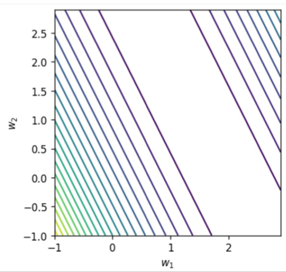
CS/ECE/ME532 Activity 18

Estimated time: 15 mins for P1, 20 mins for P2, 15 mins for P3, 20 mins for P4

1. A breast cancer gene database has approximately 8000 genes from 100 subjects. The label y_i is the disease state of the ith subject (+1 if no cancer, -1 if breast cancer). Suppose we build a linear classifier that combines the 8000 genes, say $\boldsymbol{g}_i, i = 1, 2, \ldots, 100$ to predict whether a subject has cancer $\hat{y}_i = \text{sign}\{\boldsymbol{g}_i^T \boldsymbol{w}\}$. Note that here $\boldsymbol{g}_i$ and $\boldsymbol{w}$ are 8000-by-1 vectors. You recall from the previous period that the least-squares problem for finding classifier weights has no unique solution.

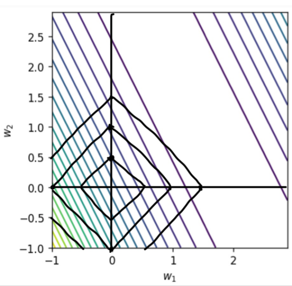
Your hypothesis is that a relatively small number of the 8000 genes are predictive of the cancer state. Identify a regularization strategy consistent with this hypothesis and justify your choice. L1 norm regularization will find a sparse solution that makes unimportant features/genes have a weight of Ø. Thus only a small number of genes are chosen.

2. Consider the least-squares problem $\min_{\boldsymbol{w}} ||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}||_2^2$ where $\boldsymbol{y} = 4$ and $\boldsymbol{X} = \begin{bmatrix} 2 & 1 \end{bmatrix}$.



a) Does this problem have a unique solution? Why or why not? No, the rank of X is 1 which will be less than the dimensions of w.

b) Sketch the contours of the cost function $f(\boldsymbol{w}) = ||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}||_2^2$ in the $w_1 - w_2$ plane.

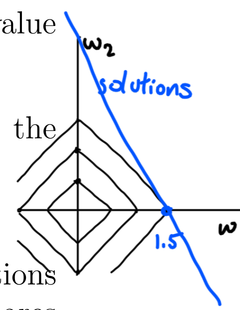


c) Now consider the LASSO $\min_{\boldsymbol{w}} ||\boldsymbol{w}||_1$ subject to $||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}||_2^2 < 1$. Find the solution using the following steps

i. Repeat your sketch from part **b)**.

ii. Add a sketch of $||\boldsymbol{w}||_1 = c$

iii. Find the $\boldsymbol{w}$ that satisfies $||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}||_2^2 = 1$ with the minimum possible value of $||\boldsymbol{w}||_1$. $w = [1.5, 0]^T$

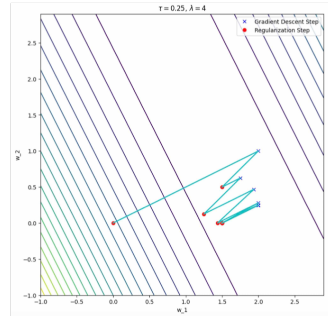


d) Use your insight from the previous part to sketch the set of solutions to the problem $\min_{\boldsymbol{w}} ||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}||_2^2 + \lambda||\boldsymbol{w}||_1$ for $0 < \lambda < \infty$.

3. The script provided has a function that will compute a specified number of iterations of the proximal gradient descent algorithm for solving the ℓ_1-regularized least-squares problem

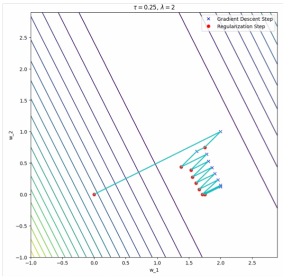
$$\min_{\boldsymbol{w}} ||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}||_2^2 + \lambda||\boldsymbol{w}||_1$$

The script will get you started displaying the path taken by the weights in the proximal gradient descent iteration superimposed on a contour plot of the squared error surface for the cost function defined in problem **2.** part **b)** starting from $\boldsymbol{w}^{(0)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$. The script assumes $\lambda = 4$ and $\tau = 1/4$.



