ELE 364: Assignment #4

1. (10 pts) Consider the following multivariate linear regression model:

   Predict($\mathbf{d}$) = $\mathbf{w}[0] + \mathbf{w}[1]\times\mathbf{d}[1]+\mathbf{w}[2]\times\mathbf{d}[2]+\mathbf{w}[3]\times\mathbf{d}[3]$ and a historical dataset:

ID	$\mathbf{d}[1]$	$\mathbf{d}[2]$	$\mathbf{d}[3]$	Target
1	1	2	3	20
2	2	3	4	30
3	3	4	5	50
4	4	5	6	60

   (a) Assume we initialize the weights as follows: $\mathbf{w}[i] = i$ for i =0, 1, 2, 3. Predict the target values for all the instances in the dataset. Then, derive the value of the sum of squared error function L_2.

   (b) Assume a learning rate of 0.0001. Calculate the new $\mathbf{w}[i]$'s for i =0, 1, 2, 3 after one iteration.

2. (10 pts) Consider the neuron shown below that executes the function $tanh(\mathbf{w}\cdot\mathbf{d})$, where $tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ and is called the activation function. Derive the weight update rule when using the gradient descent algorithm to train this neuron.

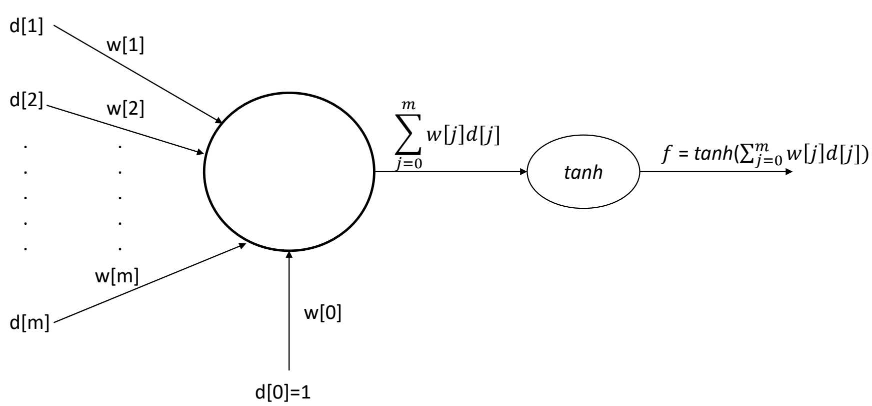

Figure 1: A neuron.

3. (10 pts) End-of-Chapter Problem 8 for ID = 2 query.

4. (10 pts) Consider the cubic kernel: $kernel(\mathbf{d},\mathbf{q}) = (\mathbf{d}\cdot\mathbf{q}+1)^3 = \phi(\mathbf{d})\cdot\phi(\mathbf{q})$. Given that $\mathbf{d} = \langle \mathbf{d}[1], \mathbf{d}[2]\rangle$ and $\mathbf{q} = \langle \mathbf{q}[1], \mathbf{q}[2]\rangle$, find $\phi(\mathbf{d})$.

5. (20 pts) In this project, you will train error-based models to identify types of glass. The dataset consists of 214 instances of glass with the following nine numerical descriptive features per instance: Refractive index, Sodium, Magnesium, Aluminum, Silicon, Potassium, Calcium, Barium, Iron.

The target class of each glass instance is one of six glass types: (i) Float processed building window, (ii) Non-float processed building window, (iii) Float processed vehicle window, (iv) Container, (v) Tableware, (vi) Headlamp.

The data will be divided into a training set and a validation set. You will train a multinomial logistic regression classifier with stochastic gradient descent as well as support vector machines. You will explore different kernel functions, such as the polynomial kernel and the radial basis kernel.

Refer to the Jupyter notebook for more details.