MIDTERM EXAM

Course: Natural Language Processing

Chapter 8: Recurrent Neural Networks (RNN) and LSTM

Time allowed: 60 minutes

Question	Points	Score
1	0	
2	0	
3	0	
4	0	
5	0	
6	0	
7	0	
8	0	
9	0	
10	0	
11	0	
12	0	
13	0	
14	0	
15	0	
16	0	
17	0	
18	0	
19	0	
20	0	
Total:	0	

Part I: Multiple Choice (8 questions, 4.0 points)

1. What key feature of a Recurrent Neural Network (RNN) makes it suitable for processing sequential data?

   A. It uses convolutional layers.

Name:..... Class:.....

B. Weight sharing across time steps and having a hidden state.

C. Each input is processed independently of the others.

D. There are no loops in its architecture.

2. In a simple RNN, what is the hidden state h_t at time step t calculated based on?

A. Only the current input x_t.

B. Only the previous hidden state h_{t-1}.

C. The current input x_t and the previous hidden state h_{t-1}.

D. Only the network's output at time step $t-1$.

3. What is the primary cause of the vanishing gradient problem in RNNs?

A. Using too many layers in the network.

B. The repeated multiplication of small values during backpropagation through time.

C. A learning rate that is too large.

D. The dataset size being too small.

4. In an LSTM, which gate is responsible for deciding what information to discard from the cell state?

A. Input Gate.

B. Forget Gate.

C. Output Gate.

D. Reset Gate.

5. Compared to an LSTM, what is a characteristic of a GRU?

A. It is more complex, with four gates.

B. It has separate cell state and hidden state vectors.

C. It is simpler, combining the forget and input gates into a single update gate.

D. It cannot solve the vanishing gradient problem.

6. Which gate in an LSTM uses a 'tanh' activation function to create new candidate values?

A. The Forget Gate.

B. Part of the Input Gate (to create $\tilde{C}_t$).

C. The Output Gate.

D. All three gates use 'tanh'.

7. What is the main purpose of using a Bidirectional RNN (Bi-RNN)?

A. To reduce training time.

B. To process two input sequences at once.

Name:..... Class:.....

C. To allow the network to use both past and future context at each time step.

D. To simplify the network architecture.

8. Which of the following tasks is a form of sequence labeling?

A. Machine Translation.

B. Text Classification.

C. Part-of-Speech Tagging.

D. Text Summarization.

Part II: Short Answer (7 questions, 10.0 points)

9. Write the formula for calculating the hidden state h_t in a simple RNN and explain the meaning of each term.

10. Explain why repeated multiplication during Backpropagation Through Time (BPTT) leads to vanishing or exploding gradients.

11. Write the full set of equations for an LSTM cell, including all gates and update equations.

12. Explain the role of the cell state (C_t) in an LSTM and how its additive update mechanism mitigates vanishing gradients.

13. Compare and contrast the architectures of a GRU and an LSTM, focusing on gates and the handling of the cell state.

14. Draw a block diagram of a Bidirectional RNN and explain how the final output at each time step is computed.

15. Describe how an RNN can be used as a language model.

Part III: Coding Practice (5 questions, 6.0 points)

16. Given $W_{xh} = \begin{pmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{pmatrix}$, $W_{hh} = \begin{pmatrix} 0.5 & 0.6 \\ 0.7 & 0.8 \end{pmatrix}$, input $x_t = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, and $h_{t-1} = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}$, compute the new hidden state h_t for a simple RNN with tanh activation and no bias.

17. Using NumPy, implement:

- a sigmoid function
- an LSTM forget gate function $lstm\_forget\_gate(xt, ht\_prev, Wf, bf)$.

18. Explain the steps to preprocess a raw text corpus for training an RNN-based language model.

Name:..... Class:.....

19. You are building a POS tagger using a Bi-LSTM. a) Why is a Bi-LSTM better than a unidirectional LSTM? b) What are the input and output tensor shapes?

20. In the Keras code below, identify the recurrent layer, number of hidden units, and activation function:

```
model = Sequential([
    Embedding(vocab_size, embedding_dim, input_length=max_length),
    SimpleRNN(32, activation='tanh'),
    Dense(1, activation='sigmoid')
])
```

Name:..... Class:.....

9. Write the formula for calculating the hidden state h_t in a simple RNN and explain the meaning of each term.

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

- h_{t-1}: previous hidden state
- W_R: Weight matrix that determine which info in the previous matrix to mix
- x_t: input at step t
- W_x: Weight matrix that map the input to the embedding space and determine which info of x_t that need to mix with th prese hidden state

10. Explain why repeated multiplication during Backpropagation Through Time (BPTT) leads to vanishing or exploding gradients.

