先端機械学習　中間レポート

提出日：2023 年 7 月 27 日

系／学科／類：情報工学系

学籍番号：20B30100

氏名：伊藤　悠馬

作成したコードは `https://github.com/gray-1to/ad_ML_2023` 上に公開している。

1 Problem 1

1.1 Derive the Hessian of the objective function of this optimization.

$$\frac{\partial J}{\partial \boldsymbol{w}} = \sum_{i=1}^{n} \left(\frac{-y_i e^{-y_i \boldsymbol{w}^T \boldsymbol{x}_i}}{1 + e^{-y_i \boldsymbol{w}^T \boldsymbol{x}_i}} \boldsymbol{x}_i \right) + 2\lambda \boldsymbol{w}$$

よって

$$\nabla^2 J(\boldsymbol{w}) = \sum_{i=1}^{n} \left(\frac{y_i^2 e^{-y_i \boldsymbol{w}^T \boldsymbol{x}_i}}{(1 + e^{-y_i \boldsymbol{w}^T \boldsymbol{x}_i})^2} \boldsymbol{x}_i \boldsymbol{x}_i^T \right) + 2\lambda \boldsymbol{I}$$

ここで $\boldsymbol{I}$ は単位行列とする。

1.2 Compare the performance of the above two optimization methods

以下の図 1 からわかる通り Batch steepest gradient method の方が少ない回数でロスを下げている。

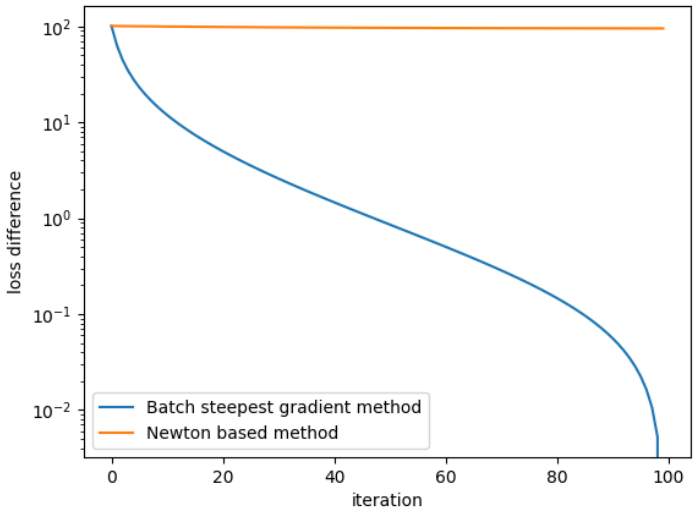

図 1: 2 つの最適化手法の比較

図 1 では分かりづらいが、下の図に示す通り各手法で収束している。

