Signed Network Embedding in Social Media Supplementary Material

Suhang Wang* Jiliang Tang† Charu Aggarwal‡ Yi Chang§ Huan Liu¶

S1 Optimization of SiNE

Following the common way, we employ the backpropagation to optimize the deep network for SiNE [1]. The key idea of backpropagation is to update the parameters in a backward direction by propagating "errors" backward to efficiently calculate the gradients. Basically, we want to optimize Eq. (3.3) w.r.t to $\mathbf{X}$, $\mathbf{x}_0$ and θ. The key step of optimizing Eq. (3.3) is to get the gradient of $\max(0, f(\mathbf{x}_i, \mathbf{x}_k) + \delta - f(\mathbf{x}_i, \mathbf{x}_j))$ and $\max(0, f(\mathbf{x}_i, \mathbf{x}_0) + \delta - f(\mathbf{x}_i, \mathbf{x}_j))$ with respect to the parameters, $\mathbf{X}, \mathbf{x}_0$ and θ. With the gradient, we then can update the parameters using gradient descent method. Let's first analyze $\max(0, f(\mathbf{x}_i, \mathbf{x}_k) + \delta - f(\mathbf{x}_i, \mathbf{x}_j))$.

- If $\max(0, f(\mathbf{x}_i, \mathbf{x}_k) + \delta - f(\mathbf{x}_i, \mathbf{x}_j)) = 0$, or equivalent, $f(\mathbf{x}_i, \mathbf{x}_k) + \delta - f(\mathbf{x}_i, \mathbf{x}_j) \leq 0$, the parameters have already been optimized for the inputs $\mathbf{x}_i$ and $\mathbf{x}_j$. In other words, the gradient of $\max(0, f(\mathbf{x}_i, \mathbf{x}_k) + \delta - f(\mathbf{x}_i, \mathbf{x}_j))$ is 0 when $f(\mathbf{x}_i, \mathbf{x}_k) + \delta - f(\mathbf{x}_i, \mathbf{x}_j) \leq 0$.
- If $\max(0, f(\mathbf{x}_i, \mathbf{x}_k) + \delta - f(\mathbf{x}_i, \mathbf{x}_j)) > 0$, $\max(0, f(\mathbf{x}_i, \mathbf{x}_k) + \delta - f(\mathbf{x}_i, \mathbf{x}_j))$ is equal to $f(\mathbf{x}_i, \mathbf{x}_k) + \delta - f(\mathbf{x}_i, \mathbf{x}_j)$.

The same idea can be applied to $\max(0, f(\mathbf{x}_i, \mathbf{x}_0) + \delta_0 - f(\mathbf{x}_i, \mathbf{x}_j)$. Based on the aforementioned analysis, we only need to take gradient of $f(\mathbf{x}_i, \mathbf{x}_j)$ w.r.t the parameters. Then we are able to get the gradient of Eq. (3.3) with some calculations. We will start from the parameters of the N-th layer and go backward to get derivatives for other layers. First, using Eq (3.8), the derivative of $f(\mathbf{x}_i, \mathbf{x}_j)$ w.r.t $\mathbf{w}$ is given as

$$\begin{aligned} \frac{\partial f(\mathbf{x}_i, \mathbf{x}_j)}{\partial \mathbf{w}} &= [1 - f^2(\mathbf{x}_i, \mathbf{x}_j)] \frac{\partial}{\partial \mathbf{w}} [\mathbf{w}^T \mathbf{z}^{N1} + b] \\ &= [1 - f^2(\mathbf{x}_i, \mathbf{x}_j)] \mathbf{z}^{N1} \end{aligned} \tag{S1}$$

and similarly, the derivative of $f(\mathbf{x}_i, \mathbf{x}_j)$ w.r.t b is

$$\frac{\partial f(\mathbf{x}_i, \mathbf{x}_j)}{\partial b} = 1 - f^2(\mathbf{x}_i, \mathbf{x}_j) \tag{S2}$$

Next, the gradient of $f(\mathbf{x}_i, \mathbf{x}_j)$ w.r.t $\mathbf{z}^{N1}$ is given as

$$\begin{aligned} \frac{\partial f(\mathbf{x}_i, \mathbf{x}_j)}{\partial \mathbf{z}^{N1}} &= [1 - f^2(\mathbf{x}_i, \mathbf{x}_j)] \frac{\partial}{\partial \mathbf{z}^{N1}} [\mathbf{w}^T \mathbf{z}^{N1} + b] \\ &= [1 - f^2(\mathbf{x}_i, \mathbf{x}_j)] \mathbf{w} \end{aligned} \tag{S3}$$

