Due Date : March 1st, 23:00 2024

Instructions

- *For all questions, show your work! Any problem without work shown will get no marks regardless of the correctness of the final answer.*
- *Use a document preparation system such as LaTeX. If hand-written, you may lose marks if your writing is illegible without any possibility of regrade.*
- *Submit your answers electronically via the course GradeScope.*
- *Please sign the agreement below.*
- *It is your responsibility to follow updates to the assignment after release. All changes will be visible on Overleaf and Piazza, so no additional time or exceptions will be granted.*
- *TA for this assignment is (theoretical part) :* ***Jerry Huang****.*

I acknowledge I have read the above instructions and will abide by them throughout this assignment. I further acknowledge that any assignment submitted without the following form completed may result in no marks being given for this portion of the assignment.

Signature : _____

Name : _____

Matricule : _____

Basics (14 points)

Question 1.

Answer 1.

1.
   - This question changed at one point so there are two potential answers.
     — (Stricly increasing) Sufficient but not necessary. It is convex as long as it's always non-decreasing. The sufficiency direction is easy as you can take the convex combination of two points and the derivative to show convexity. To disprove necessity, simply use a horizontal line.
     — (Monotically increasing) Sufficient and necessary. Sufficiency can be proved in the same way. Necessity can be proved by contradiction (take a function that decreases at some point and use the convex combination to show that the function is not convex at that point).
   - Sufficient and necessary. Sufficiency is easy to prove, as this leads to monotonic non-decreasing derivative. Necessity can be shown in a two step process by contradiction for each part : 1) show that derivative has to be monotonic non-decreasing and 2) for this to be the case, the second derivative has to be non-negative.
   - Necessary but not sufficient. Necessity is easy to prove if you show the function must be monotonically increasing. Sufficincy can be negated by using a sin or cos function.
2. Both are convex. Proof : Use the previous statement to show the derivative is greater than 0.
3. Not Lipschitz (will accept ∞ as well) and 1-Lipschitz. Proof :

   For $f(x) = x^2$, take $x_1 = 0$ and $x_2 = 1 + \rho$. Then

   $$f(x_2) - f(x_1) = (1+\rho)^2 > \rho(1+\rho) = \rho|x_2 - x_1|$$

   This is ρ-Lipshitz over the set $C = \{x : |x| \leq \rho/2\}$. For any $x_1, x_2 \in C$

   $$|x_1^2 - x_2^2| = |x_1 + x_2||x_1 - x_2| \leq \rho|x_1 - x_2|$$

   For $f(x) = \frac{\exp(x)}{1+\exp(x)}$

   $$|f'(x)| = \left|\frac{\exp(x)}{1+\exp(x)}\right| = \left|\frac{1}{1+\exp(-x)}\right| \leq 1$$

4. 2-smooth and $\frac{1}{4}$ smooth. Proof : Find second derivatives and find maximum value.
5. Fix some $(\mathbf{x}, y) \in \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}'\|_2 \leq R\} \times \{-1, 1\}$. Let $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{R}^d$. For $i \in [1, 2]$, let $\ell_i = \max\{0, 1 - y\langle \mathbf{w}_i, \mathbf{x}\rangle\}$. We wish to show that $|\ell_1 - \ell_2| \leq R\|\mathbf{w}_1 - \mathbf{w}_2\|_2$. If both $y\langle \mathbf{w}_1, \mathbf{x}\rangle \geq 1$ and $y\langle \mathbf{w}_2, \mathbf{x}\rangle \geq 1$, then it follows that $|\ell_1 - \ell_2| = 0$. Now assume that $|\{i : y\langle \mathbf{w}_i, x\rangle < 1\}| \geq 1$. Assume without loss of generality that $1 - y\langle \mathbf{w}_1, \mathbf{x}\rangle \geq 1 - y\langle \mathbf{w}_2, \mathbf{x}\rangle$. Hence

   $$\begin{aligned}
   |\ell_1 - \ell_2| &= \ell_1 - \ell_2 \\
   &= 1 - y\langle \mathbf{w}_1, \mathbf{x}\rangle - \max\{0, 1 - y\langle \mathbf{w}_2, \mathbf{x}\rangle\} \\
   &\leq 1 - y\langle \mathbf{w}_1, \mathbf{x}\rangle - 1 + y\langle \mathbf{w}_2, \mathbf{x}\rangle \\
   &= y\langle \mathbf{w}_2 - \mathbf{w}_1, \mathbf{x}\rangle \\
   &\leq \|\mathbf{w}_1 - \mathbf{w}_2\|\|\mathbf{x}\|_2 \\
   &\leq R\|\mathbf{w}_1 - \mathbf{w}_2\|
   \end{aligned}$$

Optimization (12 points)

Question 2.

