Logistic Regression notes

Robert K.

December 26, 2016

Abstract

This is just a document with notes regarding Logistic Regression.

1 Hypothesis function for logistic regression

Hypothesis function in Logistic Regression has a form:

$$h_\Theta(x^{(i)}) = g(\Theta^T x^{(i)}) \tag{1}$$

where $g()$ is a "Sigmoid Function" (a.k.a "Logistic Function")

$$g(z) = \frac{1}{1+e^{-z}} \tag{2}$$

where:
z - is scalar value $\Theta^T x^{(i)}$
Θ - is the column vector of Θ_0, Θ_1 ... Θ_n values
$x^{(i)}$ - is the column vector of $x_0^{(i)} = 1$, $x_1^{(i)}$ ... $x_n^{(i)}$ values in particular training set (i)

properties of $h_\Theta(x)$ - sigmoid function as the logistic regression hypothesis function:
a) $h_\Theta(x^{(i)})$ returns the scalar value
b) $0 < h_\Theta(x^{(i)}) < 1$
c) if $\Theta^T x^{(i)} = 0$ then $h_\Theta(x^{(i)}) = 0.5$
d) if $\Theta^T x^{(i)} \to \infty$ then $h_\Theta(x^{(i)}) = 1$
e) if $\Theta^T x^{(i)} \to -\infty$ then $h_\Theta(x^{(i)}) = 0$
f) $h_\Theta(x^{(i)})$ gives as the **probability** that our output is 1

Example: $h_\Theta(x^{(i)}) = 0.9$ gives as the 90% probability that our output is 1 or 10% that the output is 0 for the given $x^{(i)}$ and Θ column vectors.

$$h_\Theta(x^{(i)}) = P(y^{(i)} = 1 | x^{(i)}; \Theta) \tag{3}$$

$$h_\Theta(x^{(i)}) = 1 - P(y^{(i)} = 0|x^{(i)}; \Theta) \tag{4}$$
$$P(y = 1|x^{(i)}; \Theta) + P(y^{(i)} = 0|x^{(i)}; \Theta) = 1 \tag{5}$$

Excercise: Let's say that we've trained a logistic regression classifier. It means we have the Θ values. Now, for the given new x input we calculated $h_\Theta(x) = 0.3$. What does it mean?

*Answer:*It means that for given x and trained logistic regression classifier identified by Θ, estimates that the answer is positive with the probability of 30% and negative with the probability 70%.

$$P(y = 1|x; \Theta) = h_\Theta(x) = 0.3$$
$$P(y = 0|x; \Theta) = 1 - h_\Theta(x) = 1 - 0.3 = 0.7$$

2 Decision Boundary

With logistic regression classifier we should get discrete answers ($y^{(i)} = 0|1$) for the specific $x^{(i)}$ input vector. It means that we have to decide which value of $h_\Theta(x^{(i)})$ classifies as positive answer and which value classifies as the negative answer.

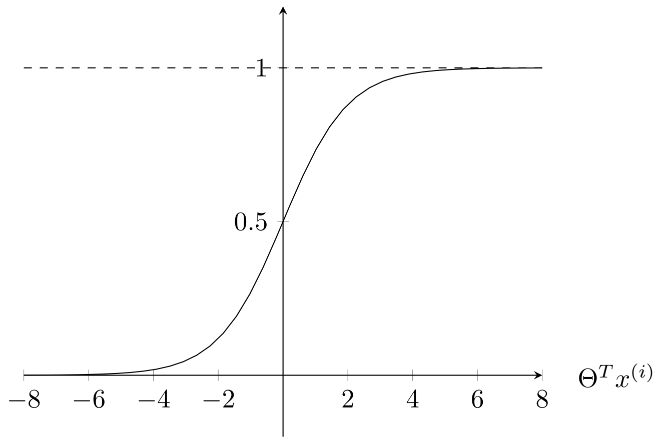
Using Sigmoid Function as the $h_\Theta(x^{(i)})$, it seems to be straight forward - since the $h_\Theta(x^{(i)})$ is the **probability** that our output is positive (a.k.a "true", 1, "yes", etc).

