Machine Learning Quick Reference

Jorge Augusto Sabaliauskas

January 3, 2014

1 Linear Regression

Training set:

input	output
$X^{(1)}$	$y^{(1)}$
$X^{(2)}$	$y^{(2)}$
$\vdots$	$\vdots$
$X^{(m)}$	$y^{(m)}$

1.1 Training Algorithm

Input vector:

$$X^{(i)} = \begin{bmatrix} 1 \\ x_1^{(i)} \\ \vdots \\ x_d^{(i)} \end{bmatrix}$$

Weight vector:

$$\Theta = \begin{bmatrix} \theta_0 \\ \vdots \\ \theta_d \end{bmatrix}$$

Hypothesis:

$$h_\Theta(X^{(i)}) = \theta_0 + \theta_1 x_1^{(i)} + \ldots + \theta_d^{(i)} x_d = \Theta^T X^{(i)}$$

Cost function:

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

Gradient:

$$\frac{\partial J(\Theta)}{\partial \theta_k} = \frac{1}{m} \sum_{i=1}^{m} (\Theta^T X^{(i)} - y^{(i)}) x_k^{(i)}$$

Update rule:

$$\theta_k \leftarrow \theta_k - \alpha \frac{\partial J(\Theta)}{\partial \theta_k}$$

$$\theta_k \leftarrow \theta_k - \alpha \frac{1}{m} \sum_{i=1}^{m} (\Theta^T X^{(i)} - y^{(i)}) x_k^{(i)}$$

$$\theta_k \leftarrow \theta_k - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\Theta(X^{(i)}) - y^{(i)}) x_k^{(i)}$$

1.2 Vectorized Implementation

$$X' = \begin{bmatrix} X_1 \\ \vdots \\ X_{i+1} \end{bmatrix}$$

Gradient:

$$\frac{\partial J(\Theta)}{\partial \theta} = \frac{1}{m} X'^T (X'\Theta - Y)$$

Update rule:

$$\theta \leftarrow \theta - \alpha \frac{\partial J(\Theta)}{\partial \theta}$$

$$\theta \leftarrow \theta - \alpha \frac{1}{m} X'^T (X'\Theta - Y)$$

2 Logistic Regression

Training set:

input	output
$X^{(1)}$	$y^{(1)}$
$X^{(2)}$	$y^{(2)}$
$\vdots$	$\vdots$
$X^{(m)}$	$y^{(m)}$

2.1 Training Algorithm

Logistic function:

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

Hypothesis:

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

Cost function:

$$J(\Theta) = -\frac{1}{m}\sum_{i=1}^{m} y^{(i)} \ln(h_\Theta(X^{(i)})) + (1-y^{(i)}) \ln(1-h_\Theta(X^{(i)}))$$

Derivative of logistic function:

$$\frac{\partial f(x)}{\partial x} = \frac{-1}{(1+e^{-x})^2} e^{-x}(-1) = \frac{e^{-x}}{(1+e^{-x})^2}$$

Gradient:

$$\frac{\partial J(\Theta)}{\partial \theta_k} = -\frac{1}{m}\sum_{i=1}^{m} \frac{\partial}{\partial h_\Theta(X^{(i)})}(y^{(i)} \ln(h_\Theta(X^{(i)}))+(1-y^{(i)}) \ln(1-h_\Theta(X^{(i)}))) \frac{\partial h_\Theta(X^{(i)})}{\partial \theta_k}$$

$$\frac{\partial h_\Theta(X^{(i)})}{\partial \theta_k} = \frac{\partial f(\Theta^T X^{(i)})}{\partial \Theta^T X^{(i)}} \frac{\partial \Theta^T X^{(i)}}{\partial \theta_k}$$

$$\frac{\partial f(\Theta^T X^{(i)})}{\partial \Theta^T X^{(i)}} = \frac{e^{-\Theta^T X^{(i)}}}{(1+e^{-\Theta^T X^{(i)}})^2}$$

