Homework 5: Due Fri 09-14-2018
Total Points (40 pts)

1. (10 pts) (Simple Bias Network)

   (a) Let $\mathbf{1}_{n\times 1}$ equal the *ones vector*, $\begin{pmatrix} 1 & 1 & \cdots & 1 \end{pmatrix}^\top$, consisting of a vector of n 1's. Show that $\mathbf{1}^\top \mathbf{1} = n$.

   (b) Let $\mathbf{y} = \begin{pmatrix} y_1 & y_2 & \cdots & y_n \end{pmatrix}^\top$. Show that $\mathbf{y}^\top \mathbf{1} = \sum_{i=1}^n y_i$.

   (c) Consider a simple bias network (i.e. no input features). Use vector operations to show that $b = \bar{\mathbf{y}} = \frac{1}{n}\sum_{i=1}^n y_i$ minimizes MSE.

   Hint: Since there are no features, the data matrix is just the ones vector, i.e. $\mathbf{X} = \mathbf{1}$. Solve the normal equations $\mathbf{X}^\top \mathbf{X}\mathbf{w} = \mathbf{X}^\top \mathbf{y}$. Note that $\mathbf{w} = b$.

2. (10 pts) (Central Difference Formula) Consider the function

$$y = f(\mathbf{x}) = x_1^2 + x_1 x_2 + 5x_2 + x_2^2.$$

   (a) Compute the gradient vector $\left.\dfrac{dy}{d\mathbf{x}}\right|_{(-1,1)}$.

   (b) Numerically approximate the gradient vector in part (a) using the central difference formula

$$\frac{\partial y}{\partial x_i} \approx \frac{f(x_1, \ldots, x_i + \epsilon, \ldots, x_n) - f(x_1, \ldots, x_i - \epsilon, \ldots, x_n)}{2\epsilon}$$

   where $\epsilon = 10^{-5}$. How accurate is the numerical approximation?

3. (10 pts) (Gradient Descent Algorithm) Use the data set `wine_quality_white.csv` to train a linear regression network to predict wine quality in the following ways:

   (a) Standardize all features (excluding the bias) and then apply the gradient descent algorithm to minimize training MSE.

   Hint: $\dfrac{d\ell}{d\mathbf{w}} = \dfrac{2}{n}\mathbf{X}^\top(\mathbf{X}\mathbf{w} - \mathbf{y})$.

   (b) Try and repeat part (a) without first standardizing the features.

   (c) Solve the normal equations.

   (d) Compare the speed and accuracy of each method and comment.

4. (10 pts) (L^2 Regularization) Use the data set `wine_quality_white.csv` to train a linear regression network to predict wine quality. Use an 80% training, 20% testing data split and L^2 regularization to minimize test MSE in the following two ways. (Don't forget to standardize the features first.)

   (a) Load only the first 100 rows of the data set using the command:

```
df = pd.read_csv('... path to file/wine_quality_white.csv',sep=';',nrows=100)
```

   (b) Load all of the data.

   (c) What effect does increasing the size of the data set have on test MSE and on the regularization parameter α_o? As the size of the data set is increased, is more or less regularization needed.