If $h_\Theta(x^{(i)})$ is greater or equal 0.5 then the answer is *positive*, and
If $h_\Theta(x^{(i)})$ is less 0.5 then the answer is *negative*

$$h_\Theta(x^{(i)}) \geq 0.5 \rightarrow y^{(i)} = 1 \tag{6}$$
$$h_\Theta(x^{(i)}) < 0.5 \rightarrow y^{(i)} = 0 \tag{7}$$

Let's draw $h_\Theta(x^{(i)})$ function:

$$h_\Theta(x^{(i)}) = \frac{1}{1+e^{-\Theta^T x^{(i)}}}$$



the following equations are true

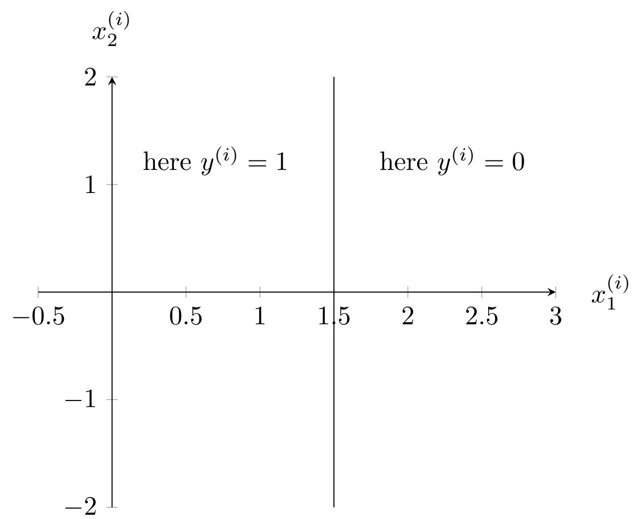
$$\Theta^T x^{(i)} \geq 0 \rightarrow y^{(i)} = 1 \tag{8}$$

$$\Theta^T x^{(i)} < 0 \rightarrow y^{(i)} = 0 \tag{9}$$

Conclusion: We do not have to calculate $h_\Theta(x^{(i)})$ to figure out if the result of our logistic regression classifier will be positive or negative. We just need to calculate $\Theta^T x^{(i)}$, if it is less the 0 then the result is negative, otherwise it's positive.

Example: Let's say that we have classifier identified by $\Theta = \begin{bmatrix} 3 \\ -2 \\ 0 \end{bmatrix}$.

The classifier will return positive answer ($y = 1$) if $\Theta^T x^{(i)} = 3x_0^{(i)} - 2x_1^{(i)} + 0x_2^{(i)} \geq 0$ (where $x_0^{(i)} = 1$). So it give us $x_1^{(i)} \leq 1.5$. So, our decision boundary is a straight vertical line placed on the graph where $x_1^{(i)} = 1.5$, and everything to the left of that denotes positive result ($y^{(i)} = 1$), while everything to the right denotes negative result ($y^{(i)} = 0$).



3 Cost function

3.1 Linear regression cost function cannot be used

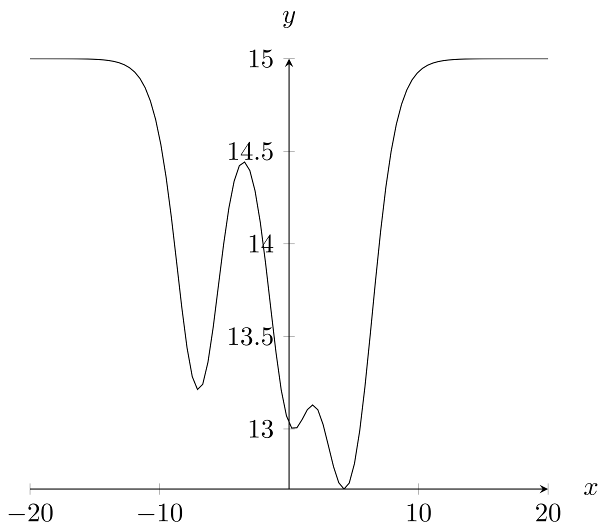
We cannot use the same cost function that we use for linear regression

$$J(\Theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

because the Logistic Function $h_\Theta(x^{(i)}) = \frac{1}{1+e^{-\Theta^T x^{(i)}}}$ will cause the output $J(\Theta)$ to be wavy.

$$J(\Theta) = \frac{1}{2m} \sum_{i=1}^{m} (\frac{1}{1 + e^{-\Theta^T x^{(i)}}} - y^{(i)})^2 \tag{10}$$

See the example figure below, where we have two local minimas next to global minimum. Formal term for this kind of function is **non-convex** function. Which is useless for gradient descend algorithm.



