COMP5212 Machine Learning
Homework 3

LO, Li-yu
20997405
e-mail: lloac@connect.hkust.hk

December 1, 2023

Problem 1 - Proximal Gradient Descent

(a)

With the objective function being:

$$\min_{\mathbf{w}} \|X\mathbf{w} - y\|_2^2 + \lambda\|\mathbf{w}\|_1,$$
$$X \in R^{n\times d}, \mathbf{w} \in R^d, y \in R^n.$$

With the L1-norm, the whole thing could be further written as:

$$\min_{\mathbf{w}} \|X\mathbf{w} - y\|_2^2 + \lambda\|\mathbf{w}\|_1,$$
$$= \min_{\mathbf{w}} \|X\mathbf{w} - y\|_2^2 + \lambda\sum_{i=1}^{d} |\mathbf{w}_i|.$$

From the term $\sum_{i=1}^{d} \|\mathbf{w}_i\|$, it could be seen that it is not a continuous function, and it is particularly not differentiable when $\mathbf{w}_i = 0$. Therefore, the objective function is non-differentiable, leading the inapplicability of gradient descent algorithm.

(b)

Here we derive the closed-form solution $\mathbf{w}_{t+1}$ based on the given $\mathbf{w}_t$, $\nabla g(\mathbf{w}_t)$, η, and λ. Also not that

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

And with:

$$\hat{g}(\mathbf{w}_t) := g(\mathbf{w}_t) + \nabla g(\mathbf{w}_t)^T(\mathbf{w} - \mathbf{w}_t) + \frac{\eta}{2}\|\mathbf{w} - \mathbf{w}_t\|_2^2 \tag{2}$$

First, we rewrite $\hat{g}(\mathbf{w}_t)$:

$$\begin{aligned}
&\hat{g}(\mathbf{w}_t) = g(\mathbf{w}_t) + \nabla g(\mathbf{w}_t)^T(\mathbf{w} - \mathbf{w}_t) + \frac{\eta}{2}\|\mathbf{w} - \mathbf{w}_t\|_2^2 \\
&= \frac{\eta}{2}[\frac{2}{\eta}g(\mathbf{w}_t) + \frac{2}{\eta}\nabla g(\mathbf{w}_t)^T(\mathbf{w} - \mathbf{w}_t) + \|\mathbf{w} - \mathbf{w}_t\|_2^2] \\
&= \frac{\eta}{2}[\frac{2}{\eta}g(\mathbf{w}_t) + \frac{2}{\eta}\nabla g(\mathbf{w}_t)^T(\mathbf{w} - \mathbf{w}_t) + \|\mathbf{w} - \mathbf{w}_t\|_2^2 + \|\frac{1}{\eta}\nabla g(\mathbf{w}_t)\|_2^2 - \|\frac{1}{\eta}\nabla g(\mathbf{w}_t)\|_2^2] \\
&= \frac{\eta}{2}[\frac{2}{\eta}g(\mathbf{w}_t) - \|\frac{1}{\eta}\nabla g(\mathbf{w}_t)\|_2^2 + \|(\mathbf{w} - \mathbf{w}_t) + \frac{1}{\eta}\nabla g(\mathbf{w}_t)\|_2^2] \\
&= g(\mathbf{w}_t) - \frac{\eta}{2}\|\frac{1}{\eta}\nabla g(\mathbf{w}_t)\|_2^2 + \frac{\eta}{2}\|(\mathbf{w} - \mathbf{w}_t) + \frac{1}{\eta}\nabla g(\mathbf{w}_t)\|_2^2
\end{aligned}$$

As $\mathbf{w}_t$ is not a variable, eqn. 1 is equivalent to:

$$\hat{g}(\mathbf{w}_t) = arg\min_{\mathbf{w}}(\frac{\eta}{2}\|(\mathbf{w}-\mathbf{w}_t)+\frac{1}{\eta}\nabla g(\mathbf{w}_t)\|_2^2+\lambda\|\mathbf{w}\|_1). \tag{3}$$

We also let $\mathbf{w}' = \mathbf{w}_t - \frac{1}{\eta}\nabla g(\mathbf{w}_t)$, then we have:

$$\begin{aligned}\mathbf{w}_{t+1} &= arg\min_{\mathbf{w}}(\frac{\eta}{2}\|\mathbf{w}-\mathbf{w}'\|_2^2+\lambda\|\mathbf{w}\|_1)\\ &= arg\min_{\mathbf{w}}(\frac{\eta}{2}\sum_{i=1}^{d}(\mathbf{w}_i-\mathbf{w}'_i)^2+\lambda\sum_{i=1}^{d}\|\mathbf{w}_i\|).\end{aligned} \tag{4}$$

