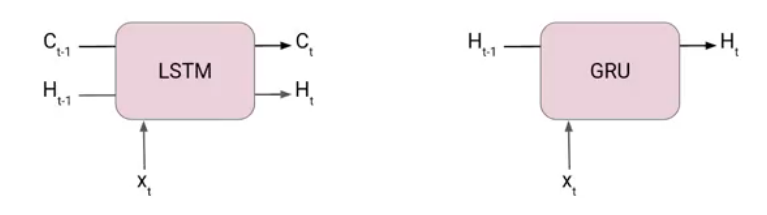
**Gated Recurrent Unit (GRU)**

* **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.**



## What does Gated Recurrent Unit (GRU) mean?

**A gated recurrent unit (GRU) is part of a specific model of recurrent neural network that intends to use connections through a sequence of nodes to perform machine learning tasks associated with memory and clustering, for instance, in speech recognition. Gated recurrent units help to adjust neural network input weights to solve the vanishing gradient problem that is a common issue with recurrent neural networks.**

**As a refinement of the general recurrent neural network structure, gated recurrent units have what's called an update gate and a reset gate. Using these two vectors, the model refines outputs by controlling the flow of information through the model. Like other kinds of recurrent network models, models with gated recurrent units can retain information over a period of time – that is why one of the simplest ways to describe these types of technologies is that they are a "memory-centered" type of neural network. By contrast, other types of neural networks without gated recurrent units often do not have the ability to retain information.**

**In addition to speech recognition, neural network models using gated recurrent units may be used for research on the human genome, handwriting analysis and much more. Some of these innovative networks are used in stock market analysis and government work. Many of them leverage the simulated ability of machines to remember information.**

## What is a Gated Recurrent Unit?

**A gated recurrent unit (GRU) is a gating mechanism in**[**recurrent neural networks**](https://deepai.org/machine-learning-glossary-and-terms/recurrent-neural-network)**(RNN) similar to a**[**long short-term memory**](https://deepai.org/machine-learning-glossary-and-terms/long-short-term-memory)**(LSTM) unit but without an output gate. GRU’s try to solve the**[**vanishing gradient problem**](https://deepai.org/machine-learning-glossary-and-terms/vanishing-gradient-problem)**that can come with standard recurrent**[**neural networks**](https://deepai.org/machine-learning-glossary-and-terms/neural-network)**. A GRU can be considered a variation of the long short-term memory (LSTM) unit because both have a similar design and produce equal results in some cases. GRU’s are able to solve the vanishing gradient problem by using an update gate and a reset gate. The update gate controls information that flows into memory, and the reset gate controls the information that flows out of memory. The update gate and reset gate are two**[**vectors**](https://deepai.org/machine-learning-glossary-and-terms/vector)**that decide which information will get passed on to the output. They can be trained to keep information from the past or remove information that is irrelevant to the prediction.**

### Why is this Useful?

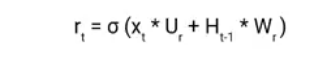
**A GRU is a very useful mechanism for fixing the vanishing gradient problem in recurrent neural networks. The vanishing gradient problem occurs in**[**machine learning**](https://deepai.org/machine-learning-glossary-and-terms/machine-learning)**when the gradient becomes vanishingly small, which prevents the weight from changing its value. They also have better performance than LSTM when dealing with smaller datasets.**

## The architecture of Gated Recurrent Unit

* **Now lets’ understand how GRU works. 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.**

****

**If you remember from the LSTM gate equation it is very similar to that. 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.**

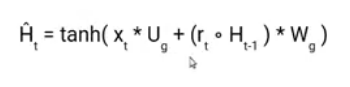
**Gated recurrent unit - Update Gate (Long Term memory)**

