Tópicos selectos de ciencia de datos Tarea 2

Iván Vega Gutiérrez

20 de septiembre 2022

1 Introduction

1. Este ejercicio es sobre word embeddings con el modelo neuronal de lenguaje que vimos según la propuesta de Mikolov et al. Puedes entregar la respuesta en un PDF anexo si gustas.

Considera el modelo skip-gram de 'word2vec', donde la predicción de la palabra w_j dada cierta palabra pivote (input) w_i se calcula usando softmax:

$$\hat{y}_j = P(w_j|w_i) = \frac{\exp(\mathbf{u}'_{w_j}\mathbf{v}_{w_i})}{\sum_{v=1}^{V}\exp(\mathbf{u}'_{w_v}\mathbf{v}_{w_i})}$$

En esta expresión, $\mathbf{v}_w$ y $\mathbf{u}_w$ son dos representaciones vectoriales de una palabra w, dadas por los pesos input $\rightarrow$ hidden $\mathbf{W}_{V\times N}$ y hidden $\rightarrow$ output $\mathbf{U}_{N\times V}$, respectívamente. V es el tamaño del vocabulario y N representa el tamaño del embedding.

Suponiendo que los parámetros (embeddings) se obtienen minimizando cross-entropy:

$$L(\mathbf{y},\hat{\mathbf{y}}) = -\sum_{v\in V} y_v \log(\hat{y}_v),$$

donde $\mathbf{y}$ es un vector 'one-hot encoding' y $\hat{\mathbf{y}}$ son las predicciones usando softmax.

a) Muestra que el gradiente respecto a $\mathbf{v}_{w_i}$ representa el error de predicción (pesado) de la capa de salida, es decir:

$$\frac{\partial L}{\partial \mathbf{v}_{w_i}} = \mathbf{U}(\hat{\mathbf{y}} - \mathbf{y}),$$

donde $\mathbf{U} = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_V)$ es la matriz de todos los vectores de salida.

b) Muestra que el gradiente de los vectores de salida $\mathbf{u}_{w_t}$'s (incluyendo $\mathbf{u}_{w_j}$) representa (nuevamente) el error de predicción con diferente peso:

$$\frac{\partial L}{\partial \mathbf{U}} = \mathbf{v}_{w_i}(\hat{\mathbf{y}} - \mathbf{y})',$$

c) Repite los dos incisos anteriores pero ahora considerando que usas la función de costo con muestreo negativo, es decir:

$$L(\mathbf{v}_{w_i}, \mathbf{u}_{w_j}) = -\log(\sigma(\mathbf{u}'_{w_j}\mathbf{v}_{w_i})) - \sum_{k=1}^{K}\log(\sigma(-\mathbf{u}'_{w_k}\mathbf{v}_{w_i})),$$

con $\sigma(\cdot)$ la función sigmoide. (Observa que la expresión anterior es la función de costo para SGNS de Mikolov que vimos en clase, pero con signo negativo, ya que queremos minimizarla)

d) Explica porqué ésta función de costo es mucho más eficiente que usar la función softmax con Cross-Entropy.

Solución de a)

Tenemos que

$$\hat{y}_j = P(w_j|w_i) = \frac{\exp(\mathbf{u}'_{w_j}\mathbf{v}_{w_i})}{\sum_{v=1}^{V}\exp(\mathbf{u}'_{w_v}\mathbf{v}_{w_i})} \tag{1}$$

y

$$L(\mathbf{y},\hat{\mathbf{y}}) = -\sum_{v\in V} y_v \log(\hat{y}_v)$$

Dado que y es un vector one hot encoding, existe algún $j \in V$ tal que $y_j = 1$ y $y_l = 0$ para $l \neq j$. Luego, por (1) y propiedades de logaritmo se tiene que

$$\begin{aligned}
L(y,\hat{y}) &= -y_j \log\left(\hat{y}_j\right) \\
&= -y_j \log\left[\frac{\exp\left(u'_{w_j} v_{w_i}\right)}{\sum_{v=1}^{V}\exp\left(u'_{\omega_v} v_{w_i}\right)}\right] \\
&= -y_j \left(u_{\omega_j}^{\top} v_{w_i} - \log\left(\sum_{v=1}^{V}\exp\left(u_{\omega_v}^{\top} v_{\omega_i}\right)\right)\right) \\
&= -u_{\omega_j}^{\top} v_{w_i} + \log\left(\sum_{v=1}^{V}\exp\left(u_{\omega_v}^{\top} v_{w_i}\right)\right)
\end{aligned}$$