Answer 2.

1. Start from the LHS and complete the square

$$\begin{aligned}\langle \mathbf{w}^{(t)} - \mathbf{w}^*, \mathbf{v}_t \rangle &= \frac{1}{\eta}\langle \mathbf{w}^{(t)} - \mathbf{w}^*, \eta \mathbf{v}_t \rangle \\ &= \frac{1}{2\eta}(-\|\mathbf{w}^{(t)} - \mathbf{w}^* - \eta \mathbf{v}_t\|^2 + \|\mathbf{w}^{(t)} - \mathbf{w}^*\|^2 + \eta^2\|\mathbf{v}_t\|^2) \\ &= \frac{1}{2\eta}(-\|\mathbf{w}^{(t+1)} - \mathbf{w}^*\|^2 + \|\mathbf{w}^{(t)} - \mathbf{w}^*\|^2 + \eta^2\|\mathbf{v}_t\|^2)\end{aligned}$$

   Now put this into the summation and you'll see that we have a telescopic sum that eventually collapses to

$$\begin{aligned}\sum_{t=1}^{T}\langle \mathbf{w}^{(t)} - \mathbf{w}^*, \mathbf{v}_t \rangle &= \frac{1}{2\eta}(\|\mathbf{w}^{(1)} - \mathbf{w}^*\|^2 - \|\mathbf{w}^{(T+1)} - \mathbf{w}^*\|^2) + \frac{\eta}{2}\sum_{t=1}^{T}\|\mathbf{v}_t\|^2 \\ &\leq \frac{1}{2\eta}(\|\mathbf{w}^{(1)} - \mathbf{w}^*\|^2) + \frac{\eta}{2}\sum_{t=1}^{T}\|\mathbf{v}_t\|^2 \\ &= \frac{1}{2\eta}\|\mathbf{w}^*\| + \frac{\eta}{2}\sum_{t=1}^{T}\|\mathbf{v}_t\|^2\end{aligned}$$

2. I accepted the answer if you had some very minor indexing issues. However these are the correct answers. You should not include a dependency from s_0 or s_{-1}.
   (a) $s_t = (1-\beta_2)\sum_{i=0}^{t}\beta_2^{t-i}g_i^2$
   (b) $\mathbb{E}[s_t] = \mathbb{E}[g_t^2](1-\beta_2)\sum_{i=1}^{t}\beta_2^{t-1} = \mathbb{E}[g_t^2](1-\beta_2^t)$
3. You should expand the loss to get something like

$$\mathop{\mathbb{E}}_{\delta\sim\mathcal{N}}[\mathcal{L}^*] = \mathop{\mathbb{E}}_{\delta\sim\mathcal{N}}\left[\frac{1}{m}\sum_{i=1}^{m}(-2(y^{(i)} - {x^{(i)}}^T\theta)({\delta^{(i)}}^T\theta) + ({\delta^{(i)}}^T\theta)^2)\right]$$

   One can take the expectation individually for each sample

$$\mathop{\mathbb{E}}_{\delta\sim\mathcal{N}}\left[-2(y^{(i)} - {x^{(i)}}^T\theta)({\delta^{(i)}}^T\theta)\right] = -2(y^{(i)} - {x^{(i)}}^T\theta)\mathop{\mathbb{E}}_{\delta\sim\mathcal{N}}\left[({\delta^{(i)}}^T\theta)\right] = 0$$

   and

$$\mathop{\mathbb{E}}_{\delta\sim\mathcal{N}}\left[({\delta^{(i)}}^T\theta)^2\right] = \mathop{\mathbb{E}}_{\delta\sim\mathcal{N}}\left[({\delta^{(i)}}^T\theta)^T({\delta^{(i)}}^T\theta)\right] = \sigma^2\|\theta\|_2^2$$

   which means

$$\mathop{\mathbb{E}}_{\delta\sim\mathcal{N}}[\mathcal{L}^*] = \mathcal{L} + \sigma^2\|\theta\|_2^2$$

   due to the noises all being IID.

