Close

Linear Regression with One Variable

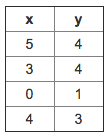
5 questions

1.

Consider the problem of predicting how well a student does in her second year of college/university, given how well they did in their first year.

Specifically, let x be equal to the number of "A" grades (including A-. A and A+ grades) that a student receives in their first year of college (freshmen year). We would like to predict the value of y, which we define as the number of "A" grades they get in their second year (sophomore year).

Questions 1 through 4 will use the following training set of a small sample of different students' performances. Here each row is one training example. Recall that in linear regression, our hypothesis is *hθ*(*x*)=*θ*0+*θ*1*x*, and we use *m* to denote the number of training examples.

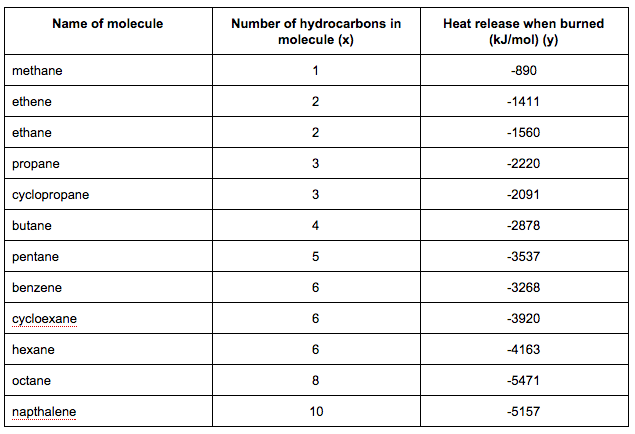


For the training set given above, what is the value of *m*? In the box below, please enter your answer (which should be a number between 0 and 10).



2.

Many substances that can burn (such as gasoline and alcohol) have a chemical structure based on carbon atoms; for this reason they are called hydrocarbons. A chemist wants to understand how the number of carbon atoms in a molecule affects how much energy is released when that molecule combusts (meaning that it is burned). The chemists obtains the dataset below. In the column on the right, “kJ/mol” is the unit measuring the amount of energy released.



You would like to use linear regression (*hθ*(*x*)=*θ*0+*θ*1*x*) to estimate the amount of energy released (y) as a function of the number of carbon atoms (x). Which of the following do you think will be the values you obtain for *θ*0 and *θ*1? You should be able to select the right answer without actually implementing linear regression.



*θ*0=−569.6,*θ*1=−530.9



*θ*0=−569.6,*θ*1=530.9



*θ*0=−1780.0,*θ*1=530.9



*θ*0=−1780.0,*θ*1=−530.9

3.

Suppose we set *θ*0=−2,*θ*1=0.5. What is *hθ*(6)?



4.

Let *f* be some function so that

*f*(*θ*0,*θ*1) outputs a number. For this problem,

*f* is some arbitrary/unknown smooth function (not necessarily the

cost function of linear regression, so *f* may have local optima).

Suppose we use gradient descent to try to minimize *f*(*θ*0,*θ*1)

as a function of *θ*0 and *θ*1. Which of the

following statements are true? (Check all that apply.)



No matter how *θ*0 and *θ*1 are initialized, so long

as *α* is sufficiently small, we can safely expect gradient descent to converge

to the same solution.



If the first few iterations of gradient descent cause *f*(*θ*0,*θ*1) to

**increase** rather than decrease, then the most likely cause is that we have set the

learning rate *α* to too large a value.



Setting the learning rate *α* to be very small is not harmful, and can

only speed up the convergence of gradient descent.



If *θ*0 and *θ*1 are initialized at

the global minimum, then one iteration will not change their values.

5.

Suppose that for some linear regression problem (say, predicting housing prices as in the lecture), we

have some training set, and for our training set we managed to find some *θ*0, *θ*1 such that *J*(*θ*0,*θ*1)=0. Which

of the statements below must then be true? (Check all that apply.)



We can perfectly predict the value of *y* even for new examples that we have not yet seen.

(e.g., we can perfectly predict prices of even new houses that we have not yet seen.)



For these values of *θ*0 and *θ*1 that satisfy *J*(*θ*0,*θ*1)=0,

we have that *hθ*(*x*(*i*))=*y*(*i*) for every training example (*x*(*i*),*y*(*i*))



For this to be true, we must have *θ*0=0 and *θ*1=0

so that *hθ*(*x*)=0



This is not possible: By the definition of *J*(*θ*0,*θ*1), it is not possible for there to exist

*θ*0 and *θ*1 so that *J*(*θ*0,*θ*1)=0

Submit Quiz

[Help Center](https://accounts.coursera.org/i/zendesk/courserahelp?return_to=https://learner.coursera.help/hc" \t "_blank)

Close

Linear Regression with One Variable

4/5 questions correct

Quiz passed!

[Continue Course](https://www.coursera.org/learn/machine-learning/lecture/38jIT/matrices-and-vectors)

[Back to Week 1](https://www.coursera.org/learn/machine-learning/home/week/1)

**Be Recognized for Your Achievements.** "Course Certificates give you the recognition you need to get the job, the material gives you the skills to do the job. It makes you look more valuable because you are more valuable." - Peter B., USA, Software Developer

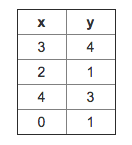
[**Showcase Your Accomplishment! Earn Your Course Certificate! $49**](https://www.coursera.org/certificate/machine-learning)

1.

Consider the problem of predicting how well a student does in her second year of college/university, given how well they did in their first year.

Specifically, let x be equal to the number of "A" grades (including A-. A and A+ grades) that a student receives in their first year of college (freshmen year). We would like to predict the value of y, which we define as the number of "A" grades they get in their second year (sophomore year).

Refer to the following training set of a small sample of different students' performances (note that this training set will also be referenced in other questions in this quiz). Here each row is one training example. Recall that in linear regression, our hypothesis is *hθ*(*x*)=*θ*0+*θ*1*x*, and we use *m* to denote the number of training examples.



For the training set given above, what is the value of *m*? In the box below, please enter your answer (which should be a number between 0 and 10).

2.

Consider the following training set of *m*=4 training examples:

|  |  |
| --- | --- |
| x | y |
| 1 | 0.5 |
| 2 | 1 |
| 4 | 2 |
| 0 | 0 |

Consider the linear regression model *hθ*(*x*)=*θ*0+*θ*1*x*. What are the values of *θ*0 and *θ*1 that you would expect to obtain upon running gradient descent on this model? (Linear regression will be able to fit this data perfectly.)

3.

Suppose we set *θ*0=−1,*θ*1=2. What is *hθ*(6)?

4.

Let *f* be some function so that

*f*(*θ*0,*θ*1) outputs a number. For this problem,

*f* is some arbitrary/unknown smooth function (not necessarily the

cost function of linear regression, so *f* may have local optima).

Suppose we use gradient descent to try to minimize *f*(*θ*0,*θ*1)

as a function of *θ*0 and *θ*1. Which of the

following statements are true? (Check all that apply.)

5.

Suppose that for some linear regression problem (say, predicting housing prices as in the lecture), we

have some training set, and for our training set we managed to find some *θ*0, *θ*1 such that *J*(*θ*0,*θ*1)=0. Which

of the statements below must then be true? (Check all that apply.)

