F-017

分割演算子法を用いた確率的勾配法の検討
Stochastic gradient descent using split operator method

鈴木 和磨 † 大久保 潤 †
Kazuma Suzuki Jun Ohkubo

1. はじめに

機械学習においては，目的関数 F を最小化する式 (1)，式 (2) のような最適化の問題を考えることが多い．

$$\min_{\boldsymbol{w}} \ F(\boldsymbol{w}) \tag{1}$$

$$F(\boldsymbol{w}) := \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{w}) \tag{2}$$

$f_i(\boldsymbol{w})$ は i 番目の学習データに対してパラメータ $\boldsymbol{w}$ を用いたときの目的関数である．機械学習では与えられた n 個の学習データの集合 $S = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$ をもとに学習をおこなう．ここでの $\boldsymbol{x}$ はデータの入力を表し，y がそれに対する出力を表す．具体的には $f_i(\boldsymbol{w}) = (\boldsymbol{w}^T\boldsymbol{x}_i - y_i)^2$ とすると二乗誤差を最小にする問題となり，$f_i(\boldsymbol{w}) = \log(1 + \exp(-\boldsymbol{w}^T\boldsymbol{x}_i y_i)) + \frac{1}{2}\lambda\boldsymbol{w}^T\boldsymbol{w}, y \in \{-1, +1\}$ とすると正則化を考慮したロジスティック回帰問題を表す形になる．

式 (1) の最適化を実現するにあたって，機械学習では勾配法がよく用いられる．勾配法は関数の一階微分の情報からその関数の最小値を探索するような手法である．最も単純な勾配法である最急降下法 (バッチ型勾配法) は式 (3) の更新式で表される．

$$\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} - \eta_t \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\boldsymbol{w}^{(t)}) \tag{3}$$

ただし，η_t は学習率と呼ばれる更新の幅を決めるハイパーパラメータである．パラメータの更新を効率良くおこなうためには，学習率はパラメータが発散しない程度に大きな値を取ることが望ましいとされる [1]．

最急降下法は，目的関数が最も減少するような勾配方向を選び更新している点において，最も単純な勾配法であると言える．ただし，データ数である n 個分の目的関数の微分値の和を計算する必要があるため，データ数が増えると計算時間が増大してしまう．その点を解消する代表的な手法として確率的勾配法 (Stochastic Gradient Descent) があげられる．SGD の更新式は式 (4) で与えられる．

$$\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} - \eta_t \nabla f_{i_t}(\boldsymbol{w}^{(t)}) \tag{4}$$

ただし，$i_t \in \{1, \ldots, n\}$ は t 回目の更新で用いられるデータ番号であり，ランダムに選ばれるものとする．最急降下法と比較しての SGD の特徴としては，最急降下法が局所的最適解に陥ることに対して比較的局所的最適解にはまりにくい点や，一反復あたりの計算時間が少ない点，勾配方向の分散が大きく反復回数が多くなる点，更新がデータ数に依存しない点などがあげられる [1]．更新がデータ数に依存しないため，SGD はディープラーニングなどの大規模なデータを扱う際にその効果を発揮する．なお，SGD においては学習率 η_t を反復ごとに減少させなければならず，$\eta_t \propto t^{-1}$ や $\eta_t \propto t^{-1/2}$ といった学習率が用いられることが多い [2]．

本研究では，この SGD に対して時間発展を扱う手法である分割演算子法 [3] を応用する理論を提案し，実際にロジスティック回帰問題を用いてその性能を評価する．

2. 連続時間の時間発展としての SGD

まず，SGD に分割演算子法を応用するために，勾配法の更新式の変形を行う．$\Delta t > 0$ を導入し，式 (3) を変形すると以下の式が得られる．

$$\frac{\boldsymbol{w}^{(t+1)} - \boldsymbol{w}^{(t)}}{\Delta t} = -\frac{\eta_t}{\Delta t}\frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\boldsymbol{w}^{(t)}) \tag{5}$$

この式において左辺を重みベクトル $\boldsymbol{w}$ を時間 t で微分している形としてみなすと，次の微分方程式を得ることができる．

$$\frac{d\boldsymbol{w}}{dt} = -\tilde{\eta}\frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\boldsymbol{w}) \tag{6}$$

