

# Нейронные сети, классификация 1 слой

С.И.Хашин

<http://math.ivanovo.ac.ru/dalgebra/Khashin/index.html>

Ивановский государственный университет

Иваново-2019

# План

Datasets

Спирали

sky\_data

MNIST

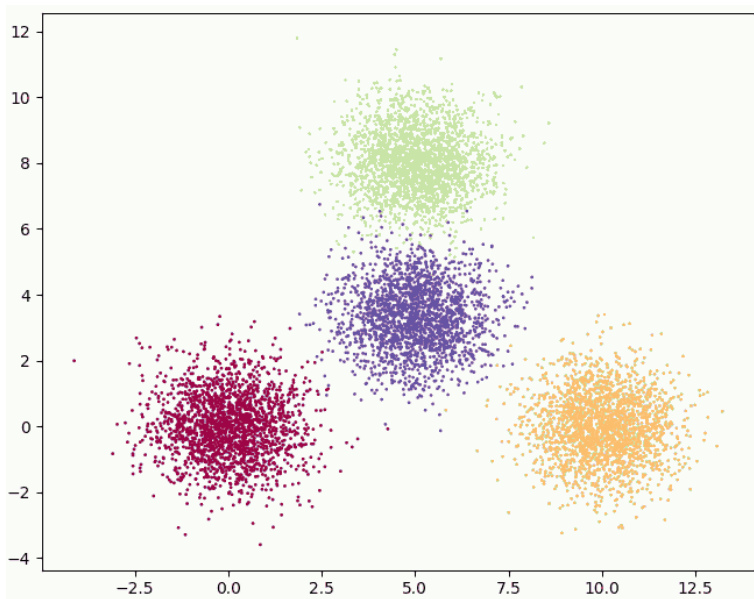
CIFAR

## Данные, 4 круга

4circles.csv

```
4 circles of radius 1.0 with 2000 points each,  
centers=((0, 0), (10, 0), (5, 8), (5, 3.3))  
0, -0.62180, -0.36799  
1, 11.97536, -0.60039  
2, 5.92334, 8.10006  
0, 0.35097, -1.95299  
2, 4.61210, 7.15570  
3, 4.77422, 2.51536  
0, 0.32185, 1.04246  
3, 4.66022, 3.60496  
3, 4.63681, 4.52674  
1, 10.65733, 0.99395  
2, 5.15129, 9.93802  
3, 5.07423, 0.87455
```

