Федеральное государственное автономное образовательное учреждение высшего образования

Университет ИТМО

Дисциплина: Системы искусственного интеллекта

Лабораторная работа 5

Выполнил студент:
Кривоносов Егор Дмитриевич

Группа: P33111

Преподаватель:
Полещук Елизавета Александровна

Санкт-Петербург

2021 г.

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 объема 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. Выведите количество верно классифицированных объектов класса номер_в_списке % 9 для тестовых данных
3. Введите вероятность отнесения 5 любых изображений из тестового набора к назначенному классу
4. Определите Accuracy, Precision, Recall и F1 для обученной модели
5. Сделайте вывод про обученную модель

Выполнение

Задание 1

Код

```
!pip install --upgrade pip
!pip install --upgrade scikit-learn==0.23.0
```

```
exp_disp = 0.8 + 14 % 10 / 100
classa = 14 % 9
```

```
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.multiclass import OneVsRestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.decomposition import PCA
from keras.datasets import mnist

(X_train, y_train), (X_pred, y_pred) = mnist.load_data()

dim = 784 # 28*28
X_train_ = X_train.reshape(len(X_train), dim)

pca = PCA(svd_solver='full')
pca = pca.fit(X_train_)

explained_variance = np.round(np.cumsum(pca.explained_variance_ratio_),3)
plt.plot(np.arange(dim), explained_variance, ls = '-')
```

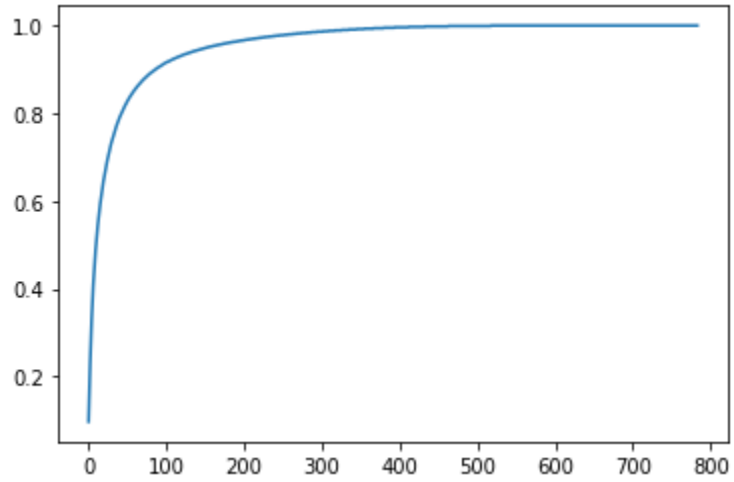
```
M = 0
for arg, val in enumerate(np.cumsum(pca.explained_variance_ratio_)):
  if val > exp_disp:
    M = arg + 1
    break

print("Количество главных компонент, чтобы доля объяснённой дисперсии превышала " + str(exp_disp) + ": " + str(M))
```

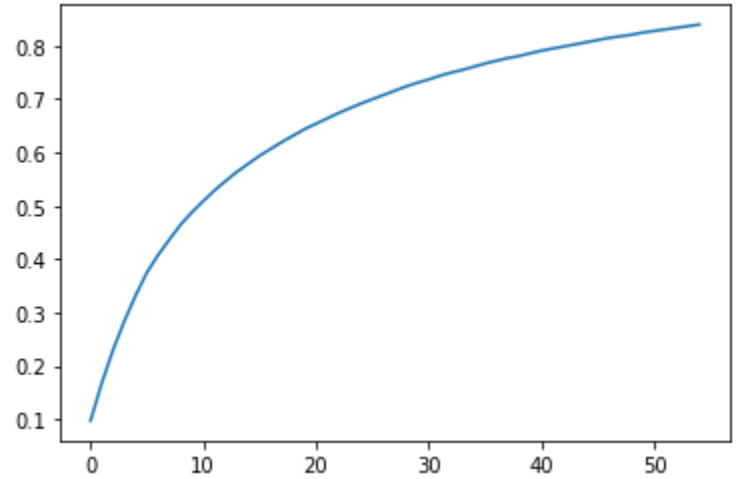
```
X_train = X_train.reshape(len(X_train), dim)
pca = PCA(n_components=M, svd_solver='full')
pca = pca.fit(X_train)

explained_variance = np.round(np.cumsum(pca.explained_variance_ratio_),3)
plt.plot(np.arange(M), explained_variance, ls = '-')
```

