Министерство науки и высшего образования РФ

Национальный исследовательский университет ИТМО

Факультет Программной инженерии и компьютерных технологий

По дисциплине:
Системы искусственного интеллекта

Лабораторная работа № 5.

«Решение задачи многоклассовой классификации - набор данных MNIST»

Выполнил: До Зыонг Мань
Группа: P33201

Санкт–Петербург
2022 год.

I. Описание задания

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.
3. Введите вероятность отнесения 5 любых изображений из тестового набора к назначенному классу.
4. Определите Accuracy, Precision, Recall или F1 для обученной модели.
5. Сделайте вывод про обученную модель.

Вариант: Номер в списке 5.

Код можно посмотреть здесь

II. Выполнение

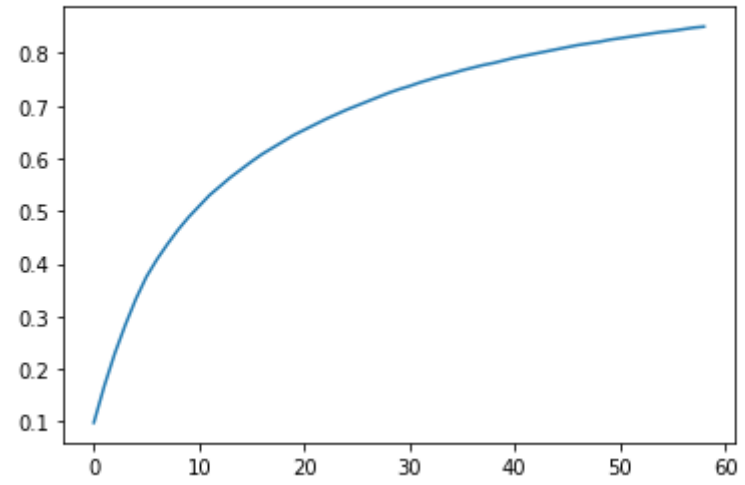
1. Используйте метод главных компонент для набора данных MNIST

Минимальное количество главных компонент необходимо использовать, чтобы доля объясненной дисперсии превышала 0.85.

```
variant_expectation = 0.8 + var_number % 10 /100
X_train = X_train.reshape(len(X_train), dim)

pca = PCA(n_components=variant_expectation, svd_solver='full')

modelPCA = pca.fit(X_train)

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

count = explained_variance.size

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


print("Explained Variance: " + str(round(variant_expectation, 2)) + " Number of Components: " + str(count))
```

График дисперсии:

```
Explained Variance: 0.85 Number of Components: 59
```



2. Введите количество верно классифицированных объектов

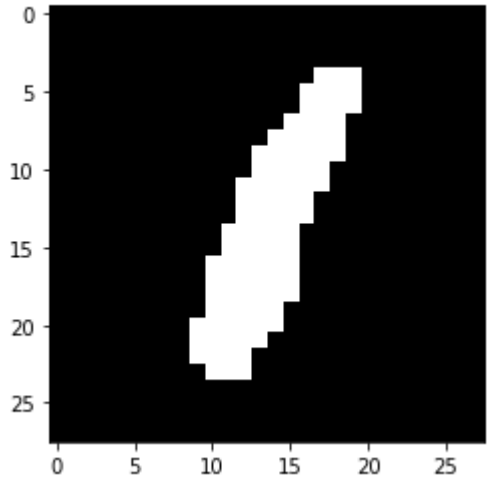
```
X_train, X_test, y_train, y_test = train_test_split(X_train, y_train, test_size=0.3, random_state=2020)

X_train = pca.transform(X_train)
X_test = pca.transform(X_test)

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)

modelPCA = PCA(n_components=count, svd_solver='full').fit(X_test)
X_test = modelPCA.transform(X_test)

class_variant = var_number % 9

y_pred = clf.predict(X_test)

CM = confusion_matrix(y_test, y_pred)

print("The number of correctly classified images contained in Class " + str(class_variant) + " is: " + str(
    CM[class_variant][class_variant]))
```

```
The number of correctly classified images contained in Class 5 is: 647
```

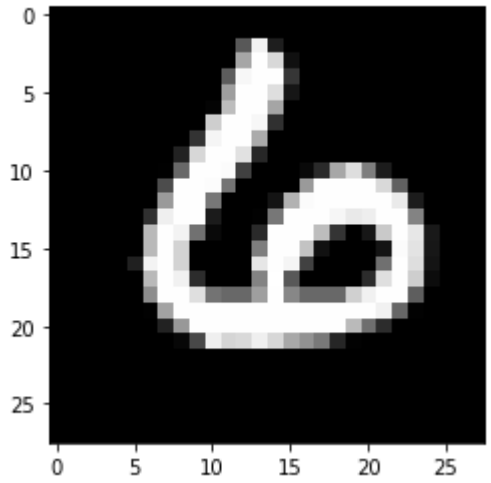
3. Введите вероятность отнесения 5 любых изображений

```
num = random.randint(0, 10000)
result = (clf.predict_proba(X_test)[num])[y_pred[num]]
plt.imshow(X_test_show[num], cmap='gray')
print("The probability that picture No." + str(num) + " belongs to Class " +
      str(y_pred[num]) + " is: " + str(round(result, 3)))
```

