1.can you think of few applications for a sequence to sequence RNN? what about a sequence to vector RNN? And a vector to sequence RNN ?

Ans : In Sequence to Sequence Learning, RNN is trained to map an input sequence to an output sequence which is not necessarily of the same length. Applications are speech recognition, machine translation, image captioning and question answering.

Sequence to vector is a feed the network a sequence of inputs, and ignore all outputs except for the last one.

Vector to sequence is a feed the network a single input at the first time step (and zeros for all other time steps), and let it output a sequence.

2.why do people use encoder-decoder RNNs rather than plain sequence to sequence RNNs for automic translation.?

Ans : seq-2-seq RNNs translate one word at a time,encoder-decoder RNNs read & translate a sentence at a time

3.How could you combine a convolutional neural network with an RNN to classify videos.?

Ans : Run a frame from each second of video through a CNN,Feed CNN outputs as input sequence to RNN,Feed RNN outputs to softmax layer for probabilities of each class.

4.what are the advantages of Bulinding an RNN using dynamic\_rnn() rather than static\_rnn()?

Ans : avoids out-of-memory errors,

directly takes single tensor as input and output (covering all time steps),no need to stack, unstack, or transpose,generates a smaller easier to visualize graph in TensorBoard.

5.How can you deal with variable length input sequences? What about variable length output sequences ?

Ans : set sequence\_length parameter when calling static\_rnn() or dynamic\_rnn(),

pad smaller input/output to make them same size as largest input/output

6.what is a common way to distribute training & Execution of a deep RNN across multiple GRUs ?

Ans : Deep Learning approach for modeling sequential data is Recurrent Neural Networks (RNN). RNNs were the standard suggestion for working with sequential data before the advent of attention models. Specific parameters for each element of the sequence may be required by a deep feedforward model.