Вывод кода



Зависимость доли объясненной дисперсии от от всех ГК



Зависимость доли объясненной дисперсии от от всех количества используемых ГК

```
Количество главных компонент, чтобы доля объяснённой дисперсии превышала 0.8400000000000001: 56
```

Задание 2

Код

```
X_train, X_test, y_train, y_test = train_test_split(X_train, y_train, test_size=0.3, random_state=2020)
X_train = pca.transform(X_train)
X_test = pca.transform(X_test)

modelPCA = pca.fit(X_test)
X_test = modelPCA.transform(X_test)

tree = RandomForestClassifier(criterion='gini', min_samples_leaf=10, max_depth=20, n_estimators=10, random_state=2020)
clf = OneVsRestClassifier(tree).fit(X_train, y_train)
y_pred = clf.predict(X_test)

CM = confusion_matrix(y_test, y_pred)
print("Количество верно классифицированных объектов класса " + str(classa) + ": " +
          str(CM[classa][classa]))
```

Вывод кода

```
Количество верно классифицированных объектов класса 5: 458
```

Задание 3

Код

```
imgs = [1337, 555, 777, 322, 17]
for img in imgs:
  print(f"Вероятность отнесение изображения №{img} к назначеному классу {y_pred[img]} = {clf.predict_proba(X_test)[img][y_pred[img]]}")
```

Вывод кода

```
Вероятность отнесение изображения №1337 к назначеному классу 4 = 0.27418700127161516
Вероятность отнесение изображения №555 к назначеному классу 5 = 0.26616674947382496
Вероятность отнесение изображения №777 к назначеному классу 8 = 0.4258460925121311
Вероятность отнесение изображения №322 к назначеному классу 5 = 0.26864957771673564
Вероятность отнесение изображения №17 к назначеному классу 6 = 0.24596904856790294
```

Задание 4

Код

```
from sklearn.metrics import classification_report, accuracy_score
target_names = ['class 0', 'class 1','class 2','class 3','class 4','class 5','class 6','class 7','class 8','class 9']
print("Accuracy:", accuracy_score(y_test, y_pred))

print(classification_report(y_test, y_pred, target_names=target_names))
```

Вывод кода

```
Accuracy: 0.5719444444444445
              precision    recall  f1-score   support

     class 0       0.76      0.85      0.80      1693
     class 1       0.91      0.79      0.85      2075
     class 2       0.33      0.45      0.38      1763
     class 3       0.60      0.71      0.65      1873
     class 4       0.60      0.73      0.65      1756
     class 5       0.40      0.29      0.33      1591
     class 6       0.33      0.23      0.28      1766
     class 7       0.66      0.72      0.69      1886
     class 8       0.44      0.43      0.44      1773
     class 9       0.59      0.45      0.51      1824

    accuracy                           0.57     18000
   macro avg       0.56      0.57      0.56     18000
weighted avg       0.57      0.57      0.57     18000
```

Вывод

В результате выполнения лабораторной работы была обучена модель для предсказания нарисованных цифр на наборе MNIST с формированием 56 главных компонент из 784 имеющихся для получения на тестовой выборке доли объяснённой дисперсии 0.84. Полученная модель имеет точность 0.57 и хорошо справляется с определением цифр 0, 1, 3, 4 и 7 относительно остальных цифр, для которых значения более информативных мерок Precision, Recall и F1 значительно меньше и составляет меньше 0.5.