PFN インターン 2019 コーディング課題
課題 3, 4

東京大学大学院 情報理工学系研究科 修士 1 年
佐伯高明

2019 年 5 月

課題 3

datasets/train 内のデータのうち、7 割を学習用データ、3 割を検証用データとした。
各種パラメータについては、資料を参考に以下の通り設定した。

表 1　SGD,MomentumSGD のパラメータ

集約ステップ数 T	2
特徴ベクトル次元数 D	8
学習率 α	0.0001
モーメント η	0.9
数値微分の摂動 ε	0.001
W の初期値	平均 0, 標準偏差 0.4 の正規乱数
A の初期値	平均 0, 標準偏差 0.4 の正規乱数
b の初期値	平均 0, 標準偏差 0.4 の正規乱数
学習のエポック数	100

バッチサイズ B:10, 100 における平均損失・平均精度の変化はそれぞれ Fig1, Fig2 のようになった。

Fig1　B:10 の場合の平均損失・平均精度

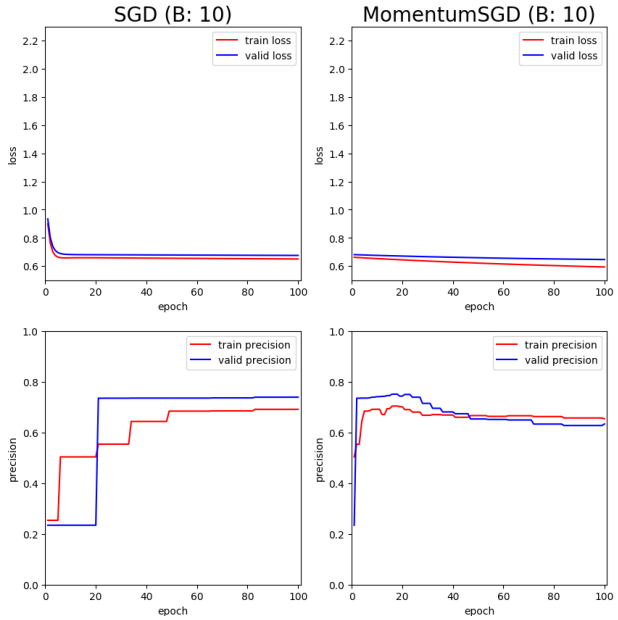


Fig2　B:100 の場合の平均損失・平均精度

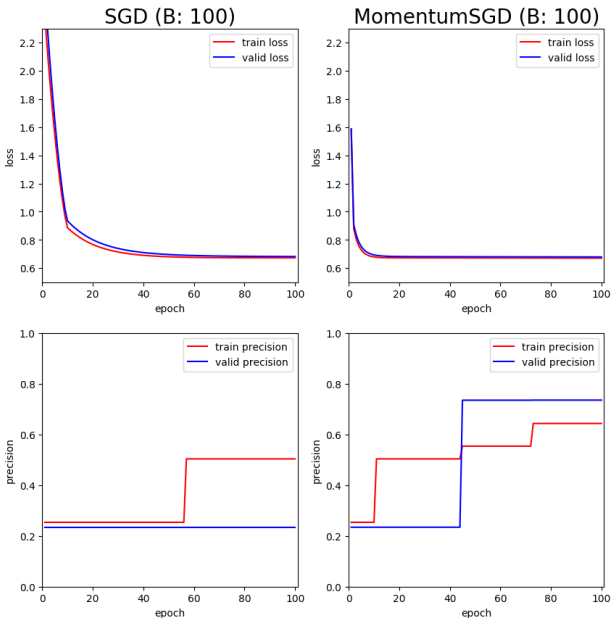


Fig1, Fig2 ともに、左側が SGD の平均損失・平均精度の変化であり、右側が MomentumSGD の平均損失・平均精度の変化である。ただし、W,A,b の初期値は全てのケースでそれぞれ等しく置いた。

平均損失の変化より、バッチサイズ B:10, 100 のいずれの場合においても、MomentumSGD は SGD よりも収束が速いことが見て取れる。また、MomentumSGD の B:10 の場合では、epoch 数 30 付近で学習用データの平均精度が最大 0.70 を超え、検証データの平均精度が最大で 0.75 を超えたが、それより大きい epoch 数では特に検証データの平均精度が減少傾向にあり、過学習を起こしていることが考えられる。
また、SGD の B:100 の場合では、平均損失が下がり続けているにも関わらず検証データの平均精度は上昇しなかったことから、この場合は epoch 数 100 では学習回数が不足していると考えられる。

課題 4

課題 3 をさらに発展させて、最適化手法に Adam[1] を用いることを考える。$\boldsymbol{g}_t = \nabla_\theta f_t(\boldsymbol{\theta}_{t-1})$ とすると、Adam では以下のようにパラメータ $\boldsymbol{\theta}$ を更新する。

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \alpha E[\boldsymbol{g}]/\sqrt{E[\boldsymbol{g}^2]} \tag{1}$$

