What are the pros and cons of using a stateful RNN versus a stateless RNN?

Ans. Stateless means there is no memory of the past. Every transaction is performed as if it were being done for the first time. Main advantage is that it simplifies the server design because there is no need to dynamically allocate storage to deal with conversations in progress. Also you can easily identify each request and each one is independent. So disadvantages is pretty obvious.

Stateful means that there is memory of the past. Previous transactions are remembered and may affect the current transaction. It is a core part to implement intelligent network service platform comprising of services like QoS, security, firewalls, bandwidth management, monitoring, metering, billing etc. It's much more complex than stateless.

Please note that all of these information you could (and honestly you should in next time) find by yourself on the internet. Don't think about this site like it is a space where you get answers without any effort. I am not sure about policy of sharing external link but if you are interest about some details this older source-link could be pretty helpful.

Why do people use Encoder–Decoder RNNs rather than plain sequence-to-sequence RNNs for automatic translation?

Ans. People use encoder decoder RNN because of following reasons :

-- The encoder is the used to map a source that is of variable length sequence to a vector which is fixed length.

-- The decoder is used to map the vector representation back into the variable length sequence.

-- This model can handle variable length of input as well as output.

-- It has a capability of training the model to direct at single source and destination.

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

Ans. The most common way people deal with inputs of varying length is padding.

You first define the desired sequence length, i.e. the input length you want your model to have. Then any sequences with a shorter length than this are padded either with zeros or with special characters so that they reach the desired length. If an input is larger than your desired length, usually you'd split it into multiple inputs.

What is beam search and why would you use it? What tool can you use to implement it?

Ans. The beam search algorithm selects multiple tokens for a position in a given sequence based on conditional probability. The algorithm can take any number of N best alternatives through a hyperparameter know as Beam width.

What is an attention mechanism? How does it help?

Ans. The idea behind the attention mechanism was to permit the decoder to utilize the most relevant parts of the input sequence in a flexible manner, by a weighted combination of all of the encoded input vectors, with the most relevant vectors being attributed the highest weights.

What is the most important layer in the Transformer architecture? What is its purpose?

Ans. The most important part here is the “Residual Connections” around the layers. This is very important in retaining the position related information which we are adding to the input representation/embedding across the network. The network displayed catastrophic results on removing the Residual Connections. Multi-head attention plays a crucial role in the recent success of Transformer models, which leads to consistent performance improvements over conventional attention in various applications. The popular belief is that this effectiveness stems from the ability of jointly attending multiple positions.

When would you need to use sampled softmax?

Ans. Sampled Softmax. Sampled softmax aims to approximate a full softmax during model training (Bengio & Sénécal, 2008; 2003). Rather than computing the loss over all classes, only the positive class and a sample of m negative classes are considered. Each negative class is sampled with probability qi with replacement.