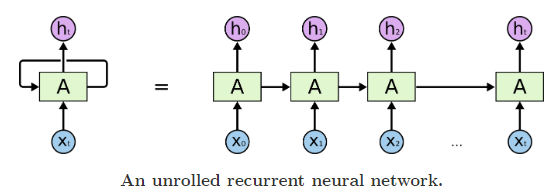
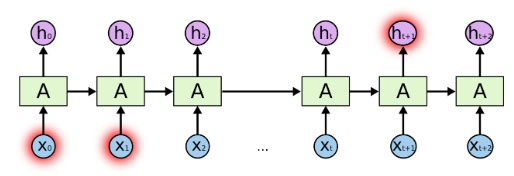
# **LSTM Networks** -Christopher Olah’s blog

**RNNs:**

* Networks with loops, allow information to persist   
  
* Chain like nature
* A chunk of neural network, A, looks at some input and outputs a value .
* A loop allows information to be passed from one step of the network to the next.

**The problem of long-term dependencies**

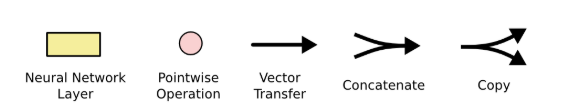
* Where the gap between the relevant information and the place that it’s needed is small, RNNs can learn to use past information
* If gap between relevant information & the point where its needed is large, RNNs become unable to connect the information   
  

**LSTM networks**

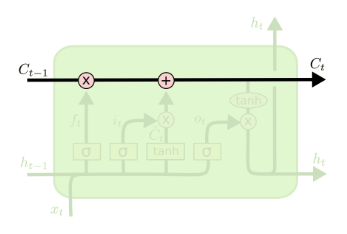
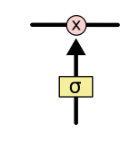
* Long Short Term Memory networks are special type of RNN capable of learning long-term dependencies
* Introduced in 1997 by Sepp Hochreiter & Jurgen Schmidhuber

|  |  |
| --- | --- |
| **RNN** | **LSTM** |
| Form of a chain of repeating modules of NN | Also have chain like structure |
| This repeating module has very simple structure,  like a single tanh layer | Repeating module has four NN layers, interacting  in a very special way |
|  |  |

**Notation**

****

**LSTM Overview**

* **Cell state:** Horizontal line running through the top of the diagram, running straight down the entire chain, with only some minor linear interactions. It’s very easy for information to just flow along it unchanged.  
  ****
* **Gates:** Provide theability to remove or add information to the cell state, by optionally letting information through. They are composed out of a sigmoid neural net layer and a pointwise multiplication operation.  
  ****

Sigmoid layer outputs numbers between 0 and 1, describing how much of each component should be let through.

0 means “let nothing through”

1 means “let everything through!”

An LSTM has three of these gates:

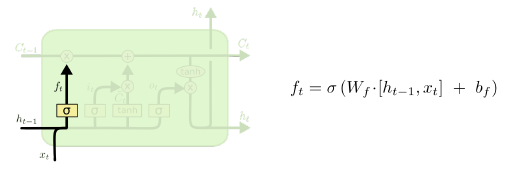
1. Forget gate
2. Input gate
3. Output gate

**Step by step LSTM walkthrough**

1. **Forget gate layer**: *What information to throw away from the cell state*
2. **Input gate layer (sigmoid layer + tanh layer)**: *What information to store in the cell state*
3. **Output gate layer (sigmoid layer + tanh layer)**: *What information to output*

**Step 1.Layer 1: Forget gate layer**

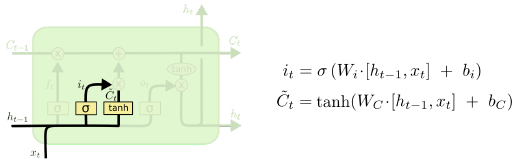
Decide what information we’re going to throw away from the cell state. Decided by a **sigmoid** layer.  
Looks at and and outputs a number between 0 and 1 for each number in the cell state .



**Step 2. Layer 2: Input gate layer**

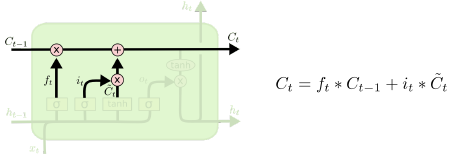
Decides what new information we’re going to store in the cell state.

1. **Sigmoid layer**: Input gate layer. Decides which values we’ll update.
2. **tanh layer**: Creates vector of new candidate values that could be added to the state

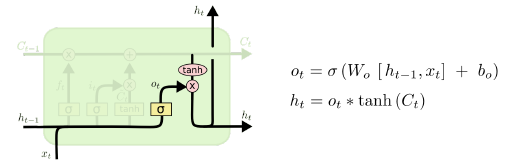
Then we combine these two to create an update to the state.

**Step 3. Update the old cell state**

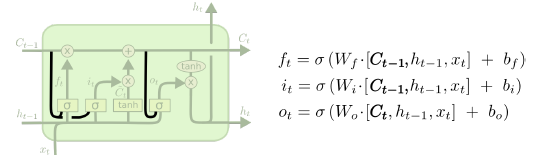
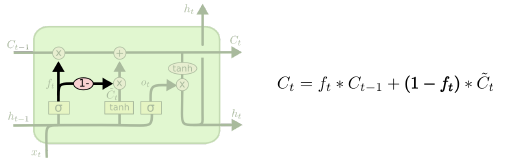
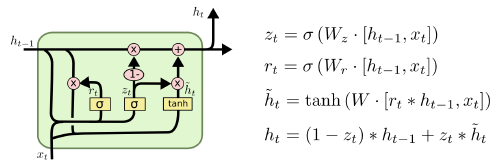
We multiply the old state by , forgetting the things we want to forget earlier.

Then add , which is the new candidate values, scaled by how much we decided to update each state value.  


**Step 4: Layer 3: Output gate layer**This gate will be based on our cell state, but will be a filtered version.

1. **Sigmoid layer**: Decides what part of the cell state we’re going to output
2. **tanh layer**: Push the values to be between -1 & 1
3. Multiply a & b, so that we only output the parts we decided to output

**Variants to LSTM**

1. **Adding peephole connections**   
   We let the gate layers look at the cell state  
   ****
2. **Use coupled forget & input gates**Instead of separately deciding what to forget and what we should add new information to, we make those decisions together. We only forget when we’re going to input something in its place. We only input new values to the state when we forget something older.  
   
3. **Gated Recurrent Unit (GRU)**Combines the forge and input gates into a single “update gate.”   
   It also merges the cell state and hidden state, and makes some other changes.   
   The resulting mode is simpler than standard LSTM models, and has been growingincreasingly popular.  
   ****