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

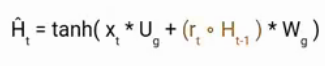
## How GRU Works

**Now let’s see the functioning of these gates. 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.**

**Hidden 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.**

**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.**

**timestamp t-1**

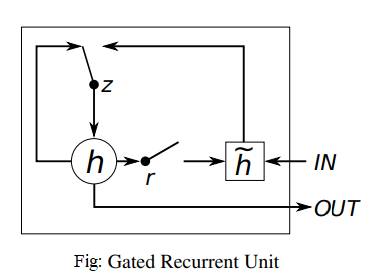
**Hence we can conclude that the value of ut is very critical in this equation and it can range from 0 to 1.**

## Applications of a Gated Recurrent Unit

* **Polyphonic music modeling**
* **Speech signal modeling**
* **Handwriting recognition**

## How It Works

**In GRU, two gates including a reset gate that adjusts the incorporation of new input with the previous memory and an update gate that controls the preservation of the precious memory are introduced. The reset gate and the update gate adaptively control how much each hidden unit remembers or forgets while reading/generating a sequence.**



**In the above figure of the Gated Recurrent Unit, *r* and *z* are known to be the reset and update gates, while *h*and *h˜* are the activations as well as the candidate activation respectively. The working of GRU proceeds such that when the reset gate is close to zero, the hidden state is forced to ignore the previous hidden state and is reset with the current input.**

**This allows the hidden state to discard any data that is found to be irrelevant in the future. This result allows a more compact representation. While the update gate controls how much data from the previous hidden state will be transferred to the current hidden state. This process performs in a similar manner to the memory cell in the Long Short-Term Memory network and helps the RNN to remember long-term information.**

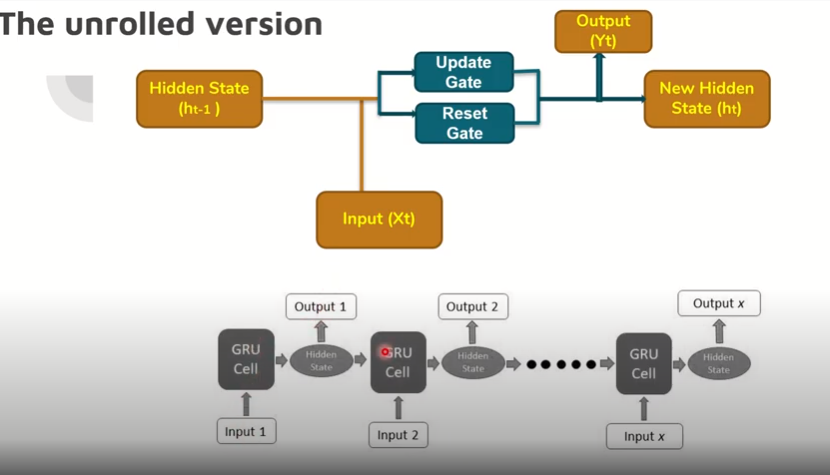
**The activation of the GRU at a particular time is a linear interpolation between the previous activation and the candidate activation, where an update gate decides how much the unit updates its activation or content.**