[Help Center](https://accounts.coursera.org/i/zendesk/courserahelp?return_to=https://learner.coursera.help/hc" \t "_blank)

Close

Linear Regression with One Variable

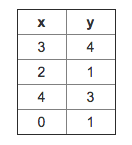
5 questions

1.

Consider the problem of predicting how well a student does in her second year of college/university, given how well they did in their first year.

Specifically, let x be equal to the number of "A" grades (including A-. A and A+ grades) that a student receives in their first year of college (freshmen year). We would like to predict the value of y, which we define as the number of "A" grades they get in their second year (sophomore year).

Refer to the following training set of a small sample of different students' performances (note that this training set will also be referenced in other questions in this quiz). Here each row is one training example. Recall that in linear regression, our hypothesis is *hθ*(*x*)=*θ*0+*θ*1*x*, and we use *m* to denote the number of training examples.



For the training set given above, what is the value of *m*? In the box below, please enter your answer (which should be a number between 0 and 10).



2.

For this question, continue to assume that we are

using the training set given above. Recall our definition of the

cost function was *J*(*θ*0,*θ*1)=12*m*∑*mi*=1(*hθ*(*x*(*i*))−*y*(*i*))2.

What is *J*(0,1)? In the box below,

please enter your answer (use decimals instead of fractions if necessary, e.g., 1.5).



3.

Suppose we set *θ*0=0,*θ*1=1.5. What is *hθ*(2)?



4.

Let *f* be some function so that

*f*(*θ*0,*θ*1) outputs a number. For this problem,

*f* is some arbitrary/unknown smooth function (not necessarily the

cost function of linear regression, so *f* may have local optima).

Suppose we use gradient descent to try to minimize *f*(*θ*0,*θ*1)

as a function of *θ*0 and *θ*1. Which of the

following statements are true? (Check all that apply.)



If *θ*0 and *θ*1 are initialized at

a local minimum, then one iteration will not change their values.



If the learning rate is too small, then gradient descent may take a very long

time to converge.



If *θ*0 and *θ*1 are initialized so that *θ*0=*θ*1, then by symmetry (because we do simultaneous updates to the two parameters), after one iteration of gradient descent, we will still have *θ*0=*θ*1.



Even if the learning rate *α* is very large, every iteration of

gradient descent will decrease the value of *f*(*θ*0,*θ*1).

5.

Suppose that for some linear regression problem (say, predicting housing prices as in the lecture), we

have some training set, and for our training set we managed to find some *θ*0, *θ*1 such that *J*(*θ*0,*θ*1)=0. Which

of the statements below must then be true? (Check all that apply.)



Our training set can be fit perfectly by a straight line,

i.e., all of our training examples lie perfectly on some straight line.



For this to be true, we must have *θ*0=0 and *θ*1=0

so that *hθ*(*x*)=0



For this to be true, we must have *y*(*i*)=0 for every value of *i*=1,2,…,*m*.



Gradient descent is likely to get stuck at a local minimum and fail to find the global minimum.

Submit Quiz

[Help Center](https://accounts.coursera.org/i/zendesk/courserahelp?return_to=https://learner.coursera.help/hc" \t "_blank)

Close

Linear Regression with Multiple Variables

5 questions

1.

Suppose *m*=4 students have taken some class, and the class had a midterm exam and a final exam. You have collected a dataset of their scores on the two exams, which is as follows:

|  |  |  |
| --- | --- | --- |
| midterm exam | (midterm exam)2 | final exam |
| 89 | 7921 | 96 |
| 72 | 5184 | 74 |
| 94 | 8836 | 87 |
| 69 | 4761 | 78 |

You'd like to use polynomial regression to predict a student's final exam score from their midterm exam score. Concretely, suppose you want to fit a model of the form *hθ*(*x*)=*θ*0+*θ*1*x*1+*θ*2*x*2, where *x*1 is the midterm score and *x*2 is (midterm score)2. Further, you plan to use both feature scaling (dividing by the "max-min", or range, of a feature) and mean normalization.

What is the normalized feature *x*(3)1? (Hint: midterm = 89, final = 96 is training example 1.) Please round off your answer to two decimal places and enter in the text box below.

 WRONG, rest are right

2.

You run gradient descent for 15 iterations

with *α*=0.3 and compute *J*(*θ*) after each

iteration. You find that the value of *J*(*θ*) **increases** over

time. Based on this, which of the following conclusions seems

most plausible?



*α*=0.3 is an effective choice of learning rate.



Rather than use the current value of *α*, it'd be more promising to try a smaller value of *α*(say *α*=0.1).



Rather than use the current value of *α*, it'd be more promising to try a larger value of *α*(say *α*=1.0).

3.

Suppose you have *m*=23 training examples with *n*=5 features (excluding the additional all-ones feature for the intercept term, which you should add). The normal equation is *θ*=(*XTX*)−1*XTy*. For the given values of *m* and *n*, what are the dimensions of *θ*, *X*, and *y* in this equation?



*X* is 23×5, *y* is 23×1, *θ* is 5×5



*X* is 23×6, *y* is 23×1, *θ* is 6×1



*X* is 23×6, *y* is 23×6, *θ* is 6×6



*X* is 23×5, *y* is 23×1, *θ* is 5×1

4.

Suppose you have a dataset with *m*=50 examples and *n*=200000 features for each example. You want to use multivariate linear regression to fit the parameters *θ* to our data. Should you prefer gradient descent or the normal equation?



Gradient descent, since (*XTX*)−1 will be very slow to compute in the normal equation.



Gradient descent, since it will always converge to the optimal *θ*.



The normal equation, since it provides an efficient way to directly find the solution.



The normal equation, since gradient descent might be unable to find the optimal *θ*.

5.

Which of the following are reasons for using feature scaling?



It speeds up solving for *θ* using the normal equation.



It prevents the matrix *XTX* (used in the normal equation) from being non-invertable (singular/degenerate).



It is necessary to prevent gradient descent from getting stuck in local optima.



It speeds up gradient descent by making it require fewer iterations to get to a good solution.

Submit Quiz

[Help Center](https://accounts.coursera.org/i/zendesk/courserahelp?return_to=https://learner.coursera.help/hc" \t "_blank)

Close

Linear Regression with Multiple Variables

5 questions

1.

Suppose *m*=4 students have taken some class, and the class had a midterm exam and a final exam. You have collected a dataset of their scores on the two exams, which is as follows:

|  |  |  |
| --- | --- | --- |
| midterm exam | (midterm exam)2 | final exam |
| 89 | 7921 | 96 |
| 72 | 5184 | 74 |
| 94 | 8836 | 87 |
| 69 | 4761 | 78 |

You'd like to use polynomial regression to predict a student's final exam score from their midterm exam score. Concretely, suppose you want to fit a model of the form *hθ*(*x*)=*θ*0+*θ*1*x*1+*θ*2*x*2, where *x*1 is the midterm score and *x*2 is (midterm score)2. Further, you plan to use both feature scaling (dividing by the "max-min", or range, of a feature) and mean normalization.

What is the normalized feature *x*(3)1? (Hint: midterm = 89, final = 96 is training example 1.) Please round off your answer to two decimal places and enter in the text box below.



2.

