Advanced Human Language Technologies
Exercises on Recurrent Neural Networks

Recurrent Neural Networks

Exercise 1.

Your friend's mood depends on the weather of the last few days. You've collected data about the weather for the past 365 days which you represent as a sequence as $x_1, \ldots, x_{365}$. You've also collected data on your friends's mood, which you represent as $y_1, \ldots, y_{365}$. You'd like to build a model to map from $x \rightarrow y$.

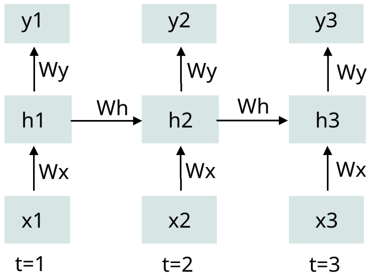
1. Should you use a Unidirectional RNN or Bidirectional RNN for this problem?
2. Explain why vanishing gradients affect more severerly RNNs than deep FNNs.

SOLUTION

1. You need a unidirectional RNN because your friend's mood does only depend on the past days weather. Having the information from the future weather would not be useful.
2. Vanishing gradients affect much more RNNs because the weight matrices are shared among all time steps, while in FNN each input position (i.e. time step) has it own matrix.

Exercise 2.

Given the unfolding of the first time steps of a RNN, where both hidden and output layers are linear:



1. Write the RNN equations.
2. Compute the values for hidden units h_1, h_2, h_3 and output units y_1, y_2, y_3, for input values $x_1 = 2, x_2 = -0.5, x_3 = 1$, assuming that the weight matrices are $W_x = W_h = W_y = 1$.
3. Explain what is the network computing.

SOLUTION

1. The RNN equations are:
   $h(t) = W_x x_t + W_h h_{t-1}$
   $y(t) = W_y h_t$

2. $h_1 = W_x x_1 + W_h h_0 = 1 \cdot 2 + 1 \cdot 0 = 2$
   $y_1 = W_y h_1 = 1 \cdot 2 = 2$

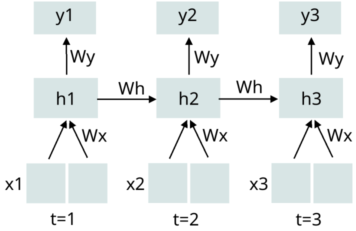
   $h_2 = W_x x_2 + W_h h_1 = 1 \cdot -0.5 + 1 \cdot 1 = 1.5$
   $y_2 = W_y h_2 = 1 \cdot 1.5 = 1.5$

   $h_3 = W_x x_3 + W_h h_2 = 1 \cdot 1 + 1 \cdot 1.5 = 2.5$
   $y_3 = W_y h_3 = 1 \cdot 2.5 = 2.5$

3. The network computes the sum of all the inputs, i.e. $y_t = \sum_{i=1}^{t} x_i$

Exercise 3.

Given the unfolding of the first time steps of a RNN, where hidden layer is linear and output layer has a sigmoid activation function $\sigma(z) = \frac{1}{1+e^{-z}}$:



1. Write the RNN equations.

2. Compute the values for hidden units h_1, h_2, h_3 and output units y_1, y_2, y_3, for input values $x_1 = (2, -2), x_2 = (0, 3.5), x_3 = (1, 2.2)$, assuming that the weight matrices are $W_x = (1, -1), W_h = W_y = 1$.

3. Explain what is the network computing.

SOLUTION

1. The RNN equations are:
   $h(t) = W_x x_t + W_h h_{t-1}$
   $y(t) = \sigma(W_y h_t) = \dfrac{1}{1 + e^{-W_y h_t}}$

2. $h_1 = W_x x_1 + W_h h_0 = (1, -1) \cdot (2, -2) + 1 \cdot 0 = 4$
   $y_1 = \sigma(W_y h_1) = \sigma(1 \cdot 4) = 1/(1 + e^{-4}) = 0.98$

