Problem Set #1

Joe Michail
DATA SCI 423 – Machine Learning
Northwestern University

April 14, 2020

Question 3.1

1. $g(w) = w \log w + (1-w)\log(1-w), w \in (0,1)$

$$\begin{aligned}\frac{dg}{dw} &= \log w \,\frac{dw}{dw} + w\frac{d\log w}{dw} + \log(1-w)\frac{d\ (1-w)}{dw} + (1-w)\frac{d\ \log(1-w)}{dw}\\ &= \log w + \frac{w}{w} + \log(1-w)\cdot -1 + \frac{(1-w)}{(1-w)}\cdot -1\\ &= \log w - \log(1-w)\end{aligned}$$

Stationary points occur when $dg/dw = 0$:

$$\begin{aligned}0 &= \log \frac{w}{1-w}\\ 1 &= \frac{w}{1-w}\end{aligned}$$

$$\boxed{w = 1/2}$$

The plot (at end of the homework) shows that this point is a minimum.

2. $g(w) = \log(1 + e^w)$

$$\begin{aligned}\frac{dg}{dw} &= \frac{\log(1+e^w)}{d(1+e^w)}\frac{d(1+e^w)}{dw}\\ &= \frac{e^w}{(1+e^w)}\\ 0 &= e^w\end{aligned}$$

$$\boxed{w = -\infty}$$

The plot shows that the derivative function tending toward $w = -\infty$ is a minimum.

3. $g(w) = w \tanh w$

$$\begin{aligned}\frac{dg}{dw} &= \tanh w \ \frac{dw}{dw} + w \ \frac{d \tanh w}{dw} \\ &= \tanh w + w \ \operatorname{sech}^2 w \\ 0 &= \sinh w \cosh w + w \\ &= \frac{1}{2} \sinh 2w + w\end{aligned}$$

$$\boxed{w = 0}$$

The plot shows that this stationary point is a minimum.

4. $g(w) = \dfrac{1}{2}\mathbf{w}^T\mathbf{C}\mathbf{w} + \mathbf{b}^T\mathbf{w}$

$$\begin{aligned}\nabla_w g(w) &= \frac{1}{2}\nabla(\mathbf{w}^T\mathbf{C}\mathbf{w}) + \nabla(\mathbf{b}^T\mathbf{w}) \\ &= \frac{1}{2}(\mathbf{C} + \mathbf{C}^T)\mathbf{w} + \mathbf{b}, \qquad \mathbf{C}^T = \mathbf{C} \\ \mathbf{0} &= \mathbf{C}\mathbf{w} + \mathbf{b} \\ -\mathbf{b} &= \mathbf{C}\mathbf{w}\end{aligned}$$

The problem states that:

$$\mathbf{C} = \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

$$\boxed{(w_1 \quad w_2)^T = (-2/5 \quad -1/5)^T}$$

The plot shows this stationary point is a minimum.

Question 3.3

Start with Rayleigh's quotient:

$$g(\mathbf{w}) = \frac{\mathbf{w}^T\mathbf{C}\mathbf{w}}{\mathbf{w}^T\mathbf{w}}.$$

Taking the gradient of the numerator and denominators separately:

$$\begin{aligned}\nabla(\mathbf{w}^T\mathbf{C}\mathbf{w}) &= (\mathbf{C} + \mathbf{C}^T)\mathbf{w}, \\ \nabla(\mathbf{w}^T\mathbf{w}) &= 2\mathbf{w}.\end{aligned}$$

Now taking the gradient of g:

$$\begin{aligned}\nabla g(\mathbf{w}) &= \frac{\nabla(\mathbf{w}^T\mathbf{C}\mathbf{w})\mathbf{w}^T\mathbf{w} - \nabla(\mathbf{w}^T\mathbf{w})\mathbf{w}^T\mathbf{C}\mathbf{w}}{(\mathbf{w}^T\mathbf{w})^2} \\ &= \frac{(\mathbf{C}+\mathbf{C}^T)\mathbf{w}\mathbf{w}^T\mathbf{w} - 2\mathbf{w}\mathbf{w}^T\mathbf{C}\mathbf{w}}{||\mathbf{w}||_2^4}.\end{aligned}$$

The stationary points occur when $\nabla g = \mathbf{0}$:

$$\mathbf{0} = (\mathbf{C}+\mathbf{C}^T)\mathbf{w}(\mathbf{w}^T\mathbf{w}) - 2\mathbf{w}(\mathbf{w}^T\mathbf{C}\mathbf{w}).$$