### C:\Users\dell\Desktop\a.jpg

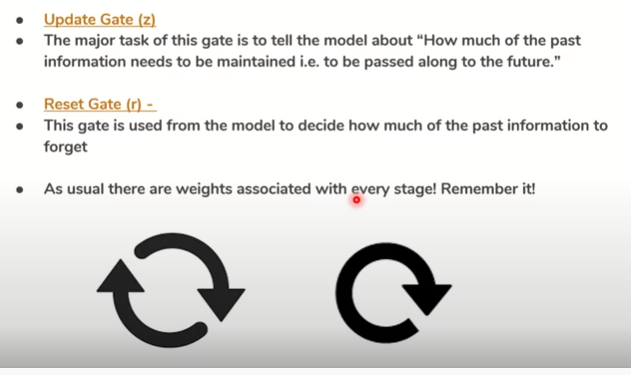
### C:\Users\dell\Desktop\qq.jpg

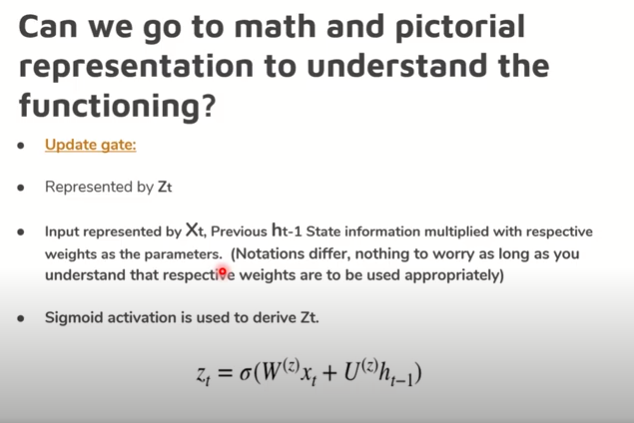
### C:\Users\dell\Desktop\2.jpg

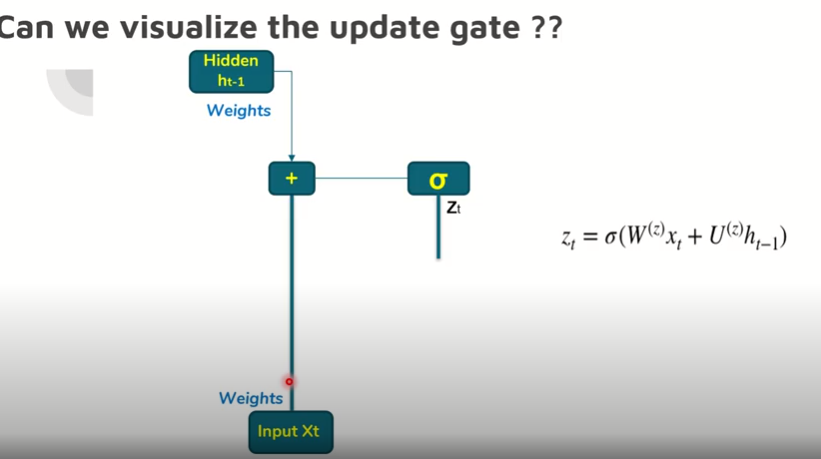
### The input state & hidden state together will get into the GRU & we have update gate & reset gate working which will generate output & new hidden state.

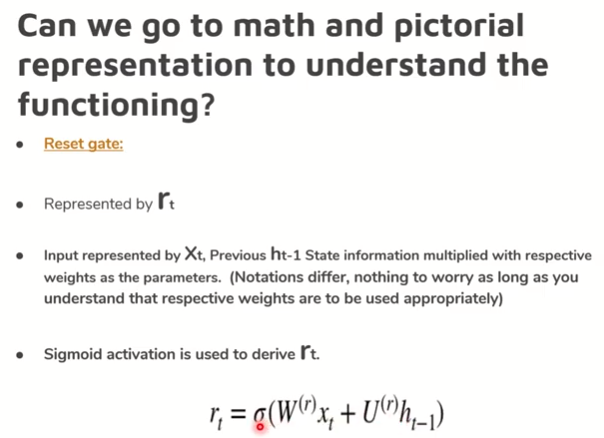


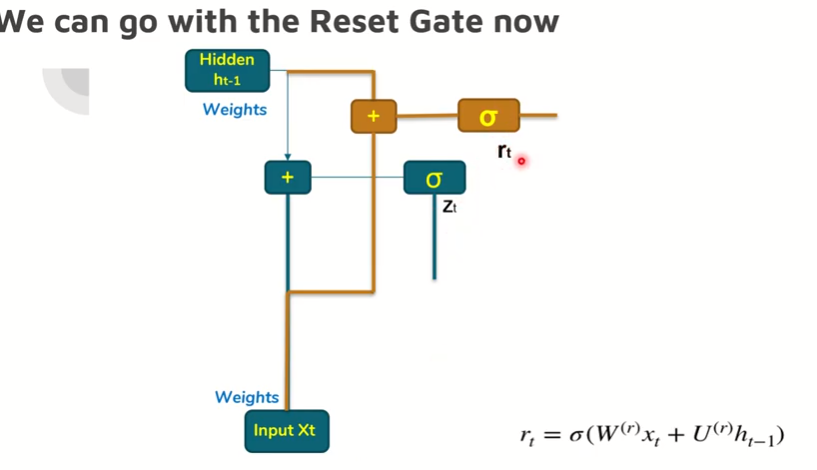
**During the first input don’t have a hidden state, so the input get into the GRU & we get the o/p, for the next GRU cell we have the hidden state as well as the i/p , here we generate new hidden state & output is coming keeps on going till the last state.**

