## 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

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

$$\therefore \frac{\partial CE(\boldsymbol{y}, \widehat{\boldsymbol{y}})}{\partial \boldsymbol{\theta}_k} = -1 + \frac{\exp(\boldsymbol{\theta}_k)}{\sum_i \exp(\boldsymbol{\theta}_i)} = \widehat{y}_k - 1,$$
$$\frac{\partial CE(\boldsymbol{y}, \widehat{\boldsymbol{y}})}{\partial \boldsymbol{\theta}_j} = \frac{\exp(\boldsymbol{\theta}_j)}{\sum_i \exp(\boldsymbol{\theta}_i)} = \widehat{y}_j, \ j \neq k.$$

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

(c) The forward propagation steps:

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

$$oldsymbol{Z_2} = oldsymbol{hW_2} + oldsymbol{b_2}, \quad \widehat{oldsymbol{y}} = sigmoid(oldsymbol{Z_2})$$

$$J = CE(y, \hat{y})$$

The backward propagation:

$$rac{\partial J}{\partial oldsymbol{Z_2}} = \widehat{oldsymbol{y}} - oldsymbol{y} riangleq oldsymbol{\delta_1}, \quad rac{\partial J}{\partial oldsymbol{h}} = oldsymbol{\delta_1} oldsymbol{W}_{oldsymbol{2}}^{oldsymbol{T}} riangleq oldsymbol{\delta_2}$$

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

$$rac{\partial J}{\partial x} = oldsymbol{\delta_3} W_1^{
m T}$$

- (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