1 Week 2

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 , where is the midterm score and is (midterm score) . 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 ? (Hint: midterm = 72, fial = 74 is training example 2.)

Please round your answer to two decimal places and enter in the text box below.

Suppose you have training examples with features (excluding the additional all-ones feature for the intercept term, which you should add). The normal equation is . For the given values of and , what are the dimensions of , , and in this equation?

1.1 Question 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?

1.2 Question 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)=θ0+θ1x1+θ2x2, where x1 is the midterm score and x2 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(1)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.

1.3 Question 1. Variant

What is the normalized feature $x_2^{(4)}$? (Hint: midterm = 69, final = 78 is training example 4.) Please round off your answer to two decimal places and enter in the text box below.

The mean of x2 is 6675.5 and the range is 8836-4761=4075 So x(1)1 is 4761-6675.54075=-0.47.

1.4 Question 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) **decreases** quickly then levels
off. Based on this, which of the following conclusions seems most plausible?

- (CORRECT) α=0.3 is an effective choice of learning rate. [YES]
- 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).

1.5 Question 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) **decreases slowly** then levels
off. Based on this, which of the following conclusions seems most plausible?

- α=0.3 is an effective choice of learning rate. [YES]
- Rather than use the current value of α, it'd be more promising to try a smaller value of α (say α=0.1).
- (CORRECT) Rather than use the current value of α, it'd be more promising to try a larger value of α (say α=1.0).

Question 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) decreases quickly then levels off. Based on this, which of the following conclusions seems most plausible?

- Rather than use the current value of α, it'd be more promising to try a larger value of α (say α=1.0).
  Incorrect 0.00 A larger value for α will make it more likely that J(θ) diverges.
- Rather than use the current value of α, it'd be more promising to try a smaller value of α (say α=0.1).
- α=0.3 is an effective choice of learning rate.

Total 0.00 / 1.00

Rather than use the current value of a, it'd be more promising to try a smaller value of α (say α=0.1). Correct 1.00 Since the cost function is increasing, we know that gradient descent is diverging, so we need a lower learning rate.

1.6 Question 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=$(X^TX)^{-1}X^Ty$. For the given values of m and n, what are the dimensions of θ, X, and y in this equation?

- X is 23×6, y is 23×1, θ is 6×1 [YES]
- X is 23×5, y is 23×1, θ is 5×5
- X is 23×5, y is 23×1, θ is 5×1
- X is 23×6, y is 23×6, θ is 6×6

1.7 Question 3. Variant

Suppose you have m=28 training examples with n=4 features (excluding the additional all-ones feature for the intercept term, which you should add). The normal equation is θ=$(X^TX)^{-1}X^Ty$. For the given values of m and n, what are the dimensions of θ, X, and y in this equation?

- X is 28×5, y is 28×1, θ is 5×1 [YES]
- X is 28×5, y is 28×1, θ is 5×5
- X is 28×5, y is 28×1, θ is 5×1
- X is 28×5, y is 28×5, θ is 5×5

Question 3

Suppose you have m=28 training examples with n=4 features (excluding the additional all-ones feature for the intercept term, which you should add). The normal equation is θ=(XTX)-1XTy. For the given values of m and n, what are the dimensions of θ, X, and y in this equation?

- X is 28×4, y is 28×1, θ is 4×4
- X is 28×4, y is 28×1, θ is4×1
- X is 28×5, y is 28×1, θ is 5×1 **Correct 1.0**
- X is 28×5, y is 28×5, θ is 5×5

Total 1.00 / 1.00 Question Explanation

X has m rows and n+1 columns (+1 because of the x0=1 term). y is an m-vector. θ is an (n+1)-vector.

1.8 Question 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. [NO]
- Gradient descent, since it will always converge to the optimal θ.
- Gradient descent, since $(X^TX)^{-1}$ will be very slow to compute in the normal equation.
- The normal equation, since gradient descent might be unable to find the optimal θ.

1.9 Question 4. variant

Suppose you have a dataset with m=50 examples and n=15 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. [NO]
- Gradient descent, since it will always converge to the optimal θ.
- Gradient descent, since $(X^TX)^{-1}$ will be very slow to compute in the normal equation.
- The normal equation, since gradient descent might be unable to find the optimal θ.

Question 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 $(X^TX)^{-1}$ will be very slow to compute in the normal equation.
  Correct
- 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.
  Incorrect 0.00
  For an appropriate choice of α, gradient descent can always find the optimal θ.

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

1.10 Question 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 $X^T X$ (used in the normal equation) from being non-invertable (singular/degenerate).

