Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет Компьютерных сетей и систем

Кафедра Информатики

ЛАБОРАТОРНАЯ РАБОТА №3
«Переобучение и регуляризация»

БГУИР 1-40 81 04

Магистрант:
гр. 858642
Кукареко А.В.

Проверил:
Стержанов М. В.

Минск, 2019

ХОД РАБОТЫ

Задание.

Набор данных ex3data1.mat представляет собой файл формата *.mat (т.е. сохраненного из Matlab). Набор содержит две переменные X (изменения уровня воды) и y (объем воды, вытекающий из дамбы). По переменной X необходимо предсказать y. Данные разделены на три выборки: обучающая выборка (X, y), по которой определяются параметры модели; валидационная выборка (Xval, yval), на которой настраивается коэффициент регуляризации; контрольная выборка (Xtest, ytest), на которой оценивается качество построенной модели.

1. Загрузите данные ex3data1.mat из файла.
2. Постройте график, где по осям откладываются X и y из обучающей выборки.
3. Реализуйте функцию стоимости потерь для линейной регрессии с L2-регуляризацией.
4. Реализуйте функцию градиентного спуска для линейной регрессии с L2-регуляризацией.
5. Постройте модель линейной регрессии с коэффициентом регуляризации 0 и постройте график полученной функции совместно с графиком из пункта 2. Почему регуляризация в данном случае не сработает?
6. Постройте график процесса обучения (learning curves) для обучающей и валидационной выборки. По оси абсцисс откладывается число элементов из обучающей выборки, а по оси ординат - ошибка (значение функции потерь) для обучающей выборки (первая кривая) и валидационной выборки (вторая кривая). Какой вывод можно сделать по построенному графику?
7. Реализуйте функцию добавления p - 1 новых признаков в обучающую выборку (X2, X3, X4, …, Xp).
8. Поскольку в данной задаче будет использован полином высокой степени, то необходимо перед обучением произвести нормализацию признаков.
9. Обучите модель с коэффициентом регуляризации 0 и p = 8.
10. Постройте график модели, совмещенный с обучающей выборкой, а также график процесса обучения. Какой вывод можно сделать в данном случае?

11. Постройте графики из пункта 10 для моделей с коэффициентами регуляризации 1 и 100. Какие выводы можно сделать?
12. С помощью валидационной выборки подберите коэффиент регуляризации, который позволяет достичь наименьшей ошибки. Процесс подбора отразите с помощью графика (графиков).
13. Вычислите ошибку (потерю) на контрольной выборке.
14. Ответы на вопросы представьте в виде отчета..

Результат выполнения:

1. Загрузите данные ex3data1.mat из файла.

```
data = scipy.io.loadmat( 'ex3data1.mat' )
```

```
X_train, y_train = data['X'], data['y']

X_valid, y_valid = data['Xval'], data['yval']

X_test, y_test = data['Xtest'], data['ytest']

# добавляем  bias

X_train = np.insert(X_train, 0, 1, axis=1)
X_valid = np.insert(X_valid, 0, 1, axis=1)
X_test = np.insert(X_test, 0, 1, axis=1)
```

2. Постройте график, где по осям откладываются X и y из обучающей выборки.

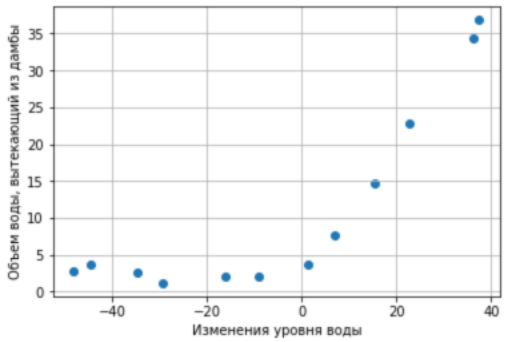


Рисунок 1 – график зависимости объёма воды, вытекающего из дамбы от изменения уровня воды.

3. Реализуйте функцию стоимости потерь для линейной регрессии с L2-регуляризацией.

