

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 $\mathbf{h}^{(t-1)} \in \mathbb{R}^D$ and a current input $\mathbf{x}^{(t)} \in \mathbb{R}^N$, the equations to update the hidden state and produce the output are

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

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

To train an RNN we need partial derivatives of loss w.r.t. the parameters: $\partial L / \partial \mathbf{W}$, $\partial L / \partial \mathbf{U}$ and $\partial L / \partial \mathbf{V}$. Prove the formula given on slide 17 for these derivatives.

Use the fact that $\partial L / \partial \mathbf{o}^{(t)} = \hat{\mathbf{y}}^{(t)} - \mathbf{y}^{(t)}$, where $\mathbf{y}^{(t)}$ 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. $\mathbf{W}^{(t)}$ – a copy of \mathbf{W} at step t , calculate $\partial L / \partial \mathbf{W}^{(t)}$ 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 $\mathbf{o}^{(t)}$ is used as an input to another neural network?
