Στατιστική Μοντελοποίηση και Αναγνώριση Προτύπων - ΤΗΛ311

Αναφορά 2ης σειράς ασκήσεων
Σοφοκλής Φιλάρετος Γαβριηλίδης
Α.Μ. : 2014030062

Θέμα 1: Λογιστική Παλινδρόμηση: Αναλυτική εύρεση κλίσης (Gradient)

a)

Στατιστική Μοντελοποίηση και Αναγνώριση Προτύπων
2η Σειρά Ασκήσεων

Σοφοκλής Φιλάρετος Γαβριηλίδης
Α.Μ.: 2014030062

Θέμα 1: Λογιστική Παλινδρόμηση: Αναλυτική Εύρεση Κλίσης (Gradient)

α) Για την βελτιστοποίηση του σφάλματος χρειάζεται να υπολογίσουμε την κλίση του σφάλματος $J(\theta)$.

Συνάρτηση σφάλματος: $J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left(-y^{(i)}\ln(h_\theta(x^{(i)})) - (1-y^{(i)})\ln(1-h_\theta(x^{(i)}))\right)$

$\theta_j, x_j^{(i)}$ η j-συνιστώσα των διανυσμάτων $\theta = [\theta_1, \theta_2, \dots \theta_n]^T$ και $x^{(i)} = [x_1^{(i)}, x_2^{(i)}, \dots x_n^{(i)}]$

Κλίση για j: $\frac{dJ(\theta)}{d\theta_j} = \frac{1}{m}\frac{d}{d\theta_j}\sum_{i=1}^{m}\left(-y^{(i)}\ln(h_\theta(x^{(i)})) - (1-y^{(i)})\ln(1-h_\theta(x^{(i)}))\right) \Leftrightarrow$

$$\frac{dJ(\theta)}{d\theta_j} = \frac{1}{m}\sum_{i=1}^{m}\left(-y^{(i)}\underbrace{\frac{d}{d\theta_j}\ln(h_\theta(x^{(i)}))} - (1-y^{(i)})\underbrace{\frac{d}{d\theta_j}\ln(1-h_\theta(x^{(i)}))}\right) : (1)$$

- $\frac{d}{d\theta_j}\ln(h_\theta(x^{(i)})) = \frac{1}{h_\theta(x^{(i)})}\cdot\frac{d}{d\theta_j}h_\theta(x^{(i)})$ (2)
- $\frac{d}{d\theta_j}\ln(1-h_\theta(x^{(i)})) = \frac{1}{1-h_\theta(x^{(i)})}\cdot\frac{d}{d\theta_j}(1-h_\theta(x^{(i)})) = \frac{-1}{1-h_\theta(x^{(i)})}\frac{d}{d\theta_j}h_\theta(x^{(i)})$ (3)
- $\boxed{\frac{d}{d\theta_j}h_\theta(x^{(i)})} = \frac{d}{d\theta_j}\left(\frac{1}{1+e^{-\theta^T x^{(i)}}}\right) = \frac{-1}{(1+e^{-\theta^T x^{(i)}})^2}\frac{d}{d\theta_j}(1+e^{-\theta^T x^{(i)}}) =$

$= -h_\theta(x^{(i)})^2\, e^{-\theta^T x^{(i)}}\frac{d}{d\theta_j}(-\theta^T x^{(i)}) = -h_\theta(x^{(i)})^2(1+e^{-\theta^T x^{(i)}}-1)\frac{d}{d\theta_j}(-\theta^T x^{(i)}) =$

$= -h_\theta(x^{(i)})^2\left(\frac{1}{h_\theta(x^{(i)})}-1\right)\frac{d}{d\theta_j}\left(-(\theta_1 x_1^{(i)} + \theta_2 x_2^{(i)} + \dots + \theta_n x_n^{(i)})\right) =$

