МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ «МИСиС»

Институт ИТАСУ
Группа: **МПИ-20-4-2**

ОТЧЕТ

по лабораторной работе №2
по курсу «Нейронные сети»

Выполнил: Хабибулин М.И.
группа МПИ-20-4-2
Проверил: Курочкин И.И.

Москва 2020

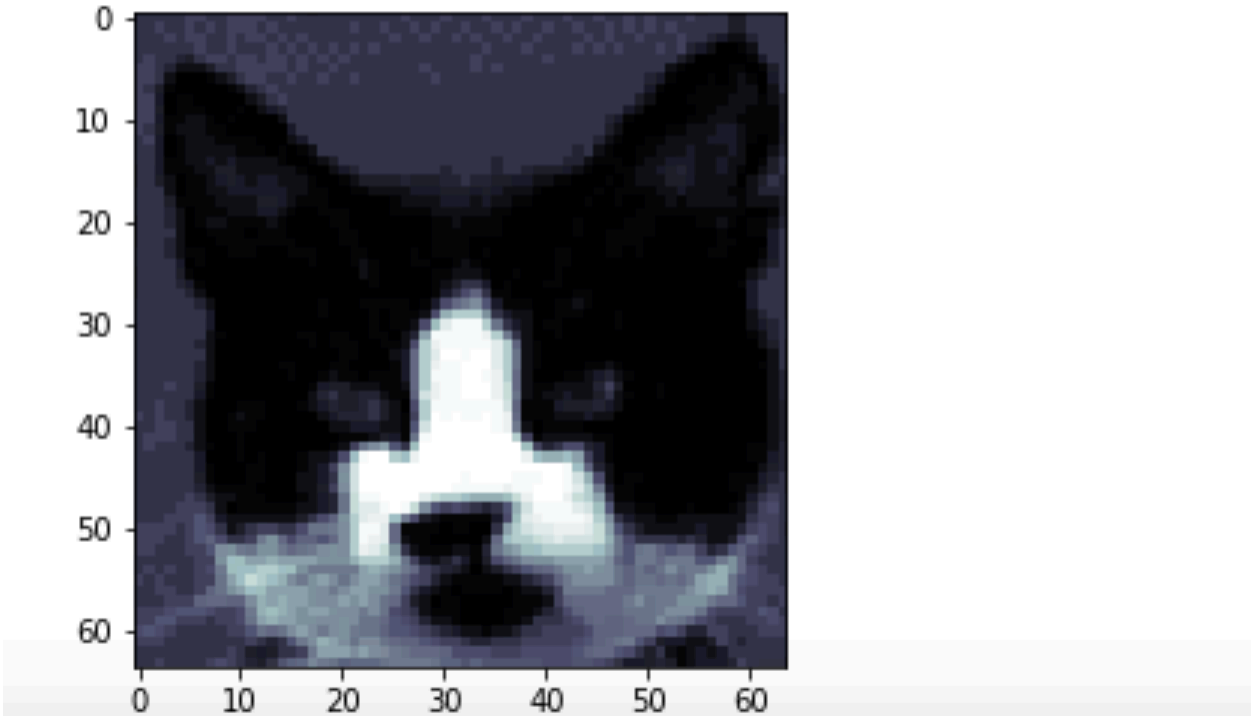
Инструментарий:
Язык программирования python 3.7

Библиотеки: matplotlib, numpy, random, pandas.