Let $\boldsymbol{\delta}^{N1}$ be a vector with its s-th element δ_s^{N1} defined as

$$\begin{aligned} \delta_s^{N1} &= \frac{\partial f(\mathbf{x}_i, \mathbf{x}_j)}{\partial z_s^{N1}} [1 - (z_s^{N1})^2] \\ &= [1 - (z_s^{N1})^2][1 - f^2(\mathbf{x}_i, \mathbf{x}_j)] w_s \end{aligned} \tag{S4}$$

where z_s^{N1} is the s-th element of $\mathbf{z}^{N1}$. $\boldsymbol{\delta}^{N1}$ is the "error" generated by the output layer and will propagate back to the N-th layer as shown later. Using the chain rule and Eq. (3.7), the derivative of $f(\mathbf{x}_i, \mathbf{x}_j)$ w.r.t. $\mathbf{W}^N$ is given as:

$$\begin{aligned} \frac{\partial f(\mathbf{x}_i, \mathbf{x}_j)}{\partial \mathbf{W}_{st}^N} &= \sum_k \frac{\partial f(\mathbf{x}_i, \mathbf{x}_j)}{\partial z_k^{N1}} \frac{\partial z_k^{N1}}{\partial \mathbf{W}_{st}^N} \\ &= \frac{\partial f(\mathbf{x}_i, \mathbf{x}_j)}{\partial z_s^{N1}} [1 - (z_s^{N1})^2] z_t^{(N-1)1} \end{aligned} \tag{S5}$$

With Eq. (S4), the above equation is simplified as

$$\frac{\partial f(\mathbf{x}_i, \mathbf{x}_j)}{\partial \mathbf{W}_{st}^N} = \delta_s^{N1} z_t^{(N-1)1} \tag{S6}$$

Similarly, the derivative of $f(\mathbf{x}_i, \mathbf{x}_j)$ w.r.t. $\mathbf{b}^N$ is given as:

$$\frac{\partial f(\mathbf{x}_i, \mathbf{x}_j)}{\partial b_s^N} = \delta_s^{N1} \tag{S7}$$

From Eqs (S6) and (S7), we can see that the "error" $\boldsymbol{\delta}^{N1}$ is propagated backwards, i.e., it is used for the calculation of the gradients of the parameters for the N-th layer.

Generally, the "error" for the n-th layer is denoted as $\boldsymbol{\delta}^{n1}, 1 \leq n < N$, with it's s-th element defined as

$$\delta_s^{n1} = \frac{\partial f(\mathbf{x}_i, \mathbf{x}_j)}{\partial z_s^{n1}} [1 - (z_s^{n1})^2] \tag{S8}$$

*Arizona State University, suhang.wang@asu.edu
†Michigan State University, jiliang.tang@cse.msu.edu
‡IBM T.J. Watson, charu.aggarwal@us.ibm.com
§Huawei Research America, yichang@acm.org
¶Arizona State University, huan.liu@asu.edu

where the derivative of $f(\mathbf{x}_i, \mathbf{x}_j)$ w.r.t $\mathbf{z}^{n1}$ is given as

$$\begin{aligned}\frac{\partial f(\mathbf{x}_i,\mathbf{x}_j)}{\partial z_s^{n1}} &= \sum_k \frac{\partial f(\mathbf{x}_i,\mathbf{x}_j)}{\partial z_k^{(n+1)1}} \frac{\partial z_k^{(n+1)1}}{\partial z_s^{n1}} \\ &= \sum_k \frac{\partial f(\mathbf{x}_i,\mathbf{x}_j)}{\partial z_k^{(n+1)1}} [1-(z_k^{n1})^2]\mathbf{W}_{ks}^{n+1} \\ &= \sum_k \delta_k^{(n+1)1}\mathbf{W}_{ks}^{(n+1)}\end{aligned} \tag{S9}$$

Thus, we have

$$\delta_s^{n1} = [1-(z_s^{n1})^2]\sum_k \delta_k^{(n+1)1}\mathbf{W}_{ks}^{(n+1)}, 1 \le n < N \tag{S10}$$