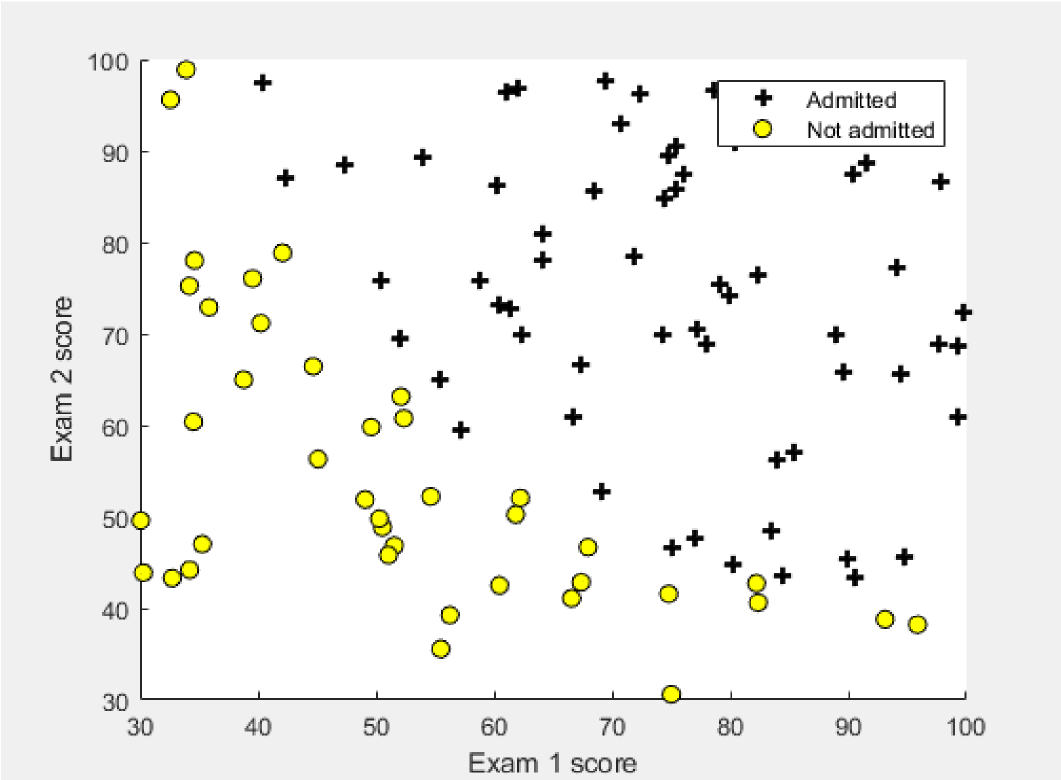
$= h_\theta(x^{(i)})^2\left(\frac{1}{h_\theta(x^{(i)})}-1\right)x_j^{(i)} = \boxed{h_\theta(x^{(i)})(1-h_\theta(x^{(i)}))x_j^{(i)}}$ (4)

(2) $\xrightarrow{(4)}$ $\boxed{\frac{d}{d\theta_j}\ln(h_\theta(x^{(i)})) = (1-h_\theta(x^{(i)}))x_j^{(i)}}$ (5)

(3) $\xrightarrow{(4)}$ $\boxed{\frac{d}{d\theta_j}\ln(1-h_\theta(x^{(i)})) = -1\cdot h_\theta(x^{(i)})x_j^{(i)}}$ (6)

(1) $\xrightarrow{(5),(6)}$ $\boxed{\frac{dJ\theta}{d\theta_j} = \frac{1}{m}\sum_{i=1}^{m}\left((h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}\right)}$

b) i)



ii)

```matlab
function f = sigmoid(z)
%SIGMOID Compute sigmoid functoon
%   J = SIGMOID(z) computes the sigmoid of z.

% You need to return the following variables correctly
f = zeros(size(z));

% ====================== YOUR CODE HERE ======================
% Instructions: Compute the sigmoid of each value of z (z can be a matrix,
%               vector or scalar).

f = 1 ./(1 + exp(-z));

% =============================================================

end
```

