§ Machine Learning — Week 4 Assignment

1. ex3.m — Multi-class Classification

(1) oneVsAll

```
function [all_theta] = oneVsAll(X, y, num_labels, lambda)
%ONEVSALL trains multiple logistic regression classifiers and returns all
%the classifiers in a matrix all_theta, where the i-th row of all_theta
%corresponds to the classifier for label i
%   [all_theta] = ONEVSALL(X, y, num_labels, lambda) trains num_labels
%   logistic regression classifiers and returns each of these classifiers
%   in a matrix all_theta, where the i-th row of all_theta corresponds
%   to the classifier for label i

% Some useful variables
m = size(X, 1);   → 5000
n = size(X, 2);   → 400

% You need to return the following variables correctly
all_theta = zeros(num_labels, n + 1);
                  10          401
                  5000 X 401

% ========================== YOUR CODE HERE
==========================
% Instructions: You should complete the following code to train num_labels
%               logistic regression classifiers with regularization
%               parameter lambda.
%
% Hint: theta(:) will return a column vector.
%
% Hint: You can use y == c to obtain a vector of 1's and 0's that tell you
%       whether the ground truth is true/false for this class.
%
% Note: For this assignment, we recommend using fmincg to optimize the cost
%       function. It is okay to use a for-loop (for c = 1:num_labels) to
%       loop over the different classes.
%
%       fmincg works similarly to fminunc, but is more efficient when we
%       are dealing with large number of parameters.
%
% Example Code for fmincg:
%
%     % Set Initial
```

```
theta
%     initial_theta = zeros(n + 1, 1);
%
%     % Set options for fminunc
%     options = optimset('GradObj', 'on', 'MaxIter', 50);
%
%     % Run fmincg to obtain the optimal theta
%     % This function will return theta and the cost
%     [theta] = ...
%         fmincg (@(t)(lrCostFunction(t, X, (y == c), lambda)), ...
%                 initial_theta, options);
%
initial_theta = zeros(n + 1, 1);
options = optimset('GradObj', 'on', 'MaxIter', 50);
for a = 1:num_labels
    all_theta(a,:) = fmincg(@(t)(lrCostFunction(t,X,(y ==
a),lambda)),initial_theta,options);
%
================================================================
================
```

goal: gradient descent

iterations

the fmincg likes fminunc, but more efficient

40*1

→ by using loop, the multi-class classification can be transfered to binary classification

the original input matrix

```
  end
```

(2) predict One Vs All

```
function p = predictOneVsAll(all_theta, X)
%PREDICT Predict the label for a trained one-vs-all classifier. The labels
%are in the range 1..K, where K = size(all_theta, 1).
%  p = PREDICTONEVSALL(all_theta, X) will return a vector of predictions
%  for each example in the matrix X. Note that X contains the examples in
%  rows. all_theta is a matrix where the i-th row is a trained logistic
%  regression theta vector for the i-th class. You should set p to a vector
%  of values from 1..K (e.g., p = [1; 3; 1; 2] predicts classes 1, 3, 1, 2
%  for 4 examples)

m = size(X, 1);
num_labels = size(all_theta, 1);

% You need to return the following variables correctly
p = zeros(size(X, 1), 1);

% Add ones to the X data matrix
X = [ones(m, 1) X];

% ====================== YOUR CODE HERE ======================
% Instructions: Complete the following code to make predictions using
%               your learned logistic regression parameters (one-vs-all).
%               You should set p to a vector of predictions (from 1 to
%               num_labels).
%
% Hint: This code can be done all vectorized using the max function.
%       In particular, the max function can also return the index of the
%       max element, for more information see 'help max'. If your examples
```

