1.

(a).

$$L = \sum_t \log P(y_t | \vec{x}_t)$$

$$= \sum_t \log \left[p_t^{y_t} (1-p_t)^{1-y_t} \right]$$

$$= \sum_t \left[y_t \log p_t + (1-y_t) \log (1-p_t) \right]$$

$$\frac{\partial L}{\partial w_i} = \sum_t \left[\frac{y_t}{p_t} \frac{\partial p_t}{\partial w_i} + \frac{(1-y_t)}{(1-p_t)} \frac{\partial (1-p_t)}{\partial w_i} \right]$$

$$= \sum_t \left[\frac{y_t}{p_t} \frac{\partial p_t}{\partial w_i} - \frac{(1-y_t)}{(1-p_t)} \frac{\partial p_t}{\partial w_i} \right]$$

$$= \sum_t \left[\left(\frac{y_t}{p_t} - \frac{1-y_t}{1-p_t} \right) \cdot \frac{\partial p_t}{\partial w_i} \right]$$

$$\because \frac{\partial p_t}{\partial w_i} = \frac{\partial g(\vec{w}, \vec{x})}{\partial w_i, \partial x_i} \cdot \partial x_i = g'(\vec{w} \cdot \vec{x}_t) \cdot x_i.$$

$$\because \frac{y_t}{p_t} - \frac{1-y_t}{1-p_t} = \frac{y_t(1-p_t) - (1-y_t)p_t}{p_t(1-p_t)}$$

$$= \frac{y_t - y_t p_t - p_t + y_t p_t}{p_t(1-p_t)} = \frac{y_t - p_t}{p_t(1-p_t)}.$$

$$\therefore \frac{\partial L}{\partial w_i} = \sum_t \frac{y_t - p_t}{p_t(1-p_t)} \cdot g'(\vec{w} \cdot \vec{x}_i) \cdot x_{it}.$$

(b). $\because g(z) = [1+e^{-z}]^{-1}$

$\therefore g(z)' = g(z)g(-z)$; $g(-z) = 1-g(z)$

$\because p_t = P(Y=1|\vec{x_t}) = g(\vec{w}\cdot\vec{x})$

$1-p_t = g(-\vec{w}\vec{x})$.

$\therefore g(\vec{w}\vec{x})' = p_t(1-p_t)$.

$\therefore$

$$\frac{\partial L}{\partial w_i} = \sum_{t=1}^{T}\left[\frac{p_t(1-p_t)}{p_t(1-p_t)}\right]\cdot(y_t - p_t)x_{it}.$$

$$= \sum_{t=1}^{T}(y_t - g(\vec{w}\cdot\vec{x}))\cdot\vec{x_t}$$

2.

$L = \sum_t \log P(y_t|\vec{x_t})$.

as for $P(Y=i|\vec{x}) = \dfrac{e^{\vec{w_i}\vec{x_t}}}{\sum_j e^{\vec{w_j}\vec{x}}}$.

and there are C kinds of Y.

$$\therefore L = \sum_t \log\left(\frac{\prod_{i=1}^{C}(e^{\vec{w_i}\vec{x}})^{y_{it}}}{\sum e^{\vec{w_j}\vec{x}}}\right)$$

$$= \sum_t\left(\sum_{i=1}^{C}\vec{w_i}\vec{x}\cdot y_{it} - \log\sum e^{\vec{w_j}\vec{x}}\right)$$

$$\therefore \frac{\partial L}{\partial \vec{w}} = \sum\left[y_{it}\vec{x} - \frac{1}{\sum_{j=1} e^{\vec{w_j}\vec{x}}}\cdot\vec{x}\cdot e^{\vec{w_j}\vec{x}}\right]$$

$$= \sum_t[(y_t - p_{it})\cdot\vec{x}]$$

3.

