1.

If the cost function is $(y-\hat{y})^4$ what would be the gradient descent step?

~~Answer: **Option D**. None~~

- ~~Look like the Option A is answer. That equation uses only one record for updating parameters (i.e. wj). That happens only in *Stochastic gradient descent*. The question doesn't mention *Stochastic*. So I believe we should consider *batch gradient descent* which uses all records. Option A don't consider all the records to update.~~
- Got more details from Balaji Sir: "**Assume just a single data point which is equivalent to having no summation sign. Alternatively, you can assume this is Stochastic Gradient Descent with a single data point.**"
- The option A look like answer. But there is a negtive sign missing.

- Accepted answer is **Option A**
- You may be thinking how it is possible. Do your karma very well and hope the God of luck will turn favour to you!!

2.

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

I believe the question is incomplete. didn't give the range of j of w. Or didn't give summation sigma for j.

Answer: **Option A**. I understood the psychology of the team who evaluate this assignment. I choose this option based on my intuation of what they may choose as answer; not based on facts or knowledge. Sorry, I cannot express my intution in words!

3 Which of the following statements are True? Check all that apply:

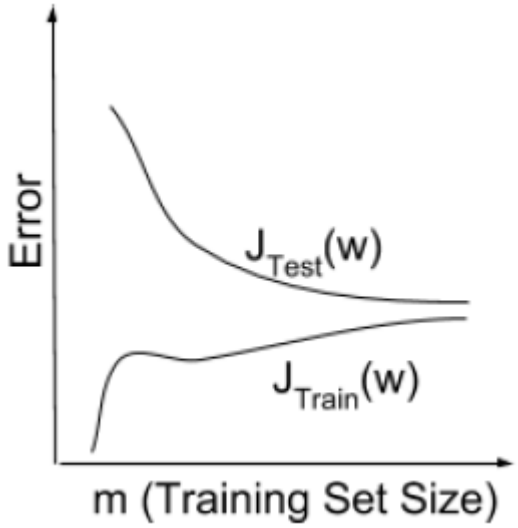
- **A. RIGHT. If a learning algorithm is suffering from high bias, only adding more training examples may not improve the test error significantly.**
  - The reason of high bias is that the learning algorithm is not flexible/complex to capture the structure of the actual hypothesis. More data can help only if the learning algorithm have enough flexibility/complexity.
- **B. RIGHT: A model with more parameters is more prone to overfitting and typically has a higher variance.**
  - There is a famous saying *"With four parameters I can fit an elephant, and with five I can make him wiggle his trunk."*" It means with more parameters you can fit any data set very well. But that may be overfit!!

- More parameters will make the model learn noise too from the training data (It is called overfitting.) Such model performs well in trainig data, but performs poorly in test data (it is called high variance).

- **C. RIGHT: When debugging learning algorithms, it is useful to plot a learning curve to understand if there is a high bias or high variance problem.**
  - Learning curve for training data and test data are used for ploting.
    - **High Variance**: Training error is very low and test error is very high
    - **High Bias**: Training error is very high

- D. WRONG: **Increasing degree of the polynomial in curve fitting will increase the bias in the model.**:
  - More degree of the polynomial will make the model more complex. Complex model will have low bias.

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. WRONG: High Variance: Training error should be low and test error high.
- **B. RIGHT. High Bias**: Here training error and test error is almost equal. That means the learning algorithm learnt some actual patterns from training data and didn't learn significant noise from the training data. The test error is high means the learning algorithm didn't capture the enough patterns of the actual hypothesis.
- C. WRONG. High Variance and Low bias: Option A is wrong.
- D. WRONG. None: One of the above options is chosen :)

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.

training error low and test error high means **high variance**. That means learning alogithm is so complex and learning noise from the training data. Here need to reduce the variance for lower the error on testing dataset.

- **A. RIGHT. Try using a smaller set of the features**:
  - Some of the feature may not be relevent fetaures. That may cause to learn noises. Remove irrelevant features.
- **B**. WRONG. Try decreasing the regularization parameter λ
  - Regularizing is the way to control the model become too complex. Reducing the regularization parameter λ means making the model more complex. The model is already enough complex! Adding more complexity will make the situation worse.
- **C. RIGHT: Get more training examples**
  - More examples will increase the chance of getting more signal and reduce the capturing the noise. So it will reduce the variance.
- **D**. WRONG. Use fewer training examples
  - It will only increase the variance.

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

Both training error and test error are high. It is the sign of **high bias**. The learning algorithm should be more complex.

- **A. RIGHT. Try to obtain an additional set of features**
  - More features will make the model more complex.
- **B**. WRONG. Try increasing the regularization parameter λ
  - Regularization parameter λ control the complexity of the model. The parameter should be decreased to make the model more complex.
- **C**. WRONG.Get more training examples
  - Adding more won't help much in *high bias* situation.
- **D. RIGHT. Try adding polynomial features**
  - It is like adding more features.

- Sorry I failed to predict the answers. I may be a bad astrologer!!

Accepted Answers:

- Try to obtain an additional set of features
- Get more training examples

---

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

- A. WRONG. The regularization parameter λ value is chosen so as to give the lowest training set error
  - If the objective is to lower the training-error, there is no point of regularization.
- **B. RIGHT: The regularization parameter λ value is chosen so as to give the lowest cross validation error**
  - choice-of-regularization-parameter
- **C**. WRONG: The regularization parameter λ value is chosen so as to give the lowest test set error
  - I assume they didn't mean test-error as valdation-error. We shouldn't tune any hyperparameter according to the performance in the test-error.
- **D. RIGHT. The performance of a learning algorithm on the training set will typically be better than its performance on the test set**
  - It is true for linear regression irrespective of regularized or not (So I am bit skeptic about why they gave this option in the context of regularized linear regression.)

8.

Consider the data provided below, which follows the linear regression model $h_w(x) = w_0 + w_1 x$, and cost function is MSE. Starting with initial weights $(w_0, w_1) = (1,1)$. What is the cost?

x	y
1	0.5
2	1
4	2
0	0

Answer: actual MSE is given below. But look like they have used **MSE/2**

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2$$

```
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

x = np.array([1,2, 4, 0])
y = np.array([0.5,1, 2, 0])

h = lambda w, x : w[0] + w[1]*x # Didn't use anywhere. Take it an example for lam

def mse(w):
  return 1.0/(2*len(x)) * sum([(yy-w[0]-w[1]*xx)**2 for xx,yy in zip(x,y)])

def gradient_mse(w):
  return np.array([
      1.0/(2*len(x)) * sum([2*(yy-w[0]-w[1]*xx)*-1   for xx,yy in zip(x,y)]),
      1.0/(2*len(x)) * sum([2*(yy-w[0]-w[1]*xx)*-xx for xx,yy in zip(x,y)])])

w = np.array([1,1])
print("ANSWER# 8 :", mse(w))

alpha = 0.1 # learning rate
updated_w = w-alpha*gradient_mse(w)
print("ANSWER# 9 :", updated_w)

print("ANSWER# 10 : %.4f" % mse(updated_w))
```