УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Отчёт по лабораторной работе №5

Дисциплина: "Система искусственного интеллекта"

Преподаватель: Королёва Юлия

Студент: Закиров Бобур

Группа: P33312

Санкт-Петербург

2021 г.

Лабораторная работа №5.

Задание

Цель: решить задачу много-классовой классификации, используя в качестве тренировочного набора данных - набор данных MNIST, содержащий образы рукописных цифр.

- Используйте метод главных компонент для набора данных MNIST (train dataset объема 60000). Определите, какое минимальное количество главных компонент необходимо использовать, чтобы доля объясненной дисперсии превышала 0.80. Построить график зависимости доли объясненной дисперсии от количества используемых ГК
- Введите количество верно классифицированных объектов класса *номер_в_списке % 9* для тестовых данных
- Введите вероятность отнесения 5 любых изображений из тестового набора к назначенному классу
- Определите Accuracy, Precision, Recall or F1 для обученной модели
- Сделайте вывод про обученную модель

Выполнение

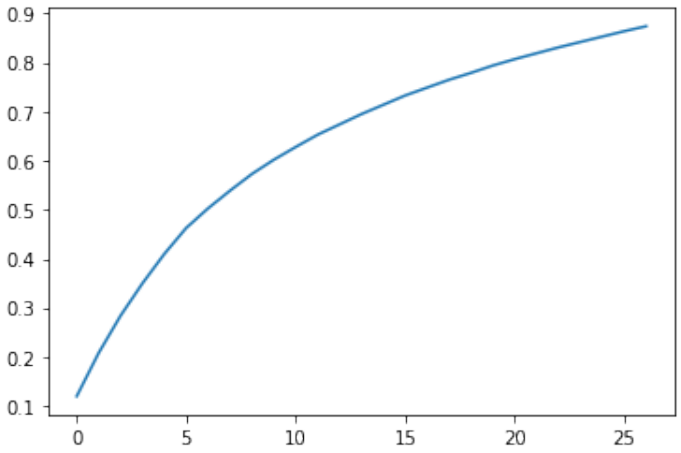
Код: https://github.com/insaniss/artificial-intelligence-system

Немного дополним код, чтобы было легче подбирать колиество компонент - нипишем функцию, которая будет выводить, достигла ли доля объясненной дисперсии необходимого значения:

```
from sklearn.decomposition import PCA

disp = 0.8 + (279307 % 10) / 100;

pca = PCA(n_components=27, svd_solver='full')
X_train = pca.fit(X_train).transform(X_train)

explained_variance = np.round(np.cumsum(pca.explained_variance_ratio_), 3)

plt.plot(np.arange(27), explained_variance, ls = '-')
```

Минимально необходимоеколичество компонент = 27.

```
array([0.12 , 0.208, 0.284, 0.35 , 0.41 , 0.463, 0.503, 0.539, 0.573,
       0.602, 0.628, 0.653, 0.674, 0.695, 0.714, 0.733, 0.749, 0.765,
       0.779, 0.794, 0.807, 0.819, 0.831, 0.842, 0.853, 0.864, 0.874])
```



Определим число объектов, отнесённых к нужному классу:

```
var_class = 279307 % 9
CM[var_class][var_class] # 1765
```

Выведем остальные параметры модели с помощью средств библиотеки `scikit-learn`:

```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred, target_names=target_names))
```

	precision	recall	f1-score	support
class 0	0.78	0.79	0.78	1693
class 1	0.94	0.85	0.89	2075
class 2	0.37	0.46	0.41	1763
class 3	0.69	0.81	0.74	1873
class 4	0.67	0.80	0.73	1756
class 5	0.39	0.38	0.39	1591
class 6	0.48	0.44	0.46	1766
class 7	0.75	0.79	0.77	1886
class 8	0.35	0.32	0.34	1773
class 9	0.64	0.39	0.49	1824
accuracy			0.61	18000
macro avg	0.61	0.60	0.60	18000
weighted avg	0.61	0.61	0.61	18000

Для оценки качества работы алгоритма на каждом из классов по отдельности введем метрики precision (точность) и recall (полнота).

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

Precision можно интерпретировать как долю объектов, названных классификатором положительными и при этом действительно являющимися положительными, а recall показывает, какую долю объектов положительного класса из всех объектов положительного класса нашел алгоритм. Оценка F1-score - это среднее гармоническое значение precision и recall.

`macro avg` – вычисление метрики для каждого класса и получение невзвешенного среднего
`weighted avg` – вычисление метрик для каждого класса и получение взвешенного среднего по числу выборок на каждый класс.

Вывод

В лабораторной работе я реализовал многоклассовую классификацию с помощью методы опорных векторов для набора данных, состоящих из рукописных цыфр, научился просматрывать параметры обученной модели – это базовые навыки, необходиые для развития в сфере ML.