

Preferred Networks 機械学習・数理分野 課題レポート

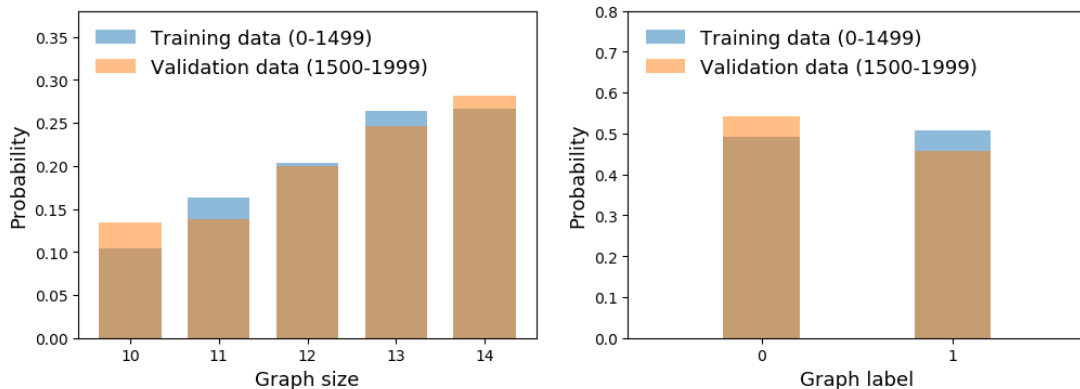
言語は Python3.6.4 を用いた. 用いたパッケージは, numpy1.14.2 と matplotlib2.1.2 である.

課題3

SGD, Momentum SGD のハイパーパラメータは課題文にあったものを採用した. ただし, ミニバッチのデータ数は $B=10$ とした.

• 学習データの確認

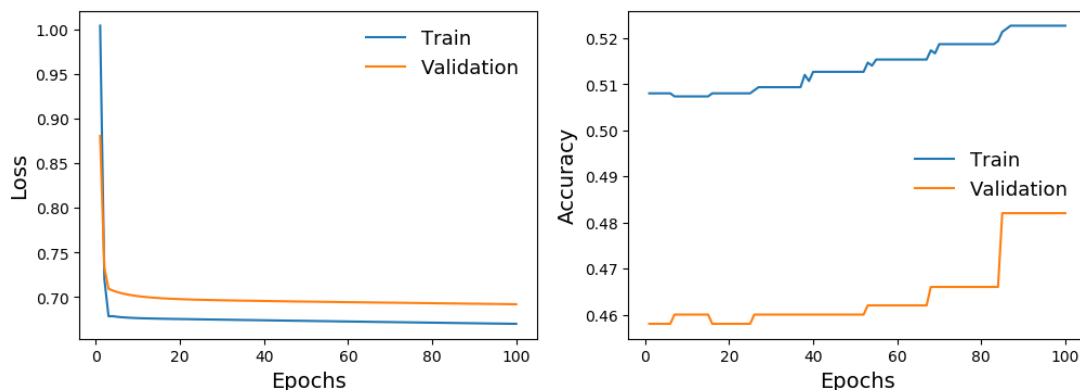
data/train 内のデータを学習用と検定用に分けるにあたって, データ内の分布を調べた. 特に, 0-1499 と 1500-1999 とでデータを分けた際にデータに偏りが無いかを調べた.



上のグラフから, 学習用と検定用データとでグラフのサイズおよびラベルについて大きな偏りはないと判断して, 1500-1999のidのデータを検定用とし, 残りを学習用データとした.

• SGDの実行

SGDに関するスクリプトは ./src/prob3_sgd.py にある. 100 epochs 回した際の学習用と検定用データに対する平均損失(Loss)と平均予測精度(Accuracy)の変化を以下に載せる.

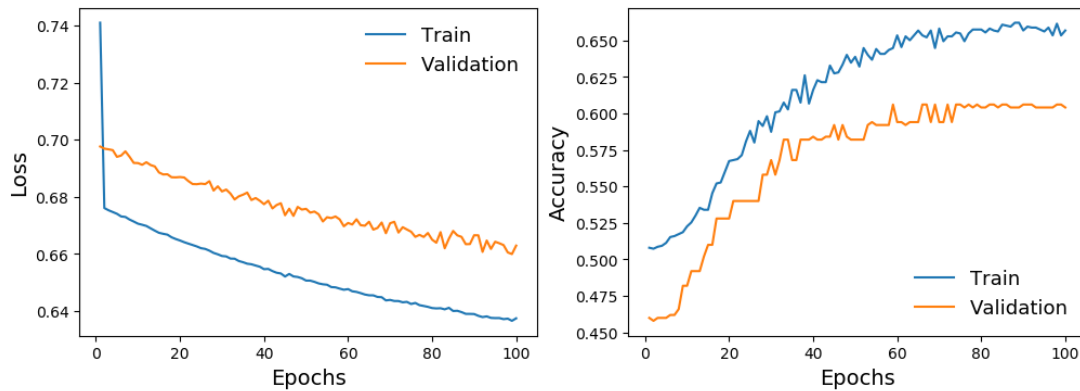


100 epochs 回したあとでも検定用データに対する平均予測精度が0.49に満たないことが分かる. データのほぼ半分のラベルが1なので, 全て1と予測した場合でも精度がおよそ50%になることを考えると, これは精度が悪い. ほかの初期値においても似た結果になることが確認された.

