

知能プログラミング演習 I 演習課題

梅津 佑太

umezu.yuta@nitech.ac.jp

2018 年 6 月 13 日

1 準備

- 前回出席できなかった人で, まだ Anaconda の仮想環境を構築していない人
 - ホームディレクトリに演習用のディレクトリを作成し, DLL に移動

```
step1: mkdir -p DLL
```

```
step2: cd ./DLL
```
 - 今日の課題を DLL にダウンロードし, 展開する

```
step1: wget http://www-als.ics.nitech.ac.jp/~umezu/Lec2.zip
```

```
step2: unzip Lec2.zip
```
 - 展開したフォルダの中に, 以下のものがすべて入っていることを確認し, Lec2 に移動
 - * NN.py
 - * task.pdf
 - * train (フォルダ)
 - * test (フォルダ)

```
step1: cd ./Lec2
```
 - Anaconda の仮想環境 (myenv) を構築

```
step1: /opt/cse/pkg/anaconda3-5.0.1/bin/conda create --name myenv python=3.6
```

```
step2: source /opt/cse/pkg/anaconda3-5.0.1/bin/activate myenv
```
 - インストール済みのパッケージを確認し, matplotlib と seaborn を (インストールされていなければ) インストール

```
step1: pip list
```

```
step2: matplotlib と seaborn が確認できなければインストール
```

 - * `pip install matplotlib`
 - * `pip install seaborn`
- すでに Anaconda の仮想環境がある人
 - 作業ディレクトリ DLL に移動

```
step1: cd ./DLL
```
 - 今日の課題を DLL にダウンロードし, 展開してから Lec2 へ移動

```
step1: wget http://www-als.ics.nitech.ac.jp/~umezu/Lec2.zip
```

- step2: `unzip Lec2.zip`
- step3: `cd ./Lec2`
- Anaconda の仮想環境上で python を扱う
 - step1: `source /opt/cse/pkg/anaconda3-5.0.1/bin/activate myenv`

2 課題

手書き文字 (28× 28 ピクセル) の多値分類 (0: 2500 枚, 1: 2500 枚, 2: 2500 枚, 3: 2500 枚)^{*1}を実装する。以下のプログラムを作成せよ。ただし, NN.py にコードを保存すること。

1. 以下の関数を定義せよ。

(a) ReLU とその微分。なお, 実数 x に対して, ReLU は以下で定義される。

$$f(x) = \max\{0, x\}$$

(b) ソフトマックス関数。なお, m 次元の実ベクトル $\mathbf{x} = (x_1, \dots, x_m)^\top$ に対して, ソフトマックス関数は以下で定義される。

$$g(\mathbf{x}) = \frac{1}{\sum_{k=1}^m e^{x_k}} \begin{bmatrix} e^{x_1} \\ \vdots \\ e^{x_m} \end{bmatrix}$$

(c) クロスエントロピー。なお, クロスエントロピーは, ソフトマックス関数 $g(\mathbf{x})$ と m 次元の出力ラベル $\mathbf{y} \in \mathbb{R}^K$ に対して以下で定義される。

$$E(\mathbf{x}, \mathbf{y}) = - \sum_{j=1}^m y_j \log g_j(\mathbf{x}).$$

ただし, $g_j(\mathbf{x})$ はソフトマックス関数の出力の j 番目の成分を表す。

(d) 3 層ニューラルネットワークの順伝播を完成させよ。ただし, パラメータと入力の線形結合 $\mathbf{v} = W\mathbf{x}$ に対して,

- 活性化関数の値 $z = f(\mathbf{v})$ とバイアス項 1 を結合した配列 $(1, z^\top)^\top$
- 活性化関数の微分の値 $f'(\mathbf{v})$

を出力すること。

(e) 3 層ニューラルネットワークの逆伝播を完成させよ。必要なら, (d) で出力した活性化関数の微分の値 $f'(\mathbf{v})$ を用いても良い。

2. 3 層ニューラルネットワークのアルゴリズムを完成させよ。ただし, 活性化関数として ReLU を用いることとし, パラメータ更新の際の学習率は確率的勾配降下法のアルゴリズム中に定義している `eta_t` を用いること。

ヒント: 順伝播, 誤差評価, 逆伝播, パラメータの更新の順に行う。ただし, 誤差評価の部分はすでに記述しているので, それ以外のパーツをうめる。

3. confusion matrix を計算するプログラムを完成させよ。必要なら, for 文を使っても良い。

4. NN.py を実行し, 訓練誤差とテスト誤差が減少している様子を確認せよ。さらに, confusion matrix で分類結果がどのようなものか, また, 誤分類されたデータがどのようなものか確認せよ。

注意: NN.py 中のこの部分は全てうめているので, 結果を確認するだけで良い。

^{*1} テストデータは各 250 枚の 0, 1, 2, 3 の計 1000 枚

3 課題の提出

Moodle を使ってファイルを提出してください。提出方法は以下の通りです。

- Moodle にログインし, 知能プログラミング演習のページへ移動。
- Lec2 の項目に, NN.py をアップロードする。

6/19(火) の 17:00 (次回の授業前日) を提出期限とします。