Recurrent Neural Networks

Shao Michael

Recap

- MLPs
- CNNs
- They all take in a fixed input and output a fixed output.
- What if we want the input to change in size.

Tasks

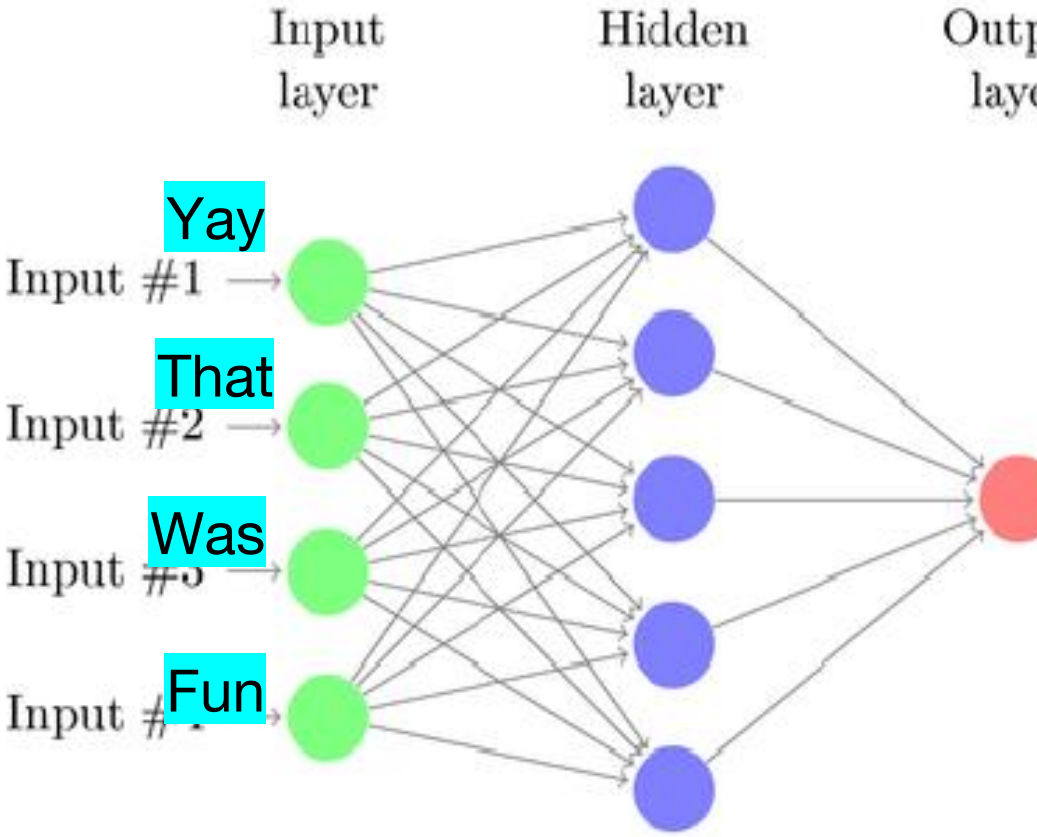
- Or what if we want to detect emotion from a sentence like:
- “Yay that was fun!”
- first we need to know how to input this sentence.

Tasks

- Or what if we want to detect emotion from a sentence like:
- “Yay that was fun!”

- first we need to know how to input this sentence:

- We can assign each word a number (details coming soon!), and use 4 inputs for the sentence!

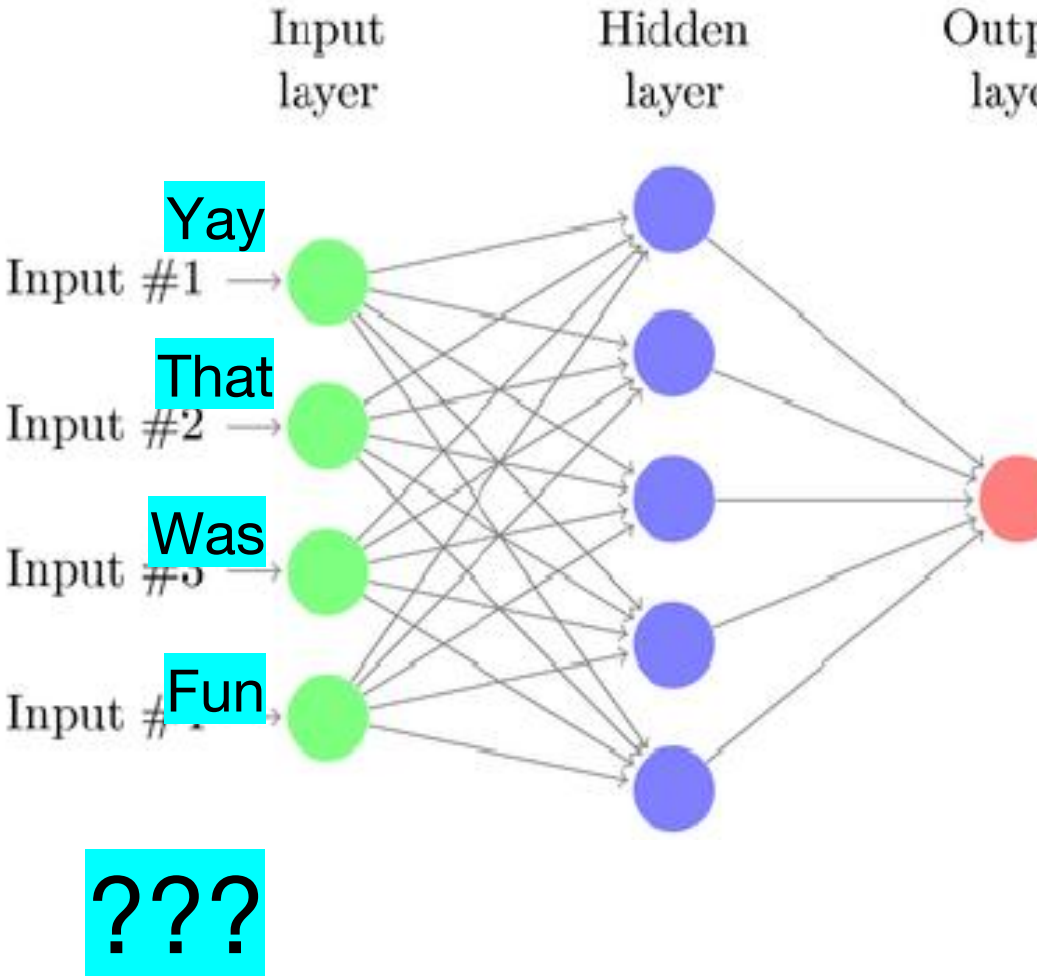


Tasks

- But sentences can vary in length.

- What do we do if we get:

  “I must declare that this particular experience proved to be an absolutely delightful and thoroughly entertaining endeavor that filled me with pure joy!”



