|  |  |
| --- | --- |
|  | import numpy  from keras.datasets import mnist |
|  | from keras.models import Sequential |
|  | from keras.layers import Dense |
|  | from keras.utils import np\_utils |
|  |  |
|  | # Загружаем данные |
|  | (X\_train, y\_train), (X\_test, y\_test) = mnist.load\_data() |
|  |  |
|  | # Преобразование размерности изображений |
|  | X\_train = X\_train.reshape(60000, 784) |
|  | X\_test = X\_test.reshape(10000, 784) |
|  | # Нормализация данных |
|  | X\_train = X\_train.astype('float32') / 255 |
|  | X\_test = X\_test.astype('float32') / 255 |
|  |  |
|  | # Преобразуем метки в категории |
|  | Y\_train = np\_utils.to\_categorical(y\_train, 10) |
|  | Y\_test = np\_utils.to\_categorical(y\_test, 10) |
|  |  |
|  | # Создаем последовательную модель |
|  | model = Sequential() |
|  | model.add(Dense(800, input\_dim=784, activation="relu", kernel\_initializer="normal")) |
|  | model.add(Dense(10, activation="softmax", kernel\_initializer="normal")) |
|  | model.compile(loss="categorical\_crossentropy", optimizer="adam", metrics=["accuracy"]) |
|  |  |
|  | # Обучаем сеть |
|  | history = model.fit(X\_train, Y\_train, batch\_size=64, epochs=25, validation\_data=(X\_test, Y\_test)) |
|  |  |
|  | # Оцениваем качество обучения сети на тестовых данных |
|  | scores = model.evaluate(X\_test, Y\_test) |
|  | print("Точность работы на тестовых данных: %.2f%%" % (scores[1]\*100)) |

import matplotlib.pyplot as plt

%matplotlib inline

plt.plot(history.history['acc'], label='Аккуратность на обучающем наборе')

plt.plot(history.history['val\_acc'], label='Аккуратность на проверочном наборе')

plt.xlabel('Эпоха обучения')

plt.ylabel('Аккуратность')

plt.legend()

plt.show()



x = X\_test[488]

x = np.expand\_dims(x, axis=0)

y = model.predict(x)

prediction = np.argmax(y)

print(prediction)

print(y\_test[488])