ただし，$\tilde{\eta} := \frac{\eta_t}{\Delta t}$ とする．

ここで，今後の議論のために演算子 $\hat{f}$ を $\hat{f}_i(\cdot)\boldsymbol{w}^{(t)} := \nabla f_i(\boldsymbol{w}^{(t)})$ と定義する．式 (6) のような微分方程式に対してはその形式的な解を求められることが知られており，その解は

$$\frac{d\boldsymbol{w}^{(T)}}{dt} = \exp\left(\int_0^T \tilde{\eta}\frac{1}{n}\sum_{i=1}^{n} \hat{f}_i(\cdot)\right)\boldsymbol{w}^{(0)} \tag{7}$$

となる．これにより勾配法を連続時間の時間発展としてみなすことができる．なお，$\boldsymbol{w}^{(0)}$ はパラメータベクトルの初期値とする．また，この式にオイラー法を施し，元の勾配法を再現できることも確認できている．

3. 分割演算子法を用いた SGD

分割演算子法 [3] は式 (7) でも登場したような，自然対数の肩に行列や演算子がかかった形を扱う手法の一つである．行列 A, B に関して $\exp(A + B)$ のように行列が和の形を持っているものを考える．これを計算すると

$$\exp(A\Delta t + B\Delta t) \approx \exp(A\Delta t)\cdot\exp(B\Delta t) + O(\Delta t^2) \tag{8}$$

† 埼玉大学，Saitama University

これに対して分割演算子法を適用すると

$$\exp(A\Delta t + B\Delta t)$$
$$\approx \exp(\frac{1}{2}A\Delta t)\cdot\exp(B\Delta t)\cdot\exp(\frac{1}{2}A\Delta t) + O(\Delta t^3) \quad (9)$$

といったように近似誤差を抑えることができる．

ここで，連続時間として捉えた勾配法においてデータ数が 2 である時の 1 ステップを考える．この時，最急降下法は $\boldsymbol{w}^{(t+1)} = \exp(\tilde{\eta}\frac{1}{2}\sum_{i=1}^{2}\hat{f}_i(\cdot)\Delta t)\boldsymbol{w}^{(0)}$ と表され，式 (8) を考えるとこれに対する SGD の更新式は $\boldsymbol{w}^{(t+1)} = \exp(\tilde{\eta}\frac{1}{2}\hat{f}_1(\cdot))\cdot\exp(\tilde{\eta}\frac{1}{2}\hat{f}_2(\cdot))\boldsymbol{w}^{(0)}$ と表される．すなわち SGD を最急降下法の近似として捉えると，式 (9) の分割演算子法の考え方を用いることでその近似誤差を小さくすることができると考えられる．実際に

$$\exp(\tilde{\eta}\frac{1}{2}\sum_{i=1}^{2}\hat{f}_i\Delta t) \quad (10)$$
$$\approx \exp(\tilde{\eta}\frac{1}{2}\hat{f}_1\Delta t)\exp(\tilde{\eta}\frac{1}{2}\hat{f}_2(\cdot)\Delta t) + O(\Delta t^2) \quad (11)$$
$$\approx \exp(\tilde{\eta}\frac{1}{4}\hat{f}_1\Delta t)\exp(\tilde{\eta}\frac{1}{2}\hat{f}_2(\cdot)\Delta t)\exp(\tilde{\eta}\frac{1}{4}\hat{f}_1\Delta t) + O(\Delta t^3) \quad (12)$$

といった更新式を得られ，最急降下法に対して Δt の高次の項を考慮した近似をおこなえる．

SGD の特徴として，最急降下法と比較すると勾配方向の分散が大きく，収束までの反復数が多くなってしまうという点があげられた．そこに分割演算子法を用いると，式 (12) のように SGD と最急降下法との誤差が少なくなっていることがわかる．このことから SGD を最急降下法の近似として考えたとき，その誤差を小さくすることで，収束までに要する反復数が少なくなることが期待される．

