|  |  |  |  |
| --- | --- | --- | --- |
| Министерство образования Республики Беларусь  Учреждение образования  Белорусский Государственный Университет Информатики и Радиоэлектроники | | | |
| Факультет | Компьютерных сетей и систем | | |
| Кафедра | Информатики  Дисциплина: Конструирование те технологии электронных вычислительных средств | | |
|  |  | | |
| ЛАБОРАТОРНАЯ РАБОТА №3  «Переобучение и регуляризация» | | | |
| Магистрант:  гр. 858341  Жишкевич С.А. | |  | 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.

1. data = scipy.io.loadmat('ex3data1.mat')
3. X\_train, y\_train = data['X'], data['y']
4. X\_valid, y\_valid = data['Xval'], data['yval']
5. X\_test, y\_test = data['Xtest'], data['ytest']
7. # добавляем  bias
9. X\_train = np.insert(X\_train, 0, 1, axis=1)
10. X\_valid = np.insert(X\_valid, 0, 1, axis=1)
11. 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:

1. **def** h(theta, X):
2. **return** np.dot(X, theta)

Функция стоимости с L2-регуляризацией:

1. **def** J(theta, X, y, lmb=0.):
2. m = len(X)
3. error = 0
4. reg = 0
6. h\_res = h(theta, X).reshape(-1, 1)
7. error = np.sum(np.power((h\_res - y), 2)) / (2 \* m)
9. **if** lmb != 0:
10. reg = np.sum(np.power(theta[1:], 2) \* lmb) / (2 \* m)
12. **return** error + reg

Пример работы функции стоимости с L2-регуляризацией:

print(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:

1. **def** gd\_step(theta, X, y, lmb=0.):
2. m = len(X)
3. gradient = 0
5. h\_res = h(theta, X).reshape(-1, 1)
7. gradient = np.dot(X.T, (h\_res - y)) / m
9. **if** lmb != 0:
10. reg = ((lmb / m) \* np.array(theta)).reshape(-1, 1)
11. gradient += reg
13. **return** gradient

16. **def** gd\_step\_flatten(theta, X, y, lmb=0.):
17. **return** gd\_step(theta, X, y, lmb).flatten()
19. **print**(gd\_step\_flatten([1, 1], X\_train, y\_train, 0.5))
20. 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?



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:

1. **def** gen\_polynom(X, p):
2. X\_new = np.ones([len(X), p + 1])
3. **for** i **in** range(1, p + 1):
4. X\_new[:, i] = X[:, 1] \*\* i;
6. **return** X\_new

9. **print**(gen\_polynom(np.array([[1, 2], [1, 3], [1, 4]]), 3))
10. array([[ 1., 2., 4., 8.],
11. [ 1., 3., 9., 27.],
12. [ 1., 4., 16., 64.]])

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

Функция нормализации признаков:

1. **def** create\_normalizer(data):
2. wo\_bias = data[:, 1:]
3. mean = np.mean(wo\_bias, axis=0)
4. range = np.max(wo\_bias, axis=0) - np.min(wo\_bias, axis=0)
5. std = np.std(wo\_bias, axis=0)
7. **def** norm\_func(val):
8. cp = val.copy()
9. cp[:, 1:] -= mean
10. # cp[:,1:] /= range
11. cp[:, 1:] /= std
13. **return** cp
15. **def** denorm\_func(val):
16. cp = val.copy()
17. # cp[:,1:] \*= range
18. cp[:, 1:] \*= std
19. cp[:, 1:] += mean
21. **return** cp
23. **return** norm\_func, denorm\_func

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

1. p = 8
2. X\_train\_poly = gen\_polynom(X\_train, p)
3. X\_valid\_poly = gen\_polynom(X\_valid, p)
5. norm\_func, denorm\_func = create\_normalizer(X\_train\_poly)
7. X\_train\_poly\_norm = norm\_func(X\_train\_poly)
8. X\_valid\_poly\_norm = norm\_func(X\_valid\_poly)

11. 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.



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.



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.



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.

1. **print**(f'train\_j = {train\_j}')
2. **print**(f'valid\_j = {valid\_j}')
3. **print**(f'test\_j = {test\_j}')
4. train\_j = 8.821571706513591
5. valid\_j = 8.876200901441967
6. 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.