9.05 Recurrent Neural Networks

What is sequential data?

- If there is a particular order in which related things follow each other, we call it as a sequence

“i am going to watch Star Wars” vs. **“am i going to watch Star Wars”**

- Do you think both sentences mean the same?
- NO! which means the position of words is very important!
- They are a sequence of words

What is a Recurrent Neural Network (RNN)?

- RNNs are a kind of NN that specialise **in processing sequences**

- They are often used in NLP tasks because of their effectiveness in handling text

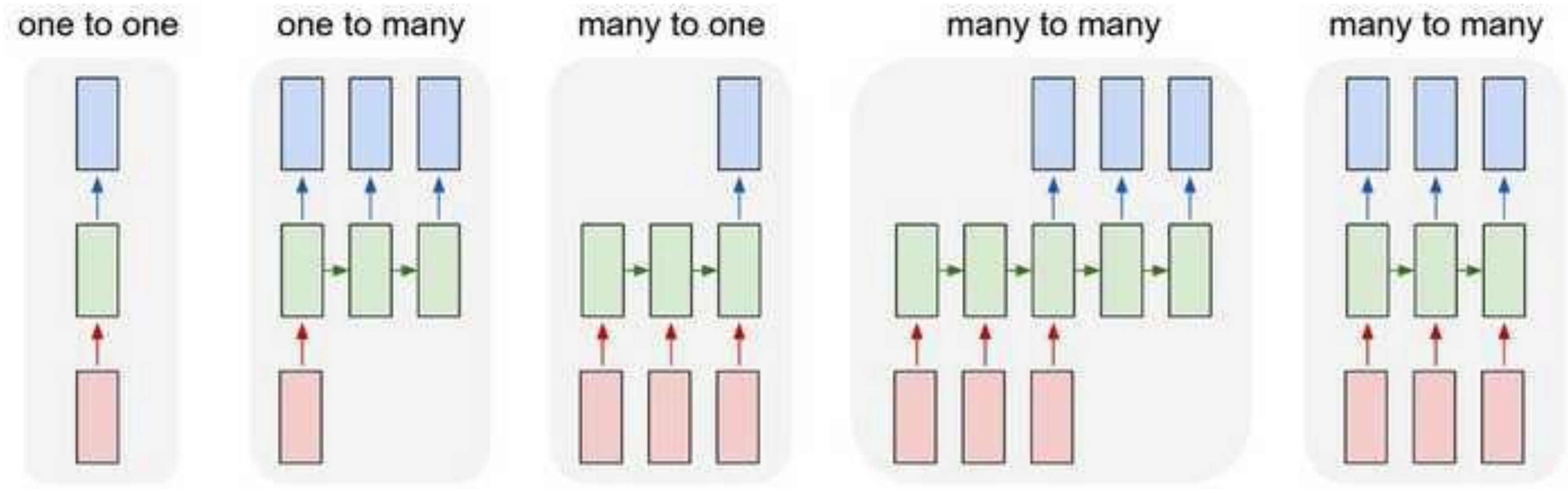
Why RNN?

- One issue with ANNs and CNNs is that they only work with predetermined sizes where they take fixed-size inputs and produce fixed-size outputs.

- RNNs are useful because they let us have variable-length sequences as both inputs and outputs.

Why RNN?

- Here are a few examples of what RNNs can look like:



Inputs are red, the RNN itself is green, and outputs are blue

Where can I use RNN?

- The ability to process sequences makes RNNs very useful. For example:
  - **Machine Translation** (e.g. Google Translate) is done with “many to many” RNNs. The original text sequence is fed into an RNN, which then produces translated text as output.
  - **Sentiment Analysis** (e.g. Is this a positive or negative review?) is often done with “many to one” RNNs. The text to be analyzed is fed into an RNN, which then produces a single output classification (e.g. This is a positive review).

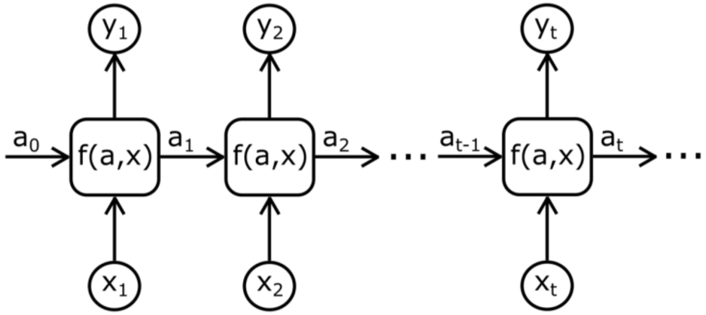
Where can I use RNN?

- **Speech Recognition:** Given an input sequence of acoustic signals from a sound wave, we can predict a sequence of phonetic segments together with their probabilities.

- **Generating Image Descriptions:** Together with CNNs, RNNs have been used as part of a model to generate descriptions for unlabeled images.

- **Chatbots:** Chatbots can give reply to your queries. When a sequence of words is given as the input, sequence of words will be generated at the output.

How does RNN work?



- Let’s consider a “many to many” RNN with inputs x_1, x_2, … x_t that generates outputs y_1, y_2, … y_t. These inputs and outputs are vectors with arbitrary dimensions.

- In an RNN, we take **2 inputs** ➔input feature and previous hidden value, and **2 outputs** ➔ output prediction and next hidden value.

- RNNs work by iteratively updating a hidden state, *a*, which encodes some information about the sequence which will help in predicting the next character.

- At any given step t:
  1. The next hidden state a_t is calculated using the previous hidden state a_{t-1} and the next input x_t
  2. The next output y_t is calculated using a_t