RNNs

- This is what RNNs can do: deal with dynamic inputs and outputs

Inventing the RNN

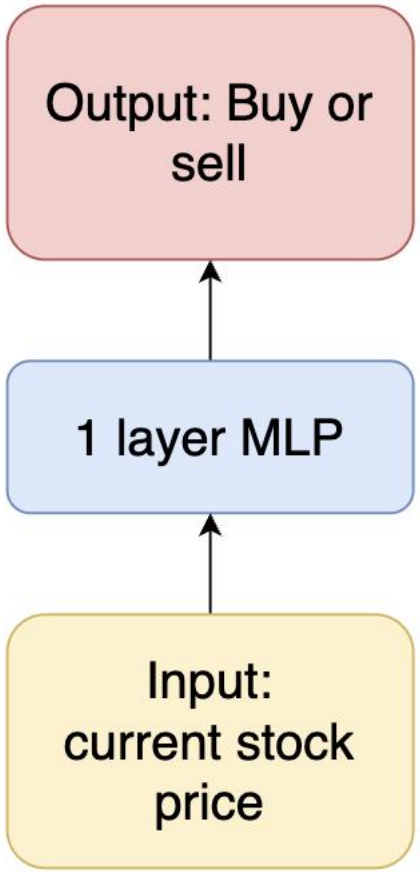
Suppose you want to get rich by buying stock:

For each day, you want to predict to buy or sell.

We are going to oversimplify all the stock things in this example.

Inventing the RNN

Using what we have, we start with this network:



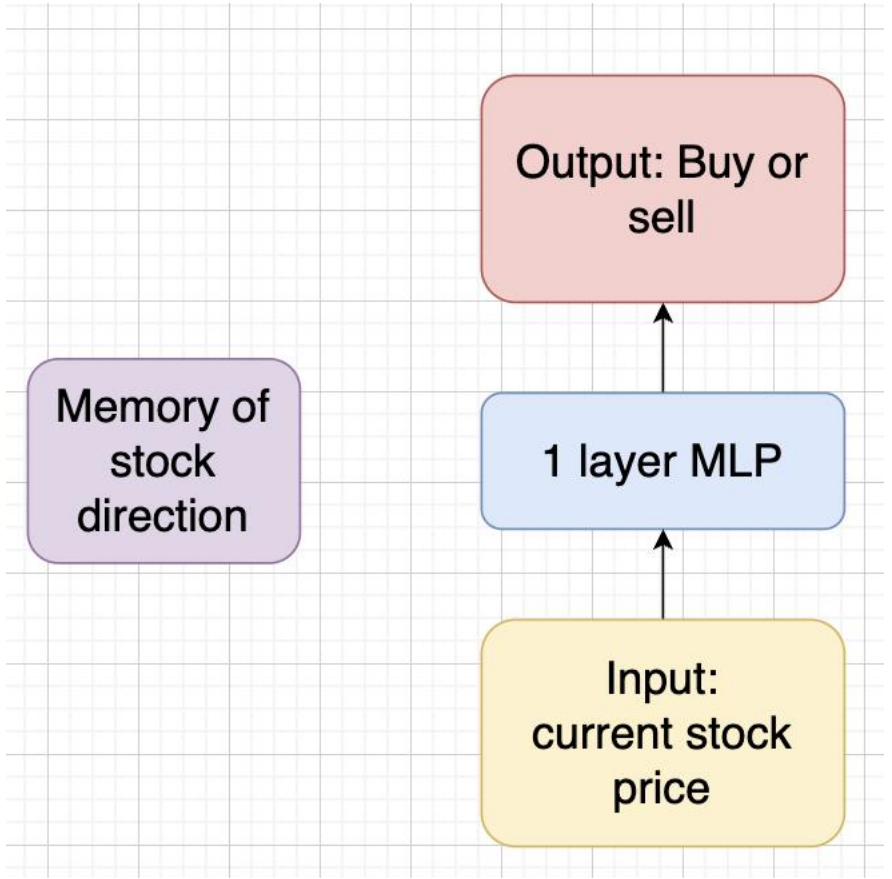
Inventing the RNN

But of course it does a bad job! Consider these two cases:

The model gives the same prediction for the two spots, even though one is clearly growing and one's falling!

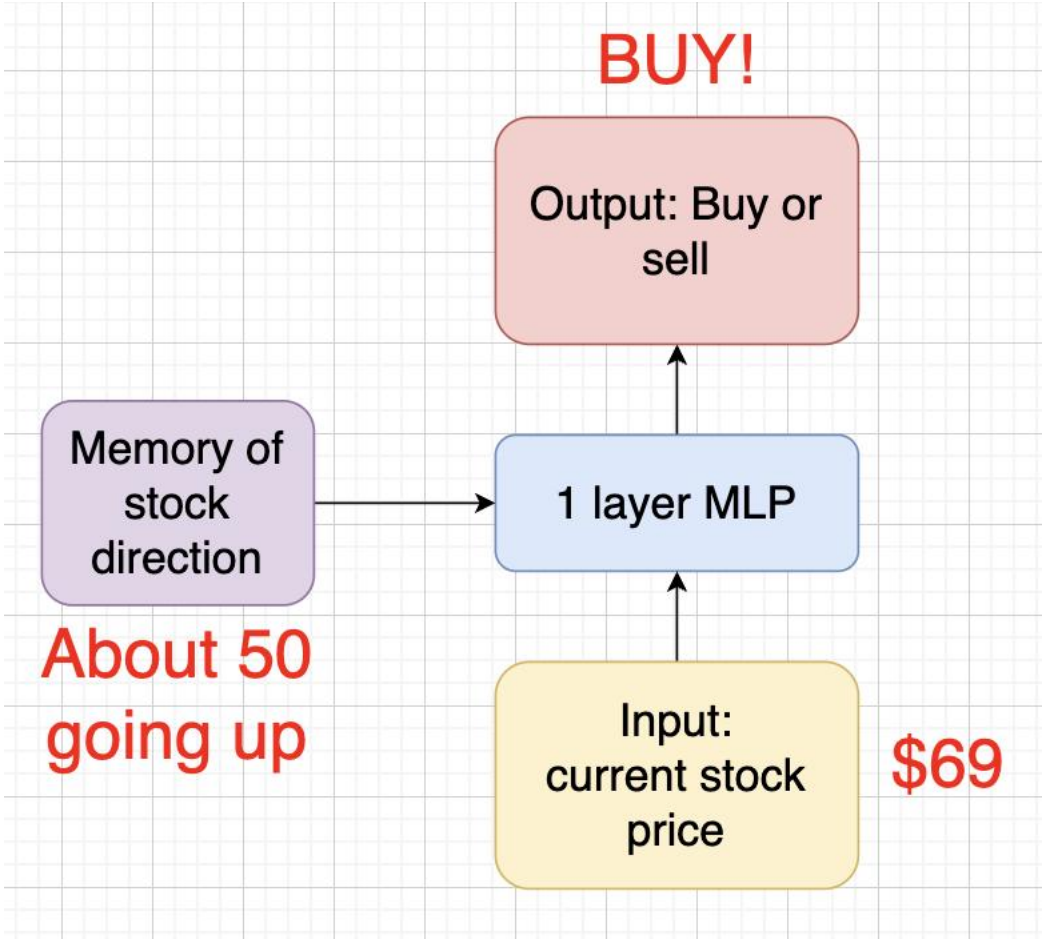
Inventing the RNN

What we need is some kind of “memory”, so the model can remember previous stock statuses.



