COMP4332 and RMBI4310 Final Exam Answers

May 21, 2019, Tuesday
Time: 8:30am-11:30am
Instructor: Yangqiu Song

Name: _____ **Student ID:** _____

Question	Score	Question	Score
1	/ 10	6	/ 9
2	/ 10	7	/ 12
3	/ 10	8	/ 10
4	/ 10	9	/ 10
5	/ 9	10	/ 10
Total:		**/ 100**	

Q1. Yes/No Questions (10 points)

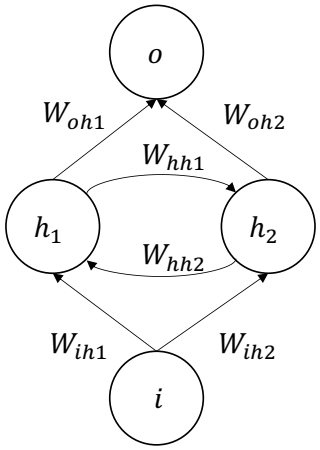
Indicate whether each statement is true (✓) or false (×).

1. Weight sharing occurs in Recurrent Neural Network (RNN).
2. Convolutional Neural Network (CNN) is designed for images; Recurrent Neural Network (RNN) is designed for sequences. Hence, CNN cannot be applied in NLP tasks.
3. CBOW and Skip-gram learn high-quality distributed vector representations that capture a large number of precise syntactic and semantic word relationships, but they are lack of dynamic, context-dependent, and contextualized information.
4. The sequence-to-sequence model (without attention mechanism) often fails to capture the complex semantic relations between words entirely as well as the long-term dependencies.
5. In a graph, the most important node (the node with highest centrality) could vary based on different measure methods.
6. Expressivity and efficiency are two advantages of graph CNN.
7. Spider trap happens because some of the nodes violate the conditions needed for the random walk theorem.
8. If we use Pearson Correlation to model user-by-user similarity, possible similarity values are between 0 and 1.
9. Item-based collaborative filtering itself solves the scalability problem of the recommendation system.
10. In Wide & Deep Learning, we jointly train a wide linear model (for generalization) alongside a deep neural network (for memorization).

Solution TFTTT,FTFFF (10 points, 1 point for each)

Q2. RNN (10 points)

Consider the following recurrent neural network, in which the bottom node $i \in \mathbb{R}^{100}$ is the input, the top node $o \in \mathbb{R}^{50}$ is the output, and the middle two nodes $h_1 \in \mathbb{R}^{128}, h_2 \in \mathbb{R}^{128}$ are hidden states.



We have following equations for this network:

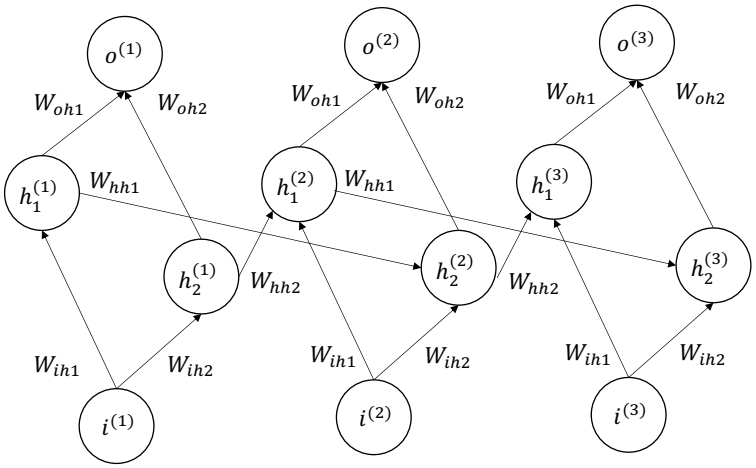
$$h_1^{(t)} = W_{ih1}^T i + W_{hh2}^T h_2^{(t-1)}$$
$$h_2^{(t)} = W_{ih2}^T i + W_{hh1}^T h_1^{(t-1)}$$
$$o = W_{oh1}^T h_1^{(t)} + W_{oh2}^T h_2^{(t)}$$

(a) (6 points) Draw an equivalent, feed-forward (non-recurrent) neural network representing three time instances.