That's because the $h_\Theta(x^{(i)}) = \frac{1}{1+e^{-\Theta^T x^{(i)}}}$ is nonlinear.

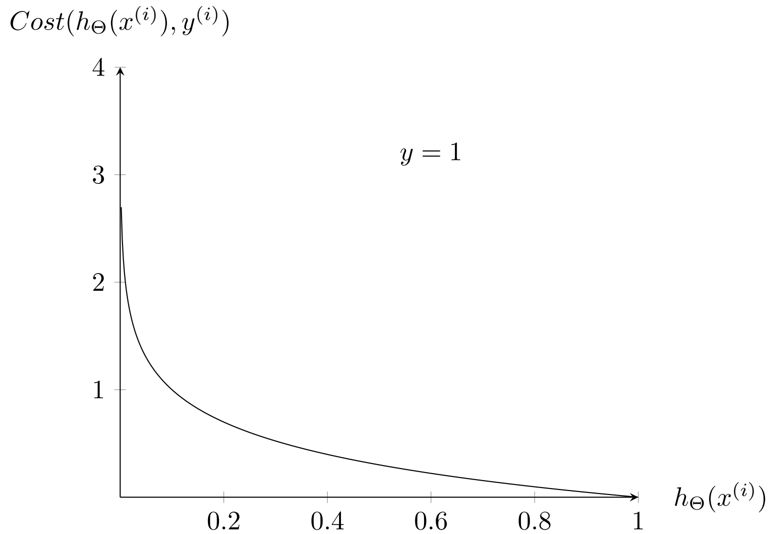
3.2 Logistic regression cost function

We are looking for function which is **convex** function - has one minimum. Let it be arithmetic mean of costs for particular training set (i=1, 2 .. m).

$$J(\Theta) = \frac{1}{m} \sum_{i=1}^{m} Cost(h_\Theta(x^{(i)}), y^{(i)}) \tag{11}$$

where single cost for particular training set (i) is:

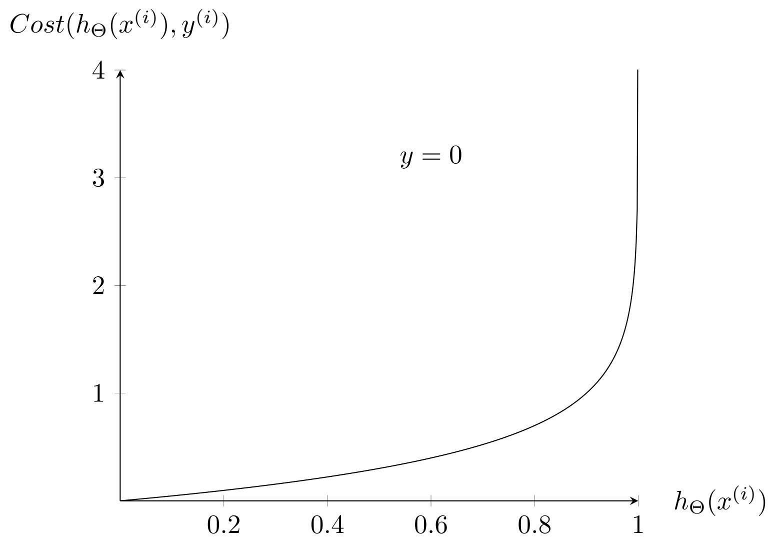
$$Cost(h_\Theta(x^{(i)}), y^{(i)}) = \begin{cases} -log(h_\Theta(x^{(i)})) & \text{if } y^{(i)} = 1 \\ -log(1 - h_\Theta(x^{(i)})) & \text{if } y^{(i)} = 0 \end{cases} \tag{12}$$



Note that $h_\Theta(x^{(i)})$ is the probability that y=1 for the given $x^{(i)}$ and Θ, other words:

$$h_\Theta(x^{(i)}) = P(y = 1|x^{(i)}; \Theta) \tag{13}$$

if probability that y=1 for the given $x^{(i)}$,Θ is 1, then $Cost(h_\Theta(x^{(i)}), y^{(i)})$ is 0
if probability that y=1 for the given $x^{(i)}$,Θ is 0, then $Cost(h_\Theta(x^{(i)}), y^{(i)})$ is ∞