В качестве датасетов будем использовать следующие: 1)набор данных о кошках 2) набор данных о собаках

```
fig, ax = plt.subplots()
ax.imshow(cats[0,:].reshape(64, 64).T, cmap=plt.
```

```
<matplotlib.image.AxesImage at 0x12c5d9190>
```



1. Предобработка данных

Используем функцию np.concatenate, чтобы объединить массивы кошек и собак. Используем axis = 1, чтобы правильно их объединить. Поставим собак на первое место.

```
conc = np.concatenate((dogs,cats,),axis=0)
```

Создадим функцию стандартизации. Эта функция найдет среднее значение и стандартное отклонение для каждого изображения. Мы вычитаем среднее значение из каждого пикселя и делим каждый пиксель на стандартное отклонение.

```
from sklearn.preprocessing import StandardScaler
def standardize(x):
    return  StandardScaler().fit_transform(x)
```

```
Xm = np.mean(conc,axis=0)
Xs = np.std(conc,axis=0)
for i in range(160):
    conc[i,:] = (conc[i,:]-Xm)/(Xs)
```

2. Построение матрицы корреляций и составление списка признаков со слабой корреляцией

```
C = np.cov(conc.T,bias = True) # Determine NxN sized Cov. Matrix

print("Covariance matrix Dimensionality is: ",C.shape)
print("Covariance matrix is\n",C)
```

```
Covariance matrix Dimensionality is:  (4096, 4096)
Covariance matrix is
 [[1.         0.96293765 0.89397791 ... 0.20667445 0.216396   0.22003534]
 [0.96293765 1.         0.96546381 ... 0.19296505 0.20148196 0.19860027]
 [0.89397791 0.96546381 1.         ... 0.18452903 0.191088   0.17540085]
 ...
 [0.20667445 0.19296505 0.18452903 ... 1.         0.9812837  0.94195789]
 [0.216396   0.20148196 0.191088   ... 0.9812837  1.         0.9768517 ]
 [0.22003534 0.19860027 0.17540085 ... 0.94195789 0.9768517  1.        ]]
```

4. Реализация метода главных компонент

5. Реализация критериев выбора числа главных компонент (Кайзера, сломанной трости, каменистой осыпи)

6. Определение числа главных компонент

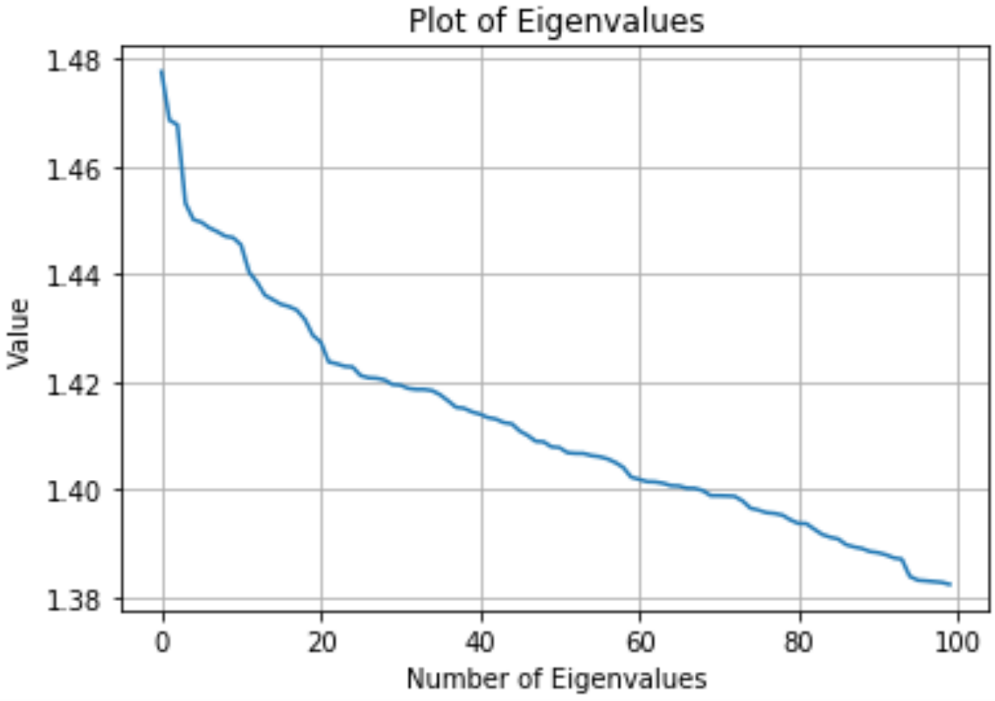
1) Критерий Кайзера. Сначала отберем только факторы, с собственными значениями, большими 1. По существу, это означает, что если фактор не выделяет дисперсию, эквивалентную, по крайней мере, дисперсии одной переменной, то он опускается. Этот критерий предложен Кайзером (Kaiser, 1960), и является, вероятно, наиболее широко используемым.

```
]: conc = np.concatenate((dogs,cats,),axis=0)
   Xv = np.var(conc, axis = 0)
   Xv=Xv[:]/4096
   Xv = Xv.tolist()
   Xv.sort(reverse = True)
   count = 0
   while (Xv[count]>1):
       count=count+1
   count
```

```
]: 2170
```

