

به نام خدا

تمرین سوم مباحث ویژه
استفاده از تکنیک تنظیم L2 در تمرین اول و دوم

استاد دولتشاهی

پریسا مبارک

40211415006

شرح تمرین:

میخواهیم دو تمرین قبل را با استفاده از تکنیک "تنظیم L2" بر روی دیتاست هایی که در تمرین ها داریم، پیاده سازی کرده و موارد زیر را انجام دهیم:

- (1) خطای تست استفاده از تنظیم را با خطای هر روش بدون استفاده از تکنیک "تنظیم" مقایسه کنیم.
- (2) مقادیر پارامترها را در هر روش با هم مقایسه کنیم.

برای این کار کافیت دو تغییر در تمرینات قبل ایجاد کنیم. تغییر اول در تابع خطا و تغییر دوم در دلتاتاهای پارامترهای (dtheta ها) تمرین مربوطه هست.

استفاده از تکنیک تنظیم L2 در تمرین اول :

مطابق توضیحات بالا درکد مربوط به تمرین اول ابتدا تابع خطا را تغییر میدهیم. برای این کار طبق کد زیر باید یک ضریب L2 که آن را λ نامیدم را در فرمول L2 (مجموع توان دوم پارامترها) ضرب کنم و آن را به کد قبلی اضافه کنم:

```
def mse(y_pred, y_true): # تغییر تابع خطا
    regularization_l2 = lambda_ * np.sum(theta[1:]**2)
    return 0.5 * ((y_pred - y_true) ** 2).mean() + regularization_l2
```

حال یک مقدار بین صفر و یک به λ میدهیم:

```
alpha = 0.01
lambda_ = 0.1 # تنظیم مقدار
```

تغییر دوم همانطور که گفتم در دلتاتاهای پارامترهای غیر ثابت (یعنی تمام پارامترها به جز عرض از مبدا) می باشد همانند کد زیر:

```

def update_step():
    global theta, costs_train
    y_pred = h(xn_train, theta)
    costs_train.append(mse(y_pred, y_train))
    dtheta = np.zeros([xn_train.shape[0], xn_train.shape[1]])
    dtheta[:, 0] = (y_pred - y_train)

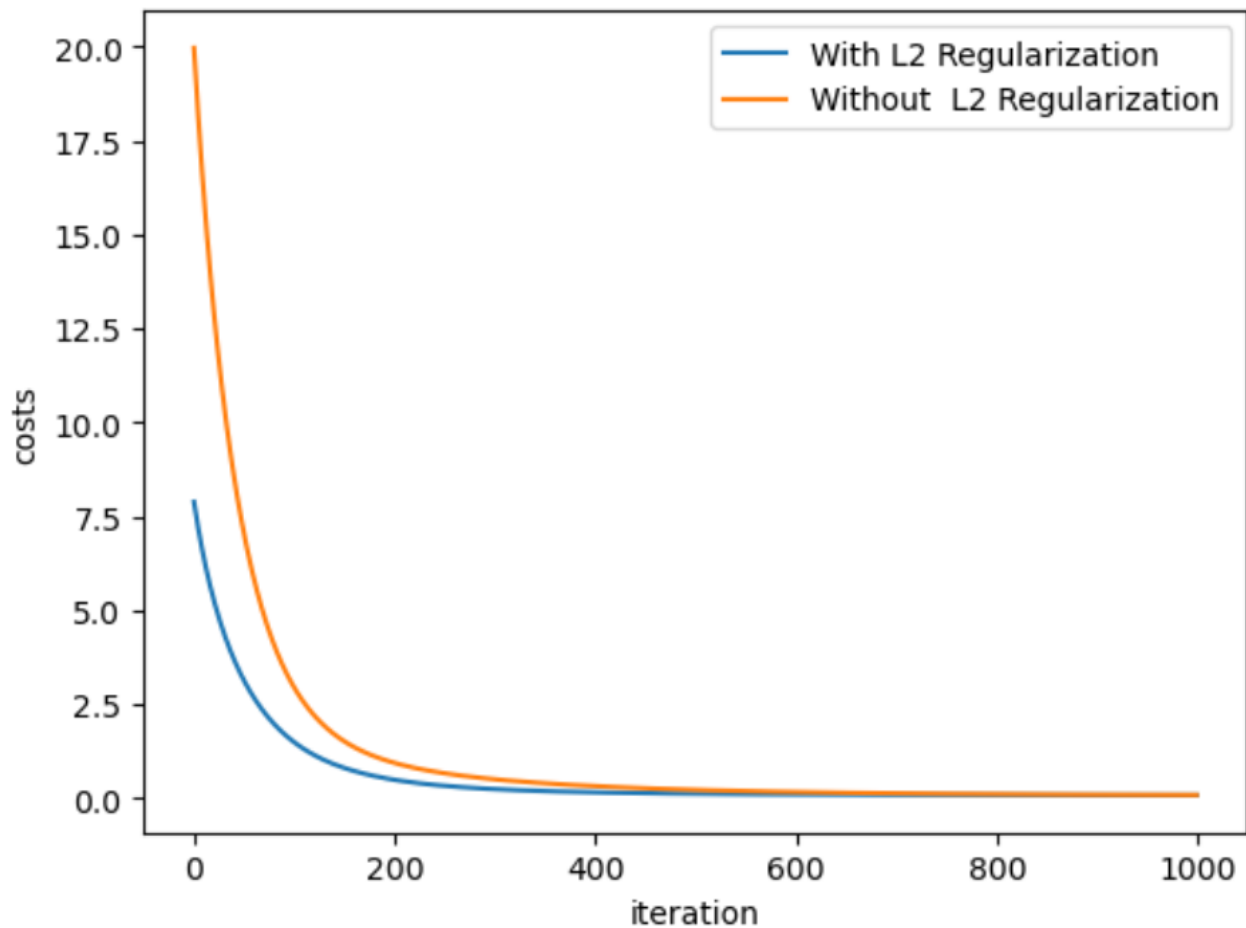
    for i in range(1, xn_train.shape[1]):
        dtheta[:, i] = xn_train[:, i] * dtheta[:, 0]
        dtheta[:, i] += lambda_ * theta[i]  # تغییر در دلتا های غیر ثابت

    for i in range(0, xn_train.shape[1]):
        theta[i] -= alpha * dtheta[:, i].mean()