Include the plots you generate below with your submission.

a) How many iterations does it take for the algorithm to converge to the solution? What is the converged value for $\boldsymbol{w}$? After 5 iterations the algorithm converges $w = [1.5, 0]^T$



b) Change to $\lambda = 2$. How many iterations does it take for the algorithm to converge to the solution? What is the converged value for $\boldsymbol{w}$? Converges after 8 interations $w = [1.75, 0]^T$

c) Explain what happens to the weights in the regularization step. Any weights from gradient descent that are between $\frac{-\lambda z}{2}$ and $\frac{\lambda z}{2}$ become 0. This means that a smaller λ will decrease this interval

4. Use the proximal gradient algorithm to solve $\min_{\boldsymbol{w}} ||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}||_2^2 + 4||\boldsymbol{w}||_1$ for the parameters defined in problem **2**.

a) What is the maximum value for the step size in the negative gradient direction, τ? $\frac{2}{||A||_{op}^2} = \frac{2}{5} = 0.4$

b) Suppose $\tau = 0.1$ and you start at $\boldsymbol{w}^{(0)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$. Calculate the first two complete iterations of the proximal gradient algorithm and depict $\boldsymbol{w}^{(0)}, \boldsymbol{z}^{(1)}, \boldsymbol{w}^{(1)}, \boldsymbol{z}^{(2)}$ and $\boldsymbol{w}^{(2)}$ on a sketch of the cost function identical to the one you created in problem **2.b)**.

$$z^k = w^k - z X^T (X w^k - y)$$

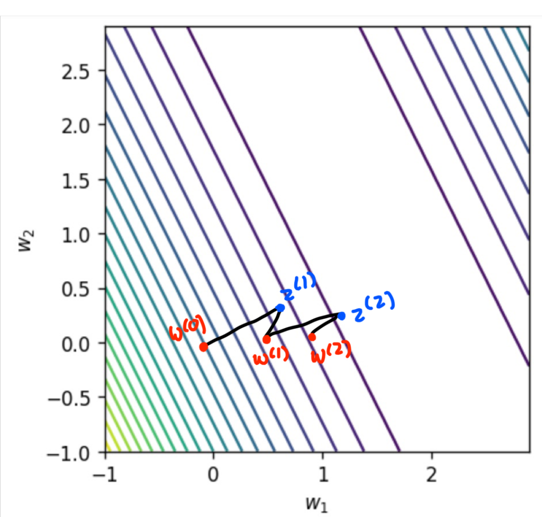
$$\underbrace{\frac{\lambda z}{2} = \frac{4(0.1)}{2} = 0.2}$$

$$w^{(k+1)} = \min_w \sum (z_i^k - w_i)^2 + \lambda z |w_i|$$

$$w_i^{(k+1)} = (|z_i| - \tfrac{\lambda z}{2})_+ \operatorname{sign}(z_i)$$

$$z^1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - 0.1 \begin{bmatrix} 2 \\ 1 \end{bmatrix} (0 - 4) = \begin{bmatrix} 0.8 \\ 0.4 \end{bmatrix}$$

$$w^1 = \begin{bmatrix} (0.8 - 0.2)(1) \\ (0.4 - 0.2)(1) \end{bmatrix} = \begin{bmatrix} 0.6 \\ 0.2 \end{bmatrix}$$



$$z^2 = \begin{bmatrix} 0.6 \\ 0.2 \end{bmatrix} - 0.1 \begin{bmatrix} 2 \\ 1 \end{bmatrix} (1.4 - 4) = \begin{bmatrix} 1.12 \\ 0.46 \end{bmatrix}$$