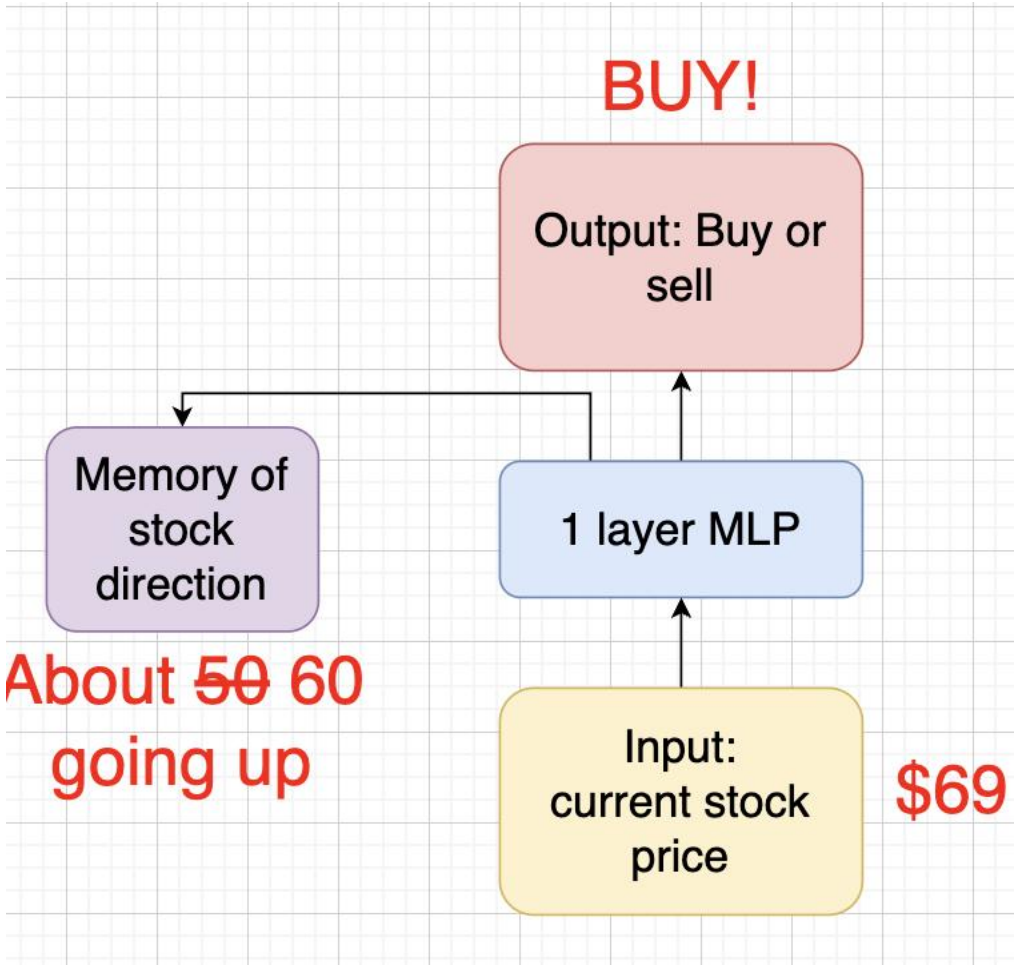
Inventing the RNN

For each prediction, we also take in the memory apart from the current stock price.



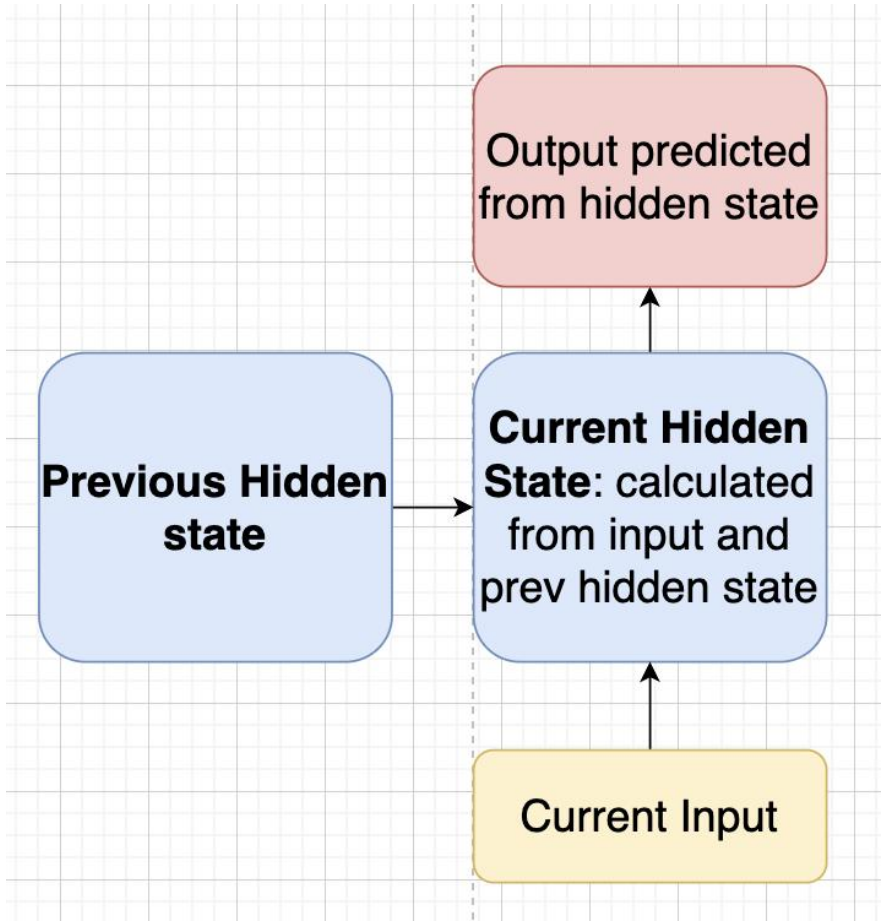
Inventing the RNN

After each prediction, we then take the intermediate calculations and update the memory:



Inventing the RNN

In the RNN, the memory part is known as the **Hidden State**



Inventing the RNN: Game of telephone

- We will play a game to help us understand how RNNs work.
- Get into groups of 6, your task is to guess the animal described by a sentence from a drawing you all drew together.
- Each person would get their assigned word, and when its your turn, use the drawing your previous teammate drew and add in whatever you can to best add in the meaning of your word to the drawing. Be creative and do not write down words!
- The last person in your group will guess the animal based on your drawings.

Inventing the RNN: Game of telephone

- In the game, the drawings you drew are the “Hidden states” of the human-rnn we made.