- (CORRECT) 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.

Question 5

Which of the following are reasons for using feature scaling?

- FALSE It prevents the matrix XTX (used in the normal equation) from being non-invertable (singular/degenerate).
  Correct 0.25 XTX can be singular when features are redundant or there are too few examples. Feature scaling does not solve these problems.

- FALSE It speeds up gradient descent by making it require fewer iterations to get to a good solution.
  Correct 0.25 Feature scaling speeds up gradient descent by avoiding many extra iterations that are required when one or more features take on much larger values than the rest.

- TRUE It speeds up solving for θ using the normal equation.
  Incorrect 0.00 The magnitude of the feature values are insignificant in terms of computational cost.

- FALSE It is necessary to prevent gradient descent from getting stuck in local optima.
  Correct 0.25 The cost function J(θ) for linear regression has no local optima.

2 Week 3 - Logistic Regression

2.1 Logistic Function

- A logistic function (or logistic curve) is a common sigmoid function, given its name (in reference to its S-shape) in 1844 or 1845 by Pierre François Verhulst who studied it in relation to population growth.

- A generalized logistic curve can model the "S-shaped" behaviour (abbreviated S-curve) of growth of some population P.

- The initial stage of growth is approximately exponential; then, as saturation begins, the growth slows, and at maturity, growth stops. The logistic function is the sigmoid curve with equation:

$$f(x) = \frac{1}{1 + \mathrm{e}^{-x}}$$

2.2 Question 1

Suppose that you have trained a logistic regression classifier, and it outputs on a new examplex a prediction $h\theta(x) = 0.7$. This means (check all that apply):

- Our estimate forPr$(y| = 1|x; \theta)$ is 0.7.

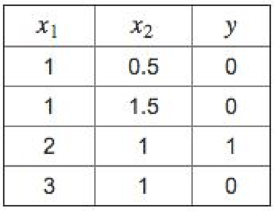
- Our estimate for$\Pr(y| = 0|x; \theta)$ is 0.3.
- Our estimate for$\Pr(y| = 1|x; \theta)$ is 0.3.
- Our estimate for$\Pr(y| = 0|x; \theta)$ is 0.7.

Solution

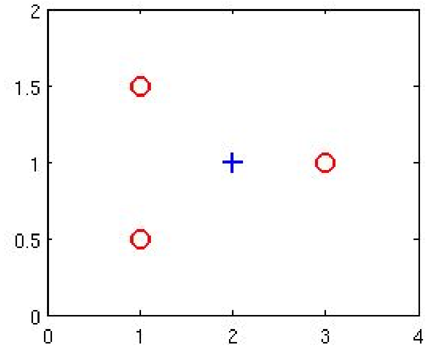
- Our estimate for $\Pr(y| = 1|x; \theta)$ is 0.7. T $h\theta(x)$ is precisely$\Pr(y| = 1|x; \theta)$, so each is 0.7.
- Our estimate for $\Pr(y| = 0|x; \theta)$ is 0.3. T Since we must have $\Pr(y| = 0|x; \theta) =$ 1-$\Pr(y| = 1|x; \theta)$, the former is 1-0.7=0.3 .
- Our estimate for $\Pr(y| = 1|x; \theta)$ is 0.3. F $h\theta(x)$ gives $\Pr(y| = 1|x; \theta)$, not 1-$\Pr(y| = 1|x; \theta)$.
- Our estimate for $\Pr(y| = 0|x; \theta)$ is 0.7. F $h\theta(x)$ is $\Pr(y| = 1|x; \theta)$, not $\Pr(y| = 0|x; \theta)$

Question 2.

Suppose you have the following training set, and fit a logistic regression classifier hθ(x) = g(θ_0 + $\theta 1 x_1$ + $\theta 2 x_2$).



x_1	x_2	y
1	0.5	0
1	1.5	0
2	1	1
3	1	0



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

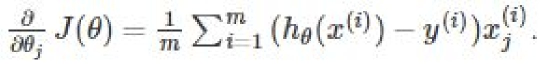
2.3 Question 3. variant

Suppose you train a logistic classifier hθ(x)=g(θ0+θ1x1+θ2x2). Suppose θ0=-6,θ1=0,θ2=1.

Which of the following figures represents the decision boundary found by your classifier?

Question 3.

For logistic regression, the gradient is given by EQUATION



$$\frac{\partial}{\partial\theta_j}J(\theta) = \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}.$$

Which of these is a correct gradient descent update for logistic regression with a learning rate of α? Check all that apply.

Question 4.