# Данные, 4 круга



## Данные, спираль-3

spiral\_02\_03.csv

Y, x[0], x[1], 3 classes of 2000 items each

2, -1.696242, -1.083956

1, -3.056333, -2.318821

0, 0.216430, -1.907722

1, -3.571905, -0.052195

2, -0.239038, -3.094834

2, -1.922705, 0.031378

0, 2.692563, -0.327081

2, 3.838949, -0.802651

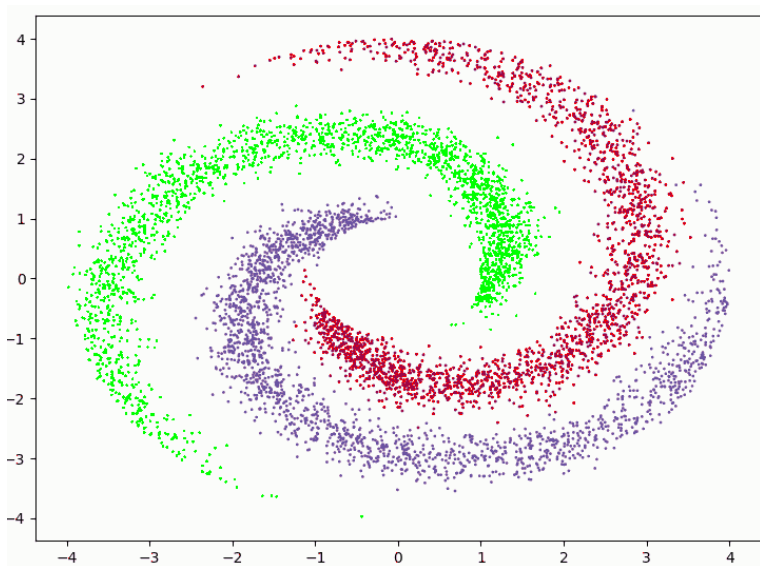
1, 1.075468, -0.074407

2, -1.511187, -0.404096

0, 2.294769, -0.970996

2, 2.117907, -2.754682

# Данные, спираль-3

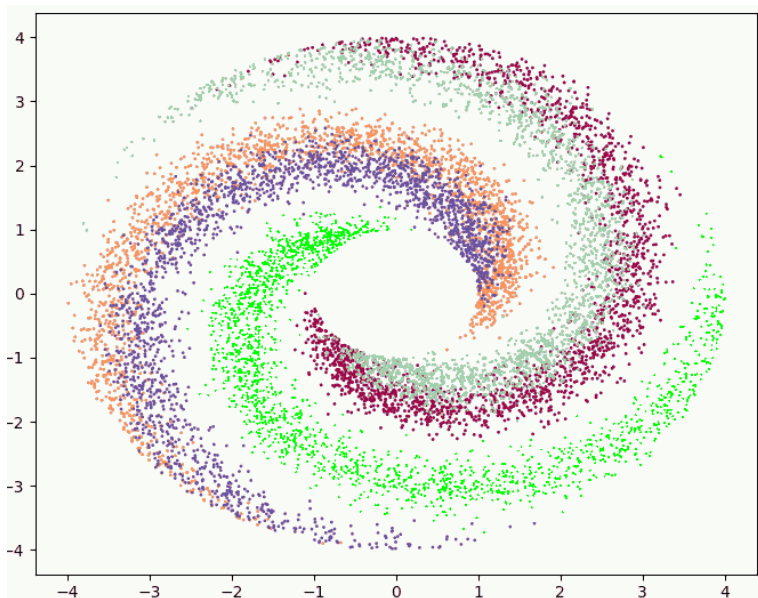


## Данные, спираль-5

spiral\_02\_05.csv

```
Y, x[0], x[1], 5 classes of 2000 items each
0, 1.357024, -1.287250
2, -1.849273, -0.397186
4, -0.659564, 2.116730
1, -2.842324, 1.848786
1, 0.342637, 1.627659
0, 2.897814, 0.111272
1, 1.146589, 0.089472
4, -3.326052, -0.347655
0, 3.028562, 0.052353
4, -0.972957, -3.558738
```

# Данные, спираль-5





## Данные, sky\_data

sky\_data.csv

10000 записей о звездах (STAR), галактиках (GALAXY) и квазарах (QSO)

Тип объекта — в столбце 'class'.

Распознавание надо вести по столбцам

['ra', 'dec', 'u', 'g', 'r', 'i', 'z', 'run', 'camcol', 'field'].

Данные в файле имеют такой вид:

```
objid,ra,dec,u,g,r,i,z,run,rerun,camcol,field,specobjid,...  
1.23765E+18,183.5313257,0.08969303,19.47406,17.0424,...  
1.23765E+18,183.5983705,0.135285032,18.6628,17.21449,...  
1.23765E+18,183.6802074,0.126185092,19.38298,18.19169,...  
1.23765E+18,183.8705294,0.049910685,17.76536,16.60272,...  
...
```

## Данные, MNIST, рукописные цифры

```
data = np.load("mnist_std.npz")
Y, X = data['Y'], data['X']
print('Y:', Y.shape)
print('X:', X.shape)
```

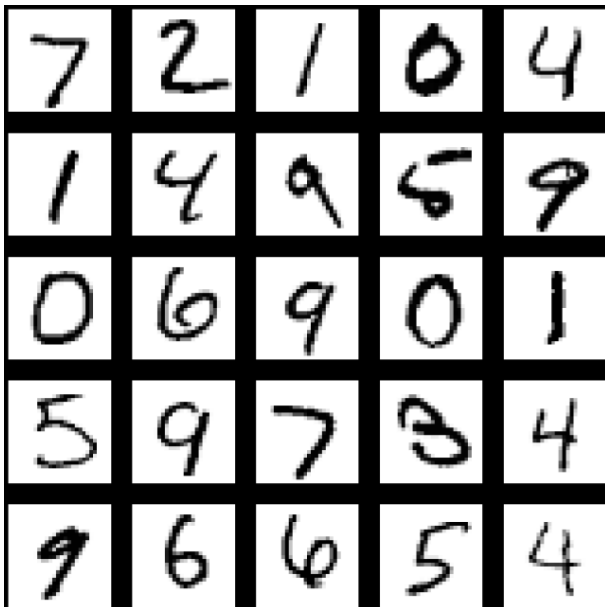
```
> Y: (70000,)
> X: (70000, 784)
```

**Замечание 1.** Данные в массивах Y, X типа np.uint8, то есть однобайтовые беззнаковые. Следовательно, из нельзя умножить, например, на 0.1.