Entonces

$$\frac{\partial L}{\partial v_{w_i}} = -u_{w_j}^{\top} + \frac{1}{\sum_{v=1}^{V}\exp\left(u_{w_v}^{\top} v_{w_i}\right)}\left(\sum_{v=1}^{V}\exp\left(u_{w_v}^{\top} v_{w_i}\right)\right) u_{w_v}^{\top}$$

Por otro lado, notemos que

$$\frac{\sum_{i=1}^{n} x_i}{\sum_{j=1}^{n} x_j} = \sum_{i=1}^{n}\left(\frac{x_i}{\sum_{j=1}^{n} x_j}\right).$$

Asi,

$$\frac{\partial L}{\partial v_{w_i}} = -u_{w_j}^{\top} + \sum_{v=1}^{V}\left(\frac{\exp\left(u_{w_v}^{\top} v_{w_i}\right)}{\sum_{v=1}^{V}\exp\left(u_{w_v}^{\top} v_{w_i}\right)}\right) u_{w_v}^{\top}$$

Por 1), se tiene que

$$\begin{aligned}
\frac{\partial L}{\partial v_{w_i}} &= -u_{w_j}^{\top} + \sum_{v=1}^{v} \hat{y_v} u_{w_\nu}^{\top} \\
&= -u_{w_j}^{\top} + \hat{y_1} u_{w_1}^{\top} + \hat{y}_2 v_{w_2}^{\top} + \ldots + \hat{y_j} u_{w_j}^{\top} + \ldots + \hat{y}_v u_{w_v}^{\top}
\end{aligned}$$

Recordemos que $y_j = 1$ y $y_l = 0$ para $l \neq j$, en consecuencia

$$\begin{aligned}\frac{\partial L}{\partial v_{w_1}} =&(\hat{y}_1 u_{w_1}^\top - y_1 u_{w_1}^\top) + (\hat{y}_2 u_{w_2}^\top - y_2 u_{w_2}^\top) + \cdots \\ &+ (\hat{y}_j u_{w_j}^\top - y_j u_{w_j}^\top) + \cdots + (\hat{y}_v u_{w_v}^\top - y_v u_{w_v}^\top) \\ &= \sum_{i=1}^{V} (\hat{y}_i u_{w_i}^\top - y_i u_{w_i}^\top) \\ &= \sum_{i=1}^{V} (\hat{y}_i - y_i) u_{w_i}^\top \\ &= (\hat{y} - y) u^\top\end{aligned}$$

Solucion de d) Al utilizar la función softmax con cross entropy se necesitan realizar las operaciones

$$L(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{v \in V} y_v \log(\hat{y}_v),$$

y

$$\hat{y}_j = P(w_j | w_i) = \frac{\exp(\mathbf{u}'_{w_j} \mathbf{v}_{w_i})}{\sum_{v=1}^{V} \exp(\mathbf{u}'_{w_v} \mathbf{v}_{w_i})}$$

Por lo tanto, se necesitan $V \times V$ operaciones, por otro lado, de la función de muestreo negativo, solo se realizan K operaciones.

$$L(\mathbf{v}_{w_i}, \mathbf{u}_{w_j}) = -\log(\sigma(\mathbf{u}'_{w_j} \mathbf{v}_{w_i})) - \sum_{k=1}^{K} \log(\sigma(-\mathbf{u}'_{w_k} \mathbf{v}_{w_i}))$$

De esta manera, si el vocabulario incrementa al utilizar softmax con croos-entropy el costo computacional es mayor que al utilizar la función de muestreO negativo.