Функция гипотезы:

```
def h(theta, X):
    return np.dot(X, theta)
```

Функция стоимости с L2-регуляризацией:

```
def J(theta, X, y, lmb = 0.):
    m = len(X)
    error = 0
    reg = 0

    h_res = h(theta, X).reshape(-1, 1)
    error = np.sum(np.power((h_res - y), 2)) / ( 2 * m )

    if lmb != 0:
        reg = np.sum(np.power(theta[1:], 2) * lmb ) / ( 2 * m )

    return error + reg
```

Пример работы функции стоимости с L2-регуляризацией:

```
J([1, 1], X_train, y_train, 0.5)
```

```
303.9723588869309
```

4. Реализуйте функцию градиентного спуска для линейной регрессии с L2-регуляризацией.

Функция градиентного спуска для линейной регрессии с L2-регуляризацией:

```
def gd_step(theta, X, y, lmb = 0.):
    m = len(X)
    gradient = 0

    h_res = h(theta, X).reshape(-1, 1)

    gradient = np.dot(X.T, (h_res - y)) / m

    if lmb != 0:
        reg = ((lmb / m) * np.array(theta)).reshape(-1, 1)
        gradient += reg

    return gradient

def gd_step_flatten(theta, X, y, lmb = 0.):
    return gd_step(theta, X, y, lmb).flatten()
```

```
gd_step_flatten([1, 1], X_train, y_train, 0.5)
```

```
array([-15.26134901, 598.20907751])
```

5. Постройте модель линейной регрессии с коэффициентом регуляризации 0 и постройте график полученной функции совместно с графиком из пункта 2. Почему регуляризация в данном случае не сработает?

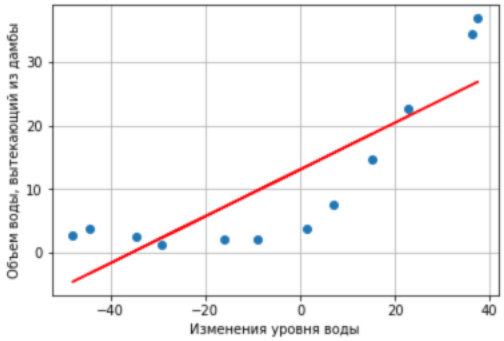
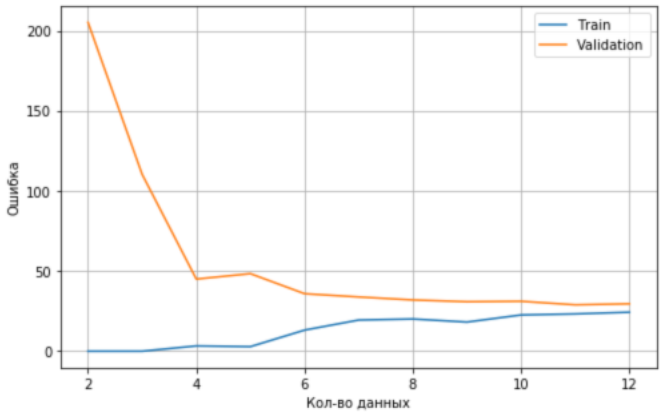


Рисунок 2 – график линейной регрессии с lambda = 0.

L2-регуляризация помогает решить проблему переобучения (overfitting, high variance), "сгладить кривую". Если посмотреть на график, то мы увидим, что наша модель страдает от недообучения (underfitting, high bias), следовательно L2-регуляризация тут не поможет.

6. Постройте график процесса обучения (learning curves) для обучающей и валидационной выборки. По оси абсцисс откладывается число элементов из обучающей выборки, а по оси ординат - ошибка (значение функции потерь) для обучающей выборки (первая кривая) и валидационной выборки (вторая кривая). Какой вывод можно сделать по построенному графику?




```
train_error: 24.317249588044152
validation_error: 29.55143162199776
```

Рисунок 3 – график “learning curves” с lambda = 0.

По графику можно сделать вывод, что модель страдает от "недообучения"(underfitting, high bias). Ошибка модели высока и добавление большего количества данных не принесет значимого прироста точности.

