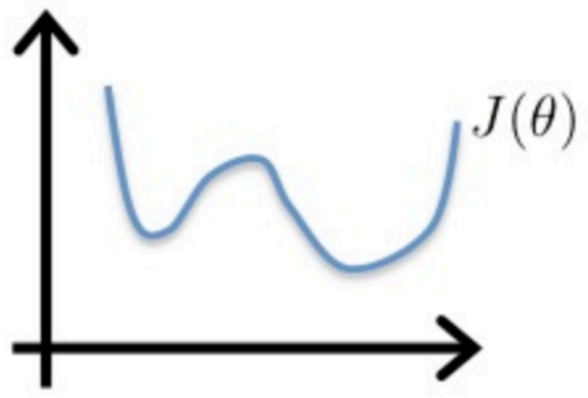
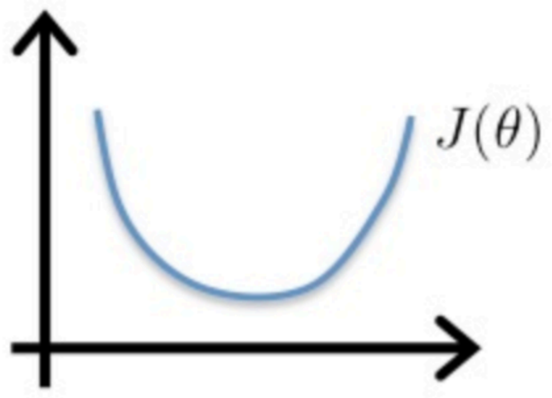
Consider minimizing a cost function $J(\theta)$. Which one of these functions is convex?

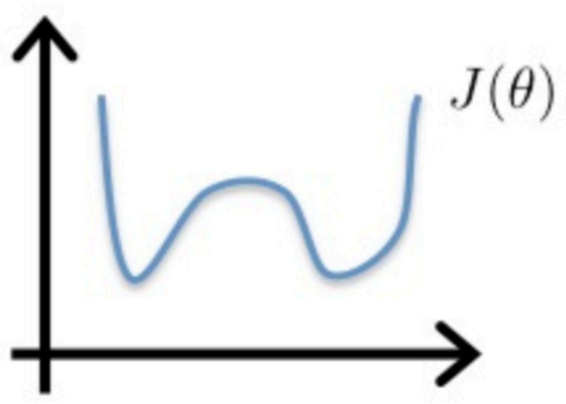


.

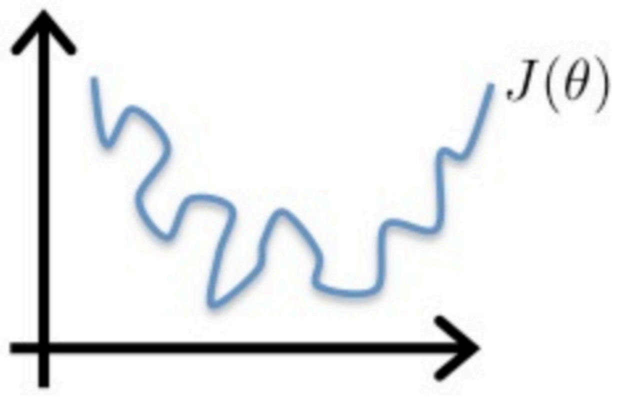


.

Correct



.



Continue

In logistic regression, the cost function for our hypothesis outputting (predicting) $h_\theta(x)$ on a training example that has label $y \in \{0, 1\}$ is:

$$\mathrm{cost}(h_\theta(x), y) = \begin{cases} -\log h_\theta(x) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

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

- [x] If $h_\theta(x) = y$, then $\mathrm{cost}(h_\theta(x), y) = 0$ (for $y = 0$ and $y = 1$).

Correct

- [x] If $y = 0$, then $\mathrm{cost}(h_\theta(x), y) \to \infty$ as $h_\theta(x) \to 1$.

Correct

- [ ] If $y = 0$, then $\mathrm{cost}(h_\theta(x), y) \to \infty$ as $h_\theta(x) \to 0$.

Un-selected is correct

- [x] Regardless of whether $y = 0$ or $y = 1$, if $h_\theta(x) = 0.5$, then $\mathrm{cost}(h_\theta(x), y) > 0$.

Correct

Suppose you are running gradient descent to fit a logistic regression model with parameter $\theta \in \mathbb{R}^{n+1}$. 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?

- ○ Plot $J(\theta) = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$ as a function of the number of iterations (i.e. the horizontal axis is the iteration number) and make sure $J(\theta)$ is decreasing on every iteration.
- ◉ 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.
- ○ Plot $J(\theta)$ as a function of θ and make sure it is decreasing on every iteration.
- ○ Plot $J(\theta)$ as a function of θ and make sure it is convex.

This should not be selected

Suppose you are running gradient descent to fit a logistic regression model with parameter $\theta \in \mathbb{R}^{n+1}$. 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?

- Plot $J(\theta) = \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$ as a function of the number of iterations (i.e. the horizontal axis is the iteration number) and make sure $J(\theta)$ is decreasing on every iteration.
- 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.

  Correct

- Plot $J(\theta)$ as a function of θ and make sure it is decreasing on every iteration.
- Plot $J(\theta)$ as a function of θ and make sure it is convex.

One iteration of gradient descent simultaneously performs these updates:

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

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

$$\vdots$$

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

We would like a vectorized implementation of the form $\theta := \theta - \alpha\delta$ (for some vector $\delta \in \mathbb{R}^{n+1}$).

What should the vectorized implementation be?

- (selected) $\theta := \theta - \alpha \frac{1}{m} \sum_{i=1}^{m} [(h_\theta(x^{(i)}) - y^{(i)}) \cdot x^{(i)}]$

Correct

- $\theta := \theta - \alpha \frac{1}{m} [\sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})] \cdot x^{(i)}$
- $\theta := \theta - \alpha \frac{1}{m} x^{(i)} [\sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})]$
- All of the above are correct implementations.

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] = costFunction(theta)
  jVal = % code to compute J(theta)
  gradient(1) = CODE#1 % derivative for theta_0
  gradient(2) = CODE#2 % derivative for theta_1
```

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

- CODE#1 and CODE#2 should compute $J(\theta)$.
- CODE#1 should be theta(1) and CODE#2 should be theta(2).
- CODE#1 should compute $\frac{1}{m}\sum_{i=1}^{m}[(h_\theta(x^{(i)}) - y^{(i)}) \cdot x_0^{(i)}](= \frac{\partial}{\partial\theta_0}J(\theta))$, and

  CODE#2 should compute $\frac{1}{m}\sum_{i=1}^{m}[(h_\theta(x^{(i)}) - y^{(i)}) \cdot x_1^{(i)}](= \frac{\partial}{\partial\theta_1}J(\theta))$.

  Correct
- None of the above.