$$\frac{\partial \Theta^T X^{(i)}}{\partial \theta_k} = \frac{\partial}{\partial \theta_k}(\theta_0 x_0 + \ldots + \theta_k x_k + \ldots + \theta_d x_d) = x_k$$

$$\frac{\partial h_\Theta(X^{(i)})}{\partial \theta_k} = \frac{e^{-\Theta^T X^{(i)}}}{(1+e^{-\Theta^T X^{(i)}})^2} x_k$$

$$\frac{\partial J(\Theta)}{\partial \theta_k} = -\frac{1}{m}\sum_{i=1}^{m}(y^{(i)} \frac{1}{h_\Theta(X^{(i)})} + (1-y^{(i)}) \frac{-1}{1-h_\Theta(X^{(i)})}) \frac{\partial h_\Theta(X^{(i)})}{\partial \theta_k}$$

$$\frac{\partial J(\Theta)}{\partial \theta_k} = -\frac{1}{m}\sum_{i=1}^{m}(y^{(i)} \frac{1}{\frac{1}{1+e^{\Theta^T X^{(i)}}}} + (y^{(i)}-1) \frac{1}{1-\frac{1}{1+e^{\Theta^T X^{(i)}}}}) \frac{\partial h_\Theta(X^{(i)})}{\partial \theta_k}$$

$$\frac{\partial J(\Theta)}{\partial \theta_k} = -\frac{1}{m}\sum_{i=1}^{m}(y^{(i)}\frac{1}{\frac{1}{1+e^{-\Theta^T X^{(i)}}}} + (y^{(i)}-1)\frac{1}{\frac{e^{-\Theta^T X^{(i)}}}{1+e^{-\Theta^T X^{(i)}}}})\frac{\partial h_\Theta(X^{(i)})}{\partial \theta_k}$$

$$\frac{\partial J(\Theta)}{\partial \theta_k} = -\frac{1}{m}\sum_{i=1}^{m}(y^{(i)}(1+e^{-\Theta^T X^{(i)}}) + (y^{(i)}-1)\frac{1+e^{-\Theta^T X^{(i)}}}{e^{-\Theta^T X^{(i)}}})\frac{\partial h_{-\Theta}(X^{(i)})}{\partial \theta_k}$$

$$\frac{\partial J(\Theta)}{\partial \theta_k} = -\frac{1}{m}\sum_{i=1}^{m}(y^{(i)}\frac{e^{-\Theta^T X^{(i)}}}{e^{-\Theta^T X^{(i)}}}(1+e^{-\Theta^T X^{(i)}}) + (y^{(i)}-1)\frac{1+e^{-\Theta^T X^{(i)}}}{e^{-\Theta^T X^{(i)}}})\frac{e^{-\Theta^T X^{(i)}}}{(1+e^{-\Theta^T X^{(i)}})^2}x_k$$

$$\frac{\partial J(\Theta)}{\partial \theta_k} = -\frac{1}{m}\sum_{i=1}^{m}(y^{(i)}e^{i-\Theta^T X^{(i)}} + y^{(i)} - 1)\frac{1}{1+e^{-\Theta^T X^{(i)}}}x_k$$

$$\frac{\partial J(\Theta)}{\partial \theta_k} = -\frac{1}{m}\sum_{i=1}^{m}(y^{(i)}(1+e^{-\Theta^T X^{(i)}}) - 1)\frac{1}{1+e^{-\Theta^T X^{(i)}}}x_k$$

$$\frac{\partial J(\Theta)}{\partial \theta_k} = -\frac{1}{m}\sum_{i=1}^{m}(y^{(i)} - \frac{1}{1+e^{-\Theta^T X^{(i)}}})x_k$$

$$\frac{\partial J(\Theta)}{\partial \theta_k} = -\frac{1}{m}\sum_{i=1}^{m}(y^{(i)} - h_\Theta(X^{(i)}))x_k = \frac{1}{m}\sum_{i=1}^{m}(h_\Theta(X^{(i)}) - y^{(i)})x_k$$

