Suppose you are running gradient descent to fit a logistic regression model with parameter. Which of the following is a reasonable way to make sure the learning rate α is set properly and that gradient descent is running correctly?

1. Plot $J(\theta) = \frac{1}{m}\sum_{i=1}^{m}[-y^{(i)}log(h_\theta(x^{(i)})) - (1-y^{(i)})log(1-h_\theta(x^{(i)}))]$ as a function of the number of iterations and make sure $J(\theta)$ is decreasing on every iteration: **True**
2. Plot $J(\theta)$ as a function of \thetaθ and make sure it is decreasing on every iteration: **False**
3. Plot $J(\theta)$ as a function of \thetaθ and make sure it is convex: **False**

One iteration of gradient descent simultaneously performs these updates:

$\theta_0 := \theta_0 - \frac{\alpha}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_0^{(i)}$

$\theta_1 := \theta_1 - \frac{\alpha}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_1^{(i)}$

..

$\theta_n := \theta_n - \frac{\alpha}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_n^{(i)}$

What should the vectorized implementation be?

1. $\theta := \theta - \frac{\alpha}{m}\sum_{i=1}^{m}[(h(x^{(i)}) - y^{(i)})x^{(i)}]$: **True**
2. $\theta := \theta - \frac{\alpha}{m}\sum_{i=1}^{m}[(h(x^{(i)}) - y^{(i)})]x^{(i)}$: **False**

Suppose you want to use an advanced optimization algorithm to minimize the cost function for logistic regression with parameters θ_0 and θ_1. You write the following code:

```
function [jVal, gradient] = costFn
    jVal = ...
    gradient(1) = ... CODE#1
    gradient(2) = ... CODE#2
```

What should CODE#1 and CODE#2 above compute?

$\frac{d}{d\theta_0}J(\theta) = \frac{1}{m}\sum_{i=1}^{m}[(h(x^{(i)}) - y^{(i)})x_0^{(i)}]$

$\frac{d}{d\theta_1}J(\theta) = \frac{1}{m}\sum_{i=1}^{m}[(h(x^{(i)}) - y^{(i)})x_1^{(i)}]$

Suppose you have a multi-class classification problem with k classes (so $y \in \{1, 2, \ldots, k\}$). Using the 1-vs.-all method, how many different logistic regression classifiers will you end up training? k

---

QUIZ

Logistic Regression

Suppose that you have trained a logistic regression classifier, and it outputs on a new example x a prediction $h_\theta(x) = 0.4$. This means (check all that apply):

1. Our estimate for $P(y = 0|x; \theta)$ is 0.6.
2. Our estimate for $P(y = 1|x; \theta)$ is 0.4.
3. Our estimate for $P(y = 0|x; \theta)$ is 0.4.
4. Our estimate for $P(y = 1|x; \theta)$ is 0.6.

2 and 4 because $h_\theta(x) = P(y = 1|x; \theta) = 1 - P(y = 0|x; \theta)$

Suppose you have the following training set, and fit a logistic regression classifier $h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$:

$x1$	$x2$	y
1	0.5	0
1	0.5	0
2	1	1
3	1	0

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

1. $J(\theta)$ will be a convex function, so gradient descent should converge to the global minimum.
2. Adding polynomial features (e.g., instead using $h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_1 x_2 + \theta_5 x_2^2)$ could increase how well we can fit the training data.
3. The positive and negative examples cannot be separated using a straight line. So, gradient descent will fail to converge.
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.

1 and 2

Setting $\theta_3 = \theta_3 4 = \theta_3 5 = 0$ makes the hypothesis the same as the original one, gradient descent will use those features only if doing so improves the training set fit.

For logistic regression, the gradient is given by $\frac{\partial}{\partial \theta_j} J(\theta) = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$. Which of these is a correct gradient descent update for logistic regression with a learning rate of \alphaα? Check all that apply.

1. θ:=θ−αm1∑i=1m(θTx−y(i))x(i).
2. θj:=θj−αm1∑i=1m(1+e−θTx(i)1−y(i))xj(i) (simultaneously update for all j).
3. θj:=θj−αm1∑i=1m(hθ(x(i))−y(i))x(i) (simultaneously update for all j).
4. θj:=θj−αm1∑i=1m(hθ(x(i))−y(i))xj(i) (simultaneously update for all j).

2 and 4

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

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).
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).
3. The cost function J(\theta)J(θ) for logistic regression trained with m \geq 1m≥1 examples is always greater than or equal to zero.
4. The one-vs-all technique allows you to use logistic regression for problems in which each y^{(i)}y(i) comes from a fixed, discrete set of values.

2 and 3

Suppose you train a logistic classifier h_\theta(x) = g(\theta_0 + \theta_1x_1 + \theta_2 x_2)hθ(x)=g(θ0+θ1x1+θ2x2). Suppose \theta_0 = - 6, \theta_1 = 1, \theta_2 = 0θ0=−6,θ1=1,θ2=0. Which of the following figures represents the decision boundary found by your classifier?

x1 >= 6 when y= 1

Consider the medical diagnosis problem of classifying tumors as malignant or benign. If a hypothesis h_\theta(x)hθ(x) has overfit the training set, it means that:

It makes accurate predictions for examples in the training set, but it does not generalize well to make accurate predictions on new, previously unseen examples.

In regularized linear regression, we choose $\boldsymbol{\theta}$ to minimize:

$$J(\theta) = 1/2m[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1} n\theta_j^2]$$

What if $\boldsymbol{\lambda}$ is set to an extremely large value (perhaps too large for our problem, say $\boldsymbol{\lambda = 10^{10}}$)?

Algorithm results in underfitting (fails to fit even the training set).