**Assignment\_4**

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?

**Ans: Applications :  speech recognition, machine translation, image captioning and question answering.**

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

**Ans: The key benefits of the approach are the ability to train a single end-to-end model directly on source and target sentences and the ability to handle variable length input and output sequences of text.**

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

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

**Ans: In tensorflow, what is the difference between tf.nn.static\_rnn and tf.nn.dynamic\_rnn, and when to use them?**

**Both take a sequence\_length argument that adapts the computation to the actual length of the input; it is not as if static\_rnn is limited to fixed-size inputs, right?**

**dynamic\_rnn has the following extra arguments:**

* **parallel\_iterations**
* **swap\_memory**
* **time\_major**

**Static**

**Internally, tf.nn.rnn creates an unrolled graph for a fixed RNN length. That means, if you call tf.nn.rnn with inputs having 200 time steps you are creating a static graph with 200 RNN steps. First, graph creation is slow. Second, you’re unable to pass in longer sequences (> 200) than you’ve originally specified.**

**Dynamic**

**tf.nn.dynamic\_rnn solves this. It uses a tf.While loop to dynamically construct the graph when it is executed. That means graph creation is faster and you can feed batches of variable size**

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

**Ans: There are two ways to run a single model on multiple GPUs, data parallelism and device parallelism. In most cases, what you need is most likely data parallelism. Data parallelism consists of replicating the target model once on each device and using each replica to process a different fraction of the input data.**