

Programming Ex.3

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

Debugging Tip

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

Open `utils/submit0000-ConfigParser.m` and replace line:



```
1  @param() Vector from File to column the code line 1601', a_stack(1,1), File,a_stack(1,1),name, a_stack(1,1),line 1)
2
```

That top line says: I Pass by again later on train, instead of that, the bottom line will give the location and the number of the error. This change can be applied to all the programming assignments.

#### 1.4.2 One-vs-all Prediction

The cell you should get 90.0% training accuracy. This might not be correct depending on how you implement your code.

The results you get may differ slightly based on how you implement your code. Sometimes, although mathematically two expressions are the same, Matlab may compute them differently. For example, expressions `prod(gauss(0*0.001))` and `sum(gauss(0*0.001))` are the same mathematically, however, Matlab does not compute them the same numerically. I tried to use the same code for these two expressions and Matlab gave me a difference about  $2.10 \times 10^{-11}$  error. Therefore, when you use different expressions to compute the gradient and then use fmin to turn the parameters, your results may be a little different. Actually, when I used the fmin function, I got the accuracy 93.4% and when I used the second one, I got 94.94%. They should be both correct in this sense. **Posted by gaussian (Student)**

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

#### 2.3 Feedforward Propagation and Prediction (Neural Network)

It is not clear to me whether when computing the hidden layer you only need to compute  $g^{(2)}$ , or should you transform it to binary values (use the value to 1 for  $g \geq 0.5$  and to 0 for  $g < 0.5$ ) like an output in logistic regression. Both solutions give almost the same results in the final predictions. From the "Submit" feature it is clear that you should transform the values to binary values. **Posted by huu (Student)**

#### Prediction of an image outside the dataset (Neural Network)

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

```
1 function p = predict(img, Theta, sig)
2     % = forward(img) runs the image_img (28 bits) (28x28)
3
4     % = double(img) converts it to double
5
6     temp = A2; create a copy for later use
7
8     % = 10; call .size() to determine the features
9     % = 1; - (temp + A1) returns the original 8 values to the X
10    % = reshape(0, 1) - reshape() converts the 28x28 matrix into a 1x400 vector
11
12    display(temp); % display the image inputted
13
14    p = predict(Theta, Theta, A1); call the neural network prediction method
```

usage:

```
1 p = predict_logit(Therapy, Therapy, "A.logit");
2
```

Obs: Because this function will use the Therapy, and Therapy created by `add_mn_col` before the first use of this function.`

**Posted by Victor Albiero (Student)**