Update rule:

$$\theta_k \leftarrow \theta_k - \alpha\frac{\partial J(\Theta)}{\partial \theta_k}$$

$$\theta_k \leftarrow \theta_k - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\Theta(X^{(i)}) - y^{(i)})x_k$$

3 Vectorized Implementation

$$X' = \begin{bmatrix} X_1 \\ \vdots \\ X_{i+1} \end{bmatrix}$$

$$\frac{\partial J(\Theta)}{\partial \theta} = \frac{1}{m}X'^T(f(X'\theta) - Y)$$

$$\theta \leftarrow \theta - \alpha\frac{1}{m}X'^T(f(X'\theta) - Y)$$

4 Neural Network

4.1 3 Layers

Neural network with 3 layers: d, q, r units on layer 1, 2, 3 respectively.

$$\text{Layer 3}\left\{\begin{array}{l} y_1^{(3)} = \theta_{0,1}^{(2)}1 + \theta_{1,1}^{(2)}a_1^{(2)} + \ldots + \theta_{q,1}^{(2)}a_q^{(2)} \\ y_r^{(3)} = \theta_{0,r}^{(2)}1 + \theta_{1,r}^{(2)}a_1^{(2)} + \ldots + \theta_{q,r}^{(2)}a_q^{(2)} \end{array}\right. \Rightarrow \begin{array}{l} a_1^{(3)} = f(y_1^{(3)}) \\ a_r^{(3)} = f(y_r^{(3)}) \end{array}$$

$$\text{Layer 2}\left\{\begin{array}{l} y_1^{(2)} = \theta_{0,1}^{(1)}1 + \theta_{1,1}^{(1)}a_1^{(1)} + \ldots + \theta_{d,1}^{(1)}a_d^{(1)} \\ y_q^{(2)} = \theta_{0,q}^{(1)}1 + \theta_{1,q}^{(1)}a_1^{(1)} + \ldots + \theta_{d,q}^{(1)}a_d^{(1)} \end{array}\right. \Rightarrow \begin{array}{l} a_1^{(2)} = f(y_1^{(2)}) \\ a_q^{(2)} = f(y_q^{(2)}) \end{array}$$

$$\Theta^{(2)} = \begin{bmatrix} \theta_{0,1}^{(2)} & \theta_{1,1}^{(2)} & \ldots & \theta_{q,1}^{(2)} \\ \vdots & \vdots & \ldots & \vdots \\ \theta_{0,r}^{(2)} & \theta_{1,r}^{(2)} & \ldots & \theta_{q,r}^{(2)} \end{bmatrix}$$

$$\Theta^{(1)} = \begin{bmatrix} \theta_{0,1}^{(1)} & \theta_{1,1}^{(1)} & \ldots & \theta_{d,1}^{(1)} \\ \vdots & \vdots & \ldots & \vdots \\ \theta_{0,q}^{(1)} & \theta_{1,q}^{(1)} & \ldots & \theta_{d,q}^{(1)} \end{bmatrix}$$

Cost function for the neural network:

$$J(\Theta^{(1)}, \Theta^{(2)}) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{r} y_j^{(i)} \ln(a_j^{(3)}) + (1 - y_j^{(i)}) \ln(1 - a_j^{(3)})$$

First we will derive $J(\Theta^{(1)}, \Theta^{(2)})$ with respect to $y_s^{(3)}$ $(s \in \{1, \ldots, r\})$.

$$\frac{\partial J(\Theta^{(1)}, \Theta^{(2)})}{\partial y_s^{(3)}} = -\frac{1}{m}\sum_{i=1}^{m}\left(a_s^{(3)} - y_s^{(i)}\right) = \frac{1}{m}\sum_{i=1}^{m}\left(y_s^{(i)} - a_s^{(3)}\right) = \frac{1}{m}\sum_{i=1}^{m}\delta_s^{(3)}\,(i)$$

