

# 知能プログラミング演習 I 演習課題

## 1 準備

### 1.1 自前の環境の場合

- Moodle から課題を各自任意のフォルダにダウンロードし, 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
  - CNN2.py
  - CNN3.py
  - CNN4.py
  - report.tex
  - report.pdf
  - task.pdf
- **torch** と **torchvision** をインストールする. pip なら

```
$ pip install torch
$ pip install torchvision
```

### 1.2 CSE の場合

- まだ演習用のフォルダを作っていない人は DLL のフォルダを作成
  - ホームディレクトリに演習用のディレクトリを作成
   
step1: `mkdir -p DLL`
- 作業ディレクトリ DLL に移動
   
step1: `cd ./DLL`
- 今日の課題を DLL にダウンロードして展開
  - 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
    - \* CNN2.py
    - \* CNN3.py
    - \* CNN4.py
    - \* report.tex
    - \* report.pdf
    - \* task.pdf

- Lec7 へ移動

step1: cd ./Lec7

## 2 課題

PyTorch を用いて、多値分類を畳み込みニューラルネットワークで実装する。今回の課題では、レポートも提出する。テンプレート report.tex を使ってもよいし、使わずに各自適当なフォーマットで用意してもよい。

- CNN2.py (講義スライドで説明したプログラム) の `forward` 関数において以下のようにすると `x` のサイズを出力して終了する。

```
def forward(self, x):
    print(x.size())
    sys.exit()
    x = self.conv(x)
    x = self.relu(x)
    x = self.maxpooling(x)
    x = self.flatten(x)
    logits = self.linear(x)
    return logits
```

`forward` の各行 (`self.conv` の前, `self.relu` の前, `self.maxpooling` の前, `self.flatten` の前, `self.linear` の前, `return logits` の前) での `x` のサイズ (最後のみ logit のサイズ) を調べて記載せよ。また、それぞれのサイズがなぜそのようになるのか説明せよ。この課題はコードは提出不要とする。

- CNN3.py の CNN クラスを編集して、以下で定義される CNN を作成せよ。指定のない設定はデフォルトでよい。ただし、(a)-(g) を `self.fe` に、(h)-(j) を `self.fc` に Sequential を使ってまとめること。

- (a) 畳み込み層: 出力チャネル数 6, フィルタサイズ 3, ストライド 1, パディング 1
- (b) ReLU 活性化関数
- (c) マックスプーリング: フィルタサイズ 2, ストライド 2
- (d) 畳み込み層: 出力チャネル数 6, フィルタサイズ 3, ストライド 1, パディング 1
- (e) ReLU 活性化関数
- (f) マックスプーリング: フィルタサイズ 2, ストライド 2
- (g) Flatten
- (h) 線形層: 出力 100 次元
- (i) ReLU 活性化関数
- (j) 線形層: 出力 10 次元

ヒント: 最初の線形層の入力サイズを指定する際には 1. で行なったサイズ確認方法で確かめることができる。プログラムを完成させて走らせると error.pdf が作成される。この図には何がプロットされていてどのようなことが読み取れるのか、なるべく詳しく説明せよ。

- CNN4.py は Fashion MNIST と呼ばれる  $28 \times 28$  のグレースケール、10 クラスのデータを読み込んで CNN を実行するためのテンプレートになっている。ただし、CNN クラスの `_init_` と `forward` は空欄になっているので、好きなようにネットワークを設計して、最終的な正解率 (最後に表示される

Final Accuracy) がなるべく高くなるような CNN を作成せよ。少なくとも 1 つの畳み込み層を使うこと。その他の設定は紹介していない機能など含めて好きに設計してよい(参考: 利用可能な層のリファレンス <https://pytorch.org/docs/stable/nn.html>, 利用可能な最適化アルゴリズムのリファレンス <https://pytorch.org/docs/stable/optim.html> )。最終的に採用した設定および結果について説明すること。



図 1 Fashion MNIST データの各クラスのサンプル (T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, Ankle boot)

### 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。
- Lec7 の項目に、CNN3.py, CNN4.py, レポートをアップロードする。

7/19(金) の 17:00 を提出期限とします。