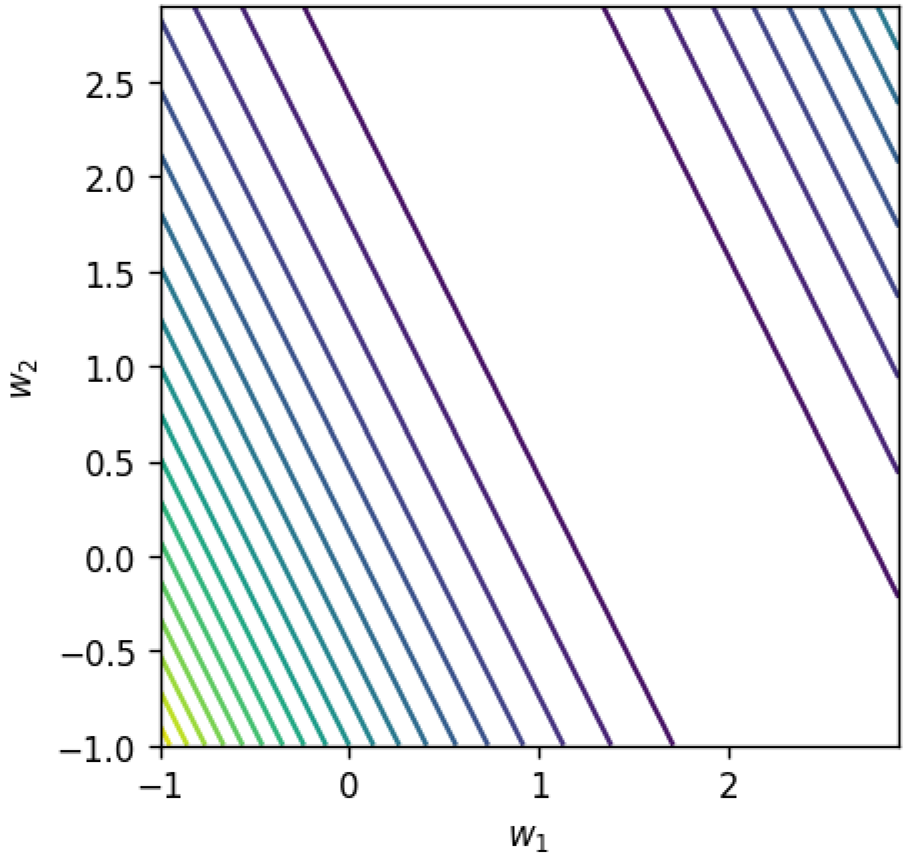
$$\begin{bmatrix} 0.52 \\ -0.26 \end{bmatrix}$$

$$w^2 = \begin{bmatrix} (1.12 - 0.2)(1) \\ (0.46 - 0.2)(1) \end{bmatrix} = \begin{bmatrix} 0.92 \\ 0.26 \end{bmatrix}$$

