Deep Learning for Speech and Language Processing
Exercise Sheet 6: RNNs, LSTMs

Maximilian Schmidt, Pascal Tilli, Dirk Väth, Ngoc Thang Vu

December 14th, 2020

Deadline

Please submit your solutions until Monday, December 21st, 2020, noon. You may also see the deadline at any time on the website under *Exercises* → *Upload.*

Notation

Notation conventions for this exercise:

Symbol	Description
$[x, y]$	concatenation of two vectors: $z = [x, y]$ for $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \mathbb{R}^2, y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \in \mathbb{R}^3 \rightarrow z = \begin{bmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} \in \mathbb{R}^{2+3}$
x_t	input *vector* at timestep/position t in a sequence of vectors (e.g. RNN input/output)
x_{t_i}	i-th entry of a vector (= scalar) at timestep t of a sequence
$\odot$	element-wise product of two vectors

Submission

Calculation Tasks

Upload (only solutions!) to https://dlcourse.ims.uni-stuttgart.de/ using your account.

Theory Tasks

You don't have to submit this part, but it will help you with the contents of this course.

Warm-Up

Exercise 1.

(1) **Calculation Task** Compute the number of parameters with input layer size 100, hidden layer size 400 and output layer size 5 for

  (1) an Elman RNN.
  (2) a Jordan RNN.

  Always include the bias!

(2) **Theory Task** In an Elman network, the output of the hidden layer at timestep t a_t is stored in the memory and applied to compute the output of the hidden layer in the next timestep $t + 1$. However, in a Jordan network, the *network's output* y_t is stored in the memory and used to compute the output of the hidden layer

at $t+1$. Give the equations to compute a_t for both network types. What is the consequence on the hidden and output layer sizes of Elman vs. Jordan networks?

(3) **Theory Task** The computations in an LSTM cell can be defined by the following equations:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \; (\sigma = \text{logistic sigmoid}) \tag{1}$$
$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{2}$$
$$\tilde{C}_t = \tanh(W_C[h_{t-1}, x_t] + b_C) \tag{3}$$
$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{4}$$
$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{5}$$
$$h_t = o_t \odot \tanh(C_t) \tag{6}$$

In what range are the entries of the input, forget and output gate? In what range are the entries of the candidate activation $\tilde{C}_t$? And in what range are the outputs of the LSTM cell?

Theory Task: RNN

In this exercise we compute the output of an Elman RNN for POS tagging, which is depicted in Figure 1. Use the logistic sigmoid as the activation function σ.

Exercise 2.

Compute the sequence of POS tags for the input sentence 'Kevin likes bananas'. Represent each word as one-hot vector given the 3-word vocabulary V = $\{v_1$: Kevin, v_2 : likes, v_3 : bananas$\}$.

The outputs y_t of the network in each timestep are 3-dimensional vectors $\begin{bmatrix} \text{p(VBZ)} \\ \text{p(NNS)} \\ \text{p(NNP)} \end{bmatrix}$ denoting the probability of the following POS tags according to the Penn Treebank Tagset: verb in 3rd person singular present, plural noun, proper noun singular. The memory (a_0) is initialized as zero vector. Use the following weights of the network, there are no biases in any layer.

$W_\text{i} = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 0 & 0.5 \end{bmatrix}, W_\text{h} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, W_{out} = \begin{bmatrix} -1 & 1 \\ 0 & -1 \\ 1 & 0 \end{bmatrix}.$

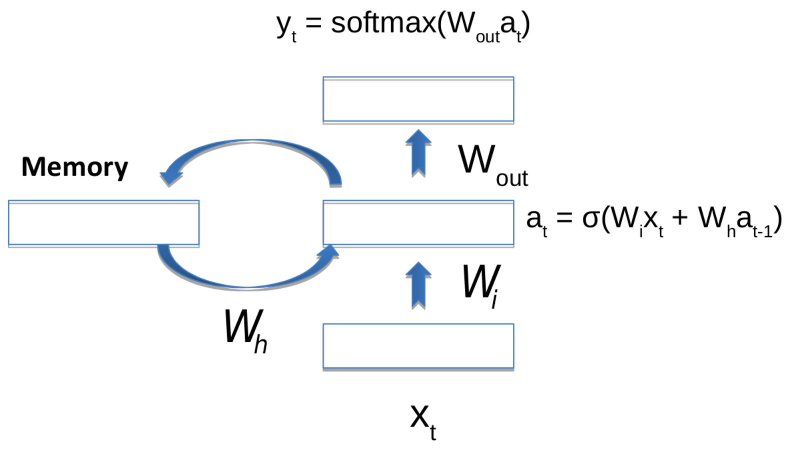

Figure 1: RNN architecture for Exercise 2.

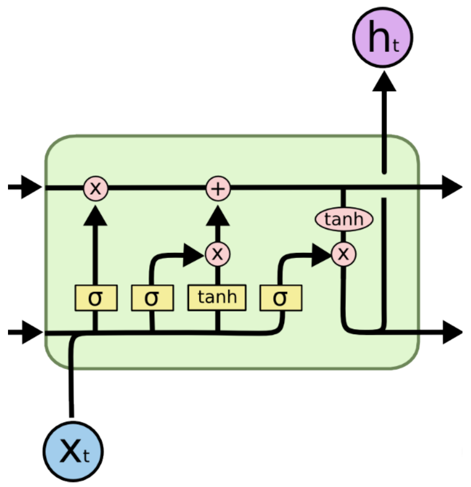

Figure 2: LSTM cell
(`http://colah.github.io/posts/2015-08-Understanding-LSTMs`).

Calculation Task: LSTM

To have more control over the memory updates in RNNs, we can replace the simple RNN cell, in the RNN architecture of Exercise 2 with an LSTM cell as depicted in Figure 2. By doing so the memory vectors a_t become tuples of a hidden state and cell state (h_t, C_t), which are computed according to Equation 1-6 given in Exercise 1.3. Note that in LSTMs only the hidden state of the memory tuple is passed on to the output layer.

Exercise 3.

Determine the predicted POS tag for the second word of the input sentence from Exercise 2. Assume as hidden state after processing the first word $h_1 = \begin{bmatrix} 0.370 \\ -0.123 \end{bmatrix}, C_1 = \begin{bmatrix} 0.557 \\ -0.338 \end{bmatrix}$. The weight matrices of the LSTM cell and for the output are given below, all bias vectors are zero vectors.

$$W_f = \begin{bmatrix} 0 & -1 & -1 & 1 & 0 \\ -1 & 1 & 0 & -1 & 1 \end{bmatrix}, W_i = \begin{bmatrix} -1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & -1 \end{bmatrix}, W_C = \begin{bmatrix} -1 & 0 & 1 & 1 & 0 \\ -1 & -1 & 1 & -1 & 1 \end{bmatrix}$$

$$W_o = \begin{bmatrix} -1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & -1 & 1 \end{bmatrix}, W_{out} = \begin{bmatrix} -1 & 1 \\ 0 & -1 \\ 0 & 0 \end{bmatrix}$$