```

اگر این تغییرات را اعمال کنیم، تمرین اول را با تنظیم L2 انجام دادیم.

حال برای مقایسه خطای تست استفاده از تنظیم با خطای تست بدون استفاده از تکنیک "تنظیم" خطای تست های دو روش را با نام های `costs_test_without_regularization` و `costs_test_with_regularization` نامگذاری کردم و آن ها را رسم کردم و خروجی مطابق تصویر زیر نمایش داده شد:



در تفسیر نمودار بالا میتوان گفت استفاده از تکنیک تنظیم، خطاها را کاهش داده است. همانطور که مشاهده میشود نمودار آبی رنگ خطای تست با تنظیم را نشان داده و نمودار نارنجی خطای تست بدون تنظیم (یعنی همان تمرین اول) را نشان داده است.

حال برای مقایسه مقادیر پارامترها در روش با تنظیم و بدون تنظیم نیز مقادیر پارامترها را در هر روش در یک لیست ذخیره کردم و آن ها را نمایش دادم و خروجی به صورت تصویر زیر چاپ شد:

```
Parameters with L2 Regularization: [ 2.7872483 -0.01175222 -0.05460419 0.08676981 0.07014689 0.03671803
0.06613355 0.06859933 0.04826353 -0.03238736 0.01522927 -0.08265752
0.37498675]
Parameters without L2 Regularization: [ 2.78715584 0.00704818 0.16169953 -0.40944877 0.030858 0.0908597
0.04336851 -0.23691753 0.05818268 -0.07038163 -0.01365087 -0.17518891
0.29499877]
```

طبق تصویر بالا مشاهده میکنیم که استفاده از تکنیک تنظیم، مقادیر پارامترها را کاهش داده است.