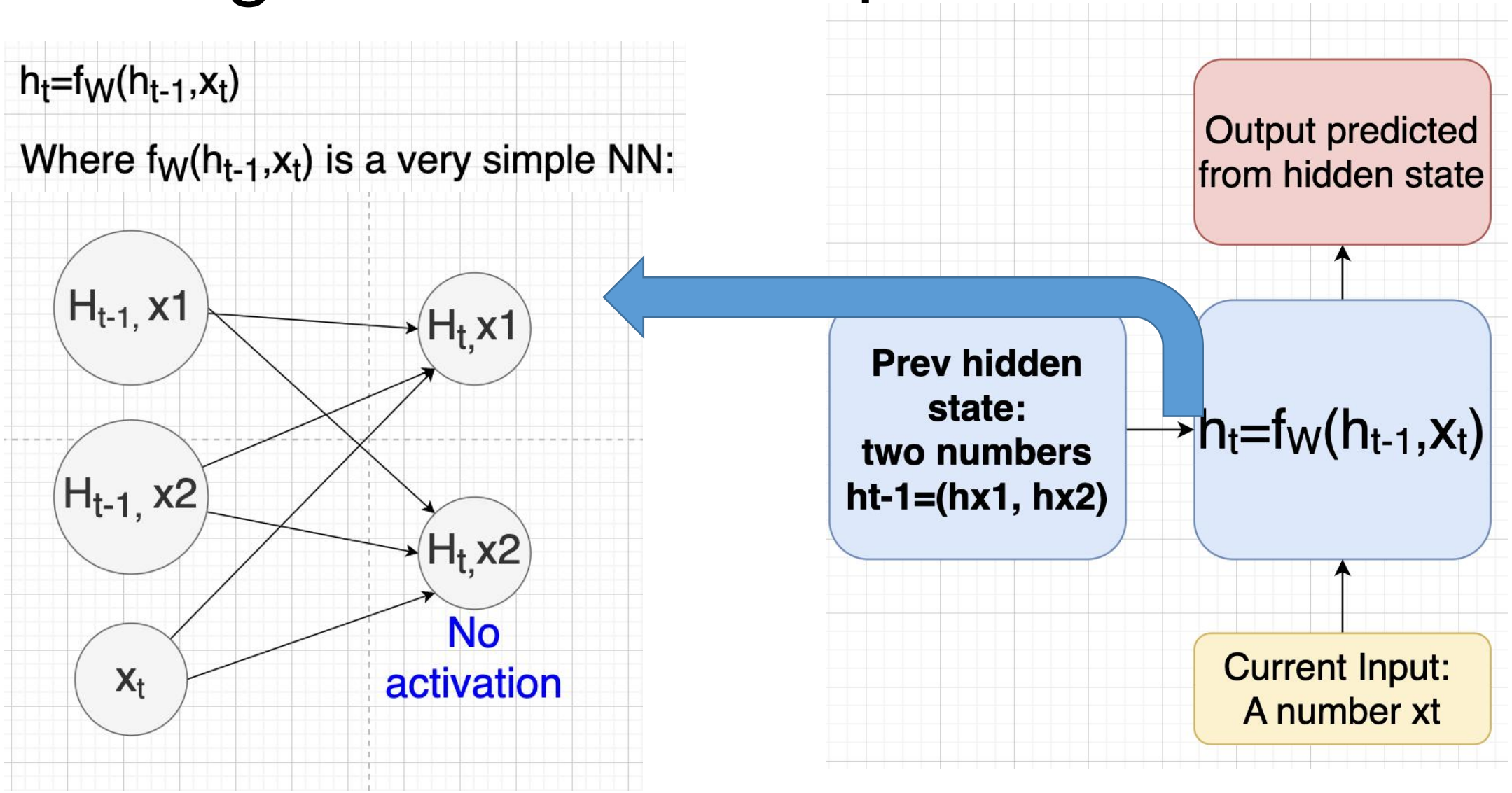
- Note that unlike in the game where each word is processed by a different person, in RNNs each processing part for each element is the **same network.** So the RNN basically plays this game for each word inputted over and over again with itself (it has no friends to play with :( ).

Inventing the RNN: example with math

- To demonstrate this how RNNs work with a concrete example, lets consider the stock example again.

- We will see how an RNN can effectively predict when to buy or sell stocks.