Adam の特徴として、更新幅の絶対値が学習率 α 以下となることがある。また、勾配の向きの変化が小さい時は $E[\boldsymbol{g}]/\sqrt{E[\boldsymbol{g}^2]}$ が 1 に近い値を取り、α に近い更新幅となる。反対に勾配の向きの変化が大きいと、$E[\boldsymbol{g}]/\sqrt{E[\boldsymbol{g}^2]}$ が小さくなり、更新幅も小さくなる。
この期待値を指数移動平均で近似することを考える。ここで β をパラメータとすると指数移動平均 m_t は

$$\boldsymbol{m}_t = \beta \boldsymbol{m}_{t-1} - (1-\beta)\boldsymbol{g}_t \tag{2}$$

となり

$$\boldsymbol{m}_t = (1-\beta)\sum_{i=1}^{t} \beta^{t-i}\boldsymbol{g}_i \tag{3}$$

となる。ここで、

$$E[\boldsymbol{m}_t] \approx E[\boldsymbol{g}_t](1-\beta)\sum_{i=1}^{t}\beta^{t-i} = E[\boldsymbol{g}_t](1-\beta^t) \tag{4}$$

より、$(1-\beta^t)$ の項の補正をする必要がある。これは $E[\boldsymbol{g}^2]$ についても同様だから、更新式の全体は以下のように書ける。

$$\boldsymbol{m}_t = \beta_1 \boldsymbol{m}_{t-1} + (1-\beta_1)\boldsymbol{g}_t \tag{5}$$
$$\boldsymbol{v}_t = \beta_2 \boldsymbol{v}_{t-1} + (1-\beta_2)\boldsymbol{g}_t \tag{6}$$
$$\hat{\boldsymbol{m}}_t = \boldsymbol{m}_t/(1-\beta_1^t) \tag{7}$$
$$\hat{\boldsymbol{v}}_t = \boldsymbol{v}_t/(1-\beta_2^t) \tag{8}$$
$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \alpha\hat{\boldsymbol{m}}_t/\sqrt{\hat{\boldsymbol{v}}_t} \tag{9}$$

これを optimizers.py の Adam() に実装した。Adam で学習を行う際に用いる各種パラメータとしては、[1] を参考に以下のように決定した。ただし、集約ステップ数 T、特徴ベクトル次元数、学習エポック数、W,A,b の初期値は課題 3 の場合と同一のものを用いた。

表 2　Adam のパラメータ

学習率 α	0.0001
β_1	0.9
β_2	0.99
数値微分の摂動 ε	0.001
初期値 $\boldsymbol{m_0}$	全ての成分が 0
初期値 $\boldsymbol{v_0}$	全ての成分が 0

Fig3　B:10 の場合の平均損失・平均精度

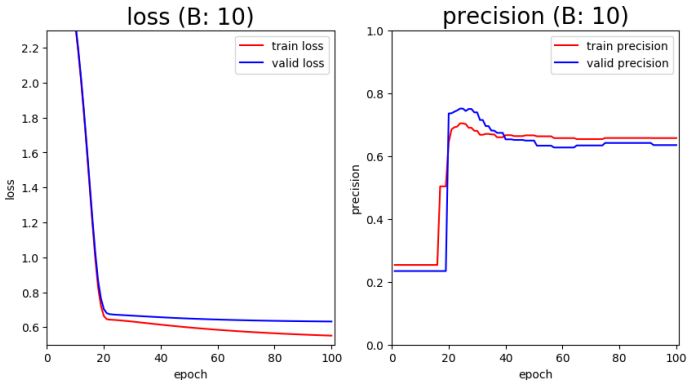
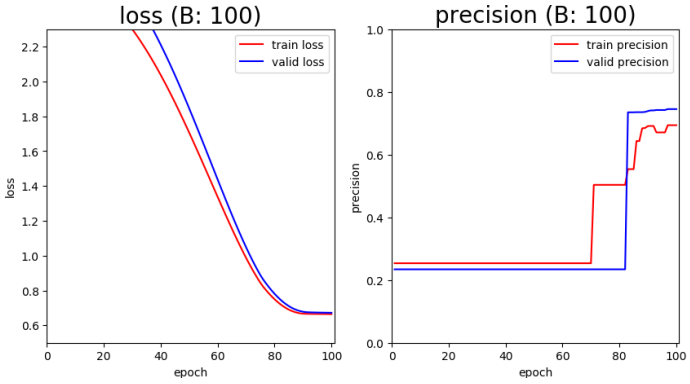


Fig4　B:100 の場合の平均損失・平均精度



B:10 の場合で SGD・MomentumSGD と Adam を比較すると、Adam は学習初期での loss の値が SGD・MomentumSGD に比べて大きいが、epoch 数 100 における loss の値は MomentumSGD よりも小さくなっており、MomentumSGD で loss が下がりにくくなった領域でも高い性能を発揮していることが分かる。また、初期の loss が大きい要因としては、Adam では更新幅の絶対値が学習率以下となるため、学習率 α の値が小さいと更新に時間がかかるためであると考えられる。
B:10 の場合では、epoch 数 24 で学習用データの平均精度が 0.7039、検証用データの平均精度が 0.7509 で共に最大になっており、この epoch 数での重み W,A,b の値を用いてテストデータのラベル予測を行なった。予測したラベルは prediction.txt に示してある。

参考文献

[1] Kingma, Diederik P., and Jimmy Ba. ”Adam: A method for stochastic optimization.” ICLR 2014.