Programming Ex.3

ML:Programming Exercise 3:Multi-class Classification and Neural Networks

Debugging Tip

The submit script, for all the programming assignments, does not report the line number and location of the error when it crashes. The follow method can be used to make it do so which makes debugging easier.

Open ex3/lib/submitWithConfiguration.m and replace line:

```
fprintf('!! Please try again later.\n');

```

(around 28) with:

```
fprintf('Error from file:%s\nFunction:%s\nOn line:%d\n', e.stack(1,1).file,e
  .stack(1,1).name, e.stack(1,1).line );

```

That top line says '!! Please try again later' on crash, instead of that, the bottom line will give the location and line number of the error. This change can be applied to all the programming assignments.

1.4.1 One-vs-all Prediction

The pdf says you should get 94.9% training accuracy. This might not be correct depending on how you implement your code.

"The result you will get may differ a little bit based on how you implement your code. Sometimes, although mathematically two expressions are the same, Matlab may compute them differently. For example, expressions X'(sigmoid(X*theta)-y) and sum((sigmoid(X*theta)-y)*ones(1,size(X,2)).*X) are the same mathematically; however, Matlab does not compute them the same numerically. I tried to use the same input for these two expressions and Matlab gave me a difference about 2*10^(-10) in 1 norm. Therefore, when you use different expressions to compute the gradient and then use fmincg to learn the parameters, your result may be a little different. Actually, when I used the first expression, I got the accruacy 95.14% and when I used the second one, I got 94.94%. They should be both correct in this sense."* **-Posted by guoxian (Student)**

Use the submit feature to find out if you are correct even if you get a different answer for training accuracy.

2.2 Feedforward Propagation and Prediction (Neural network)

It wasn't clear to me weather when computing the hidden layer you only need to compute $g(z^1)$, or should you transform it to binary values (set the value to 1 for g>0.5 and to 0 for g<0.5), like we learned in logistic regression. Both solutions give almost the same results in the final predictions. From the "submit" feature it is clear that you shouldn't transform the values to binary values. **-Posted by inna (Student)**

Prediction of an image outside the dataset (Neural Network)

To test the prediction with images outside the dataset, below is a code that I wrote to import the image and use the prediction.

```
function p = predictImg(Theta1, Theta2, Img)
X = imread(Img);% reads the image .bmp (24 bits) (20x20)

X = double(X);% converts it to double
temp = X;% creates a copy for later use

X = (X.-128)./255;%normalize the features
X = X .* (temp > 0);%return the original 0 values to the X
X = reshape(X, [], numel(X));%converts the 20x20 matrix into a 1x400 vector

displayData(X);%display the image imported

p = predict(Theta1, Theta2, X);% calls the neural network prediction method

```

Usage:

```
1  p = predictImg(Theta1, Theta2, '1.bmp');
2
```


Obs: Because this function will use the Theta1, and Theta2 created my **ex3_nn**, call it before the first use of this function.

-Posted by Vítor Albiero (Student)