iii)

```
function [J, grad] = costFunction(theta, X, y)
%COSTFUNCTION Compute cost and gradient for logistic regression
%   J = COSTFUNCTION(theta, X, y) computes the cost of using theta as the
%   parameter for logistic regression and the gradient of the cost
%   w.r.t. to the parameters.

% Initialize some useful values
m = length(y); % number of training examples

% You need to return the following variables correctly
J = 0;
grad = zeros(size(theta));

% ====================== YOUR CODE HERE ======================
% Instructions: Compute the cost of a particular choice of theta.
%               You should set J to the cost.
%               Compute the partial derivatives and set grad to the partial
%               derivatives of the cost w.r.t. each parameter in theta
%
% Note: grad should have the same dimensions as theta
%

h = sigmoid(X*theta);

s=-y.*log(h)-(1-y).*log(1-h);

J=1/m*(sum(sum(s)));
J=sum(J);

for j=1:size(theta)
    grad(j) = (1/m) * sum( sum((h-y).*X(:,j)) );
end

% =============================================================

end
```

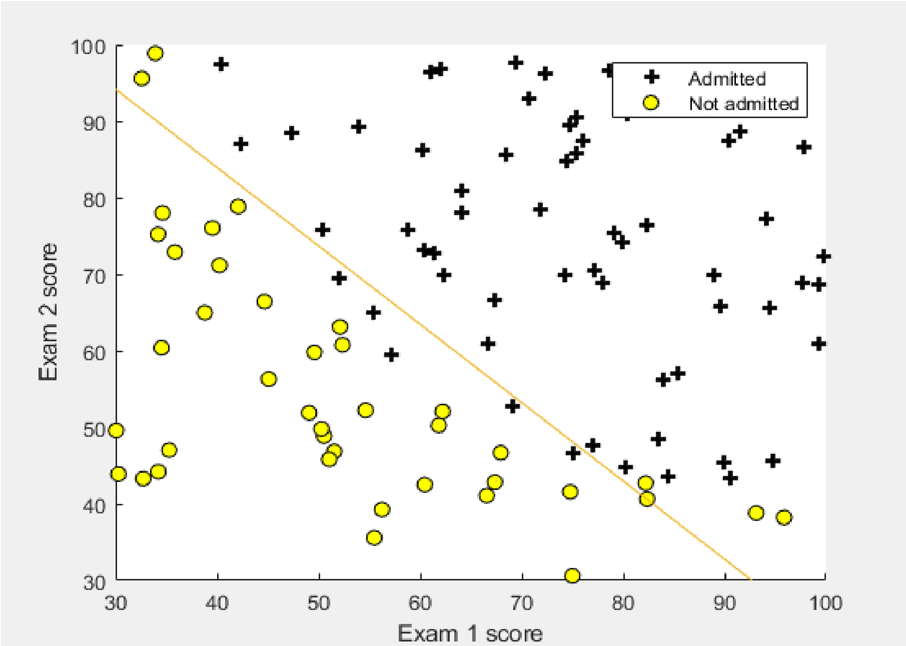
```
Program paused. Press enter to continue.
Cost at initial theta (zeros): 0.693147
Gradient at initial theta (zeros):
 -0.100000
 -12.009217
 -11.262842
```

```
Cost at theta found by fminunc: 0.203498
theta:
 -25.161343
 0.206232
 0.201472
```

iv)