You run gradient descent for 15 iterations

with *α*=0.3 and compute *J*(*θ*) after each

iteration. You find that the value of *J*(*θ*) **increases** over

time. Based on this, which of the following conclusions seems

most plausible?



*α*=0.3 is an effective choice of learning rate.



Rather than use the current value of *α*, it'd be more promising to try a larger value of *α*(say *α*=1.0).



Rather than use the current value of *α*, it'd be more promising to try a smaller value of *α*(say *α*=0.1).

3.

Suppose you have *m*=14 training examples with *n*=3 features (excluding the additional all-ones feature for the intercept term, which you should add). The normal equation is *θ*=(*XTX*)−1*XTy*. For the given values of *m* and *n*, what are the dimensions of *θ*, *X*, and *y* in this equation?



*X* is 14×3, *y* is 14×1, *θ* is 3×1



*X* is 14×3, *y* is 14×1, *θ* is 3×3



*X* is 14×4, *y* is 14×4, *θ* is 4×4



*X* is 14×4, *y* is 14×1, *θ* is 4×1

4.

Suppose you have a dataset with *m*=1000000 examples and *n*=200000 features for each example. You want to use multivariate linear regression to fit the parameters *θ* to our data. Should you prefer gradient descent or the normal equation?



The normal equation, since it provides an efficient way to directly find the solution.



The normal equation, since gradient descent might be unable to find the optimal *θ*.



Gradient descent, since it will always converge to the optimal *θ*.



Gradient descent, since (*XTX*)−1 will be very slow to compute in the normal equation.

5.

Which of the following are reasons for using feature scaling?



It speeds up solving for *θ* using the normal equation.



It prevents the matrix *XTX* (used in the normal equation) from being non-invertable (singular/degenerate).



It speeds up gradient descent by making it require fewer iterations to get to a good solution.



It is necessary to prevent gradient descent from getting stuck in local optima.

Submit Quiz

[Help Center](https://accounts.coursera.org/i/zendesk/courserahelp?return_to=https://learner.coursera.help/hc" \t "_blank)

Close

# Octave Tutorial

5 questions

1.

Suppose I first execute the following Octave commands:

A = [1 2; 3 4; 5 6];

B = [1 2 3; 4 5 6];

Which of the following are then valid Octave commands? Check all that apply and assume all options are written in an Octave command. (Hint: A' denotes the transpose of A.)



C = A' + B;



C = B \* A;



C = A + B;



C = B' \* A;

2.

Let *A*=⎡⎣⎢⎢16594211714310615138121⎤⎦⎥⎥.

Which of the following indexing expressions gives *B*=⎡⎣⎢⎢16594211714⎤⎦⎥⎥? Check all that apply.



B = A(:, 1:2);



B = A(1:4, 1:2);



B = A(:, 0:2);



B = A(0:4, 0:2);

3.

Let *A* be a 10x10 matrix and*x* be a 10-element vector. Your friend wants to compute the product *Ax*and writes the following code:

v = zeros(10, 1);

for i = 1:10

for j = 1:10

v(i) = v(i) + A(i, j) \* x(j);

end

end

How would you vectorize this code to run without any for loops? Check all that apply.



v = A \* x;



v = Ax;



v = x' \* A;



v = sum (A \* x);

4.

Say you have two column vectors *v* and *w*, each with 7 elements (i.e., they have dimensions 7x1). Consider the following code:

z = 0;

for i = 1:7

z = z + v(i) \* w(i)

end

Which of the following vectorizations correctly compute z? Check all that apply.



z = sum (v .\* w);



z = w' \* v;



z = v \* w;



z = w \* v;

5.

In Octave, many functions work on single numbers, vectors, and matrices. For example, the sin function when applied to a matrix will return a new matrix with the sin of each element. But you have to be careful, as certain functions have different behavior. Suppose you have an 7x7 matrix *X*. You want to compute the log of every element, the square of every element, add 1 to every element, and divide every element by 4. You will store the results in four matrices, *A*,*B*,*C*,*D*. One way to do so is the following code:

for i = 1:7

for j = 1:7

A(i, j) = log(X(i, j));

B(i, j) = X(i, j) ^ 2;

C(i, j) = X(i, j) + 1;

D(i, j) = X(i, j) / 4;

end

end

Which of the following correctly compute *A*,*B*,*C*, or *D*? Check all that apply.



C = X + 1;



D = X / 4;



A = log (X);



B = X ^ 2;

Submit Quiz

[Help Center](https://accounts.coursera.org/i/zendesk/courserahelp?return_to=https://learner.coursera.help/hc" \t "_blank)

Close

# Logistic Regression

5 questions

1.

Suppose that you have trained a logistic regression classifier, and it outputs on a new example *x* a prediction *hθ*(*x*) = 0.7. This means (check all that apply):



Our estimate for *P*(*y*=0|*x*;*θ*) is 0.3.



Our estimate for *P*(*y*=0|*x*;*θ*) is 0.7.



Our estimate for *P*(*y*=1|*x*;*θ*) is 0.7.



Our estimate for *P*(*y*=1|*x*;*θ*) is 0.3.

2.

Suppose you have the following training set, and fit a logistic regression classifier *hθ*(*x*)=*g*(*θ*0+*θ*1*x*1+*θ*2*x*2).

Which of the following are true? Check all that apply.



*J*(*θ*) will be a convex function, so gradient descent should converge to the global minimum.



Adding polynomial features (e.g., instead using *hθ*(*x*)=*g*(*θ*0+*θ*1*x*1+*θ*2*x*2+*θ*3*x*21+*θ*4*x*1*x*2+*θ*5*x*22) ) could increase how well we can fit the training data.



The positive and negative examples cannot be separated using a straight line. So, gradient descent will fail to converge.



Because the positive and negative examples cannot be separated using a straight line, linear regression will perform as well as logistic regression on this data.

3.

For logistic regression, the gradient is given by ∂∂*θjJ*(*θ*)=∑*mi*=1(*hθ*(*x*(*i*))−*y*(*i*))*x*(*i*)*j*. Which of these is a correct gradient descent update for logistic regression with a learning rate of*α*? Check all that apply.



*θj*:=*θj*−*α*1*m*∑*mi*=1(*hθ*(*x*(*i*))−*y*(*i*))*x*(*i*)*j* (simultaneously update for all *j*).



*θj*:=*θj*−*α*1*m*∑*mi*=1(*hθ*(*x*(*i*))−*y*(*i*))*x*(*i*) (simultaneously update for all *j*).



*θj*:=*θj*−*α*1*m*∑*mi*=1(11+*e*−*θTx*(*i*)−*y*(*i*))*x*(*i*)*j* (simultaneously update for all *j*).



*θ*:=*θ*−*α*1*m*∑*mi*=1(*θTx*−*y*(*i*))*x*(*i*).

4.  \*\* Wrong \*\*

Which of the following statements are true? Check all that apply.



The cost function *J*(*θ*) for logistic regression trained with *m*≥1 examples is always greater than or equal to zero.



For logistic regression, sometimes gradient descent will converge to a local minimum (and fail to find the global minimum). This is the reason we prefer more advanced optimization algorithms such as fminunc (conjugate gradient/BFGS/L-BFGS/etc).



The sigmoid function *g*(*z*)=11+*e*−*z* is never greater than one (>1).



Linear regression always works well for classification if you classify by using a threshold on the prediction made by linear regression.