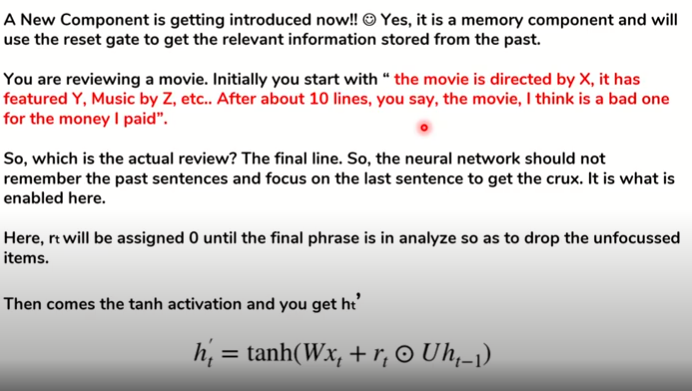


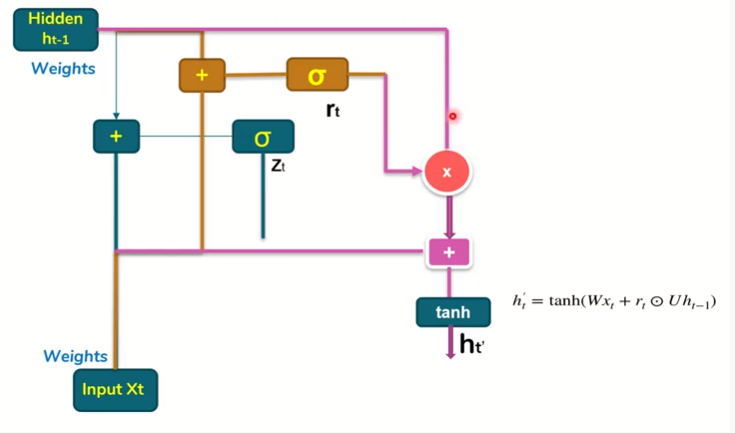


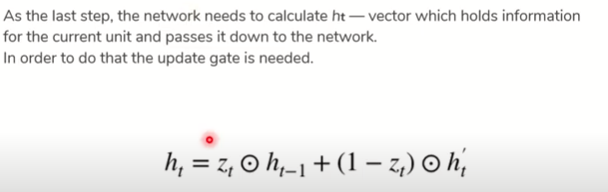


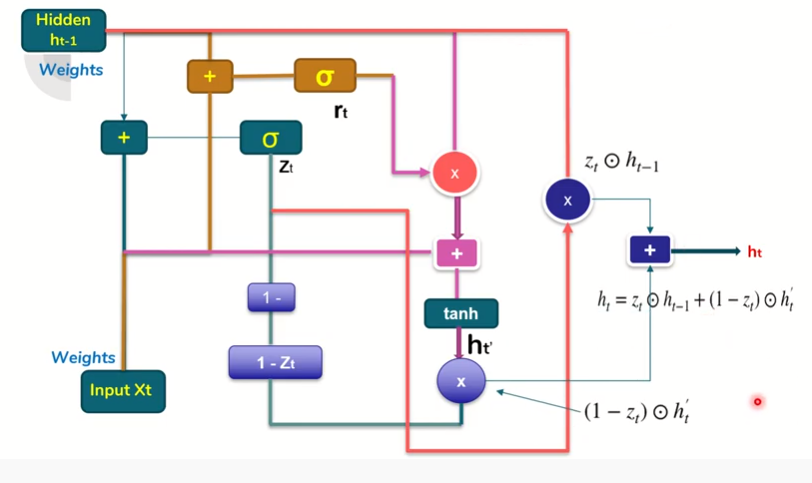


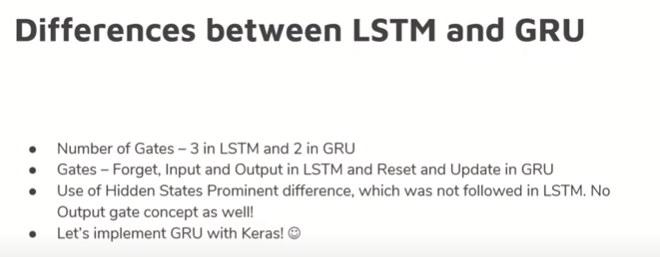


