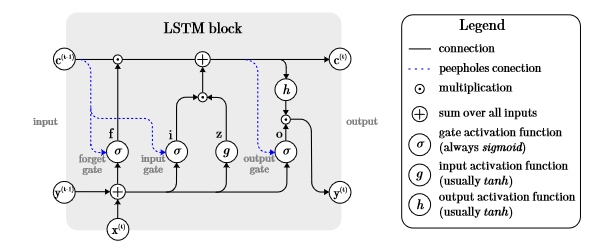
**Abstract**: The fields of machine learning and neurocomputing have both been revolutionized by long short-term memory (LSTM). Several internet sources claim that this methodology has significantly enhanced Google Translate's machine translations, Alexa's responses, and speech recognition. Facebook also uses this neural network, and as of 2017, it was performing more than 4 billion LSTM-based translations every day. It's interesting to note that recurrent neural networks performed rather discretely before the appearance of LSTM. The capacity of this recurrent network to handle the exploding/vanishing gradient problem, which is a challenging problem to be avoided when training recurrent or very deep neural networks, is one factor in its success.

**Introduction**: It is challenging to train recurrent or extremely deep neural networks because they frequently experience the exploding/vanishing gradient problem. The LSTM architecture was developed to address this issue while learning long-term dependencies. The learning capability of LSTM had a significant theoretical and practical impact on many domains, making it a cutting-edge model. Due to its high applicability and popularity, this neural architecture has also found its way into the world of gaming.

**Long Short-Term Memory**: The LSTM model is an effective recurrent neural system that was created specifically to solve the exploding/vanishing gradient difficulties that frequently occur while learning long-term dependencies, even when the smallest time lags are quite long. Using a constant error carousel (CEC), which keeps the error signal within each unit's cell, may often prevent this. In actuality, these cells are recurrent networks in and of themselves, with an intriguing architecture that results from the CEC's extension to include the input gate and output gate, which together make up the memory cell. Feedback with a one-time step lag is indicated by the self-recurrent connections.

A cell, an input gate, an output gate, and a forget gate make up an LSTM unit. The LSTM network did not originally include this forget gate, but this was suggested so that the network could reset its state. The three gates control the flow of information associated with the cell, and the cell remembers values across arbitrary time intervals.

In a nutshell, the LSTM architecture is made up of several memory blocks that are recurrently connected to sub-networks. The memory block's purpose is to keep its state constant over time and control information flow using non-linear gating units.



2.1.1. LSTM Block

The so-called highway networks used the gating technique to enable unrestricted information flow across numerous tiers. Another proof-of-concept demonstrating the functionality of the gates may be said to be. Although LSTM already performs very well, various works investigated the potential for performance enhancement.

The LSTM model discussed employs Graves and Schmidhuber’s complete gradient training method to modify the network’s learnable parameters (weights). The weights connecting the various parts of the network are calculated via Backpropagation Through Time.

**Relevant applications**: Both individually and in conjunction with other deep learning designs, the LSTM network is used in a wide range of problem domains. One of the most sophisticated networks for processing temporal sequences is the LSTM, as was previously mentioned. Since it is possible to integrate the LSTM with other networks to generate hybrid models, it is still one of the most widely used network options. Any issue requiring temporal memory can be handled by LSTM, including time-series forecasts.

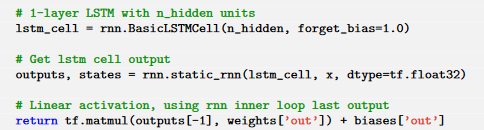
**Time series prediction**: Time series data come to mind first when thinking about temporal sequences in data. This is a broad concept, though. The LSTM model has been applied to financial market predictions using a standard random forest, a conventional deep neural network, and a standard logistic regression in the more literal sense of time-series predictions. Receiving a time series as input does not necessarily mean the model will predict the next values in the series, as it can also be used to train a classifier.

**Natural Language Processing**: LSTM is a formidable force in the field of context-free and context-sensitive language acquisition. The study of natural language understanding and manipulation by computers to carry out useful tasks is known as natural language processing. For instance, dialogue systems, usually referred to as conversational agents, let people communicate verbally with a machine. The LSTM model was used for the first time to do speech recognition because its key advantage is its ability to handle lengthy time delays. In this experiment, outcomes similar to those of the hidden Markov model (HMM) were attained.

**Sentiment analysis**: Natural language processing and sentiment analysis go hand in hand. Emotions can be detected using a variety of data sources, including physiological information, environmental factors, movies, and more. Sensor signals from these multi-modal data sources were used by many. These signals came specifically from wearable technology and smartphones. They were the first to recognize emotions based on physiological, environmental, and geographic information. Four models, each based on a CNN-LSTM architecture, were created to analyze all the data: one for the on-body data, one for the environment, one for location, and one for the fusion of all the data inputs.

The accuracy level was found to have enhanced by more than 20% when employing this hybrid network as opposed to a conventional multi-layer perception model.

**Implementation of LSTM**: The softmax() function is used to normalize the LSTM’s output vector of probabilities for the subsequent symbol. The anticipated index of the symbol in the reverse dictionary is that of the element with the highest probability. The application’s main component is this model, which Tensorflow makes incredibly easy to implement.



2.1.2 LSTM Code

In the training process, at each step, three symbols are retrieved from the training data to form the input vector. These three symbols are converted to numeric values.