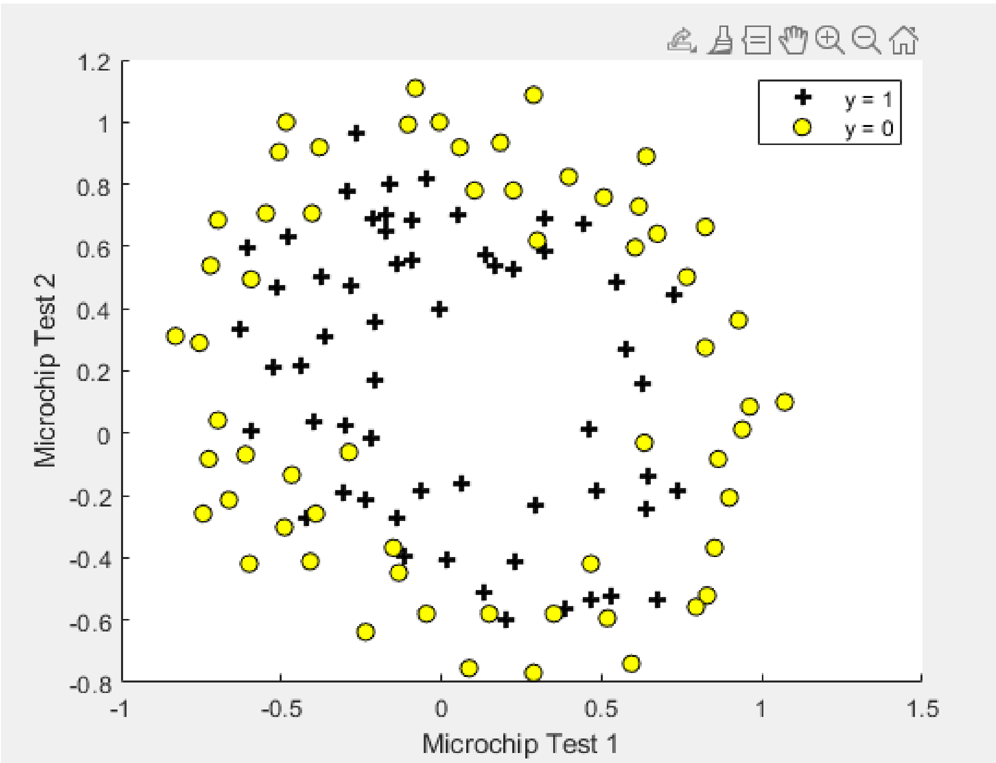
```
For a student with scores 45 and 85, we predict an admission probability of 0.776291

Train Accuracy: 89.000000
```



Θέμα 2: Λογιστική Παλινδρόμηση με Ομαλοποίηση

a)



b)

```
function out = mapFeature(X1, X2) C:\Users\Sophocles\Desktop\HW2_2022\exercise2_2\mapFeature.m
% MAPFEATURE Feature mapping function to polynomial features
%
%   MAPFEATURE(X1, X2) maps the two input features
%   to quadratic features used in the regularization exercise.
%
%   Returns a new feature array with more features, comprising of
%   X1, X2, X1.^2, X2.^2, X1*X2, X1*X2.^2, etc..
%
%   Inputs X1, X2 must be the same size
%
counter = 1;
degree = 6;
out = ones(length(X1(:,1)), 28);

for i = 0:degree
  for j = 0:i
    out(:, counter) = X2.^(j).*X1.^(i-j);
    counter = counter + 1;
  end
end


end
```

c),e)

```
function [J, grad] = costFunctionReg(theta, X, y, lambda)
%COSTFUNCTIONREG Compute cost and gradient for logistic regression with regularization
%   J = COSTFUNCTIONREG(theta, X, y, lambda) computes the cost of using
%   theta as the parameter for regularized logistic regression and the
%   gradient of the cost w.r.t. to the parameters.

% Initialize some useful values
m = length(y); % number of training examples

% You need to return the following variables correctly
J = 0;
grad = zeros(size(theta));

% ====================== YOUR CODE HERE ======================
% Instructions: Compute the cost of a particular choice of theta.
%               You should set J to the cost.
%               Compute the partial derivatives and set grad to the partial
%               derivatives of the cost w.r.t. each parameter in theta

h = sigmoid(X*theta);

s=-y.*log(h)-(1-y).*log(1-h);

J=1/m*(sum(sum(s)))+ (lambda/(2*m)).*sum(theta.^2);
J=sum(J);

for j=1:size(theta)
    grad(j) = (1/m) * sum( sum((h-y).*X(:,j))) + (lambda/m)*theta(j) ;
end

% =============================================================

end
```

d)