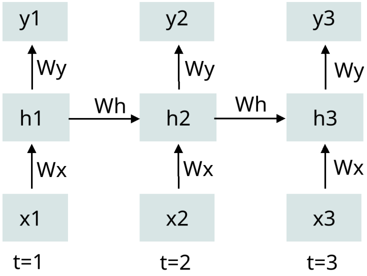
   $h_2 = W_x x_2 + W_h h_1 = (1, -1) \cdot (0, 3.5) + 1 \cdot 4 = 0.5$
   $y_2 = \sigma(W_y h_2) = \sigma(1 \cdot 0.5) = 1/(1 + e^{-0.5}) = 0.62$

   $h_3 = W_x x_3 + W_h h_2 = (1, -1) \cdot (1, 2.2) + 1 \cdot 0.5 = -0.7$
   $y_3 = \sigma(W_y h_3) = \sigma(1 \cdot -0.7) = 1/(1 + e^{0.7}) = 0.33$

3. The network computes in the hidden layer the sum over all time steps of the difference between both x_i dimensions, i.e. $h_t = \sum_{i=1}^{t}(x_i[0] - x_i[1])$.
   The output layer is the sigmoid function of that value, that is, a value in $[0, 1]$ that is 0.5 when the difference is zero, -1 when it is largely negative and $+1$ if it is largely positive.

Exercise 4.

Given the unfolding of the first time steps of a RNN, where hidden layer has a sigmoid activation function $\sigma(z) = \frac{1}{1+e^{-z}}$, and output layer is linear.



1. Write the RNN equations.
2. Compute the values for hidden units h_1, h_2, h_3 and output units y_1, y_2, y_3, for input values $x_1 = 18, x_2 = 9, x_3 = -8$, assuming that the weight matrices are $W_x = -0.1, W_h = 0.5, W_y = 0.25$ and biases are $b_h = 0.4, b_y = 0$.

SOLUTION

1. The RNN equations are:
   $h(t) = \sigma(W_x x_t + W_h h_{t-1} + b_h) = \dfrac{1}{1 + e^{-(W_x x_t + W_h h_{t-1} + b_h)}}$
   $y(t) = W_y h_t + b_y$
2. $h_1 = \sigma(W_x x_1 + W_h h_0 + b_h) = \sigma(-0.1 \cdot 18 + 0 + 0.4) = \sigma(-1.4) = 1/(1 + e^{1.4}) = 0.2$
   $y_1 = W_y h_1 + b_y = 0.25 * 0.2 + 0 = 0.05$

   $h_2 = \sigma(W_x x_2 + W_h h_1 + b_h) = \sigma(-0.1 \cdot 9 + 0.5 \cdot 0.2 + 0.4) = \sigma(-0.4) = 1/(1 + e^{0.4}) = 0.4$
   $y_2 = W_y h_2 + b_y = 0.25 * 0.4 + 0 = 0.1$

   $h_3 = \sigma(W_x x_3 + W_h h_2 + b_h) = \sigma(-0.1 \cdot -8 + 0.5 \cdot 0.4 + 0.4) = \sigma(1.4) = 1/(1 + e^{-1.4}) = 0.8$
   $y_3 = W_y h_3 + b_y = 0.25 * 0.8 + 0 = 0.2$

Exercise 5.

Consider a simple RNN with the following equations:

$$h_t = \tanh(W_x x_t + W_h h_{t-1} + b_h)$$

$$y_t = W_y h_t + b_y$$

where:

- x_t is the input at time t,
- h_t is the hidden state at time t, set to zero for $t = 0$
- W_h, W_x, W_y are weight matrices, and
- b_h, b_y are biases.

1. Given the following values, compute the hidden states h_1 and h_2 for the input sequence $x_1 = 1$, $x_2 = 2$

$$W_x = \begin{bmatrix} 0.6 \\ 0.1 \end{bmatrix} \qquad W_h = \begin{bmatrix} 0.5 & 0.2 \\ 0.3 & 0.7 \end{bmatrix} \qquad b_h = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad b_y = 0.1$$

2. Given the values above, which is the dimensionality of h_t, y_t, and W_y in this RNN? Justify your answer.

SOLUTION

1. Computing h_1 and h_2

   For $t = 1$:

$$h_1 = \tanh(W_x x_1 + W_h h_0 + b_h) = \tanh\left(\begin{bmatrix}0.6\\0.1\end{bmatrix} \cdot 1 + \begin{bmatrix}0.5 & 0.2\\0.3 & 0.7\end{bmatrix} \cdot \begin{bmatrix}0\\0\end{bmatrix} + \begin{bmatrix}0\\0\end{bmatrix}\right)$$

$$= \tanh\left(\begin{bmatrix}0.6\\0.1\end{bmatrix}\right) = \begin{bmatrix}\tanh(0.6)\\\tanh(0.1)\end{bmatrix} \approx \begin{bmatrix}0.5370\\0.0997\end{bmatrix}$$

