

卒業研究 現在の進行状況

2026年 月 日 923044 高宮 悠聖

1. 現在の進行状況

1.1. Social-LSTM プログラム把握

- ・ 内容を変更するにあたって使用するプログラムの中身を確認
 - ▶ コメントアウトで分かりやすいように説明を記述

```
social-lstm M visualizer.py M
social-lstm > utils.py > DataLoader > frame_preprocess
1 import os # OS操作ライブラリ フォルダ作成,ファイル存在確認,パス結合などに使用
2 import pickle # Pythonオブジェクト保存用,Social-LSTMでは,txtデータを前処理後,.cpkl ファイルとして保存などに使用
3 import numpy as np # 数値計算ライブラリ,座標配列管理,歩行軌跡保存,フレーム操作,行列計算
4 import pandas as pd # 表形式データ処理ライブラリ,txt/csv読み込みに使用
5 import random # ランダム処理用,validationデータ選択などに使用
6 import torch # PyTorch本体,AI計算ライブラリ,Tensor計算,LSTM,GPU計算,誤差逆伝播
7 import math # 数学関数ライブラリ
8 from torch.autograd import Variable # PyTorchの学習可能Tensor,現在のPyTorchではTensorと統合されているため本来不要,ただし古いSocial-LSTM実装では使用
9 from helper import * # helper.py にある便利関数を使用,utils.py の補助役
10
11
12 # DataLoaderクラス
13 # 役割:「実行データをAI学習できる形に変換」,train.py から呼ばれる
14 # 流れ:txtデータ→DataLoader→sequence化→model.pyへ入力
15 class DataLoader():
16     def __init__(self,
17                 f_prefix, # Social-LSTM のプロジェクトのルートフォルダ位置,絶対パスを絶対パスに変換している
18                 batch_size=5, # 一度に学習する系列数
19                 seq_length=20, # 1系列のフレーム長
20                 num_of_validation=0, # validationに使用するデータ数
21                 forcePreProcess=False, # 前処理を強制再実行
22                 infer=False, # 推論(test)モードか
23                 generate=False # データ生成モード
24             ):
25
26     # テストデータ一覧
27     # 学習後の性能評価用,各txt:frame_id ped_id y x を持つ
28     base_test_dataset = [
29         '/data/test/biwi/biwi_eth.txt', # BIWI dataset
30         '/data/test/crowds/crowds_zara01.txt', # crowds dataset
31         '/data/test/crowds/uni_examples.txt',
32     ]
```

1.2. システム内容

1. 人の座標を入力

- ・ (x_t^i, y_t^i) , i : 人物番号, t : 時刻

2. 各人物ごとに LSTM を持つようにする

- ・ LSTM:歩く速度、曲がり方、加速、減速などを記憶

3. Social Pooling を使う

- ・ Social Pooling:周囲の人の LSTM 状態(hidden state)を共有するシステム

$$H_t^i(m, n, :) = \sum_{j \in N_i} 1_{mn} [x_t^j - x_t^i, y_t^j - y_t^i] h_{t-1}^j$$

$$H_t^i(m, n, :)$$

時刻 t のとき i 番目の人がグリッド位置 m, n 内にいるときの hidden state を集めた周囲情報マップ

$$\sum_{j \in N_i} 1_{mn} [x_t^j - x_t^i, y_t^j - y_t^i] h_{t-1}^j$$

i 番目の人の近くにいる人が縦横どのくらいの距離にいるか, (m, n) マスにいるのか, いるならその人の LSTM 記憶(hidden state)をまとめる

4. LSTM に周囲情報を入力

- ・ 現在位置 + 周囲の人の状態 → LSTM 更新 → 次の位置予測
- ・ $e_t^i = \varphi(x_t^i, y_t^i; W_e)$: 座標を特徴ベクトルに変換する式
- ・ $a_t^i = \varphi(H_t^i; W_a)$: 周囲の人の hidden state を位置ごとに並べる式
- ・ $h_t^i = \text{LSTM}(h_{t-1}^i, e_t^i, a_t^i; W_l)$:

前回 LSTM、自分の位置情報、周囲の人情報を組み合わせて LSTM 更新

5. 次の位置を確率分布として予測

- ・ 二次元ガウス分布を使用

$$(\hat{x}, \hat{y})_t^i \sim N(\mu_t^i, \sigma_t^i, \rho_t^i)$$

μ_t^i : 予測位置の中心, σ_t^i : 予測の広がり, ρ_t^i : x 方向と y 方向の関連

2. 今後の予定

- ・ 使用するプログラムの中身確認の続き
- ・ 数値変えて何回か実行