(b) (4 points) What are the shapes of $W_{ih1}, W_{hh1}, W_{hh2}$, and W_{oh1} ?

Solution

(a)



(b)

$$W_{ih1} \in \mathbb{R}^{100\times128}$$
$$W_{hh1} \in \mathbb{R}^{128\times128}$$
$$W_{hh2} \in \mathbb{R}^{128\times128}$$
$$W_{oh1} \in \mathbb{R}^{128\times50}$$ (4 points, 1 point for each)

Q3. CNN (10 points)

We have an input image which is a $100 \times 100 \times 1$ tensor, and convolve it with 10 filters each of size 3×3, 10 filters each of size 4×4, and 10 filters each of size 5×5, using a stride of 2 and 'valid' padding. What are the output sizes of convolutional layers? How many parameters are required?

If we also apply the max-pooling after the convolution layer with a stride of 2, a filter of size 2, and 'same' padding after the convolution operation. What are the output sizes of max-pooling layers? How many parameters are required totally?

Solution

Output sizes of convolutional layers (1 point for each):

$$(\lfloor \frac{100-3}{2} \rfloor + 1, \lfloor \frac{100-3}{2} \rfloor + 1, 10) = (49, 49, 10)$$
$$(\lfloor \frac{100-4}{2} \rfloor + 1, \lfloor \frac{100-4}{2} \rfloor + 1, 10) = (49, 49, 10)$$
$$(\lfloor \frac{100-5}{2} \rfloor + 1, \lfloor \frac{100-5}{2} \rfloor + 1, 10) = (48, 48, 10)$$

Number of parameters (2 points):

$$(3 \times 3 + 1) \times 10 + (4 \times 4 + 1) \times 10 + (5 \times 5 + 1) \times 10 = 530$$

Output sizes of max-pooling layers (1 point for each):

$$(\lceil \frac{49-2}{2} \rceil + 1, \lceil \frac{49-2}{2} \rceil + 1, 10) = (25, 25, 10)$$
$$(\lceil \frac{49-2}{2} \rceil + 1, \lceil \frac{49-2}{2} \rceil + 1, 10) = (25, 25, 10)$$
$$(\lceil \frac{48-2}{2} \rceil + 1, \lceil \frac{48-2}{2} \rceil + 1, 10) = (24, 24, 10)$$

Number of parameters (2 points):

530 (max-pooling layers do not require parameters)

Q4. Attention (10 points)

Sequence to Sequence (Seq2Seq) is about training models to convert sequences from one domain (e.g. sentences in French) to sequences in another domain (e.g. the same sentences translated to English), introduced in 2014 by Sutskever et al. But the simple version Seq2Seq utilizes limited information. Attention mechanism is a simple but effective way to improve the performance of Seq2Seq.

The details of dot-product attention are as follows:

1. Compute the dot product: $e_{t,j} = \boldsymbol{s_{t-1}^T h_j}$, where $\boldsymbol{s_{t-1}^T}$ is the transpose of $\boldsymbol{s_{t-1}}$ which is the decoder hidden state at time $t-1$, $\boldsymbol{h_j}$ is the encoder hidden state at time j.
2. Compute the weight: $\alpha_{t,j} = \frac{\exp(e_{t,j})}{\sum_{k=0}^{T-1} \exp(e_{t,k})}$
3. Compute the attention output: $\boldsymbol{c_t} = \sum_{j=0}^{T-1} \alpha_{t,j} \boldsymbol{h_j}$
4. Concatenate the attention output with decoder hidden state before sending the decoder RNN, or concatenate the attention output with the decoder output.

