Machine Learning for Graphs and Sequential Data Exercise Sheet 08

Neural Network Approaches for Sequential Data

Problem 1: Word2vec defines a mapping from a single word to a single fixed vector. Explain and provide an example why this will not be expressive enough regarding homographs (i.e., words with the same spelling but having more than one meaning). Propose an alternative solution.

Problem 2: Given a previous hidden state $\boldsymbol{h}^{(t-1)} \in \mathbb{R}^D$ and a current input $\boldsymbol{x}^{(t)} \in \mathbb{R}^N$, the recurrent neural network equations to update the hidden state and produce the output are:

$$\begin{aligned}
\boldsymbol{z}^{(t)} &= \boldsymbol{W}\boldsymbol{h}^{(t-1)} + \boldsymbol{U}\boldsymbol{x}^{(t)} \\
\boldsymbol{h}^{(t)} &= \tanh\left(\boldsymbol{z}^{(t)}\right) \\
\boldsymbol{o}^{(t)} &= \boldsymbol{V}\boldsymbol{h}^{(t)} \\
\hat{\boldsymbol{y}}^{(t)} &= \operatorname{softmax}\left(\boldsymbol{o}^{(t)}\right)
\end{aligned}$$

where parameters $\boldsymbol{W} \in \mathbb{R}^{D\times D}$, $\boldsymbol{U} \in \mathbb{R}^{D\times N}$ and $\boldsymbol{V} \in \mathbb{R}^{M\times D}$ are shared at every step.

To train an RNN we need gradients of loss w.r.t. the parameters: $\partial L/\partial \boldsymbol{W}$, $\partial L/\partial \boldsymbol{U}$ and $\partial L/\partial \boldsymbol{V}$. Your task is to arrive at the equations given on slide 17 in the lecture.

Use the fact that $\partial L/\partial \boldsymbol{o}^{(t)} = \hat{\boldsymbol{y}}^{(t)} - \boldsymbol{y}^{(t)}$, where $\boldsymbol{y}^{(t)}$ is the true output.

Hint: Since parameters are shared, the total gradient is the sum of the contributions over all the steps. Because of that, it might be easier to introduce copies of parameters, e.g. $\boldsymbol{W}^{(t)}$ *– a copy of* $\boldsymbol{W}$ *at step* t*, calculate* $\partial L/\partial \boldsymbol{W}^{(t)}$ *and sum over all* t*.*

Problem 3: What do you need to change in the equations that you got in the previous exercise if the output $\boldsymbol{o}^{(t)}$ is used as an input to another neural network?

Instead of $\frac{\partial L}{\partial o^{(t)}} = \hat{y}^{(t)} - y^{(t)}$ calculate $\frac{\partial L}{\partial o^{(t)}}$ based on next network

Problem 1: Word2vec defines a mapping from a single word to a single fixed vector. Explain and provide an example why this will not be expressive enough regarding homographs (i.e., words with the same spelling but having more than one meaning). Propose an alternative solution.

Word2vec will use same vector to represent the same word even it has different meanings ⇒ find same neighboring words

RNN, LSTM, Transformer ⇒ capture different sentence pattern for word has differ meanings

Problem 2: Given a previous hidden state $\boldsymbol{h}^{(t-1)} \in \mathbb{R}^D$ and a current input $\boldsymbol{x}^{(t)} \in \mathbb{R}^N$, the recurrent neural network equations to update the hidden state and produce the output are:

$$\begin{aligned}
\boldsymbol{z}^{(t)} &= \boldsymbol{W}\boldsymbol{h}^{(t-1)} + \boldsymbol{U}\boldsymbol{x}^{(t)} \\
\boldsymbol{h}^{(t)} &= \tanh\left(\boldsymbol{z}^{(t)}\right) \\
\boldsymbol{o}^{(t)} &= \boldsymbol{V}\boldsymbol{h}^{(t)} \\
\hat{\boldsymbol{y}}^{(t)} &= \mathrm{softmax}\left(\boldsymbol{o}^{(t)}\right)
\end{aligned}$$

$$\frac{\partial h_i^{(t)}}{\partial z_i^{(t)}} = 1 - \tanh\left(z_i^{(t)}\right)^2$$

where parameters $\boldsymbol{W} \in \mathbb{R}^{D\times D}$, $\boldsymbol{U} \in \mathbb{R}^{D\times N}$ and $\boldsymbol{V} \in \mathbb{R}^{M\times D}$ are shared at every step.

To train an RNN we need gradients of loss w.r.t. the parameters: $\partial L/\partial \boldsymbol{W}$, $\partial L/\partial \boldsymbol{U}$ and $\partial L/\partial \boldsymbol{V}$. Your task is to arrive at the equations given on slide 17 in the lecture.

Use the fact that $\partial L/\partial \boldsymbol{o}^{(t)} = \hat{\boldsymbol{y}}^{(t)} - \boldsymbol{y}^{(t)}$, where $\boldsymbol{y}^{(t)}$ is the true output.

Hint: Since parameters are shared, the total gradient is the sum of the contributions over all the steps. Because of that, it might be easier to introduce copies of parameters, e.g. $\boldsymbol{W}^{(t)}$ – a copy of $\boldsymbol{W}$ at step t, calculate $\partial L/\partial \boldsymbol{W}^{(t)}$ and sum over all t.

$$\frac{\partial L}{\partial V} = \sum_t \frac{\partial L}{\partial V^{(t)}} = \sum_t \frac{\partial L}{\partial o^{(t)}} \cdot \frac{\partial o^{(t)}}{\partial V} = \sum_t (\hat{y}^{(t)} - y^{(t)}) \cdot h^{(t)T}$$

$$\frac{\partial L}{\partial W} = \sum_t \frac{\partial L}{\partial W^{(t)}} = \sum \frac{\partial L}{\partial o^{(t)}} \cdot \frac{\partial o^{(t)}}{\partial h^{(t)}} \cdot \underbrace{\frac{\partial \tanh(z^{(t)})}{\partial z^{(t)}}}_{\mathrm{diag}(1-(h_i^{(t)})^2)} \cdot \underbrace{\frac{\partial Wh^{(t-1)} + Ux^{(t)}}{\partial W}}_{(h^{(t-1)})^T}$$

$$= \sum \underbrace{(\hat{y}^{(t)} - y^{(t)})\, V}_{} \,.$$

$$= \frac{\partial L}{\partial h^{(t)}}$$