Gradient for $\theta_{s,t}^{(2)}$ $(s \in \{0, \ldots, q+1\},\, t \in \{1, \ldots, r\})$:

$$\frac{\partial J(\Theta^{(1)}, \Theta^{(2)})}{\partial \theta_{s,t}^{(2)}} = \sum_{j=1}^{r}\frac{\partial J(\Theta^{(1)}, \Theta^{(2)})}{\partial y_j^{(3)}}\frac{\partial y_j^{(3)}}{\theta_{s,t}^{(2)}} = \frac{\partial J(\Theta^{(1)}, \Theta^{(2)})}{\partial y_t^{(3)}}\frac{\partial y_t^{(3)}}{\theta_{s,t}^{(2)}} = \frac{1}{m}\sum_{i=1}^{m}\delta_t^{(3)}\,(i)\,a_s^{(2)}$$

The error for each estimation on layer 3 is given by:

$$\delta_t^{(3)}\,(i) = y_t^{(i)} - a_t^{(3)}$$

Gradient for $\theta_{s,t}^{(1)}$ $(s \in \{0, \ldots, d+1\},\, t \in \{1, \ldots, q\})$

$$
\begin{aligned}
\frac{\partial J(\Theta^{(1)},\Theta^{(2)})}{\partial \theta^{(1)}_{s,t}} &= \textstyle\sum_{j=1}^{r} \frac{\partial J(\Theta^{(1)},\Theta^{(2)})}{\partial y^{(3)}_j} \frac{\partial y^{(3)}_j}{\partial a^{(2)}_t} \frac{\partial a^{(2)}_t}{\partial y^{(2)}_t} \frac{\partial y^{(2)}_t}{\partial \theta^{(1)}_{s,t}} \\
&= \tfrac{1}{m} \textstyle\sum_{j=1}^{r} \sum_{i=1}^{m} \delta^{(3)}_j \theta^{(2)}_{t,j} . f'(y^{(2)}_t) a^{(1)}_s \\
&= \tfrac{1}{m} \textstyle\sum_{i=1}^{m} \left(\sum_{j=1}^{r} \delta^{(3)}_j \theta^{(2)}_{t,j} . f'(y^{(2)}_t) \right) a^{(1)}_s \\
&= \tfrac{1}{m} \textstyle\sum_{i=1}^{m} \delta^{(2)}_t \, (i) \, a^{(1)}_s
\end{aligned}
$$

The error for each estimation on layer 2 is given by:

$$
\delta^{(2)}_t \, (i) = \sum_{j=1}^{r} \delta^{(3)}_j \, (i) \, \theta^{(2)}_{t,j} f'(y^{(2)}_t)
$$

The update rule for each layer is given by:

$$
\begin{cases}
\theta^{(2)}_{s,t} \leftarrow \theta^{(2)}_{s,t} - \alpha \frac{1}{m} \sum_{i=1}^{m} \delta^{(3)}_t \, (i) \, a^{(2)}_s \\
\theta^{(1)}_{s,t} \leftarrow \theta^{(1)}_{s,t} - \alpha \frac{1}{m} \sum_{i=1}^{m} \delta^{(2)}_t \, (i) \, a^{(1)}_s
\end{cases}
$$

4.2 L layers

Generalizing to a neural network with L layers where n_l = number of units on layer m. Outputs for layers l ($l \in \{1, \ldots, L-1\}$)

$$
\text{Layer (l+1)} \begin{cases}
y^{(l+1)}_1 = \theta^{(l)}_{0,1} 1 + \theta^{(l)}_{1,1} a^{(l)}_1 + \ldots + \theta^{(l)}_{n_l,1} a^{(l)}_{n_l} \\
y^{(l+1)}_{n_{l+1}} = \theta^{(l)}_{0,n_{l+1}} 1 + \theta^{(l)}_{1,n_{l+1}} a^{(l)}_1 + \ldots + \theta^{(l)}_{n_l,n_{l+1}} a^{(l)}_{n_l}
\end{cases}
\Rightarrow \quad
\begin{aligned}
a^{(l+1)}_1 &= f(y^{(l+1)}_1) \\
a^{(l+1)}_{n_{l+1}} &= f(y^{(l+1)}_{n_{l+1}})
\end{aligned}
$$