It is clear from the above equation that the "error" $\delta_k^{(n+1)1}$ from the $(n+1)$-th layer is back propagated to the n-th layer for the calculation of δ_s^{n1}. With δ_s^{n1}, the derivative of $f(\mathbf{x}_i, \mathbf{x}_j)$ w.r.t $\mathbf{W}^n$ and $\mathbf{b}^n$ is given as

$$\begin{aligned}\frac{\partial f(\mathbf{x}_i,\mathbf{x}_j)}{\partial \mathbf{W}_{st}^{n}} &= \sum_k \frac{\partial f(\mathbf{x}_i,\mathbf{x}_j)}{\partial z_k^{n1}} \frac{\partial z_k^{n1}}{\partial \mathbf{W}_{st}^{n}} \\ &= \frac{\partial f(\mathbf{x}_i,\mathbf{x}_j)}{\partial z_s^{n1}} [1-(z_s^{n1})^2] z_t^{(n-1)1} \\ &= \delta_s^{n1} z_t^{(n-1)1}, 1 < n < N\end{aligned} \tag{S11}$$

and

$$\frac{\partial f(\mathbf{x}_i,\mathbf{x}_j)}{\partial b_s^{n}} = \delta_s^{n1}, 1 \le n < N \tag{S12}$$

The derivative of $f(\mathbf{x}_i, \mathbf{x}_j)$ w.r.t $\mathbf{W}^{11}$ and $\mathbf{W}^{12}$ are

$$\frac{\partial f(\mathbf{x}_i,\mathbf{x}_j)}{\partial \mathbf{W}^{11}} = \sum_k \frac{\partial f(\mathbf{x}_i,\mathbf{x}_j)}{\partial z_k^{11}} \frac{\partial z_k^{11}}{\partial \mathbf{W}^{11}} = \boldsymbol{\delta}^{11}(\mathbf{x}_i)^T \tag{S13}$$

and

$$\frac{\partial f(\mathbf{x}_i,\mathbf{x}_j)}{\partial \mathbf{W}^{12}} = \boldsymbol{\delta}^{11}(\mathbf{x}_j)^T \tag{S14}$$

Finally, the derivative of $f(\mathbf{x}_i, \mathbf{x}_j)$ w.r.t $\mathbf{x}_i$ is

$$\frac{\partial f(\mathbf{x}_i,\mathbf{x}_j)}{\partial \mathbf{x}_i} = \sum_k \frac{\partial f(\mathbf{x}_i,\mathbf{x}_j)}{\partial \mathbf{z}_k^{11}} \frac{\partial \mathbf{z}_k^{11}}{\partial \mathbf{x}_i} = (\mathbf{W}^{11})^T\boldsymbol{\delta}^{11} \tag{S15}$$

and the derivative of $f(\mathbf{x}_i, \mathbf{x}_j)$ w.r.t $\mathbf{x}_j$ is

$$\frac{\partial f(\mathbf{x}_i,\mathbf{x}_j)}{\partial \mathbf{x}_j} = (\mathbf{W}^{12})^T\boldsymbol{\delta}^{11} \tag{S16}$$

Similarly, for $f(\mathbf{x}_i, \mathbf{x}_k) = \tanh((\mathbf{w}^{N+1})^T\mathbf{z}^{N2} + \mathbf{b}^{N+1})$, we define δ_s^{n2} as

$$\delta_s^{n2} = \frac{\partial f(\mathbf{x}_i,\mathbf{x}_k)}{\partial z_s^{n2}}[1-(z_s^{n2})^2] \tag{S17}$$

With the same procedure as $f(\mathbf{x}_i, \mathbf{x}_j)$, we can get the derivatives of $f(\mathbf{x}_i, \mathbf{x}_k)$ w.r.t the parameters. We omit the details here and just

With these derivatives, it's easy to get the derivatives of the objective in Eq (3.3) w.r.t to the parameters. We denote the objective as $\mathcal{L}(\mathbf{X}, \mathbf{x}_0, \theta)$. In each iteration, the parameters are updated using gradient descent. Taking $\mathbf{x}_0$ as an example, the update rule is given as

$$\mathbf{x}_0 \leftarrow \mathbf{x}_0 - \gamma\frac{\partial \mathcal{L}(\mathbf{X},\mathbf{x}_0,\theta)}{\partial \mathbf{x}_0} \tag{S18}$$

where γ is the learning rate.