The information that we can keep after processing long sequences are minimal over times by multi multiplicating operations. After unrolling the network, the network is usually deep with large amount of layers, causing the gradients to be vanished or exploded to the first layers. after repeating multiplications of chain rule and Jacobian-

11. Write the full set of equations for an LSTM cell, including all gates and update equations.

Forget gate:

$$f_t = \sigma\left(U_f h_{t-1} + W_f x_t\right)$$

$$h_t = c_{t-1} \odot f_t$$

- The candidate vector to be added to cell state

$$g_t = \tanh\left(U_g h_{t-1} + W_g x_t\right)$$

Add gate (Select the actual information that need to be extracted)

$$i_t = \sigma\left(U_i h_{t-1} + W_i x_t\right)$$

$$j_t = g_t \odot i_t$$

cell state:

$$c_t = j_t + h_t$$

Output gate

$$o_t = \sigma\left(U_o h_{t-1} + W_o x_t\right)$$

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

12. Explain the role of the cell state (C_t) in an LSTM and how its additive update mechanism mitigates vanishing gradients.

The cell state hold the long-term information of the previous steps. In order to maintain the relevant context, the cell state captures the important information and drop the unnecessary information from a long distance. Due to this unit, The input in the very first steps still can significantly contribute to the future decisions, allowing the gradients in the future can back-propagated back to the first layers after unrolling the network and the additive update mechanism of the cell state avoid multiple multiplications when back-prop

$$C_t = f_t \odot C_{t-t} + i_t \odot g_t$$

13. Compare and contrast the architectures of a GRU and an LSTM, focusing on gates and the handling of the cell state.

GRU

Update gate: decide how much the units update it's content

$$z_t = \sigma(W_z x_t + U_z h_{t-1})$$

Reset gate: decide to forget which information of the previous computed state

$$r_t = \sigma(W_r x_t + U_r h_{t-1})$$

Candidate activation:

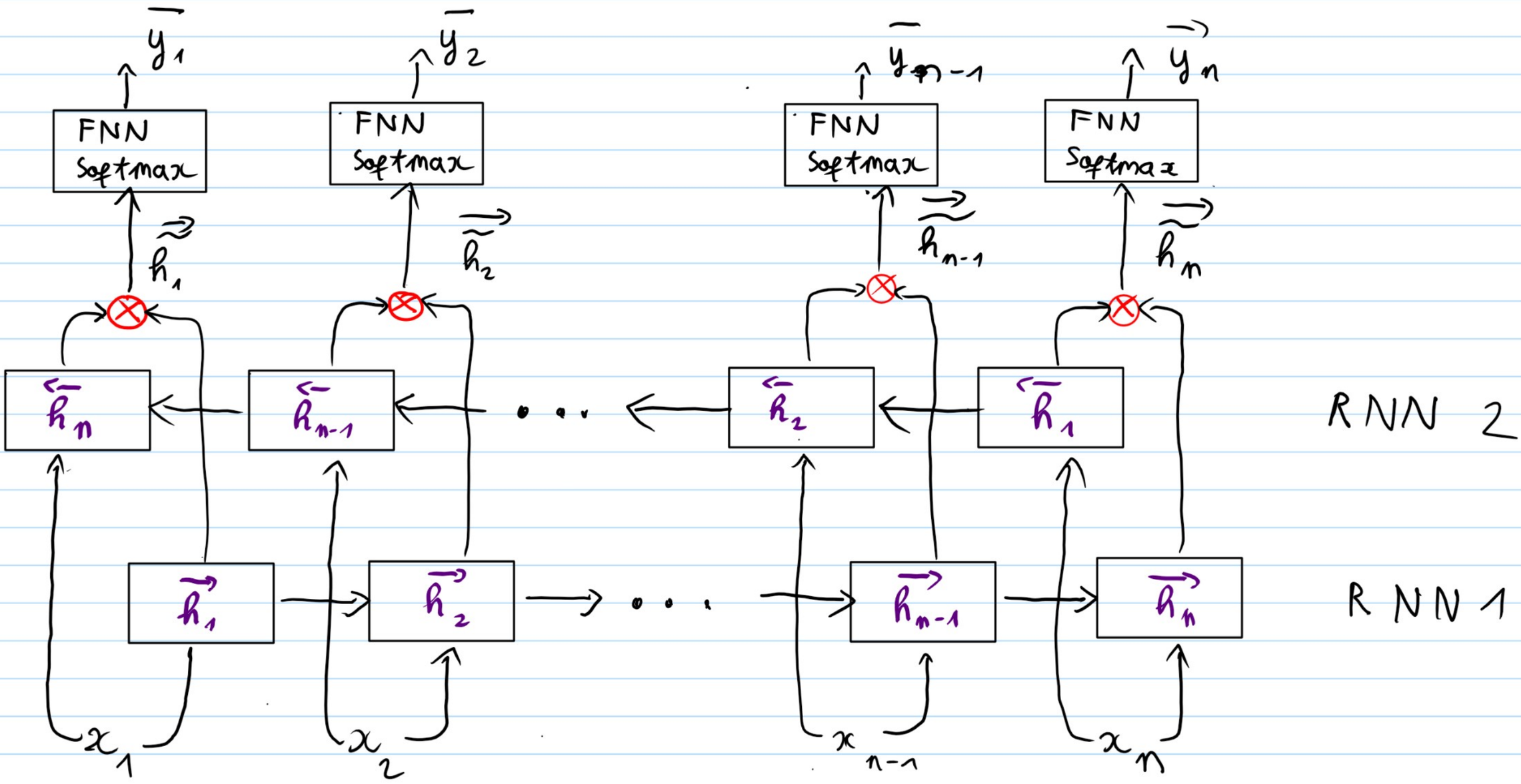
$$\tilde{h}_t = \tanh(W x_t + U.(r_t \odot h_{t-1}))$$

