Assignment 2. Recurrent Neural Networks and Graph Neural Networks

Maurice Frank
11650656
maurice.frank@posteo.de
Code: github

1 Vanilla RNN versus LSTM

1.1 RNN derivatives

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}_{ph}} = \prod \frac{\partial}{\partial} \cdot \frac{\partial}{\partial \boldsymbol{W}_{ph}} \tag{1}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}_{hh}} = \dots \tag{2}$$

1.2 Vanilla RNN code

Find the code inside `vanilla_rnn.py` and `train.py`.

1.3 Vanilla RNN experiment

See Figure 1.6 for a overview plot of the results and Section 1.6 for a discussion/comparsion of the results.

1.4 Optimizers

1.5 LSTM theory

a) LSTM Gates

input modulation gate $\boldsymbol{g}^{(t)}$ The input modulation gate determines candidate information from the new input (using also the old hidden state). We want our state values normalized but need also negative values (otherwise the cell values would only increase) which, as in this case, can be done with a $\tanh$, squashing the input to $[-1, 1]$.

input gate $\boldsymbol{i}^{(t)}$ The input regulates which and how much information of the input of this time step should be included in the cell and hidden state. As the input gate regulates the flow it is necessary to have its values bounded to $[0, 1]$ which can most directly achieved by squashing the values with the sigmoid.

forget gate $\boldsymbol{f}^{(t)}$ The forget gate regulates which and how much information from the old cell state should be disregared under the new information from the input (and the old hidden state). As the forget gate only changes the importance (magnitude) of the information in the cell state it should be in $[0, 1]$ which is achieved with the sigmoid.

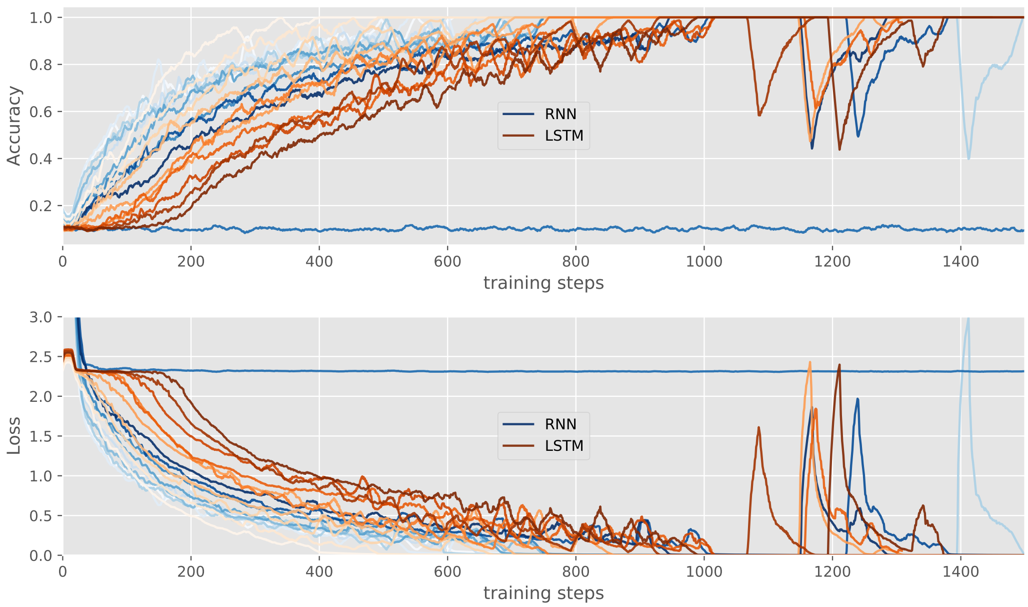

Figure 1

output gate $\boldsymbol{o}^{(t)}$ The output gate regulates which and how much information from the new cell state should go into the new hidden state. Again its gating the values from the other tensor which is asking for a range $[0, 1]$ achieved by the sigmoid.

b) Number of parameters

We have given $\boldsymbol{x} \in \mathbb{R}^{T \times d}$ with T sequence length and d feature dimension. Further we have n hidden units. Then we have

$$4 \cdot (d \cdot n + n \cdot n + n)$$

trainable parameters for one LSTM cell.

1.6 LSTM practice

Find the code inside `lstm.py` and `train.py`.

2 Recurrent Nets as Gernerative Model

2.1

a)

b)

c)

2.2

3 Graph Neural Networks

3.1

a)

b)

3.2

3.3

a)

b)