Since $\mathbf{w}^T\mathbf{w} \in \mathbb{R}$, it can freely be divided:

$$\begin{aligned}\mathbf{0} &= (\mathbf{C}+\mathbf{C}^T)\mathbf{w} - 2\mathbf{w}\frac{\mathbf{w}^T\mathbf{C}\mathbf{w}}{\mathbf{w}^T\mathbf{w}} \\ &= (\mathbf{C}+\mathbf{C}^T)\mathbf{w} - 2g(\mathbf{w})\mathbf{w} \\ \\ &= (\mathbf{C}+\mathbf{C}^T - 2g(\mathbf{w})\mathbf{I})\mathbf{w} \qquad (1)\end{aligned}$$

There, the last equation is multiplied by the identity matrix $\mathbf{I}$ to make the sum inside of the parenthesis a matrix. This is the classic eigenvalue problem which can more easily been shown if $\mathbf{C} = \mathbf{C}^T$:

$$\mathbf{0} = (2\mathbf{C} - 2g(\mathbf{w})\mathbf{I})\mathbf{w} \implies \det(\mathbf{C} - 2g(\mathbf{w})\mathbf{I}) = 0.$$

Consider $\mathbf{C} \in \mathbb{R}^{N\times N}$. This implies that $|\mathbf{C}+\mathbf{C}^T - 2g(\mathbf{w})\mathbf{I}| = 0$ can be solved to determine N eigenvalues, each of which correspond to one eigenvectors (N total eigenvectors). *Therefore, the stationary points of Rayleigh's quotient correspond to the N eigenvectors of the matrix $\mathbf{C}$.*

Question 3.5

Here we start with the function:

$$g(w) = \frac{w^4 + w^2 + 10w}{50}$$

The derivative of this is trivial:

$$\frac{dg}{dw} = \frac{4w^3 + 2w + 10}{50}$$

The cost function plots for this problem are at the end of this document. With this combination of step length and initial position, $\alpha = 1$ converges the quickest to the minimum $w \sim -1.234, g(w) \sim -0.17$.

Question 3.6

See end of document for plot. Please note that the fixed-step line stops at $k = 4$ where it encounters the minimum which is a non-differentiable point.

Question 3.8

The cost function we're trying to minimize is:

$$g(\mathbf{w}) = \mathbf{w}^T \mathbf{w}$$

and has a gradient of the form:

$$\nabla_{\mathbf{w}} g(\mathbf{w}) = 2\mathbf{w}.$$

The cost function history plots are below and a step length of $\alpha = 0.1$ converges quickest to the minimum located at $\mathbf{w} = \mathbf{0}$.

