

機械学習を用いたポケモンの タイプ判定

Group 4

上原 由宇駆・上原 一真・白石 貴祥





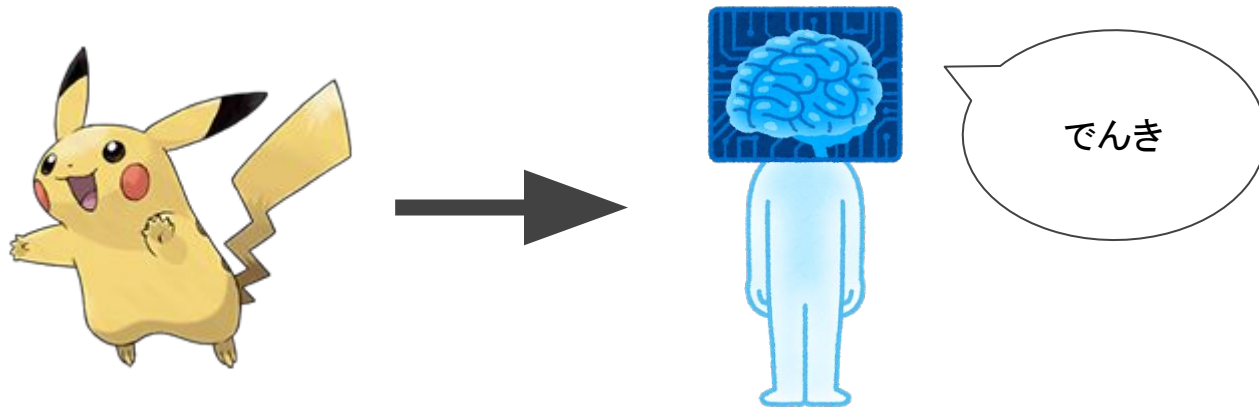
Agenda

- 目的・目標
- アプローチ
- 実施計画
- データ
- 学習方法
- 実行結果
- 実験結果
- 考察

目的・目標

ポケモンの画像から、そのポケモンが持つタイプ (18種)を判定したい

例:ピカチュウの画像を入力すると、でんきタイプだと判定される





アプローチ

- ポケモンのタイプ判別できそうな手法を考える
- classificationが良さそう
- CNNやろう！



実験計画

- 必要そうなデータの収集
- 学習用データセットの構築
- CNN等を用いて機械学習
- 様々な手法の検証
- 評価

データ(1)

- ポケモンの画像
- 形状: 215×215 RGBA
- 枚数: 801枚 (1ポケモン1枚)*第7世代まで



0001.png



0002.png



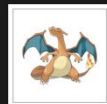
0003.png



0004.png



0005.png



0006.png



0007.png



0008.png



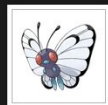
0009.png



0010.png



0011.png



0012.png



0013.png



0014.png



データ(2)

- ポケモンの図鑑番号や名前、タイプデータ、種族値など (第7世代まで)

1	Number	Name	Type1	Type2	Total	HP	Attack	Defense	SpecialAtk	SpecialDef	Speed
2	1	Bulbasaur	Grass	Poison	318	45	49	49	65	65	45
3	2	Ivysaur	Grass	Poison	405	60	62	63	80	80	60
4	3	Venusaur	Grass	Poison	525	80	82	83	100	100	80
5	3	VenusaurMega Venusaur	Grass	Poison	625	80	100	123	122	120	80
6	4	Charmander	Fire	NA	309	39	52	43	60	50	65
7	5	Charmeleon	Fire	NA	405	58	64	58	80	65	80
8	6	Charizard	Fire	Flying	534	78	84	78	109	85	100
9	6	CharizardMega Charizard X	Fire	Dragon	634	78	130	111	130	85	100
10	6	CharizardMega Charizard Y	Fire	Flying	634	78	104	78	159	115	100



データ前処理

- 今回使わないデータを省く
 - 種族値やメガ進化, 色違い etc.
- データの正規化
 - 0~255を0~1にする
- 次元削減
 - RGBAをRGBに変換
 - 画像サイズを215*125から64*64に変換



