Solutions to assignment1 of CS224n

陈承勃

May 22, 2018

1 Softmax

(a) Omitted. (b) See **q1_softmax.py**

2 Neural Network Basics

(a) $\sigma'(x) = \sigma(x)\sigma(1-x)$

(b) Assume k is the correct class, then

$CE(\boldsymbol{y}, \widehat{\boldsymbol{y}}) = -y_k \log \widehat{y_k} = -\log \widehat{y_k} = -\log \frac{\exp(\boldsymbol{\theta}_k)}{\sum_i \exp(\boldsymbol{\theta}_i)} = -\boldsymbol{\theta}_k + \log \sum_i \exp(\boldsymbol{\theta}_i)$.

$\therefore \frac{\partial CE(\boldsymbol{y}, \widehat{\boldsymbol{y}})}{\partial \boldsymbol{\theta}_k} = -1 + \frac{\exp(\boldsymbol{\theta}_k)}{\sum_i \exp(\boldsymbol{\theta}_i)} = \widehat{y_k} - 1,$

$\frac{\partial CE(\boldsymbol{y}, \widehat{\boldsymbol{y}})}{\partial \boldsymbol{\theta}_j} = \frac{\exp(\boldsymbol{\theta}_j)}{\sum_i \exp(\boldsymbol{\theta}_i)} = \widehat{y_j},\ j \neq k.$

$\therefore \frac{\partial CE(\boldsymbol{y}, \widehat{\boldsymbol{y}})}{\partial \boldsymbol{\theta}} = \widehat{\boldsymbol{y}} - \boldsymbol{y}$

(c) The forward propagation steps:

$\boldsymbol{Z_1} = \boldsymbol{x}\boldsymbol{W_1} + \boldsymbol{b_1}, \quad \boldsymbol{h} = sigmoid(\boldsymbol{Z_1})$

$\boldsymbol{Z_2} = \boldsymbol{h}\boldsymbol{W_2} + \boldsymbol{b_2}, \quad \widehat{\boldsymbol{y}} = sigmoid(\boldsymbol{Z_2})$

$\boldsymbol{J} = CE(\boldsymbol{y}, \widehat{\boldsymbol{y}})$

The backward propagation:

$\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{Z_2}} = \widehat{\boldsymbol{y}} - \boldsymbol{y} \triangleq \boldsymbol{\delta_1}, \quad \frac{\partial \boldsymbol{J}}{\partial \boldsymbol{h}} = \boldsymbol{\delta_1}\boldsymbol{W_2^{\mathrm{T}}} \triangleq \boldsymbol{\delta_2}$

$\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{Z_1}} = \boldsymbol{\delta_2} * \sigma'(\boldsymbol{Z_1}) \triangleq \boldsymbol{\delta_3},\ *$ denotes element-wise product.

$\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{x}} = \boldsymbol{\delta_3}\boldsymbol{W_1^{\mathrm{T}}}$

(d) $(1 + D_x) \times H + (1 + H) \times D_y$

(e) See **q2_sigmoid.py**

(f) See **q2_gradcheck.py**

(g) See **q2_neural.py**