Problem Set 1

1. Word2Vec

word o and c → predict $P(O=o \mid C=c)$

context window of c : W_c

2 2D Vectors : $\vec{u}_w$ outside vector
w is outsid word

$\vec{v}_w$ center vector
w is center word

$\vec{u}_k$ outside vectors word indexed by k

$\vec{v}_k$ center . . .

U k^{th} col of U and V are $\vec{u}_k$ and $\vec{v}_k$

V

Softmax

$$P(O=o \mid C=c) = \frac{\exp(\vec{u}^T \vec{c})}{\sum_{w \in Vocab} \exp(\vec{u}^T \vec{v}_c)}$$

To minimize −log / maximize log likelihood

To minimize $-\log$ / maximize $\log$ likelihood

$$-\sum_{c \in C} \sum_{o \in W_c} \log P(O=o \mid C=c)$$

Single pair of words c and o loss:

$$\mathcal{L}(\vec{v}_c, o, U) = -\log P(O=o \mid C=c)$$

i.e. $y = 1$ at position of word o, 0 everywhere else

$$\Rightarrow y_o = 1, \; y_w = 0 \; (w \neq o)$$

$$-\sum_{w \in Vocab} y_w \log \hat{y}_w$$

$$= -\left[y_o \log \hat{y}_o + \sum_{w \neq o} y_w \log \hat{y}_w \right]$$

$$= -\left[\overset{1}{y_o} \log \hat{y}_o + \sum_{w \neq o} 0 \cdot \log \hat{y}_w \right]$$

$$= -\log \hat{y}_o$$

since: $\hat{y}_o = P(O=o \mid C=c)$

since: $\hat{y}_o = P(O=o \mid C=c)$

$$\Rightarrow -\log \hat{y}_o = -\log P(O=o \mid C=c)$$

this part is $\mathcal{L}(\vec{v}_c, o, U)$

So all 3 parts are equal

1.2 Gradient of loss function

$$\mathcal{L}(\vec{v}_c, o, U) = -\log P(O=o \mid C=c)$$

$$= -\log \left(\frac{\exp(\vec{u}_o^T \vec{v}_c)}{\sum_{w \in Vocab} \exp(\vec{u}_w \vec{v}_c)} \right)$$

$$= -\vec{u}_o \vec{v}_c + \log \left(\sum_{w \in Vocab} \exp(\vec{u}_w \vec{v}_c) \right)$$

$$\nabla_{\vec{v}_c} (-\vec{u}_o \vec{v}_c) = -\vec{u}_o$$

$$\nabla_{\vec{v}_c} \log \left(\sum_{w \in vocab} \exp(\vec{u}_w^T \vec{v}_c) \right)$$

$$= \frac{1}{\sum_{w \in vocab} \exp(\vec{u}_w^T \vec{v}_c)} \cdot \nabla_{\vec{v}_c} \left(\sum_{w \in vocab} \exp(\vec{u}_w^T \vec{v}_c) \right)$$

$$= \frac{1}{\sum_{w \in vocab} \exp(\vec{u}_w^T \vec{v}_c)} \cdot \sum_{w \in vocab} \exp(\vec{u}_w^T \vec{v}_c) \cdot \vec{u}_w$$

$$= \sum_{w \in vocab} \frac{\exp(\vec{u}_w \vec{v}_c)}{\sum_{w \in vocab} \exp(\vec{u}_w^T \vec{v}_c)} \cdot \vec{u}_w \qquad \rightarrow \hat{y}_w$$

$$= \sum_{w \in vocab} \hat{y}_w \vec{u}_w$$

$$\Rightarrow \nabla_{\vec{v}_c} \mathcal{L} = -\vec{u}_o \,(1) + \sum_{w \in vocab} \hat{y}_w \vec{u}_w \,(2)$$

U : matrix, col k is $\vec{u}_k$

$\vec{y}$: 1 at O, 0 everywhere else

$$U\vec{y} = \vec{u}_o \cdot 1 + \sum_{w \neq o} \vec{u}_w \cdot 0 = \vec{u}_o \,(1)$$

$\hat{\vec{y}}$ is p vector, w is $\hat{y}_w$

$U\hat{\vec{y}} = \sum_{w \in vocab} \vec{U}_w \cdot \vec{y}_w$ (2)

$\Rightarrow \nabla_{\vec{v}_c} \mathcal{L} = U\hat{\vec{y}} - U\vec{y}$

1.3 Improvement

$\vec{v}_c - \nabla_{\vec{v}_c} \mathcal{L}$

$= \vec{v}_c - \alpha \nabla_{\vec{v}_c} \mathcal{L}$

$= \vec{v}_c - \alpha (U\hat{\vec{y}} - U\vec{y})$

$= \vec{v}_c - \alpha U\hat{\vec{y}} + \alpha U\vec{y}$

$- \alpha U\vec{y}$: subtract weighted average of all vectors to move $\vec{v}_c$ away from average of all other outside vectors to reduce

- $\alpha U \vec{y}$: subtract weighted average of all vectors to move $\vec{v}_c$ away from average of all other outside vectors to reduce potencial false positive words.