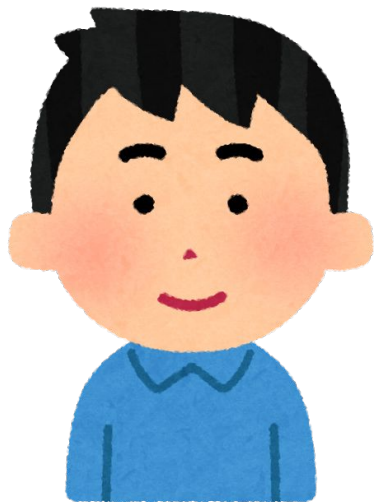
試した手法

- 多クラス分類 Multi class classification
- 多ラベル分類 Multi label classification
- クラスタリング
- ランダムフォレスト



Multi class classification

- 多クラス分類。1つの画像を複数クラスの中から1つに分類する。
- 問題点として、1つのクラスに分類するので、複数タイプに対応できない。



男性 90%



Multi class classification

- KerasのSequentialモデルを利用
 - 畳み込み層やプーリング層を重ねてモデルを構築
 - 活性化関数はrelu
 - 最適化関数はAdam
 - 出力層にsoftmax
 - 損失関数はcategorical_crossentropy
- データ数不足の対策として、sklearnのKFoldを使って交差検証



Multi class classification

実行結果:

部分点ありの場合: 約95%

完全解答の場合: 約1%



Multi class classification

部分点とは？

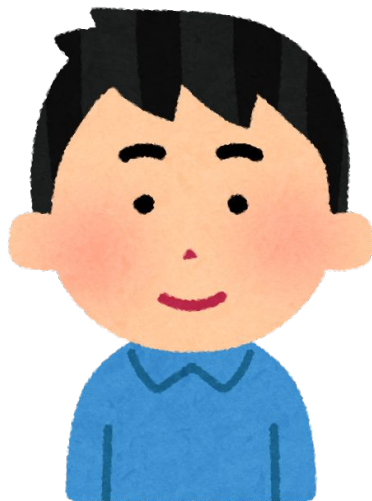
→ タイプはバイナリクラスで表現される。(e.g.: [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])

このタイプの精度を評価する際、一致する割合を精度とする (部分点を与える) 方法と完全一致で正解とみなす方法の2種類がある。



Multi label classification

- 多ラベル分類。1枚の画像から2つ以上のクラスを出力出来る。



男性 90%
ポロシャツ 70%
青色 98%



Multi class classification

- KerasのSequentialモデルを利用
 - 畳み込み層やプーリング層を重ねてモデルを構築
 - 活性化関数はrelu
 - 最適化関数はAdam
 - 出力層にsigmoid
 - 損失関数はbinary_crossentropy
- データ数の不足対策
 - ズラし, 反転等での水増し
 - 交差検証



Multi label classification

試した事

1. とりあえず参考サイトのコードを試す
 - a. 水タイプとノーマルタイプとしか出力されなかった
2. class weight
 - a. タイプはバラけたけど精度はほぼ変わらなかった
3. L2正則化
 - a. 過学習はしなくなったがlossが下がらなくなった



Multi label classification

実行結果:

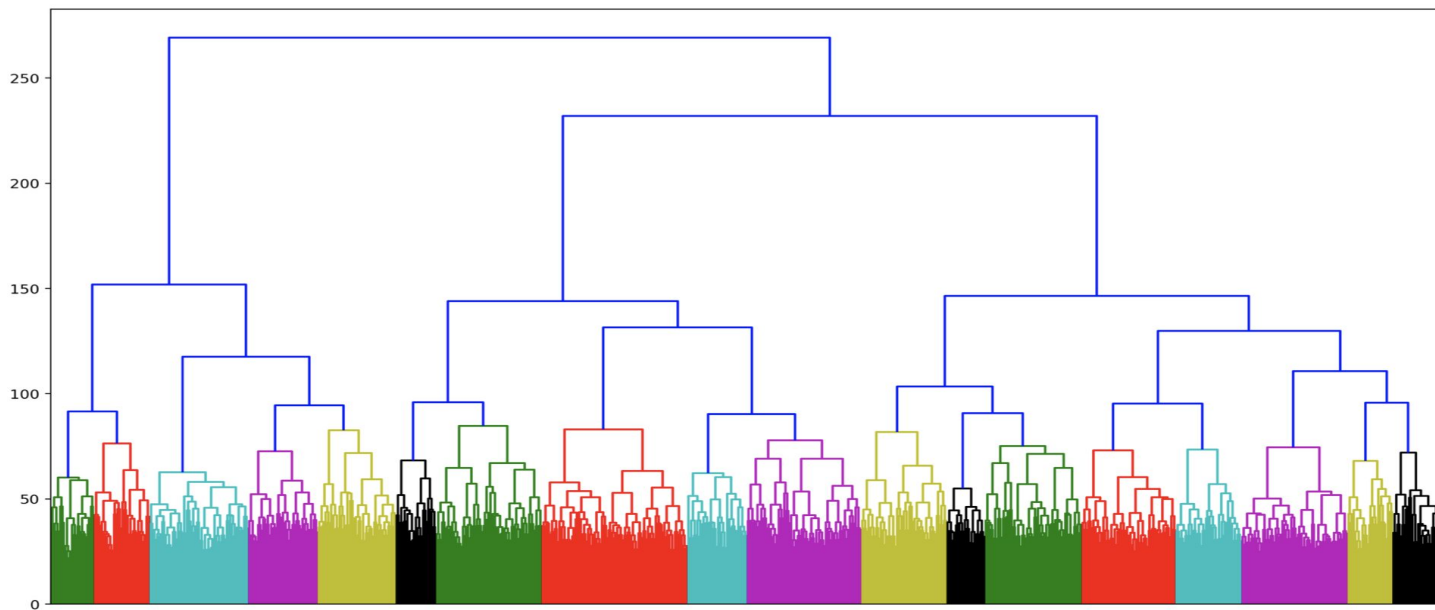
部分点ありの場合: 約90%





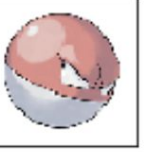



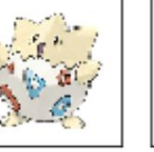


















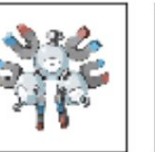




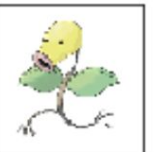














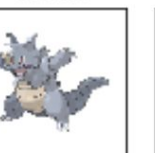


完全解答の場合: 約1%



クラスタリング

- 階層型クラスタリング
 - scipy の linkage を使ってクラスタリング



0	Poison 	Fairy 	Fairy 	Fire 	Electric 	Electric 	Normal 	Water 	Fairy 	Ground 
1	Water 	Water 	Poison 	Ground 	Ground 	Water 	Ghost 	Normal 	Normal 	Normal 
2	Grass 	Grass 	Ground 	Fire 	Grass 	Fire 	Rock 	Electric 	Water 	Water 
3	Normal 	Fighting 	Grass 	Grass 	Water 	Ground 	Water 	Water 	Electric 	Bug 
4	Fire 	Fire 	Normal 	Poison 	Ground 	Normal 	Water 	Ground 	Water 	Normal 



クラスタリングの考察

- あまりタイプごとに別れてくれない
- 進化系統が同じポケモンは同じクラスターに分類できているっぽい
- 色だけでなく、体形や姿勢が似ているものでクラスタリングされている



実行結果

- 多クラス分類
 - 2タイプもちのポケモンの時、一つのタイプしか予測できない為に、正解はType1だが予測はType2を出力してしまい、当たっているのに当たっていない状況になった.
- 多ラベル分類
 - 18タイプ中4タイプで確率20%ずつとなり2タイプに絞り込めなかった.
- クラスタリング
 - 色以外の要素が思ったより大きく、うまくタイプで分かれなかった



考察

精度が出せなかった要因

- ・データが少ない
- ・データに偏りがある (クラスごとのサンプル数が違う、e.g.:水タイプは多いがフェアリータイプは少ない)
- ・そもそもクラスタ数が多すぎる (18タイプあるのでかなりばらけたっぽい)



参考文献

image_data

<https://www.kaggle.com/dollarakshay/pokemon-images>

type_data

<https://github.com/lgreski/pokemonData>