Note that $1 - h_\Theta(x^{(i)})$ is the probability that y=0 for the given $x^{(i)}$ and Θ, other words:

$$1 - h_\Theta(x^{(i)}) = P(y = 0|x^{(i)}; \Theta) \tag{14}$$

if $h_\Theta(x^{(i)}) = 1$ then there is no chance that y=0 - $Cost(h_\Theta(x^{(i)}), y^{(i)})$ is ∞
if $h_\Theta(x^{(i)}) = 0$ then there is 100% chance that y=0 - $Cost(h_\Theta(x^{(i)}), y^{(i)})$ is 0

We can simplify the equation. So **cost function for logistic regression** can be written with equation:

$$J(\Theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}log(h_\Theta(x^{(i)})) + (1-y^{(i)})log(1-h_\Theta(x^{(i)}))] \tag{15}$$

where:

$y \in \{1, 2\}$

$h_\Theta(x^{(i)}) = \frac{1}{1+e^{-\Theta^T x^{(i)}}}$

Derivative of the cost function for logistic regression looks like that (see: Appendix B):

$$\frac{\delta}{\delta\Theta_j}J(\Theta) = \frac{1}{m}\sum_{i=1}^{m}(h_\Theta(x^{(i)}) - y^{(i)})x_j^{(i)} \tag{16}$$

where:

$y \in \{1, 2\}$

$h_\Theta(x^{(i)}) = \frac{1}{1+e^{-\Theta^T x^{(i)}}}$

3.3 Vectorised form of cost function

Let's assume that we have m training elements.
Single training element is represented by column vector $x^{(i)}$.
Each training element $x^{(i)} = \begin{bmatrix} x_0^{(i)} \\ x_1^{(i)} \\ \vdots \\ x_n^{(i)} \end{bmatrix}$ has $n+1$ features.

Let's insert all training input values (a.k.a. independent) into matrix X and dependant values ($y^{(i)}$) to column vector y.

$$X = \begin{bmatrix} (x^{(1)})^T \\ (x^{(2)})^T \\ \vdots \\ (x^{(m)})^T \end{bmatrix} = \begin{bmatrix} x_0^{(1)} & x_1^{(1)} & \dots & x_n^{(1)} \\ x_0^{(2)} & x_1^{(2)} & \dots & x_n^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_0^{(m)} & x_1^{(m)} & \dots & x_n^{(m)} \end{bmatrix}; y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix}$$

since $x_0^{(i)} = 1$ for every i

$$X = \begin{bmatrix} (x^{(1)})^T \\ (x^{(2)})^T \\ \vdots \\ (x^{(m)})^T \end{bmatrix} = \begin{bmatrix} 1 & x_1^{(1)} & x_2^{(1)} & \dots & x_n^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} & \dots & x_n^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(m)} & x_2^{(m)} & \dots & x_n^{(m)} \end{bmatrix}$$

$$X\Theta = \begin{bmatrix} 1 & x_1^{(1)} & x_2^{(1)} & \dots & x_n^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} & \dots & x_n^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(m)} & x_2^{(m)} & \dots & x_n^{(m)} \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix} = \begin{bmatrix} \theta_0 + x_1^{(1)} * \theta_1 + x_2^{(1)} * \theta_2 + \dots + x_n^{(1)} * \theta_n \\ \theta_0 + x_1^{(2)} * \theta_1 + x_2^{(2)} * \theta_2 + \dots + x_n^{(2)} * \theta_n \\ \vdots \\ \theta_0 + x_1^{(m)} * \theta_1 + x_2^{(m)} * \theta_2 + \dots + x_n^{(m)} * \theta_n \end{bmatrix}$$

in result we have

$$X\Theta = \begin{bmatrix} \Theta^T x^{(1)} \\ \Theta^T x^{(2)} \\ \vdots \\ \Theta^T x^{(m)} \end{bmatrix}$$

so, if we want to vectorise following equation:

$$J(\Theta) = -\frac{1}{m} \left[\sum_{i=1}^{m} y^{(i)} log(h_\Theta(x^{(i)})) + \sum_{i=1}^{m} (1 - y^{(i)}) log(1 - h_\Theta(x^{(i)})) \right] \tag{17}$$