   For $t = 2$:

$$h_2 = \tanh(W_x x_2 + W_h h_1 + b_h) = \tanh\left(\begin{bmatrix}0.6\\0.1\end{bmatrix} \cdot 2 + \begin{bmatrix}0.5 & 0.2\\0.3 & 0.7\end{bmatrix} \cdot \begin{bmatrix}0.5370\\0.0997\end{bmatrix} + \begin{bmatrix}0\\0\end{bmatrix}\right)$$

$$= \tanh\left(\begin{bmatrix}1.2\\0.2\end{bmatrix} + \begin{bmatrix}(0.5 \times 0.5370) + (0.2 \times 0.0997)\\(0.3 \times 0.5370) + (0.7 \times 0.0997)\end{bmatrix}\right) = \tanh\left(\begin{bmatrix}1.2 + (0.2685 + 0.01994)\\0.2 + (0.1611 + 0.06979)\end{bmatrix}\right)$$

$$= \tanh\left(\begin{bmatrix}1.4884\\0.4309\end{bmatrix}\right) = \begin{bmatrix}\tanh(1.4884)\\\tanh(0.4309)\end{bmatrix} \approx \begin{bmatrix}0.9032\\0.4063\end{bmatrix}$$

2. Dimensionality of h_t, y_t, and W_y

   The hidden state h_t is a 2-dimensional vector since W_x has two rows, and W_h is a 2×2 matrix.

   The output y_t depends on W_y. Since $W_y h_t$ must result in a scalar (as implied by the scalar bias b_y), W_y must be a 1×2 matrix.

   Therefore, the dimensions are:

   - $h_t \in \mathbb{R}^2$
   - $y_t \in \mathbb{R}$
   - $W_y \in \mathbb{R}^{1\times 2}$

Exercise 6.

Given an LSTM cell with the following equations:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)$$
$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$$
$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)$$
$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c)$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$$
$$h_t = o_t \odot \tanh(c_t)$$

where:
$\sigma(x) = \frac{1}{1+e^{-x}}$ (sigmoid activation)
$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ (tanh activation)
$\odot$ represents element-wise multiplication.

1. Manually compute f_t, i_t, o_t, c_t, h_t, assuming the following values

   - $x_t = 1 \qquad h_{t-1} = \begin{bmatrix}0.1\\0.2\end{bmatrix} \qquad c_{t-1} = \begin{bmatrix}0.05\\0.1\end{bmatrix}$
   - Forget gate: $W_f = \begin{bmatrix}0.4\\-0.2\end{bmatrix} \qquad U_f = \begin{bmatrix}0.1 & 0.3\\0.2 & 0.5\end{bmatrix} \qquad b_f = \begin{bmatrix}0.1\\-0.1\end{bmatrix}$
   - Input gate: $W_i = \begin{bmatrix}0.2\\-0.3\end{bmatrix} \qquad U_i = \begin{bmatrix}0.4 & 0.2\\-0.1 & 0.3\end{bmatrix} \qquad b_i = \begin{bmatrix}0.05\\-0.05\end{bmatrix}$
   - Output gate: $W_o = \begin{bmatrix}-0.3\\0.5\end{bmatrix} \qquad U_o = \begin{bmatrix}0.2 & -0.2\\0.1 & 0.4\end{bmatrix} \qquad b_o = \begin{bmatrix}0.02\\0.01\end{bmatrix}$

- Candidate cell state: $W_c = \begin{bmatrix}0.3\\0.1\end{bmatrix} \qquad U_c = \begin{bmatrix}-0.2 & 0.3\\0.4 & -0.1\end{bmatrix} \qquad b_c = \begin{bmatrix}0.0\\0.1\end{bmatrix}$

Solution

1. Computing f_t, i_t, o_t, c_t, h_t

   First, compute the gate activations:

   Forget gate:

$$\begin{aligned}
f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) &= \sigma\left(\begin{bmatrix}0.4\\-0.2\end{bmatrix} \cdot 1 + \begin{bmatrix}0.1 & 0.3\\0.2 & 0.5\end{bmatrix} \cdot \begin{bmatrix}0.1\\0.2\end{bmatrix} + \begin{bmatrix}0.1\\-0.1\end{bmatrix}\right)\\
&= \sigma\left(\begin{bmatrix}0.4\\-0.2\end{bmatrix} + \begin{bmatrix}(0.1 \times 0.1) + (0.3 \times 0.2)\\(0.2 \times 0.1) + (0.5 \times 0.2)\end{bmatrix} + \begin{bmatrix}0.1\\-0.1\end{bmatrix}\right) = \sigma\left(\begin{bmatrix}0.4 + (0.01 + 0.06) + 0.1\\-0.2 + (0.02 + 0.1) - 0.1\end{bmatrix}\right)\\
&= \sigma\left(\begin{bmatrix}0.57\\-0.18\end{bmatrix}\right) \approx \begin{bmatrix}0.639\\0.455\end{bmatrix}
\end{aligned}$$

   Input gate:

$$\begin{aligned}
i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) &= \sigma(\begin{bmatrix}0.2\\-0.3\end{bmatrix} \cdot 1 + \begin{bmatrix}0.4 & 0.2\\-0.1 & 0.3\end{bmatrix} \cdot \begin{bmatrix}0.1\\0.2\end{bmatrix} + \begin{bmatrix}0.05\\-0.05\end{bmatrix})\\
&= \sigma\left(\begin{bmatrix}0.2 + 0.4 \times 0.1 + 0.2 \times 0.2 + 0.05\\-0.3 - 0.1 \times 0.1 + 0.3 \times 0.2 - 0.05\end{bmatrix}\right) = \sigma\left(\begin{bmatrix}0.33\\-0.3\end{bmatrix}\right) \approx \begin{bmatrix}0.582\\0.426\end{bmatrix}
\end{aligned}$$

   Output gate:

$$\begin{aligned}
o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) &= \sigma(\begin{bmatrix}-0.3\\0.5\end{bmatrix} \cdot 1 + \begin{bmatrix}0.2 & -0.2\\0.1 & 0.4\end{bmatrix} \cdot \begin{bmatrix}0.1\\0.2\end{bmatrix} + \begin{bmatrix}0.02\\0.01\end{bmatrix})\\
&= \sigma\left(\begin{bmatrix}-0.3 + 0.2 \times 0.1 - 0.2 \times 0.2 + 0.02\\0.5 + 0.1 \times 0.1 + 0.4 \times 0.2 + 0.01\end{bmatrix}\right) = \sigma\left(\begin{bmatrix}-0.3\\0.6\end{bmatrix}\right) \approx \begin{bmatrix}0.426\\0.646\end{bmatrix}
\end{aligned}$$

   Candidate cell state:

$$\begin{aligned}
\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) &= \tanh(\begin{bmatrix}0.3\\0.1\end{bmatrix} \cdot 1 + \begin{bmatrix}-0.2 & 0.3\\0.4 & -0.1\end{bmatrix} \cdot \begin{bmatrix}0.1\\0.2\end{bmatrix} + \begin{bmatrix}0.0\\0.1\end{bmatrix})\\
&= \tanh\left(\begin{bmatrix}0.3 - 0.2 \times 0.1 + 0.3 \times 0.2 + 0.0\\0.1 + 0.4 \times 0.1 - 0.1 \times 0.2 + 0.1\end{bmatrix}\right) = \tanh\left(\begin{bmatrix}0.34\\0.22\end{bmatrix}\right) \approx \begin{bmatrix}0.327\\0.217\end{bmatrix}
\end{aligned}$$

   Cell state:

$$\begin{aligned}
c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t &= \begin{bmatrix}0.639\\0.455\end{bmatrix} \odot \begin{bmatrix}0.05\\0.1\end{bmatrix} + \begin{bmatrix}0.582\\0.426\end{bmatrix} \odot \begin{bmatrix}0.327\\0.217\end{bmatrix}\\
&= \begin{bmatrix}0.639 \times 0.05 + 0.582 \times 0.327\\0.455 \times 0.1 + 0.426 \times 0.217\end{bmatrix} \approx \begin{bmatrix}0.222\\0.138\end{bmatrix}
\end{aligned}$$