Update function:

$$h_t = (1 - z_t).h_{t-1} + z_t.\tilde{h}_t$$

GRU manages long-term dependencies by directly using the hidden state h_t and sequently updating hidden state by linear interpolation. On the other hand, LSTM uses additional cell state to capture long-term relevant context. The reset gate of GRU has the similar effects as the forget gate of LSTM, which is forgetting un-relevant context. The update gate of GRU not only can select the information that are important like the add gate of LSTM, but also can forget the unecessary information of the previous hidden state. LSTM has the additional output gate that control which parts of cell state are exposed to the hidden state, acting like a filter on the cell state fore producing the final output.

14. Draw a block diagram of a Bidirectional RNN and explain how the final output at each time step is computed.



- The architecture processes the sequences from two directions. The outputs of bi-directional RNN are produced by concatenating hidden states from two directions. The outputs are then passed to a feed-forward neural network with softmax activation to predict the tokens.

15. Describe how an RNN can be used as a language model.

input texts are word embeddings that are added to the RNN, producing hidden states and probabilities to predict the next tokens. The produced hidden states and the predictions are added as the next input to generate probabilities for the next predictions

16. Given $W_{xh} = \begin{pmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{pmatrix}$, $W_{hh} = \begin{pmatrix} 0.5 & 0.6 \\ 0.7 & 0.8 \end{pmatrix}$, input $x_t = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, and $h_{t-1} = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}$, compute the new hidden state h_t for a simple RNN with tanh activation and no bias.

$$h_t = \tanh\left(W_{xh} x_t + W_{hh} \cdot h_{t-1} \right)$$

$$= \tanh\left(\begin{pmatrix} 0,1 \\ 0,3 \end{pmatrix} + \begin{pmatrix} 0,55 \\ 0,75 \end{pmatrix} \right)$$

$$= \tanh\left(\begin{pmatrix} 0,65 \\ 1,05 \end{pmatrix} \right) = \begin{pmatrix} 0,572 \\ 0,782 \end{pmatrix}$$

18. Explain the steps to preprocess a raw text corpus for training an RNN-based language model.

Tokenize → Create Vocabulary → Create (or download) embedding
→ Embed the tokens.

19. You are building a POS tagger using a Bi-LSTM. a) Why is a Bi-LSTM better than a unidirectional LSTM? b) What are the input and output tensor shapes?

a) By adapting information in the past and future, Bi-LSTM can learn richer context for more accurate predictions.

b) input : (N, T, embedding dim)

output : (N, T, num_classes)

N : Batch size

T : Sequence length.

20. In the Keras code below, identify the recurrent layer, number of hidden units, and activation function:

```
1 model = Sequential([
2     Embedding(vocab_size, embedding_dim, input_length=max_length),
3     SimpleRNN(32, activation='tanh'),
4     Dense(1, activation='sigmoid')
5 ])
```

Recurrent layer.

Activation: tanh (in recurrent layer)

sigmoid (in FNN)

num hidden units: 32