Homework V Report

Name: Dazhi Li
RUID:197007456
Course: Sp.20 Software Engineering Web Application
Professor: Yinglung Liang

Question 1

Source code:

```
# -*- coding: utf-8 -*-


import numpy as np
#import matplotlib.pyplot as plt



def sigmoid(x):
   return 1/(1+np.exp(-x))


def s_prime(z):
   return np.multiply(z, 1.0-z)


def init_weights(layers, epsilon):
   weights = []
   for i in range(len(layers)-1):
      w = np.random.rand(layers[i+1], layers[i]+1)
      w = w * 2*epsilon - epsilon
      weights.append(np.mat(w))
   return weights


def fit(X, Y, w):
   # now each para has a grad equals to 0
   w_grad = ([np.mat(np.zeros(np.shape(w[i])))
           for i in range(len(w))])  # len(w) equals the layer number
   m, n = X.shape
   h_total = np.zeros((m, 1))  # 所有样本的预测值, m*1, probability
   for i in range(m):
      x = X[i]
      y = Y[0,i]
      # forward propagate
      a = x
      a_s = []
      for j in range(len(w)):
         a = np.mat(np.append(1, a)).T
```

```
            a_s.append(a)  # 这里保存了前L-1层的a值
            z = w[j] * a
            a = sigmoid(z)
        h_total[i, 0] = a
        # back propagate
        delta = a - y.T
        w_grad[-1] += delta * a_s[-1].T  # L-1层的梯度
        # 倒过来，从倒数第二层开始到第二层结束，不包括第一层和最后一层
        for j in reversed(range(1, len(w))):
            delta = np.multiply(w[j].T*delta, s_prime(a_s[j]))  # 这里传递的参数是a，而不是z
            w_grad[j-1] += (delta[1:] * a_s[j-1].T)
    w_grad = [w_grad[i]/m for i in range(len(w))]
    J = (1.0 / m) * np.sum(-Y * np.log(h_total) - (np.array([[1]]) - Y) * np.log(1 - h_total))
    return {'w_grad': w_grad, 'J': J, 'h': h_total}
    #w_grad return gradient, j return cost, h_total return predict_result


def error_cal(y_true,y_pre):
    return (np.power(y_true-y_pre.reshape(1,4),2)).sum()/4


def main():
    X = np.mat([[0,0],
             [0,1],
             [1,0],
             [1,1]])
    Y = np.mat([0,1,1,0])
    layers = [2,2,1]
    #error rate
    e=0.001
    #learning rate
    lr = 0.5
    w = init_weights(layers, 1)
    print("The initial weights are \n"+str(w))
    result = {'J': [], 'h': []}
    error_rate=1
    errors=[]
    while(error_rate>e):
        fit_result = fit(X, Y, w)
        w_grad = fit_result.get('w_grad')
        J = fit_result.get('J')
        h_current = fit_result.get('h')
        error_rate=error_cal(Y,h_current)
        errors.append(error_rate)
        result['J'].append(J)
        result['h'].append(h_current)
```

```
        for j in range(len(w)):
            w[j] -= lr * w_grad[j]
    print("The final weights are \n"+str(w))

    print("\nThe first batch error is: "+str(errors[0]))
    print("The final batch error is: "+str(errors[-1]))
    print("\nThe number of batches are: "+str(len(result['J']))+"\n")
    # plt.plot(result.get('J'))
    # plt.show()
    print(result.get('h')[0])
    print(result.get('h')[-1])

if __name__=="__main__":
    main()
```

Running result:

Changing learning rate

```
The initial weights are
[matrix([[-0.98194159,  0.82275552,  0.1315229 ],
        [ 0.35865445, -0.48369968,  0.44818687]]), matrix([[ 0.07679642,  0.93179079, -0.77369805]])]
The final weights are
[matrix([[-5.18672   ,  6.74205756, -7.21496421],
        [ 0.26857745, -2.94645589,  1.64315518]]), matrix([[-2.04741226,  6.78854881,  2.4091137 ]])]

The first batch error is: 0.2511032519539631
The final batch error is: 0.09998193548809717

The number of batches are: 2554

[[0.46884747]
 [0.45530975]
 [0.53568777]
 [0.52184456]]
[[0.34499453]
 [0.5100644 ]
 [0.97622645]
 [0.20075966]]

Process finished with exit code 0
```

