases}$

if $f_t(x_i) = \hat{y}_i \Rightarrow L_1 = \sum_{i : f_t(x_i) = \hat{y}_i} w_i \cdot \exp\left(-\alpha_t \cdot (k-1)/(k-1)\right) = \sum_{i : f_t(x_i) = \hat{y}_i} w_i \cdot \exp(-\alpha_t)$

if $f_t(x_i) \neq y_i \Rightarrow L_2 = \sum_{i : f_t(x_i) \neq \hat{y}_i} w_i \cdot \exp(\alpha_t \cdot 1/(k-1)) = \sum_{i : f_t(x_i) \neq \hat{y}_i} w_i \cdot \exp\left(\frac{\alpha_t}{k-1}\right)$

$\Rightarrow L(\alpha_t) = L_1 + L_2$

最小化 $L(\alpha_t)$ 求解 $\alpha_t \Rightarrow \frac{dL}{d\alpha_t} = -L_1 \exp(-\alpha_t) + \frac{1}{k-1} L_2 \cdot \exp\left(\frac{\alpha_t}{k-1}\right) = 0 \Rightarrow -L_1 \cdot \exp(-\alpha_t) + \frac{L_2}{k-1} \cdot \exp\left(\frac{\alpha_t}{k-1}\right) = 0$

$\Rightarrow -(k-1)L_1 + L_2 \exp\left(\alpha_t \left(\frac{k}{k-1}\right)\right) = 0 \Rightarrow \exp\left(\alpha_t \left(\frac{k}{k-1}\right)\right) = \frac{(k-1)L_1}{L_2} \Rightarrow \alpha_t = \frac{k-1}{k} \ln\left(\frac{(k-1)L_1}{L_2}\right)$

$L_1 : \sum_{i : f_t(x_i) = \hat{y}_i} w_i$

$L_2 : \sum_{i : f_t(x_i) \neq \hat{y}_i} w_i$

$g_{T+1}^k(x) = g_T^k(x) + \alpha_t f_t^k(x)$

- Sample weights $w_i = \exp\left(\frac{1}{k-1} \sum_{k \neq \hat{y}_i} \left[g_T^k(x_i) - g_T^{\hat{y}_i}(x_i)\right]\right)$

  fit f_t to minimize weighted misclassification error using these weights.

  $\alpha_t = \frac{k-1}{k} \left[\ln\left((k-1) \sum_{i : f_t(x_i) = \hat{y}_i} w_i\right) - \ln\left(\sum_{i : f_t(x_i) \neq \hat{y}_i} w_i\right)\right]$

Problem 4 (AdaBoosting)(0.5%)

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

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.

(a) 在第一次迭代結束時，只有被錯誤分類的樣本權重增加，初始時所有樣本的權重相等

第一次迭代中，最佳AdaBoost將所有樣本預測為藍色叉號，可最小化錯誤分類的樣本數量

∴只有紅色圈圈的權重有增加

(b) 4個藍叉：+1（樣本1~4）

1個紅圈：-1（樣本5）

初始權重：0.2

第一次迭代：

h_1：弱分類器：所有樣本都預測為+1

$\epsilon_1 = w_5^{(1)} = 0.2$

$\alpha_1 = \frac{1}{2}\ln\left(\frac{1-\epsilon_1}{\epsilon_1}\right) = \frac{1}{2}\ln\left(\frac{0.8}{0.2}\right) = 0.6931$

更新：錯誤：$w_5^{(2)} = w_5^{(1)} \times e^{\alpha_1} = 0.2 \times e^{0.6931} = 0.4$

正確：$w_i^{(2)} = w_i^{(1)} \times e^{-\alpha_1} = 0.2 \times e^{-0.6931} = 0.1$

總權重：$0.4 + 4\times 0.1 = 0.8$

樣本1-4：$w_i^{(2)} = \frac{0.1}{0.8} = 0.125$　樣本5：$w_5^{(2)} = \frac{0.4}{0.8} = 0.5$

第二次迭代：

h_2：紅圈正確分類為-1，誤分2個藍叉（含樣本1和2）

$\epsilon_2 = w_1^{(2)} + w_2^{(2)} = 0.125 + 0.125 = 0.25$

$\alpha_2 = \frac{1}{2}\ln\left(\frac{1-\epsilon_2}{\epsilon_2}\right) = \frac{1}{2}\ln 3 = 0.5493$

