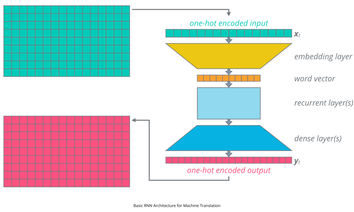
Basic RNN Architecture

Once we have the sequence of word vectors, we can feed them in one at a time to the neural network.



Basic RNN Architecture for Machine Translation

Embedding Layer

The first layer of the network is typically an embedding layer that helps enhance the representation of the word. This produces a more compact word vector that is then fed into one or more recurrent layers.

Recurrent Layer(s)

This is where the magic happens! The recurrent layer(s) help incorporate information from across the sentence, allowing each output word to be affected by potentially any (or no) input word.

Note: You could skip the embedding step, and feed in the one-hot encoded vectors directly to the recurrent layer(s). This may reduce the complexity of the model and make it easier to train, but the quality of translation may suffer as one-hot encoded vectors cannot exploit similarities and differences between words.

Dense Layer(s)

The output of the recurrent layer(s) is fed into one or more fully-connected dense layers that produce softmax output, which can be interpreted as one-hot encoded words in the target language.

As each word is passed in as input, its corresponding translation is obtained from the final output. The output words are then collected in a sequence to produce the complete translation of the input sentence.

Note: For efficient processing we would like to capture the output in a matrix of fixed size, and for that we need to have output sequences of the same length. Again, we can achieve this by using the same padding technique as we used for input.

Recurrent Layer: Internals

Let's take a closer look at what is going on inside a recurrent layer.