$$
\Theta^{(l)} = \begin{bmatrix}
\theta^{(l)}_{0,1} & \theta^{(l)}_{1,1} a^{(l)}_1 & \ldots & \theta^{(l)}_{n_l,1} \\
\theta^{(l)}_{0,n_{l+1}} & \theta^{(l)}_{1,n_{l+1}} & \ldots & \theta^{(l)}_{n_l,n_{l+1}}
\end{bmatrix}
$$

Error on layer l ($l \in \{2, \ldots, L\}$):

$$
\delta^{(l)}_t = \begin{cases}
y^{(i)}_t - a^{(L)}_t & \text{if } l = L \\
\sum_{j=1}^{n^{l+1}} \delta^{(l+1)}_j \, (i) \, \theta^{(l)}_{t,j} f'(y^{(l)}_t) & \text{otherwise}
\end{cases}
$$

Gradient for layer l ($l \in \{1, \ldots, L-1\}$):

$$
\frac{\partial J(\Theta^1, \ldots, \Theta^L)}{\partial \theta^{(l)}_{s,t}} = \frac{1}{m} \sum_{i=1}^{m} \delta^{(l+1)}_t \, (i) \, a^{(l)}_s
$$

Update rule for layer l ($l \in \{1, \ldots, L-1\}$):

$$
\theta^{(l)}_{s,t} \leftarrow \theta^{(l)}_{s,t} - \alpha \frac{1}{m} \sum_{i=1}^{m} \delta^{(l+1)}_t \, (i) \, a^{(l)}_s
$$

4.3 Vectorized Implementation

Forward propagation for layer l ($l \in \{1, \ldots, L-1\}$):

$$A^{(l)} = \begin{bmatrix} a_1^{(l)} \\ \vdots \\ a_{n_l}^{(l)} \end{bmatrix}$$

$$Y^{(l+1)} = \Theta^{(l)} \begin{bmatrix} 1 \\ A^{(l)} \end{bmatrix}$$

$$A^{(l+1)} = \begin{bmatrix} f(y_1^{(l+1)}) \\ \vdots \\ f(y_{n_{l+1}}^{(l+1)}) \end{bmatrix}$$

Back propagation for layer l

$$\Delta^{(l)}(i) = \begin{bmatrix} \delta_1^{(l)}(i) \\ \vdots \\ \delta_{n_l}^{(l)}(i) \end{bmatrix}$$

$$\Theta'^{(l)} = \begin{bmatrix} \theta_{1,1}^{(l)} & \cdots & \theta_{n_l,1}^{(l)} \\ \theta_{1,n_{l+1}}^{(l)} & \cdots & \theta_{n_l,n_{l+1}}^{(l)} \end{bmatrix}$$

$$\Delta^{(l)}(i) = (\Theta'^{(l)})^T \Delta^{(l+1)}(i) \circ f'(y^{(l)})$$

$$D^{(l)}(i) = \Delta^{(l+1)}(i) \begin{bmatrix} 1 \\ A^{(l)}(i) \end{bmatrix}^T$$

$$\frac{\partial J(\Theta^1, \ldots, \Theta^L)}{\partial \Theta^{(l)}} = \frac{1}{m} \sum_{i=1}^{m} D^{(l)}(i)$$

$$\Theta^{(l)} \leftarrow \Theta^{(l)} - \alpha \frac{\partial J(\Theta^1, \ldots, \Theta^L)}{\partial \Theta^{(l)}}$$

$$\Theta^{(l)} \leftarrow \Theta^{(l)} - \alpha \frac{1}{m} \sum_{i=1}^{m} D^{(l)}(i)$$