Activity 18

Setup

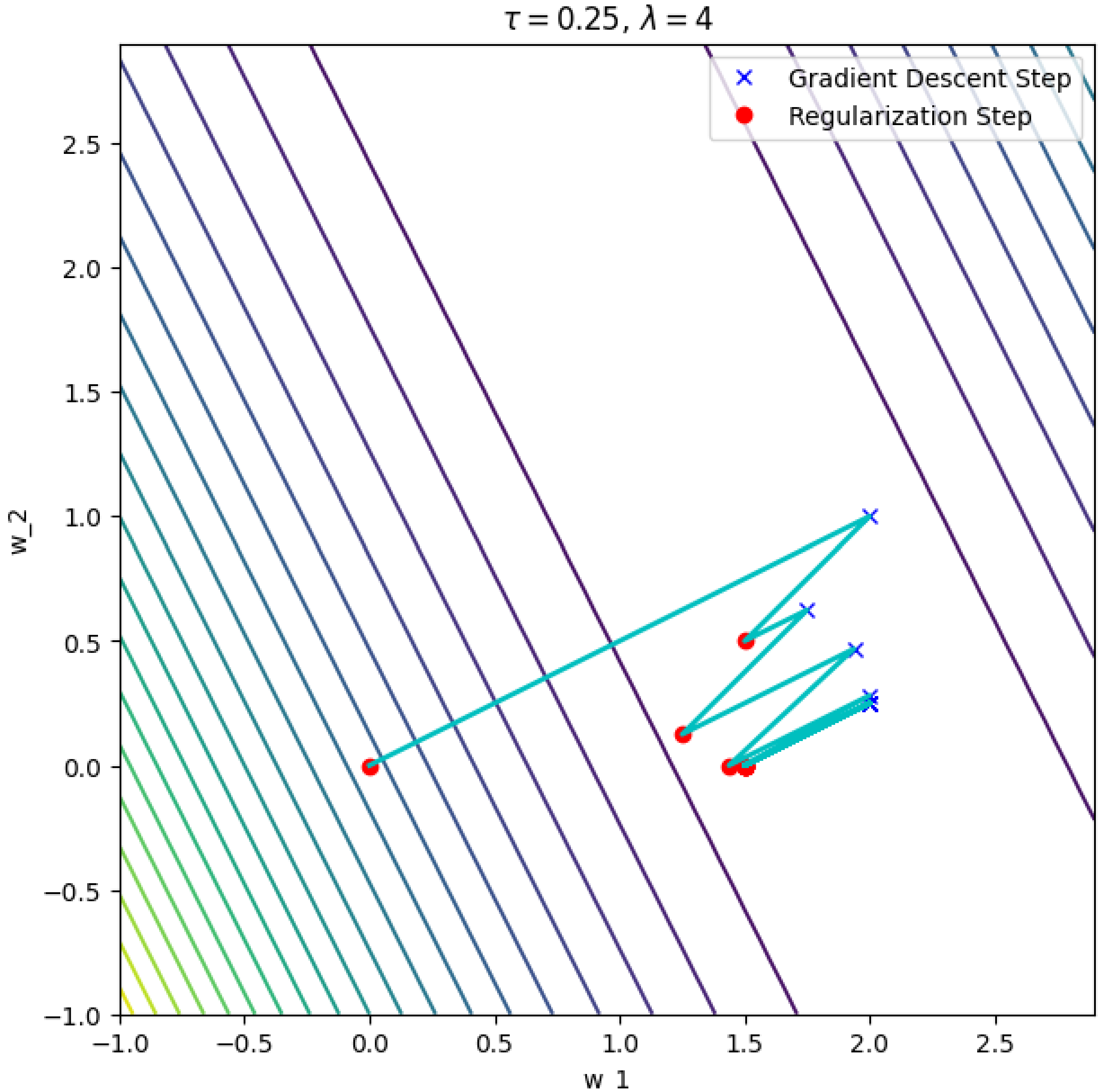
```
import numpy as np
import matplotlib.pyplot as plt


def prxgraddescent_l1(X,y,tau,lam,w_init,it):

## compute it iterations of L2 proximal gradient descent starting at w1
## w_{k+1}= (w_k - tau*X'*(X*w_k - y)/(1+lam*tau)
## step size tau
    W = np.zeros((w_init.shape[0], it+1))
    Z = np.zeros((w_init.shape[0], it+1))
    W[:,[0]] = w_init
    for k in range(it):
        Z[:,[k+1]] = W[:,[k]] - tau * X.T @ (X @ W[:,[k]] - y);
        W[:,[k+1]] = np.sign(Z[:,[k+1]])* np.clip(
            np.abs(Z[:,[k+1]])-lam*tau/2,0,float("inf"))

    return W,Z


## Proximal gradient descent trajectories
## Least Squares Problem
X = np.array([[2, 1]])
y = np.array([[4]])

### Find values of f(w), the contour plot surface for
w1 = np.arange(-1,3,.1)
w2 = np.arange(-1,3,.1)
fw = np.zeros((len(w1), len(w2)))
for i in range(len(w2)):
    for j in range(len(w1)):
        w = np.array([ [w1[j]], [w2[i]] ])
        fw[i,j] = (X @ w - y)**2

### Plot the countours
plt.figure(num=None, figsize=(4, 4), dpi=120)
plt.contour(w1,w2,fw,20)
plt.xlim([-1,3])
plt.ylim([-1,3])
plt.xlabel('$w_1$')
plt.ylabel('$w_2$')
plt.axis('square');
```