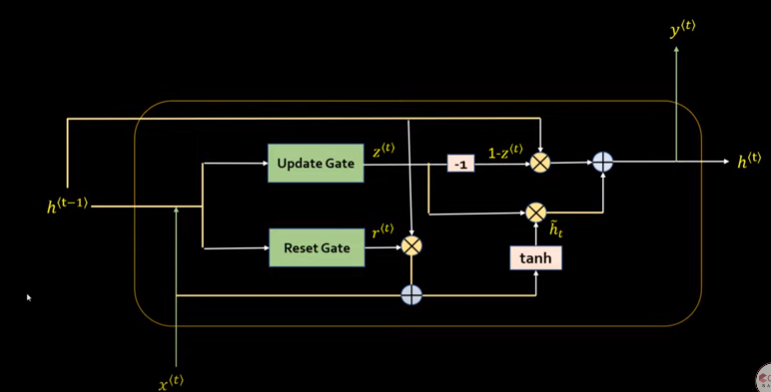


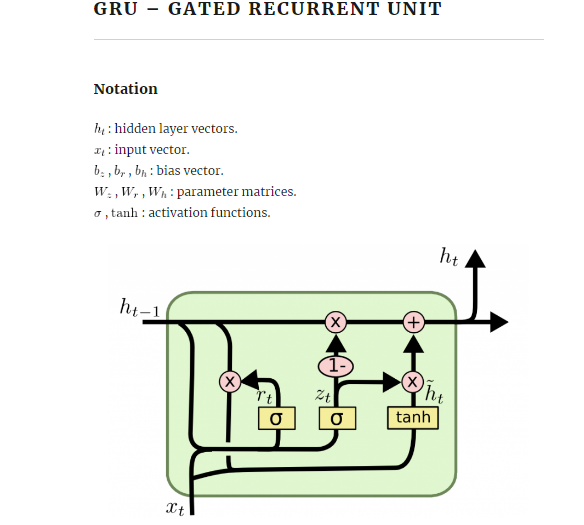


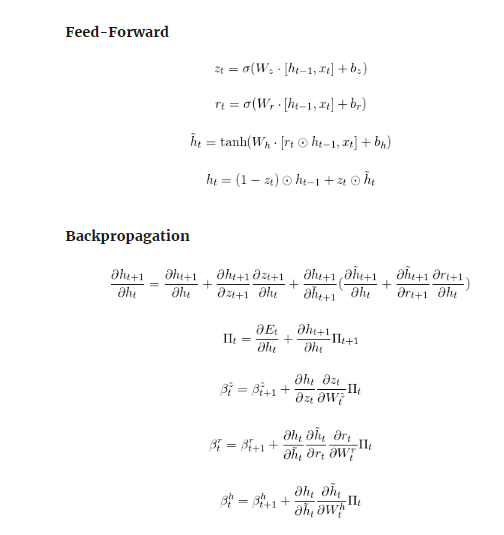










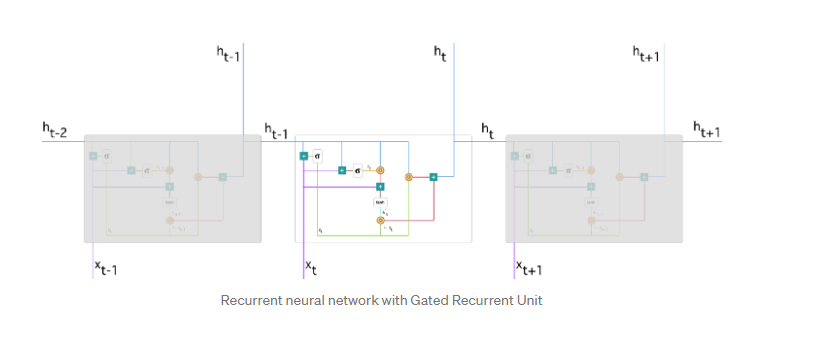


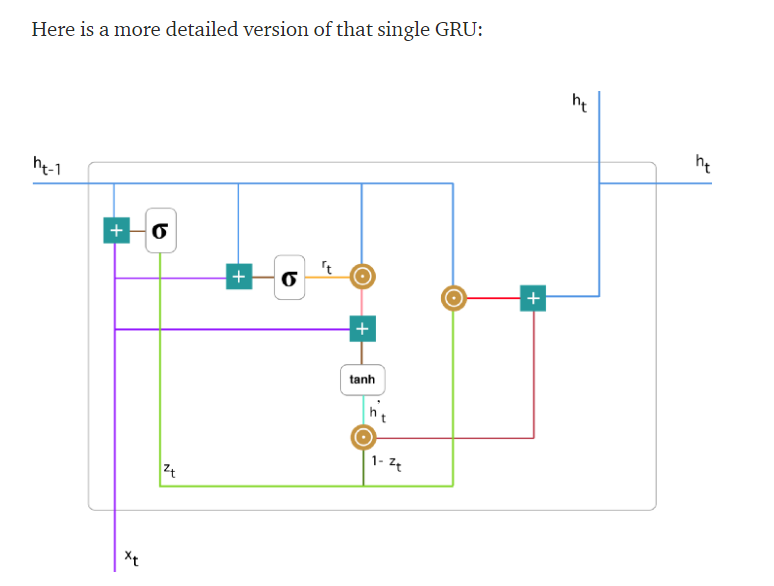
