MGM-Assignment-2

February 2024

1 Theory

1. Find the conjugates of the following:
   - Negative Entropy. $f(x) = x \log x$
   - Inverse. $f(x) = 1/x$

2. (5 points) Choose $f(x) = x \log x - (x+1)\log(x+1)$, then show that

$$D_f(p \parallel q) = 2\,\text{JSD}\,(p, q) - \log(4).$$

   Recall that the Jensen-Shannon Divergence (JSD) is given by

$$\text{JSD}(p, q) = \frac{1}{2} D_{\text{KL}}\left(p \parallel \frac{p+q}{2}\right) + \frac{1}{2} D_{\text{KL}}\left(q \parallel \frac{p+q}{2}\right).$$

3. (5 points) Show that the convex conjugate of f above is

$$f^*(t) = -\log(1 - e^t).$$

4. (2 marks) If f is 1-Lipschitz, then show that

$$\|\nabla f(x)\| \leq 1, \quad \text{for all } x \in \text{dom}(f).$$

5. (2 marks) Consider $A \subset \mathbb{R}^n$. If A is closed and bounded, then it is compact. Show that for all 1-Lipschitz functions $f : A \to \mathbb{R}$, we must have a function f^* such that the following holds:

$$|f^*(x) - f^*(y)| = \|x - y\|.$$

   You are allowed to guess a function.

6. (2 marks)($\epsilon - \delta$ definition) A function is defined to continuous if for every $\epsilon > 0$, there exists a δ such that whenever $d(x, y) < \delta$, then $d(f(x), f(y)) < \epsilon$. Here d is any valid metric. Prove or disprove that any L Lipschitz continuous function is continuous.

7. (3 marks) We define two vectors x and y to be orthogonal if $x^T y = 1$. Given any unit vector v, show that v and $(I - vv^T)v$ are orthogonal. Given a unit vector $v = [1/\sqrt{2}, 1/\sqrt{2}]$ in $\mathbb{R}^2$, write down the component of vector $[1, 2]$ along v and its orthogonal vector $(I - vv^T)v$. Plot the vector v, and the projected components.

8. (4 points) Consider the following optimization problem

$$\begin{aligned} &\text{minimize} \quad x^T x \\ &\text{subject to} \quad Ax = b, \quad A \in \mathbb{R}^{p \times n} \end{aligned}$$

- Find the Lagrangian function
- Find the dual function
- Check whether the dual function is concave
- Check whether the dual function is lower bound to p^*
- Does Slater condition hold?

9. (3 points) Show that the distance function defined as $d(x, y) = \|x - y\|$, where $x, y \in \mathbb{R}^n$ is a convex function. Can you think of a subset A of $\mathbb{R}^n$, where the distance function is not necessarily convex? [Hint: if the distance function is convex then there exists (x^*, y^*) for which, $d(x, y)$ is minimum.] Can this uniqueness be violated on some special subset of $\mathbb{R}^n$? If the subset A is convex, is it necessary that $d(x, y)$ is convex function?

2 MLP in Python

Instructions:

- Copy the following python code in your google colab: `https://colab.research.google.com/drive/1fEi5TAZHjCFZrSsHUURYlakdlbrsVhlg?usp=sharing`
- Run the examples shown in this colab notebook.

Answer the following:

1. Generate a tensor data from

$$\begin{aligned}&[[1,2],[3,4]],\\&[[5,6],[2,8]],\\&[[9,0],[5,2]],\\&[[1,7],[2,11]].\end{aligned}$$

using appropriate function, for example, `torch.tensor` as shown in colab notebook, and store this tensor in `mytensor` variable.

2. Report the device. Function used?
3. Report the shape of tensor. Function used?
4. Report total number of elements. Function used?
5. Print third slice of this tensor. Function used?
6. Print top right elements of your tensor. Function used?
7. Create a tensor of all ones and all zeros, but same shape as your tensor `mytensor`. Function used?
8. Create a tensor sampled from standard normal. Function used? How you would create this tensor on GPU?
9. Subtract all entries of your tensor `mytensor` by 10, and multiply by 4.
10. Compute mean and standard deviation of your tensor.
11. The linear layer `nn.linear(m,n)` takes in matrix of dimension $\ell \times m$ and outputs matrix of dimension $\ell \times n$. We want to perform the following operation:

$$y = Ax + b, \qquad A \in \mathbb{R}^{m \times n}, \quad b, y \in \mathbb{R}^n.$$

Set $m = 5, n = 3$, create A, b random. Show the output y using the linear layer.

12. Create a normal random tensor of dimension $3 \times 4 \times 3$. Apply ReLU activation on this.

13. Apply batch normalization on this tensor you just created.

14. Create a sequential container that implement: linear layer that takes in input vector of length 10, and produces output vector of length 5, followed by a ReLU activation, followed by linear layer of input dimension (guess?) and output dimension 3, followed by batch normalization, followed by ReLU.

15. Create a random batch: 100×10 then add one to this. Then use the same loss function as in colab notebook. Run one step of ADAM optimizer and show the loss. Do 3 more iterations, then print the loss.