   Hidden state:

$$h_t = o_t \odot \tanh(c_t) = \begin{bmatrix}0.426\\0.646\end{bmatrix} \odot \tanh(\begin{bmatrix}0.222\\0.138\end{bmatrix}) = \begin{bmatrix}0.426 \times \tanh(0.222)\\0.646 \times \tanh(0.138)\end{bmatrix} = \begin{bmatrix}0.426 \times 0.218\\0.646 \times 0.137\end{bmatrix} \approx \begin{bmatrix}0.093\\0.089\end{bmatrix}$$

Exercise 7.

We have an LSTM trained to perform sentiment analysis. At each step, the LSTM cell has different values for the **forget gate** (f_t) which decide what fraction of the previous cell state should be kept, the **input gate** (i_t), that controls how much new information is added, and the **output gate** (o_t), which controls what is sent to the hidden state.

Given the input sentence *I love this movie*, gate values at each time step are:

t	Word	f_t	i_t	o_t
1	I	0.9	0.1	0.4
2	love	0.1	0.9	0.8
3	this	0.7	0.3	0.5
4	movie	0.8	0.5	0.6

1. If the initial cell state is $c_0 = 1.0$, compute the final cell state c_t after processing *I love this movie* using only the forget gate values and ignoring the input gate values. Interpret the impact of these values on memory retention.

2. Now take into account the input gate information at step $t = 2$, assuming that he candidate cell state computed from the input is 0.8, and compute the new c_t. What does this tell about how LSTMs handle important words?

3. Assume the final c_t is 2.0 before the output gate is applied. Compute the final hidden state h_t. How does the output gate affect the final representation?

4. Discussion Questions

   (a) **Forget vs. Input Gates**: How do forget and input gates work together to control memory in an LSTM?

   (b) **Effect of Sentiment Words**: How does the LSTM handle sentiment words like *love*?

   (c) **Limitations**: What are some weaknesses of LSTMs compared to Transformers in sequence processing?

SOLUTION

1. The cell state update (using only the forget gate and ignoring the input gate) follows the formula:

$$c_t = f_t \cdot c_{t-1}$$

   Given $c_0 = 1.0$, we compute each step:

$$\begin{aligned} c_1 &= f_1 \cdot c_0 = 0.9 \cdot 1.0 = 0.9 \\ c_2 &= f_2 \cdot c_1 = 0.1 \cdot 0.9 = 0.09 \\ c_3 &= f_3 \cdot c_2 = 0.7 \cdot 0.09 = 0.063 \\ c_4 &= f_4 \cdot c_3 = 0.8 \cdot 0.063 = 0.0504 \end{aligned}$$

   The forget gate has a strong impact at $t = 2$ (where it drastically reduces memory), showing how an LSTM can ”forget” past information based on learned gate values.

2. The new cell state update formula is:

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t$$

   For $t = 2$, with an additional new input of 0.8 and input gate $i_2 = 0.9$:

$$c_2' = f_2 \cdot c_1 + i_2 \cdot 0.8 = 0.1 \cdot 0.9 + 0.9 \cdot 0.8 = 0.09 + 0.72 = 0.81$$

   This demonstrates that if sentiment words (like *love*) receive high input gate values, it allows LSTMs to amplify their importance.

3. The hidden state is computed using the output gate and the cell state:

$$h_t = o_t \cdot \tanh(c_t)$$

   Given $c_4 = 2.0$ and $o_4 = 0.6$:

$$h_4 = 0.6 \cdot \tanh(2.0) = 0.6 \cdot 0.964 \approx 0.578$$

4. (a) The forget and input gates jointly control memory retention. Forget gates selectively discard past information, while input gates determine how much new information is incorporated.

   (b) If the LSTM has been trained on a sentiment analysis task, sentiment words like *love* will receive high input values, ensuring their strong influence on the cell state.

   (c) LSTMs struggle with long-term dependencies due to vanishing gradients and sequential processing constraints, while Transformers are able to access any information in the past sequence thanks to the attention mechanism.