・課題

sklearn.datasets.load_digits という 8×8=64 次元の白黒データが 1797 個入っているデータセットを用いて、手書き数字 (0~9 までの数字)の画像を分類する教師ありの機械学習を行い、その性能評価実験を行う。

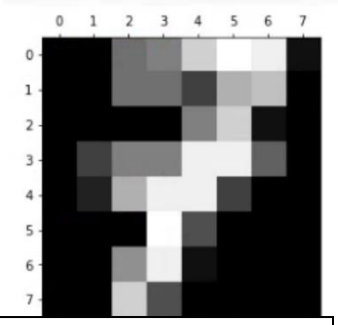


図 1　手書き数字 7 の画像データの例

・分析の概要

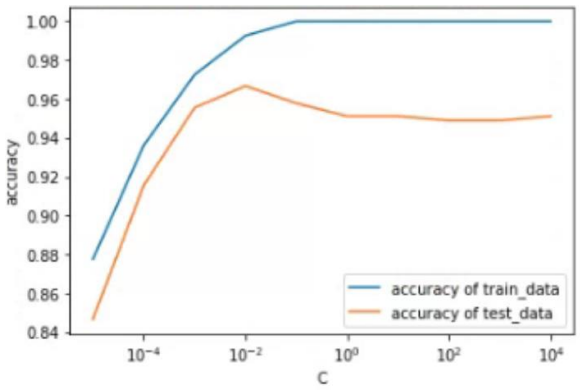
6 つの学習モデル（ロジスティック回帰、サポートベクトルマシン(SVM)、K 点最近傍法(K-NN)、決定木、ランダムフォレスト、ニューラルネットワーク(NNW)）を用いて性能評価を行う。なお、解析には NNW は Keras を用い、それ以外の学習モデルでは sklearn を用いている。まず、それぞれのモデルにおけるハイパーパラメータを変化させたときにテストエラーがどのように変化するか検証する。続いて、ハイパーパラメータを固定したうえで異なる学習モデル間の性能を比較したうえで学習データの量を減少に対してエラー率がどのくらい上がるか検証する。

(やり残したこと：非線形 SVM　＞　多項式カーネル・RBF カーネル)

ロジスティック回帰においてソフトマージンの厳しさを表す正則化強度の逆数である C を変化させてみた。図 2 より C＝0.01 の時にテストデータの正確度が最大となり、それ以上 C を増大させても過学習になる傾向が見られた。また、低すぎる C ではソフトマージンを厳しく定めすぎるために正解率が低下した。

SVM においてロジスティック回帰と同様のパラメータ C を変化させた。図 3 より C＝2.5^-3 の時に正確度が最大となり、それよりも大小の C ではロジスティック回帰と同様の理由で正解率が下がることが分かった。また、SVM の正解率の最大値はロジスティック回帰のそれと同程度である。

K-NN において K を変化させてみた。図 4 より K=1 の時に最大の正解率を取り、その後 K が増えるに従って正解率が単調減少することが分かった。ロジスティック回帰、SVM、K-NN の結果から、このデータの手書き数字は互いの画像が比較的分離していて曖昧な画像が比較的少ないと考えられる。



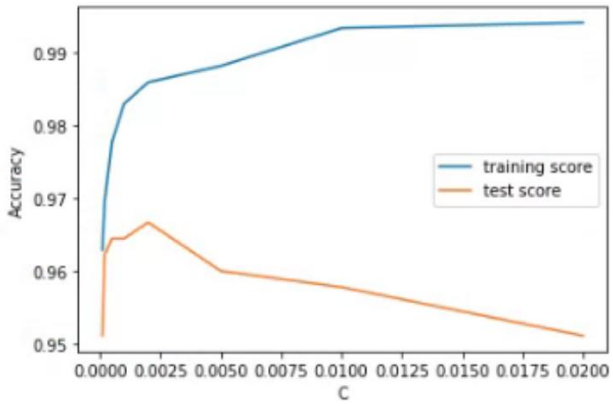


図２（左）　ロジスティック回帰のハイパーパラメータ（正則化強度の逆数 C）依存性

図３（右）　SVM のハイパーパラメータ C 依存性

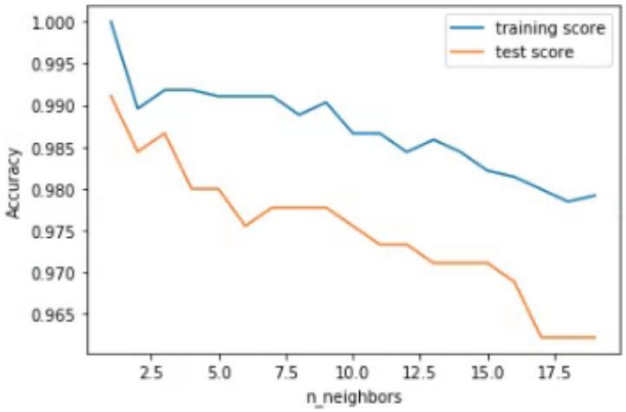
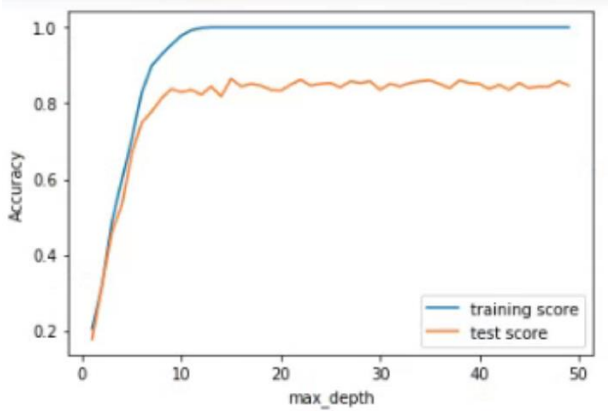


