COMP5212: Machine Learning **Fall 2022**

Homework 3: Due Friday Dec. 1, 11:59 PM

Instructions: upload a PDF report using LaTeX containing your answers to Canvas (remember to include your name and ID number).

Problem 1. Proximal Gradient Descent

Consider solving the following problem

$$\min_{\boldsymbol{w}} \|X\boldsymbol{w} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{w}\|_1,$$

where $X \in R^{n\times d}$ is the feature matrix (each row is a feature vector), $\boldsymbol{y} \in R^n$ is the label vector, $\|\boldsymbol{w}\|_1 := \sum_i |w_i|$ and $\lambda > 0$ is a constant to balance loss and regularization. This is known as the Lasso regression problem and our goal is to derive the "proximal gradient method" for solving this.

- (10 pt) The gradient descent algorithm cannot be directly applied since the objective function is non-differentiable. Discuss why the objective function is non-differentiable.
- (30 pt) In the class we showed that gradient descent is based on the idea of function approximation. To form an approximation for non-differentiable function, we split the differentiable part and non-differentiable part. Let $g(\boldsymbol{w}) = \|X\boldsymbol{w} - \boldsymbol{y}\|_2^2$, as discussed in the gradient descent lecture we approximate $g(\boldsymbol{w})$ by

$$g(\boldsymbol{w}) \approx \hat{g}(\boldsymbol{w}) := g(\boldsymbol{w}_t) + \nabla g(\boldsymbol{w}_t)^T(\boldsymbol{w} - \boldsymbol{w}_t) + \frac{\eta}{2}\|\boldsymbol{w} - \boldsymbol{w}_t\|^2.$$

  In each iteration of proximal gradient descent, we obtain the next iterate ($\boldsymbol{w}_{t+1}$) by minimizing the following approximation function:

$$\boldsymbol{w}_{t+1} = \arg\min_{\boldsymbol{w}} \hat{g}(\boldsymbol{w}) + \lambda\|\boldsymbol{w}\|_1.$$

  Derive the close form solution of $\boldsymbol{w}_{t+1}$ given $\boldsymbol{w}_t, \nabla g(\boldsymbol{w}_t), \eta, \lambda$.

Problem 2. (60 pt) Implementing LSTM

In this homework, you are asked to implement a sequence model with LSTM to conduct sentiment analysis on SST-2 dataset with postive and negative sentiments (a binary classification problem). You need to finish the pipeline for training and evaluating the model. We provide a skeleton code for data loading and iterations of training data. You are asked to implement the rest of training in Pytorch code. Detailed submission requirements are written in the final section.

You can follow the setup instructions at `https://pytorch.org/get-started/locally/`. A useful tutorial on learning building LSTM at `https://pytorch.org/tutorials/beginner/nlp/sequence_models_tutorial.html`. Details of LSTM can be found here: `https://pytorch.org/docs/stable/generated/torch.nn.LSTM.html`.

We use SST-2 for sentiment analysis. SST-2 has only two sentiments, positive and negative. Pytorch/torchtext has provide a useful dataloader to automatically download and load the data into batches. We have written the data loader for you as follow. You can find it in the attached file.

Write a report containing your experiment results. Some requirements for your report:

- You are asked to implement a sequence model with a **bidirectional** LSTM. At most two layers of LSTM can be used. The model accuracy should be around 80%.
- Describe your model structure including vocabulary size and embedding dimension for the embedding layer; input size, hidden size, number of layers for LSTM layer; drouout layer if employed; input dim and out dim for fully connected layer.
- For each of the model, report the $(\sum_{b=1}^{B}\sum_{d=1}^{D_b}\frac{loss(label_{b,d}, f_b(data_{b,d})}{D_b})$/B for each training epoch, where B is the total number of batches, f_b is the model after updated by b-th batch and D_b is the number of data points in b-th batch. An epoch is defined as one iteration of all dataset. Essentially, during a training epoch, you record down the average training loss of that batch after you update the model, and then report the average of all such batch-averaged losses after one iteration of whole dataset. You could plot the results as a figure our simply list down. Please at least report 10 epochs.

