

全体ゼミでの振り返り

2026 年 6 月 19 日

923044 高宮悠聖

今週行ったこと(6/19)

- ・ Social-LSTM の複数回実行
- ・ 今後の予定

Social-LSTM の複数回実行

実行手順

1. プロジェクトディレクトリに移動
2. Python 仮想環境を起動する
[source venv/bin/activate]
3. モデルの学習

python train.py: 学習メイン処理を起動する

[python train.py -num_epochs 30 -batch_size 5 -learning_rate 0.003]

- ・ python train.py: 学習メイン処理を起動
- ・ -num_epochs 30: データを何周させて重みを鍛えるか
- ・ -batch_size 5: 1 回のパラメータ更新で同時に計算するシーケンスの数
- ・ -learning_rate 0.003: 予測間違いを修正するときどれだけパラメータを動かすか

結果の確認方法: ターミナルにログが表示される

Social-LSTM の複数回実行 (ii)

4. 未知の未来の予測テスト

```
[ python test.py -epoch 29 -obs_length 8 -pred_length 12 -method 1 ]
```

- ・ python test.py: テスト処理を起動
- ・ -epoch 29: train.py で保存された中から、29 番目のエポック重みを指定してロード
- ・ -obs_length 8: 最初の 8 フレーム分を過去に目撃したデータとして AI に入力
- ・ -pred_length 12: 続く 12 フレーム分の未来の軌跡を AI に完全自動予測する
- ・ -method 1: 予測モデルの種類として Social-LSTM を指定する

結果の確認方法:

- ・ ターミナルに数値成績が表示される
- ・ result/SOCIALLSTM/ に予測された座標が記録されたテキストファイル (.txt) が出力される
- ・ plot/SOCIALLSTM/LSTM/test/ に、次の可視化ステップに引き渡すための予測データ (.pkl) が生成される

Social-LSTM の複数回実行 (iii)

5. 予測結果のグラフ画像・動画化

```
[ python visualize.py -num_of_data 5 -method 1 ]
```

- ・ python visualize.py: 可視化スクリプトを起動
- ・ -num_of_data 5: test.py の予測結果から、ランダムに5つ選んでビジュアル化する
- ・ -method 1: 対象モデルとして「Social-LSTM」を指定する

結果の確認方法:

- ・ plot/SOCIALLSTM/LSTM/plots/ に正解の線と AI の予測線が描かれた PNG 画像 (.png) が生成される
- ・ plot/SOCIALLSTM/LSTM/videos/ に歩行者の動きを再現した MP4 動画 (.mp4) が自動生成される

6. 結果のバックアップ

- ・ 実験ごとに ADE(平均変位誤差) や FDE(最終変位誤差) の数値をメモする
- ・ 生成された plots/ フォルダの画像などの名前を変えて保存し直す
- ・ 一から学習する場合は、model/ フォルダの中身を空にするか、別の場所に移動させてから train.py を動かす

実験結果

epochs 30 batch_size 5 learning_rate 0.003 の場合

[epoch 29]

valid_loss = 1.636,

valid_mean_err = 0.882,

valid_final_err = 1.776

Best epoch 12

Best valid_loss = 1.273

Best valid_mean_err = 0.886

Best valid_final_err = 1.699

epochs 15 batch_size 5 learning_rate 0.003 の場合

[epoch 14]

valid_loss = 3.407,

valid_mean_err = 0.910,

valid_final_err = 1.848

Best epoch 3

Best valid_loss = 2.516

実験結果 (ii)

Best valid_mean_err = 1.014,

Best valid_final_err = 2.057

epochs 30 batch_size 5 learning_rate 0.0015 の場合

[epoch 29],

valid_loss = 1.735,

valid_mean_err = 0.899,

valid_final_err = 1.843

Best epoch 14

Best valid_loss = 1.460

Best valid_mean_err = 0.918,

Best valid_final_err = 1.849

epochs 15 batch_size 5 learning_rate 0.0015 の場合

[epoch 14],

valid_loss = 1.681,

valid_mean_err = 0.871,

valid_final_err = 1.709

Best epoch 7

実験結果 (iii)

Best valid_loss = 1.607

Best valid_mean_err = 0.933,

Best valid_final_err = 1.885

Epochs / LR	Best Epoch	Best Loss	Best ADE	Best FDE	評価
30 / 0.003	12	1.273	0.886	1.699	◎
15 / 0.003	3	2.516	1.014	2.057	△
30 / 0.0015	14	1.460	0.918	1.849	○
15 / 0.0015	7	1.607	0.933	1.885	○

考察

1. epochs を 15 に減らすと性能が悪化した

epoch12 付近から過学習しているように見えたと考えたが結果を見ると、

lr=0.003 の場合

epochs=30 → ADE 0.886 epochs=15 → ADE 1.014

lr=0.0015 の場合

実験結果 (iv)

epochs=30 $\rightarrow$ ADE 0.918 epochs=15 $\rightarrow$ ADE 0.933

15epoch では十分に学習できていなかった可能性が高いと考えられる

実験結果 (v)

2. learning_rate=0.003 の方が良い

学習率を下げることで安定化ができるかと考えたが今回は学習が遅くなりすぎて十分な最適化ができなかったと思われる

lr	Best ADE	Best FDE
0.003	0.886	1.699
0.0015	0.918	1.849

3. 「Best epoch」 がかなり早い

どの実験も、全学習終了時 ≠ 最高性能時になっている

つまり、学習後半では性能改善がほとんど起きていないと思われる

設定	Best epoch
30	0.003 12
15	0.003 3
30	0.0015 14

実験結果 (vi)

15	0.0015 7
----	----------

今後の予定

- ・ もう少し数値を変更して実験してみる
- ・ 現在のデータセットではなく、外部にあるデータセットを用いて実装する
- ・ Yolo と接続して動画から人物の座標を取得し、取得した座標をもとに Social-LSTM の実行を行う