According to the description above, please compute the attention output at t=3.

h_0	h_1	h_2	h_3	s_0	s_1	s_2	s_3
$\begin{bmatrix}0.1\\0.2\\0.3\\0.4\end{bmatrix}$	$\begin{bmatrix}0.2\\0.3\\0.1\\0.4\end{bmatrix}$	$\begin{bmatrix}0.1\\0.4\\0.3\\0.2\end{bmatrix}$	$\begin{bmatrix}0.3\\0.4\\0.2\\0.1\end{bmatrix}$	$\begin{bmatrix}0.4\\0.1\\0.2\\0.3\end{bmatrix}$	$\begin{bmatrix}0.2\\0.3\\0.4\\0.1\end{bmatrix}$	$\begin{bmatrix}0.4\\0.2\\0.1\\0.3\end{bmatrix}$	$\begin{bmatrix}0.3\\0.1\\0.4\\0.2\end{bmatrix}$

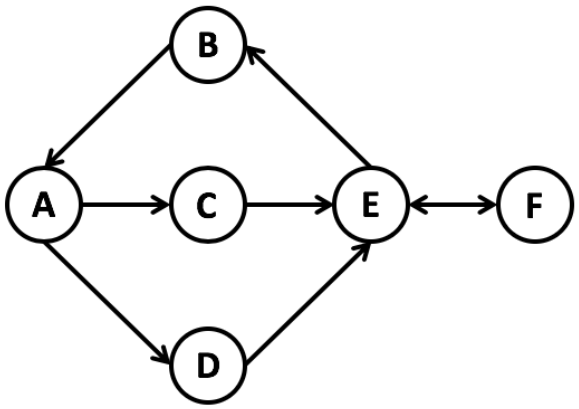
Solution

$$e_{3,:} = \begin{bmatrix}0.23 & 0.27 & 0.21 & 0.25\end{bmatrix} \text{ (4 points, 1 point for each)}$$

$$\alpha_{3,:} = \begin{bmatrix}0.2475 & 0.2575 & 0.2426 & 0.2524\end{bmatrix} \text{ (4 points, 1 point for each)}$$

$$\boldsymbol{c_3} = \begin{bmatrix}0.1762\\0.3248\\0.2233\\0.2758\end{bmatrix} \text{ (2 points)}$$

Q5. Graph Centrality (9 points)



In this question, you are required to compute the **normalized** degree, betweenness, and closeness centrality of all the nodes in the given directed graph. You should leave your answer in Table 1 and explain how do you compute these centralities in detail in the blank space. (0.5 point each)

Note: For the degree centrality, you should only use the out-degree to calculate the centrality and normalize the resulted centrality with the maximum possible degree.

Centrality Type	node A	node B	node C	node D	node E	node F
degree centrality						
betweenness centrality						
closeness centrality						

Table 1: Answer Space of Q6.

Solution

(1) degree centrality:

$$A : 2/5 = 0.4$$
$$B : 1/5 = 0.2$$
$$C : 1/5 = 0.2$$
$$D : 1/5 = 0.2$$
$$E : 2/5 = 0.4$$
$$F : 1/5 = 0.2$$

(2) betweenness centrality:

$$A : (1+1+1+1+0+1+0+0+0+1+0+0+0+1+1+0+0+1+1+0)/20 = 0.5$$
$$B : (0+0+0+0+1+1+0+0+1+1+0+0+1+1+1+0+1+1+1+0)/20 = 0.5$$
$$C : (0.5+0+0.5+0.5+0+0+0.5+0.5+0+0+0+0+0+0+0+0+0+0+0+0)/20 = 0.125$$
$$D : (0.5+0+0.5+0.5+0+0+0.5+0.5+0+0+0+0+0+0+0+0+0+0+0+0)/20 = 0.125$$
$$E : (1+0+0+1+0+0+0+1+1+1+1+1+1+1+1+1+1+1+1+1)/20 = 0.75$$
$$F : (0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0)/20 = 0$$

(3) closeness centrality:

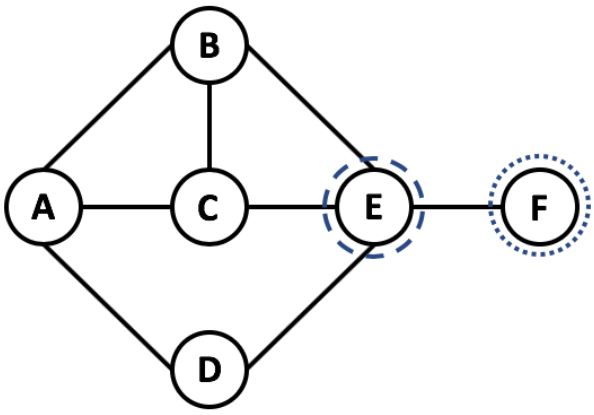
$$A : 5/(3+1+1+2+3) = 0.5$$
$$B : 5/(1+2+2+3+4) = 0.42$$
$$C : 5/(3+2+4+1+2) = 0.42$$
$$D : 5/(1+2+2+3+4) = 0.42$$
$$E : 5/(2+1+3+3+1) = 0.5$$
$$F : 5/(3+2+4+4+1) = 0.36$$

Thus the final result should be:

Centrality Type	node A	node B	node C	node D	node E	node F
degree centrality	0.4	0.2	0.2	0.2	0.4	0.2
betweenness centrality	0.5	0.5	0.125	0.125	0.75	0
closeness centrality	0.5	0.42	0.42	0.42	0.5	0.36

Table 2: Q6 result

Q6. Node2vec (9 points)



In this question, you are required to list all the paths generated from the biased random walk process. Assume that E is the last node which is indicated with the dashed circle, and F is the second last node which is indicated with the dotted circle. Assume p is 2, q is 0.5, and the remaining walk length is 2. You should list all possible paths and their associated probabilities.

Please write down all the necessary steps to compute the probabilities. The probability should be in the format of fraction rather than decimal (e.g., 3/4 rather 0.75). (1 point each)

Hints: your path should start with node E and contain 3 nodes.

Solution

(1) $E->B->A$: 4/13*4/7=16/91

(2) $E->B->C$: 4/13*2/7=8/91

(3) $E->B->E$: 4/13*1/7=4/91

(4) $E->C->A$: 4/13*4/7=16/91

(5) $E->C->B$: 4/13*2/7=8/91

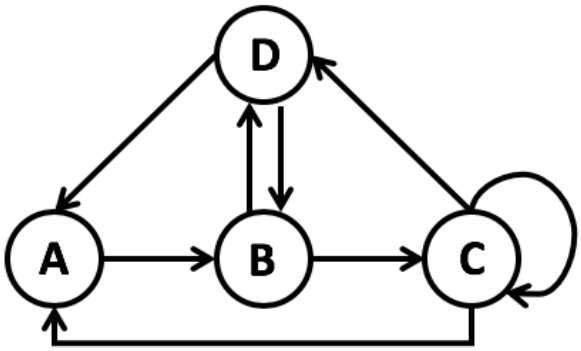
(6) $E->C->E$: 4/13*1/7=4/91

(7) $E->D->A$: 4/13*4/5=16/65

(8) $E->D->E$: 4/13*1/5=4/65

(9) $E->F->E$: 1/13*1=1/13 (1 point for each)

Q7. PageRank (12 points)



In this question, you need to compute the PageRank value for all the nodes in the given graph. You are required to use L1-norm to compute the difference between two iterations and the stopping threshold $\varepsilon = 0.1$. Your final answer should be in the format of fraction rather than decimal (e.g., 3/4 rather 0.75).

Hint 1 : When the L1-norm of the difference between current and last iteration is smaller than the threshold, the iteration should stop.

Hint 2 : We do not consider advanced cases that dealing with spider traps or dead ends.

