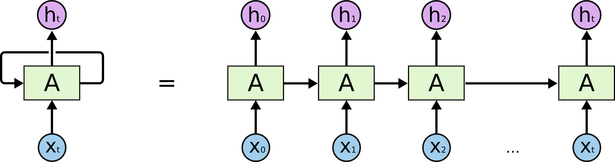
1. **Explain the basic architecture of RNN cell.**

**Ans:** The fundamental feature of a Recurrent Neural Network (RNN) is that the network contains at least one feed-back connection, so the activations can flow round in a loop. That enables the networks to do temporal processing and learn sequences, e.g., perform sequence recognition/reproduction or temporal association/prediction. Recurrent neural network architectures can have many different forms. One common type consists of a standard Multi-Layer Perceptron (MLP) plus added loops. These can exploit the powerful non-linear mapping capabilities of the MLP, and also have some form of memory. Others have more uniform structures, potentially with every neuron connected to all the others, and may also have stochastic activation functions. For simple architectures and deterministic activation functions, learning can be achieved using similar gradient descent procedures to those leading to the back-propagation algorithm for feed-forward networks.

In sequential tasks such as natural language and speech processing, there is always dependence of present input data upon the previous applied inputs. Task of RNNs is to find the relationship between current input and the previous applied inputs. In theory RNNs can make use of information sequence of any arbitrarily length, but in practice they are limited to looking back only a few steps.

****

The above figure shows a RNN being unfolded into a full network. By unfolding we simply mean that we are repeating the same layer structure of network for the complete sequence. Xt is the input at time step t. Xt is a vector of any size N. A is the hidden state at time step t. It’s the “memory” of the network. It is calculated based on the previous hidden state and the input at the current step.

Represented by At= f (W Xt +U At-1)

Here W and U are weights for input and previous state value input. And f is the non-linearity applied to the sum to generate final cell state.

1. **Explain Backpropagation through time (BPTT)**

**Ans:** Backpropagation through time (BPTT) is a gradient-based technique for training certain types of recurrent neural networks. It can be used to train Elman networks. The algorithm was independently derived by numerous researchers

1. **Explain Vanishing and exploding gradients**

**Ans:** The following is the difference between vanishing and Exploding gradients

**Gradient:** The Gradient is nothing but a derivative of loss function with respect to the weights. It is used to updates the weights to minimize the loss function during the back propagation in neural networks.

**Vanishing Gradient:** Vanishing Gradient occurs when the derivative or slope will get smaller and smaller as we go backward with every layer during backpropagation.

When weights update is very small or exponential small, the training time takes too much longer, and in the worst case, this may completely stop the neural network training.

A vanishing Gradient problem occurs with the sigmoid and tanh activation function because the derivatives of the sigmoid and tanh activation functions are between 0 to 0.25 and 0–1. Therefore, the updated weight values are small, and the new weight values are very similar to the old weight values. This leads to Vanishing Gradient problem. We can avoid this problem using the ReLU activation function because the gradient is 0 for negatives and zero input, and 1 for positive input.

**Exploding gradient:** Exploding gradient occurs when the derivatives or slope will get larger and larger as we go backward with every layer during backpropagation. This situation is the exact opposite of the vanishing gradients.

This problem happens because of weights, not because of the activation function. Due to high weight values, the derivatives will also higher so that the new weight varies a lot to the older weight, and the gradient will never converge. So it may result in oscillating around minima and never come to a global minima point.

**4.Explain Long short-term memory (LSTM)**

**Ans:** Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems.

This is a behavior required in complex problem domains like machine translation, speech recognition, and more.

LSTMs are a complex area of deep learning. It can be hard to get your hands around what LSTMs are, and how terms like bidirectional and sequence-to-sequence relate to the field.

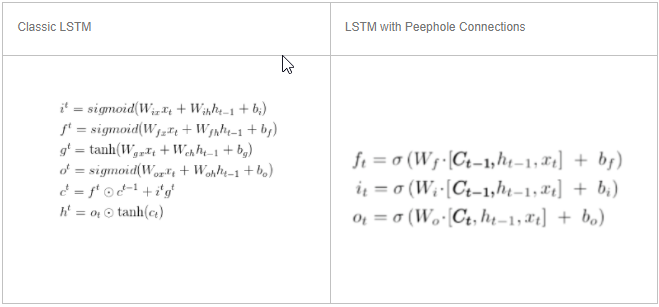
**5.Explain Gated recurrent unit (GRU)**

**Ans:** 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

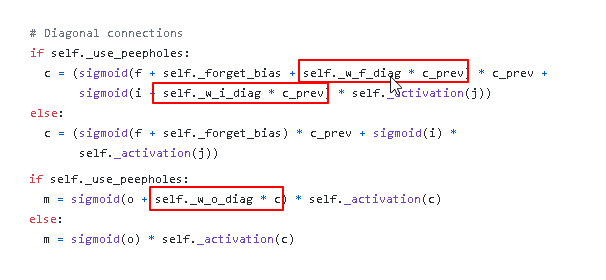
**6.Explain Peephole LSTM**

**Ans:** LSTM peephole conncections is one of improvements for classic LSTM network.

* Difference between LSTM peephole conncections and classic LSTM
* We should compare them with their formulas.

