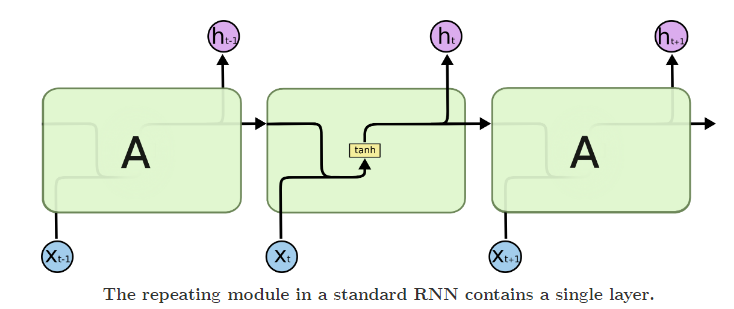
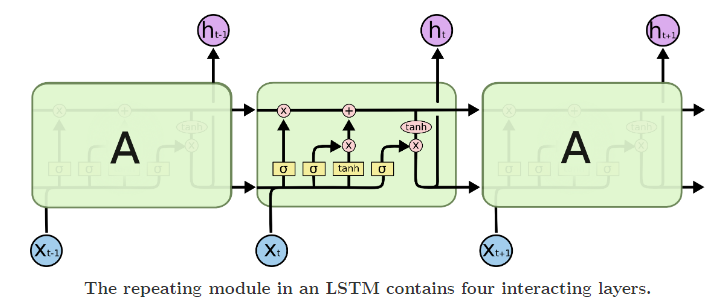
**LSTM**

Long Short Term Memory networks – usually just called “LSTMs” – are a special kind of RNN, capable of learning long-term dependencies. They were introduced by Hochreiter & Schmidhuber (1997), and were refined and popularized by many people in following work.1 They work tremendously well on a large variety of problems, and are now widely used.

LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn!

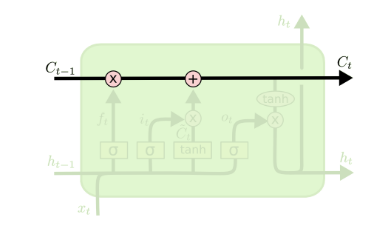
All recurrent neural networks have the form of a chain of repeating modules of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer.



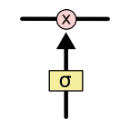
LSTMs also have this chain like structure, but the repeating module has a different structure. Instead of having a single neural network layer, there are four, interacting in a very special way. 

***The Core Idea Behind LSTMs***

The key to LSTMs is the cell state, the horizontal line running through the top of the diagram.

The cell state is kind of like a conveyor belt. It runs straight down the entire chain, with only some minor linear interactions. It’s very easy for information to just flow along it unchanged.

The LSTM does have the ability to remove or add information to the cell state, carefully regulated by structures called gates.

Gates are a way to optionally let information through. They are composed out of a sigmoid neural net layer and a pointwise multiplication operation.

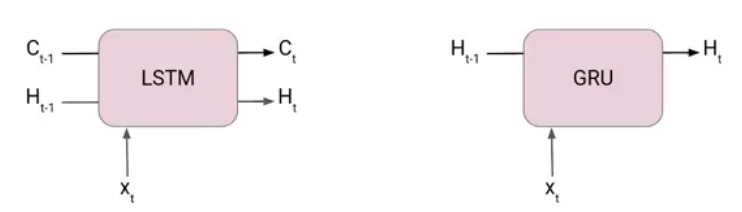
The sigmoid layer outputs numbers between zero and one, describing how much of each component should be let through. A value of zero means “let nothing through,” while a value of one means “let everything through!”

An LSTM has three of these gates, to protect and control the cell state.

**GRU**

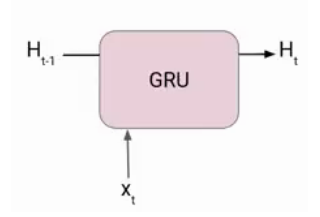
GRU or Gated recurrent unit is an advancement of the standard RNN i.e recurrent neural network. It was introduced by Kyunghyun Cho et al in the year 2014.