**Замечание 2.** Данные по каждой цифре развернуты в строку. Для получения матрицы 3-мерного массива  $70000 \times 28 \times 28$ :

```
X = X.reshape((-1, 28, 28))
```

## Данные, MNIST, рукописные цифры



## Данные, MNIST, рукописные цифры

Для борьбы с переобучением данные разбиваются на 2 части: train (тренировочные) и test (проверочные):

```
X_train = X[:60000] # первые 60К изображений  
Y_train = Y[:60000]
```

```
X_test  = X[60000:] # остальные 10К изображений  
Y_test  = Y[60000:]
```

## Сохранить монохромное png

```
from PIL import Image

def save_png(a, fname):
    b = a.copy()
    b[b<0]=0
    b[b>255]=255
    b = b.round().astype(np.uint8)
    imgOut = Image.fromarray(b, 'L')
    imgOut.save(fname)

for i in range(10):
    save_png(X[i].reshape((28,28)), f'MNIST_{i:04d}.png')
```

## Данные, CIFAR-10

```
data = np.load("CIFAR10_std.npz")
Y, X = data['Y'], data['X']
print('Y:', Y.shape)
print('X:', X.shape)
```

```
> Y: (60000,)
> X: (60000, 3072)
```

**Замечание 1.** Данные в массивах Y, X типа np.uint8, то есть однобайтовые беззнаковые. Следовательно, из нельзя умножить, например, на 0.1.

**Замечание 2.** Данные по каждой цифре развернуты в строку. Для получения матрицы 4-мерного массива  $60000 \times 3 \times 32 \times 32$ :

```
X = X.reshape((-1, 3, 32, 32))
```

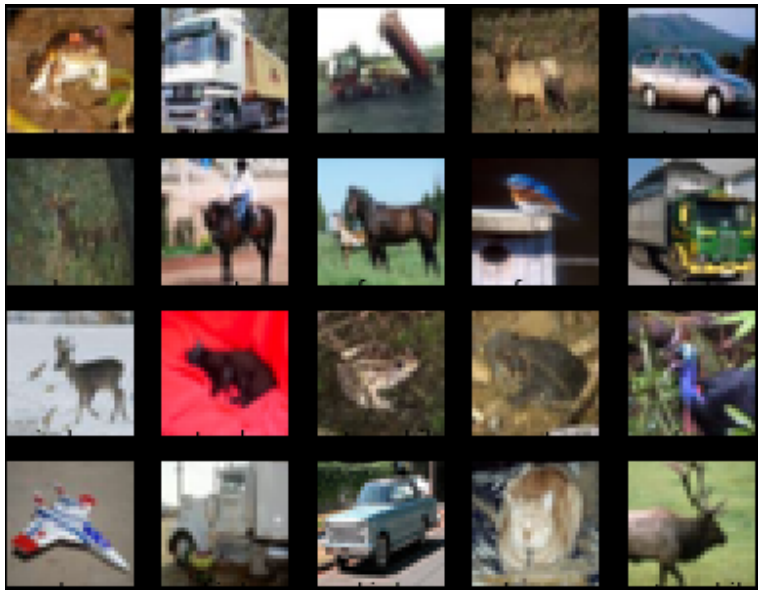
## Данные, CIFAR-10

Все картинки разбиты на 10 типов:

<i>Y</i>	<i>Description</i>
0	<i>airplane</i>
1	<i>automobile</i>
2	<i>bird</i>
3	<i>cat</i>
4	<i>deer</i>
5	<i>dog</i>
6	<i>frog</i>
7	<i>horse</i>
8	<i>ship</i>
9	<i>truck</i>

Метки — в массиве *Y*.

# Данные, CIFAR-10





## Данные, CIFAR-10

Для борьбы с переобучением данные разбиваются на 2 части:  
train (тренировочные) и test (проверочные):

```
X_train = X[:50000] # первые 50К изображений  
Y_train = Y[:50000]
```

```
X_test  = X[50000:] # остальные 10К изображений  
Y_test  = Y[50000:]
```

## Сохранить RGB-png

```
from PIL import Image

def save_png3(a, fname): # сохранить a в виде png
    a = a.astype(np.uint8)
    if a.shape[0]==3:
        b = np.zeros((a.shape[1],a.shape[2],3),dtype=np.uint8)
        for i in range(3):
            b[:, :, i] = a[i]
    else: b=a
    imgOut = Image.fromarray(b, 'RGB')
    imgOut.save(fname)

for i in range(10):
    save_png3(X[i].reshape((3,32,32)), f'CIFAR_{i:04d}.png')
```