更新：錯誤：$w_i^{(3)} = w_i^{(2)} \times e^{\alpha_2} = 0.125 \times 1.732 = 0.2165$

正確：

樣本3和4：$w_i^{(3)} = w_i^{(2)} \times e^{-\alpha_2} = 0.125 \times 0.577 = 0.0722$

樣本5：$w_5^{(3)} = w_5^{(2)} \times e^{-\alpha_2} = 0.5 \times 0.577 = 0.2887$

總權重：$2\times 0.2165 + 2\times 0.0722 + 0.2887 = 0.8661$

樣本1和2：$w_i^{(3)} = \frac{0.2165}{0.8661} = 0.25$　樣本3和4：$w_i^{(3)} = \frac{0.0722}{0.8661} = 0.0833$

樣本5：$w_5^{(3)} = \frac{0.2887}{0.8661} = 0.3334$

第三次迭代：

h_3：重點關注樣本1和2，可能誤分樣本5

ϵ_3：$\epsilon_3 = w_5^{(3)} = 0.3334$

α_3：$\alpha_3 = \frac{1}{2}\ln\left(\frac{1-\epsilon_3}{\epsilon_3}\right) = \frac{1}{2}\ln\left(\frac{0.6666}{0.3334}\right) = \frac{1}{2}\ln 2 = 0.3466$

更新：錯誤（樣本5）：$w_5^{(4)} = w_5^{(3)} \times e^{\alpha_3} = 0.3334 \times e^{0.3466} = 0.4715$

正確：

樣本1和2：$w_i^{(4)} = 0.25 \times e^{-0.3466} = 0.1768$

樣本3和4：$w_i^{(4)} = 0.0833 \times e^{-0.3466} = 0.0589$

總權重：$0.4715 + 2\times 0.1768 + 2\times 0.0589 = 0.9429$

新權重分佈：樣本1和2：$w_i^{(4)} = \frac{0.1768}{0.9429} = 0.1877$

樣本3和4：$w_i^{(4)} = \frac{0.0589}{0.9429} = 0.0625$

樣本5：$w_5^{(4)} = \frac{0.4715}{0.9429} = 0.5$

最終強分類器：$H(x) = \mathrm{sign}(\alpha_1 h_1(x) + \alpha_2 h_2(x) + \alpha_3 h_3(x))$

樣本1和2（+1）：

$h_1(x) = +1$，$h_2(x) = -1$（誤分類），$h_3(x) = +1$

$\alpha_1(+1) + \alpha_2(-1) + \alpha_3(+1) = 0.4934 > 0$　$H(x) = +1 \Rightarrow$ 分類正確

樣本3和4（+1）：

$h_1(x) = +1$，$h_2(x) = +1$，$h_3(x) = -1$

$\alpha_1(+1) + \alpha_2(+1) + \alpha_3(-1) = 0.8958 > 0$　$H(x) = +1 \Rightarrow$ 分類正確

樣本5（-1）：

$h_1(x) = +1$（誤分類），$h_2(x) = -1$，$h_3(x) = -1$

$\alpha_1(+1) + \alpha_2(-1) + \alpha_3(-1) = -0.2038 < 0$　$H(x) = -1 \Rightarrow$ 分類正確

$\Rightarrow$ 3次迭代，組合成的強分類器可正確分類所有訓練樣本，達0錯誤

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?)

$E_{train} = \frac{1}{N}\sum_{i=1}^{N} I(y_i \neq sign(F(x_i)))$ ，$F(x)=\sum_{t=1}^{T}\alpha_t h_t(x)$ 为组合分类器，α_t 表第 t 次的分类器权重，$h_t(x)$ 表第 t 个的分类器

$E_{exp} = \sum_{i=1}^{N} e^{-y_i F(x_i)}$ 0-1为损失上界：$I(y_i \neq sign(F(x_i))) \leq e^{-y_i F(x_i)} \Rightarrow E_{train} \leq \frac{1}{N} E_{exp}$

