EEE 498 ML with application to FPGAs

Homework 4

1) Starting with the Least squares cost function, find the first and second derivative, show your work.

2) Show that the derivative of the Sigmoid function is as shown here $\nabla \sigma(t) = \sigma(t)\left(1-\sigma(t)\right)$. show all steps

3) Using the cross-entropy cost function prove the equation for the gradient given in the lecture, show all steps

4) Write your own cross entropy cost function logistic regression algorithm. Use the iris dataset as shown below.

```
from sklearn import datasets
iris = datasets.load_iris()
X = iris.data[:,0:4]
y = iris.target
```

You can simplify this by choosing the first two sets of 50 points so that there are only two classes. If you wish you can use all three classes (all 150 observations) which will require multi-classifications, this will receive extra credit.

The one hot coding algorithm will make this easier and is provided.

If you want to show the highest correlated (optional)

```
##
##  Most Highly Correlated
##
def mosthighlycorrelated(mydataframe, numtoreport):
```

… in the dataanalysis code provided previously

To use this which is in the panda dataframe

```
import pandas as pd
irisp = pd.DataFrame(iris.data,columns=iris.feature_names)
print("Most Highly Correlated")
print(mosthighlycorrelated(irisp,Nfeatures))
print('\n',irisp.head())
```

5) Compare it to a canned algorithm logistic regression algorithm like the one in sklearn.

Turn in all your code files and results organized so that the grader can easily find and test your work