Solution

	A	B	C	D
A	0	0	1/3	1/2
B	1	0	0	1/2
C	0	1/2	1/3	0
D	0	1/2	1/3	0

Table 3: M-matrix (3 points)

	A	B	C	D	Difference	Decision
Initial Value	1/4	1/4	1/4	1/4	-	Keep iterating
After iteration 1	5/24	3/8	5/24	5/24	1/4	Keep iterating
After Iteration 2	25/144	5/16	37/144	37/144	7/36	Keep iterating
After Iteration 3	185/864	29/96	209/864	209/864	35/432	Stop

Table 4: Iteration Process (9 points, 3 points for each iteration)

As $35/432 < 0.1$, the final answer should be : A (185/864=0.22); B (29/96=0.30); C (209/864=0.24); D (209/864=0.24).

Q8. User CF (10 points)

The hottest movie *Avengers: Endgame* has been on screen. Alice needs to decide whether she should come to watch this movie. She has a very closed friend B, who has not watched this movie yet. But three of B's friends C, D, E have watched it and rated this movie into 1, -1, -1 separately. We also have historical movie ratings from them as follows:

	Movie 1	Movie 2	Movie 3	Movie 4
Alice	1	1	1	1
B	1	1	-1	1
C	-1	-1	1	1
D	-1	1	-1	-1
E	1	1	-1	1

Assuming we want to use Recursive User CF to determine whether to recommend movie *Avengers: Endgame* to Alice. Can you predict the rating for Alice?

Hints: The similarity measure in this problem is cosine similarity, and the prediction function is as follows:

$$pred(a,p) = \overline{r}_a + \frac{\sum_{b \in N} sim(a,b) \times (r_{b,p} - \overline{r}_b)}{\sum_{b \in N} sim(a,b)}$$

Here for the average rating we do not consider the new movie *Avengers: Endgame*.

Solution

$$cosine\_sim(x,y) = \frac{x \cdot y}{\|x\| \, \|y\|}$$

$sim(B,C) = -0.5,$
$sim(B,D) = 0.0,$
$sim(B,E) = 1.0$ (3 points, 1 point for each)

$\overline{r}_A = (1+1+1+1)/4 = 1,$
$\overline{r}_B = (1+1+-1+1)/4 = 0.5,$
$\overline{r}_C = (-1+-1+1+1)/4 = 0$
$\overline{r}_D = (-1+1+-1+-1)/4 = -0.5,$
$\overline{r}_E = (1+1+-1+1)/4 = 0.5$

Rating for B: $0.5 + \frac{(-0.5*(1-0))+(1.0*(-1-0.5)}{-0.5+1.0} = -3.5$ (1 points)

$sim(A, B) = 0.5, sim(A, C) = 0.0,$
$sim(A, D) = -0.5, sim(A, E) = 0.5$ (4 points, 1 point for each)

Rating for A: $1.0 + \frac{(0.5*(-3.5-0.5))+0+(-0.5*(-1+0.5))+(0.5*(-1-0.5))}{0.5+0.0-0.5+0.5} = -4.0$ (2 points)

Q9. Item CF (10 points)

In the same situation of Q9, if B has watched *Avengers: Endgame* today and rated it as 1, can you predict the rating for Alice using Item CF?

	Movie 1	Movie 2	Movie 3	Movie 4
Alice	1	1	1	1
B	1	1	-1	1
C	-1	-1	1	1
D	-1	1	-1	-1
E	1	1	-1	1

Hints: In this problem, the similarity measure is cosine similarity and you should only consider top 2 similar movies. The prediction function is as follws:

$$pred(u,p) = \frac{\sum_{i \in ratedItem(u)} sim(i,p) * r_{u,i}}{\sum_{i \in ratedItem(u)} sim(i,p)}$$

