Homework4 For Machine Learning

Vergil/Zijun Li李子骏

1. **Suppose you trained your logistic regression classifier which takes an image as input and outputs either dog (class 0) or cat (class 1). Given the input image x, the hypothesis outputs 0.2. What is the probability that the input image corresponds to a dog? (2 points)**

   0.8
   P(Dog) = 1 - P(Cat) = 1 - 0.2 = 0.8

2. **You are trying to build a logistic regression classifier which predicts whether the price of a house is less than 100K USD (class 0: cheap house) or greater than or equal to 100K USD (class 1: expensive house). The training examples given to you have two features: the depth and the frontage of the house and the associated class (0 or 1) for each training example. Assume that the price of a house is linearly dependent upon the area of the house. How would you make logistic regression work in this case? Write down the full hypothesis. (8 points)**

   $h_\theta = \frac{1}{1+e^{-\theta_0-\theta_1*(x_1*x_2)}}$ where x_1 is depth and x_2 is frontage. θ are the parameters
   $h_\theta = P(class1) = sigmoid(z) = \frac{1}{1+e^{-z}}$
   $z = logit(P) = \theta_0 + \theta_1 * Area$ (linearly dependent)
   $Area = x_1 * x_2$ where x_1 is depth and x_2 is frontage
   thus $h_\theta = \frac{1}{1+e^{-\theta_0-\theta_1*(x_1*x_2)}}$ where x_1 is depth and x_2 is frontage.

3. **Can the logistic regression cost function ever be negative? Please explain your answer.**

   No
   Cost function: $J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log(1-h_\theta(x^{(i)}))]$
   As $h_\theta(x^{(i)}) \in [0,1]$ and $1-h_\theta(x^{(i)}) \in [0,1]$
   $\log h_\theta(x^{(i)}) and \log(1-h_\theta(x^{(i)})) \in (-\infty, 0]$
   thus $\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log(1-h_\theta(x^{(i)}))$ is always negative
   so $-\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log(1-h_\theta(x^{(i)}))]$ is always positive(can never be negative)

4. **You have a neural network which has k layers of logistic units. You generate k random numbers r_1, r_2, ..., r_k. Next, for every i, you initialize all theta's in layer i with the random value r_i. Will this work? (3 points)**

   No,
   If initialized to the same value: all nodes within a given layer become identical and may remain identical

5. **Is it possible that the cost function goes up with each iteration of gradient descent in a neural network? If no, explain why. If yes, explain how you would fix it. (3 points)**

Yes, it is possible

1. The learning rate is too high. Solution: find a more suitable learning rate

2. The Stochastic gradient descent (SGD) gives a random sample which may result the higher cost. Solution: Monitor the values and deal with it in time

6. **Is it possible that depending on your random initialization, your gradient descent may converge to different local optima (i.e., if you run the algorithm twice with different random)**

   Yes, it is possible

   Different random initialization may result differenct result(one or more local optima)

7. **In each iteration of k-means clustering, we map each point to the centroid which is closest to this point. Prove that this step can only reduce the cost function. (3 points)**

   Cost function: $J(c^1, ...c^m, \mu_1, ..., \mu_k) = \frac{1}{m} \sum_{i=1}^{m} ||x^i - \mu_{c^i}||^2$

   Every iteration of k-means clustering, it will reassign all the points and get the lower mean distance which is exactly as same as the cost function

8. **Suppose you have a large number of points on the graph and the value of k is large. On the left side, the points are very dense and close to each other. On the right side, the points are further away from each other. Are you likely to see bigger clusters on the left side or the right side? Why? Note: By bigger clusters, we mean bigger in terms of size (or diameter) rather than number of points. (5 points)**

   We might see bigger clusters on the right side

   K-means are trying to minimize the within-cluster sum of squares . So the cluster will try to get the minimum average squared distances for all the data points but not for the data points in dense area, and it could be on right side.