****

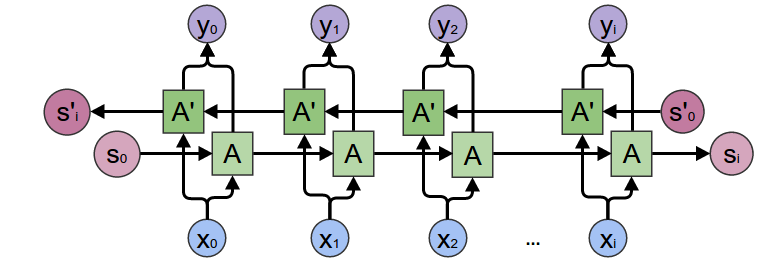
* We can find the main differences between classic LSTM and LSTM with peephole connections are in three gates.
* LSTM with peephole connections add hidden state Ct to three gates in classic lstm.
* We also can find the detail in tensorflow source code.

****

**7.Bidirectional RNNs**

**Ans:**  Bidirectional recurrent neural networks(RNN) are really just putting two independent RNNs together. The input sequence is fed in normal time order for one network, and in reverse time order for another. The outputs of the two networks are usually concatenated at each time step, though there are other options, e.g. summation.

This structure allows the networks to have both backward and forward information about the sequence at every time step. The concept seems easy enough. But when it comes to actually implementing a neural network which utilizes bidirectional structure, confusion arises…

****

**8.Explain the gates of LSTM with equations.**

**Ans:** **LSTM is made up of Gates:**

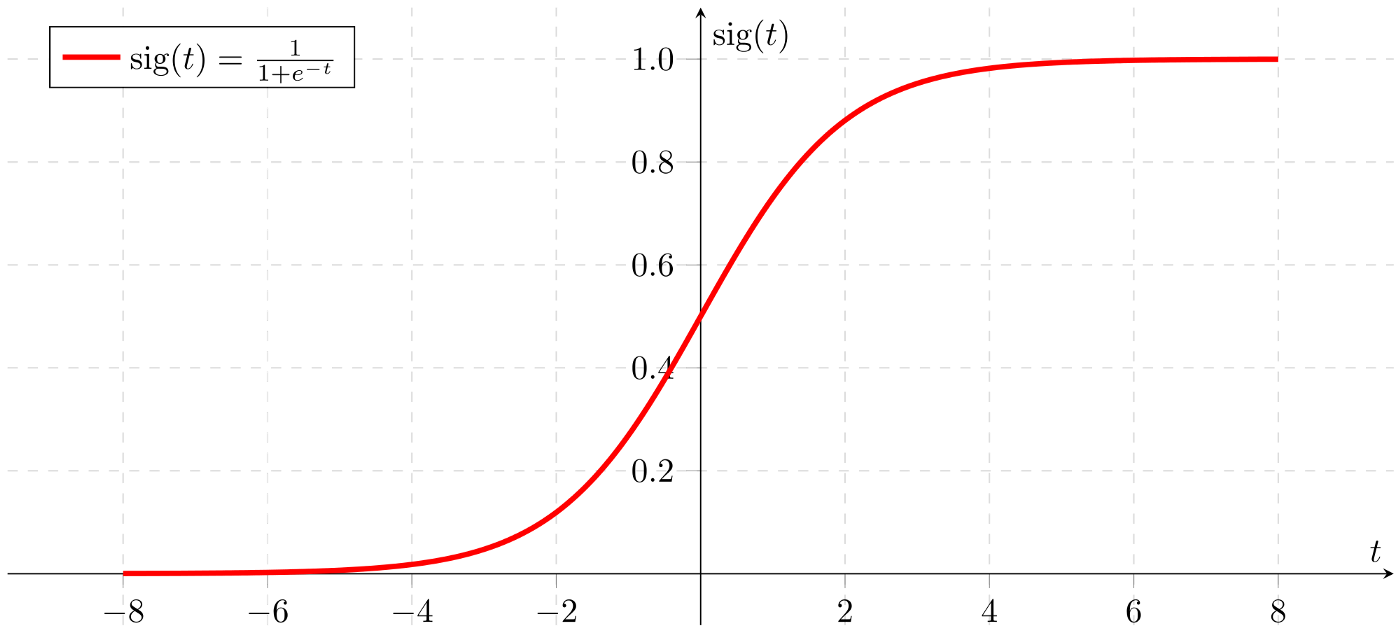
In LSTM we will have 3 gates:

**1) Input Gate.**

**2) Forget Gate.**

**3) Output Gate.**

Gates in LSTM are the sigmoid activation functions i.e they output a value between 0 or 1 and in most of the cases it is either 0 or 1.



