

ロボットインテリジェンス レポート課題

06150176 農学部生物環境工学専修 4 年 川田桃子

1. 本レポートの概要

本レポートでは、MNISTの手書き数字の画像セットを学習データとして用い、4層のBP学習で重みの更新を行うフィードフォワード型ニューラルネットの実装を行った。また、ノード数、活性化関数、初期値、学習係数を以下に記すように設定をし、学習データにおける判定精度と、テストデータにおける判定精度についてグラフで表した。並行して、今回は実装にはいたらなかったが現在のニューラルネットの精度向上や学習測度向上のために用いられている方法についても簡単にまとめた。しかしながら、今回のレポートにおいてはニューラルネットで使用されている手法について比較・考察ができるほどの実装ができなかった。

1.1 使用したデータセット

MNIST(<http://yann.lecun.com/exdb/mnist/>)の手書き数字の画像セットを学習と評価に使用した。このデータでは0~9までの手書きの数字の画像が、28x28のデータと正解ラベルとともに用意されている。画像は60000枚、テスト画像は10000枚ある。本レポートでは、あらかじめダウンロードしてnumpy配列に変換した(訓練データ、訓練ラベル)と(テストデータ、テストラベル)に分けたデータを.pklファイルとして保存して、データセットとして使用した。

1.2 設定した条件

- ・ 中間層のノード数: 30
- ・ 重みの初期値: 標準偏差1のガウス分布の乱数を使用
- ・ 活性化関数
中間層のニューロンモデルの出力値に作用させる活性化関数としてシグモイド関数を用いた。
また比較としてReLU関数も使用し精度の比較を行った。
- ・ 出力層の活性化関数
出力層の活性化関数としてはソフトマックス関数を用いた。また損失関数(誤差評価関数)としては交叉エントロピー誤差を用いた。
- ・ ミニバッチ処理: バッチ(200程度)でバッチ処理を行った。
- ・ 学習係数: 0.1
- ・ レイヤーの作り方: 積の計算レイヤ(重みとバイアスのかかる場所)、活性化関数のレイヤ(sigmoid, Relu)、評価関数のレイヤ(Softmax)に分けて作った。

2. 調べたことのまとめ・工夫点

2.1 初期値の設定について

初期値の工夫については、今回の実装では標準偏差1のガウス分布に0.01をかけたものを用いた。
重みの初期値に使用する関数ごとに適するものがあり、例えばReLUの場合には「Heの初期値」Sigmoidやtanh関数などのy軸に対して左右対称で中央付近で線形関数にみなせるものに対しては「Xavierの初期値」が適していると言われています。

2.2 one-hot 表現について

ラベルの表示の仕方。正解ラベルが、1の場合[0,1,0,0,0,0,0,0]のように表される表し方。今回の実装ではこれを用いた。

2.3 パラメーターの更新の方法:

私の実装においては、「stochastic gradient descent-SGD」を重みの更新の方法として使用したが他にもMomentum、AdaGradさらにMomentumとAdaGradを組み合わせたAdamという更新の方法もある。

2.4 各レイヤーの逆伝播の勾配を求める式について

計算グラフの考え方をういて、各レイヤーごとに順伝播（推論）方向の値の計算と逆伝播（BP）方向の値を求めた。
 $x*W + b$ の計算を行うレイヤーを DotLayer とした。DotLayer からの値を活性化関数に通すときのレイヤーを SigmoidLayer または Relu レイヤーとした。さらに、最後のレイヤーでは推論によって求められた値を SoftMax 関数に通した後に交叉エントロピー誤差を用いて最終的な推論を求めるように実装をした。
各レイヤーにおいて順方向の関数 foward() と逆方向の関数 back_propagation() を用意し、推論では各レイヤーの foward() を掛けあわせ、BP 学習では後ろのレイヤーから back_propagation()（各レイヤーの微分）を掛け合わせる実装をした。

3. 実行結果

まったく収束しなかった。活性化関数として Sigmoid 関数を用いたあとに Relu 関数で試してみたが、それも全く収束しなかったのので、BP 学習の実装のところで何か重大な欠陥があるのだと思うのだが、自分では見つけれなかった。

原因がわからず、それを苦闘していた結果ニューラルネットの実装だけに終わってしまった。

```
momoko@momoko-thinkpad-t430s:~/Documents/robot_intelligence$ python main.py
train accuracy: 0.09915 test accuracy: 0.1009
train accuracy: 0.11236666667 test accuracy: 0.1135
train accuracy: 0.11236666667 test accuracy: 0.1135
train accuracy: 0.10441666667 test accuracy: 0.1028
train accuracy: 0.11236666667 test accuracy: 0.1135
train accuracy: 0.11236666667 test accuracy: 0.1135
train accuracy: 0.11236666667 test accuracy: 0.1135
train accuracy: 0.11236666667 test accuracy: 0.1135
train accuracy: 0.11236666667 test accuracy: 0.1135
train accuracy: 0.11236666667 test accuracy: 0.1135
train accuracy: 0.11236666667 test accuracy: 0.1135
train accuracy: 0.11236666667 test accuracy: 0.1135
train accuracy: 0.11236666667 test accuracy: 0.1135
train accuracy: 0.11236666667 test accuracy: 0.1135
train accuracy: 0.11236666667 test accuracy: 0.1135
train accuracy: 0.11236666667 test accuracy: 0.1135
train accuracy: 0.11236666667 test accuracy: 0.1135
```

バッチ数や、イタレーション数を変えて計算してみても結果がほとんど変わらなかった。

3. 入門 Python3(オーム社)

4. <http://colah.github.io/posts/2014-10-Visualizing-MNIST/>

6. 講義の感想

Deep Learning についての説明をしている書籍やウェブサイトでは、各種ライブラリの使用についてのみ説明をしているものが多いと感じています。ニューラルネットの基本的考え方がどのようにできているのかについての解説はあまり見たことがなかったのですが、この講義では、「知能とは何か」からスタートしニューラルネットだけではなく生き物の行動がどのようなプロセスを経て行われているのかについて学ぶことができ、新鮮でした。

実装にあまり慣れていない中で A に挑戦してみた結果、見事に玉砕してしまい、最後まで課題をやり通せなかったことが残念でした。

締め切りには間に合わなかったのですが、今後も自分で学習を進めていきたいと思っています。

人工知能関連の技術の敷居が、この授業のおかげでさがった気がします。ありがとうございました。