LipSDP using CVXPY

In [1]:
```
# importing necessary libraries
import cvxpy as cp
import numpy as np
```

In [2]:
```
# find_l_1 computes the lipschitz constant simply by computing the spec
# since it is a single layer neural network

def find_l_1(W):
    # computes 2-norm also known as Spectral norm
    l = np.linalg.norm(W, ord=2)
    return l
```

In [3]:
```
# test case 1
W = np.array([[1, 0],[0, 1]]) #weight matrix

lipschitz_constant = find_l_1(W)
print(f"Lipschitz constant: {lipschitz_constant}")
```

```
Lipschitz constant: 1.0
```

In [4]:
```
# test case 2
W = np.array([[62, 10],[-19, 10]]) #weight matrix

lipschitz_constant = find_l_1(W)
print(f"Lipschitz constant: {lipschitz_constant}")
```

```
Lipschitz constant: 65.19698634456758
```

In [5]:
```
# test case 3
W = np.array([[1020, 35783],[3, 371]]) #weight matrix

lipschitz_constant = find_l_1(W)
print(f"Lipschitz constant: {lipschitz_constant}")
```

```
Lipschitz constant: 35799.45644372084
```

In [6]:
```
# find_l_2 computes the lipschitz constant by using the formula
#||f(x)-f(y)|| <= L||x-y||

def find_l_2(W):

    # defining the input dimension and output dimension
    n = W.shape[1]
    m = W.shape[0]

    # defining the variables
    x = cp.Variable(n)
    y = cp.Variable(n)
    t = cp.Variable(m)

    # defining the objective function
    obj = cp.Maximize(-cp.max(t))

    # defining the constraints
    constraints = [
        t >= 0,
        t >= cp.sum(cp.abs(W.T @ (x - y))),
        x >= 0,
        y >= 0,
        x <= 1,
        y <= 1
    ]

    # defining the problem
    prob = cp.Problem(obj, constraints)

    # solving the problem
    prob.solve()

    # getting the Lipschitz constant
    L = prob.value

    # returning the Lipschitz constant
    return L
```

In [7]:
```
# test case 1
W = np.array([[1, 0],[0, 1]]) # weight matrix

lipschitz_constant = find_l_2(W)
print(f"Lipschitz constant: {lipschitz_constant}")
```

```
Lipschitz constant: -1.9386620890319104e-09
```

In [8]:
```
# test case 2
W = np.array([[62, 10],[-19, 10]]) #weight matrix

lipschitz_constant = find_l_2(W)
print(f"Lipschitz constant: {lipschitz_constant}")
```

```
Lipschitz constant: -2.48541684482281e-10
```

In [9]:

```
# test case 3
W = np.array([[1020, 35783],[3, 371]]) #weight matrix

lipschitz_constant = find_l_2(W)
print(f"Lipschitz constant: {lipschitz_constant}")
```

```
Lipschitz constant: -3.508112080076276e-10
```