

## Домашнее задание №9: «Каракули и нейросети»

Дедлайн 1 (20 баллов): 14 мая, 23:59

Дедлайн 2 (10 баллов): 21 мая, 23:59

Домашнее задание нужно написать на Python и сдать в виде одного файла. Правило именования файла: name\_surname\_9.py. Например, если вас зовут Иван Петров, то имя файла должно быть: ivan\_petrov\_9.py.

---



В этом домашнем задании мы продолжим тему распознавания образов. Для тестирования нашего алгоритма будем использовать датасет MNIST<sup>1</sup>.

MNIST (Mixed National Institute of Standards and Technology database) является основной базой при тестировании систем распознавания образов, а также широко используемой для обучения и тестирования алгоритмов машинного обучения. Она была создана перегруппировкой образов из оригинальной базы NIST, которая являлась достаточно сложной для распознавания. Кроме этого, были выполнены определенные преобразования (образы были нормализованы и сглажены для получения градаций серого цвета).

1. Реализуйте обучение нейронной сети с помощью метода обратного распространения ошибки (Back Propagation).

В качестве функции активации рекомендуется использовать сигмоиду.

---

<sup>1</sup>[http://en.wikipedia.org/wiki/MNIST\\_database](http://en.wikipedia.org/wiki/MNIST_database)

Структура класса приведена ниже:

```
class NeuralNetwork:
    def __init__(self, layers):
        self.num_layers = len(layers)
        self.layers = layers
        ...

    def train(self, X, y, n_iter=100, learning_rate=1):
        ...

    def feedforward(self, X):
        ...

    def backpropagation(self, X, y):
        ...

    def predict(self, X):
        ...
```

При реализации метода `train` может быть полезно обратиться к своей реализации метода стохастического градиента из предыдущих домашних заданий.

2. Дополните реализацию методом `predict`, который прогоняет все объекты из переданной матрицы `X` через полученную нейросеть. Предполагается, что метод возвращает индекс нейрона, на котором значение на выходном слое максимально.

3. Так как на этот раз мы используем стандартный датасет, в `scikit-learn` есть функция, позволяющая его легко получить:

```
from sklearn import datasets
from sklearn.cross_validation import train_test_split
dataset = datasets.fetch_mldata("MNIST Original")
trainX, testX, trainY, testY = train_test_split(
    dataset.data / 255.0, dataset.target.astype("int0"), test_size = 0.3)
```

4. Для визуализации датасета можно воспользоваться функцией `visualize_mnist` из файла по ссылке.<sup>2</sup>

Картинка должна выглядеть подобным образом:



5. Пример использования полученной сети:

```
nn = NeuralNetwork([train_X.shape[1], 10])
nn.train(trainX, trainY)
nn.predict(testY)
```

<sup>2</sup><https://gist.github.com/ktisha/95fcee0ed79236c7e6e5>

Обратите внимание, что на первом слое нам нужно число нейронов, равное количеству признаков (в данном случае, количеству пикселей), а на выходном слое количество нейронов, равное количеству классов объектов (в нашем случае это цифры 0-9).

6. Оценивать качество классификации в этот раз мы будем простым подсчетом отношения правильно классифицированных объектов к общему количеству объектов в выборке.

7. Ответьте на вопрос: как меняется качество классификации при изменении количества слоев сети и количества нейронов на каждом слое?