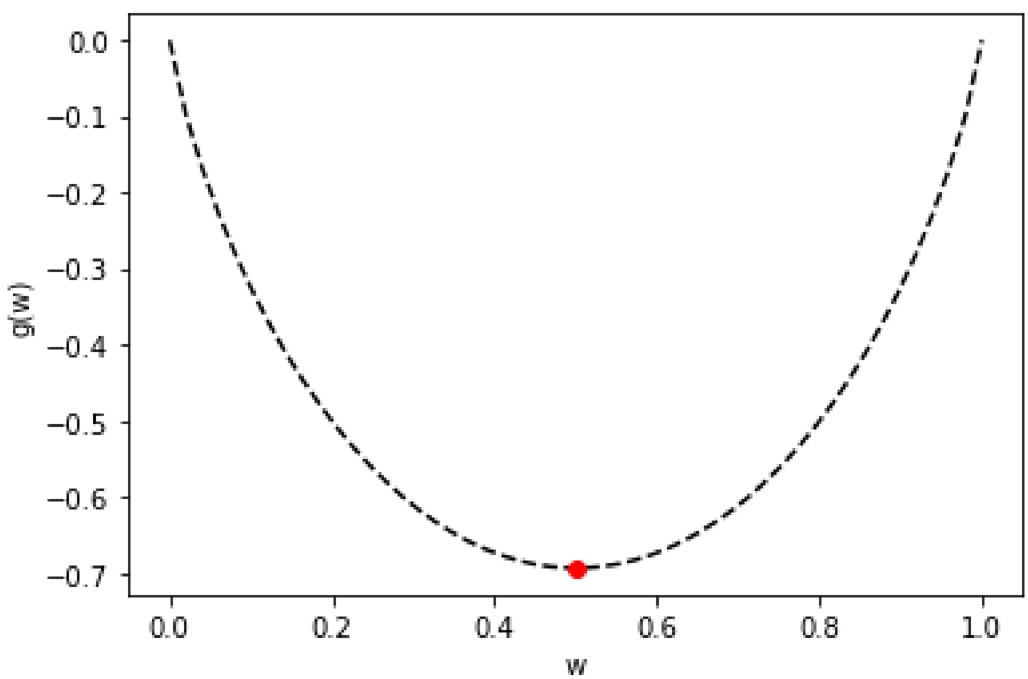
HW 1

April 14, 2020

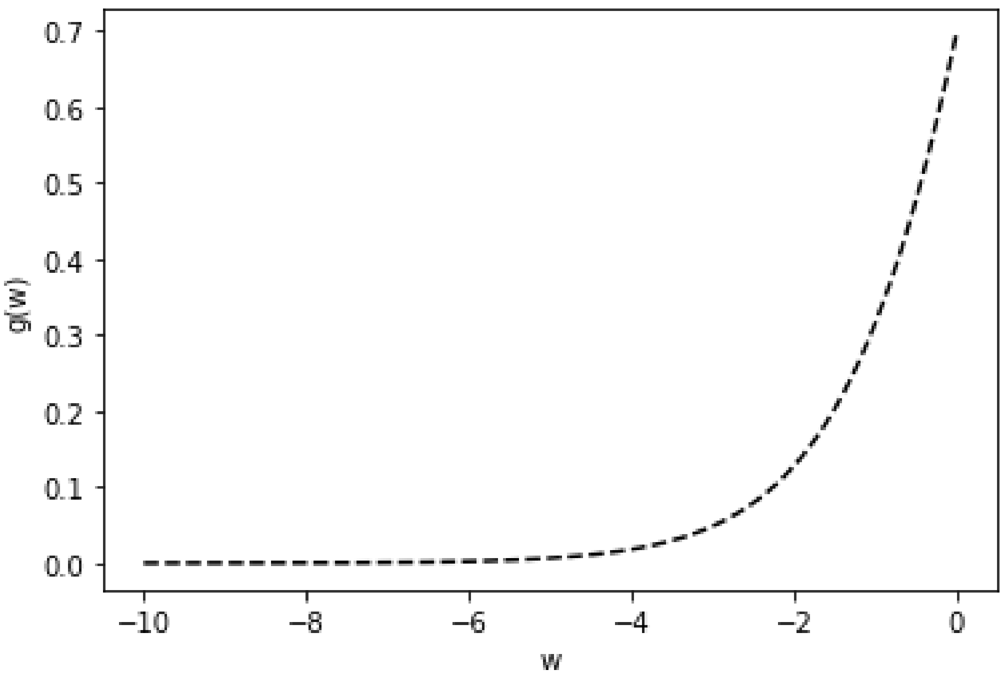
[1]:
```
#Import modules
import numpy as np
import matplotlib
matplotlib.rcParams['font.size'] = 16
import matplotlib.pyplot as plt
```

1 Problem 3.1

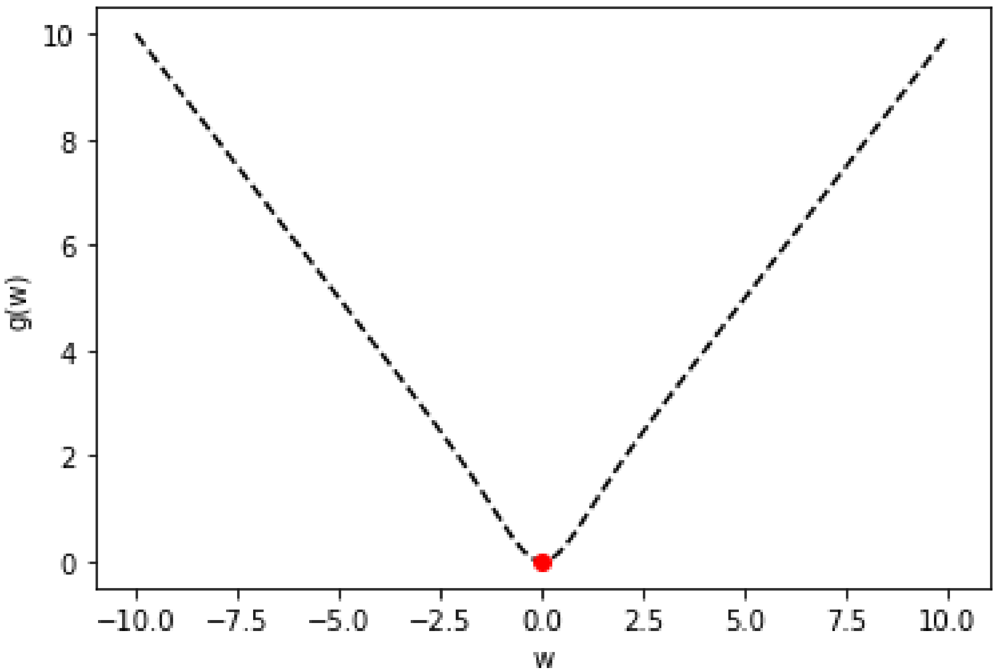
[2]:
```
#Part A
xs = np.linspace(0.0001, 0.9999)

plt.plot(xs, xs * np.log(xs) + (1 - xs) * np.log(1 - xs), 'k--')
plt.plot(0.5, np.log(0.5), 'ro')
plt.xlabel("w")
plt.ylabel("g(w)")
plt.show()
```



