EEE 498 ML with application to FPGAs

All homework is handed in online. Don't hand in pictures of the code files, hand in the actual code files so I can see if they work. Hand in images of outputs of your code (displays, plots …) converted to pdf. If there are questions to be answered answer them in PowerPoint, Word, or on paper and convert to pdf (It doesn't actually matter how you get to the pdf).

Organize your work so that the answer to each problem can be easily identified. You will not get credit if your submission is poorly organized, or if the grader cannot find the answers.

You can use any language python, c, C++, MATLAB,…, to do coding though Python is likely the easiest and easiest to translate from lectures.

Homework 1

1) What are the zeroth, first, and second conditions of optimality?
2) What are the zeroth, first conditions of convexity?
3) What is the Least Squares Cost function? Use the compact notation. Show that it is convex.
4) Use the dataset regressionprob1_train0.csv.

   a) Using a matrix inversion (such as lecture 4, slide 12 … or MLR p. 108 eqn 5.17) find the weights and intercept to predict F from the other columns. Determine the residual squared, or R^2. Use compact notation as described in the lecture so that you get an intercept b (or w0). You can read the file in using pandas and create A in compact notation using this code:
```
import numpy as np
import pandas as pd
df = pd.read_csv('regressionprob1_train0.csv')
X = df.iloc[:,0:4].values
y = df['F'].values
ones_ = np.ones(len(y),float)
```

```
## compact notation
A = np.column_stack((ones_,X))
```

This Python code will calculate R^2

```
##
##  unexplained variance squared
##  or R squared
##  Y is the target value
##  Yp is the predicted value
##
import numpy as np
def Rsquared(Y,Yp):
    V = Y-Yp
    Ymean = np.average(Y)
    totvar = np.sum((Y-Ymean)**2)
    unexpvar = np.sum(np.abs(V**2))
    R2 = 1-unexpvar/totvar
    return(R2)
```

you can also use

```
from sklearn.metrics import r2_score
R2 = r2_score(y,Yp)
```

b) Solve again using a linear solver like numpy.linalg.solve and compare results.

c) Now use the model trained in a) on dataset regressionprob1_test0.csv. Determine the residual squared, or R^2. What can you conclude about your trained model from a) and b)?

5) Now write your own Gradient descents code and code the gradient to find the weights to minimize the cost function in (3) using training data in 4 (regressionprob1_train0.csv). (hint: see Lecture 4 slide 26, and Lecture 6 slide 23) Solve for the weights and intercept using this using this training dataset. Compare with your solution from (4). Also, use compact notation.

6) Use a canned linear regression similar to sklearn.linear_model.LinearRegression and extract the weights for the training

dataset. Apply this model to the training dataset. Calculate and compare your resulting R^2 and model weights to those in (4), and (5). You can use compact notation, or you can tell your regression to extract an intercept.

Hand in all output as a screen grab for each problem converted to pdf. It is a good practice to paste well organized and labeled images into PowerPoint, answer all questions in the PowerPoint file, and then convert the PowerPoint file to pdf. Hand in all your code files. If code files are missing you will not get credit. It is typically easier to grade individual files rather than a zipped file, but either is accepted.