## How do GRUs work?

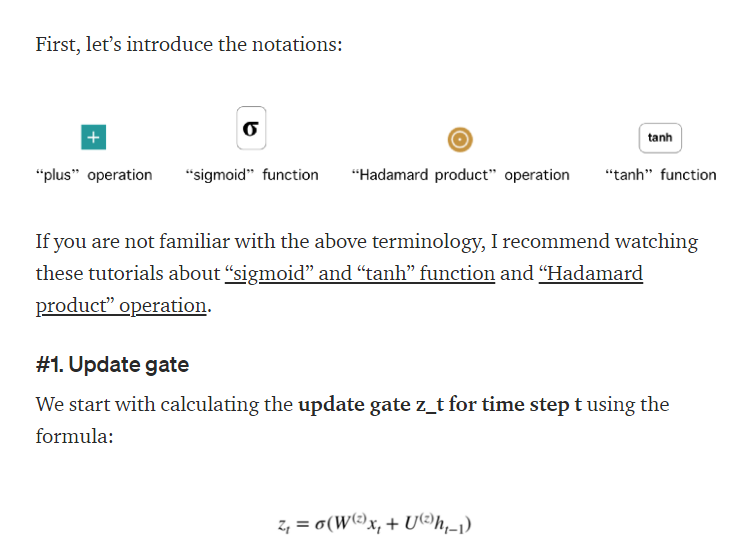
**As mentioned above, GRUs are improved version of standard recurrent neural network. But what makes them so special and effective?**