5.

Suppose you train a logistic classifier *hθ*(*x*)=*g*(*θ*0+*θ*1*x*1+*θ*2*x*2). Suppose *θ*0=−6,*θ*1=0,*θ*2=1. Which of the following figures represents the decision boundary found by your classifier?



Figure:



Figure:



Figure:

x2 = 6, y = 1 above

Figure:

Submit Quiz

[Help Center](https://accounts.coursera.org/i/zendesk/courserahelp?return_to=https://learner.coursera.help/hc" \t "_blank)

Close

# Logistic Regression

5 questions TAKE 2, first 4 Qs seem identical,

1.

Suppose that you have trained a logistic regression classifier, and it outputs on a new example *x* a prediction *hθ*(*x*) = 0.7. This means (check all that apply):



Our estimate for *P*(*y*=1|*x*;*θ*) is 0.7.



Our estimate for *P*(*y*=0|*x*;*θ*) is 0.3.



Our estimate for *P*(*y*=0|*x*;*θ*) is 0.7.



Our estimate for *P*(*y*=1|*x*;*θ*) is 0.3.

2.

Suppose you have the following training set, and fit a logistic regression classifier *hθ*(*x*)=*g*(*θ*0+*θ*1*x*1+*θ*2*x*2).

Which of the following are true? Check all that apply.



*J*(*θ*) will be a convex function, so gradient descent should converge to the global minimum.



Adding polynomial features (e.g., instead using *hθ*(*x*)=*g*(*θ*0+*θ*1*x*1+*θ*2*x*2+*θ*3*x*21+*θ*4*x*1*x*2+*θ*5*x*22) ) could increase how well we can fit the training data.



The positive and negative examples cannot be separated using a straight line. So, gradient descent will fail to converge.



Because the positive and negative examples cannot be separated using a straight line, linear regression will perform as well as logistic regression on this data.

3.

For logistic regression, the gradient is given by ∂∂*θjJ*(*θ*)=∑*mi*=1(*hθ*(*x*(*i*))−*y*(*i*))*x*(*i*)*j*. Which of these is a correct gradient descent update for logistic regression with a learning rate of*α*? Check all that apply.



*θ*:=*θ*−*α*1*m*∑*mi*=1(*θTx*−*y*(*i*))*x*(*i*).



*θj*:=*θj*−*α*1*m*∑*mi*=1(*hθ*(*x*(*i*))−*y*(*i*))*x*(*i*) (simultaneously update for all *j*).



*θj*:=*θj*−*α*1*m*∑*mi*=1(11+*e*−*θTx*(*i*)−*y*(*i*))*x*(*i*)*j* (simultaneously update for all *j*).



*θj*:=*θj*−*α*1*m*∑*mi*=1(*hθ*(*x*(*i*))−*y*(*i*))*x*(*i*)*j* (simultaneously update for all *j*).

4.

Which of the following statements are true? Check all that apply.



The sigmoid function *g*(*z*)=11+*e*−*z* is never greater than one (>1).



The cost function *J*(*θ*) for logistic regression trained with *m*≥1 examples is always greater than or equal to zero.



Linear regression always works well for classification if you classify by using a threshold on the prediction made by linear regression.



For logistic regression, sometimes gradient descent will converge to a local minimum (and fail to find the global minimum). This is the reason we prefer more advanced optimization algorithms such as fminunc (conjugate gradient/BFGS/L-BFGS/etc).

5.

Suppose you train a logistic classifier *hθ*(*x*)=*g*(*θ*0+*θ*1*x*1+*θ*2*x*2). Suppose *θ*0=6,*θ*1=−1,*θ*2=0. Which of the following figures represents the decision boundary found by your classifier?

Y = 1 to the left of the line at x1=6

Figure:



Figure:



Figure:



Figure:

Submit Quiz

[Help Center](https://accounts.coursera.org/i/zendesk/courserahelp?return_to=https://learner.coursera.help/hc" \t "_blank)

Close

# Regularization

5 questions

1.

You are training a classification model with logistic

regression. Which of the following statements are true? Check

all that apply.



Adding many new features to the model helps prevent overfitting on the training set.



Introducing regularization to the model always results in equal or better performance on examples not in the training set.



Adding a new feature to the model always results in equal or better performance on the training set.



Introducing regularization to the model always results in equal or better performance on the training set.

2.

Suppose you ran logistic regression twice, once with *λ*=0, and once with *λ*=1. One of the times, you got

parameters *θ*=[81.4712.69], and the other time you got

*θ*=[13.010.91]. However, you forgot which value of

*λ* corresponds to which value of *θ*. Which one do you

think corresponds to *λ*=1?



*θ*=[13.010.91]



*θ*=[81.4712.69]

3.

Which of the following statements about regularization are

true? Check all that apply.



Using too large a value of *λ* can cause your hypothesis to overfit the data; this can be avoided by reducing *λ*.



Consider a classification problem. Adding regularization may cause your classifier to incorrectly classify some training examples (which it had correctly classified when not using regularization, i.e. when *λ*=0).



Because logistic regression outputs values 0≤*hθ*(*x*)≤1, it's range of output values can only be "shrunk" slightly by regularization anyway, so regularization is generally not helpful for it.



Using a very large value of *λ* cannot hurt the performance of your hypothesis; the only reason we do not set *λ* to be too large is to avoid numerical problems.

4.

In which one of the following figures do you think the hypothesis has overfit the training set?

Fits perfectly a misshapen W.

Figure:



Figure:



Figure:



Figure:

5.

In which one of the following figures do you think the hypothesis has underfit the training set?

Data make a deep bowl, hypothesis a shallow one.

Figure:



Figure:



Figure:



Figure:

Submit Quiz

[Help Center](https://accounts.coursera.org/i/zendesk/courserahelp?return_to=https://learner.coursera.help/hc" \t "_blank)

Close

# Neural Networks: Representation

5 questions \*\* 3 right

1. \*\*\*\* Wrong

Which of the following statements are true? Check all that apply.



In a neural network with many layers, we think of each successive layer as being able to use the earlier layers as features, so as to be able to compute increasingly complex functions.



If a neural network is overfitting the data, one solution would be to decrease the regularization parameter *λ*.



Suppose you have a multi-class classification problem with three classes, trained with a 3 layer network. Let *a*(3)1=(*h*Θ(*x*))1 be the activation of the first output unit, and similarly *a*(3)2=(*h*Θ(*x*))2 and *a*(3)3=(*h*Θ(*x*))3. Then for any input *x*, it must be the case that *a*(3)1+*a*(3)2+*a*(3)3=1.



If a neural network is overfitting the data, one solution would be to increase the regularization parameter *λ*.

2.

Consider the following neural network which takes two binary-valued inputs *x*1,*x*2∈{0,1} and outputs *h*Θ(*x*). Which of the following logical functions does it (approximately) compute?



AND



NAND (meaning "NOT AND")



OR



XOR (exclusive OR)

3. \*\*\* Wrong

Consider the neural network given below. Which of the following equations correctly computes the activation *a*(3)1? Note: *g*(*z*) is the sigmoid activation function.



