Факультет программной инженерии и компьютерной техники

Системы искусственного интеллекта

Лабораторная работа №5

Преподаватель: Болдырева Елена Александровна

Выполнил: Кульбако Артемий Юрьевич P33112

Санкт-Петербург
2020

Задание

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

1. Используйте метод главных компонент для набора данных MNIST (train dataset объемом 60000). Определите, какое минимальное количество главных компонент необходимо использовать, чтобы доля объясненной дисперсии превышала 0.80 + номер_в_списке % 10. Построить график зависимости доли объясненной дисперсии от количества используемых ГК.

2. Введите количество верно классифицированных объектов класса номер_в_списке%9 для тестовых данных.

4. Определите Accuracy, Precision, Recall или F1 для обученной модели.

5. Сделайте вывод про обученную модель.

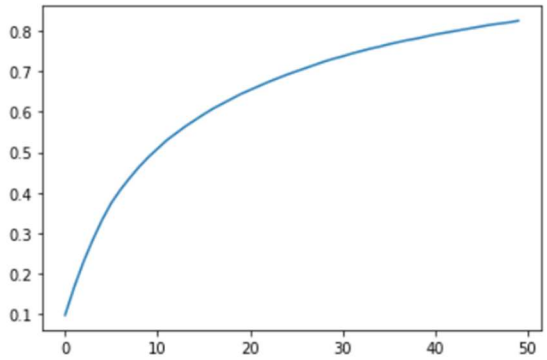
Выполнение

Немного дополним код, чтобы было легче подбирать количество компонент – напишем функцию, которая будет выводить, достигла ли доля объяснённой дисперсии необходимого значения:

```
from sklearn.decomposition import PCA
gold_disp = 0.8 + 265570 % 10 # необходимая долю объясненной дисперсии
pca = PCA(n_components=50, svd_solver='full') # перебираем n_components, пока один из элементов exp
lained_variance не станет больше gold_disp
modelPCA = pca.fit(X_train)
X_train = modelPCA.transform(X_train)
explained_variance = np.round(np.cumsum(pca.explained_variance_ratio_),3)
print(any(i >= gold_disp for i in explained_variance)) # функция, которая выведет True если Д.О.С.
достигла нужного значения
print(explained_variance)
plt.plot(np.arange(50), explained_variance, ls = '-')
```

Минимально необходимое количество компонент = 50.

```
True
[0.098 0.168 0.23  0.284 0.333 0.375 0.408 0.437 0.464 0.488 0.509 0.53
 0.547 0.564 0.579 0.594 0.608 0.62  0.632 0.644 0.654 0.664 0.674 0.683
 0.692 0.7   0.708 0.716 0.724 0.731 0.737 0.744 0.75  0.756 0.761 0.767
 0.772 0.777 0.781 0.786 0.791 0.795 0.799 0.803 0.807 0.811 0.815 0.818
 0.821 0.825]
[<matplotlib.lines.Line2D at 0x7fe184ab19e8>]
```



Определим число объектов, отнесённых к нужному классу:

```
var_class = 265570 % 9
CM[var_class][var_class]
```

Результат = 1362

Выведем остальные параметры модели с помощью средств библиотеки sklearn:

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

	precision	recall	f1-score	support
0	0.79	0.81	0.80	1693
1	0.91	0.86	0.89	2075
2	0.40	0.55	0.46	1763
3	0.65	0.80	0.72	1873
4	0.66	0.74	0.70	1756
5	0.40	0.37	0.39	1591
6	0.41	0.34	0.37	1766
7	0.75	0.72	0.74	1886
8	0.35	0.31	0.33	1773
9	0.64	0.49	0.55	1824
accuracy			0.61	18000
macro avg	0.60	0.60	0.59	18000
weighted avg	0.61	0.61	0.60	18000

Вывод

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