S2 Summary of Derivatives

In this section, we summarize the derivatives. For $f(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\mathbf{w}^T\mathbf{z}^{N1} + b)$, we have

$$\delta_s^{n1} = \begin{cases} [1-(z_s^{n1})^2]\sum_k \delta_k^{(n+1)1}\mathbf{W}_{ks}^{(n+1)}, & 1 \le n < N \\ w_s[1-(z_s^{N1})^2][1-f^2(\mathbf{x}_i,\mathbf{x}_j)], & n = N\end{cases} \tag{S19}$$

and the derivatives of $f(\mathbf{x}_i, \mathbf{x}_j)$ w.r.t θ are given as

$$\begin{aligned}\frac{\partial f(\mathbf{x}_i,\mathbf{x}_j)}{\partial \mathbf{w}} &= [1-f^2(\mathbf{x}_i,\mathbf{x}_j)]\mathbf{z}^{N1} \\ \frac{\partial f(\mathbf{x}_i,\mathbf{x}_j)}{\partial \mathbf{W}_{st}^{n}} &= \delta_s^{n1} z_t^{(n-1)1}, 1 < n < N \\ \frac{\partial f(\mathbf{x}_i,\mathbf{x}_j)}{\partial \mathbf{W}^{11}} &= \boldsymbol{\delta}^{11}(\mathbf{x}_i)^T \\ \frac{\partial f(\mathbf{x}_i,\mathbf{x}_j)}{\partial \mathbf{W}^{12}} &= \boldsymbol{\delta}^{11}(\mathbf{x}_j)^T \\ \frac{\partial f(\mathbf{x}_i,\mathbf{x}_j)}{\partial b} &= 1-f^2(\mathbf{x}_i,\mathbf{x}_j) \\ \frac{\partial f(\mathbf{x}_i,\mathbf{x}_j)}{\partial b_s^{n}} &= \delta_s^{n1}, 1 \le n < N\end{aligned} \tag{S20}$$

the derivatives of $f(\mathbf{x}_i, \mathbf{x}_j)$ w.r.t $\mathbf{x}_i, \mathbf{x}_j$ are given as

$$\frac{\partial f(\mathbf{x}_i,\mathbf{x}_j)}{\partial \mathbf{x}_i} = (\mathbf{W}^{11})^T\boldsymbol{\delta}^{11} \frac{\partial f(\mathbf{x}_i,\mathbf{x}_j)}{\partial \mathbf{x}_j} = (\mathbf{W}^{12})^T\boldsymbol{\delta}^{11} \tag{S21}$$

For $f(\mathbf{x}_i, \mathbf{x}_k) = \tanh(\mathbf{w}^T\mathbf{z}^{N2} + b)$, we have

$$\delta_s^{n2} = \begin{cases} [1-(z_s^{n2})^2]\sum_k \delta_k^{(n+1)2}\mathbf{W}_{ks}^{(n+1)}, & 1 \le n < N \\ w_s[1-(z_s^{N2})^2][1-f^2(\mathbf{x}_i,\mathbf{x}_k)], & n = N\end{cases} \tag{S22}$$

and the derivatives of $f(\mathbf{x}_i, \mathbf{x}_k)$ w.r.t θ are given as

$$
\begin{aligned}
\frac{\partial f(\mathbf{x}_i, \mathbf{x}_k)}{\partial \mathbf{w}} &= [1 - f^2(\mathbf{x}_i, \mathbf{x}_k)]\mathbf{z}^{N2} \\
\frac{\partial f(\mathbf{x}_i, \mathbf{x}_k)}{\partial \mathbf{W}^n_{st}} &= \delta^{n2}_s z^{(n-1)2}_t, 1 < n < N \\
\frac{\partial f(\mathbf{x}_i, \mathbf{x}_k)}{\partial \mathbf{W}^{11}} &= \boldsymbol{\delta}^{12}(\mathbf{x}_i)^T \\
\frac{\partial f(\mathbf{x}_i, \mathbf{x}_k)}{\partial \mathbf{W}^{12}} &= \boldsymbol{\delta}^{12}(\mathbf{x}_j)^T \\
\frac{\partial f(\mathbf{x}_i, \mathbf{x}_k)}{\partial b} &= 1 - f^2(\mathbf{x}_i, \mathbf{x}_k) \\
\frac{\partial f(\mathbf{x}_i, \mathbf{x}_k)}{\partial b^n_s} &= \delta^{n2}_s, 1 \leq n < N
\end{aligned}
\tag{S23}
$$

the derivatives of $f(\mathbf{x}_i, \mathbf{x}_k)$ w.r.t $\mathbf{x}_i, \mathbf{x}_k$ are given as

$$
\frac{\partial f(\mathbf{x}_i, \mathbf{x}_k)}{\partial \mathbf{x}_i} = (\mathbf{W}^{11})^T \boldsymbol{\delta}^{12} \quad \text{and} \quad \frac{\partial f(\mathbf{x}_i, \mathbf{x}_k)}{\partial \mathbf{x}_k} = (\mathbf{W}^{12})^T \boldsymbol{\delta}^{12}
\tag{S24}
$$

References

[1] Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.