(a) $\varepsilon_n = x_n - x_*$

$f'(x) = \alpha(x - x_*) = \alpha\varepsilon_n$

$\because$ $\varepsilon_n = x_n - x_*$, $x_n = x_{n-1} - \eta \cdot \alpha \varepsilon_{n-1}$

$\therefore$ $\varepsilon_n = x_{n-1} - \eta\alpha\varepsilon_{n-1} - x_*$

$= (x_{n-1} - x_*) - \eta\alpha\varepsilon_{n-1}$

$= \varepsilon_{n-1} - n\alpha\varepsilon_{n-1} = (1-n\alpha)\varepsilon_{n-1}$

$= (1-n\alpha)[(1-n\alpha)\varepsilon_{n-2}] \cdots$

$= (1-n\alpha)^n \varepsilon_0$

(b) we can see from part (a). $\varepsilon_n \propto \varepsilon_0$,

$\therefore$ if $|(1-n\alpha)| < 0$, then it converges.

as for $f''(x)$:

$f''(x) = \alpha$.

when $1-n\alpha = 0 \Rightarrow n = \frac{1}{\alpha}$, it converge fast.

$\therefore$ when $f''(x) = \alpha$ and $n = \frac{1}{\alpha}$, converge fast

(c) $\varepsilon_{n+1} = x_{n+1} - x_*$

$= x_n - \eta f'(x_n) + \beta(x_n - x_{n-1}) - x_*$

$= (1-\eta\alpha)(x_n - x_*) + \beta(x_n - x_{n-1})$

$= (1-\eta\alpha)\varepsilon_n + \beta((x_n - x_*) - (x_{n-1} - x_*))$

$= (1-\eta\alpha)\varepsilon_n + \beta(\varepsilon_n - \varepsilon_{n-1})$

$= (1-\eta\alpha+\beta)\varepsilon_n - \beta\varepsilon_{n-1}$

1d). $\alpha=1,\ \eta=\frac{4}{9},\ \beta=\frac{1}{9}$

$$\varepsilon_{n+1}=(1-\alpha\eta+\beta)\varepsilon_n-\beta\varepsilon_{n-1}$$
$$=\frac{6}{9}\varepsilon_n-\frac{1}{9}\varepsilon_{n-1}$$
$$=\frac{2}{3}\varepsilon_n-\frac{1}{9}\varepsilon_{n-1}$$

assume $\varepsilon_n=\lambda^n\varepsilon_0$.

$\therefore$

$$\lambda^{n+1}\varepsilon_0=\frac{2}{3}\lambda^n\varepsilon_0-\frac{1}{9}\lambda^{n-1}\varepsilon_0$$

$$\therefore 9\lambda^2-6\lambda+1=0$$
$$\lambda=\frac{1}{3}.$$

$$\therefore \varepsilon_n=\frac{1}{3}^n\varepsilon_0$$

And:

as for $\eta=\frac{4}{9},\ \beta=0,\ \varepsilon_n=(\frac{5}{9})^n\varepsilon_0$, which faster.

4

(a). $\varepsilon_n=|x_n-x_*|$

$$=\left|x_{n-1}-\frac{g'(x_{n-1})}{g''(x_{n-1})}-x_*\right|$$

$$=\left|x_{n-1}-\frac{\cancel{2k}(x_{n-1}-x_*)^{\cancel{2k-1}}}{\cancel{2k}\cdot(2k-1)\cdot\cancel{(x_{n-1}-x_*)^{2k-2}}}-x_*\right|$$

$$=\left|(1-\frac{1}{2k-1})(x_{n-1}-x_*)\right|$$

$$=(1-\frac{1}{2k-1})\cdot\varepsilon_{n-1}=(1-\frac{1}{2k-1})^n\varepsilon_0.$$

(b). $\because \varepsilon_n \le \delta \varepsilon_0$.

$\therefore \frac{\varepsilon_n}{\varepsilon_0} \le \delta$

$\therefore \left(\frac{2k-2}{2k-1}\right)^n \le \delta$

$\therefore n \log\left(\frac{2k-2}{2k-1}\right) \le \log(\delta)$

$\therefore n \left(\frac{2k-2}{2k-1} - 1\right) \le \log(\delta)$

$\therefore n \left(\frac{-1}{2k-1}\right) \le \log(\delta)$

$\therefore n \ge -(2k-1)\log(\delta) = (2k-1)\log\left(\frac{1}{\delta}\right)$

(c) $h(x) = x_* \log(x_*/x) - x_* + x$

$h'(x) = x_* \frac{d}{dx}\left(\log \frac{x_*}{x}\right) + 1$

$= x_* \cdot \frac{\cancel{x}}{\cancel{x_*}} \cdot \cancel{x_*} \cdot \frac{1}{-x^{\cancel{2}}} + 1$