しかしながら実際に SGD に分割演算子法を応用するには，連続時間の時間発展としての勾配法の近似に 4 次のルンゲクッタ法を用いる必要が生じた．これはオイラー法による近似では近似誤差が大きく，分割演算子法を応用したことによる誤差の減少が確認できなかったためである．また，4 次のルンゲクッタ法を用いることにより通常の勾配法よりも 1 反復あたりの計算時間が 4 倍になってしまい，その影響で SGD の性能の改善が難しくなった．

4. 実験

実際の例としてロジスティック回帰問題を用いて分割演算子法を用いた SGD の性能を評価する．実験では学習データ数を 1000，学習率の初期値を $\eta_0 = 10.0$，t 回目のイタレーション時の学習率を $\eta_0(t+1)^{-1/2}$ とした．また，ミニバッチのサイズを 2 とした．すなわち学習データの中からランダムにデータ 2 つ取り出して式 (10),(11),(12) でそれぞれ更新をおこなった．この条件の下で実験を 1000 回試行した平均を図 1 に示す．図 1 の縦軸は最適解との差とみなせる全ての学習データについての勾配の大きさの平均，すなわち，t 回目のイタレーション時の $\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\boldsymbol{w}^{(t)})$ を表す．図より，通常の SGD よりも分割演算子法を用いた SGD の方が勾配の大きさが早く減少していること，すなわち，より早く最適解に近づいていることが確認できた．ただし，式 (11),(12) を見ても分かるように，一回のイタレーションにおいて通常の SGD は勾配の計算を 2 回おこなっており，分割演算子法を用いた SGD は 3 回おこなっている．このことから実際の計算時間としては分割演算子法を用いた SGD の方が 1.5 倍の計算時間を要することが見込まれる点には注意が必要である．

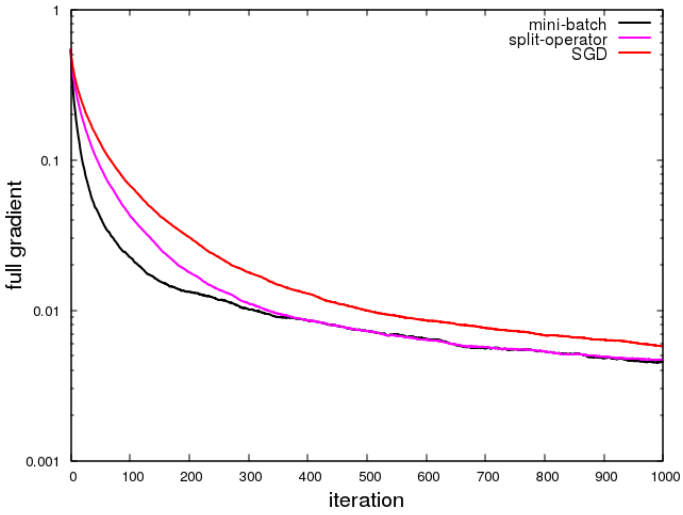

図 1: イタレーション毎の学習データについての勾配の大きさ．mini-batch は式 (10)，split-operator は式 (12)，SGD は式 (11) で更新した結果を表す．

5. 終わりに

本研究では勾配法を微分方程式とみなすことから，連続時間の時間発展として捉え，演算子形式で書き換える方法を紹介した．また，その上で分割演算子法を応用し確率的勾配法の最急降下法との誤差を減少させる方法を提案し，4 次のルンゲクッタ法を用いた場合には性能の改善が見込まれた．しかしながら，4 次のルンゲクッタ法を用いることで計算時間が増加するため実質的な性能の改善には至っていない．ただし，勾配法に時間発展を扱う手法である分割演算子法を応用できた点において，今後の性能改善への理論的基礎づけは得ることができた．

謝辞

本研究の一部は JSPS 科研費 JP16K00323 の助成を受けたものです．

参考文献

[1] 鈴木 大慈,“確率的最適化”, 講談社 (2015).

[2] L. Bottou, “Stochastic gradient descent tricks.” In Neural Networks: Tricks of the Trade, 421–436. Springer (2012).

[3] M. D. Feit, J. A. Fleck, Jr., and A. Steiger, “Solution of the Schrdinger equation by a spectral method.” J. Comput. Phys. 47, 412–433 (1982).