where:

$$h_\Theta(x^{(i)}) = g(\Theta^T x^{(i)})$$
$$g(z) = \frac{1}{1 + e^{-z}}$$

so **final vectorised form of cost function for logistic regression**:

$$J(\Theta) = -\frac{1}{m} \cdot \left[y^T log(g(X\Theta)) + (1 - y)^T log(1 - g(X\Theta)) \right] \tag{18}$$

where:

$$g(X\Theta) = \begin{bmatrix} g(\Theta^T x^{(1)}) \\ g(\Theta^T x^{(2)}) \\ \vdots \\ g(\Theta^T x^{(m)}) \end{bmatrix}$$

$log(g(X\Theta))$ - is a column vector of size m. *Log* function is done on every element of $g(X\Theta)$ column vector.

3.4 Vectorised derivative of the cost function

As we know:

$$X\Theta = \begin{bmatrix} \Theta^T x^{(1)} \\ \Theta^T x^{(2)} \\ \vdots \\ \Theta^T x^{(m)} \end{bmatrix}$$

so, if we want to vectorise following equation:

$$\frac{\delta}{\delta\Theta_j} J(\Theta) = \frac{1}{m} \sum_{i=1}^{m} x_j^{(i)} (h_\Theta(x^{(i)}) - y^{(i)}) \tag{19}$$

where:

$$h_\Theta(x^{(i)}) = g(\Theta^T x^{(i)})$$

$$g(z) = \frac{1}{1+e^{-z}}$$

so **final vectorised form of cost function derivative for logistic regression**:

$$\nabla J(\Theta) = \begin{bmatrix} \frac{\delta}{\delta\Theta_0} J(\Theta) \\ \frac{\delta}{\delta\Theta_1} J(\Theta) \\ \vdots \\ \frac{\delta}{\delta\Theta_n} J(\Theta) \end{bmatrix} = \frac{1}{m} \cdot \left[X^T (g(X\Theta) - y) \right] \tag{20}$$

4 Gradient descent algorithm

Simplified gradient descent algorithm

repeat until convergence {

$\Theta_j := \Theta_j - \alpha \frac{\delta}{\delta\Theta_j} J(\Theta)$

}

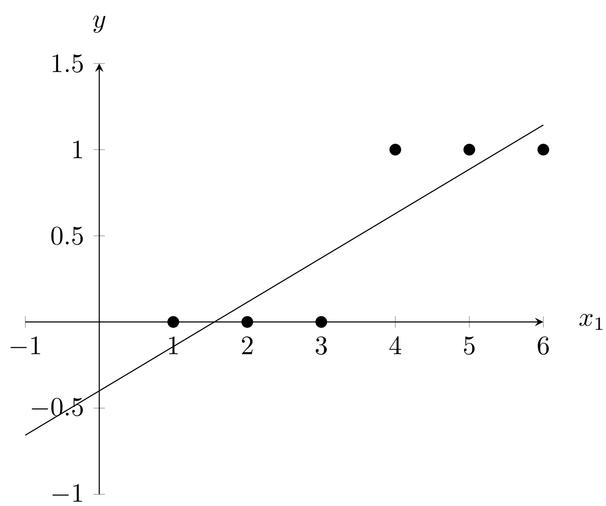
$$\frac{\delta}{\delta\Theta_j} J(\Theta) = \frac{1}{m} \sum_{i=1}^{m} (h_\Theta(x^{(i)}) - y^{(i)}) x_j^{(i)} \tag{21}$$

the equation above is exactly the same as for linear regression gradient descent

5 Appendix

A Normal Equations

Using Normal Equations for logistic regression is not a good idea. *Example* For training set $x = \begin{bmatrix} 1,1 \\ 1,2 \\ 1,3 \\ 1,4 \\ 1,5 \\ 1,6 \end{bmatrix}$; $y = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$ according to Normal Equations optimal Θ is $\Theta_{optimal} = (x^T x)^{-1} x^T y = \begin{bmatrix} -0.4 \\ 0.25714 \end{bmatrix}$, in the graph it looks like that:



B Derivatives

Derivative of sigmoid function (it will be used while finding partial derivative of $J(\Theta)$):

$$\sigma(x)' = \left(\frac{1}{1+e^{-x}}\right)' = \frac{-(1+e^{-x})'}{(1+e^{-x})^2} = \frac{-1' - (e^{-x})'}{(1+e^{-x})^2} = \frac{0 - (-x)'(e^{-x})}{(1+e^{-x})^2} =$$

$$= \frac{-(-1)(e^{-x})}{(1+e^{-x})^2} = \frac{e^{-x}}{(1+e^{-x})^2} = \left(\frac{1}{1+e^{-x}}\right)\left(\frac{e^{-x}}{1+e^{-x}}\right) =$$

$$= \sigma(x)\left(\frac{+1-1+e^{-x}}{1+e^{-x}}\right) = \sigma(x)\left(\frac{1+e^{-x}}{1+e^{-x}} - \frac{1}{1+e^{-x}}\right) = \sigma(x)(1-\sigma(x))$$

partial derivative of $J(\Theta)$):

$$\frac{\partial}{\partial\theta_j}J(\theta) = \frac{\partial}{\partial\theta_j}\frac{-1}{m}\sum_{i=1}^{m}\left[y^{(i)}log(h_\theta(x^{(i)})) + (1-y^{(i)})log(1-h_\theta(x^{(i)}))\right] =$$

$$= -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\frac{\partial}{\partial\theta_j}log(h_\theta(x^{(i)})) + (1-y^{(i)})\frac{\partial}{\partial\theta_j}log(1-h_\theta(x^{(i)}))\right] =$$

$$= -\frac{1}{m}\sum_{i=1}^{m}\left[\frac{y^{(i)}\frac{\partial}{\partial\theta_j}h_\theta(x^{(i)})}{h_\theta(x^{(i)})} + \frac{(1-y^{(i)})\frac{\partial}{\partial\theta_j}(1-h_\theta(x^{(i)}))}{1-h_\theta(x^{(i)})}\right] =$$

$$= -\frac{1}{m}\sum_{i=1}^{m}\left[\frac{y^{(i)}\frac{\partial}{\partial\theta_j}\sigma(\theta^T x^{(i)})}{h_\theta(x^{(i)})} + \frac{(1-y^{(i)})\frac{\partial}{\partial\theta_j}(1-\sigma(\theta^T x^{(i)}))}{1-h_\theta(x^{(i)})}\right] =$$

$$= -\frac{1}{m}\sum_{i=1}^{m}\left[\frac{y^{(i)}\sigma(\theta^T x^{(i)})(1-\sigma(\theta^T x^{(i)}))\frac{\partial}{\partial\theta_j}\theta^T x^{(i)}}{h_\theta(x^{(i)})} + \frac{-(1-y^{(i)})\sigma(\theta^T x^{(i)})(1-\sigma(\theta^T x^{(i)}))\frac{\partial}{\partial\theta_j}\theta^T x^{(i)}}{1-h_\theta(x^{(i)})}\right] =$$

$$= -\frac{1}{m}\sum_{i=1}^{m}\left[\frac{y^{(i)}h_\theta(x^{(i)})(1-h_\theta(x^{(i)}))\frac{\partial}{\partial\theta_j}\theta^T x^{(i)}}{h_\theta(x^{(i)})} - \frac{(1-y^{(i)})h_\theta(x^{(i)})(1-h_\theta(x^{(i)}))\frac{\partial}{\partial\theta_j}\theta^T x^{(i)}}{1-h_\theta(x^{(i)})}\right] =$$

$$= -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}(1-h_\theta(x^{(i)}))x_j^{(i)} - (1-y^{(i)})h_\theta(x^{(i)})x_j^{(i)}\right] =$$

$$= -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}(1-h_\theta(x^{(i)})) - (1-y^{(i)})h_\theta(x^{(i)})\right]x_j^{(i)} =$$

$$= -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)} - y^{(i)}h_\theta(x^{(i)}) - h_\theta(x^{(i)}) + y^{(i)}h_\theta(x^{(i)})\right]x_j^{(i)} =$$

$$= -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)} - h_\theta(x^{(i)})\right]x_j^{(i)} =$$

$$= \frac{1}{m}\sum_{i=1}^{m}\left[h_\theta(x^{(i)}) - y^{(i)}\right]x_j^{(i)}$$