Question 1. What are the main messages you learned from this chapter?

Memory network is an exapnsion of the attention models, whch consists of memory ***m*** and 4 components:

1. Input feature map I: input space -> representation of features
2. Generalization G: input space -> update old memories iteratively
3. Output feature map O: feature representation + current memory state -> output embeddings
4. Response: output embeddings -> response variable

The memory network uses the argmax function to determine the optimal memory slots and the best output, which could potentially break the gradient flows required for backpropagation. The end-to-end network describes the variation of the memory network that uses the softmax function in the final step.

Question 2. What related resources (book, paper, blog, link) do you recommend your classmates to checkout?

This appears to be an interesting website with examples and applications for learning the basic concepts.

http://wiki.pathmind.com/attention-mechanism-memory-network

Question 3. Which part do you want to improve in this chapter?

ipad

- Page 185: why don't we compute context vector over $$x_i$$ directly -> why don't we compute context vector $$c_i$$ over $$x_i$$ directly
- Page 190: their educational history, length of practice, and length of practice into the feature vector -> their educational history, length of education, and length of practice into the feature vector

Question 4. What is the main difference between the end-to-end memory network and the self-attention/transformer?

- End-to-end memory network: The final prediction uses the softmax function of the response vector ***o*** and the query vector ***u***.
- Self-attention / transformer: replaces the RNN encoder and decoder with an attention mechanism with the input to itself. The idea combines the similarity search of the query and the input then based on the similarity score, we can retrieve the input to generate the output embedding. Due to the independence of each query and input, the self attention mechanism enables parallel training. The transformer also includes multiple attention weights for the same input, which also known as the multi-attention weight matrix. The last component is the positional embeddings related to the position of $$x_i$$ in the input sequence. Observe that the positional encoding can also be used to identify a long-term trend in a time series without the time dependence axis.