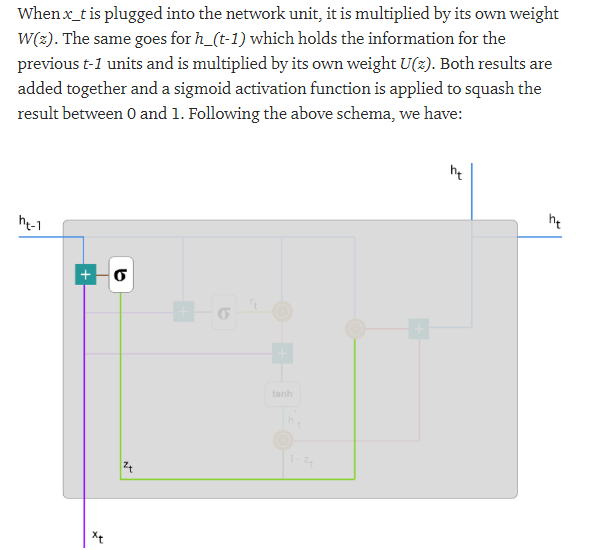
**To solve the vanishing gradient problem of a standard RNN, GRU uses, so-called,**update gate and reset gate**. Basically, these are two vectors which decide what information should be passed to the output. The special thing about them is that they can be trained to keep information from long ago, without washing it through time or remove information which is irrelevant to the prediction.**

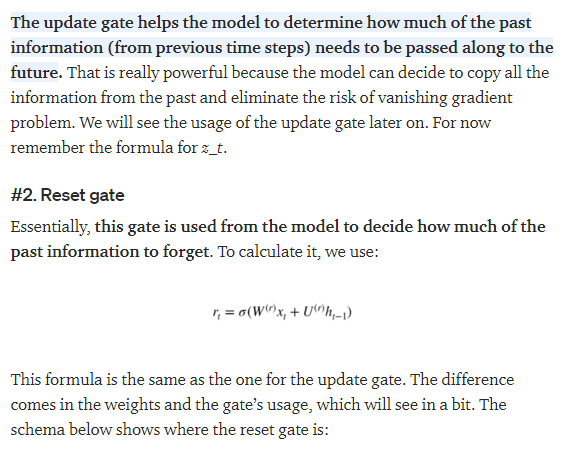
**To explain the mathematics behind that process we will examine a single unit from the following recurrent neural network:**