Question 3a)

```
## Find and display weights generated by gradient descent

w_init = np.array([[0],[0]])
lam = 4;
it = 10
tau = 0.25
W,Z = prxgraddescent_l1(X,y,tau,lam,w_init,it)
# Concatenate gradient and regularization steps to display trajectory
G = np.zeros((2,0))
for i in range(it):
    G = np.hstack((G,np.hstack((W[:,[i]],Z[:,[i+1]]))))

print(W)
print(W[:, it])

plt.figure(figsize=(7,7))
plt.contour(w1,w2,fw,20)
plt.plot(Z[0,1::],Z[1,1:],'bx',linewidth=2, label="Gradient Descent Step")
plt.plot(W[0,:],W[1,:],'ro',linewidth=2, label="Regularization Step")
plt.plot(G[0,:],G[1,:],'-c',linewidth=2)
plt.legend()
plt.xlabel('w_1')
plt.ylabel('w_2')
plt.title('$\\tau = $'+str(tau)+', $\lambda = $'+str(lam));
```

```
[[0.     1.5    1.25   1.4375 1.5    1.5    1.5    1.5    1.5    1.5
  1.5   ]
 [0.     0.5    0.125  0.     0.     0.     0.     0.     0.     0.
  0.    ]]
[1.5 0. ]
```

