SUID:06349270

Solutions for CS 224N Assignment#2 word2vec written:

(a). Since y is one hot vector, so $-\sum_{w \in Vocab} y_w \log(\hat{y}_w) = -\log(\hat{y}_o) + 0$ (when w = o ,it`s $-\log(\hat{y}_o)$, else 0)

(b) $\frac{\partial J}{\partial v_c} = U(\hat{y} - y)$

(c) $\frac{\partial J}{\partial u_w} = \begin{cases} (\hat{y}_w - 1) v_c & w = o \\ \hat{y}_w v_c & w \neq o \end{cases}$

(d) $\sigma(\acute{x}) = \sigma(x)(1 - \sigma(x))$

(e) $\frac{\partial J}{\partial v_c} = (\sigma(u_o^T v_c) - 1) u_o - \sum_{k=1}^{K} (\sigma(-u_k^T v_c) - 1) u_k$

$\frac{\partial J}{\partial u_o} = (\sigma(u_o^T v_c) - 1) v_c$

$\frac{\partial J}{\partial u_k} = (1 - \sigma(-u_k^T v_c)) v_c$ for k = 1, 2, ,3 …. K

Q:Why this neg-sample is much more efficient?

A:From native softmax $\frac{\partial J}{\partial v_c} = U(\hat{y} - y)$, we could see for each *calaction for* $\frac{\partial J}{\partial v_c}$ it needs compute with the all ‘outside’ vectors U contains a vector for every word in vocabulary. however neg-sample loss function only needs calculate fixed size matrices rather than calculate all words vectors like U and V, so neg-sample is much more efficient and scale-able.

(f)

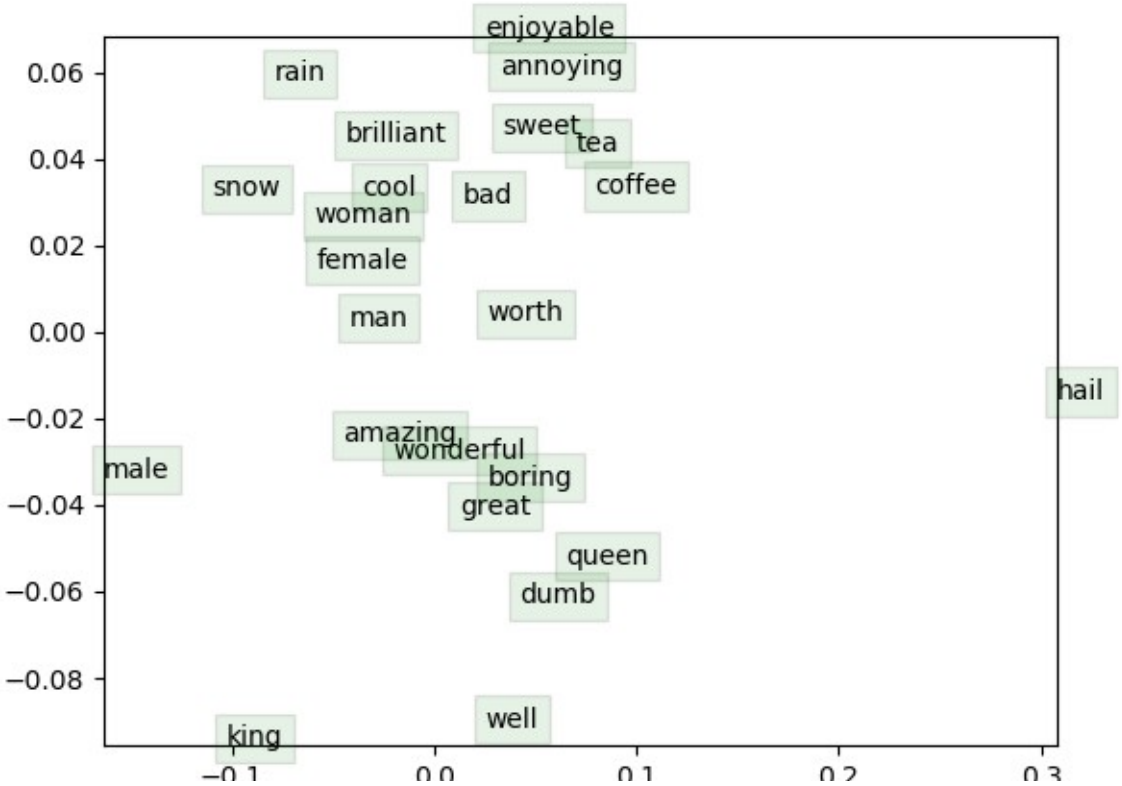
(i) $\partial J_{skip-gram}(v_c, w_{t-m}), \ldots w_{t+m}, U / \partial U = \sum_{-m \le j \le m, j \neq 0} \frac{\partial J(v_o, w_{w+j}, U)}{\partial U}$

(ii) $\partial J_{skip-gram}(v_c, w_{t-m}), \ldots w_{t+m}, U / \partial v_c = \sum_{-m \le j \le m, j \neq 0} \frac{\partial J(v_o, w_{w+j}, U)}{\partial v_c}$

(iii) $\partial J_{skip-gram}(v_c, w_{t-m}), \ldots w_{t+m}, U / \partial v_w (when\, w \neq c) = 0$

SUID:06349270

2. (c)



Loss :