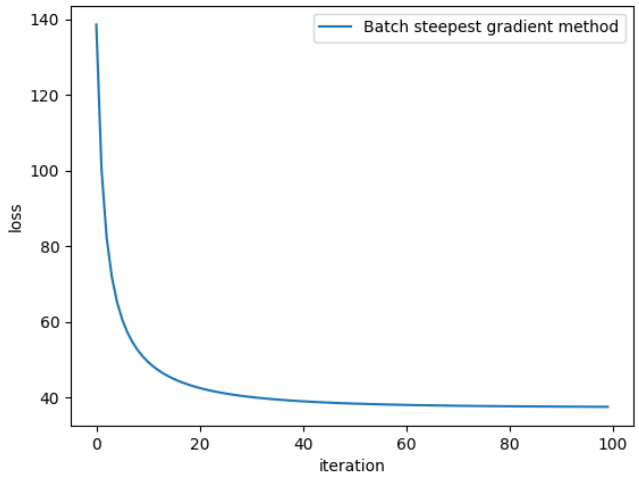

図 2: Batch steepest gradient method の損失の遷移

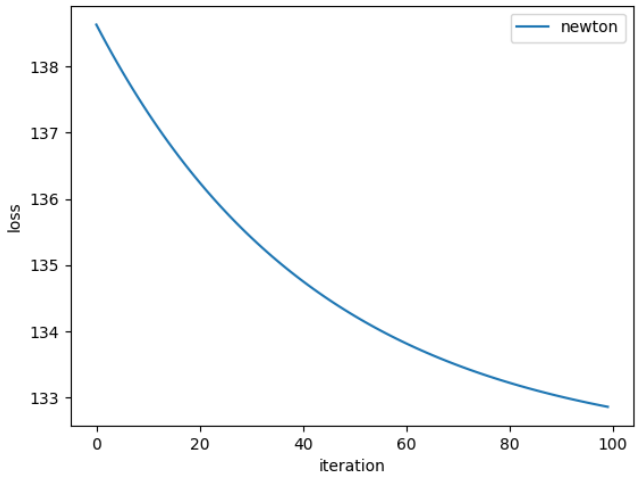

図 3: newton based method の損失の遷移

2 Problem 2

2.1 Show the result of PG

$\lambda = 0.1$ で PG を行った際の損失の動きは以下の通りである。
損失は単調減少となり滑らかに収束している。

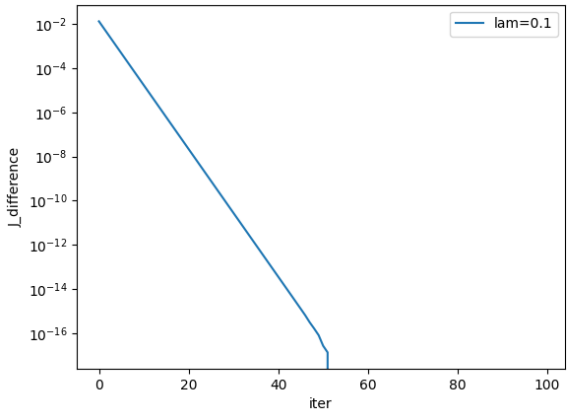

図 4: PG の損失の遷移

λを変えた際の最終 $\boldsymbol{w}$ を比較する。以下のグラフは各 λ における最終 $\boldsymbol{w}$ での損失をプロットしたものである。

λ が大きくなるということは損失中の係数が大きくなるということなので、λ が小さい時には λ の増加に従って損失が増加していく。

λ が一定程度まで大きくなると目的関数の 2 次形式部分よりも L1 正則化の部分の方が異常に強く影響するようになり、$\boldsymbol{w}$ は (0,0) に収束してしまうようになるということがわかる。

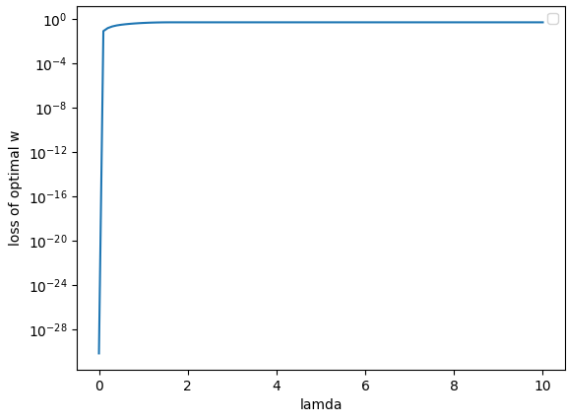

図 5: 各 λ での最終 $\boldsymbol{w}$ での損失

2.2 Run standard proximal gradient method and the advanced one (such as AdaGrad) in case

PG と AdamGrad での比較を行うために各手法の損失の動きを以下にグラフ化した。

最適解との差分をプロットするために結果の良かった AdamGrad の最終 $\boldsymbol{w}$ での損失と各手法各イテレーションの損失の差分をプロットしている。

安定解に至るまでの回数は PG の方が早いものの最終的な $\boldsymbol{w}$ における損失は AdaGrad を用いた方が小さいという結果となった。