Solution

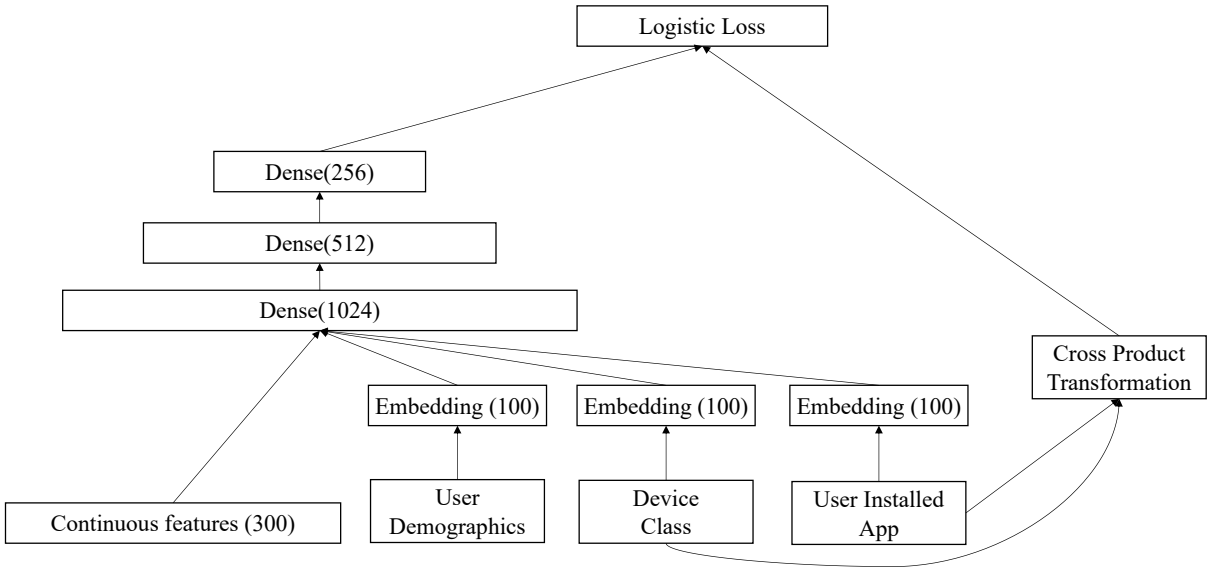
$$cosine\_sim(x,y) = \frac{x \cdot y}{\|x\| \, \|y\|}$$

$sim(Ave, M1) = 0.0$
$sim(Ave, M2) = -0.5$
$sim(Ave, M3) = 0.5$
$sim(Ave, M4) = 0.5$ (8 points, 2 points for each)

top 2 similar movies are M3 and M4.
Rating for *Avengers: Endgame*: $\frac{0.5*1+0.5*1}{0.5+0.5} = 1.0$ (2 points)

Q10. Wide and Deep (10 points)

Consider the following Wide & Deep Network.



Assume that

- we have 300 continuous features.
- we have 323 unique user demographics.
- we have 32 unique device class.
- we have 24 unique user installed app.
- we use all the combinations of device class and user installed app as cross product transformation
- all the bias terms can be ignored.
- each embedding layer size is 100,
- output size of three dense layer are 1024, 512, 256 separately. Activation function is ReLu.
- the final output is 1-dimension probability.

Compute the number of all trainable parameters of this network.

$$
\begin{aligned}
N_{emb1} &= 323 * 100 = 32,300, \\
N_{emb2} &= 32 * 100 = 3,200, \\
N_{emb3} &= 24 * 100 = 2,400 \text{ (3 points, 1 point for each)}, \\
N_{fc1} &= 600 * 1024 = 614,400 \text{ (2 points)}, \\
N_{fc2} &= 1024 * 512 = 524,288 \text{ (1 point)}, \\
N_{fc3} &= 512 * 256 = 131,072 \text{ (1 point)}, \\
N_{fc_{out}} &= 32 * 24 + 256 = 1,024 \text{ (2 points)},
\end{aligned}
$$

$$
N_{total} = 32,300+3,200+2,400+614,400+524,288+131,072+1,024 = 1,308,684 \text{ (1 point)},
$$