*a*(3)1=*g*(Θ(2)1,0*a*(2)0+Θ(2)1,1*a*(2)1+Θ(2)1,2*a*(2)2)



*a*(3)1=*g*(Θ(2)1,0*a*(1)0+Θ(2)1,1*a*(1)1+Θ(2)1,2*a*(1)2)



*a*(3)1=*g*(Θ(1)1,0*a*(2)0+Θ(1)1,1*a*(2)1+Θ(1)1,2*a*(2)2)



*a*(3)1=*g*(Θ(2)2,0*a*(2)0+Θ(2)2,1*a*(2)1+Θ(2)2,2*a*(2)2)

4.

You have the following neural network:

You'd like to compute the activations of the hidden layer *a*(2)∈R3. One way to do so is the following Octave code:

You want to have a vectorized implementation of this (i.e., one that does not use for loops). Which of the following implementations correctly compute *a*(2)? Check all that apply.



z = Theta1 \* x; a2 = sigmoid (z);



a2 = sigmoid (x \* Theta1);



a2 = sigmoid (Theta2 \* x);



z = sigmoid(x); a2 = sigmoid (Theta1 \* z);

5.

You are using the neural network pictured below and have learned the parameters Θ(1)=[110.51.21.92.7] (used to compute *a*(2)) and Θ(2)=[1−0.2−1.7] (used to compute *a*(3)} as a function of *a*(2)). Suppose you swap the parameters for the first hidden layer between its two units so Θ(1)=[111.20.52.71.9] and also swap the output layer so Θ(2)=[1−1.7−0.2]. How will this change the value of the output *h*Θ(*x*)?



It will stay the same.



It will increase.



It will decrease



Insufficient information to tell: it may increase or decrease.

Submit Quiz

[Help Center](https://accounts.coursera.org/i/zendesk/courserahelp?return_to=https://learner.coursera.help/hc" \t "_blank)

Close

# Neural Networks: Representation

5 questions

1. WRONG????

Which of the following statements are true? Check all that apply.



If a neural network is overfitting the data, one solution would be to increase the regularization parameter *λ*.



If a neural network is overfitting the data, one solution would be to decrease the regularization parameter *λ*.



In a neural network with many layers, we think of each successive layer as being able to use the earlier layers as features, so as to be able to compute increasingly complex functions.



Suppose you have a multi-class classification problem with three classes, trained with a 3 layer network. Let *a*(3)1=(*h*Θ(*x*))1 be the activation of the first output unit, and similarly *a*(3)2=(*h*Θ(*x*))2 and *a*(3)3=(*h*Θ(*x*))3. Then for any input *x*, it must be the case that *a*(3)1+*a*(3)2+*a*(3)3=1.

2.

Consider the following neural network which takes two binary-valued inputs *x*1,*x*2∈{0,1} and outputs *h*Θ(*x*). Which of the following logical functions does it (approximately) compute?

-20,30,30



OR



AND



NAND (meaning "NOT AND")



XOR (exclusive OR)

3.

Consider the neural network given below. Which of the following equations correctly computes the activation *a*(3)1? Note: *g*(*z*) is the sigmoid activation function.



*a*(3)1=*g*(Θ(2)1,0*a*(2)0+Θ(2)1,1*a*(2)1+Θ(2)1,2*a*(2)2)



*a*(3)1=*g*(Θ(2)1,0*a*(1)0+Θ(2)1,1*a*(1)1+Θ(2)1,2*a*(1)2)



*a*(3)1=*g*(Θ(1)1,0*a*(2)0+Θ(1)1,1*a*(2)1+Θ(1)1,2*a*(2)2)



*a*(3)1=*g*(Θ(2)2,0*a*(2)0+Θ(2)2,1*a*(2)1+Θ(2)2,2*a*(2)2)

4.

You have the following neural network:

You'd like to compute the activations of the hidden layer *a*(2)∈R3. One way to do so is the following Octave code:

You want to have a vectorized implementation of this (i.e., one that does not use for loops). Which of the following implementations correctly compute *a*(2)? Check all that apply.



a2 = sigmoid (Theta1 \* x);



a2 = sigmoid (x \* Theta1);



a2 = sigmoid (Theta2 \* x);



z = sigmoid(x); a2 = Theta1 \* z;

5.

You are using the neural network pictured below and have learned the parameters Θ(1)=[1111.72.43.2] (used to compute *a*(2)) and Θ(2)=[10.3−1.2] (used to compute*a*(3)} as a function of *a*(2)). Suppose you swap the parameters for the first hidden layer between its two units so Θ(1)=[111.713.22.4] and also swap the output layer so Θ(2)=[1−1.20.3]. How will this change the value of the output *h*Θ(*x*)?



It will stay the same.



It will increase.



It will decrease



Insufficient information to tell: it may increase or decrease.

Submit Quiz

[Help Center](https://accounts.coursera.org/i/zendesk/courserahelp?return_to=https://learner.coursera.help/hc" \t "_blank)

Close

# Neural Networks: Representation

5 questionsWRONG <sigh> Suck!!!

1.

Which of the following statements are true? Check all that apply.



Suppose you have a multi-class classification problem with three classes, trained with a 3 layer network. Let *a*(3)1=(*h*Θ(*x*))1 be the activation of the first output unit, and similarly *a*(3)2=(*h*Θ(*x*))2 and *a*(3)3=(*h*Θ(*x*))3. Then for any input *x*, it must be the case that *a*(3)1+*a*(3)2+*a*(3)3=1.



A two layer (one input layer, one output layer; no hidden layer) neural network can represent the XOR function.



The activation values of the hidden units in a neural network, with the sigmoid activation function applied at every layer, are always in the range (0, 1).



Any logical function over binary-valued (0 or 1) inputs *x*1 and *x*2 can be (approximately) represented using some neural network.

2.

Consider the following neural network which takes two binary-valued inputs *x*1,*x*2∈{0,1} and outputs *h*Θ(*x*). Which of the following logical functions does it (approximately) compute?



NAND (meaning "NOT AND")



AND



OR



XOR (exclusive OR)

3.

Consider the neural network given below. Which of the following equations correctly computes the activation *a*(3)1? Note: *g*(*z*) is the sigmoid activation function.



*a*(3)1=*g*(Θ(2)1,0*a*(2)0+Θ(2)1,1*a*(2)1+Θ(2)1,2*a*(2)2)



*a*(3)1=*g*(Θ(2)1,0*a*(1)0+Θ(2)1,1*a*(1)1+Θ(2)1,2*a*(1)2)



*a*(3)1=*g*(Θ(1)1,0*a*(2)0+Θ(1)1,1*a*(2)1+Θ(1)1,2*a*(2)2)



*a*(3)1=*g*(Θ(2)2,0*a*(2)0+Θ(2)2,1*a*(2)1+Θ(2)2,2*a*(2)2)

4.

You have the following neural network:

You'd like to compute the activations of the hidden layer *a*(2)∈R3. One way to do so is the following Octave code:

You want to have a vectorized implementation of this (i.e., one that does not use for loops). Which of the following implementations correctly compute *a*(2)? Check all that apply.



a2 = sigmoid (Theta1 \* x);



a2 = sigmoid (x \* Theta1);



a2 = sigmoid (Theta2 \* x);



z = sigmoid(x); a2 = Theta1 \* z;

5.

