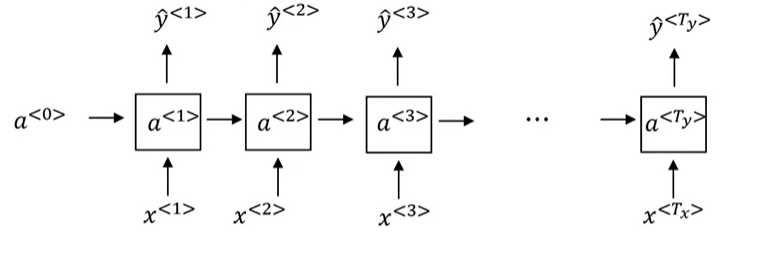
LSTM

Vanishing Gradient in RNN

*vanishing gradient problem*, an effect that is similar to what is observed with non-recurrent networks (feedforward networks) that are many layers deep: as you keep adding layers to a network, the network eventually becomes untrainable



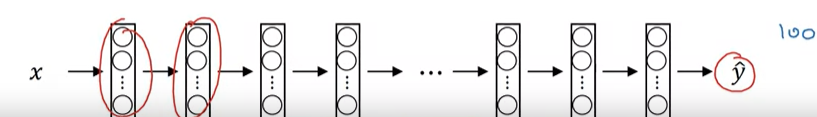
The dog ate bone ----------- was full

The dogs ate bone ----------------- were full

Singular

Plural

Long term dependencies are not so reliable in remembering states due to vanishing gradients



LSTM

a token’s *effect* is almost completely lost by the time two tokens have passed.[[1](https://learning.oreilly.com/library/view/natural-language-processing/9781617294631/kindle_split_020.html#ch09fn01)] Any effect the first node has on the third node (two time steps after the first time step) will be thoroughly stepped on by new data introduced in the intervening time step. This is important to the basic structure of the net, but it prevents the common case in human language that the tokens may be deeply interrelated even when they’re far apart in a sentence.

Example

The young woman went to the movies with her friends.

The young woman, having found a free ticket on the ground, went to the movies.

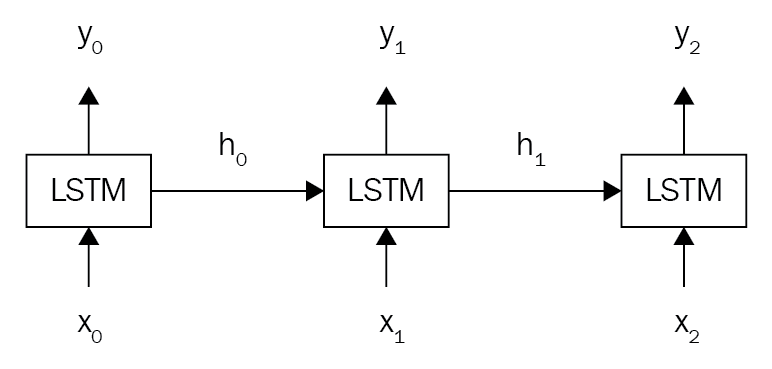
Example

Archie lived in China for 13 years. He loves listening to good music. He is a fan of comics. He is fluent in \_\_\_\_.

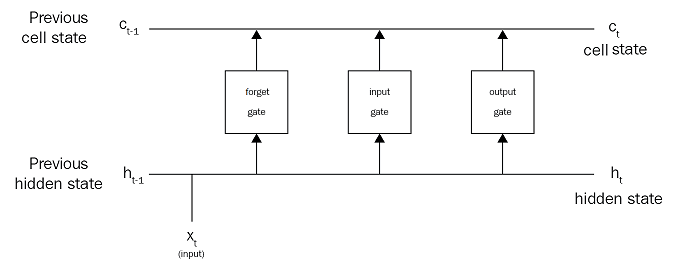
Now, if we were asked to predict the missing word in the preceding sentence, we would predict it as Chinese, but how did we predict that? We simply remembered the previous sentences and understood that Archie lived for 13 years in China. This led us to the conclusion that Archie might be fluent in Chinese. An RNN, on the other hand, cannot retain all of this information in its memory to say that Archie is fluent in Chinese. Due to the vanishing gradient problem, it cannot recollect/remember information for a long time in its memory. That is, when the input sequence is long, the RNN memory (hidden state) cannot hold all the information. To alleviate this, we use an LSTM cell.

LSTMs introduce the concept of a state for each layer in the recurrent network. The state acts as its memory

LSTM is a variant of an RNN that resolves the vanishing gradient problem and retains information in the memory as long as it is required. Basically, RNN cells are replaced with LSTM cells in the hidden units, as shown in the following diagram:



special structures called **gates**. As shown in the following diagram, a typical LSTM cell consists of three special gates called the input gate, output gate, and forget gate:

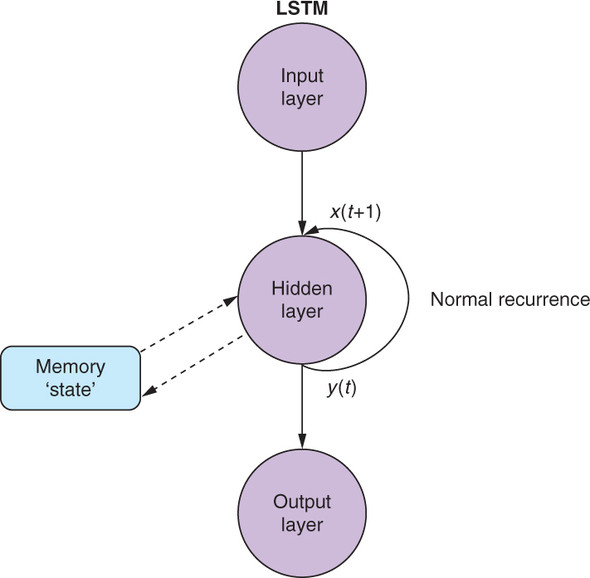


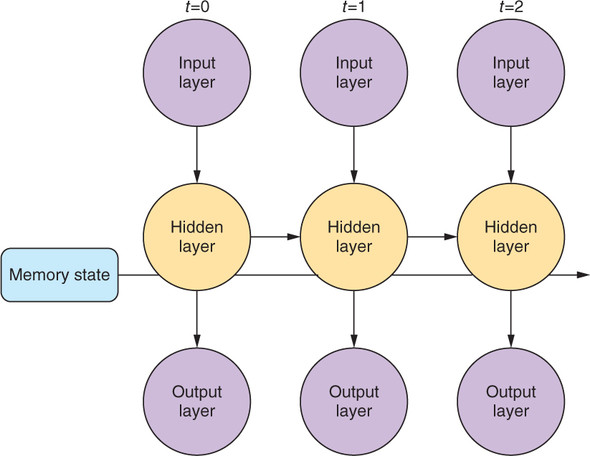
gates are responsible for deciding what information to add, output, and forget from the memory. With these gates, an LSTM cell effectively keeps information in the memory only as long as required.

In an RNN cell, we used the hidden state, , for two purposes: one for storing the information and the other for making predictions. Unlike RNN, in the LSTM cell we break the hidden states into two states, called the **cell state** and the **hidden state**:

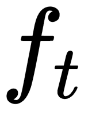
* The cell state is also called internal memory and is where all the information will be stored
* The hidden state is used for computing the output, that is, for making predictions

Both the cell state and hidden state are shared across every time step



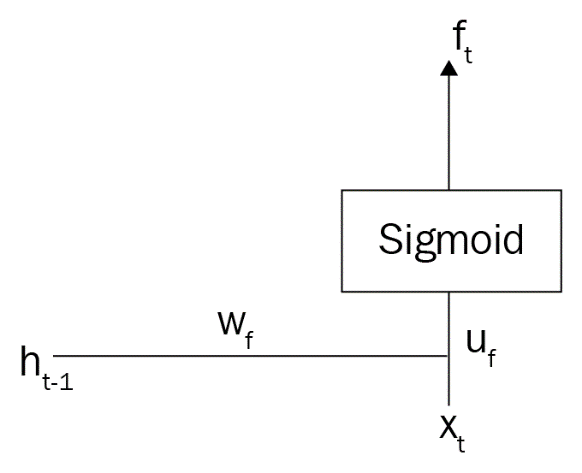


Gate Examples

The forget gate, , is responsible for deciding what information should be removed from the cell state (memory). Consider the following sentence:

*Harry* *is a good singer*. *He lives in New York. Zayn is also a good singer*.

As soon as we start talking about Zayn, the network will understand that the subject has been changed from Harry to Zayn, and the information about Harry is no longer required. Now, the forget gate will remove/forget information about Harry from the cell state.



Input gate

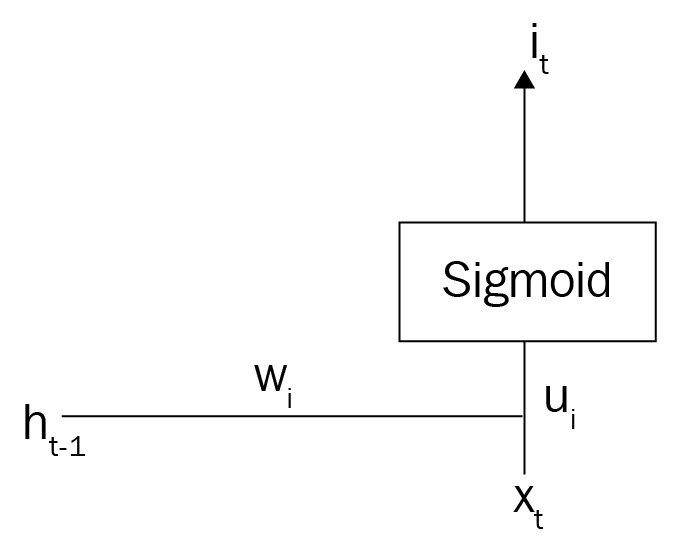
The input gate is responsible for deciding what information should be stored in the cell state. Let's consider the same example:



*Harry is a good singer. He lives in New York. Zayn is also a good singer.*

After the forget gate removes information from the cell state, the input gate decides what information it has to keep in the memory. Here, since the information about Harry is removed from the cell state by the forget gate, the input gate decides to update the cell state with information about *Zayn*.





Output gate

The output gate is responsible for deciding what information should be taken from the cell state to give as an output. Consider the following sentence:

*Zayn's debut album was a huge success. Congrats \_\_\_\_*.

The output gate will look up all the information in the cell state and select the correct information to fill the blank. Here, congrats is an adjective that is used to describe a noun. So, the output gate will predict *Zayn* (noun) to fill the blank. Similar to other gates, it is also controlled by a sigmoid function.