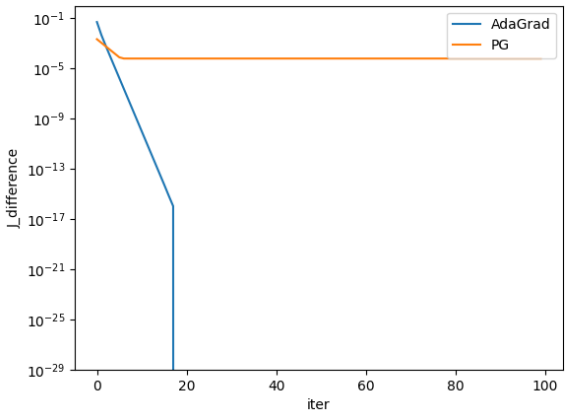

図 6: PG, AdamGrad の比較

3 Problem 10

3.1 Prove the following inequality used in proximal gradient method

仮定より f は $\beta - smooth$ であるから、問題に示されている通り、

$$\| \boldsymbol{\nabla} f(\boldsymbol{x_1}) - \boldsymbol{\nabla} f(\boldsymbol{x_1}) \|_2 \leq \beta \| \boldsymbol{x_1} - \boldsymbol{x_1} \|_2$$

さらに一般に、f が連続であるなら

$$f(\boldsymbol{x} + \boldsymbol{a}) = f(\boldsymbol{a}) + \int_0^1 \boldsymbol{\nabla} f(\boldsymbol{a} + t\boldsymbol{h})\boldsymbol{h} dt$$

以上の式より、

$$\begin{aligned}
& f(\boldsymbol{x_2}) - f(\boldsymbol{x_1}) - \boldsymbol{\nabla} f(\boldsymbol{x_1})(\boldsymbol{x_2} - \boldsymbol{x_1}) \\
= \quad & f(\boldsymbol{x_1}) + \int_0^1 \boldsymbol{\nabla} f(\boldsymbol{x_1} + t(\boldsymbol{x_2} - \boldsymbol{x_1}))(\boldsymbol{x_2} - \boldsymbol{x_1}) dt - f(\boldsymbol{x_1}) - \boldsymbol{\nabla} f(\boldsymbol{x}_1)(\boldsymbol{x_2} - \boldsymbol{x_1}) \\
= \quad & \int_0^1 \{\boldsymbol{\nabla} f(\boldsymbol{x_1} + t(\boldsymbol{x_2} - \boldsymbol{x_1})) - \boldsymbol{\nabla} f(\boldsymbol{x}_1)\}(\boldsymbol{x_2} - \boldsymbol{x_1}) dt \\
\leq \quad & \int_0^1 \| \boldsymbol{\nabla} f(\boldsymbol{x_1} + t(\boldsymbol{x_2} - \boldsymbol{x_1})) - \boldsymbol{\nabla} f(\boldsymbol{x}_1) \|_2 \cdot \| \boldsymbol{x_2} - \boldsymbol{x_1} \|_2 \, dt \\
\leq \quad & \int_0^1 \beta \| t(\boldsymbol{x_2} - \boldsymbol{x_1}) \|_2 \cdot \| \boldsymbol{x_2} - \boldsymbol{x_1} \|_2 \, dt \\
= \quad & \frac{\beta}{2} \| (\boldsymbol{x_2} - \boldsymbol{x_1}) \|_2^2
\end{aligned}$$

よって示された。

4 Topics of machine learning you want to take

自分が今後授業で扱って欲しい内容は Chat GPT についてである。授業で扱われる予定である transformer のデコーダ部分と構造的にはほぼ同じだが、transformer からの進化点について知りたい。話題になる程滑らかな応答能力があり、様々なタスクにも対応している汎用的なモデルを支えているのはどのような技術なのかに興味がある。

特に Chat GPT のモデルでは一般的な教師あり学習に加え、強化学習を行うことでより精度の高い応答を可能にしている点が面白いと思う。

強化学習について授業で扱われているのをあまり見ないので強化学習についての勉強も行えたら嬉しい。