- Report the final testing accuracy of trained model.

Also, upload your code in a zip file and show how to run your code in README.

$$w_{t+1} = \arg\min_w \underbrace{\hat{g}(w) + \lambda\|w\|_1}_{\hookrightarrow}$$

$$w_{t+1} = \arg\min_w \left(\hat{g}(w) + \lambda\|w\|_1 \right)$$

$$= \arg\min_w \left(g(w_t) + \nabla g(w_t)^T (w - w_t) + \frac{\eta}{2}\|w - w_t\|_2^2 + \lambda\|w\|_1 \right)$$

$$g(w_t) + \nabla g(w_t)^T (w - w_t) + \frac{\eta}{2}\|w - w_t\|^2$$

$$w_{t+1} = \arg\min_w \left(\hat{g}(w) + \lambda\|w\|_1 \right)$$

$f(x) = g(x) + h(x)$

$g(x) = g(x^k) + \nabla g(x^k)(x - x^k) + \frac{1}{2\alpha}\|x - x^k\|_2^2$

$h(x) = \lambda\|x\|_1$

$$\frac{1}{2\alpha} = \frac{\eta}{2}$$

$$g(x) = g(x^k) + \nabla g(x^k)(x - x^k) + \frac{\eta}{2}\|x - x^k\|_2^2$$

$$= \frac{\eta}{2}\left(\frac{2}{\eta} g(x^k) + \frac{2}{\eta}\nabla g(x^k)(x - x^k) + \|x - x^k\|_2^2 \right) = \frac{\eta}{2}\left(\frac{2}{\eta} g(x^k) + \frac{2}{\eta}\nabla \right.$$

$$= \frac{\eta}{2}\left[\frac{2}{\eta} g(x^k) + \left(2\frac{1}{\eta}\nabla g(x^k)(x - x^k) + \|x - x^k\|_2^2 + \left\|\frac{1}{\eta}\nabla g(x^k)\right\|_2^2 \right) - \left\|\frac{1}{\eta}\nabla g(x^k)\right\|_2^2 \right]$$

$$= \frac{\eta}{2}\left[\frac{2}{\eta} g(x^k) + \left\|(x - x^k) + \frac{1}{\eta}\nabla g(x^k)\right\|_2^2 - \left\|\frac{1}{\eta}\nabla g(x^k)\right\|_2^2 \right]$$

$$= g(x^k) + \frac{\eta}{2}\left\|(x - x^k) + \frac{1}{\eta}\nabla g(x^k)\right\|_2^2 - \frac{\eta}{2}\left\|\frac{1}{\eta}\nabla g(x^k)\right\|_2^2$$

$$= g(x^k) + \frac{\eta}{2}\left\| x - (x^k - \frac{1}{\eta}\nabla g(x^k))\right\|_2^2 - \frac{\eta}{2}\left\|\frac{1}{\eta}\nabla g(x^k)\right\|_2^2$$

$$= \underbrace{g(x^k)}_{\text{not related to } x} + \frac{\eta}{2}\| x - \hat{x}\|_2^2 - \underbrace{\frac{\eta}{2}\left\|\frac{1}{\eta}\nabla g(x^k)\right\|_2^2}_{\text{not related to } x}$$

$$w - w_t + \frac{1}{\eta}\nabla g(w_t)$$

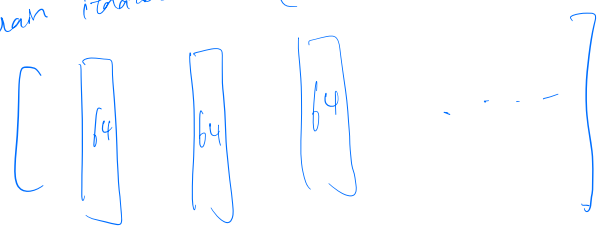
$$w - \left(w_t - \frac{1}{\eta}\nabla g(w_t)\right)$$

minimize $g(x) + h(x)$ is equivalent to

minimize $\frac{1}{2\alpha}\|x - \hat{x}\|_2^2 + \lambda\|x\|_1$

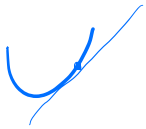
LSTM

train iterator (batches)



convex i.f.f.