AdaBoost 中：$E_{exp} = \prod_{t=1}^{T} Z_t$ $Z_t = 2\sqrt{\epsilon_t(1-\epsilon_t)}$

由于 $\epsilon_t \leq \frac{1}{2} - \gamma \Rightarrow 1-\epsilon_t \geq \frac{1}{2}+\gamma$

$Z_t \leq 2\sqrt{(\frac{1}{2}-\gamma)(\frac{1}{2}+\gamma)} = 2\sqrt{\frac{1}{4}-\gamma^2} = \sqrt{1-4\gamma^2} \Rightarrow \sqrt{1-4\gamma^2} \leq e^{-2\gamma^2}$

$\because 0 < \gamma < \frac{1}{2} \Rightarrow \sqrt{1-4\gamma^2} \leq e^{-2\gamma^2}$ $\therefore Z_t \leq 2 \times \frac{1}{2} e^{-2\gamma^2} = e^{-2\gamma^2}$

$\because$ 每次 $Z_t \leq e^{-2\gamma^2}$ $\therefore E_{exp} = \prod_{t=1}^{T} Z_t \leq (e^{-2\gamma^2})^T = e^{-2\gamma^2 T}$

$E_{train} \leq \frac{1}{N} E_{exp} \leq \frac{1}{N} e^{-2\gamma^2 T}$ （$E_{train} \leq \frac{1}{N}$，所有样本都被正确分类）

$e^{-2\gamma^2 T} \leq 1 \Rightarrow -2\gamma^2 T \leq -\ln N \Rightarrow T \geq \frac{\ln N}{2\gamma^2}$

至少 $T \geq \frac{\ln N}{2\gamma^2}$ 次迭代，才能在 N 个训练样本上达 0 训练误差

Problem 5 (Gradient Descent Convergence) (1.5%)

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

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

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

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

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

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

   (b) Show that $g'(t) = \nabla f(\boldsymbol{x} + t(\boldsymbol{y} - \boldsymbol{x}))^T (\boldsymbol{y} - \boldsymbol{x})$.

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

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

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

1. (a). $g(t) = f(x + t(y-x))$

$\Rightarrow \begin{cases} g(0) = f(x + 0(y-x)) = f(x) \\ g(1) = f(x + 1(y-x)) = f(y) \end{cases}$ $\Rightarrow f(y) - f(x) = g(1) - g(0) = \int_0^1 g'(t)\, dt$

(b) 链式: $g'(t) = \frac{d}{dt} f(x + t(y-x)) = \nabla f(x + t(y-x))^T \cdot \frac{d}{dt}(x + t(y-x)) = \nabla f(x + t(y-x))^T (y-x)$

(c) 由(a),(b)知: $f(y) - f(x) = \int_0^1 \nabla f(x + t(y-x))^T (y-x)\, dt$

$\Rightarrow f(y) - f(x) = \nabla f(x)^T (y-x) + \int_0^1 [\nabla f(x + t(y-x))^T (y-x) - \nabla f(x)^T (y-x)]\, dt$

$f(y) - f(x) - \nabla f(x)^T (y-x) = \int_0^1 [\nabla f(x + t(y-x)) - \nabla f(x)]^T (y-x)\, dt$

$|f(y) - f(x) - \nabla f(x)^T (y-x)| \leq \int_0^1 |[\nabla f(x + t(y-x)) - \nabla f(x)]^T (y-x)|\, dt$

(d) Apply the Cauchy-Schwarz inequality to the integrand:

$|[\nabla f(x + t(y-x)) - \nabla f(x)]^T (y-x)| \leq \|\nabla f(x + t(y-x)) - \nabla f(x)\|_2 \cdot \|y-x\|_2$

β-smoothness property:

$\|\nabla f(x + t(y-x)) - \nabla f(x)\|_2 \leq \beta \|x + t(y-x) - x\|_2 = \beta t \|y-x\|_2$

$\Rightarrow |[\nabla f(x + t(y-x)) - \nabla f(x)]^T (y-x)| \leq \beta t \|y-x\|_2^2$

$|f(y) - f(x) - \nabla f(x)^T (y-x)| \leq \int_0^1 \beta t \|y-x\|_2^2\, dt = \beta \|y-x\|_2^2 \int_0^1 t\, dt$ $\quad \int_0^1 t\, dt = \left[\frac{t^2}{2}\right]_0^1 = \frac{1}{2}$