استفاده از تکنیک تنظیم L2 در تمرین دوم :

دقیقا همانند بخش بالا عمل میکنیم. یعنی فقط تابع خطا و دلتاتهای پارامترهای غیر ثابت در تمرین دوم را تغییر میدهیم. به این صورت:

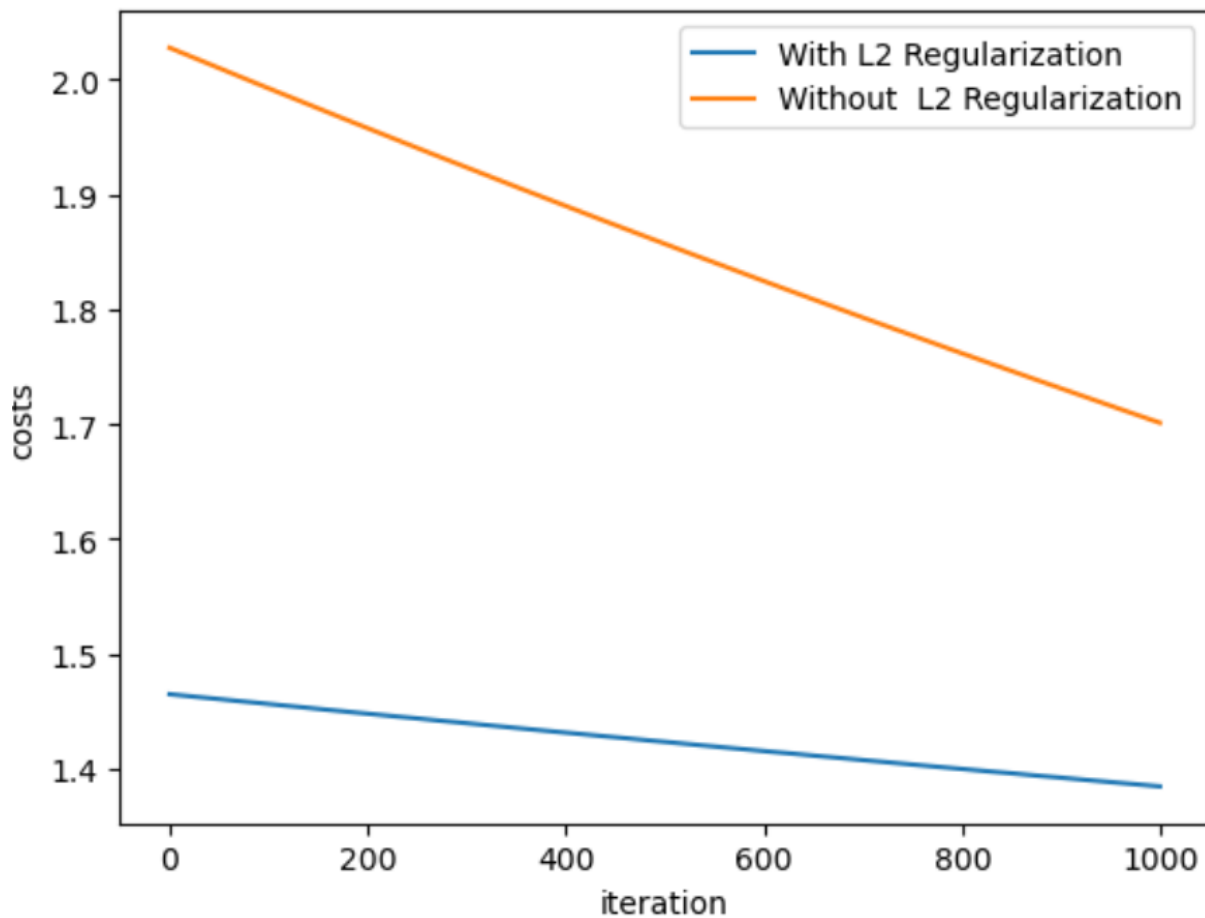
```
def log_loss(y_pred, y_true): #تغییر تابع خطا
    epsilon = 1e-10
    regularization_l2 = lambda_ * np.sum(theta[1:]**2)
    loss = -y_true * np.log(y_pred + epsilon) - (1 - y_true) * np.log(1 - y_pred + epsilon) + regularization_l2
    return loss.mean()
```

```
def update_step():
    global theta, costs_train
    y_pred = h(xn_train, theta)
    costs_train.append(log_loss(y_pred, y_train))
    dtheta = np.zeros([xn_train.shape[0], xn_train.shape[1]])
    dtheta[:, 0] = (y_pred - y_train)

    for i in range(1, xn_train.shape[1]):
        dtheta[:, i] = xn_train[:, i] * dtheta[:, 0]
        dtheta[:, i] += lambda_ * theta[i] # تغییر در دلتاتهای غیرثابت

    for i in range(0, xn_train.shape[1]):
        theta[i] -= alpha * dtheta[:, i].mean()
```