$= 1 - \frac{x_*}{x}$

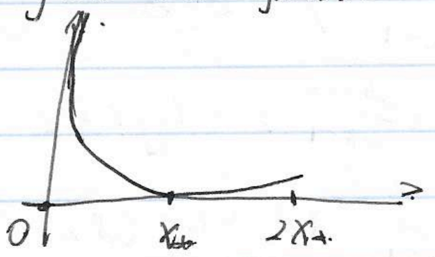
$h''(x) = -x_* \cdot -\frac{1}{x^2} = \frac{x_*}{x^2}$

$\therefore$ when $h'(x) = 0$, $h''(x) > 0$, get the minimum:

$1 - \frac{x_*}{x} = 0$, $x = x_*$

graph:

since x_* is any number > 0.
after pick up sereral number $x_* = 5, 100, 1000$,
and get the function plots should like.



(d). $x_{n+1} = x_n - \dfrac{g'(x_n)}{g''(x_n)}$

$$= x_n - \frac{1 - \frac{x_*}{x}}{\frac{x_*}{x^2}} = x_n - \frac{x - x_*}{x} \cdot \frac{x^2}{x_*}$$

$$= x_n - \frac{x_n(x_n - x_*)}{x_*} = \frac{x_n x_* - x_n^2 + x_n x_*}{x_*}$$

$$x_{n+1} = \frac{-x_n^2 + 2x_n x_* - x_*^2}{x_*} + \frac{x_*^2}{x_*}$$

$$\therefore x_{n+1} - x_* = -\frac{(x_n - x_*)^2}{x_*}$$

$$\therefore \frac{x_{n+1} - x_*}{x_*} = e_{n+1} = -\left(\frac{x_n - x_*}{x_*}\right)^2 = -e_n^2$$

$$\therefore e_{n+1} = -e_n^2 = -(e_0)^{2n}$$

$$\therefore |e_0| < 1 \Rightarrow 0 < x < 2x_*$$

4.5
(a)
[0.94520006 0.01974237 -0.01364498 0.04678134]

(b)
for 2000: MSE = 13918.63
for 2001: MSE = 2973.02

(c)
```
import numpy as np

def calculate(array):
	A = []
	B = []

	for i in range(4):
		B.append(float(0))
		A.append([])
		for j in range(4):
			A[i].append(float(0))

	for i in range(4):
		for j in range(4):
			cur = 0.0
			for k in range(4, len(array)):
				cur += array[k-j-1]*array[k-i-1]
			A[i][j] = cur

	for i in range(4):
		cur = 0.0
		for k in range(4,len(array)):
			cur += array[k] * array[k-i-1]
		B[i] = cur

	A = np.array(A,float)
	B = np.array(B,float)
	X = np.dot(np.linalg.inv(A) , B)
	return X

def error(array,X):
	Y = []
	for i in range(4, len(array)):
		cur = X[0]*array[i-1] + X[1]*array[i-2] + X[2]*array[i-3] + X[3]*array[i-4]
		Y.append(cur)

	e = 0.0
```

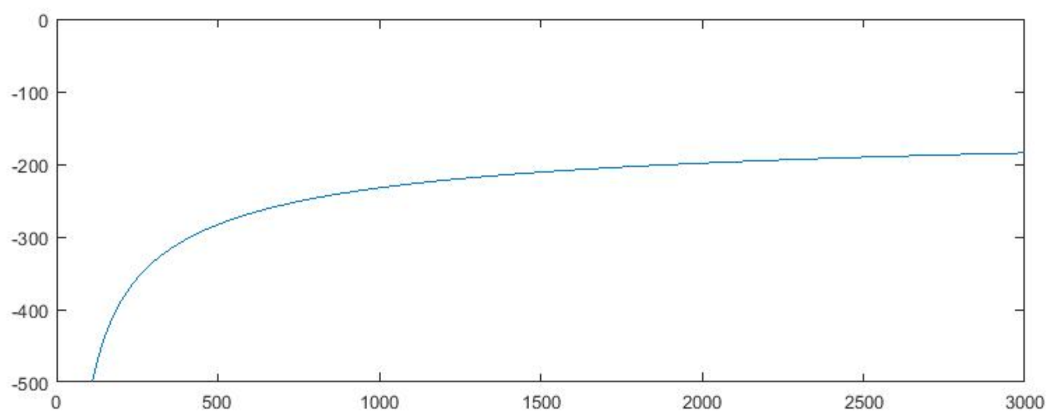
```
    for i in range(len(Y)):
        e += (Y[i] - array[i+4]) **2
    e /= len(Y)
    print e



fp = open('nasdaq00.txt', 'r')
#fp = open('nasdaq01.txt', 'r')
array2000 = []
for line in fp.readlines():
    line = line.strip('\n').split(' ')
    array2000.append(float(line[0]))

X = []
X = calculate(array2000)
print X

error(array2000,X)
```