$\alpha U \vec{y}$: add true outside vector so that $\vec{v}_c$ move towards the correct word.

1.4 single pair C cenel 0

$$\nabla_{\vec{u}_w} \mathcal{L} = \begin{cases} \vec{y}_w \vec{v}_c & \text{if } w \neq o \\ (\hat{y}_o - 1)\vec{v}_c & \text{if } w = o \end{cases}$$

$$\mathcal{L} = -\log \frac{\exp(\vec{u}_o^T \vec{v}_c)}{\sum_{w \in vocab} \exp(\vec{u}_w^T \vec{v}_c)}$$

$$= -\vec{u}_o^T \vec{v}_c + \log \sum_{w \in vocab} \exp(\vec{u}_w^T \vec{v}_c)$$

when $w \neq 0$: $\nabla_{\vec{u}_w} (\vec{y}_w \vec{v}_c)$

1.4 single pair C and O

$$\nabla_{\vec{u}_w} \mathcal{L} = \begin{cases} \vec{y}_w \vec{v}_c & \text{if } w \neq o \\ (\hat{y}_o - 1)\vec{v}_c & \text{if } w = o \end{cases}$$

$$\mathcal{L} = -\log \frac{\exp(\vec{u}_o^T \vec{v}_c)}{\sum_{w \in vocab} \exp(\vec{u}_w^T \vec{v}_c)}$$

$$= -\vec{u}_o^T \vec{v}_c + \log \sum_{w \in vocab} \exp(\vec{u}_w^T \vec{v}_c)$$

when $w \neq o$: $\nabla_{\vec{u}_w}(\vec{y}_w \vec{v}_c)$

$$= \nabla_{\vec{u}_w}\left(-\vec{u}_o \vec{v}_c + \log \sum_{w \in vocab} \exp(\vec{u}_w^T \vec{v}_c)\right)$$

($\vec{u}_o \vec{v}_c \to 0$)

$$= \frac{1}{\sum_{w \in vocab} \exp(\vec{u}_w^T \vec{v}_c)} \cdot \nabla_{\vec{u}_w} \sum_{w \in vocab} \exp(\vec{u}_w^T \vec{v}_c)$$

$$= \frac{1}{\sum_{w \in vocab} \exp(\vec{u}_w^T \vec{v}_c)} \cdot \sum_{w \in vocab} \exp(\vec{u}_w^T \vec{v}_c)(\vec{v}_c)$$

$$= \frac{1}{\sum_{w \in vocab} \exp(\vec{u}_w^T \vec{v}_c)} \cdot \sum_{w \in vocab} \exp(\vec{u}_w^T \vec{v}_c)(\vec{v}_c)$$

$$= \hat{y}_w \vec{v}_c$$

when $w = o$: $\nabla_{\vec{u}_w} (\hat{y}_o - 1) \vec{v}_c$

$$\nabla_{\vec{u}_o} \left(-\vec{u}_o^T \vec{v}_c + \log \sum_{w \in vocab} \exp(\vec{u}_w^T \vec{v}_c) \right)$$

$-\vec{v}_c$ $\hat{y}_o \vec{v}_c$

same as above

$$= -\vec{v}_c + \hat{y}_o \vec{v}_c$$

$$= (\hat{y}_o - 1) \vec{v}_c$$

$\nabla_u \mathcal{L}$

$$U = \begin{bmatrix} \vec{u}_1 \\ \vec{u}_2 \\ \vdots \\ \vec{u}_o \\ \vdots \\ \vec{u}_{|V|} \end{bmatrix} \quad \nabla_U \mathcal{L} = \begin{bmatrix} \nabla_{\vec{u}_1} \mathcal{L} \\ \nabla_{\vec{u}_2} \mathcal{L} \\ \vdots \\ \nabla_{\vec{u}_o} \mathcal{L} \\ \vdots \\ \nabla_{\vec{u}_{|V|}} \mathcal{L} \end{bmatrix} \begin{array}{l} \rightarrow \hat{y}_w \vec{v}_c \\ \\ \\ \rightarrow (\hat{y}_o - 1) \vec{v}_c \\ \\ \end{array}$$