خروجی نمودار خطاهای تست با تنظیم و بدون تنظیم نیز همانند تصویر زیر نمایش داده شد:



طبق تصویر بالا مشاهده میکنیم که استفاده از تکنیک تنظیم، خطای تست را کاهش داده است.

```
Parameters with L2 Regularization: [-0.90419064 -1.14409705 -1.60050845 -0.26949957 -0.8359296  0.58633823
-0.67542494  0.13153101  0.46370915  0.02666088 -0.94043998 -1.49908276
 0.189767 ]
Parameters without L2 Regularization: [ 1.51376594 -0.95343185  0.00544739  0.94592661 -0.21165835  0.94111196
 0.39160364 -0.60477932 -0.32422143  0.47098268  0.52748121 -0.94632785
 1.56911587]
test_accuracy_With_L2 Regularization: 82.02%
test_accuracy_without_L2 Regularization: 58.43%
```

طبق تصویر بالا مشاهده میکنیم که استفاده از تکنیک تنظیم، مقادیر بعضی پارامترها را کاهش داده است و همچنین دقت با تنظیم بهتر شده است.

استفاده از تکنیک تنظیم L2 در رگرسیون خطی با رابطه نرمال :

برای این بخش نیز همانند بخش های قبل به تابع هزینه lamda ضربدر مجموع توان دوم پارامترها(بجز پارامتر theta0) را اضافه میکنیم. اما چون به جای گرادیان کاهشی از رابطه نرمال استفاده میکنیم فرمول مربوط به معادله نرمال را به این صورت تغییر میدهیم:

```
I = np.eye(xn_train.shape[1])
I[0, 0] = 0
theta_with_regularization = np.linalg.pinv(xn_train.T.dot(xn_train) + lambda_ * I).dot(xn_train.T).dot(y_train)
```

مشاهده میکنیم که مقادیر بعضی از پارامترها با تنظیم، کاهش یافته است:

```
dataset:
[[52.  45.  48.   ...  2.   32.    1.94]
 [57.  90.  72.   ...  1.   32.    1.95]
 [57.  55.  43.   ...  3.   32.    1.95]
 ...
 [68.  95.  72.   ...  1.   45.    3.81]
 [52.  92.  72.   ...  1.   45.    3.81]
 [52.  90.  72.   ...  1.   45.    3.81]]
Cost with L2 Regularization: 0.08329263973144521
Cost without L2 Regularization: 0.05860494592806733
Parameters with L2 Regularization: [ 2.78736585e+00  1.97725543e-03 -1.27841739e-01 -1.68139308e-02
  9.33650031e-02  4.80138586e-02  1.27979540e-01  7.24782089e-03
  7.60076402e-02 -4.82724726e-02  6.00777976e-02 -1.22890000e-01
  4.19857826e-01]
Parameters without L2 Regularization: [ 2.78736585e+00  2.03046512e-03 -1.28388935e-01 -1.68772543e-02
  9.37140798e-02  4.79782045e-02  1.28387941e-01  7.32389304e-03
  7.61533397e-02 -4.83318029e-02  6.02779285e-02 -1.23017629e-01
  4.20373781e-01]
```

نتیجه گیری:

بطور کلی، تکنیک تنظیم L2 برای کاهش اثر بیش‌برازش (overfitting) در مدل‌های رگرسیون و کلاسیفایر استفاده می‌شود و برای اعمال آن به کد ، کافیت جمله تنظیم L2 را به تابع خطا اضافه کنیم و همچنین در به روزرسانی مراحل آموزش مدل، جمله تنظیم L2 را به گرادیان اضافه کنیم.

بیش برآزش زمانی به وجود می آید که مدل بیش از حد روی داده های آموزشی فیت شده باشد یعنی خطای آموزشی کم هست اما خطای داده های تست زیاد هست. در این حالت میتوانیم از تنظیم استفاده کنیم. این روش با کم کردن مقادیر پارامترها، اثر بیش برآزش را کاهش و یا حذف می کند و خطای تست را کاهش می دهد.