You are using the neural network pictured below and have learned the parameters Θ(1)=[11−1.55.13.72.3] (used to compute *a*(2)) and Θ(2)=[10.6−0.8] (used to compute *a*(3)} as a function of *a*(2)). Suppose you swap the parameters for the first hidden layer between its two units so Θ(1)=[115.1−1.52.33.7] and also swap the output layer so Θ(2)=[1−0.80.6]. How will this change the value of the output *h*Θ(*x*)?



It will stay the same.



It will increase.



It will decrease



Insufficient information to tell: it may increase or decrease.

Submit Quiz

[Help Center](https://accounts.coursera.org/i/zendesk/courserahelp?return_to=https://learner.coursera.help/hc" \t "_blank)

Close

# Neural Networks: Learning

5 questions

1. WRONG

You are training a three layer neural network and would like to use backpropagation to compute the gradient of the cost function. In the backpropagation algorithm, one of the steps is to update

Δ(2)*ij*:=Δ(2)*ij*+*δ*(3)*i*∗(*a*(2))*j*

for every *i*,*j*. Which of the following is a correct vectorization of this step?



Δ(2):=Δ(2)+(*a*(3))*T*∗*δ*(2)



Δ(2):=Δ(2)+(*a*(2))*T*∗*δ*(3)



Δ(2):=Δ(2)+*δ*(3)∗(*a*(3))*T*



Δ(2):=Δ(2)+*δ*(3)∗(*a*(2))*T*

2.

Suppose Theta1 is a 5x3 matrix, and Theta2 is a 4x6 matrix. You set thetaVec=[Theta1(:);Theta2(:)]. Which of the following correctly recovers Theta2?



reshape(thetaVec(16:39),4,6)



reshape(thetaVec(15:38),4,6)



reshape(thetaVec(16:24),4,6)



reshape(thetaVec(15:39),4,6)



reshape(thetaVec(16:39),6,4)

3. WRONG

Let *J*(*θ*)=2*θ*4+2. Let *θ*=1, and *ϵ*=0.01. Use the formula *J*(*θ*+*ϵ*)−*J*(*θ*−*ϵ*)2*ϵ* to numerically compute an approximation to the derivative at *θ*=1. What value do you get? (When *θ*=1, the true/exact derivative is *dJ*(*θ*)*dθ*=8.)



7.9992



8.0008



10



8

4.

Which of the following statements are true? Check all that apply.



If our neural network overfits the training set, one reasonable step to take is to increase the regularization parameter *λ*.



Using a large value of *λ* cannot hurt the performance of your neural network; the only reason we do not set *λ* to be too large is to avoid numerical problems.



Gradient checking is useful if we are using gradient descent as our optimization algorithm. However, it serves little purpose if we are using one of the advanced optimization methods (such as in fminunc).



Using gradient checking can help verify if one's implementation of backpropagation is bug-free.

5.

Which of the following statements are true? Check all that apply.



Suppose we have a correct implementation of backpropagation, and are training a neural network using gradient descent. Suppose we plot *J*(Θ) as a function of the number of iterations, and find that it is **increasing** rather than decreasing. One possible cause of this is that the learning rate *α* is too large.



If we are training a neural network using gradient descent, one reasonable "debugging" step to make sure it is working is to plot *J*(Θ) as a function of the number of iterations, and make sure it is decreasing (or at least non-increasing) after each iteration.



Suppose we are using gradient descent with learning rate *α*. For logistic regression and linear regression, *J*(*θ*) was a convex optimization problem and thus we did not want to choose a learning rate *α* that is too large. For a neural network however, *J*(Θ) may not be convex, and thus choosing a very large value of *α* can only speed up convergence.



Suppose that the parameter Θ(1) is a square matrix (meaning the number of rows equals the number of columns). If we replace Θ(1) with its transpose (Θ(1))*T*, then we have not changed the function that the network is computing.

Submit Quiz

[Help Center](https://accounts.coursera.org/i/zendesk/courserahelp?return_to=https://learner.coursera.help/hc" \t "_blank)

Close

# Neural Networks: Learning

5 questions

1.

You are training a three layer neural network and would like to use backpropagation to compute the gradient of the cost function. In the backpropagation algorithm, one of the steps is to update

Δ(2)*ij*:=Δ(2)*ij*+*δ*(3)*i*∗(*a*(2))*j*

for every *i*,*j*. Which of the following is a correct vectorization of this step?



Δ(2):=Δ(2)+*δ*(3)∗(*a*(3))*T*



Δ(2):=Δ(2)+*δ*(3)∗(*a*(2))*T*



Δ(2):=Δ(2)+(*a*(3))*T*∗*δ*(3)



Δ(2):=Δ(2)+*δ*(2)∗(*a*(2))*T*

2.

Suppose Theta1 is a 5x3 matrix, and Theta2 is a 4x6 matrix. You set thetaVec=[Theta1(:);Theta2(:)]. Which of the following correctly recovers Theta2?



reshape(thetaVec(16:39),4,6)



reshape(thetaVec(15:38),4,6)



reshape(thetaVec(16:24),4,6)



reshape(thetaVec(15:39),4,6)



reshape(thetaVec(16:39),6,4)

3.

Let *J*(*θ*)=3*θ*3+2. Let *θ*=1, and *ϵ*=0.01. Use the formula *J*(*θ*+*ϵ*)−*J*(*θ*−*ϵ*)2*ϵ* to numerically compute an approximation to the derivative at *θ*=1. What value do you get? (When *θ*=1, the true/exact derivative is *dJ*(*θ*)*dθ*=9.)



11



9



9.0003



8.9997

4.

Which of the following statements are true? Check all that apply.



Gradient checking is useful if we are using gradient descent as our optimization algorithm. However, it serves little purpose if we are using one of the advanced optimization methods (such as in fminunc).



Using gradient checking can help verify if one's implementation of backpropagation is bug-free.



If our neural network overfits the training set, one reasonable step to take is to increase the regularization parameter *λ*.



Using a large value of *λ* cannot hurt the performance of your neural network; the only reason we do not set *λ* to be too large is to avoid numerical problems.

5.

Which of the following statements are true? Check all that apply.



Suppose that the parameter Θ(1) is a square matrix (meaning the number of rows equals the number of columns). If we replace Θ(1) with its transpose (Θ(1))*T*, then we have not changed the function that the network is computing.



Suppose we are using gradient descent with learning rate *α*. For logistic regression and linear regression, *J*(*θ*) was a convex optimization problem and thus we did not want to choose a learning rate *α* that is too large. For a neural network however, *J*(Θ) may not be convex, and thus choosing a very large value of *α* can only speed up convergence.



Suppose we have a correct implementation of backpropagation, and are training a neural network using gradient descent. Suppose we plot *J*(Θ) as a function of the number of iterations, and find that it is **increasing** rather than decreasing. One possible cause of this is that the learning rate *α* is too large.



If we are training a neural network using gradient descent, one reasonable "debugging" step to make sure it is working is to plot *J*(Θ) as a function of the number of iterations, and make sure it is decreasing (or at least non-increasing) after each iteration.