$\nabla_U \mathcal{L}$

$$U = \begin{bmatrix} \vec{u}_1 \\ \vec{u}_2 \\ \vdots \\ \vec{u}_0 \\ \vdots \\ \vec{u}_{|V|} \end{bmatrix} \quad \nabla_U \mathcal{L} = \begin{bmatrix} \nabla_{\vec{u}_1} \mathcal{L} \\ \nabla_{\vec{u}_2} \mathcal{L} \\ \vdots \\ \nabla_{\vec{u}_0} \mathcal{L} \\ \vdots \\ \nabla_{\vec{u}_{|V|}} \mathcal{L} \end{bmatrix} \begin{array}{l} \rightarrow \hat{y}_w \vec{v}_c \\ \nearrow \\ \nearrow \\ \rightarrow (\hat{y}_0 - 1)\vec{v}_c \end{array}$$

$$\nabla_U \mathcal{L} = \vec{v}_c (\hat{\vec{y}} - \vec{y})^T$$

1.5 L^2 normalization

L^2 makes all words with same magnitude but keeps the sign of the words. So if we only care about positivity or negativity of the words, L^2 will not impact the result. But if we need word's magnitude, L^2 would make all words same mag, so very harmful.

2 Adam Optimizer and Dropout

SGD $\vec{\theta}^{(t+1)} = \vec{\theta}^{(t)} - \alpha \nabla_{\vec{\theta}} \mathcal{L}(\theta^{(t)})$

limitation: too slow, might diverge.

same step for all, ignore freq & steepness

may jitter

2.1 Momentum

rolling average:

$$m^{(t+1)} = \beta_1 m^{(t)} + (1-\beta_1) \nabla_\theta \mathcal{L}(\theta^{(t)})$$

$$\vec{\theta}^{(t+1)} = \vec{\theta}^{(t)} - \alpha m^{(t+1)}$$

β_1 : hyperparameter 0 - 1 (usually 0.9)

$f(\vec{\theta}) = \frac{1}{2}(1000\theta_1^2 + \theta_2^2)$, $\vec{\theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$

covergence tolorance $\epsilon = 0.001$

initial $\vec{\theta}^{(0)} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, at most $i_{max} = 1000$

convergence tolerance $\varepsilon = 0.001$

initial $\vec{\theta}^{(0)} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, at most $i_{max} = 1000$

SGD with $\alpha = 0.1$

SGD with $\alpha = 0.005$

SGD with $\alpha = 0.1$ $\beta_1 = 0.9$

optimal ?

iteractions ?

2.4 Dropout

$\vec{h}$: hiddden layer with k elements.

randomly set to 0, with P_{drop}, then $\vec{h}_{drop} = \gamma \vec{d} * \vec{h}$

$\vec{d}$ same size $\vec{h}$, random

- 0 : P_{drop}
- 1 : $1 - P_{drop}$

$- \quad 1 \quad : \quad 1 - P_{drop}$

(a) y to make $\vec{h}_{drop}$ is $\vec{h}$

$$E_{P_{drop}}[(\vec{h}_{drop})_i] = \vec{h}_i$$

$i = 1, 2, \ldots, k$

prove $y = \frac{1}{1 - P_{drop}}$

since $d = 0$ or $d = 1$

$$E = y * P(d_i = 0) * h_i + y * P(d_i = 1) * h_i$$

$$= y * 0 * h_i + y * (1 - P_{drop}) * h_i$$

$$E = y * h_i (1 - P_{drop}) = \vec{h}_i$$

$$\Rightarrow y(1 - P_{drop}) = 1$$

$$\Rightarrow y = \frac{1}{1 - P_{drop}}$$

b. If $P_{drop} = 0.5$, meaning half

$$E = \gamma * h_i (1 - P_{drop}) = \vec{h_i} \quad 1$$

$$\Rightarrow \quad \gamma (1 - P_{drop}) = 1$$

$$\Rightarrow \quad \gamma = \frac{1}{1 - P_{drop}}$$

b. If $P_{drop} = 0.5$, meaning half will be dropped, $\gamma = \frac{1}{1-0.5} = 2$, meaning doubling back. So $\gamma = \frac{1}{1-P_{drop}}$ makes sense.

C. Having dropout during training can help lower the overfitting chance.
We don't do dropout during evaluation is because we need consistent input in order to compare the results to see if they remain consistent.