Solution to Assignment - 4

1. If the cost function is $(y - \hat{y})^4$ what would be the gradient descent step? Given $\hat{y} = w^T x$

$$a). \quad w_j := w_j - 4\alpha(y^{(i)} - \hat{y}^{(i)}))^3 . \frac{\partial}{\partial w_j}\hat{y}^{(i)}; \quad j := 0, \ldots, n$$
$$b). \quad w_j := w_j - 4\alpha(y^{(i)} - \hat{y}^{(i)}))^3; \quad j := 0, \ldots, n$$
$$c). \quad w_j := w_j - \alpha(y^{(i)} - \hat{y}^{(i)}))^4; \quad j := 0, \ldots, n$$
$$d). \quad None$$

Answer (a)

Solution:

The gradient descent steps are:

$$J(\mathbf{w}) := (\mathbf{y} - \hat{\mathbf{y}})^4$$
$$\mathbf{w} := \mathbf{w} - \alpha \nabla_w (J(\mathbf{w}))$$
$$w_j := w_j - \alpha \nabla_{w_j} (J^i(w_j))$$
$$J^i(w_j) := (y^{(i)} - \hat{y}^{(i)})^4$$
$$\nabla_{w_j} J^i(w_j) := 4(y^{(i)} - \hat{y}^{(i)})^3 \frac{\partial}{\partial w_j}\hat{y}^{(i)}$$
$$w_j := w_j - 4\alpha(y^{(i)} - \hat{y}^{(i)}))^3 \frac{\partial}{\partial w_j}\hat{y}^{(i)}; \quad j := 0, \ldots, n$$

2. For the quadratic cost function with L_2 regularizer what would be the gradient step i.e., the expression. Given $L_2 = (\lambda/2m) \sum_{i=1}^{m} w_j^2$

$$a). \quad w_0 = w_0 - \frac{\alpha}{m} \sum_{i=1}^{m} \left(\hat{y}^{(i)} - y^{(i)}\right) x_0{}^{(i)}$$
$$w_j = w_j - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} \left(\hat{y}^{(i)} - y^{(i)}\right) x_j{}^{(i)} + \frac{\lambda}{m} \sum_{j=1}^{n} w_j \right]; \; j = 1, ..., n$$

b). $w_j = w_j - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} \left(\hat{y}^{(i)} - y^{(i)} \right) x_j^{(i)} + \frac{\lambda}{m} \sum_{j=1}^{n} w_j \right] ; j = 0, ..., n$

c). $w_j = w_j - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} \left(\hat{y}^{(i)} - y^{(i)} \right) x_j^{(i)} + \frac{\lambda}{m} \sum_{j=1}^{n} w_j^2 \right] ; j = 0, ..., n$

d). *None*

Answer (a)

Solution:

Regularization term does not include bias term (here w_0)

$$J(w) = \frac{1}{2m} \sum_{i=1}^{m} \left(\left(\hat{y}^{(i)} - y^{(i)} \right)^2 \right) + \frac{\lambda}{2m} \sum_{j=1}^{n} w_j^2$$

$$w_0 = w_0 - \frac{\alpha}{m} \sum_{i=1}^{m} \left(\hat{y}^{(i)} - y^{(i)} \right) x_0^{(i)}$$

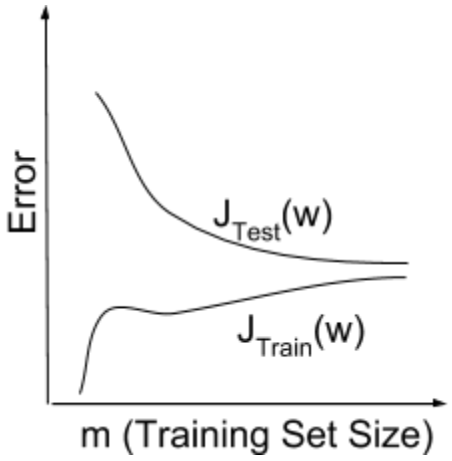
$$w_j = w_j - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} \left(\hat{y}^{(i)} - y^{(i)} \right) x_j^{(i)} + \frac{\lambda}{m} \sum_{j=1}^{n} w_j \right] ; j = 1, ..., n$$

3. Which of the following statements are True? Check all that apply:
   A) If a learning algorithm is suffering from high bias, only adding more training examples may **not** improve the test error significantly.
   B) A model with more parameters is more prone to overfitting and typically has a higher variance.
   C) When debugging learning algorithms, it is useful to plot a learning curve to understand if there is a high bias or high variance problem.
   D) Increasing degree of the polynomial in curve fitting will increase the bias in the model

Answer: A, B, C

Solution: Option D is incorrect as increasing the degree of polynomial will increase the variance of the system and decrease the bias (for training data).