7. Реализуйте функцию добавления p - 1 новых признаков в обучающую выборку (X2, X3, X4, …, Xp).

Функция добавления (p – 1) новых признаков в обучающую выборку:

```
def gen_polynom(X, p):
    X_new = np.ones([len(X), p+1])
    for i in range(1, p+1):
        X_new[:, i] = X[:, 1] ** i ;

    return X_new

gen_polynom(np.array([[1, 2], [1, 3], [1, 4]]), 3)
```

```
array([[ 1.,  2.,  4.,  8.],
       [ 1.,  3.,  9., 27.],
       [ 1.,  4., 16., 64.]])
```

8. Поскольку в данной задаче будет использован полином высокой степени, то необходимо перед обучением произвести нормализацию признаков..

Функция нормализации признаков:

```
def create_normalizer(data):
    wo_bias = data[:,1:]
    mean = np.mean(wo_bias, axis=0)
    range = np.max(wo_bias, axis=0) - np.min(wo_bias, axis=0)
    std = np.std(wo_bias, axis=0)

    def norm_func(val):
        cp = val.copy()
        cp[:,1:] -= mean
        #cp[:,1:] /= range
        cp[:,1:] /= std

        return cp

    def denorm_func(val):
        cp = val.copy()
        #cp[:,1:] *= range
        cp[:,1:] *= std
        cp[:,1:] += mean

        return cp

    return norm_func, denorm_func
```

9. Обучите модель с коэффициентом регуляризации 0 и p = 8.

```
p = 8
X_train_poly = gen_polynom(X_train, p)
X_valid_poly = gen_polynom(X_valid, p)

norm_func, denorm_func = create_normalizer(X_train_poly)

X_train_poly_norm = norm_func(X_train_poly)
X_valid_poly_norm = norm_func(X_valid_poly)


theta_poly_2 = lin_reg(X_train_poly_norm, y_train, 0)
```

10. Постройте график модели, совмещенный с обучающей выборкой, а также график процесса обучения. Какой вывод можно сделать в данном случае?

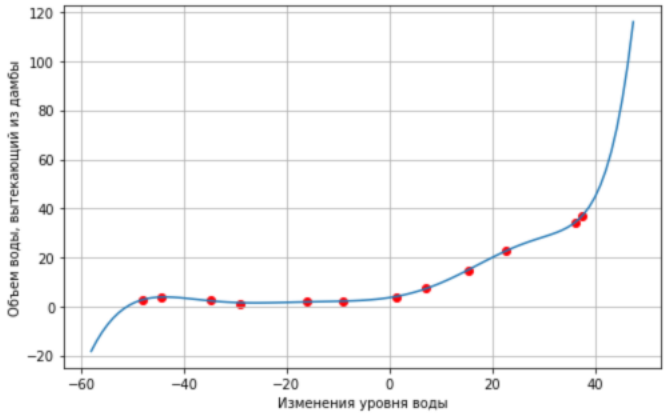
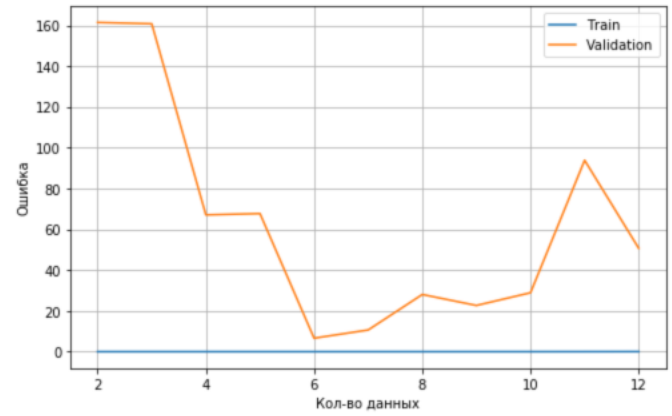


Рисунок 4 – график полученной модели с lambda = 0.




```
train_error: 0.03121924484764571
validation_error: 50.69359056624195
```

Рисунок 5 – график “learning curves” полученной модели с lambda = 0.

