米粒分類モデルの実装

1. 目的

2年の機械学習から今のデータマイニング概論までに学んだことを復習するため. また, 松尾研の講義で学んだことの復習も兼ねて, 実装できそうなことを実践する.

2. 概要

1粒ずつ写真に撮られた整流米と割米の画像データを分類するプログラムを作成する. 訓練データのみ与えられ, それぞれ146枚(整流米), 231枚(割米)用意されている.

3. 手順

3つファイルがあるため, 順番に実行する必要がある.

3.1 全てのファイルの用意

- 訓練データは"./data/train/broken", "./data/train/proper"に配置する.
- 新たに分類を確認したい画像データは, "./data/img"に配置する.

3.2 データ拡張

1. プログラム内にある「target_count」という変数に目標とする画像の枚数を指定する.
2. コマンドラインで「python data_augmentation.py」を実行する.
3. "./data/augmented"の中に, プログラム内で指定した枚数まで画像が増えている.

3.3 モデル学習と閾値決定

1. コマンドラインで「python model_training.py」を実行する.
2. モデルの学習と調整ができる. 交差検証を観察し, 必要なハイパーパラメータに調整する.
3. モデル「best_rice_classification_model.keras」が生成されたことを確認し, コマンドラインに表示された閾値の値を控えておく(ROCとPRの2つがあるが, どちらを試しても良い).

3.4 分類と精度確認

- 分類したいデータのみを見たい場合： コマンドラインで「python .\kometsubu.py ./data/img」を実行する.
- 訓練データで使ったデータで精度も確認したい場合： 「python .\kometsubu.py ./data/img ./data/train」を実行する.
- 分類したいデータは, 整流米の場合「proper」, 割米の場合「broken」と表示される.
- 訓練データの精度については, 正解率と詳細なレポートがコマンドラインに表示される.

4. 方法

4.1 データ拡張

- 2年の手書き図形バトルで同じような画像水増しプログラムを組んだため, それを改良している.

- なるべくランダムにするために, 回転, フリップ, 明るさ, ズーム, ノイズなどを加えながら水増しする方法を取る.

4.2 モデル学習

- 畳み込みニューラルネットワークの訓練, 評価, 最適化を, K-fold交差検証を用いながら行う.
- モデルの構造は, 2つの畳み込み層, 2つのプーリング層, 2つの全結合層で構成している.
- 過学習を防ぐために, L2正則化とドロップアウトを使用, 学習率の調整とEarlyStoppingを利用している.

4.3 分類と評価

- 生成したkerasファイルのモデルとROC曲線, PR曲線を用いて最適化された閾値を利用し, 二値分類を行う.
- また, 訓練データを用いてモデルの性能評価を行い, AccuracyとF1スコアを算出する.

5. 結果

5.1 データ拡張

data_augmentation.pyではそれぞれ2000枚ずつのデータの生成に成功した.

5.2 モデル学習

model_training.pyの最終モデルの学習の進捗データは以下の通り：

- accuracy: 0.9048
- loss: 0.5088
- val_accuracy: 0.9187
- val_loss: 0.4516
- learning_rate: 1.0000e-04

ROC曲線とPR曲線については以下の通り：

- ROC曲線に基づく最適な閾値: 0.5939
- PR曲線に基づく最適な閾値: 0.5339
- ROC閾値の性能 - 精度: 0.9437, F1スコア: 0.9466
- PR閾値の性能 - 精度: 0.9437, F1スコア: 0.9470

5.3 分類結果

訓練データでの分類の正確性：Accuracy: 0.9841

6. 考察

- 非常に高い精度で米粒の分類を行うことができた.
- loss値の停滞域を見てepoch数やコールバックの調整をしたため, 過学習も避けられているのではないかと思われる.
- 交差検証を用いることがモデルの汎化性能の向上の手助けになった.
- データ拡張により, 元のデータセットの不均衡を解消し, モデルの学習に十分なデータ量を確保できたことで, 過学習のリスクを低減し, より堅牢なモデルを構築することができたと考えられる.

7. 感想

- 詰め込めるもの詰め込んだので, 途中から自分でも全体を把握するのが難しくなった.
- 要改善な点と言えば, コールバック使用時の監視対象かなと思う. 二値分類なのでaccuracy中心に見ればいいが, 今回検証データが用意されていないので, 過学習していないかどうかの見極めがaccuracyだけではできない.
- 正直もっとハイパーパラメータの改善は出来たと思う.
- 色々な復習の機会として課題を使わせていただきました.