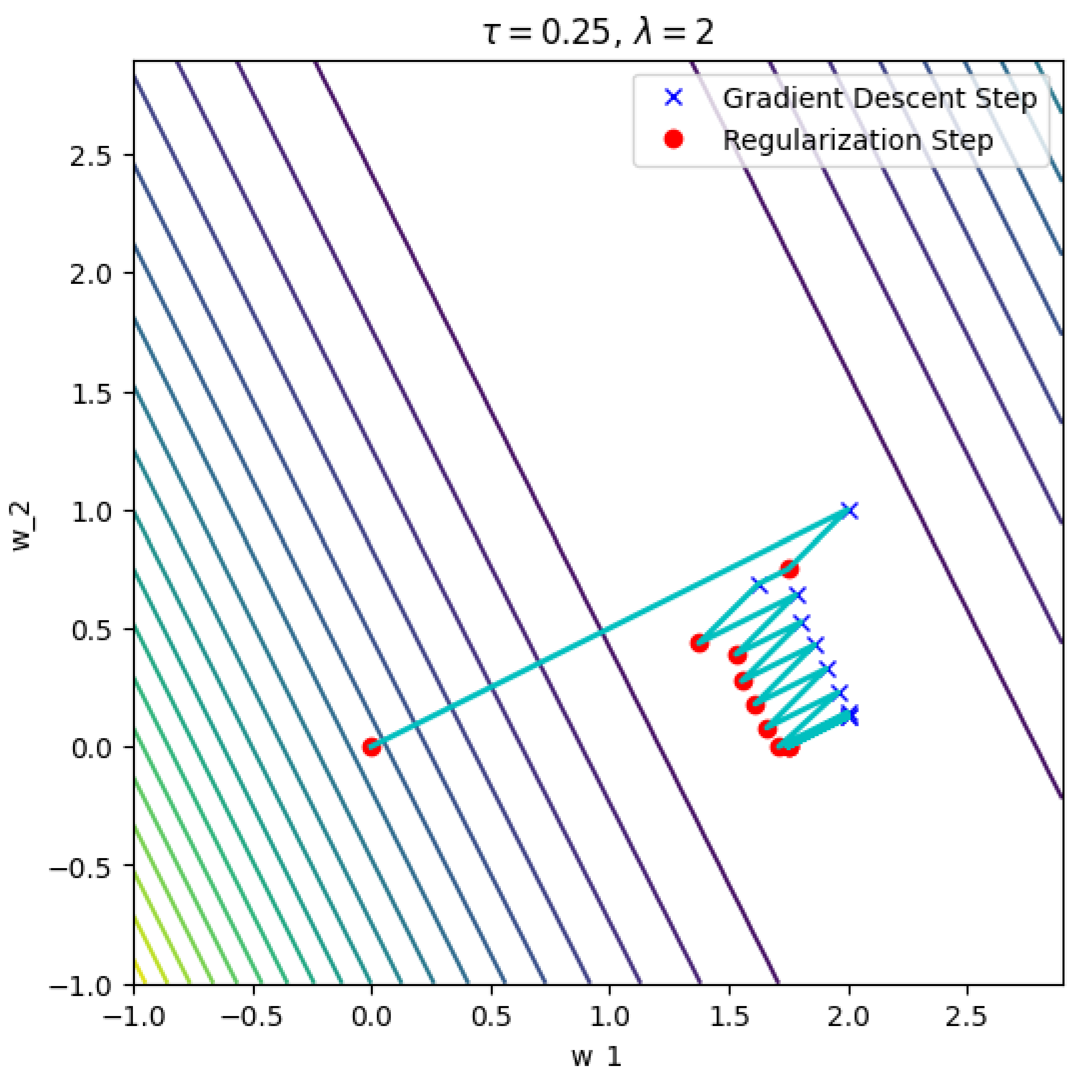


Question 3b)

```
## Find and display weights generated by gradient descent

w_init = np.array([[0],[0]])
lam = 2;
it = 10
tau = 0.25
```

```
W,Z = prxgraddescent_l1(X,y,tau,lam,w_init,it)
# Concatenate gradient and regularization steps to display trajectory
G = np.zeros((2,0))
for i in range(it):
    G = np.hstack((G,np.hstack((W[:,[i]],Z[:,[i+1]]))))

print(W)
print(W[:, it])

plt.figure(figsize=(6,6))
plt.contour(w1,w2,fw,20)
plt.plot(Z[0,1::],Z[1,1:],'bx',linewidth=2, label="Gradient Descent Step")
plt.plot(W[0,:],W[1,:],'ro',linewidth=2, label="Regularization Step")
plt.plot(G[0,:],G[1,:],'-c',linewidth=2)
plt.legend()
plt.xlabel('w_1')
plt.ylabel('w_2')
plt.title('$\\tau = $'+str(tau)+', $\lambda = $'+str(lam));
```