4.6
(a)
error rate for train3: 3.57%
error rate for train5: 3.85%
error rate overall: 3.71%

w =

0.6971	-0.1080	-1.0490	-0.7983	-0.4133	-0.7576	-0.4498	-0.0479
0.8010	-0.0417	-1.1509	-0.9043	-0.3120	0.2457	-0.1742	-0.4225
0.9076	-0.3412	-0.7329	-0.6148	0.1957	0.1171	-0.3187	-0.3872
1.0184	0.1839	-0.3500	0.1273	0.2159	-0.3595	-0.3523	-0.6077
1.0018	-0.0668	-0.1303	0.5902	0.3105	-0.1145	-0.0138	-0.3400
-0.2049	-0.4751	1.3796	0.7316	0.2290	0.2519	0.1153	-0.5644
-0.9889	0.6730	2.5369	-0.1523	0.0097	-0.2046	0.0092	0.2550
-1.4022	0.6319	1.9222	0.2368	0.3302	0.8040	0.7367	0.1802

The converge of L using gradient:



(b)
error rate for test3: 5.25%
error rate for test5: 6.75%
error rate overall:6.00%

(c)

```
%0 stands for 5
%1 stands for 3
%sigmf(x,[1 0])
Train3=load('Train3.txt');
Train5=load('Train5.txt');
Test3 =load('Test3.txt');
Test5 =load('Test5.txt');

% initial values of weight.
w= 0.5 * ones(64,1);
L0= sum(log(sigmf(Train3 * w,[1 0])))+sum(log(sigmf(-Train5 * w,[1 0])));
Y1 = 1* ones(700,1);
Y2 = 0* ones(700,1);
Gradient= Train3' *(Y1 - sigmf(Train3*w,[1,0])) + Train5' *(Y2 - sigmf(Train5 *w,[1 0]));
Lr=ones(10000,1);

for i= 1:10000
   w=w+0.02/1400*Gradient;
   Gradient= Train3' *(Y1 - sigmf(Train3*w,[1,0])) + Train5' *(Y2 - sigmf(Train5 *w,[1 0]));
   Lr(i)=sum(log(sigmf(Train3 * w,[1 0])))+sum(log(sigmf(-Train5 * w,[1 0])));
end
sigmf(Train3 *w,[1 0])
L= sum(log(sigmf(Train3 * w,[1 0])))+sum(log(sigmf(-Train5 * w,[1 0])));

T3=sigmf(Train3 * w,[1 0]);
T5=sigmf(Train5 * w,[1 0]);
R3=sigmf(Test3 * w,[1 0]);
R5=sigmf(Test5 * w,[1 0]);
c3_5=0;
t3_5=0;
c5_3=0;
t5_3=0;
for i= 1:400
    if(R3(i)<=0.5)
        c3_5=c3_5+1;
    end
    if(R5(i)>0.5)
        c5_3=c5_3+1;
    end
end
for i= 1:700
    if(T3(i)<=0.5)
        t3_5=t3_5+1;
    end
    if(T5(i)>0.5)
```

```
        t5_3=t5_3+1;
    end
end
fprintf('Error rate of train3: %f\n',t3_5/700);
fprintf('Error rate of train5: %f\n',t5_3/700);
fprintf('Error rate overall:%f\n',t3_5/1400+t5_3/1400);
fprintf('Error rate of test3: %f\n',c3_5/400);
fprintf('Error rate of test5: %f\n',c5_3/400);
fprintf('Error test rate overall:%f\n',c3_5/800+c5_3/800);

x=1:10000;
y=Lr(x);
plot(x,y)

w= reshape(w, [8,8])
```