1 Convexity and Local Minimas

1. Let $\mathcal{H} = \{f_w(x) = \langle x, w\rangle | w \in \mathbb{R}^n\}$ be a hypothesis class of linear functions. Let $\ell(f_w(x), y) := \mathbf{1}[y f_w(x) < 0]$ be the 0-1 loss. The empirical error: $L_S[w] := \frac{1}{m}\sum_{i=1}^{m} \ell(f_w(x_i), y_i)$.

Construct an example showing that linear classification with the 0-1 loss may suffer from local minima. Namely, construct a dataset $S = \{(x_i, y_i)\}_{i=1}^{m}$ (where $y_i \in \{\pm 1\}$ and say $x_i \in \mathbb{R}^2$), for which there exists a vector w and some $\epsilon > 0$, such that:

- For any w', such that, $\|w - w'\| \leq \epsilon$, we have: $L_S[w] \leq L_S[w']$.
- There exists some w^* such that, $L_S[w^*] < L_S[w]$. This means w is not the global minima of L_S.

2. Let $\mathcal{H} = \{f_w(x) = \langle x, w\rangle | w \in \mathbb{R}^n\}$ be a hypothesis class of linear functions. Assume that $\|x\| \leq B$ and $\|w\| \leq B$, for some scalar $B > 0$. Labels are ±1. Let the loss function be the logistic loss: $\ell(f_w(x), y) = \log(1+\exp(-y \cdot f_w(x))$. Show that $\ell(f_w(x), y)$ is convex and Lipschitz with respect to w (for any $x \in \mathbb{R}^n$ such that $\|x\| \leq B$ and $y \in \{\pm 1\}$). Specify the parameter of Lipschitzness.

3. Show that neural networks with the logistic loss is a non-convex learning task. Namely, take a hypothesis class $\mathcal{H} = \{f_w(x) = W_d \circ \sigma \circ \cdots \circ \sigma \circ W_1 : \mathbb{R}^n \to \mathbb{R} | w = (W_1, \dots, W_d)\}$ of neural networks and construct a dataset $S = \{(x_i, y_i)\}_{i=1}^{m}$. Let the loss function be defined as $\ell(f_w(x), y) = \log(1 + \exp(-y \cdot f_w(x))$. Show that the empirical error $\frac{1}{m}\sum_{i=1}^{m} \ell(f_w(x_i), y_i)$ is non-convex with respect to w.

Clue: if $f(x, y)$ is convex in x, y (simultaneously), then it is convex in x for any fixed y.

Note: you can choose the the activation function to be ReLU, tanh or sigmoid. You can assume each layer is of size 2 (including the input dimension). You can choose S and m. Partial score will be given if you fix the number of layers.

2 Backpropagation

1. Run (on paper) step-by-step the Backpropagation algorithm as efficient as you can for calculating the gradient of: $\|W_3(\sigma(W_2(\sigma(W_1 x)))) - y\|_2^2$ with respect to W_1, W_2, W_3 (for arbitrary x and y). Here, σ is the sigmoid activation function.

Explain each step, the computational cost of each step (in terms of runtime and space complexities) and prove any necessary identities (excluding anything we proved in class).

3 Stochastic Gradient Descent

In this exercise we prove the analysis of SGD from class.

Step 1: prove the following lemma.

Lemma 1. *Let $v_1, \ldots, v_T$ be an arbitrary sequence of vectors. Any algorithm with an initialization $w^{(1)} = 0$ and an update rule of the form*

$$w^{(t+1)} = w^{(t)} - \mu v_t \tag{1}$$

satisfies:

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

Step 2: prove the following lemma.

Lemma 2. *Let $g(w) = \frac{1}{m} \sum_{i=1}^{m} g_i(w)$ be a function. Assume that g_i is convex, continuosly differentiable, ρ-Lipschitz, and let $w^* = \arg\min_{\|w\| \leq B} g(w)$. If we run the SGD algorithm on g for T iterations with $\mu = \sqrt{\frac{B^2}{\rho^2 T}}$, then, we have:*

$$\mathbb{E}_{v_1,\ldots,v_T} \left[\frac{1}{T} \sum_{t=1}^{T} \langle w^{(t)} - w^*, v_t \rangle \right] \leq \frac{B\rho}{\sqrt{T}} \tag{3}$$

Here, v_t is the gradient used by the SGD in iteration number t.

Step 3: prove the following lemma.

Lemma 3. *Let $g(w) = \frac{1}{m} \sum_{i=1}^{m} g_i(w)$ be a function. Assume that g_i is convex, continuosly differentiable and let $w^* = \arg\min_{\|w\| \leq B} g(w)$. If we run the SGD algorithm on g for T steps, then:*

$$\mathbb{E}_{v_1,\ldots,v_T} \left[\sum_{t=1}^{T} (g(w^{(t)}) - g(w^*)) \right] \leq \mathbb{E}_{v_1,\ldots,v_T} \left[\sum_{t=1}^{T} \langle w^{(t)} - w^*, v_t \rangle \right] \tag{4}$$

Here, v_t is the gradient used by the SGD in iteration number t.

Step 4: conclude that the theorem from class regarding SGD holds.