CS 224n: Assignment #1

Kristian Hartikainen

November 18, 2017

1 Softmax

(a)

Softmax is invariant to constant offsets in the input:

$$
\begin{aligned}
softmax(x)_i &= \frac{e^{x_i}}{\sum_j e^{x_j}} = \frac{e^{x_i}}{\sum_j e^{x_j}}\frac{e^c}{e^c} \\
&= \frac{e^{x_i}e^c}{\sum_j e^{x_j}e^c} = \frac{e^{x_i+c}}{\sum_j e^{x_j+c}} \\
&= softmax(x+c)_i
\end{aligned}
$$

2 Neural Network Basics

(a)

$$
\begin{aligned}
\frac{\partial}{\partial x}\sigma(x) &= \frac{\partial}{\partial x}\frac{1}{1+e^{-x}} \\
&= -1 * (1+e^{-x})^{-2} * \frac{\partial}{\partial x}(1+e^{-x}) \\
&= -\frac{-e^{-x}}{(1+e^{-x})^2} \\
&= \frac{e^{-x}+1-1}{(1+e^{-x})^2} \\
&= \frac{1+e^{-x}-1}{1+e^{-x}}\frac{1}{1+e^{-x}} \\
&= (1-\frac{1}{1+e^{-x}})\frac{1}{1+e^{-x}} \\
&= (1-\sigma(x))\sigma(x)
\end{aligned}
$$

(b)

Let $\hat{y} = softmax(\theta)$ and $CE(y, \hat{y}) = -\sum_i ylog(\hat{y})$. Then,

$$\begin{aligned}
\frac{\partial}{\partial \theta_k} CE(y, \hat{y}) &= \frac{\partial}{\partial \theta_k} - \sum_i y_i log(\hat{y}_i) \\
&= -\sum_i y_i \frac{\partial}{\partial \theta_k} log(\hat{y}_i) \\
&= -\sum_i y_i \frac{\partial}{\partial \theta_k} log(\frac{e^{\theta_i}}{\sum_j e^{\theta_j}}) \\
&= -\sum_i y_i \frac{\partial}{\partial \theta_k} (log(e^{\theta_i}) - log(\sum_j e^{\theta_j})) \\
&= -\sum_i y_i \frac{\partial}{\partial \theta_k} (\theta_i - log(\sum_j e^{\theta_j})) \\
&= -\sum_i y_i (\frac{\partial \theta_i}{\partial \theta_k} - \frac{\partial}{\partial \theta_k} log(\sum_j e^{\theta_j})) \\
&= -\sum_i y_i (\mathbf{1}_{i=k} - \frac{1}{\sum_j e^{\theta_j}} \sum_j \frac{\partial}{\partial \theta_k} e^{\theta_j}) \\
&= -\sum_i y_i (\mathbf{1}_{i=k} - \frac{e^{\theta_k}}{\sum_j e^{\theta_j}}) \\
&= -\sum_i y_i (\mathbf{1}_{i=k} - \hat{y}_k) \\
&= -y_k(1 - \hat{y}_k) + \sum_{i \neq k} y_i \hat{y}_k \\
&= -y_k + y_k \hat{y}_k + \sum_{i \neq k} y_i \hat{y}_k \\
&= -y_k + \sum_i y_i \hat{y}_k && \sum_i y_i = 1 \\
&= \hat{y}_k - y_k
\end{aligned}$$

Thus,

$$\frac{\partial}{\partial \theta} CE(y, \hat{y}) = \hat{y} - y$$

(c)

$$\begin{aligned}\hat{y} &= softmax(z_2)\\ z_2 &= hW_2 + b_2\\ h &= \sigma(z_1)\\ z_1 &= XW_1 + b_1\end{aligned}$$

Given the output $\hat{y}$, the cross-entropy loss for the network is defined as:

$$J(\theta) = CE(y, \hat{y}) = -\sum_i y_i log(\hat{y}_i)$$

The gradients of the network can be defined using chain-rule. Starting from $\frac{\partial J}{\partial z_2} = \frac{\partial CE(y,\hat{y})}{\partial z_2} = y - \hat{y}$ we get derivatives for J w.r.t. h and z_2 as follows:

$$\begin{aligned}\frac{\partial J}{\partial h} &= \frac{\partial J}{\partial z_2}\frac{\partial z_2}{\partial h} = (y - \hat{y})W_2^T\\ \frac{\partial J}{\partial z_1} &= \frac{\partial J}{\partial h}\frac{\partial h}{\partial z_1} = (y - \hat{y})W_2^T diag(\sigma'(z_1))\end{aligned}$$

And the derivative w.r.t. the network parameters W_1, $b1$, W_2 and b_2 and input X as follows:

$$\begin{aligned}\frac{\partial J}{\partial b_2} &= \frac{\partial J}{\partial z_2}\frac{\partial z_2}{\partial b_2} = (y - \hat{y})[11\ldots1]^T = \sum_i y_i - \hat{y}_i\\ \frac{\partial J}{\partial W_2} &= \frac{\partial J}{\partial z_2}\frac{\partial z_2}{\partial W_2} = (y - \hat{y})h_1.\\ \frac{\partial J}{\partial b_1} &= \frac{\partial J}{\partial z_1}\frac{\partial z_1}{\partial b_1} = \sum_i (\frac{\partial J}{\partial z_1})_i\\ \frac{\partial J}{\partial W_1} &= \frac{\partial J}{\partial z_1}\frac{\partial z_1}{\partial W_1} = (y - \hat{y})W_2^T diag(\sigma'(z_1))X\\ \frac{\partial J}{\partial X} &= \frac{\partial J}{\partial z_1}\frac{\partial z_1}{\partial X} = (y - \hat{y})W_2^T diag(\sigma'(z_1))W_1^T\end{aligned}$$

(d)

Assuming the input for this neural network is D_x-dimensional, the output is D_y-dimensional, and there are H hidden units, the contains $D_xH + H + HD_y + D_y$ parameters.

3 word2vec

(a)

(b)

(c)

(d)

4 Sentiment Analysis

(b)

(d)