Neural Networks (23 points)

Question 3.

Answer 3. For each of these questions, you were deducted marks if you did not explicitly mention the number of units in each layer you constructed, the weights for connections and bias terms. Everything should be explained explicitly.

1.
- Construct the graph with $k+1$ units at the input, 2^k+1 units in the hidden layer and a single output unit.
- Let $\{\mathbf{u}_i\}_{i=1}^p$ be all the vectors in $\{\pm 1\}^k$ on which f outputs 1. Observe that for every i and every $\mathbf{x} \in \{\pm 1\}^k$, if $\mathbf{x} \neq \mathbf{u}_i$ then $\langle \mathbf{x}, \mathbf{u}_i \rangle \leq k-2$ and if they are equal, $\langle \mathbf{x}, \mathbf{u}_i \rangle = k$. It follows that the function $g_i(\mathbf{x}) = \text{sign}(\langle \mathbf{x}, \mathbf{u}_i \rangle - k + 1)$ equals 1 if and only if $\langle \mathbf{x}, \mathbf{u}_i \rangle = k$.
- Then we can adapt the weights between the input and hidden layer so that for every $i \in [1, p]$, the i-th hidden neuron implements $g_i(\mathbf{x})$. We can do this by using a weight of -1 for any component that is in that specific element be -1 and a weight of 1 for all others, with the bias $-k+1$.
- Next, we observe that $f(\mathbf{x})$ is the disjunction of the functions $g_i(\mathbf{x})$ and therefore
- The hidden to output layer connection should therefore be defined by

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{p} g_i(\mathbf{x}) + p - 1 \right)$$

meaning that we can simply use a weight of 1 for all relevant g_i to our output (with the rest 0) with a bias of $p-1$, which concludes our proof. Alternatively you can have the bias as $2^k - 1$ and have the weights from the irrelevant g_i to be -1 instead. Both of these are correct.

Commons reasons for deducted marks

- You don't explicitly say what the weights are within either of the layers.
- If you don't mention the 0 weights when you use a summation over the p variable, you lost marks as you're not specifying what happens to these connections.

2.
- Observe that if $f : \{\pm 1\}^n \to \{\pm 1\}$ is the conjunction, then it can be written as $f(\mathbf{x}) = \text{sign}(1 - k + \sum_{i=1}^k x_i)$.
- Similarly, the disjunction can be written as $f(\mathbf{x}) = \text{sign}(k - 1 + \sum_{i=1}^k x_i)$. Hence weights should always be 1 and only the bias changes depending on the conjunction or disjunction.

Commons reasons for deducted marks

- If your bias terms were off.

3.
- Let $\epsilon > 0$. Following the hint, we can separate the domain $[-1, 1]^n$ by disjoint intervals such that for every pair $\mathbf{x}_0, \mathbf{x}_1$ which lie in the same box, we have $|f(\mathbf{x}_0) - f(\mathbf{x}_1)| \leq \epsilon$. Since we only aim at approximating f to an accuracy of ϵ, we can pick an arbitrary point from each box.
- We know that f is ρ-Lipshitz, therefore you can replace into the formula

$$\rho \|\mathbf{x}_0 - \mathbf{x}_1\| \leq \epsilon$$

This means you need to ensure your step size $\delta = \|\mathbf{x}_0 - \mathbf{x}_1\|$ needs to satisfy

$$\rho\delta \leq \epsilon$$

However note that there are k different possible directions to move. To ensure that the norm of the step direction is bounded by δ, you need to ensure that movement in any of the directions is bounded as well. The easiest way to do so is to restrict

$$\delta \leq \frac{\epsilon}{\rho\sqrt{k}}$$

by the Cauchy-Schwartz inequality.