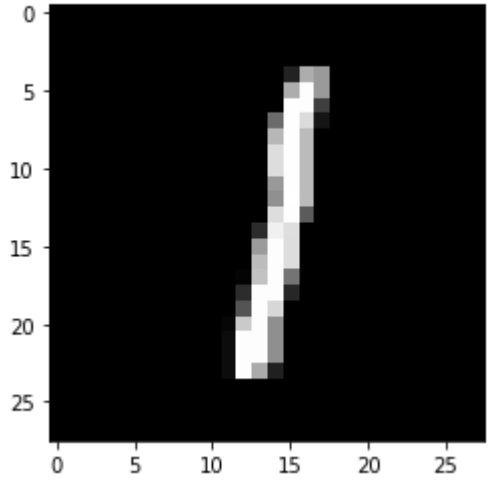
```
The probability that picture No.6185 belongs to Class 6 is: 0.391
```



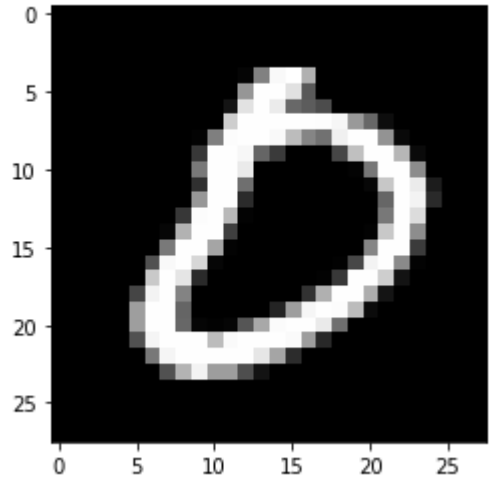
```
The probability that picture No.2001 belongs to Class 0 is: 0.264
```



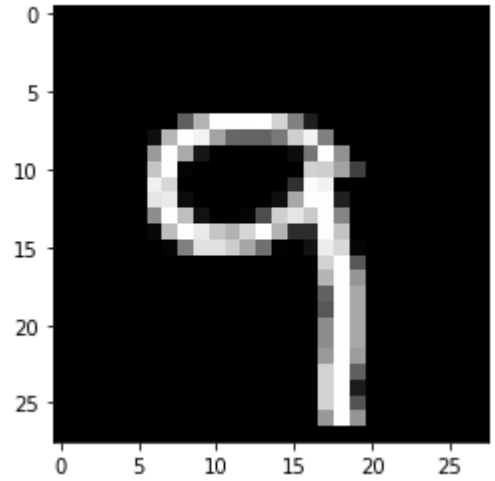
```
The probability that picture No.8587 belongs to Class 1 is: 0.952
```



```
The probability that picture No.7861 belongs to Class 6 is: 0.477
```



```
The probability that picture No.7708 belongs to Class 6 is: 0.51
```



4. **Определите Accuracy, Precision, Recall или F1**

a. **Accuracy** - доля правильных ответов

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{P + N} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

Тем не менее, у этой метрики есть одна особенность, которую необходимо учитывать. Она присваивает всем документам одинаковый вес, что может быть не корректно в случае, если распределение документов в обучающей выборке сильно смещено в сторону какого-то одного или нескольких классов.

b. **Precision** – точность

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

Точность показывает, какая доля объектов, выделенных классификатором как положительные, действительно является положительными.

c. **Recall** – полнота

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

Полнота показывает, какая часть положительных объектов была выделена классификатором.

d. F1 Score

$$F_1 = \frac{2}{\text{recall}^{-1} + \text{precision}^{-1}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{\text{tp}}{\text{tp} + \frac{1}{2}(\text{fp} + \text{fn})}$$

Существует несколько способов получить один критерий качества на основе точности и полноты. Один из них — F-мера, гармоническое среднее точности и полноты

```
print("Accuracy: " + str(accuracy_score(y_test, y_pred)))
print()
print(classification_report(y_test, y_pred, target_names=targets))
```

```
Accuracy: 0.801031746031746

              precision    recall  f1-score   support

     Class 0       0.89      0.88      0.89      1293
     Class 1       0.93      0.96      0.94      1416
     Class 2       0.81      0.80      0.80      1262
     Class 3       0.71      0.76      0.73      1290
     Class 4       0.72      0.83      0.77      1214
     Class 5       0.67      0.56      0.61      1158
     Class 6       0.86      0.89      0.88      1204
     Class 7       0.87      0.88      0.88      1318
     Class 8       0.76      0.79      0.77      1188
     Class 9       0.75      0.62      0.68      1257

    accuracy                           0.80     12600
   macro avg       0.80      0.80      0.79     12600
weighted avg       0.80      0.80      0.80     12600
```

III. Вывод

Я использовал анализ основных компонентов, установил долю объясненной дисперсии выше 85% и обучил модель с основными компонентами 59. Общая точность этой модели составляет 80%, и модель распознает цифры 0, 1 точно высокая степень.