2016年 Preferred Infrastructure / Preferered Networks インターン選考コーディング課題

変更履歴

- 2016年5月13日：初版
- 2016年5月16日：第2版
  - 課題3の疑似コードにバイアス項の学習を追加
  - 課題4の損失関数平均の目標を10から0.1に変更

- 2016年5月20日：第3版
  - 課題3の疑似コードでバイアス項の更新式を訂正
  - 課題4の問題文を修正
    - 損失関数平均の目標を0.1から2.5に変更
    - 隠れ層の次元を明記
    - 疑似コードの補足説明を追加

  - 課題4の疑似コードを修正
    - 訓練と評価を分離
    - オプショナルな訓練データのシャッフルを追加

回答にあたっての留意事項

制約

- 課題1-4では、各言語の標準ライブラリの関数のみを使用し、NumPy, Eigenなど多次元配列ライブラリは使用しないで実装してください。
- 課題5ではライブラリを自由に使用して頂いて構いません。
- プログラムの回答には以下のいずれかの言語を利用してください。
  - C, C++, Python, Ruby, Go, Java

提出物

- 課題1-4はプログラムを、課題5はプログラムとレポートを提出してください。
- プログラムはできるだけ他人が読んでも分かりやすいものになっており、また追試がしやすい形になっていることが望ましく、レポートはわかりやすくまとまっているのが望ましいです。
- プログラムのビルドに必要な環境・実行環境・実行手順について記述してください。

- プログラムの説明を記述した補足資料（A4用紙1枚以内）を添付しても構いません。

提出先

- intern-apply@preferred.jp（応募時と同一のメールアドレスです）
- 課題に関する質問などもこのアドレス宛にお願いいたします

問題文

本課題では、Auto Encoderを実装します。Auto Encoderとは、ニューラルネットワークの一種で、入力に（一般的に非線形な）変換を繰り返し行い、出力が入力を復元するように訓練されるモデルです。具体的には入力をベクトル x とすると、Auto Encoderはパラメータとして n 個の行列 W_i と n 個のベクトル b_i $(i = 1, \dots, n)$ を用いて、

$$h_i = f_i(W_i h_{i-1} + b_i) \quad \text{for} \quad i = 1, \dots, n$$

と変換し、$y = h_n$ を出力します。但し、$h_0 = x$ と定義し、f_i は活性化関数と呼ばれるある固定の（一般的には非線形な）関数です。Auto Encoderの学習とは、訓練データと呼ばれるデータセット $\{x_j\}_{j=1}^N$ を用いて、Auto Encoderの出力値 y が入力値 x に近づくように W_i, b_i 達を調節することを指します。各 h_i が x よりも高次元の場合、恒等関数を学習すれば良いので、通常何らかの制限（h_i 達は x よりも次元を小さくするなど）を課します。

今回は最も単純なAuto Encoderである、隠れ層1層（すなわち、$n = 2$）で、活性化関数が恒等関数 ($f(x) = x$)の場合を実装し、訓練データを用いて学習します。

課題1

ベクトル x とベクトル y の外積を計算する関数を書いてください。ベクトル $x = (x_1, \dots, x_n)$ とベクトル $y = (y_1, \dots, y_m)$ の外積は $n \times m$の行列でその (i, j) 成分は $x_i \times y_j$ で定義されます。関数の定義例は以下の通りです。

Python

```
def outer(x, y): pass
```

C++

```
std::vector<std::vector<float> > outer(const std::vector<float>& x, const std::vec
tor<float>& y);
```

課題2

入力ベクトル x を重み行列 W とバイアスベクトル b を用いて、アファイン変換を行う関数を書いてください。アファイン変換とは $y = Wx + b$ という変換を指します。関数の定義例は以下の通りです。

Python

```
def affine(x, W, b): pass
```

C++

```
std::vector<float> affine(const std::vector<float>& x, const std::vector<std::vect
or<float> >& W, const std::vector<float>& b);
```

課題3

学習の1イテレーションを実装してください。Auto Encoderの隠れ層の次元は5次元としてください。課題1, 2で作成した関数を利用しても構いません。1イテレーションはニューラルネットの順方向に計算を進めて損失を計算する順伝播、逆方向に計算を進めて損失の各変数に関する勾配を計算する逆伝播、勾配を用いたパラメータ更新の3ステップからなります。

1訓練データ x を用いた1イテレーションの疑似コードは以下の通りです：

順伝播

$$
\begin{aligned}
h &= W_1 x + b_1 \\
y &= W_2 h + b_2 \\
loss &= \frac{1}{2} \sum_i (x_i - y_i)^2
\end{aligned}
$$