Fig 1-1 Learning rate=0.5, target error=0.1

```
The initial weights are
[matrix([[-0.63120543,  0.50157731, -0.17654968],
        [ 0.91419564, -0.89062192, -0.05418265]]), matrix([[ 0.38163574, -0.55742933, -0.45640915]])]
The final weights are
[matrix([[ 0.48217791,  2.66319157, -1.82942516],
        [ 2.99343509, -4.87111184,  5.00604792]]), matrix([[ 4.55686872, -2.65906149, -3.85878273]])]

The first batch error is: 0.2510081954042159
The final batch error is: 0.09994662608034956

The number of batches are: 5496

[[0.46562039]
 [0.47228837]
 [0.47254419]
 [0.48014711]]
[[0.31850004]
 [0.53725538]
 [0.8169319 ]
 [0.22516153]]

Process finished with exit code 0
```

Fig 1-2 Learning rate=0.1, target error=0.1

```
The initial weights are
[matrix([[-0.41101354,  0.58109597, -0.92265163],
        [ 0.7556794 ,  0.68976771, -0.1151709 ]]), matrix([[0.22795061, 0.43408902, 0.0792987 ]])]
The final weights are
[matrix([[-2.18216721,  3.41278553, -3.74333927],
        [ 1.19266506,  2.55074989, -2.34560142]]), matrix([[ 0.94480788,  3.9688842 , -2.79839578]])]

The first batch error is: 0.2614601205022659
The final batch error is: 0.09999405192391432

The number of batches are: 25699

[[0.61181894]
 [0.59155846]
 [0.6289196 ]
 [0.60579939]]
[[0.31008426]
 [0.57046172]
 [0.78289818]
 [0.26867755]]

Process finished with exit code 0
```

Fig 1-3 Learning rate=0.01, target error=0.1

Changing target error

```
The initial weights are
[matrix([[ 0.55490093, -0.15595622,  0.23157222],
        [ 0.47139592, -0.61311369, -0.84631309]]), matrix([[ 0.87197564, -0.36914317,  0.60603011]])]
The final weights are
[matrix([[ 3.06643074, -5.20501112,  5.14923974],
        [-1.54324481, -3.69404597,  3.16031205]]), matrix([[ 2.61879717, -5.53842914,  5.36364307]])]

The first batch error is: 0.2949334088883934
The final batch error is: 0.019834085321984746

The number of batches are: 1255

[[0.73313964]
 [0.70374918]
 [0.71761208]
 [0.68900829]]
[[0.15166315]
 [0.8247898 ]
 [0.88694402]
 [0.11337707]]

Process finished with exit code 0
```

Fig 2-1 Learning rate=0.5, target error=0.02

```
The initial weights are
[matrix([[ 0.16766631,  0.89931247,  0.10622387],
        [-0.98838594, -0.21484535,  0.31882199]]), matrix([[0.68499841, 0.16732353, 0.91505002]])]
The final weights are
[matrix([[-2.76260211,  5.01545667, -5.34068317],
        [-3.40481426, -6.11254583,  5.98281193]]), matrix([[-4.017493  ,  8.25569708,  8.21156562]])]

The first batch error is: 0.3080213284556741
The final batch error is: 0.0009972236775223694

The number of batches are: 1516

[[0.73572134]
 [0.74835857]
 [0.73512385]
 [0.74653643]]
[[0.0369083 ]
 [0.97380537]
 [0.96928872]
 [0.03158055]]

Process finished with exit code 0
```

Fig 2-2 Learning rate=0.5, target error=0.001

Question 2

Source code:

Html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>Homework5</title>
   <script src="calculate.js"></script>
</head>
<body style="background-color:lightgray">
<h1 style="text-align:center">This is a website which will calculate the volume of objects</h1>
<hr/>
<h3>select the units first</h3>
<input name="unit" type="radio" value="0" onclick="getvalue(this.value)">English
<input name="unit" type="radio" value="1" onclick="getvalue(this.value)">SI unit
<p style="font-family:arial;color:red;font-size:20px;">A paragraph.</p>
<form>
   PLase select a shape:
<select id="shape">
   <option value="0" selected="selected">-none-</option>
  <option value ="1">Cylinder</option>
  <option value ="2">Cone</option>
  <option value="3">Sphere</option>
</select>
</form>
<br/>
<form action="">
 Enter the radius:<br>
<input id="radius" type="text" name="radius">
<br>
 For the Cylinder and Cone, please enter the height:<br>
<input id="height" type="text" name="height">
</form>
<br/>
<button type="button" onclick="calculate()">Click To Calculate!</button>
<br/>
<hr/>
<h2>Result</h2>
<p id="phase_1" style="font-family:arial;color:balck;font-size:16px;">You choose to use units</p>
<p id="phase_2" style="font-family:arial;color:balck;font-size:16px;">You choose to find the volume of </p>
<br/>
<table border="1">
   <tr>
```