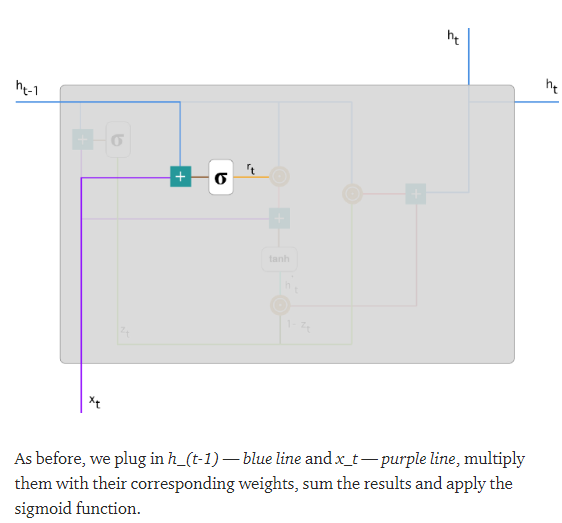


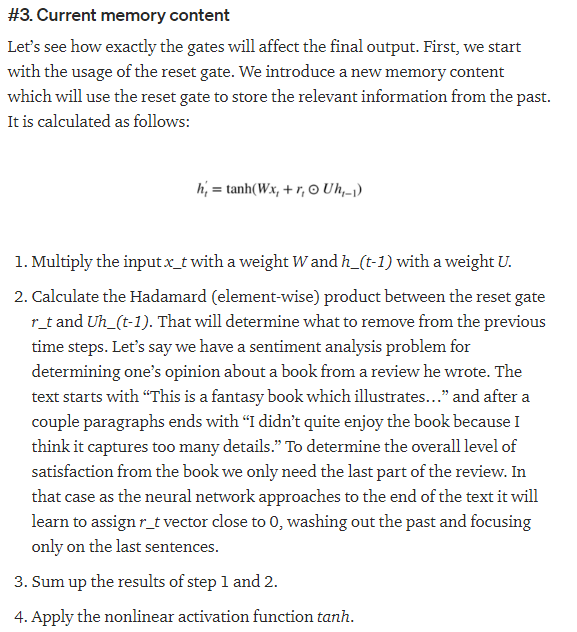


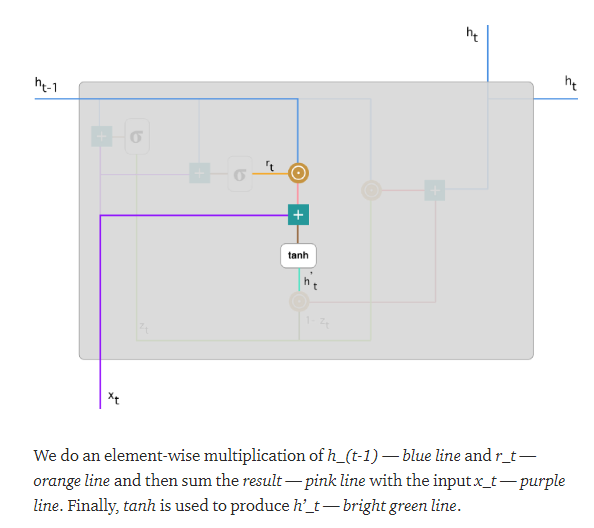


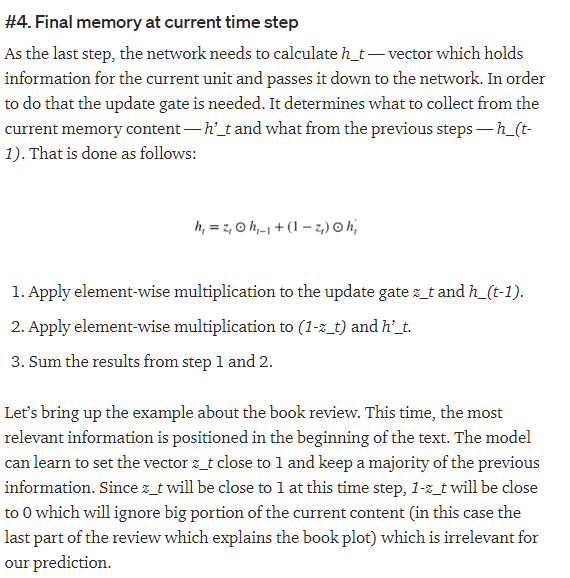


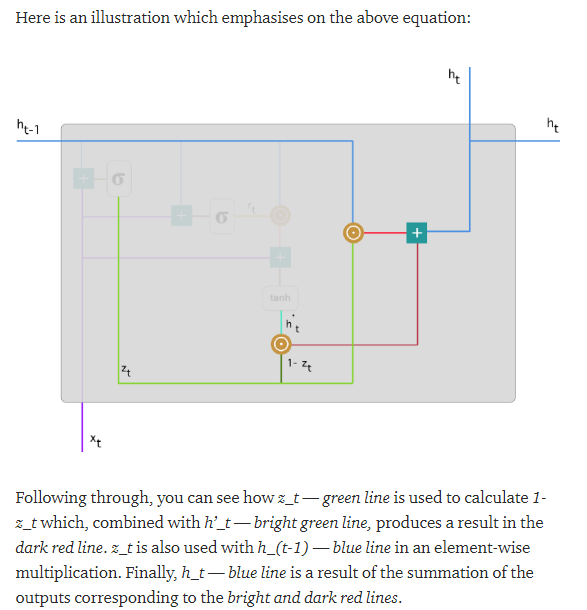


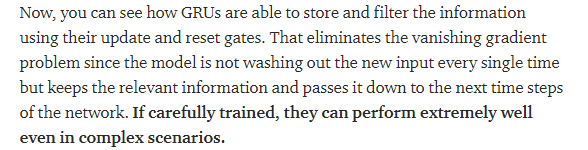












### Advantages of Gated Recurrent Unit

**Gated Recurrent Unit can be used to improve the memory capacity of a recurrent neural network as well as provide the ease of training a model. The hidden unit can also be used for settling the vanishing gradient problem in recurrent neural networks. It can be used in various applications, including speech signal modelling, machine translation, handwriting recognition, among others.**