図４　K-NN のハイパーパラメータ K 依存性

決定木において最大の決定木の深さを変化させてみた。図５より決定木の最大深さが 8 になるまで急激に正解率が上昇するのに対して、それ以降は正解率が変化しなかった。このことから、このデータの手書き数字を区別するには 10 程度の次元があれば充分であると考えられる。

ランダムフォレストにおいても最大の深さを変化させてみた。図６より大まかな結果は決定木と同じであったが、ランダムフォレストでは決定木と比べて最大深さが１の時から一貫して正解率が高い傾向にあり、訓練データの正確度が飽和する最大深さ７以上においても微量の正解率の増加があることが分かった。



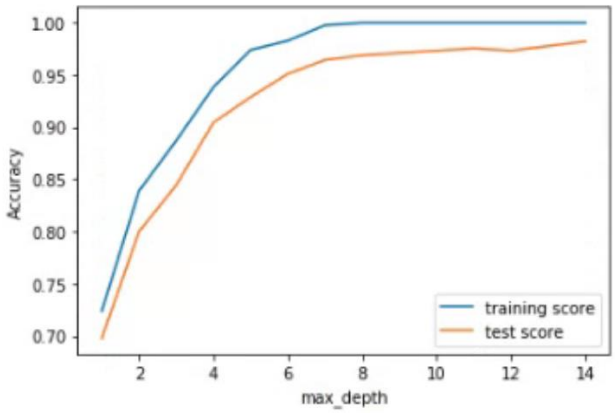


図５（左）　決定木のハイパーパラメータ（最大の決定木の深さ）依存性

図６（右）　ランダムフォレストのハイパーパラメータ依存性

三層で構成される古典的な NNW において、第一層、第二層のニューロン数(1,2)をそれぞれ変化させてみた。なお、NNW の活性化関数は Sigmoid 関数であり、出力に対しては Softmax 関数を用いている。発展させた逆誤差伝搬法である RMSprop を用いて、最適化関数 categorical_cross-entropy を 20 ほどに分けた train_data のミニバッチを 100step 学習させることにより最小化している。NNW を用いた機械学習はそれぞれのケースで分単位の時間がかかり、そのほかの教師あり学習の方法に比べて格段に計算時間が長かった。図７ではニューロン数を 4,8,16,32,64,128 と変化させており、そのうち正解率が 95%以上のケースを色分けしている。図７よりニューロン数が最大のケースで最高の正解率 0.968 を出している。また、 (1,2)=(64,32) のケースにおいて周囲に比べて高い正解率を出していることが分かる。このデータでは入力の次元が 64 で出力が 10 このデータであることから、比較的小さな計算資源で NNW を実装するためには入力の次元から出力の次元に向かって単調減少するように各層のニューロン数を設定すればよいことが分かる。

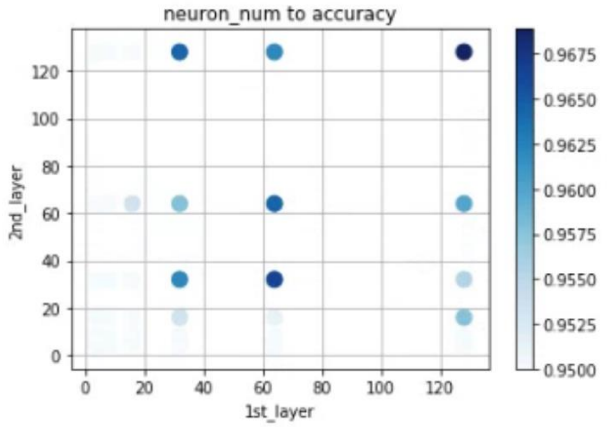


図７　NNW のハイパーパラメータ（第一層、第二層のニューロン数）依存性

図 2 ～図 7 により決定木を除いたすべての学習モデルで 95%以上の高い正解率が得られた。特に、K-NN の K=1 の場合すべての学習モデルの中で最も高い 99%の正解率が得られた。これは与えられた手書き数字のデータの画像が互いに比較的分離していて曖昧な画像が比較的少ないことに起因すると考えられる。

続いて、学習データを極端に少なくした時に(train_size=100)正解率の変化について調べた結果を表 1 に示す。十分な学習データを与えたときに正解率の高かったランダムフォレスト・NNW において、学習データを少なくしたときに他の学習モデルと比較して正解率が大きく低下した。このことから、ランダムフォレスト・NNW は他の学習モデルに比べてより多くの学習データが必要となると考えられる。

表 1　6 つの学習モデルにおける学習データを減少時の正解率の変化

	train_data_num=1347	train_data_num=100
LogisticRegression	0.962	0.886
LinearSVC	0.957	0.878
KNeighborsClassifier	0.986	0.873
DecisionTreeClassifier	0.844	0.597
RandomForestClassifier	0.971	0.868
Neural NetWork	0.966	0.693

・補足

画像データは 64 ピクセルのデータそのものが特徴量であるから、特徴量を変えることができなかった。また、画像認識においてピクセルはすでに規格化されているのでスケーリングはやりようがなかった。そして、分類を誤った事例の手書き数字は人間が見ても数字そのものが分かりにくい傾向にあった。

・参考文献

[1] 塚本邦尊, 山田典一, 大澤文孝　東京大学のデータサイエンティスト育成講座 ~Python で手を動かして学ぶデ―タ分析~

[2] 斎藤 康毅　ゼロから作る Deep Learning ―Python で学ぶディープラーニングの理論と実装

[3] その他インターネット上の記事多数