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

- WRONG J(θ) will be a convex function, so gradient descent should converge to the global minimum.
- CORRECT Adding polynomial features (e.g., instead using hθ(x)=g(θ0+θ1x1+θ2x2+θ3x21+θ4x1x2+
) could increase how well we can fit the training data.
- WRONG The positive and negative examples cannot be separated using a straight line. So, gradient descent will fail to converge.
- WRONG 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.
- Adding polynomial features (e.g., instead using h(x)=g(0+1x1+2x2+3x12+4x1x2+5x22)
) 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.
- At the optimal value of θ (e.g., found by fminunc), we will have $J(\theta)$ $\geq$0 .
- 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. solution

Discussion

1. Adding polynomial features (e.g., instead using h(x)=g(0+1x1+2x2+3x12+4x1x2+5x22)) could increase how well we can fit the training data.

TRUE Adding new features can only improve the fit on the training set: since setting $\theta_3 = \theta_4 = \theta_5 = 0$ makes the hypothesis the same as the original one, gradient descent will use those features (by making the corresponding θ_j non-zero) only if doing so improves the training set fit.

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

FALSE While it is true they cannot be separated, gradient descent will still converge to the optimal fit. Some examples will remain misclassified at the optimum.

3. At the optimal value of θ (e.g., found by fminunc), we will have $J(\theta) \geq 0$.

TRUE The cost function $J(\theta)$ is always non-negative for logistic regression.

4. 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.

FALSE While it is true they cannot be separated, logistic regression will outperform linear regression since its cost function focuses on classification, not prediction.

2.4 Question 5

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

- 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).
- Since we train one classifier when there are two classes, we train two classifiers when there are three classes (and we do one-vs-all classification).

- The sigmoid function g(z) is never greater than one (¿1). g(z)=11+e-z
- The one-vs-all technique allows you to use logistic regression for problems in which eachy(i) comes from a fixed, discrete set of values.

Solutions

1. 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). 0.00

   The cost function for logistic regression is convex, so gradient descent will always converge to the global minimum. We still might use a more advanded optimization algorithm since they can be faster and don't require you to select a learning rate.

2. Since we train one classifier when there are two classes, we train two classifiers when there are three classes (and we do one-vs-all classification).

   We need to train three classifiers if there are three classes; each one treats one of the three classes as the y=1 examples and the rest as the y=0 examples.

3. The sigmoid function g(z) is never greater than one (¿1).

   The denominator ranges from to1 asz grows, so the result is always in(0,1) .

4. The one-vs-all technique allows you to use logistic regression for problems in which each y(i) comes from a fixed, discrete set of values. If each y(i) is one of k different values, we can give a label to each $y(i) \in \{1, 2, \ldots, k\}$ and use one-vs-all as described in the lecture.

3 Week 3 - Regularization

3.1 Question 1.

You are training a classification model with logistic
regression. Which of the following statements are true? Check all that apply.

- Adding a new feature to the model always results in equal or better performance on examples not in the training set.
- Adding many new features to the model makes it more likely to overfit the training set.
- Introducing regularization to the model always results in equal or better performance on examples not in the training set.
- Introducing regularization to the model always results in equal or better performance on the training set.

3.2 Question 2.

Suppose you ran logistic regression twice, once with λ=0, and once with λ=1. One of the times, you got
parameters θ=[26.2965.41], and the other time you got
θ=[2.751.32]. However, you forgot which value of
λ corresponds to which value of θ. Which one do you
think corresponds to λ=1?
θ=[26.2965.41]
θ=[2.751.32] YES

3.3 Question 3.

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

- Because logistic regression outputs values $0 leq$hθ(x)$leq 1$, it's range of output values can only be "shrunk" slightly by regularization anyway, so regularization is generally not helpful for it.
- YES Using too large a value of λ can cause your hypothesis to overfit the data; this can be avoided by reducing λ.
- NO 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).
- 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.

3.4 Question 4.

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

Figure:
Figure:
Figure:
Figure:

3.5 Question 5.

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

Figure:
Figure:
Figure:
Figure:

```
4
0.5
3

NO No matter how $\theta_0$ and $\theta_1$ are initialized, so long as $\alpha$ is suffic
can safely expect gradient descent to converge to the same solution.
Correct 0.25
This is not true, because depending on the initial condition, gradient descent may end up

YES If the learning rate is too small, then gradient descent may take a very long time to
Correct 0.25
If the learning rate is small, gradient descent ends up taking an extremely small step on

YES If $\theta_0$ and $\theta_1$ are initialized at the global minimum, the one iteration
Correct 0.25
At the global minimum, the derivative (gradient) is zero, so gradient descent will not ch

NO Setting the learning rate $\alpha$ to be very small is not harmful, and can only speed
Correct 0.25
If the learning rate is small, gradient descent ends up taking an extremely small step on

If \theta0 and \theta1 are initialized at a local minimum, the one iteration will not
change their values. Inorrect 0.00
At a local minimum, the derivative (gradient) is zero, so gradient descent
will not change the parameters.

YES If the first few iterations of gradient descent cause f(\theta0,\theta1) to increase
Inorrect 0.00 If alpha were small enough, then gradient descent should always successfull

YES If \theta0 and \theta1 are initialized at the global minimum, the one iteration will
Correct 0.25 At the global minimum, the derivative (gradient) is zero, so gradient descen

YES No matter how \theta0 and \theta1 are initialized, so long as \alpha  is sufficiently
Correct 0.25 This is not true, because depending on the initial condition, gradient desce

NO Even if the learning rate \alpha  is very large, every iteration of gradient descent w

Inorrect 0.00 If the learning rate \alpha  is too large, one step of gradient descent can

YES If \theta0 and \theta1 are initialized so that \theta0=\theta1, then by symmetry (bec
Inorrect 0.00 The updates to \theta0 and \theta1 are different (even though we're doing s

Suppose that for some linear regression problem (say, predicting housing prices as in the
managed to find some $\theta_0$, $\theta_1$ such that $J(\theta_0, \theta_1)$=0. Which of
```

```
Your Answer Score Explanation

NO We can perfectly predict the value of y even for new examples that we have not yet see
Inorrect 0.00 Even though we can fit our training set perfectly, this does not mean that

NO This is not possible: By the definition of J(\theta0,\theta1), it is not possible for
Correct 0.25 If all of our training examples lie perfectly on a line, then J(\theta0,\the

YES Our training set can be fit perfectly by a straight line, i.e., all of our training e
Inorrect 0.00
If J(\theta0,\theta1)=0, that means the line defined by the equation "y=\theta0+\theta1x"

NO Gradient descent is likely to get stuck at a local minimum and fail to find the global
Inorrect 0.00 The cost function J(\theta0,\theta1) for linear regression has no local opt

NO For this to be true, we must have y(i)=0 for every value of i=1,2,...,m.
Correct 0.25 So long as all of our training examples lie on a straight line, we will be a
NO For this to be true, we must have $\theta_0$=0 and $\theta_1$=0 so that $h_\theta$(x)=
Correct 0.25

If $J(\theta_0, \theta_1)$=0, that means the line defined by the equation "y=$\theta_0$+$
There’s no particular reason to expect that the values of $\theta_0$ and $\theta_1$ that

NO This is not possible: By the definition of $J(\theta_0, \theta_1)$, it is not possible
Correct 0.25 If all of our training examples lie perfectly on a line, then $J(\theta_0, \

YES Gradient descent is likely to get stuck at a local minimum and fail to find the globa
Incorrect 0.00 The cost function $J(\theta_0, \theta_1)$ for linear regression has no loc
so gradient descent will not get stuck at a bad local minimum.

NO Our training set can be fit perfectly by a straight line, i.e., all of our training ex
Incorrect 0.00
If $J(\theta_0, \theta_1)$=0, that means the line defined by the equation "y=$\theta_0$+$
of our data.

YES For these values of \theta0 and \theta1 that satisfy J(\theta0,\theta1)=0, we have th
Inorrect 0.00 J(\theta0,\theta1)=0, that means the line defined by the equation "y=\theta
```

4 Week 4 - Neural Networks : Representation

Topics: Neural Networks: Representation

Programming Exercise 3 : Multi-class classification and neural networks. Close Neural Networks: Representation

Close Neural Networks: Representation

Question 1.

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

SELECTED If a neural network is overfitting the data, one solution would be to increase 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.

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

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

Question 2.

Consider the following neural network which takes two binary-valued inputs x1,x2 $\in \{0,1\}$ and outputs hΘ(x). Which of the following logical functions does it (approximately) compute?

- OR
- AND
- NAND (meaning "NOT AND")
- XOR (exclusive OR)

Question 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)1,0a(2)0+Θ(2)1,1a(2)1+Θ(2)1,2a(2)2)

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

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

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

Question 4.

You have the following neural network:

You'd like to compute the activations of the hidden layer a(2)$\in R^3$. 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);

Question 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 hΘ(x)?

- It will stay the same.
- It will increase.
- It will decrease
- Insufficient information to tell: it may increase or decrease.