```
      <th>Object Volume</th>
   </tr>
   <tr>
      <td>Shape</td>
      <td>Radius</td>
      <td>Height</td>
      <td>Volume</td>
   </tr>
   <tr>
      <td> </td>
      <td id="2-2">(ft)</td>
      <td id="2-3">(ft)</td>
      <td id="2-4">(ft^3)</td>
   </tr>
   <tr>
      <td id="3-1">object</td>
      <td id="3-2">num1</td>
      <td id="3-3">num2</td>
      <td id="3-4">num3</td>
   </tr>
</table>
</body>
</html>
```

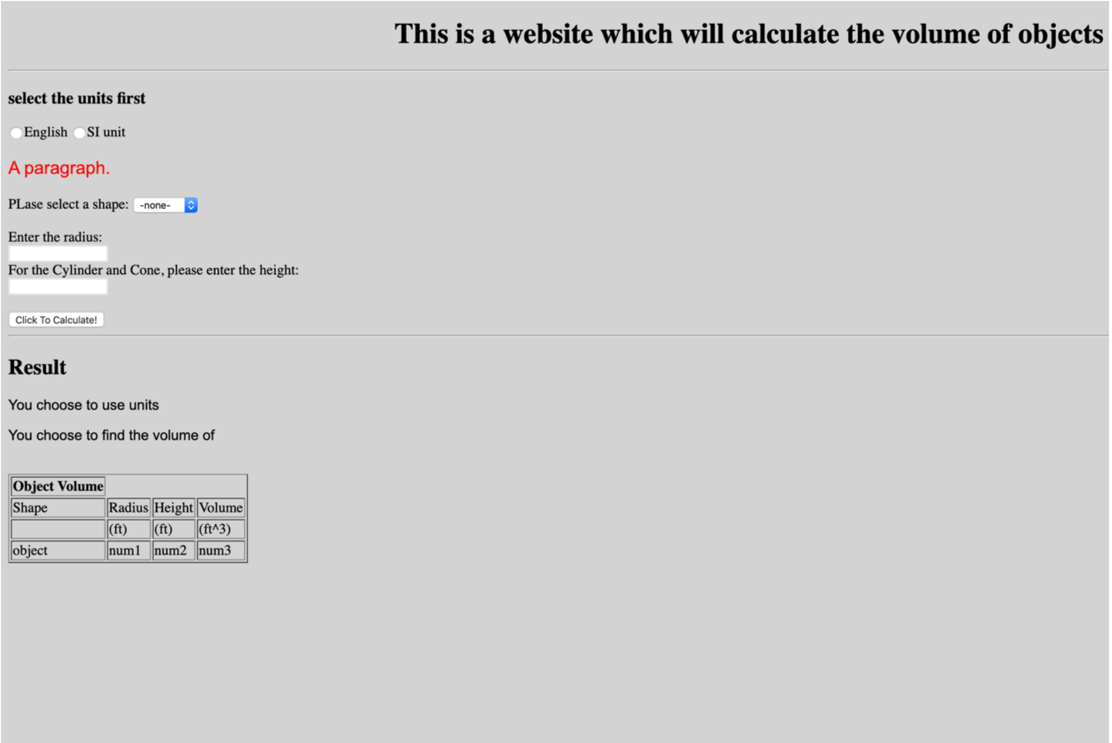
JavaScript

```
var unit
function calculate()
{
   console.log(unit);
   var shape=document.getElementById("shape");
   console.log(shape.value);
   var r=document.getElementById("radius");
   console.log(r.value);
   var h=document.getElementById("height");
   console.log(h.value);
   var v;
   var pi=3.1415926;
   if (unit==0)
   {
      document.getElementById("phase_1").innerHTML = "You choose to use English units";
      document.getElementById("2-2").innerHTML = "(ft)";
      document.getElementById("2-3").innerHTML = "(ft)";
      document.getElementById("2-4").innerHTML = "(ft^3)";
   }
   else if (unit==1)
```