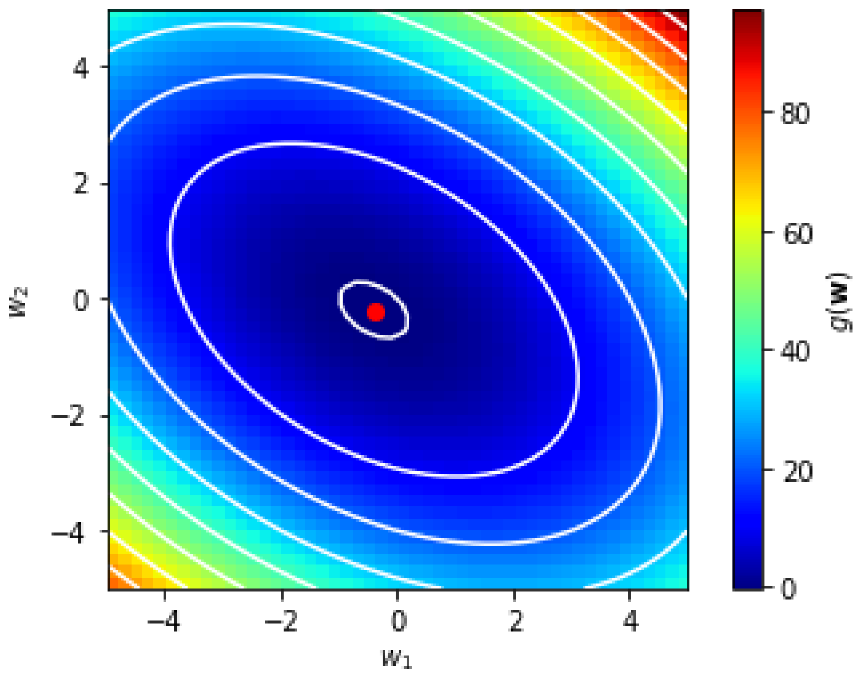
[3]:
```
#Part B: Minimum is at -infinity
xs = np.linspace(-10, 0, num=10000)

plt.plot(xs, np.log(1 + np.exp(xs)), 'k--')
plt.xlabel("w")
plt.ylabel("g(w)")
plt.show()
```



[4]:
```
#Part C
xs = np.linspace(-10, 10, num=10000)

plt.plot(xs, xs * np.tanh(xs), 'k--')
plt.plot(0, 0, 'ro')
plt.xlabel("w")
plt.ylabel("g(w)")
plt.show()
```



[5]:
```
#Part D
def g(x, y):
    C = np.matrix('2 1; 1 3')
    w = np.matrix([[x], [y]])
    b = np.matrix('1; 1')
    return np.sum(0.5 * w.T*C*w + b.T * w)

xs, ys = np.linspace(-5.0, 5.0), np.linspace(-5.0, 5.0)
xm, ym = np.meshgrid(xs, ys)

gs = np.zeros((xs.size, xs.size))
for i, x in enumerate(xs):
    for j, y in enumerate(ys):
        gs[j, i] = g(x, y)

plt.imshow(gs, origin='lower', cmap='jet', extent=[-5.0, 5.0, -5.0, 5.0])
cbar = plt.colorbar()
plt.contour(xm, ym, gs, levels=10, colors='white')
cbar.set_label("$g(\\mathbf{w})$")
plt.plot(-2/5, -1/5, 'ro')
plt.xlabel("$w_1$")
plt.ylabel("$w_2$")
plt.show()
```

