

LAB ACTIVITY 7: MULTI-CLASS CLASSIFICATION and NEURAL NETWORKS

CARANDO, Paolo Creas
GOMENDOZA, Frank
LIM, Theo Paolo
LORENA, Neil John
ZABAT, Juan Paolo

I. INTRODUCTION

In multi-class classification and neural networks, the group used several methods for determining the actual number of a hand written digit. In this experiment it is necessary to implement a one vs all logistic regression and neural networks.

II. PROCEDURE

1. Download the provided code and training data, place the files to your desired directory where you will access it in your Matlab/Octave.
2. Implement the unregularized cost function in the source file `lrCostFunction.m`.
3. Update and insert your code in `oneVsAll.m` to train logistic regression classifiers with regularization parameter.
4. After you have added your code in `oneVsAll.m`, the source file `ex3.m` will call the `oneVsAll` function to train a multi-class classifier.
5. Complete the code in `predictOneVsAll.m` to create predictions using your learned logistic regression parameters `oneVsAll`.
6. Lastly, complete the code in `predict.m` to create predictions using your learned neural network.

III. RESULTS AND DISCUSSION

Training Set Accuracy: 97.52

Prediction:

Actual	Prediction
7	7
1	1
9	9
4	4
10	0
9	9
7	7

Regression:

Training Set Accuracy: 95.14

IV. CONCLUSION

In this laboratory work, the group was able to implement a prediction algorithm of a data set. The algorithm was able to produce a 97.52% accuracy. The group used some of the functions used in the previous experiments to be able to implement an accurate prediction algorithm. These functions are included in the Appendices section of this report.

V. APPENDICES

[1]<http://briandolhansky.com/blog/2013/9/23/artificial-neural-nets-linear-multiclass-part-3>

```
H = sigmoid(X*theta);
T = y.*log(H) + (1 - y).*(log(1 - H));
J = -1/m*sum(T) +
lambda/(2*m)*sum(theta(2:end).^2);

ta = [0; theta(2:end)];
grad = X'*(H - y)/m + lambda/m*ta;

grad = grad(:);
```

Code 1. lrCostFunction

```
function g = sigmoid(z)
g = 1.0 ./ (1.0 + exp(-z));
end
```

Code 2. Sigmoid

```
for c = 1 : num_labels,
    initial_theta = zeros(n + 1 , 1);
    options =
optimset('GradObj' , 'on' ,
'MaxIter' , 50);
    [theta] =
fmincg(@(t)(lrCostFunction(t , X ,
(y == c) , lambda)), initial_theta
, options);
    all_theta(c,:) = theta';
end
```

Code 3. oneVsAll

```
C = sigmoid(X*all_theta');
[M , p] = max(C , [], 2);
```

Code 4. predictOneVsAll

```
a1 = X;
a2 = sigmoid(a1*Theta1');
a2 = [ones(m , 1) a2];
a3 = sigmoid(a2*Theta2');
[M , p] = max(a3 , [], 2);
```

Code 5. Predict

VI. BIBLIOGRAPHY