GRUs are very similar to Long Short Term Memory(LSTM). Just like LSTM, GRU uses gates to control the flow of information. They are relatively new as compared to LSTM. This is the reason they offer some improvement over LSTM and have simpler architecture.



Another Interesting thing about GRU is that, unlike LSTM, it does not have a separate cell state (Ct). It only has a hidden state(Ht). Due to the simpler architecture, GRUs are faster to train.

***The architecture of Gated Recurrent Unit***

Here we have a GRU cell which more or less similar to an LSTM cell or RNN cell.

At each timestamp t, it takes an input Xt and the hidden state Ht-1 from the previous timestamp t-1. Later it outputs a new hidden state Ht which again passed to the next timestamp.

Now there are primarily two gates in a GRU as opposed to three gates in an LSTM cell. The first gate is the Reset gate and the other one is the update gate.

Reset Gate (Short term memory)

The Reset Gate is responsible for the short-term memory of the network i.e the hidden state (Ht). Here is the equation of the Reset gate.



The value of rt will range from 0 to 1 because of the sigmoid function. Here Ur and Wr are weight matrices for the reset gate.

Update Gate (Long Term memory)

Similarly, we have an Update gate for long-term memory and the equation of the gate is shown below.



The only difference is of weight metrics i.e Uu and Wu.

***How GRU Works***

To find the Hidden state Ht in GRU, it follows a two-step process. The first step is to generate what is known as the candidate hidden state. As shown below

Candidate Hidden State



It takes in the input and the hidden state from the previous timestamp t-1 which is multiplied by the reset gate output rt. Later passed this entire information to the tanh function, the resultant value is the candidate’s hidden state.



The most important part of this equation is how we are using the value of the reset gate to control how much influence the previous hidden state can have on the candidate state.

If the value of rt is equal to 1 then it means the entire information from the previous hidden state Ht-1 is being considered. Likewise, if the value of rt is 0 then that means the information from the previous hidden state is completely ignored.

Hidden state

Once we have the candidate state, it is used to generate the current hidden state Ht. It is where the Update gate comes into the picture. Now, this is a very interesting equation, instead of using a separate gate like in LSTM in GRU we use a single update gate to control both the historical information which is Ht-1 as well as the new information which comes from the candidate state.



Now assume the value of ut is around 0 then the first term in the equation will vanish which means the new hidden state will not have much information from the previous hidden state. On the other hand, the second part becomes almost one that essentially means the hidden state at the current timestamp will consist of the information from the candidate state only.



Similarly, if the value of ut is on the second term will become entirely 0 and the current hidden state will entirely depend on the first term i.e the information from the hidden state at the previous timestamp t-1.



Hence we can conclude that the value of ut is very critical in this equation and it can range from 0 to 1.

Compare LSTM vs GRU vs RNN

|  |  |  |  |
| --- | --- | --- | --- |
|  | RNN | LSTM | GRU |
| Structure | Simple | More complex | Simpler than LSTM |
| Training | Can be difficult | Can be more difficult | Easier than LSTM |
| Performance | Good for simple tasks | Good for complex tasks | Can be intermediate between simple and complex tasks |
| Hidden state | Single | Multiple (memory cell) | Single |
| Gates | None | Input, output, forget | Update, reset |
| Ability to retain long-term dependencies | Limited | Strong | Intermediate between RNNs and LSTMs |

Summarizing the Difference Between RNN vs LSTM vs GRU

RNNs are a type of neural network that are designed to process sequential data, such as text, audio, or time series data. They can “remember” or store information from previous inputs, which allows them to use context and dependencies between time steps.

LSTMs are a type of RNN that use special type of memory cell and gates to store and output information. The gates in an LSTM network are controlled by sigmoid activation functions. These gates allow the network to selectively store or forget information. LSTMs are effective at storing and accessing long-term dependencies. They are slower to train and run than other types of RNNs.

GRUs are simplified version of LSTMs that use single “update gate” to control the flow of information into the memory cell. GRUs are easier to train and faster to run than LSTMs, but they may not be as effective at storing and accessing long-term dependencies.

There is no one “best” type of RNN for all tasks, and the choice between LSTMs, GRUs, and other types of RNNs will depend on the specific requirements of the task at hand. It is often a good idea to try multiple types of RNNs and see which one performs best on your specific task.