Θέμα 2 Λογιστική Παλινδρόμηση με Ομαλοποίηση

δ) Αν θ_j και $x_j^{(i)}$ είναι η συνιστώσα των διανυσμάτων $\theta=[\theta_1,\theta_2,\dots\theta_n]^T$ κ' $x^{(i)}=[x_1^{(i)},x_2^{(i)},\dots x_n^{(i)}]^T$ το j-στοιχείο της κλίσης του σφάλματος είναι:

$$\frac{dJ(\theta)}{d\theta_j} = \frac{d}{d\theta_j}\left(\frac{1}{m}\sum_{i=1}^{m}\left(-y^{(i)}\ln(h_\theta(x^{(i)})) - (1-y^{(i)})\ln(1-h_\theta(x^{(i)}))\right) + \frac{\lambda}{2m}\sum_{j=1}^{m}\theta_j^2\right) =$$

$$= \frac{d}{d\theta_j}\left(\frac{1}{m}\sum_{i=1}^{m}\left(-y^{(i)}\ln(h_\theta(x^{(i)})) - (1-y^{(i)})\ln(1-h_\theta(x^{(i)}))\right)\right) + \frac{d}{d\theta_j}\left(\frac{\lambda}{2m}\sum_{j=1}^{m}\theta_j^2\right)$$

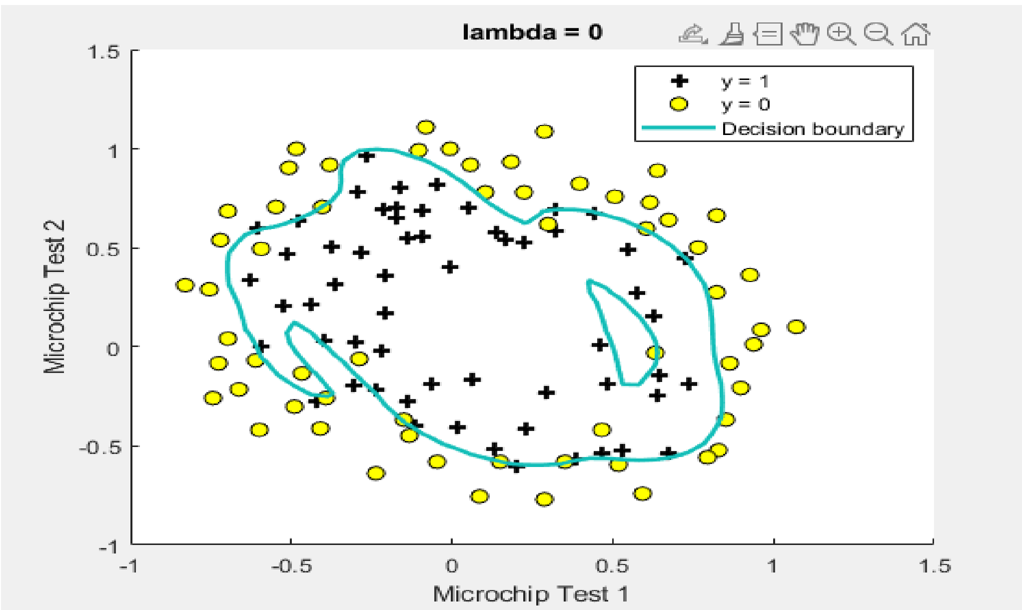
από το θέμα 1a η 1η παράγωγος γίνεται

$$\frac{1}{m}\sum_{i=1}^{m}\left((h_\theta(x^{(i)})-y^{(i)})x_j^{(i)}\right) + \frac{\lambda}{2m}\sum_{j=1}^{m}\frac{d}{d\theta_j}\theta_j^2 =$$

$$= \frac{1}{m}\sum_{i=1}^{m}\left((h_\theta(x^{(i)})-y^{(i)})x_j^{(i)}\right) + \frac{\lambda}{2m}\sum_{j=1}^{m}2\theta_j \Rightarrow$$

$$\boxed{\frac{dJ(\theta)}{d\theta_j} = \frac{1}{m}\sum_{i=1}^{m}\left((h_\theta(x^{(i)})-y^{(i)})x_j^{(i)}\right) + \frac{\lambda}{m}\theta_j}$$