Таким образом по критерию Кайзера нам нужно взять 2170 факторов

2) Критерий каменистой осыпи. Критерий каменистой осыпи является графическим методом, впервые предложенным Кэттелем (Cattell, 1966). Нужно изобразить собственные значения, в виде простого графика. Кэттель предложил найти такое место на графике, где убывание собственных значений слева направо максимально замедляется. Предполагается, что справа от этой точки находится только "факториальная осыпь" - "осыпь" является геологическим термином, обозначающим обломки горных пород, скапливающиеся в нижней части скалистого склона.



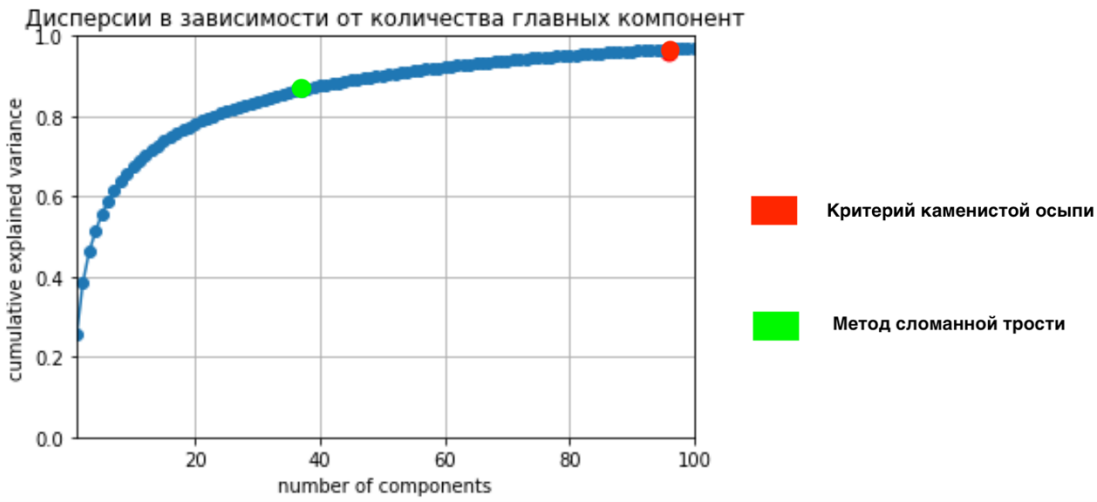
Смотря на график, по критерию каменистой осыпи я бы оставил **96** факторов

3) Метод сломанной трости. Набор нормированных собственных чисел сравнивается с распределением длин обломков трости единичной длины, сломанной в n-1-й случайно выбранной точке (точки разлома выбираются независимо и равнораспределены по длине трости).

По методу сломанной трости было получено 27 главных компонент.

In [166]:

2. График % описываемой дисперсии в зависимости от количества главных компонент. (пометить на графике числа ГК по разным критериям)



Из графика "Дисперсии в зависимости от количества главных компонент" видно что 64 фактора достаточно чтобы описать более 90% дисперсии