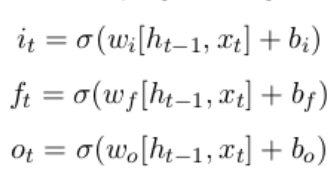
we use sigmoid function for gates because, we want a gate to give only positive values and should be able to give us a clear cut answer whether, we need to keep a particular feature or we need to discard that feature.

“0” means the gates are blocking everything.

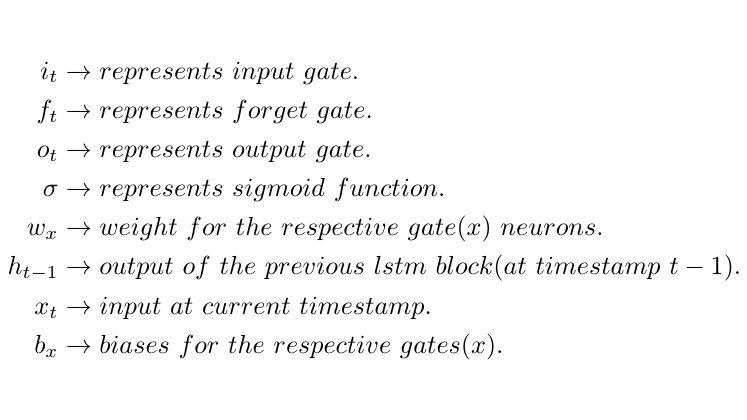
“1” means gates are allowing everything to pass through it.

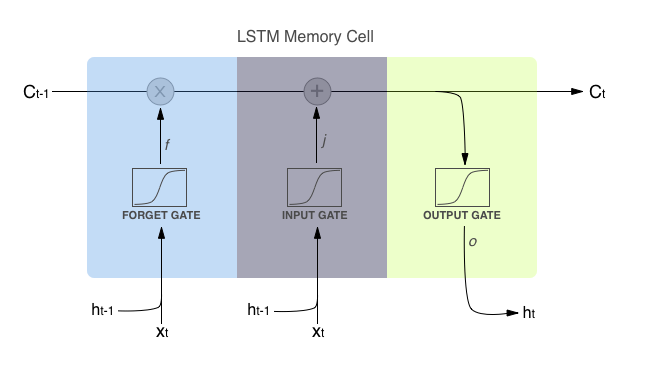
The equations for the gates in LSTM are:

 are:



**Equation of Gates**



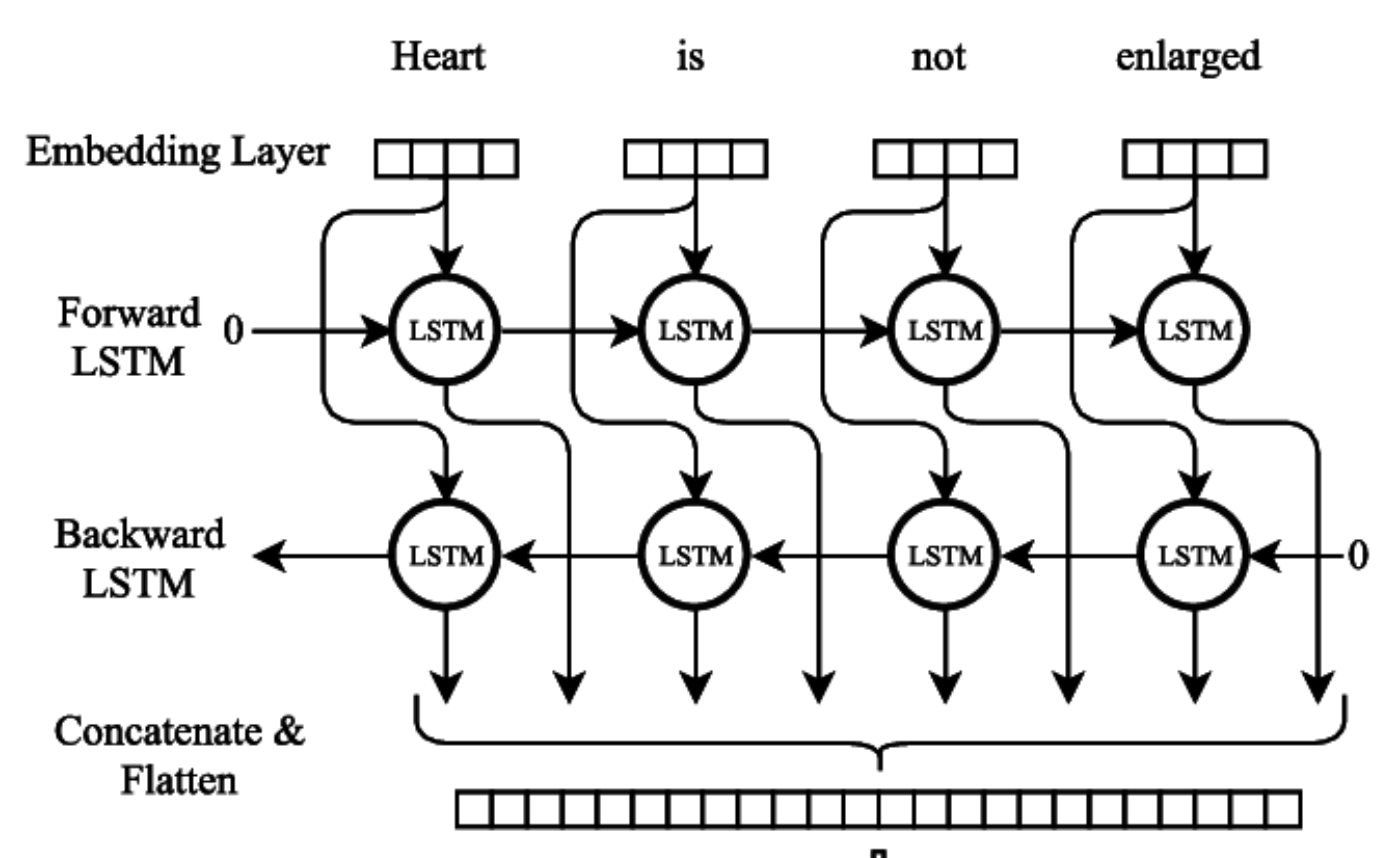


*First equation is for*Input Gate*which tells us that what new information we’re going to store in the cell state(that we will see below).*

*Second is for the forget gate which tells the information to throw away from the cell state.*

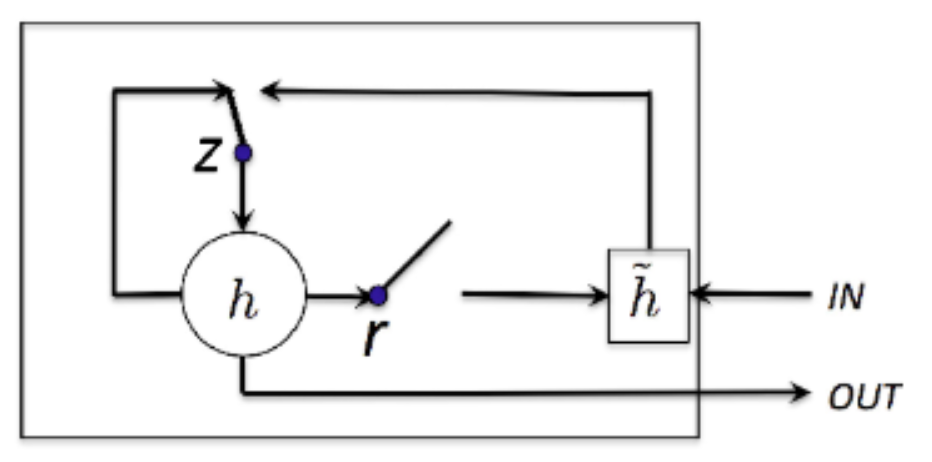
*Third one is for the output gate which is used to provide the activation to the final output of the lstm block at timestamp ‘t’.*

**9.Explain BiLSTM**

**Ans:** A Bidirectional LSTM, or biLSTM, is a sequence processing model that consists of two LSTMs: one taking the input in a forward direction, and the other in a backwards direction. BiLSTMs effectively increase the amount of information available to the network, improving the context available to the algorithm (e.g. knowing what words immediately follow and precede a word in a sentence). ****

**10.Explain BiGRU**

**Ans:** A Bidirectional GRU, or BiGRU, is a sequence processing model that consists of two GRUs. one taking the input in a forward direction, and the other in a backwards direction. It is a bidirectional recurrent neural network with only the input and forget gates.

****