e) lambda=0

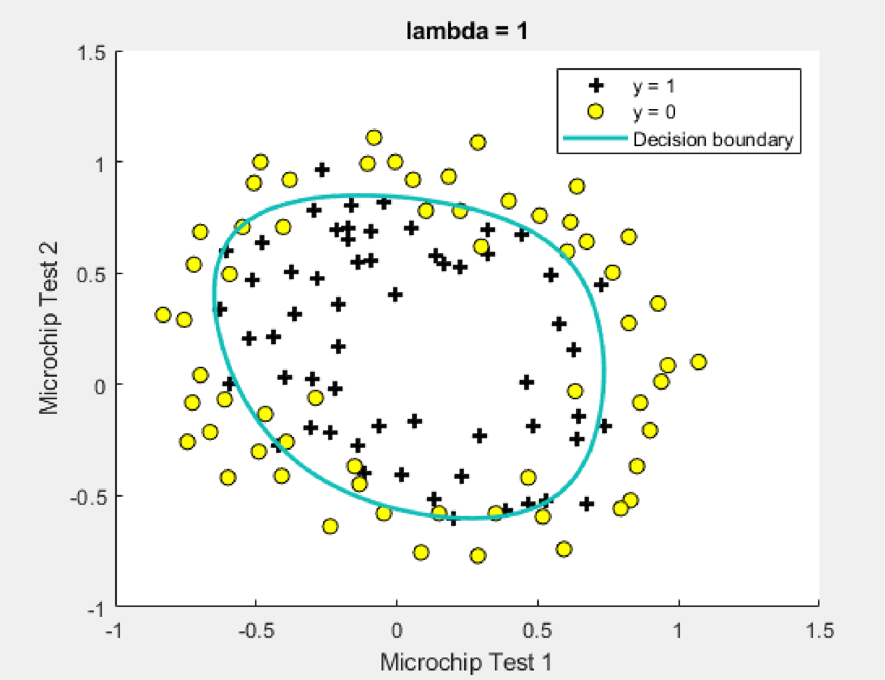


Solver stopped prematurely.

fminunc stopped because it exceeded the iteration limit,
options.MaxIterations = 4.000000e+02.

Train Accuracy: 88.983051
>>

lambda=1



Cost at initial theta (zeros): 0.693147

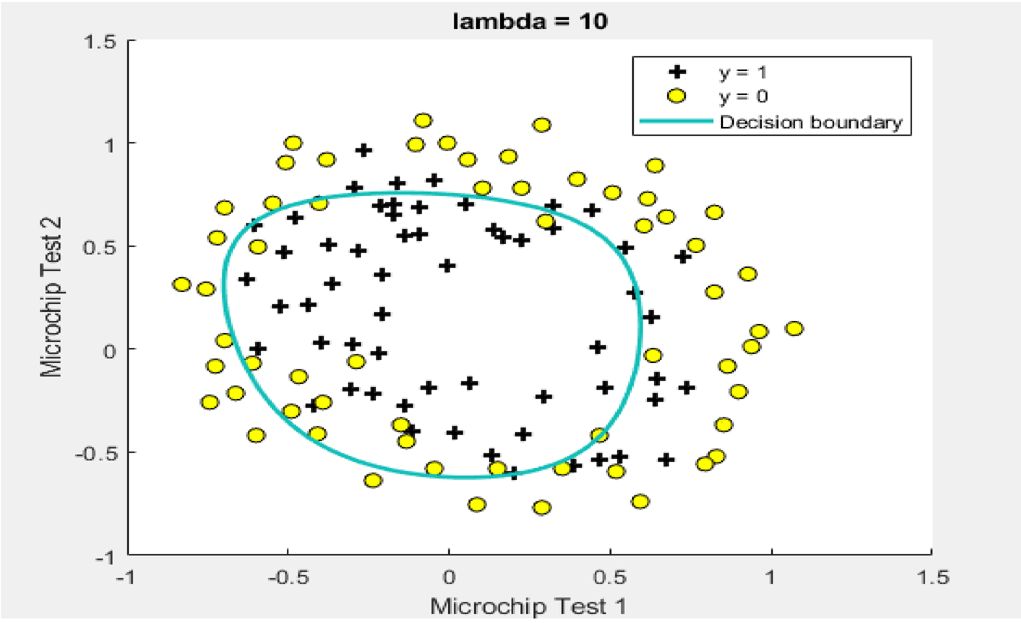
Program paused. Press enter to continue.