Если посмотреть на график функции, то видно, что она очень хорошо обучилась на "train" данных и имеет большой процент точности, в то же время, если посмотреть на график "learning curves", то мы увидим, что ошибка "train" очень мала - 0.03, а ошибка "validation" большая - 45.5. Эти показатели свидетельствуют о том, что модель "переобучилась" (overfitting, high variance). Для того, чтобы модель не переобучилась, можно использовать L2-регуляризацию.

11. Постройте графики из пункта 10 для моделей с коэффициентами регуляризации 1 и 100. Какие выводы можно сделать?

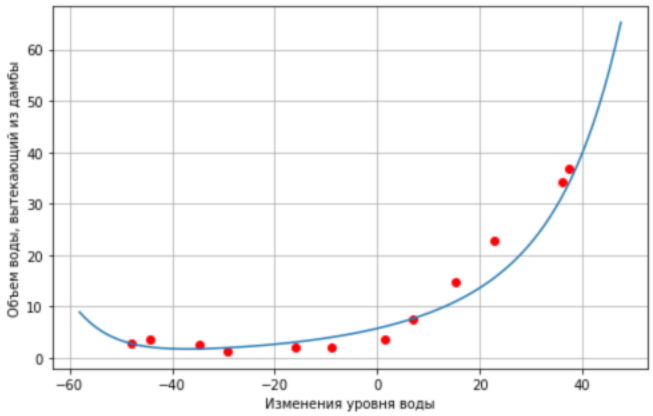
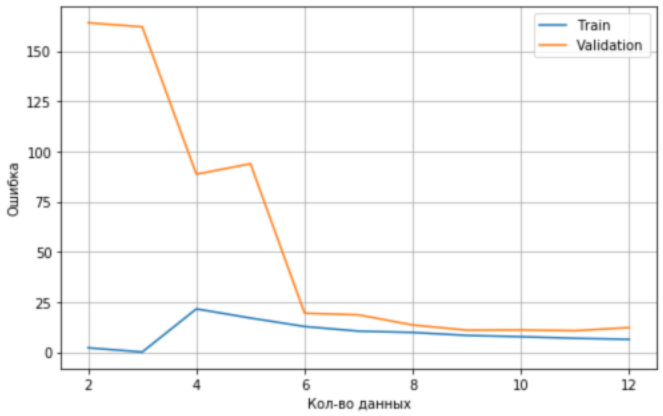


Рисунок 6 – график полученной модели с lambda = 1.




```
train_error: 6.493171435615369
validation_error: 12.367444036173632
```

Рисунок 7 – график “learning curves” полученной модели с lambda = 1.

Если посмотреть на график функции при (lambda = 1), то можно заметить, что хоть модель и не имеет такой процент точности как при (lambda = 0) однако она довольно точно обобщает входящие данных. Если посмотреть на график "learning curves" то мы увидим, что хоть ошибка "train" возросла с 0.03 до 6.49, зато ошибка "validation" уменьшилась с 45.5 до 12.3. Даныне метрики говорят о том, что у модели отсутствует "overfitting, high variance" и "underfitting, high bias".