```
    {
        document.getElementById("phase_1").innerHTML = "You choose to use SI units";
        document.getElementById("2-2").innerHTML = "(m)";
        document.getElementById("2-3").innerHTML = "(m)";
        document.getElementById("2-4").innerHTML = "(m^3)";
    }
    else
    {
        alert("Please choose unit first");
        location.reload();
    }
    switch(Number(shape.value))
    {
        case 1:
            v=pi*Math.pow(r.value,2)*h.value;
            document.getElementById("phase_2").innerHTML = "You choose to find the volume of Cylinder";
            document.getElementById("3-1").innerHTML = "Cylinder";
            document.getElementById("3-2").innerHTML = r.value;
            document.getElementById("3-3").innerHTML = h.value;
            document.getElementById("3-4").innerHTML = v;
            break;
        case 2:
            v=pi*Math.pow(r.value,2)*h.value/3;
            document.getElementById("phase_2").innerHTML = "You choose to find the volume of Cone";
            document.getElementById("3-1").innerHTML = "Cone";
            document.getElementById("3-2").innerHTML = r.value;
            document.getElementById("3-3").innerHTML = h.value;
            document.getElementById("3-4").innerHTML = v;
            break;
        case 3:
            v=4/3*pi*Math.pow(r.value,2);
            document.getElementById("phase_2").innerHTML = "You choose to find the volume of Sphere";
            document.getElementById("3-1").innerHTML = "Sphere";
            document.getElementById("3-2").innerHTML = r.value;
            document.getElementById("3-3").innerHTML = " ";
            document.getElementById("3-4").innerHTML = v;
            break;
        default:
            alert("Please choose shape ");
            //location.reload();
    }
}
function getvalue(value)
{
```

```
  unit=value
}
```

Web page result:



This is a website which will calculate the volume of objects

select the units first

English SI unit

A paragraph.

PLase select a shape: -none-

Enter the radius:

For the Cylinder and Cone, please enter the height:

Click To Calculate!

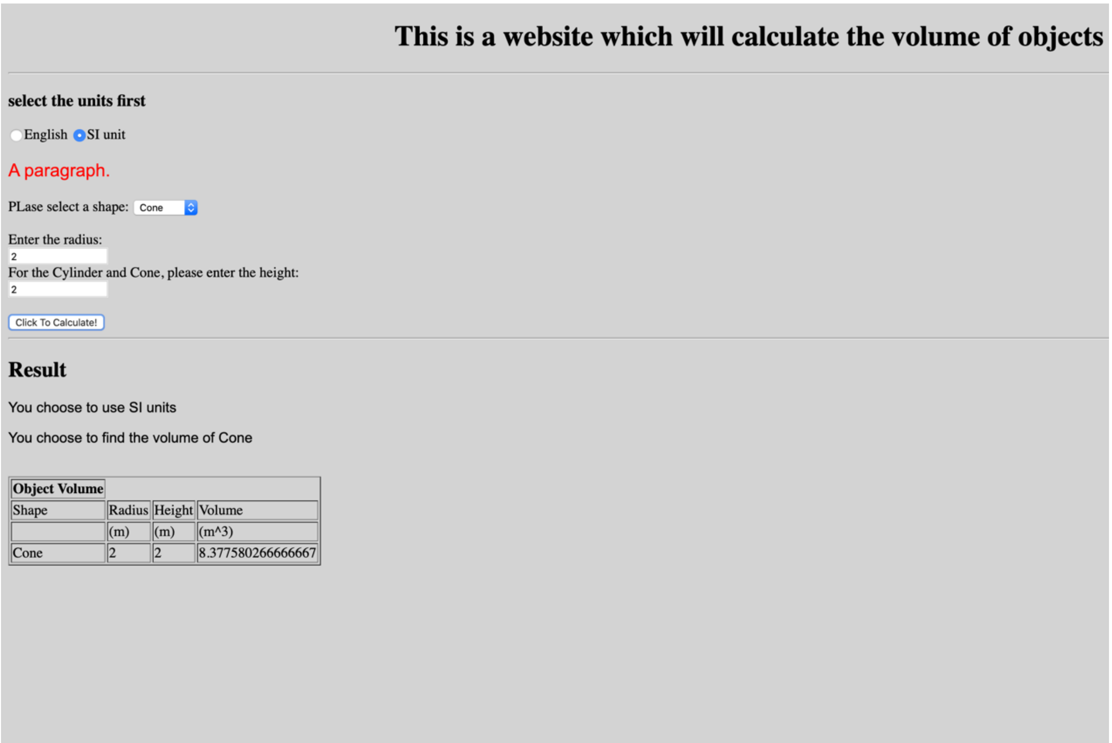
Result

You choose to use units

You choose to find the volume of

Object Volume			
Shape	Radius	Height	Volume
	(ft)	(ft)	(ft^3)
object	num1	num2	num3

Fig 3-1 Initial page



This is a website which will calculate the volume of objects

select the units first

English SI unit

A paragraph.

PLase select a shape: Cone

Enter the radius:
2
For the Cylinder and Cone, please enter the height:
2

Click To Calculate!

Result

You choose to use SI units

You choose to find the volume of Cone

Object Volume			
Shape	Radius	Height	Volume
	(m)	(m)	(m^3)
Cone	2	2	8.377580266666667

Fig 3-2 Calculated page