Local minimum found.

Optimization completed because the size of the gradient is less than
the value of the optimality tolerance.

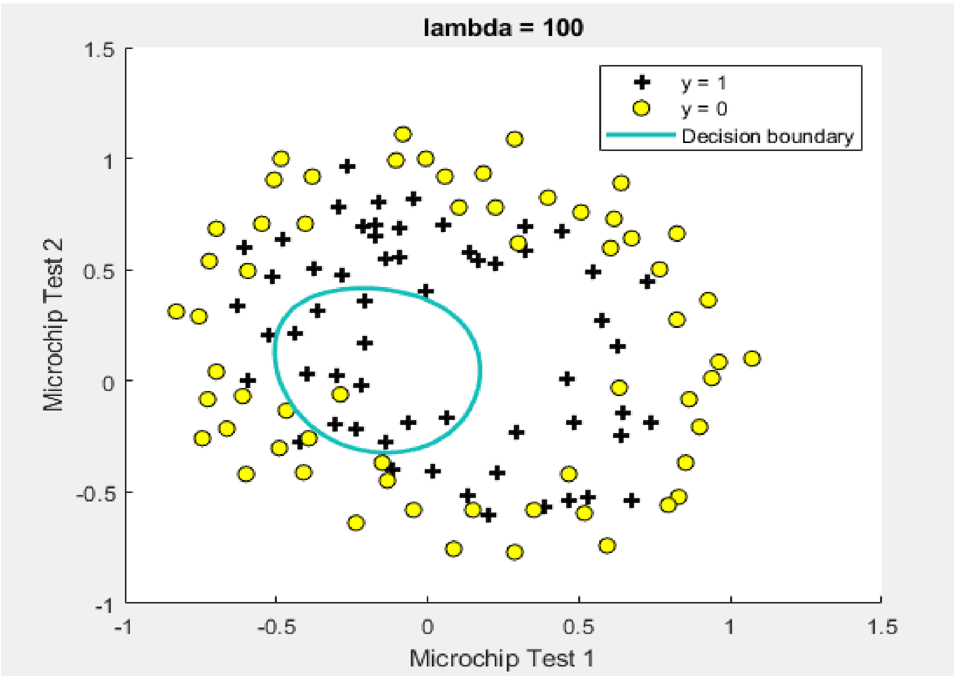
<stopping criteria details>
Train Accuracy: 82.203390
>>

lambda=10



Train Accuracy: 74.576271

lambda=100



Train Accuracy: 60.169492

Θέμα 3: Εκτίμηση Παραμέτρων (Maximum Likelihood)

Θέμα 3: Εκτίμηση Παραμέτρων (Maximum Likelihood)

n δείγματα $D=\{x_1,\dots,x_n\}$: Κατανομή Poisson $p(x|\lambda)=\frac{\lambda^x e^{-\lambda}}{x!}$, $x=0,1,\dots,\lambda>0$

Συνάρτηση Πιθανοφάνειας: $p(D|\lambda)=\prod_{k=1}^{n}p(x_k|\lambda)=\prod_{k=1}^{n}\frac{\lambda^{x_k}\cdot e^{-\lambda}}{x_k!}$

Log likelihood: $L(\theta)=\ln(p(D|\lambda))=\ln\left(\prod_{k=1}^{n}\frac{\lambda^{x_k}\cdot e^{-\lambda}}{x_k!}\right)=\sum_{k=1}^{n}\ln\left(\frac{\lambda^{x_k}e^{-\lambda}}{x_k!}\right)=$

$=\sum_{i=1}^{n}\left(\ln\lambda^{x_k}+\ln e^{-\lambda}-\ln(x_k!)\right)=\sum_{i=1}^{k}\left(x_k\ln\lambda-\lambda-\ln(x_k!)\right)$ σταθερό όρο έξω