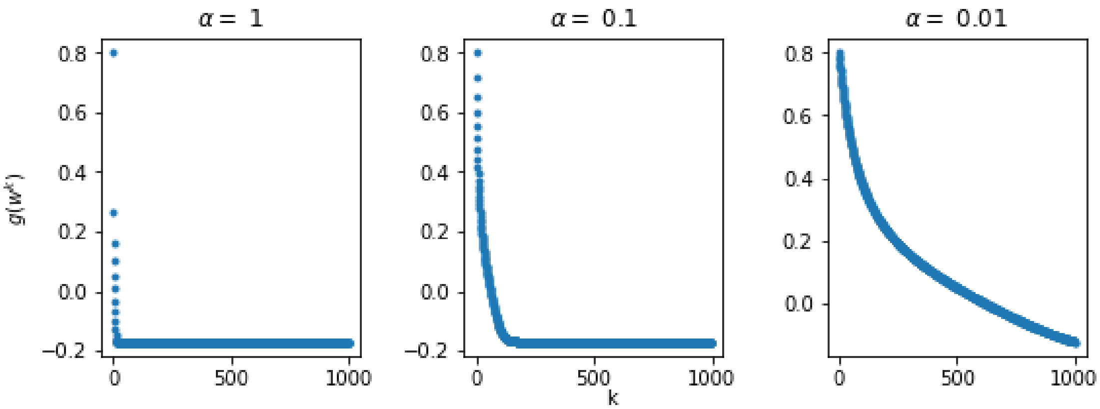


2 Problem 3.5

[6]:
```
def g(w):
    return (w**4.0 + w**2.0 + 10 * w) / 50.0

def grad_g(w):
    return (4 * w**3.0 + 2 * w + 10) / 50.0

def grad_descent(w0, alpha, n_iter):
    ws = np.array([])
    for i in range(n_iter):
        ws = np.append(ws, w0)
        w0 = w0 - alpha * grad_g(w0)
    return ws

fig, ax = plt.subplots(1, 3, figsize=(9, 3))
ax[0].plot(np.arange(1000), g(grad_descent(2, 1, 1000)), '.')
ax[0].set_ylabel("$g(w^k)$")
ax[0].set_title("$\\alpha = $ 1")

ax[1].plot(np.arange(1000), g(grad_descent(2, 0.1, 1000)), '.')
ax[1].set_title("$\\alpha = $ 0.1")

ax[2].plot(np.arange(1000), g(grad_descent(2, 0.01, 1000)), '.')
```

```
ax[2].set_title("$\\alpha = $ 0.01")

plt.subplots_adjust(left=0.1, right=0.88, wspace=0.4)
fig.text(0.5, 0.01, "k")
```

[6]: Text(0.5, 0.01, 'k')