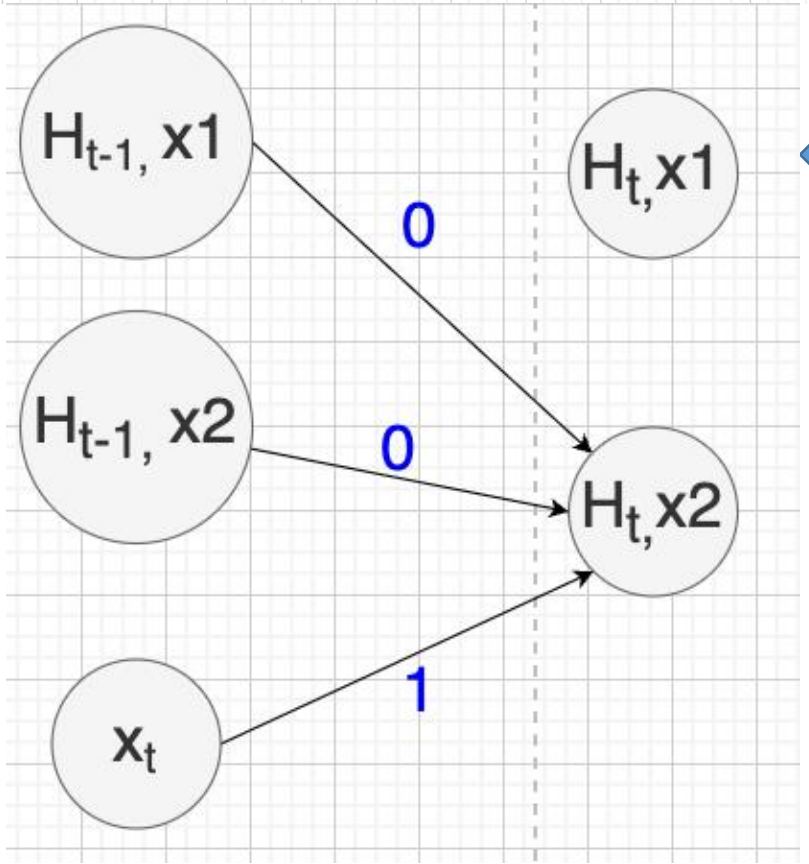
Inventing the RNN: example with math

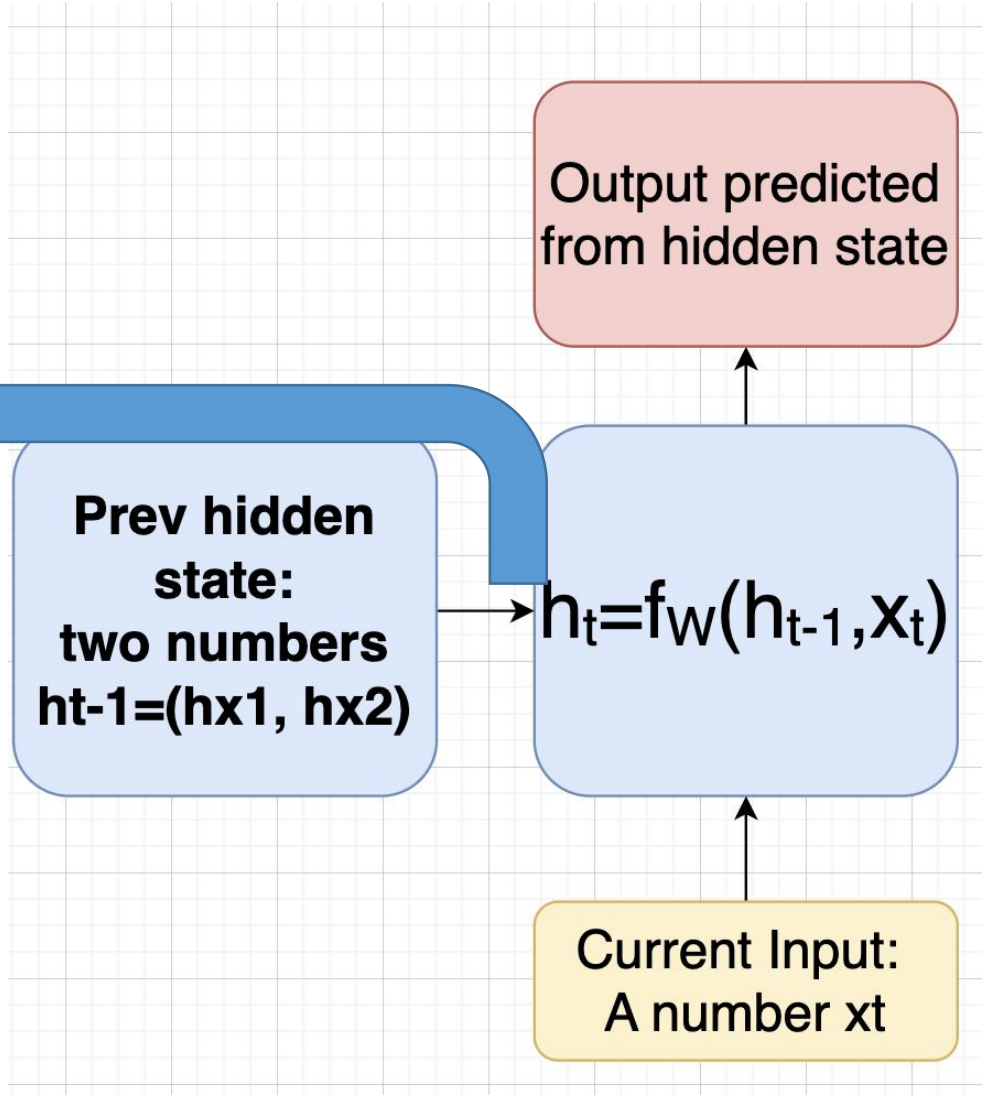


Inventing the RNN: example with math

$h_t=f_W(h_{t-1},x_t)$

Where $f_W(h_{t-1},x_t)$ is a very simple NN:

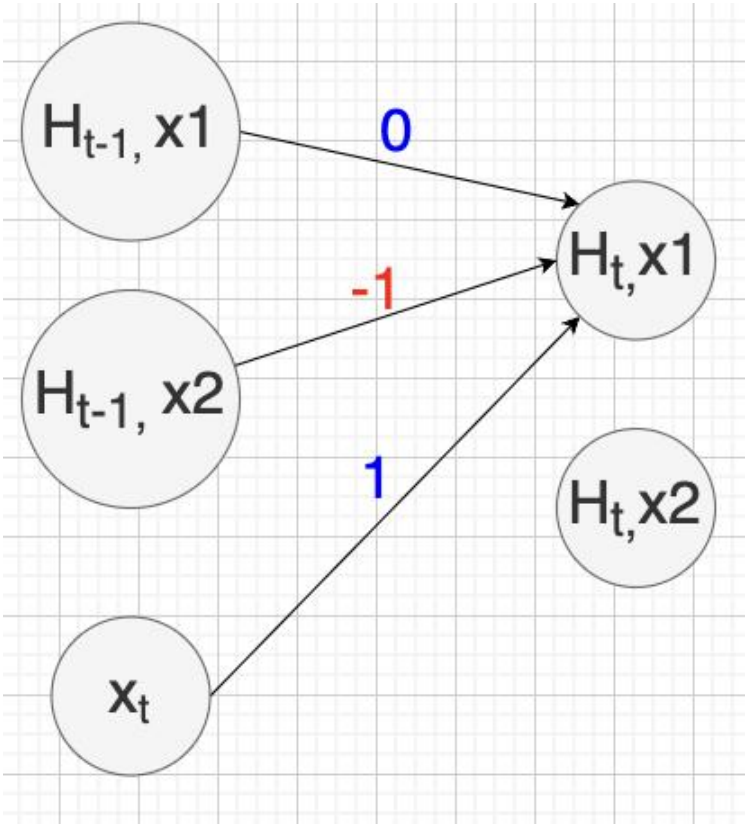


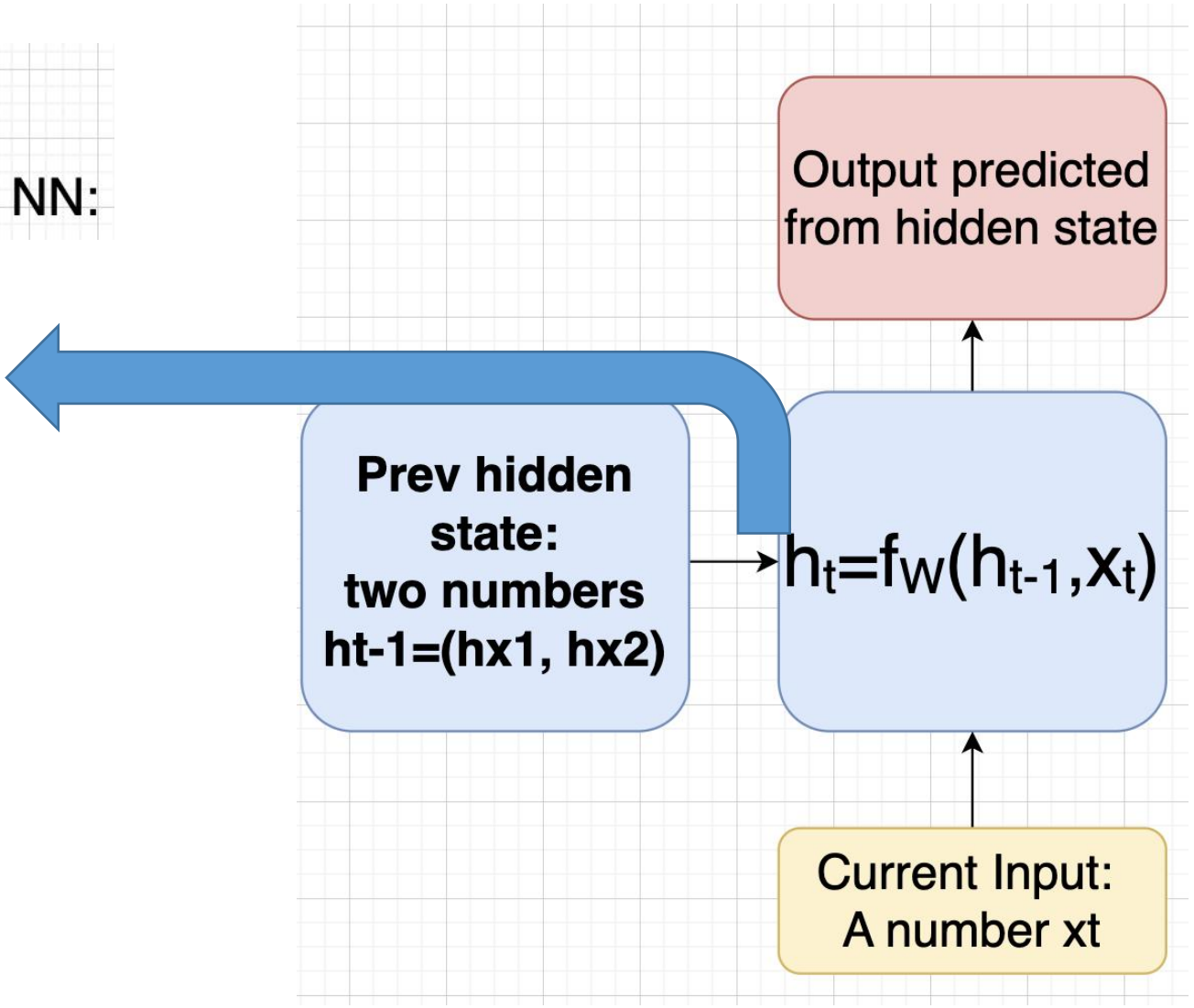


Inventing the RNN: example with math

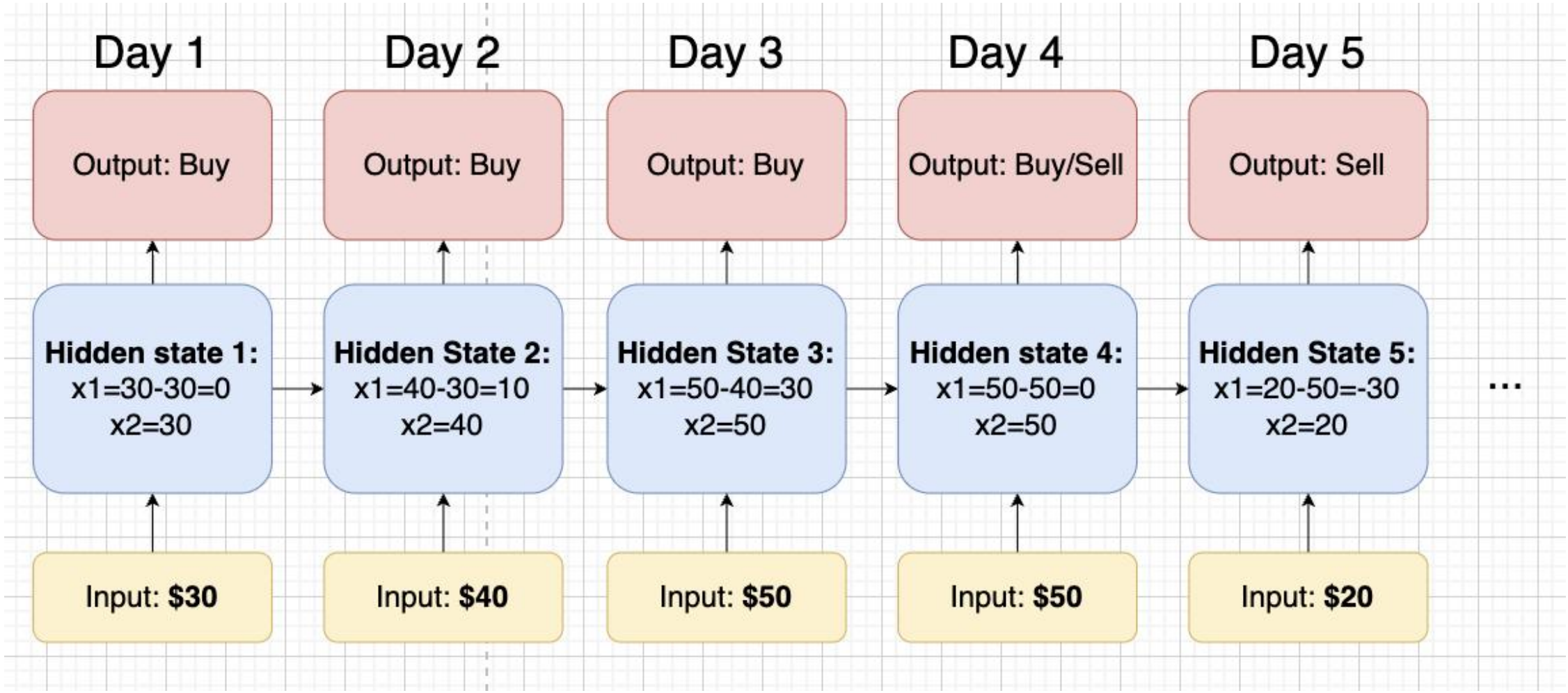
$h_t = f_W(h_{t-1}, x_t)$

Where $f_W(h_{t-1}, x_t)$ is a very simple NN:





Inventing the RNN: example with math



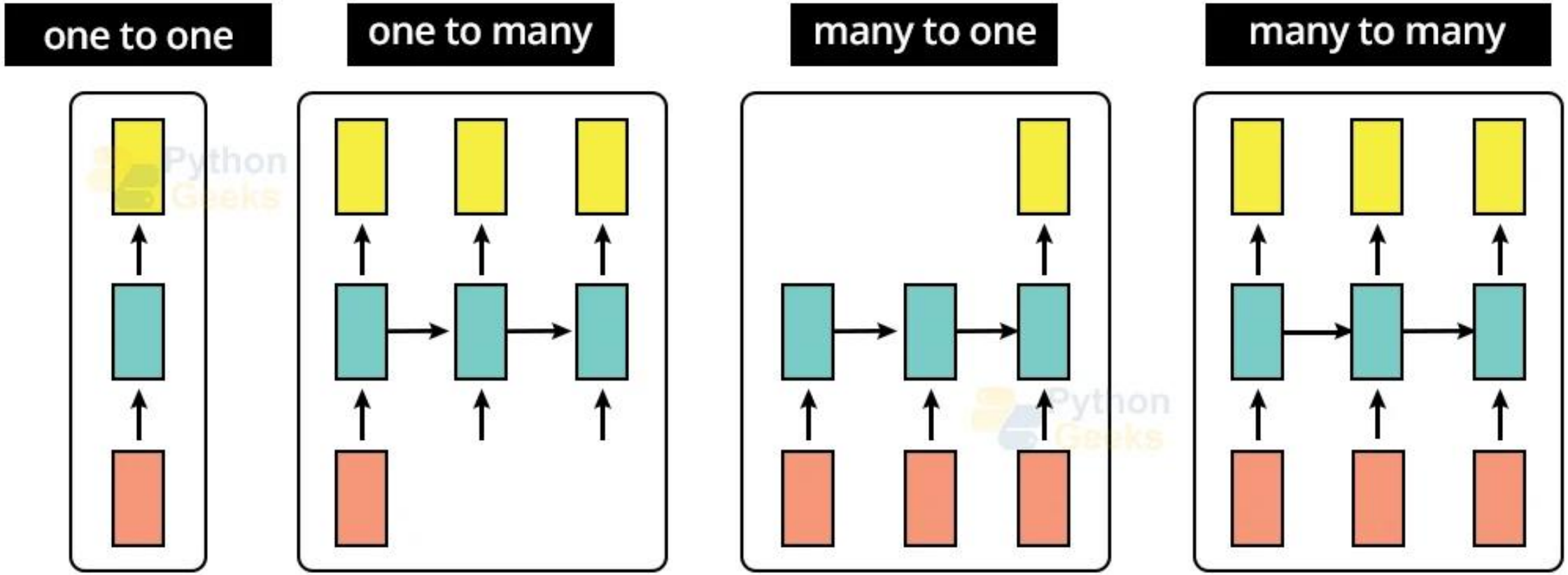
Inventing the RNN: example with math

- Now we go back to the hidden states. Can you try to see what hx1, hx2 are?

- hx1: difference between previous day and today
- hx2: current day price

- In practice, the hidden states can be anything and any size (how many numbers), and they are trained to represent different things (often way more complex than the example given).

Different types of RNNs



source: Python Geeks

Different types of RNNs: One to One

- Just your normal MLP or CNN

source: Python Geeks

Different types of RNNs: One to Many

- Takes in a fixed input, outputs a sequence of things
- Used for things like image captioning

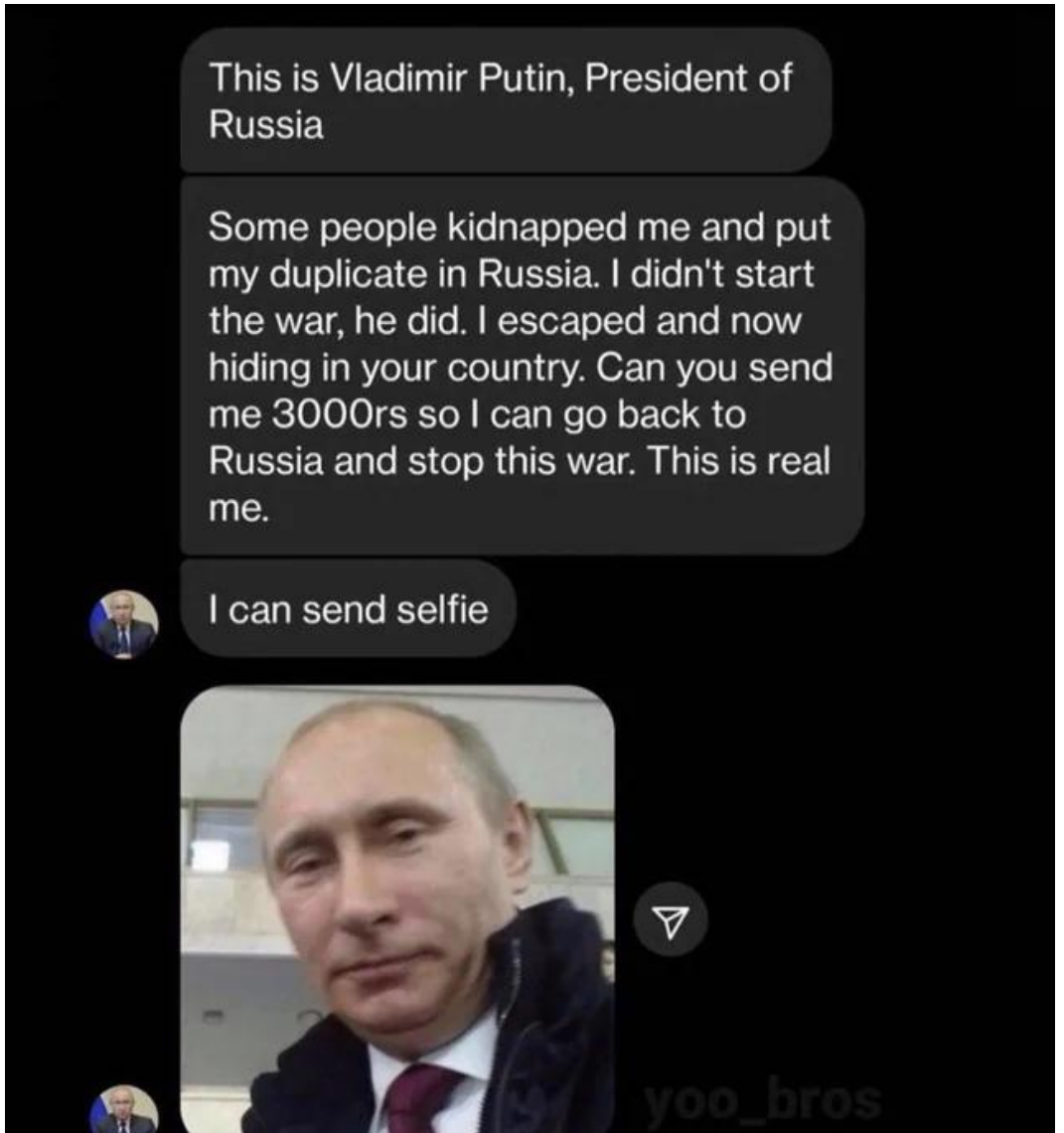
"Man in black shirt is playing guitar."

"Construction worker in orange safety vest is working on road."

"Two young girls are playing with lego toy."

