CS224n HW1

Xinyu Tan

June 4, 2017

1 Softmax

(a)

Take a look at any element i,

$$\text{softmax}(\boldsymbol{x}+c)_i = \frac{e^{\boldsymbol{x}_i+c}}{\sum_j e^{\boldsymbol{x}_j+c}} = \frac{e^c \cdot e^{\boldsymbol{x}_i}}{e^c \cdot \sum_j e^{\boldsymbol{x}_j}} = \text{softmax}(\boldsymbol{x})_i$$

Therefore, we have $\text{softmax}(\boldsymbol{x}+c) = \text{softmax}(\boldsymbol{x})$

(b)

Note the first case illustrate the broadcasting principle (dimension match) in numpy:

```
if len(x.shape) > 1:
  #matrix
  x = x - np.max(x, axis=1, keepdims=True)
  x = np.exp(x) / np.sum(np.exp(x), axis=1, keepdims=True)
else:
  # vector
  x = x - x.max() # normalize
  x = np.exp(x)/np.sum(np.exp(x))
```

2 Neural Network Basics

(a)

$$\frac{d\sigma(x)}{dx} = \frac{e^{-x}}{(1-e^{-x})^2} = \sigma(x)(1-\sigma(x))$$

(b)

First, we have

$$\hat{y}_i = \frac{e^{\theta_i}}{\sum_j e^{\theta_j}}$$

For one-hot encoding, only k-th element in $\boldsymbol{y}$ is *one*, so we have

$$\begin{aligned} CE(\boldsymbol{y}, \hat{\boldsymbol{y}}) &= -\log \hat{y}_k = -\log \frac{e^{\theta_k}}{\sum_j e^{\theta_j}} \\ &= -\theta_k + \log \sum_j e^{\theta_j} \end{aligned}$$

Therefore,

$$\frac{\partial CE(\boldsymbol{y}, \hat{\boldsymbol{y}})}{\partial \theta_k} = -1 + \frac{e^{\theta_k}}{\sum_j e^{\theta_j}}$$
$$\frac{\partial CE(\boldsymbol{y}, \hat{\boldsymbol{y}})}{\partial \theta_i} = \frac{e^{\theta_i}}{\sum_j e^{\theta_j}}, \forall i \neq k$$

Put them altogether,

$$\frac{\partial CE(\boldsymbol{y}, \hat{\boldsymbol{y}})}{\partial \boldsymbol{\theta}} = -\boldsymbol{y} + \mathrm{softmax}(\boldsymbol{\theta}) = \hat{\boldsymbol{y}} - \boldsymbol{y}$$