# Seq2seq Model

This is a simple model that uses an rnn encoder and and rnn decoder to to go from one sequence of values to another.

## Attention in seq2seq

There are a couple of notable ways to use attention in seq2seq networks, the first method is the most simple. We run the inputs through an encoder rnn and return a sequence of hidden states for each time point. We can then use the final hidden state as query, and all the other hidden states as keys and values to get an attention based starting hidden state, this can then be used as the initial hidden state in a decoder, or it can be passed as inputs to a decoder. This is a less effective method since we only have attention at one time point, it might be more effective at removing the effect of outliers or weird data points from our state consideration, however the model itself should learn to do so when managing its own internal recurrent states.

The next method is to do the same sort of thing for the encoder, but now we run the decoder, initializing the decoder states in some manner either based upon the final states of the encoder or some zero-based vector potentially. In translation tasks sometimes this can be initialized with the start seq vector, this is most common when we use some form of attention tho as we do need to pass the context of the inputs in somehow. One method of doing so is to repeat the context-vector we get form the encoder N times for as many outputs as we have, or simply using the last hidden state from the encoder as the input to the first step of the decoder, the iteratively using the last steps hidden state as input and passing the states through each step of the rnn. The attention now comes in as we use the output at each step from the decoder as the query and the context-vectors of the encoder to get an attention-based result.

The last method I read about is one that makes the most sense to me in how it feels like these should operate.

Diagram

Description automatically generated