Submit Quiz

[Help Center](https://accounts.coursera.org/i/zendesk/courserahelp?return_to=https://learner.coursera.help/hc" \t "_blank)

Close

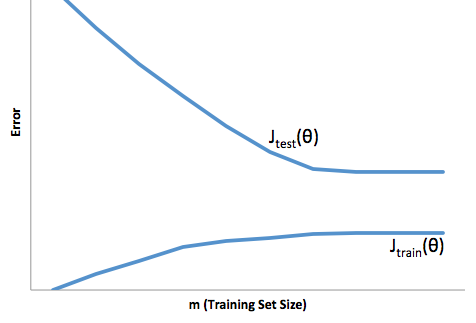
# Advice for Applying Machine Learning

# Boom! 5/5 first time.

5 questions

1.

You train a learning algorithm, and find that it has unacceptably high error on the test set. You plot the learning curve, and obtain the figure below. Is the algorithm suffering from high bias, high variance, or neither?





High variance



High bias



Neither

2.

Suppose you have implemented regularized logistic regression

to classify what object is in an image (i.e., to do object

recognition). However, when you test your hypothesis on a new

set of images, you find that it makes unacceptably large

errors with its predictions on the new images. However, your

hypothesis performs **well** (has low error) on the

training set. Which of the following are promising steps to

take? Check all that apply.



Try using a smaller set of features.



Use fewer training examples.



Get more training examples.



Try adding polynomial features.

3.

Suppose you have implemented regularized logistic regression

to predict what items customers will purchase on a web

shopping site. However, when you test your hypothesis on a new

set of customers, you find that it makes unacceptably large

errors in its predictions. Furthermore, the hypothesis

performs **poorly** on the training set. Which of the

following might be promising steps to take? Check all that

apply.



Try adding polynomial features.



Try increasing the regularization parameter *λ*.



Try using a smaller set of features.



Try to obtain and use additional features.

4.

Which of the following statements are true? Check all that apply.



Suppose you are training a logistic regression classifier using polynomial features and want to select what degree polynomial (denoted *d* in the lecture videos) to use. After training the classifier on the entire training set, you decide to use a subset of the training examples as a validation set. This will work just as well as having a validation set that is separate (disjoint) from the training set.



A typical split of a dataset into training, validation and test sets might be 60% training set, 20% validation set, and 20% test set.



Suppose you are using linear regression to predict housing prices, and your dataset comes sorted in order of increasing sizes of houses. It is then important to randomly shuffle the dataset before splitting it into training, validation and test sets, so that we don’t have all the smallest houses going into the training set, and all the largest houses going into the test set.



It is okay to use data from the test set to choose the regularization parameter *λ*, but not the model parameters (*θ*).

5.

Which of the following statements are true? Check all that apply.



When debugging learning algorithms, it is useful to plot a learning curve to understand if there is a high bias or high variance problem.



A model with more parameters is more prone to overfitting and typically has higher variance.



If a neural network has much lower training error than test error, then adding more layers will help bring the test error down because we can fit the test set better.



If a learning algorithm is suffering from high bias, only adding more training examples may **not** improve the test error significantly.

Submit Quiz

[Help Center](https://accounts.coursera.org/i/zendesk/courserahelp?return_to=https://learner.coursera.help/hc" \t "_blank)

Close

Machine Learning System Design

BAM! 100% first time again.

5 questions

1.

You are working on a spam classification system using regularized logistic regression. "Spam" is a positive class (y = 1) and "not spam" is the negative class (y = 0). You have trained your classifier and there are m = 1000 examples in the cross-validation set. The chart of predicted class vs. actual class is:

|  |  |  |
| --- | --- | --- |
|  | **Actual Class: 1** | **Actual Class: 0** |
| **Predicted Class: 1** | 85 | 890 |
| **Predicted Class: 0** | 15 | 10 |

For reference:

* Accuracy = (true positives + true negatives) / (total examples)
* Precision = (true positives) / (true positives + false positives)
* Recall = (true positives) / (true positives + false negatives)
* *F*1 score = (2 \* precision \* recall) / (precision + recall)

What is the classifier's recall (as a value from 0 to 1)?

Enter your answer in the box below. If necessary, provide at least two values after the decimal point.



2.

Suppose a massive dataset is available for training a learning algorithm. Training on a lot of data is likely to give good performance when two of the following conditions hold true.

Which are the two?



A human expert on the application domain

can confidently predict *y* when given only the features *x*

(or more generally, if we have some way to be confident

that *x* contains sufficient information to predict *y*

accurately).



When we are willing to include high

order polynomial features of *x* (such as *x*21, *x*22,

*x*1*x*2, etc.).



Our learning algorithm is able to

represent fairly complex functions (for example, if we

train a neural network or other model with a large

number of parameters).



The classes are not too skewed.

3.

Suppose you have trained a logistic regression classifier which is outputing *hθ*(*x*).

Currently, you predict 1 if *hθ*(*x*)≥threshold, and predict 0 if *hθ*(*x*)<threshold, where currently the threshold is set to 0.5.

Suppose you **increase** the threshold to 0.7. Which of the following are true? Check all that apply.



The classifier is likely to have unchanged precision and recall, but

higher accuracy.



The classifier is likely to have unchanged precision and recall, and

thus the same *F*1 score.



The classifier is likely to now have higher precision.



The classifier is likely to now have higher recall.

4.

Suppose you are working on a spam classifier, where spam

emails are positive examples (*y*=1) and non-spam emails are

negative examples (*y*=0). You have a training set of emails

in which 99% of the emails are non-spam and the other 1% is

spam. Which of the following statements are true? Check all

that apply.



If you always predict spam (output *y*=1),

your classifier will have a recall of 0% and precision

of 99%.



If you always predict non-spam (output

*y*=0), your classifier will have a recall of

0%.



If you always predict spam (output *y*=1),

your classifier will have a recall of 100% and precision

of 1%.



If you always predict non-spam (output

*y*=0), your classifier will have an accuracy of

99%.

5.

Which of the following statements are true? Check all that apply.



After training a logistic regression

classifier, you **must** use 0.5 as your threshold

for predicting whether an example is positive or

negative.



Using a **very large** training set

makes it unlikely for model to overfit the training

data.



If your model is underfitting the

training set, then obtaining more data is likely to

help.



It is a good idea to spend a lot of time

collecting a **large** amount of data before building

your first version of a learning algorithm.



The "error analysis" process of manually

examining the examples which your algorithm got wrong

can help suggest what are good steps to take (e.g.,

developing new features) to improve your algorithm's

performance.

Submit Quiz

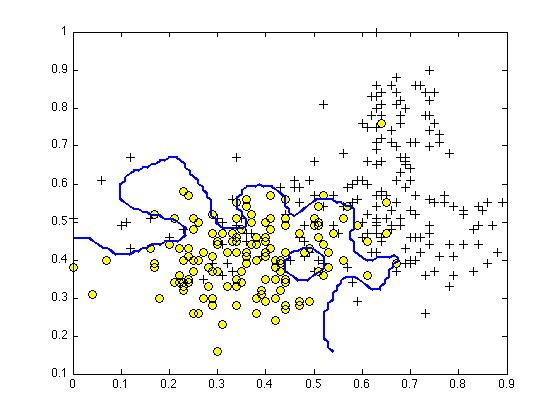
Close

# Support Vector Machines \* Boom! First time again

5 questions

1.

Suppose you have trained an SVM classifier with a Gaussian kernel, and it learned the following decision boundary on the training set:



When you measure the SVM's performance on a cross validation set, it does poorly. Should you try increasing or decreasing *C*? Increasing or decreasing *σ*2?