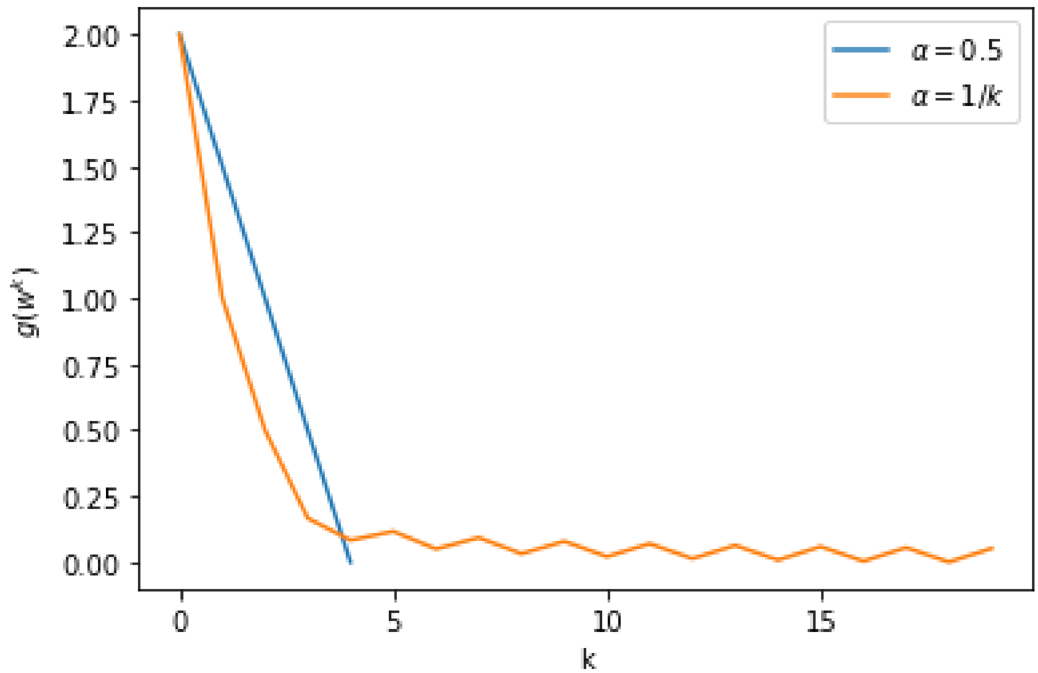


3 Problem 3.6

[7]:
```
def g(w):
    return np.abs(w)

def grad_g(w):
    if w < 0:
        return -1.0
    elif w > 0:
        return 1.0
    else:
        return np.nan #Stop if we hit a non-differentiable point

# Diminshing step length
def grad_descent_diminish(w0, n_iter):
    ws = []
    for i in range(1, n_iter+1):
        ws.append(w0)
        w0 = w0 - (1 / i) * grad_g(w0)
    return ws

def grad_descent(w0, alpha, n_iter):
    ws = []
    for i in range(n_iter):
```

```
        ws.append(w0)
        w0 = w0 - alpha * grad_g(w0)
    return ws

plt.plot(g(grad_descent(2.0, 0.5, 20)), label="$\\alpha = 0.5$")
plt.plot(g(grad_descent_diminish(2, 20)), label="$\\alpha = 1/k$")
plt.xlabel("k")
plt.ylabel("$g(w^k)$")
plt.xticks(np.arange(0, 20, 5))
plt.legend()
```

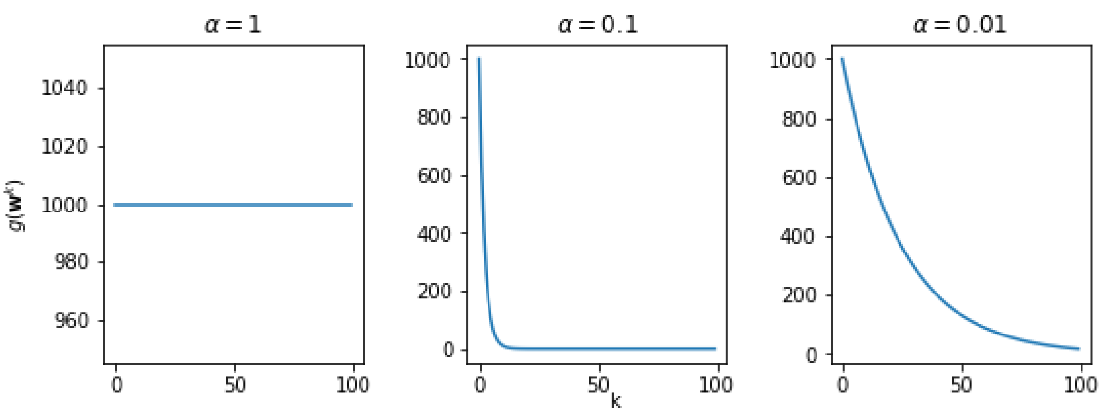
[7]: <matplotlib.legend.Legend at 0x2bc9a7e2608>



4 Problem 3.8

[8]:
```
def g(w):
    return np.sum(w**2.0)

def grad_w(w):
    return 2 * w

def grad_descent(w0, alpha, n_iter):
    ws = []
    for i in range(n_iter):
```

```
        ws.append(g(w0))
        w0 = w0 - alpha * grad_w(w0)
    return ws

fig, ax = plt.subplots(1, 3, figsize=(9, 3))
ax[0].plot(grad_descent(10*np.ones((10, 1)), 1.0, 100))
ax[0].set_ylabel("$g(\\mathbf{w}^k)$")
ax[0].set_title("$\\alpha = 1$")

ax[1].plot(grad_descent(10*np.ones((10, 1)), 0.1, 100))
ax[1].set_title("$\\alpha = 0.1$")

ax[2].plot(grad_descent(10*np.ones((10, 1)), 0.01, 100))
ax[2].set_title("$\\alpha = 0.01$")

fig.text(0.5, 0.02, "k")

plt.subplots_adjust(left=0.1, right=0.88, wspace=0.4)
```



[ ]:

```
```