- Now that you have your step size, split each direction into different sets, namely you now have a series of bounds by $[-1 + \delta, -1 + 2\delta, \ldots, 1]$ for each of the k directions.
- The first layer has $k\lceil 2/\delta \rceil$ nodes which correspond to the intervals that make up our boxes. Adjust the weights between the input and the hidden layer such that given an input $\mathbf{x}$, the output of each neuron is close enough to 1 if the corresponding coordinate of $\mathbf{x}$ lies in the corresponding interval (here, we can approximate the indicator function using the sigmoid function). The weights here don't have to be very specific since the function f is not given to you. You can also add a bias term whose value depends on how you set the weights in the next hidden layer.
- In the next layer, we construct a neuron for each box, meaning we again have $\lceil 2/\delta \rceil^k$ units (**You must mention this number explicitly**). We can adjust the weights such that the output of each neuron is 1 if $\mathbf{x}$ belongs to the corresponding box, and 0 otherwise (this follows from part 1, where we can represent membership through a boolean approximator).
- Finally, we can easily adjust the weights between the second layer and the output layer such that the desired output is obtained. You need to say how you can find these values. The easiest is to find the coordinates of the specific hyper-cube/box the point falls within and then use the centroid. The weight from each unit in the second hidden layer to the output is simply f evaluated at these centroid.

Commons reasons for deducted marks

- Your step size is improperly calculated, namely missing the $\sqrt{k}$ component.
- You don't mention the number of units in the second hidden layer.
- You don't mention what the weights from the second hidden layer to the output.

Convolutional Neural Networks (16 points)

Question 4.

Answer 4.

1. This can be replaced by a 12×12 convolution. Assume the first layer has weights $w^1_{0,0}$ to $w^1_{2,2}$ and the third layer has weights $w^2_{0,0}$ to $w^2_{1,1}$. The pooling layer is the same as having weights $\frac{1}{4}$ everywhere, as it acts on a 2×2 input and takes the average of all these elements. Weights for each cell will simply be

$$w_{ij} = \frac{1}{4} \cdot w^1_{i\%3,j\%3} \cdot w^2_{\lfloor \frac{i}{6} \rfloor, \lfloor \frac{j}{6} \rfloor}$$

   where $\%$ is the modulus operator and $\lfloor \cdot \rfloor$ is the floor function.

   Commons reasons for deducted marks

   - You only provide an image rather than a formula.
   - Your formula depends on the input.
   - You use the ceiling function rather than the floor function.

2. See Table 1

Layer	Output Dimensions	Number of Parameters
Input	$32 \times 32 \times 3$	0
Conv(3,8)	$32 \times 32 \times 8$	$8 * (3 * 3 * 3 + 1) = 224$
BatchNorm	$32 \times 32 \times 8$	16
ReLU	$32 \times 32 \times 8$	0
MaxPooling(2)	$16 \times 16 \times 8$	0
Conv(3,16)	$16 \times 16 \times 16$	$16 * (3 * 3 * 8 + 1) = 1168$
BatchNorm	$16 \times 16 \times 16$	$2 * 16 = 32$
LeakyReLU	$16 \times 16 \times 16$	0
MaxPooling(2)	$8 \times 8 \times 16$	0
Flatten	$16 * 8 * 8$	0
FC(10)	10	$(8 * 8 * 16 + 1) * 10 = 10250$

Table 1 – Table to fill out

3. The bias in the convolutional layers, as the normalization will remove their effect. Batchnorm is incorrect because removing the scaling and the bias correction paramters will cause the two convolutional layers to compound in a non-linear fashion and lead to a different output.
4. See Table 2
5. The simplest answer is just 3 rows of `|1 0 -1|` for light-to-dark transitions and `|-1 0 1|` for the other direction (rows have to be the same). **If you multiplied each row by a constant (namely $\frac{1}{3}$) that is correct as well**. The important property is that the rows are the same and the gradient is positive in the right/left direction (**Mentioning the gradient is necessary for full marks**) and unchanging in the direction we don't want to detect (vertical changes). Symmetry is also necessary. This means that when the transition in the image is in the same direction as the gradient, these changes are enhanced by the filter and will produce positive results. For other transitions, these get cancelled out by the filter transition and therefore isn't detected by the filter.

Layer	Receptive Field
Conv(5,1) with stride 3	5
MaxPooling(2) with stride 2	8
BatchNorm	8
Conv(4,1) with stride 2	26
MaxPooling(2) with stride 1	38
Conv(3,1) with stride 1	62

Table 2 – Table to fill out