1. Can you think of a few applications for a sequence-to-sequence RNN? What about a sequence-to-vector RNN? And a vector-to-sequence RNN?  
   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.

A variable-length context vector can be used instead of a ﬁxed-size vector. An Attention mechanism can be used to produces a sequence of vectors from the encoder RNN from each time step of the input sequence. The Decoder learns to pay selective attention to the vectors to produce the output at each time step.

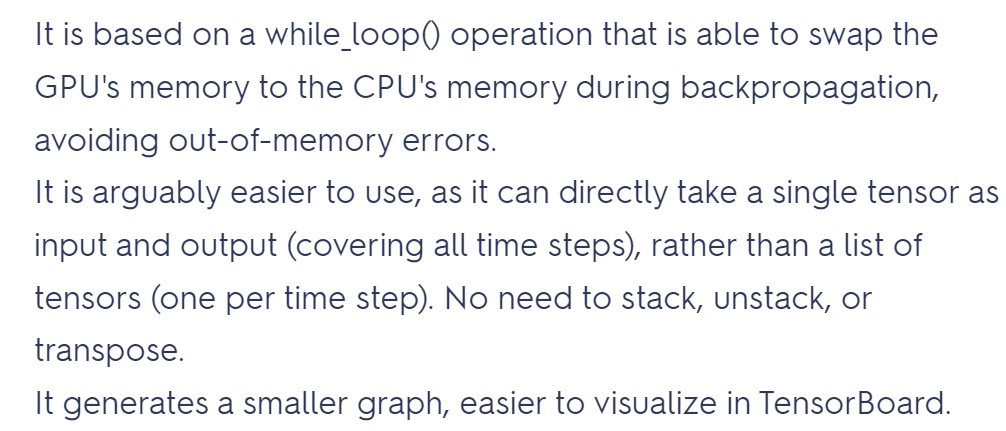
1. Why do people use encoder–decoder RNNs rather than plain sequence-to-sequence RNNs for automatic translation?

Encoder decoder models allow for a process in which a machine learning model generates a sentence describing an image. It receives the image as the input and outputs a sequence of words. This also works with videos.

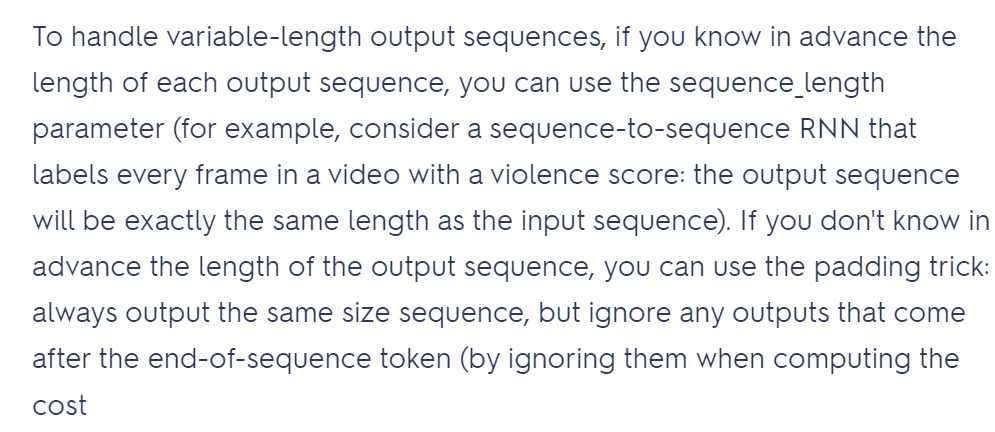
1. How could you combine a convolutional neural network with an RNN to classify videos?

RNN is a class of artificial neural network where connections between nodes form a directed graph along a sequence. It is basically a sequence of neural network blocks that are linked to each other like a chain.

1. What are the advantages of building an RNN using dynamic\_rnn() rather than static\_rnn()?



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



1. What is a common way to distribute training and execution of a deep RNN across multiple GPUs?