It would be reasonable to try **decreasing** *C*. It would also be reasonable to try**decreasing** *σ*2.



It would be reasonable to try **increasing** *C*. It would also be reasonable to try **decreasing***σ*2.



It would be reasonable to try **decreasing** *C*. It would also be reasonable to try **increasing***σ*2.

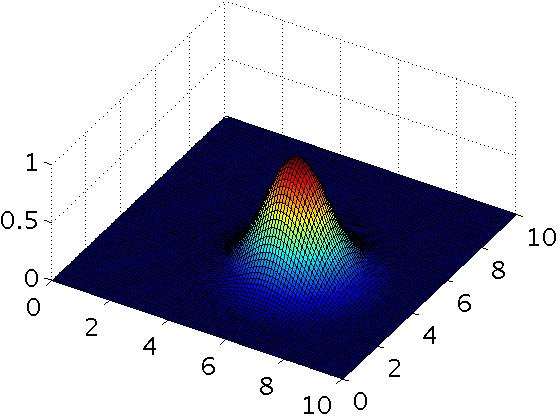


It would be reasonable to try **increasing** *C*. It would also be reasonable to try **increasing** *σ*2.

2.

The formula for the Gaussian kernel is given by similarity(*x*,*l*(1))=exp(−||*x*−*l*(1)||22*σ*2) .

The figure below shows a plot of *f*1=similarity(*x*,*l*(1)) when *σ*2=1.



Which of the following is a plot of *f*1 when *σ*2=0.25?



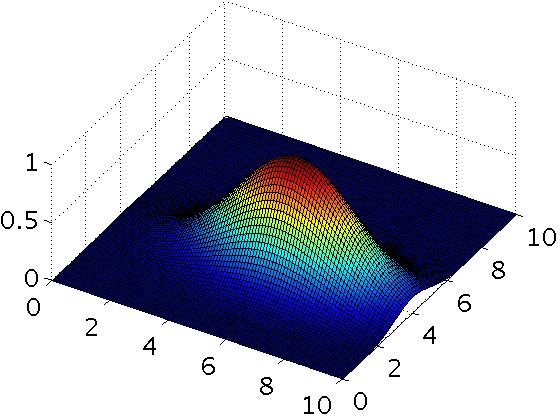
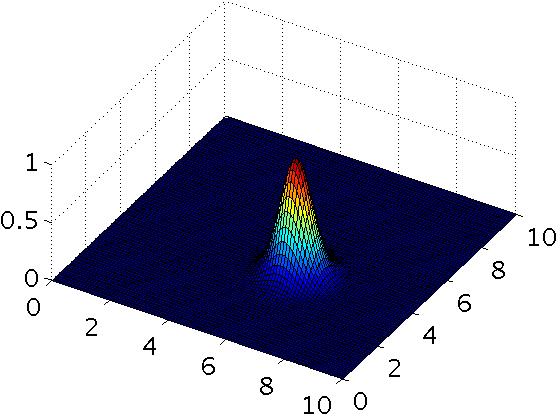


Figure 3.







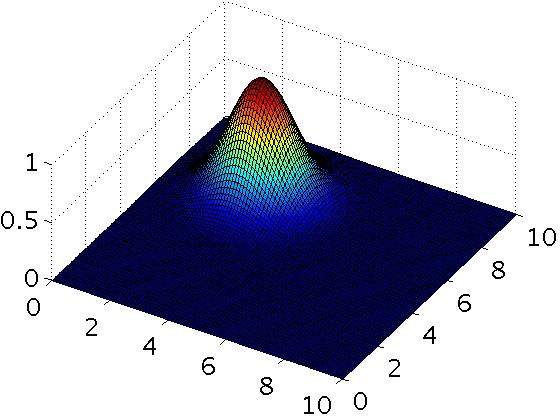


Figure 4.



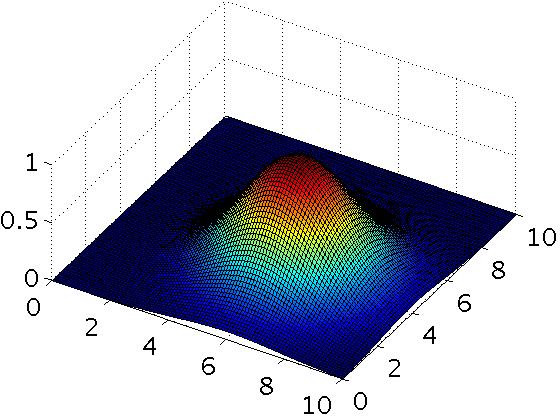


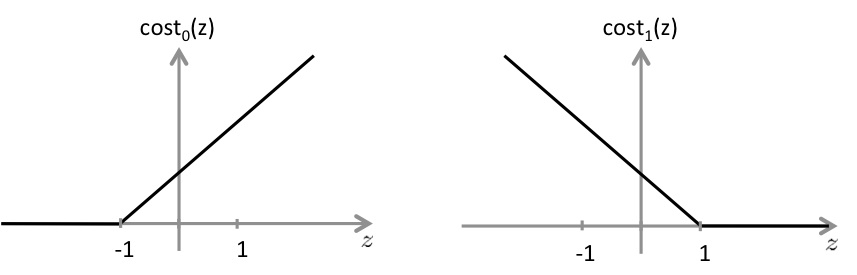
Figure 2.

3.

The SVM solves

min*θ* *C*∑*mi*=1*y*(*i*)cost1(*θTx*(*i*))+(1−*y*(*i*))cost0(*θTx*(*i*))+∑*nj*=1*θ*2*j*

where the functions cost0(*z*) and cost1(*z*) look like this:



The first term in the objective is:

*C*∑*mi*=1*y*(*i*)cost1(*θTx*(*i*))+(1−*y*(*i*))cost0(*θTx*(*i*)).

This first term will be zero if two of the following four conditions hold true. Which are the two conditions that would guarantee that this term equals zero?



For every example with *y*(*i*)=1, we have that *θTx*(*i*)≥1.



For every example with *y*(*i*)=0, we have that *θTx*(*i*)≤0.



For every example with *y*(*i*)=0, we have that *θTx*(*i*)≤−1.



For every example with *y*(*i*)=1, we have that *θTx*(*i*)≥0.

4.

Suppose you have a dataset with n = 10 features and m = 5000 examples.

After training your logistic regression classifier with gradient descent, you find that it has underfit the training set and does not achieve the desired performance on the training or cross validation sets.

Which of the following might be promising steps to take? Check all that apply.



Use a different optimization method since using gradient descent to train logistic regression might result in a local minimum.



Create / add new polynomial features.



Try using a neural network with a large number of hidden units.



Reduce the number of examples in the training set.

5.

Which of the following statements are true? Check all that apply.



The maximum value of the Gaussian kernel (i.e., *sim*(*x*,*l*(1))) is 1.



Suppose you have 2D input examples (ie, *x*(*i*)∈R2). The decision boundary of the SVM (with the linear kernel) is a straight line.



If the data are linearly separable, an SVM using a linear kernel will

return the same parameters *θ* regardless of the chosen value of

*C* (i.e., the resulting value of *θ* does not depend on *C*).



If you are training multi-class SVMs with the one-vs-all method, it is

not possible to use a kernel.

2 questions unanswered

Submit Quiz