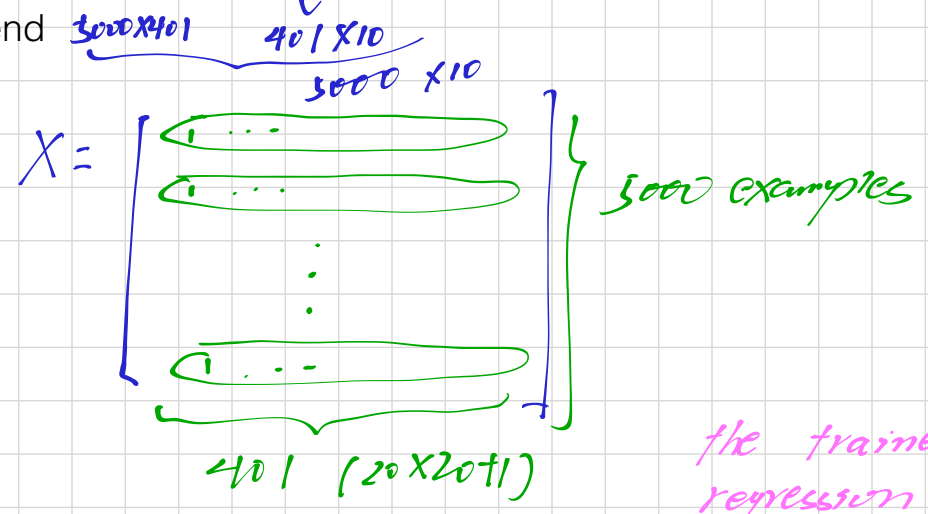
→ 5000

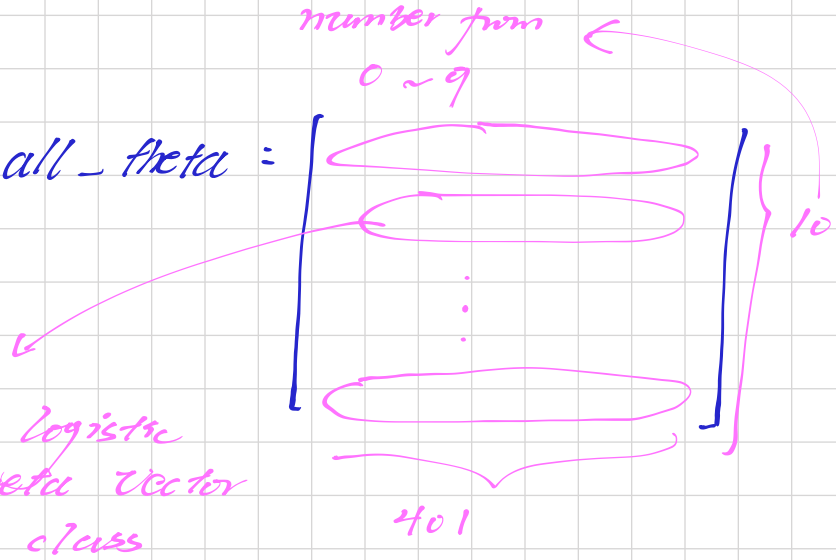
→ 10

→ 5000 * 1

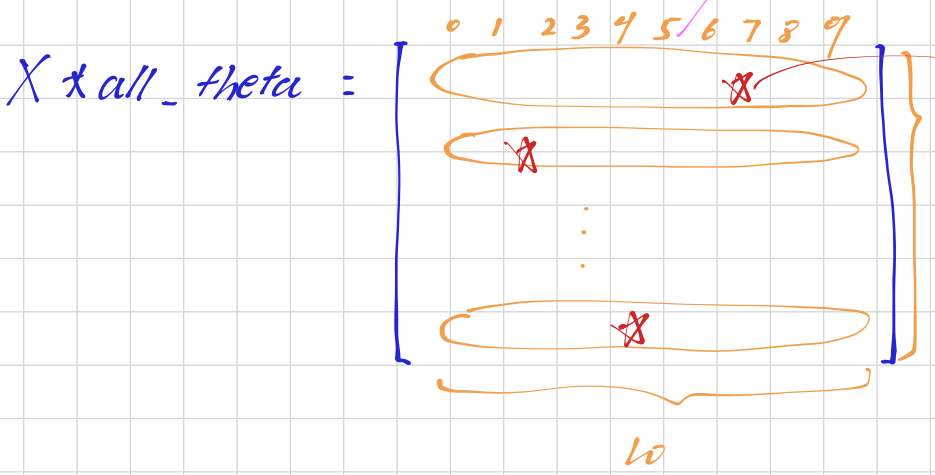
→ 5000 * 401

```
%       are in rows, then, you can use max(A, [], 2) to obtain the max
%       for each row.
%
[z,p]=max(X*all_theta',[],2);
%
=========================================================================
end
```





the trained logistic regression theta vector for the i-th class



use max function to find the max number in each column, (the number "2" determains the dimension that we find)

for [z, p], z is the max value that we've found, and p is the index of the max value

what we need

2. ex3_nn.m — Neural Network

predict.m

```
function p = predict(Theta1, Theta2, X)
%PREDICT Predict the label of an input given a trained neural network
%   p = PREDICT(Theta1, Theta2, X) outputs the predicted label of X given the
%   trained weights of a neural network (Theta1, Theta2)

% Useful values
m = size(X, 1);
num_labels = size(Theta2, 1);

% You need to return the following variables correctly
p = zeros(size(X, 1), 1);

% ====================== YOUR CODE HERE ======================
% Instructions: Complete the following code to make predictions using
%               your learned neural network. You should set p to a
%               vector containing labels between 1 to num_labels.
%
% Hint: The max function might come in useful. In particular, the max
%       function can also return the index of the max element, for more
%       information see 'help max'. If your examples are in rows, then, you
%       can use max(A, [], 2) to obtain the max for each row.
%
X = [ones(m,1) X];
h1 = sigmoid(X*Theta1');
X2 = [ones(m,1) h1];
h2 = sigmoid(X2*Theta2');
[z p] = max(h2 [] 2);
```

just use data to calculate each layer

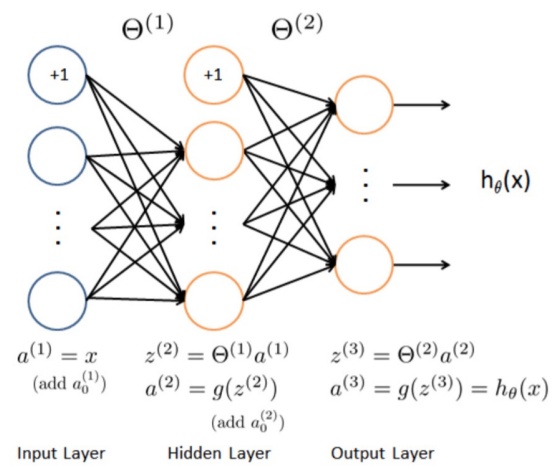


Figure 2: Neural network model.