Secondly, as we know the optimality condition for a non-differentiable objective function could be derived with subgradients. We say w^* is a minimizer i.f.f.:

$$0 \in \{g|g \in R^d, subgradient\ of\ \mathbf{w}^*\} \tag{5}$$

From eqn. 4, we could treat the problem element-wise, and with cond. 5, in our case, the following condition should be true:

$$0 \in \{\eta(\mathbf{w}_i-\mathbf{w}'_i)+\lambda\partial|\mathbf{w}_i|\} \tag{6}$$

Finally, we derive the minimizer. As

$$\partial|\mathbf{w}_i| = \begin{cases}1, & if\ \mathbf{w}>0.\\ [-1,1], & if\ \mathbf{w}=0.\\ -1, & if\ \mathbf{w}<0.\end{cases} \tag{7}$$

And we let:

$$0 = \eta(\mathbf{w}_i-\mathbf{w}'_i)+\lambda\partial|\mathbf{w}_i| \tag{8}$$

$$\rightarrow \mathbf{w}_i = \mathbf{w}'_i - \frac{1}{\eta}\lambda\partial|\mathbf{w}_i| \tag{9}$$

Therefore, we have:

$$\mathbf{w}_i = \begin{cases}\mathbf{w}'_i - \frac{1}{\eta}\lambda, & if\ \mathbf{w}'_i > \frac{1}{\eta}\lambda.\\ 0, & if\ |\mathbf{w}'_i| \le \frac{1}{\eta}\lambda.\\ \mathbf{w}'_i + \frac{1}{\eta}\lambda, & if\ \mathbf{w}'_i < -\frac{1}{\eta}\lambda.\end{cases} \tag{10}$$

The above is equivalent to:

$$\mathbf{w}_i = \begin{cases}[\mathbf{w}_t - \frac{1}{\eta}\nabla g(\mathbf{w}_t)]_i - \frac{1}{\eta}\lambda, & if\ [\mathbf{w}_t - \frac{1}{\eta}\nabla g(\mathbf{w}_t)]_i > \frac{1}{\eta}\lambda.\\ 0, & if\ |[\mathbf{w}_t - \frac{1}{\eta}\nabla g(\mathbf{w}_t)]_i| \le \frac{1}{\eta}\lambda.\\ [\mathbf{w}_t - \frac{1}{\eta}\nabla g(\mathbf{w}_t)]_i + \frac{1}{\eta}\lambda, & if\ [\mathbf{w}_t - \frac{1}{\eta}\nabla g(\mathbf{w}_t)]_i < -\frac{1}{\eta}\lambda.\end{cases} \tag{11}$$

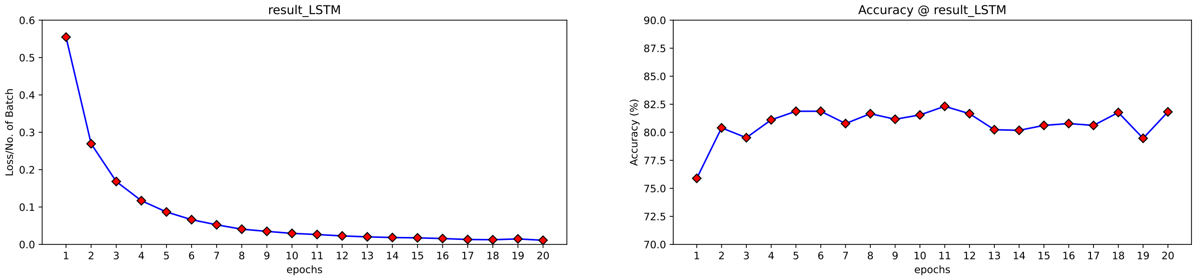
Problem 2 - Implementing LSTM

In problem two, we implemented a Bi-LSTM for SST-2 dataset. In particular, We adopt PyTorch library as our main programming framework. Below then reports the training results. Note that all the training was carried on GeForce RTX 3090.

Bi-LSTM

The adopted embedding layer is as follows: the vocabulary size is 18003, and embedding dimension is 100. As for the LSTM structure, the input size is the same as embedding dimension, which is 100; the hidden states size is 150, and we only deployed 1 layer of LSTM. No dropout layer is utilized, yet a fully connected layer with input size 300, and output size 2 is added to output the final classification results.

Training Results



(a) LSTM Loss v. Epochs

(b) LSTM Accuracy

Figure 1: Long Short-Term Memory Network training results.

The above shows the final training results. It can be observed that, during traning, the loss gradually decreased with the increase of epoch number, while the accuracy had a trend of improving. Nevertheless, it could also be seen that the accuracy experienced a fluctuation after reaching around 80%. It could be improved by adding dropout layer, or a better embedding method; or even improve the entire performance with different framework.