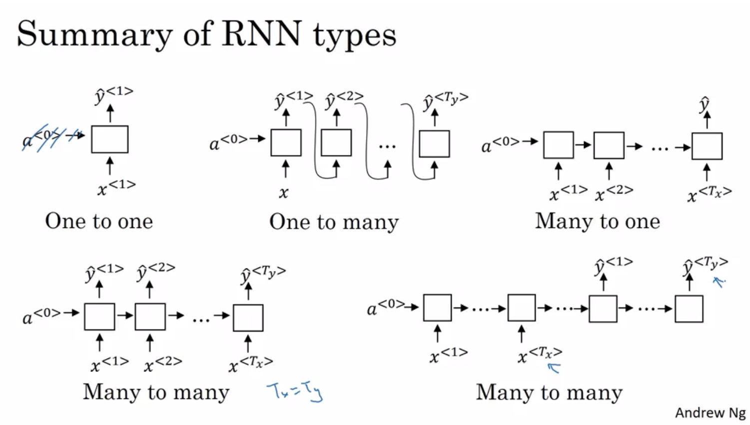
Prev Next



Different types of RNNs

Interactive Transcript

Search Transcript

English

0:00

So far, you've seen an RNN architecture where the number of inputs, Tx, is equal to the number of outputs, Ty. It turns out that for other applications, Tx and Ty may not always be the same, and in this video, you'll see a much richer family of RNN architectures. You might remember this slide from the first video of this week, where the input x and the output y can be many different types. And it's not always the case that Tx has to be equal to Ty. In particular, in this example, Tx can be length one or even an empty set. And then, an example like movie sentiment classification, the output y could be just an integer from 1 to 5, whereas the input is a sequence. And in name entity recognition, in the example we're using, the input length and the output length are identical, but there are also some problems were the input length and the output length can be different.They're both our sequences but have different lengths, such as machine translation where a French sentence and English sentence can mean two different numbers of words to say the same thing. So it turns out that we could modify the basic RNN architecture to address all of these problems. And the presentation in this video was inspired by a blog post by Andrej Karpathy, titled, The Unreasonable Effectiveness of Recurrent Neural Networks. Let's go through some examples. The example you've seen so far use Tx equals Ty, where we had an input sequence x(1), x(2) up to x(Tx), and we had a recurrent neural network that works as follows when we would input x(1) to compute y hat (1), y hat (2), and so on up to y hat (Ty), as follows. And in early diagrams, I was drawing a bunch of circles here to denote neurons but I'm just going to make those little circles for most of this video, just to make the notation simpler. So, this is what you might call a many-to-many architecture because the input sequence has many inputs as a sequence and the outputs sequence is also has many outputs. Now, let's look at a different example. Let's say, you want to address sentiments classification. Here, x might be a piece of text, such as it might be a movie review that says, "There is nothing to like in this movie." So x is going to be sequenced, and y might be a number from 1 to 5, or maybe 0 or 1. This is a positive review or a negative review, or it could be a number from 1 to 5. Do you think this is a one-star, two-star, three, four, or five-star review? So in this case, we can simplify the neural network architecture as follows. I will input x(1), x(2). So, input the words one at a time. So if the input text was, "There is nothing to like in this movie." So "There is nothing to like in this movie," would be the input. And then rather than having to use an output at every single time-step, we can then just have the RNN read into

hen it has already input the entire sentence. So,
3ecause as many inputs, it inputs many words and
ess, there is also a one-to-one architecture. So this
ıral network, we have some input x and we just had
work that we covered in the first two courses in
so have a one-to-many architecture. So an example
generation. And in fact, you get to implement this
re you go is have a neural network, output a set of
ould be maybe just an integer, telling it what genre
want, and if you don't want to input anything, x
well. For that, the neural network architecture
e first value, and then, have that, with no further
The third value, and so on, until you synthesize the
is input a(0) as well. One technical now what you
g sequences, often you take these first synthesized
architecture actually ends up looking like that. So,
any, as well as one-to-one. It turns out there's one
describing. Which is when the input and the output
saw just now, the input length and the output
nachine translation, the number of words in the
words in the output sentence, say the translation
here's an alternative new network architecture
entence. So first, reading the input, say French
done that, you then, have the neural network
h this architecture, Tx and Ty can be different
And so, this that collinear network architecture has
:, say a French sentence, and then, there's is a
decoder, which having read in the sentence, outputs the translation into a different language. So this would be an example of a many-to-many architecture. So by the end of this week, you have a good understanding of all the components needed to build these types of architectures. And then, technically, there's one other architecture which we'll talk about only in week four, which is attention based architectures. Which maybe isn't cleanly captured by one of the diagrams we've drawn so far. So, to summarize the wide range of RNN architectures, there is one-to-one, although if it's one-to-one, we could just give it this, and this is just a standard generic neural network. Well, you don't need an RNN for this. But there is one-to-many. So, this was a music generation or sequenced generation as example. And then, there's many-to-one, that would be an example of sentiment classification. Where you might want to read as input all the text with a movie review. And then, try to figure out that they liked the movie or not. There is many-to-many, so the name entity recognition, the example we've been using, was this where Tx is equal to Ty. And then, finally, there's this other version of many-to-many, where for applications like machine translation, Tx and Ty no longer have to be the same. So, now you know most of the building blocks, the building are pretty much all of these neural networks except that there are some subtleties with sequence generation, which is what we'll discuss in the next video. So, I hope you saw from this video that using the basic building blocks of an RNN, there's already a wide range of models that you might be able put together. But as I mentioned, there are some subtleties to sequence generation, which you'll get to implement yourself as well in this week's primary exercise where you implement a language model and hopefully, generate some fun sequences or some fun pieces of text. So, what I want to do in the next video, is go deeper into sequence generation. Let's see the details in the next video.

Downloads

Lecture Video mp4

Subtitles (English) WebVTT

Transcript (English) txt

Powerpoint slides pptx

Would you like to help us translate the transcript and subtitles into additional languages?