$\Rightarrow |f(y) - f(x) - \nabla f(x)^T (y-x)| \leq \beta \|y-x\|_2^2 \cdot \frac{1}{2} = \frac{\beta}{2} \|y-x\|_2^2$

∵ 左式绝非负的: $f(y) - f(x) - \nabla f(x)^T (y-x) \leq \frac{\beta}{2} \|y-x\|_2^2$

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

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

and

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

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

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

2. $y-x = (x-\frac{1}{\beta}\nabla f(x)) - x = -\frac{1}{\beta}\nabla f(x)$

$\nabla f(x)^T(y-x) = \nabla f(x)^T(-\frac{1}{\beta}\nabla f(x)) = \frac{1}{\beta}\|\nabla f(x)\|_2^2$

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

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

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

$f(x^*) \leq f(y)$ ∵ x^* 是 f 的全局最小值點，對任何 y 都有 $f(x^*) \leq f(y)$

$\exists\, y = x - \frac{1}{\beta}\nabla f(x)$: $f(x^*) \leq f(x - \frac{1}{\beta}\nabla f(x))$

由前面的推導知: $f(x-\frac{1}{\beta}\nabla f(x)) - f(x) \leq -\frac{1}{2\beta}\|\nabla f(x)\|_2^2$

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

3. 梯度下降 : $\theta^{n+1} = \theta^n - \eta\nabla_\theta f(\theta^n)$

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

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

$= (\theta^n - \theta^*)^T(\theta^n - \theta^*) - 2\eta(\theta^n - \theta^*)^T\nabla_\theta f(\theta^n) + \eta^2(\nabla_\theta f(\theta^n))^T\nabla_\theta f(\theta^n)$

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

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

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

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

4. 梯度下降: $\theta^{n+1} = \theta^n - \eta \nabla f(\theta^n)$ $\eta = \frac{1}{\beta}$

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

α-strongly convex:

$f(y) \geq f(x) + \nabla f(x)^T(y-x) + \frac{\alpha}{2}\|y-x\|_2^2$ 令 $x=\theta^*, y=\theta_n$ ⟹ 由于 $\nabla f(\theta^*)=0$ 得 $f(\theta^n) \geq f(\theta^*) + 0 + \frac{\alpha}{2}\|\theta^n - \theta^*\|_2^2$

$\Rightarrow f(\theta^n) - f(\theta^*) \geq \frac{\alpha}{2}\|\theta^n - \theta^*\|_2^2$

β-smoothness: $\|\nabla f(\theta^n)\|_2^2 \leq 2\beta(f(\theta^n) - f(\theta^*))$, $\because \nabla f(\theta^*) = 0 \Rightarrow f(\theta^{n+1}) \leq f(\theta^n) - \frac{1}{2\beta}\|\nabla f(\theta^n)\|_2^2$

$\|\nabla f(\theta^n)\|_2^2 \geq 2\alpha(f(\theta^n) - f(\theta^*))$

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

$\because f(\theta^n) - f(\theta^*) \geq \frac{\alpha}{2}\|\theta^n - \theta^*\|_2^2 \therefore \|\theta^n - \theta^*\|_2^2 \leq \frac{2}{\alpha}(f(\theta^n) - f(\theta^*))$

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

$\forall n \geq 0 : \|\theta^{n+1} - \theta^*\|_2^2 \leq (1 - \frac{\alpha}{\beta})\|\theta^n - \theta^*\|_2^2$

5. $\|\theta_{n+1} - \theta^*\|_2^2 \leq (1 - \frac{\alpha}{\beta})\|\theta_n - \theta^*\|_2^2$ $\because 0 < \frac{\alpha}{\beta} \leq 1$, $0 \leq q = 1 - \frac{\alpha}{\beta} < 1$

$\|\theta_n - \theta^*\|_2^2 \leq q^n \|\theta^0 - \theta^*\|_2^2$

$n \to \infty$, $q^n \to 0$ $\lim_{n\to\infty} \|\theta_n - \theta^*\|_2^2 = 0$

θ_n converges to θ^*