$\Leftrightarrow L(\theta)=\ln(\lambda)\sum_{k=1}^{N}x_k-n\lambda-\sum_{k=1}^{N}\ln(x_k!)$

Maximum Likelihood $\hat{\theta}$: $\hat{\theta}_{ML}=\arg\max_{\theta}L(\theta)$ Θέλουμε κλίση ως προς το λ ίση με 0 ①

$\nabla_\lambda L_D(\lambda)=\nabla_\lambda\left(\ln(\lambda)\sum_{k=1}^{n}x_k-n\lambda-\sum_{k=1}^{n}\ln(x_k!)\right)=\frac{1}{\lambda}\sum_{k=1}^{n}x_k-n$ ① $\Rightarrow$

$\Rightarrow\frac{1}{\lambda}\sum_{k=1}^{n}x_k-n=0\Leftrightarrow\sum_{k=1}^{n}x_k=n\lambda\Leftrightarrow\boxed{\lambda=\frac{1}{n}\sum_{k=1}^{n}x_k}$

Θέμα 4 -

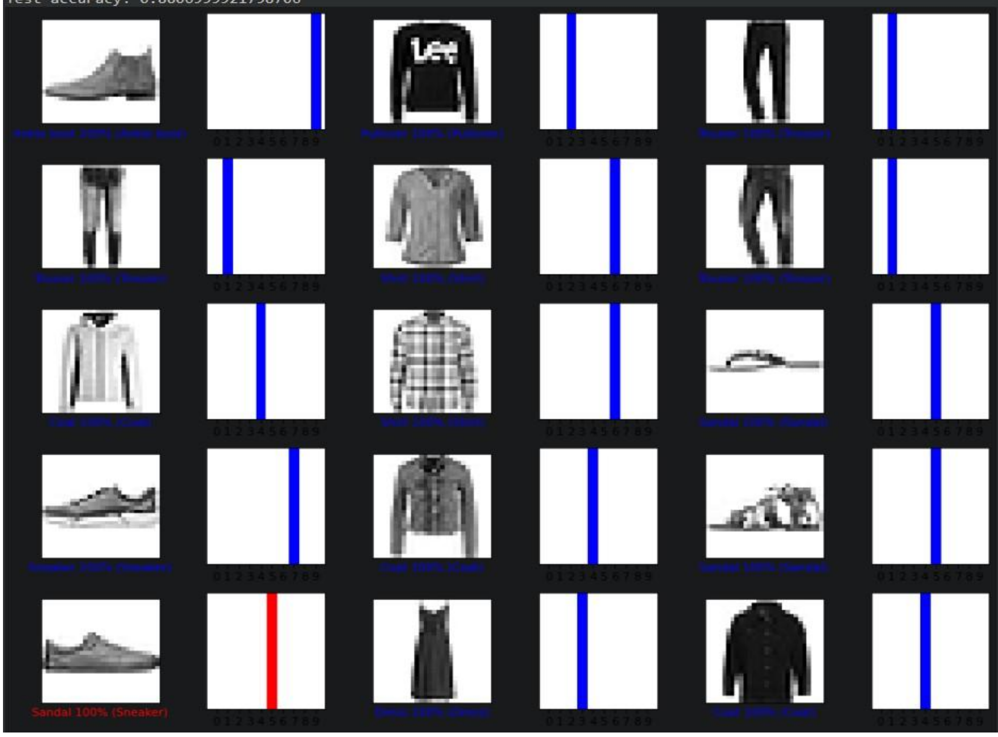
Θέμα 5 -

Θέμα 6

1)epoch = 400

```
Epoch 400/400
1875/1875 [==============================] - 7s 4ms/step - loss: 0.0188 - accuracy: 0.9943 - val_loss: 1.9313 - val_accuracy: 0.8807
313/313 - 0s - loss: 1.9313 - accuracy: 0.8807 - 390ms/epoch - 1ms/step

Test accuracy: 0.8806999921798706
```



epoch = 50

```
Test accuracy: 0.8866000175476074
```