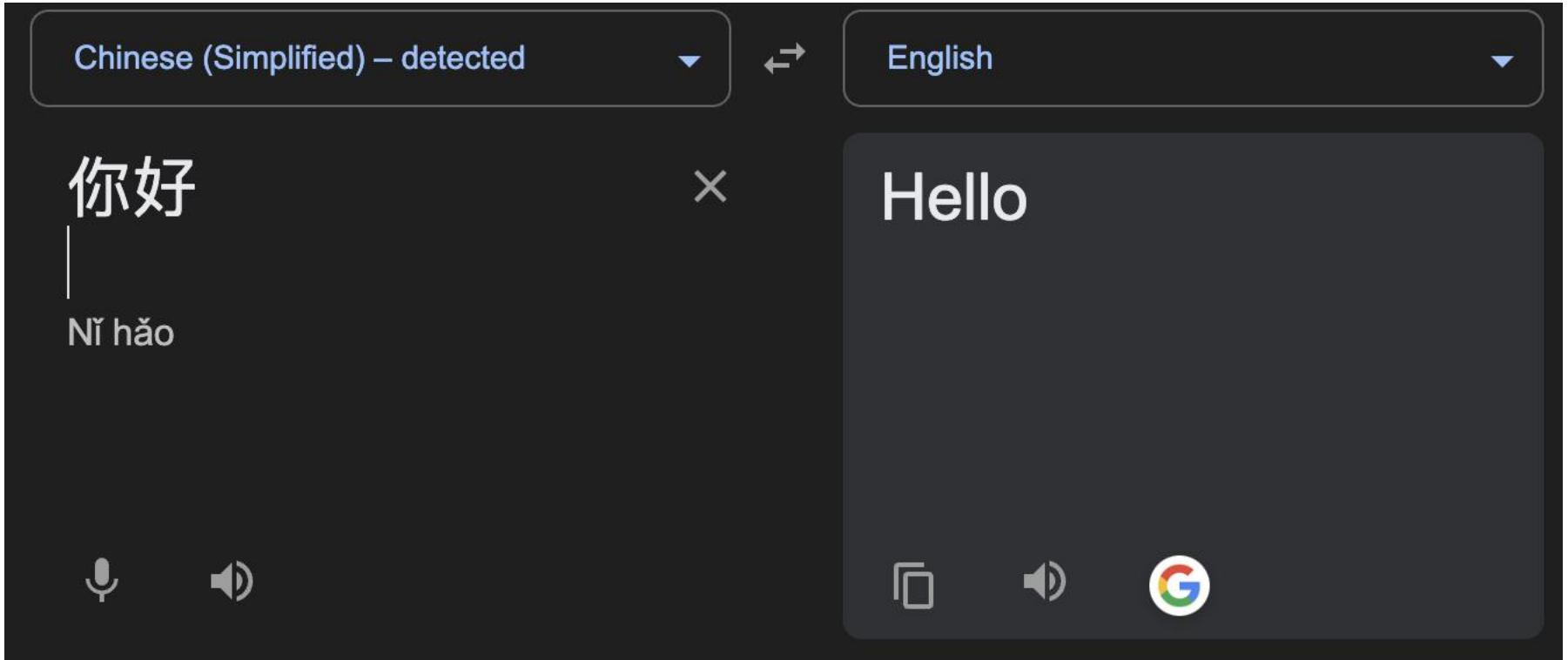
Different types of RNNs: Many to One

- Takes in a sequence of inputs, outputs one thing
- Application: Emotion detection from text, spam/scam detection



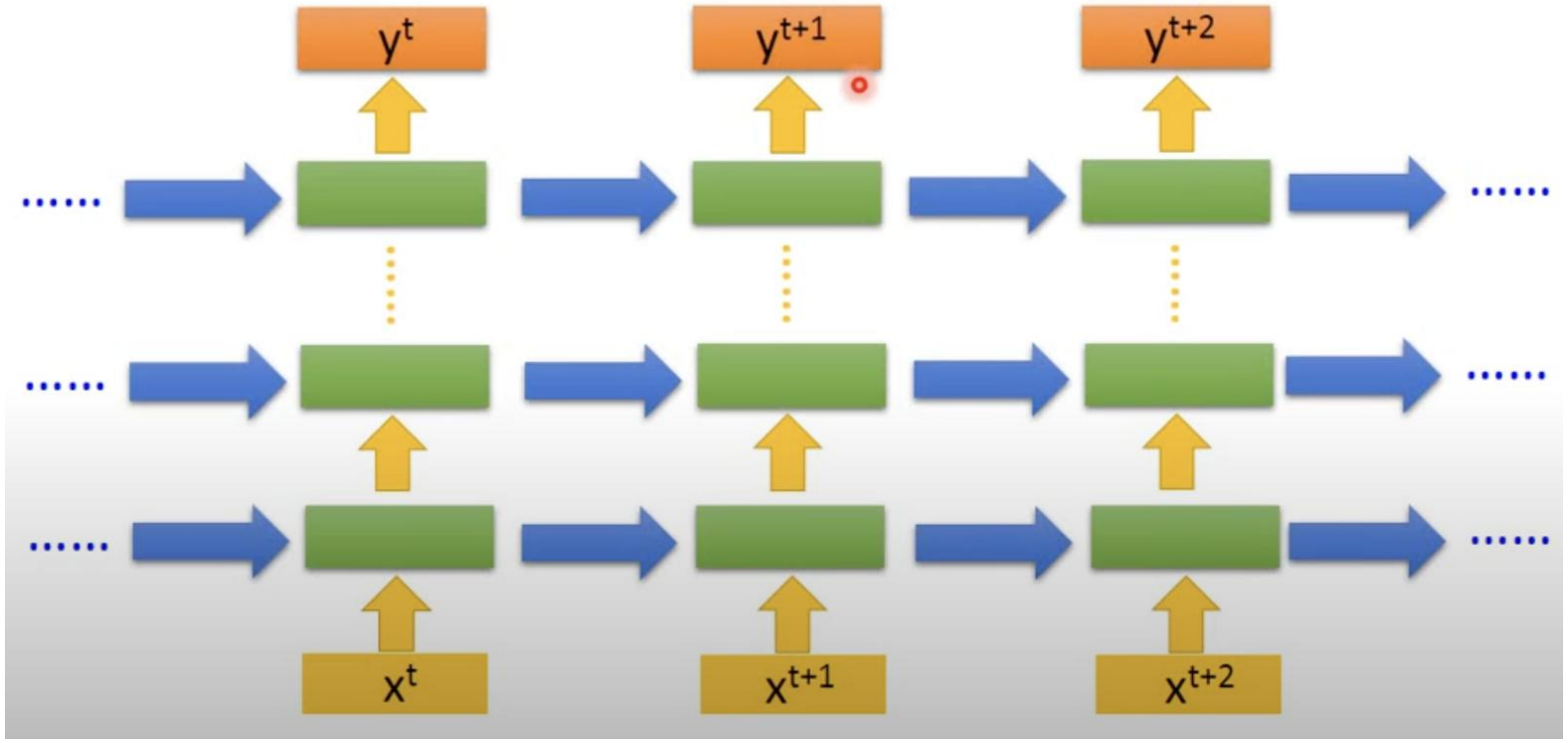
Different types of RNNs: Many to Many

- Inputs a sequence of many elements, outputs a sequence
- Example application: translation (though we do not use RNNs nowdays because transformers are way better)



Deeeeeep Networks!

- We can also have multiple layers of hidden states, just like how we can add hidden layers in MLPs:



source: https://www.youtube.com/watch?v=xCGidAeyS4M