$f(y) \geq f(x) + \nabla f(x)^T (y - x) \quad \forall x, y \in \mathbb{R}^n$



$$\underset{x}{\text{minimize}} \quad \frac{1}{2\alpha} \|x - \hat{x}\|_2^2 + \lambda \|x\|_1$$

$$x_{k+1} = \underset{x}{\operatorname{argmin}} \quad \frac{1}{2\alpha} \|x - \hat{x}\|_2^2 + \lambda \|x\|_1$$

---

proximal mapping

$$\text{prox}_{\alpha f}(x^k) = \underset{x}{\operatorname{argmin}} \left(f(x) + \frac{1}{2\alpha} \|x - x^k\|_2^2 \right)$$

---

soft-thresholding operator

$$S_\lambda(\hat{x}) = \begin{cases} \hat{x} - \lambda \\ \circ \\ \hat{x} + \lambda \end{cases}$$

$h(x) = \lambda \|x\|_1$

$$\text{prox}_{\alpha f}(x^k) = \underset{x}{\operatorname{argmin}} \; \lambda \|x\|_1 + \frac{1}{2\alpha} \|x - x^k\|_2^2$$

$$\Rightarrow \text{prox}_{\alpha f}(x^k) = \underset{x}{\operatorname{argmin}} \; \lambda \sum_{i=1}^{d} |x_i| + \frac{1}{2\alpha} \sum_{i=1}^{n} (x_i - x_i^k)^2$$

$\Rightarrow$ element-wise: $\text{prox}_{\alpha f}(x_i^k) =$

$$\text{prox}_{\alpha f}(x_i^k) = \arg\min_{x_i} \lambda|x_i| + \frac{1}{2\alpha}(x_i - x_i^k)^2$$

$$0 \in \partial\left(\lambda|x_i| + \frac{1}{2\alpha}(x_i - \hat{x}_i)^2\right)$$

你微分的這把來來要 可以 = 0

optimality condition:

$$0 \in \left\{\frac{1}{\alpha}(x_i - \hat{x}_i) + \lambda \partial|x_i|\right\}$$

$$x_i^* = \text{prox}_{\alpha f}(x_i^k)$$

i.f.f $\quad 0 \in \left\{\frac{1}{\alpha}(x_i - \hat{x}_i) + \lambda \partial|x_i|\right\}$ (set)

(gradient) (gradient)

x_i condition

$$0 \in \begin{cases} x_i > 0 & \frac{1}{\alpha}(x_i - \hat{x}_i) + \lambda \\ x_i \geq 0 & \frac{1}{\alpha}(x_i + \hat{x}_i) + \lambda \\ & \frac{1}{\alpha}(x_i + \hat{x}_i) + 0.9999\ldots\lambda \\ & \vdots \\ & \frac{1}{\alpha}(x_i + \hat{x}_i) + 0.00\ldots1\lambda \\ & \frac{1}{\alpha}(x_i + \hat{x}_i) + 0 \\ x_i < 0 & \frac{1}{\alpha}(x_i - \hat{x}_i) - \lambda \end{cases}$$

$$0 = \eta(w_i - w_i') + \lambda V$$

$$0 = w_i - w_i' + \frac{1}{\eta}\lambda V$$

$$w_i = w_i' - \frac{1}{\eta}\lambda \cdot v \qquad v = \begin{cases} 1 & \text{if } w_i > 0 \\ [-1, 1] & \text{if } w_i = 0 \\ -1 & \text{if } w_i < 0 \end{cases}$$

$$w_i \begin{cases} w_i' - \frac{1}{\eta}\lambda & , \quad w_i' > \frac{1}{\eta}\lambda \\ 0 & , \quad |w_i'| \le \frac{1}{\eta}\lambda \\ w_i' + \frac{1}{\eta}\lambda & , \quad w_i' < -\frac{1}{\eta}\lambda \end{cases}$$

$$w_i' + \frac{1}{\eta}\lambda < 0$$