• Momentum SGD

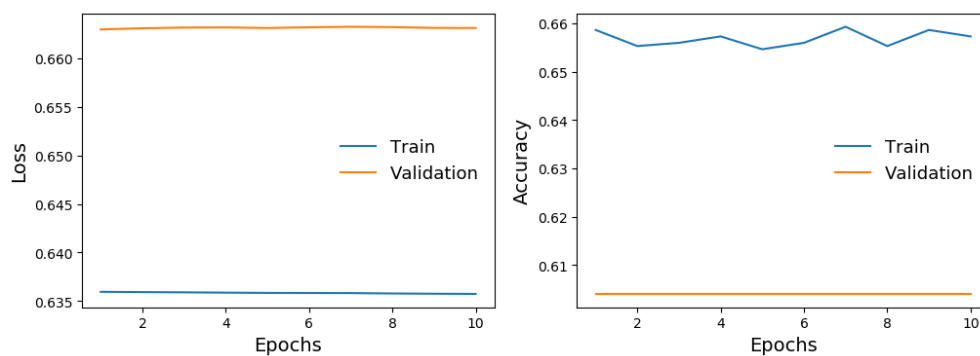
SGDに関するスクリプトは ./src/prob3_msgd.py にある. Momentum SGDを用いて100 epochs 回した際の学習用と検定用データに対する平均損失(Loss)と平均予測精度(Accuracy)の変化を以下に載せる.





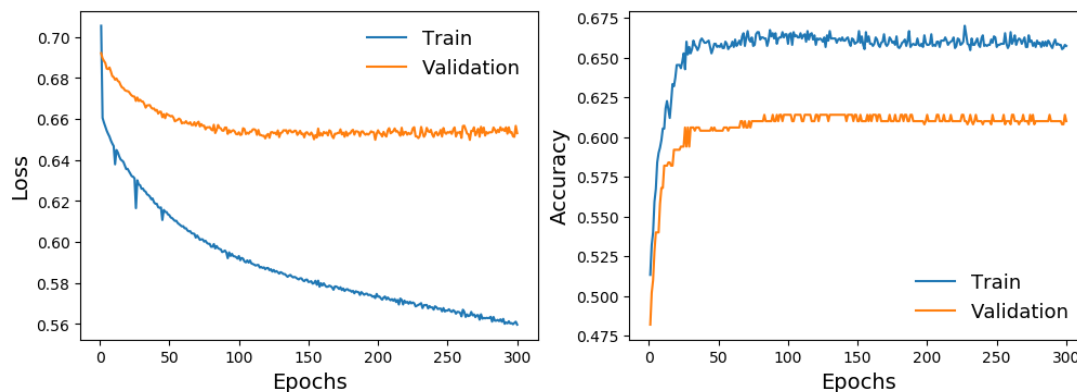
平均損失, 平均予測精度の両方に関して momentum SGD の方がより速くより良い成績を出していることが確認された (100epochsでの平均精度:0.604). 他の初期値に対してもおおよそ同様の傾向が見られた.

Momentum をつけたことによってノイズの影響を抑え, 小さい勾配を下れるようになったのだと考えられる. ちなみに momentum SGD で訓練したあとの W, A, b の値を初期値として SGD を回すと, 以下の結果が得られた. Momentum SGD で得られた解が SGD のノイズに対して安定だということを示唆している.



課題4 (Adam)

Adam を実装した. ハイパーパラメータは Kingma et al, 2014 を参考に $\beta_1 = 0.9, \beta_2 = 0.999$, 学習率 $\alpha = 0.001$, 発散を防ぐ定数 $\epsilon_2 = 10^{-8}$ とした. Adamを用いて300 epochs 回した際の学習用と検定用データに対する平均損失(Loss)と平均予測精度(Accuracy)の変化を以下に載せる (./src/prob4_adam.py).



• Momentum SGD との比較

平均損失, 平均予測精度の両方に関して Adam の方がより速くより良い成績を出していることが確認された (100epochsでの平均精度:0.614). 勾配の2次のモーメントを利用した学習率のadaptiveな更新によって, loss の振動を抑えられていることなどが効いていると考えられる.

• Overfitting の確認

学習用データで100 epochs を超えても loss が減り続けているのに対して, 検定用データでは100 epochs 以降 loss が下がらないことが確認された. 100 epochs程度を超えたところから学習用データに対する overfitting が進んでいると考え, 100 epochs 回したあとの W, A, b の値を用いて, テスト用データの予測を行った(./src/prob4_prediction.py , prediction.txt).