Very Deep Learning | **Technische Universität Kaiserslautern**
WS 2022/23 | Dr. Muhammad Zeshan Afzal

Exercise 3 - Recurrent Networks and NLP

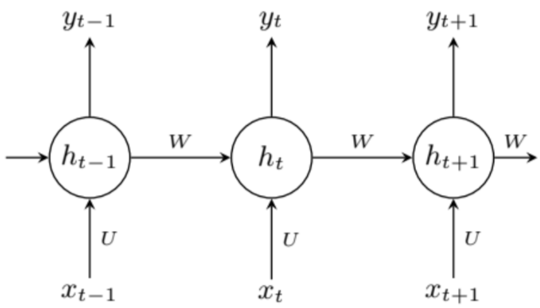
Deadline: 12.12.2022 **Total Marks: 30**

Submission

- Submissions through OLAT. Only one group member needs to submit it.
- Your submission should contain a PDF with the solutions to the exercise questions (and any Python code files) zipped together in a single file.
- Include the group number along with the names and matriculation numbers of all group members on the PDF.
- For Jupyter notebooks, please save them with the outputs of your code displayed.
- Question 2.6 is not mandatory. Bonus points can be used to make up for lost points in other exercises as well.

3.1. Backpropagation through Time [4+4=8]

Consider the following RNN:



Each state h_t is given by:

$$h_t = \sigma(Wh_{t-1} + Ux_t), \quad \sigma(z) = \frac{1}{1+e^{-z}}$$

Let L be a loss function defined as the sum over the losses L_t at every time step until time $T : L = \sum_{t=0}^{T} L_t$, where L_t is a scalar loss depending on h_t.

In the following, we want to derive the gradient of this loss function with respect to the parameter W.

i) Given $y = \sigma(Wx)$ where $y \in \mathbb{R}^n, x \in \mathbb{R}^d$ and $W \in \mathbb{R}^{n \times d}$. Derive the Jacobian $\frac{\partial y}{\partial x} = diag(\sigma')W \in \mathbb{R}^{n \times d}$.

ii) Derive the quantity $\frac{\partial L}{\partial W} = \sum_{t=0}^{T} \sum_{k=1}^{t} \frac{\partial L_t}{\partial h_t} \frac{\partial h_t}{\partial h_k} \frac{\partial h_k}{\partial W}$.

3.2. Vanishing/Exploding Gradients in RNNs [2+3=5]

In this exercise, we want to understand why RNNs are especially prone to the Vannishing/Exploding Gradients problem and what role the eigenvalues of the weight matrix play. Consider part 3.1(ii) again.

i) Write down $\frac{\partial L}{\partial W}$ as expanded sum for $T = 3$. You should see that if we want to backpropagate through n timesteps, we have to multiply the matrix $diag(\sigma')W$ n times with itself.

ii) Consider the weight matrix $W = \begin{pmatrix} 0.58 & 0.24 \\ 0.24 & 0.72 \end{pmatrix}$. Its eigendecomposition is:

$$W = Q\Lambda Q^{-1} = \begin{pmatrix} 0.8 & -0.6 \\ 0.6 & 0.8 \end{pmatrix} \begin{pmatrix} 0.4 & 0 \\ 0 & 0.9 \end{pmatrix} \begin{pmatrix} 0.8 & 0.6 \\ -0.6 & 0.8 \end{pmatrix}$$

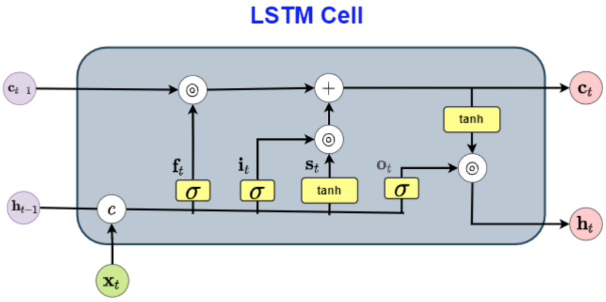
Calculate W^{30}. What do you observe? What happens in general if the absolute value of all eigenvalues of W is smaller than 1? What happens if the absolute value of any eigenvalue of W is larger than 1? What if all eigenvalues are 1?

3.3. LSTMs [3+4=7]

In this exercise we try to implement LSTM cell

i) What do the gates f_t, i_t and o_t do?

ii) Implement LSTM cell using pytorch tensors. Please create a new cell in the notebook of task 3.5 and write your code there.



$$\begin{aligned}
\mathbf{f}_t &= \sigma(\mathbf{W}_{fh}\,\mathbf{h}_{t-1} + \mathbf{W}_{fx}\,\mathbf{x}_t + \mathbf{b}_f)\\
\mathbf{i}_t &= \sigma(\mathbf{W}_{ih}\,\mathbf{h}_{t-1} + \mathbf{W}_{ix}\,\mathbf{x}_t + \mathbf{b}_i)\\
\mathbf{o}_t &= \sigma(\mathbf{W}_{oh}\,\mathbf{h}_{t-1} + \mathbf{W}_{ox}\,\mathbf{x}_t + \mathbf{b}_o)\\
\mathbf{s}_t &= \tanh(\mathbf{W}_{sh}\,\mathbf{h}_{t-1} + \mathbf{W}_{sx}\,\mathbf{x}_t + \mathbf{b}_s)\\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{s}_t\\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t)
\end{aligned}$$

3.4. Visualizing Neural Networks [5]

In this exercise, you will learn how to visualize the hidden layers of convolutional neural networks. Follow the instructions in the `Task3.4_Visualization.ipynb` notebook and complete all tasks marked as `TODO`.

3.5. NLP with RNNs [5]

In this exercise, you will use recurrent neural networks to perform sentiment analysis on Amazon's review dataset. Follow the instructions in the `Task3.5_NLP.ipynb` notebook and complete all tasks marked as `TODO`.