4. The figure below shows the plot of the learning curves of a learning algorithm. It is found that it has an unacceptably high error on the test set. What is the algorithm suffering?



A) High Variance
B) High Bias
C) High Variance and Low bias
D) None

Answer: (B)
Solution: This learning curve shows high error on both the training and test sets, so the algorithm is suffering from high bias.

5. Suppose you have implemented a regularized linear regression model. You observe that on the held out testing set, the model makes unacceptably large errors with its predictions. However, you observe that the model performs well (has a low error) on the training set. Which of the following steps can be incorporated to lower the error on testing dataset. Select all that apply.

A) Try using a smaller set of the features
B) Try decreasing the regularization parameter λ
C) Get more training examples

D) Use fewer training examples

Answers: A, C

Solution:

Option **A** is True: The problem suggests high variance and hence overfitting of the training set. Reducing the feature set might alleviate the variance problem.

Option **B** is False: As decreasing the regularization parameter will increase the overfitting, not decrease it.

Option **C** is True: Adding more training data will help to fit increasingly complex models. For a limited dataset complex models will tend to overfit while adding more data points will reduce oscillations in the fitted curve/surface.

Option **D** is False.

6. Suppose you have implemented a regularized linear regression model. You observe that on the held out testing set, the model makes unacceptably large errors with its predictions. Furthermore, you observe that the model performs **poorly** on the training set. Which of the following steps can be incorporated to lower the error on the testing dataset. Select all that apply.

A) Try to obtain an additional set of features

B) Try increasing the regularization parameter λ

C) Get more training examples

D) Try adding polynomial features

Answers: A&C

Solution:

Option **A** is True: The poor performance on the training set suggests high bias problem which could be alleviated using more features.

Option **B** is False: Increasing regularization might underfit the training data and worsen the performance on testing data.

Option **C** is True: Having more training examples helps to reduce variance and improves performance on testing data. More training examples is most of the time beneficial to improve the model performance.

Option **D** is False: As adding polynomial features will lead to overfitting i.e. high variance model- increase variance problem.

7. Suppose you are training a regularized linear regression model. Check which of the following statements are true? Select all that apply.

A) The regularization parameter λ value is chosen so as to give the lowest training set error.
B) The regularization parameter λ value is chosen so as to give the lowest cross validation error.
C) The regularization parameter λ value is chosen so as to give the lowest test set error.
D) The performance of a learning algorithm on the training set will typically be better than its performance on the test set.

Answers: B & D

Solution:

Option **A** is false: Low training set error can be obtained using a weak regularization ie. a small value for lambda.

Option **B** is True: For a fixed model parameters learned from the training set, cross-validation lets in fine tuning regularization parameters.

Cross-validation helps in testing for high variance, as in every fold you choose a different combination of training and validation.

Option C is false: Regularization parameters cannot be tuned on test data. As in practical situations test data is not available.Model will just end up fitting to the test data.

Option **D** is True: The parameters are tuned to minimize the training set error, so the performance on the training set would mostly be better than the testing set.

Consider the data provided below, which follows linear regression model $h_w(x) = w_0 + w_1 x$, and the cost function is MSE.

x	**y**
1	0.5
2	1
4	2
0	0

8). What is the cost?

A. 1.03125
B. 2.03125
C. 3.03125
D. 4.03125

ans). B

9). Run gradient descent algorithm with learning rate 0.1. What are the updated weights now?

A. $(w_0, w_1) = (-1.875, -4.375)$
B. $(w_0, w_1) = (0.8125, 1.245)$
C. $(w_0, w_1) = (1.875, 0.5625)$
D. $(w_0, w_1) = (0.8125, 0.5625)$

ans). D

10). What is the cost now?

A. 0.4292
B. 1.4292
C. 2.4292

D. 3.4292

ans). A

Matlab code for questions 8,9, and 10:

```
clc
clear
x = [1;2;4;0];              % defining x vector
y = [0.5;1;2;0];            % defining y vector
m = size(x,1);              % defining m, which is the number of examples
X = [ones(m,1),x];          % appending ones (bias) to input vector
w = ones(size(X,2),1);% defining weight matrix
J = 1/(2*m)*sum((y-X*w).^2); % defining cost function
fprintf('cost is %0.4f\n',J)
alpha = 0.1;    % defining learning rate
dJ_w = [1/m*sum((X*w-y).*X)]';        % gradient of cost function with respect to weights
w = w - alpha*dJ_w;             % updating weights using gradient descent method
fprintf('(w_0,w_1) = (%0.4f,%0.4f)\n',w(1),w(2))
J = 1/(2*m)*sum((y-X*w).^2); %  finding cost with updated weights
fprintf('cost with updated weights is %0.4f\n',J)
```

8).
ans). cost is 2.0312
9).
ans). (w_0,w_1) = (0.8125,0.5625)
10).
ans). cost with updated weights is 0.4292