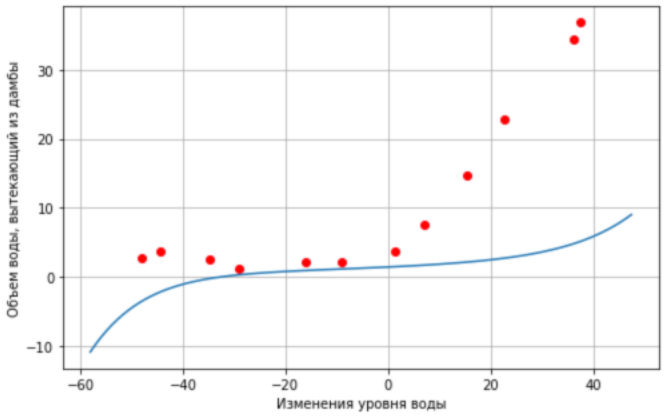
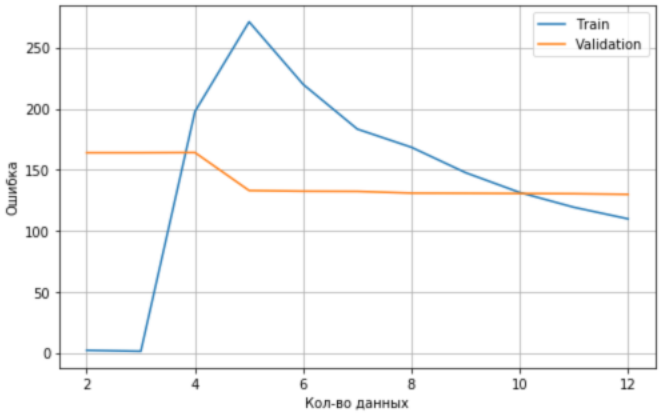


Рисунок 8 – график полученной модели с lambda = 100.




```
train_error: 109.80372494878571
validation_error: 129.86636806575024
```

Рисунок 9 – график “learning curves” полученной модели с lambda = 100.

Если посмотреть на график функции при (lambda = 100), то можно заметить, что хоть модель очень плохо обобщает входящие данные, и по графику она похожа на модель без полиномиальных признаков. Если посмотреть на график "learning curves" то мы увидим, что ошибки "train" и "validation" очень вклики, что говорит нам о том, что модель "недообучена"(underfitting, high bias). Решить проблему можно уменьшив параметр lambda.

12. С помощью валидационной выборки подберите коэффициент регуляризации, который позволяет достичь наименьшей ошибки. Процесс подбора отразите с помощью графика (графиков).

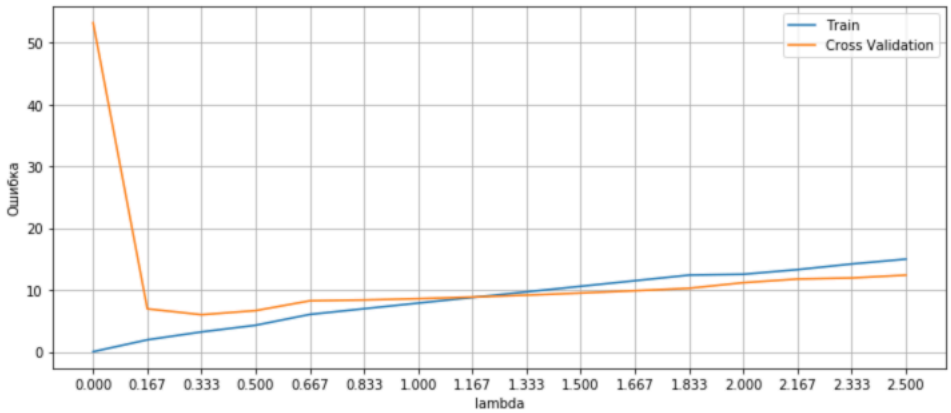


Рисунок 10 – график зависимости ошибки на валидационном и тренировочном сете от lambda.

По графику видно, что наименьшая ошибка достигается примерно при значении lambda ~ 1.17.

13. Вычислите ошибку (потерю) на контрольной выборке.

```
print(f'train_j = {train_j}')
print(f'valid_j = {valid_j}')
print(f'test_j = {test_j}')
```

```
train_j = 8.821571706513591
valid_j = 8.876200901441967
test_j = 7.111563401085762
```

При значении lambda = 1.17 ошибка на контрольной выборке составила 7.11.

Вывод.

В ходе выполнения лабораторной работы я ознакомился с понятием «переобучение», «недообучение» и узнал, какими способами можно решать эти проблемы. Так же в ходе работы изучил принципы работы L2-регуляризации, её влияние на «ошибку» а так же изучил технику «кросс-валидации», которая позволяет избежать зависимости модели от величины lambda.