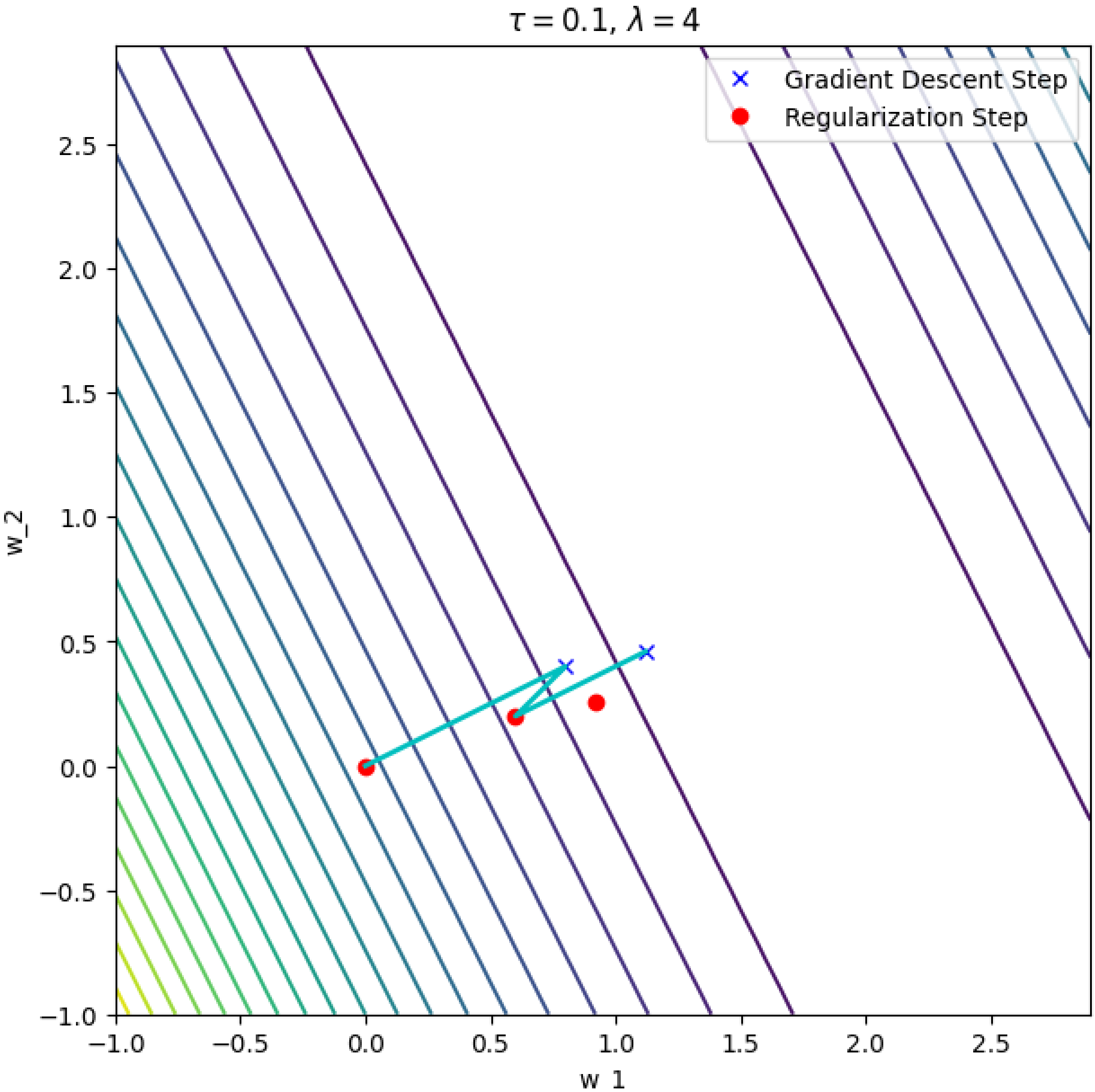
```
[[0.         1.75       1.375      1.53125    1.5546875  1.61132812
  1.65966797 1.71008301 1.75       1.75       1.75      ]
 [0.         0.75       0.4375     0.390625   0.27734375 0.18066406
  0.07983398 0.         0.         0.         0.        ]]
[1.75 0.  ]
```



Question 4)

```
## Find and display weights generated by gradient descent

w_init = np.array([[0],[0]])
lam = 4;
it = 2
tau = 0.1
W,Z = prxgraddescent_l1(X,y,tau,lam,w_init,it)
# Concatenate gradient and regularization steps to display trajectory
G = np.zeros((2,0))
for i in range(it):
    G = np.hstack((G,np.hstack((W[:,[i]],Z[:,[i+1]]))))

print(W)
print(W[:, it])

plt.figure(figsize=(7,7))
plt.contour(w1,w2,fw,20)
plt.plot(Z[0,1::],Z[1,1:],'bx',linewidth=2, label="Gradient Descent Step")
plt.plot(W[0,:],W[1,:],'ro',linewidth=2, label="Regularization Step")
plt.plot(G[0,:],G[1,:],'-c',linewidth=2)
plt.legend()
plt.xlabel('w_1')
plt.ylabel('w_2')
plt.title('$\\tau = $'+str(tau)+', $\lambda = $'+str(lam));
```

```
[[0.   0.6  0.92]
 [0.   0.2  0.26]]
[0.92 0.26]
```



✓ 0s completed at 12:19 PM