Machine Learning Course Notes
Stanford – Coursera

Bartol Freskura

29-12-2015

Contents

1 Week 5

1.1 Cost Function and Backpropagation

1.1.1 Cost Function for neural networks

$$J(\Theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{k=1}^{K} = y_k^{(i)} \log(h(x^{(i)})) + (1 + y_k^{(i)}) \log\left(1 - (h(x^{(i)}))_k\right)\right] + \frac{\lambda}{2m}\sum_{l=1}^{L-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}\left(\theta_{ji}^{(l)}\right)^2$$

1.1.2 Calculating cost error

$$\delta_j^{(L)} = a_j^{(L)} - y_j \text{ — error of node } j \text{ in layer } L$$
$$g'(z) = \tfrac{1}{1+e^{-z}} .* \left(1 - \tfrac{1}{1+e^{-z}}\right) \text{ — derivation}$$
$$\delta^{(L-1)} = (\theta^{(L-1)})^T \delta^{(L)} .* g'(z^{(L-1)}) \text{ — error vector for level } L-1$$

1.1.3 Backpropagation algorithm

Algorithm 1: Backpropagation

Training set — $\left\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\right\}$
Set gradient — $\Delta_{ij}^{(l)} = 0$ for all l, i, j

for $i = 1$ *to* m **do**
  Set $a^{(1)} = x^{(i)}$
  Perform forward propagation to compute $a^{(l)}$ for $l = 2, 3, \ldots, L$
  Using $y^{(i)}$ compute $\delta^{(i)} = a^{(L)} - y^{(i)}$
  Compute $\delta^{(L-1)}, \delta^{(L-2)}, \delta^{(L-3)}, \ldots, \delta^{(2)}$
  $\Delta_{ij}^{(l)} = \Delta_{ij}^{(l)} + a_j^{(l)} \delta_i^{(l+1)}$
end

1.1.4 Gradient Checking

Suppose you have a funciton $f_i(\theta)$ that computes $\frac{\partial}{\partial\theta_i} J(\theta)$; you'd like to check if f_i is outputting correct derivative values.

$$\text{Let } \theta^{(i+)} = \theta + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \epsilon \\ \vdots \\ 0 \end{bmatrix} \qquad \text{and } \theta^{(i-)} = \theta - \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \epsilon \\ \vdots \\ 0 \end{bmatrix}$$

You can now numerically verify $f_i(\theta)$'s correctness by checking, for each i, that:

$$f_i(\theta) = \frac{J(\theta^{(i+)}) - J(\theta^{(i-)})}{2\epsilon}$$