逆伝播

$$
\begin{aligned}
gy &= y - x \\
gW_2 &= (gy) \otimes h \\
gb_2 &= gy \\
gh &= W_2^T (gy) \\
gW_1 &= (gh) \otimes x \\
gb_1 &= gh
\end{aligned}
$$

パラメータ更新

$$
\begin{aligned}
W_2 &\leftarrow W_2 - \eta(gW_2) \\
b_2 &\leftarrow b_2 - \eta(gb_2) \\
W_1 &\leftarrow W_1 - \eta(gW_1) \\
b_1 &\leftarrow b_1 - \eta(gb_1)
\end{aligned}
$$

ここで、gX は $loss$ の X に関する勾配 $\frac{\partial loss}{\partial X}$ を表し（Xがベクトルもしくは行列の場合には要素ごとの偏微分を表します）、η は学習率を表す定数です。また、$a \otimes b$ はベクトル a とベクトル b の外積、X^T は行列 X の転置行列です。

課題4

本文書に添付されているデータセット（dataset.dat）を用いて、Auto Encoderを学習するコードを実装してください。また、全訓練データに渡る損失関数の平均（下の疑似コードの `average_loss` に対応）を2.5以下まで減らし、最終的な学習パラメータ（ `W1` , `W2` , `b1` , `b2` ）を提出してください(下記参照)。課題3で作成したコードを利用しても構いません。隠れ層の次元は課題3と同様に5次元としてください。

学習のプロセスの疑似コードは以下の通りです。疑似コード中の `xs` は訓練データセットを表し、 `number of elements in xs` は、訓練データセット `xs` のデータ数（後述の N）を表します。

```
initialize W1, W2, b1, b2
initialize eta
for i = 1 to epoch_num
    # 学習
    # ...（オプショナル：訓練データセットxsをシャッフル）...
    for each x in xs
        # ...（学習の1イテレーション）...

    # 評価
    total_loss = 0
    for each x in xs
        # ...（順伝播によりlossを計算）...
        total_loss <- total_loss + loss
    average_loss = total_loss / (number of elements in xs)
```

添付データセット（dataset.dat）の形式は以下の通りです。1行目は訓練データ数 N と、次元数 D を表す2つの整数がスペース区切りでこの順番に並んでいます。2行目から $N+1$ 行目は訓練データであり、1行が1訓練データを表します。1行はスペース区切りで、d 番目の実数値は訓練データの d 次元目の値を表します。以下は訓練データセットの例です。

```
1000 10
-0.59013806759 -0.0241665967954 0.601114241516 0.976326346575 -0.448981979213 0.91
6407144004 -0.641780347732 -0.805810167716 0.84013636913 -0.1506155968
0.934965712733 0.991708435645 0.445449116474 0.505073413079 -0.584120911354 -0.745
970828327 -0.676474864127 0.127201137168 0.996230538804 -0.21455657636
...（997行省略）...
-0.675096614497 -0.852168820875 -0.577230899139 -0.438220803993 -0.133447379578 0.
0839661500525 -0.375295865497 -0.973532769499 0.656025794121 0.803297676031
```

学習したパラメータは以下の形式でファイルに出力して提出ください。それぞれのパラメータの間はスペース区切りにしてください。

```
W1_{1,1} ... W1_{1,10}
...
W1_{5,1} ... W1_{5,10}
W2_{1,1} ... W2_{1,5}
...
W2_{10,1} ... W2_{10,5}
b1_{1} ... b1_{5}
b2_{1} ... b2_{10}
```

課題5

課題4のパラメータや実装を変えた場合に、実験結果がどのように変わるか報告してください。また、変更したパラメータの中から一番良いパラメータを決定してください。精度は交差検定で測定するのが望ましいです。レポートはA4用紙2枚以内（図・表などを含む）に収めてください。

変更する項目としては例えば以下のようなものがあります。

- AutoEncoder の隠れ層の層数・各層の次元数を変えた場合に、収束に掛かる計算時間がどのように変化するか
- モデルをAuto Encoderの亜種である Denoising Auto Encoder に変えた場合、混入させるノイズの量によって収束に掛かる計算時間がどのように変化するか
- モデルの初期化方法を変えた場合に精度がどのように変化するか
- 活性化関数を変えた場合に精度がどのように変化するか
- 入力をミニバッチ化した場合に計算時間・精度がどのように変化するか

これらは一例であり、全てを実施する必要はありません。時間が足りない場合は重要だと思われる項目だけを実験してください。また、これら以外のパラメータを変更しても構いません。

（課題文ここまで）