Simple Recurrent Cell

Recurrent cells, which are the building blocks of Recurrent Neural Networks, can be of three different types: LSTM, GRU, and Simple RNN Cell. While first two ones have quite complicated internal structure to learn and sustain long-term dependencies in input sequence, simple cells tend to operate in primitive manner. This provides poorer learning capability, but decreases computational complexity. Simple illustration of its operative structure is given in Figure 1.

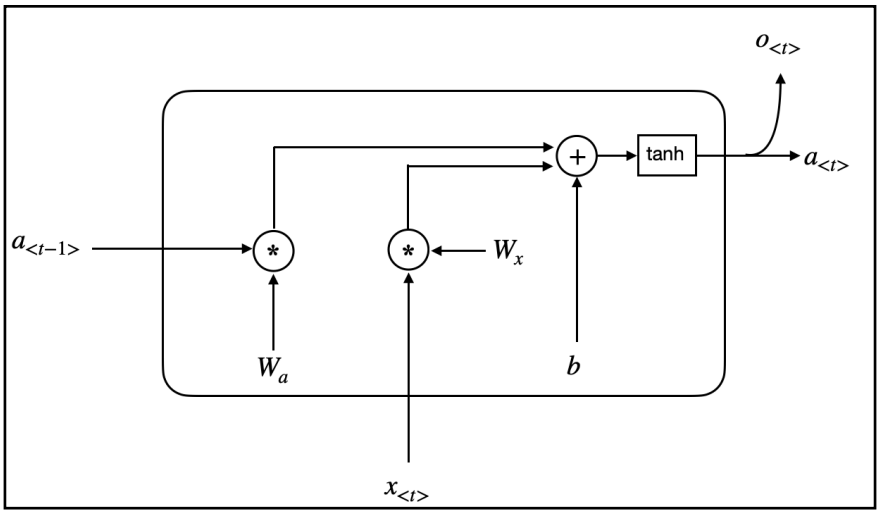

Figure 1: Internal Structure of Simple RNN Cell

To describe neatly, each simple cell in a recurrent network actually consumes two different input arguments to perform its internal computation and thereby producing its output, which are one representative vector in input sequence $x_{<t>}$ and state information inherited from its ancestor $a_{<t-1>}$, which is its preceding cell. These two arguments are at first multiplied by cell's parameter matrices W_x and W_a respectively, and then bias factor b is added to the result of this multiplication. In final step, the output of this operation is activated by pre-determined activation function, which is generally chosen as hyperbolic-tangent. The important thing that we need to know in this point is the fact that parameter matrices are shared along all cells in one recurrent layer. In other words, each cell does not have its own individual parameter system, instead all of them deploys same parameter matrices. This parameter sharing enables variable-length input sequences to be processed by RNN layers. All computations from the aspect of linear algebra is depicted in Figure 2.

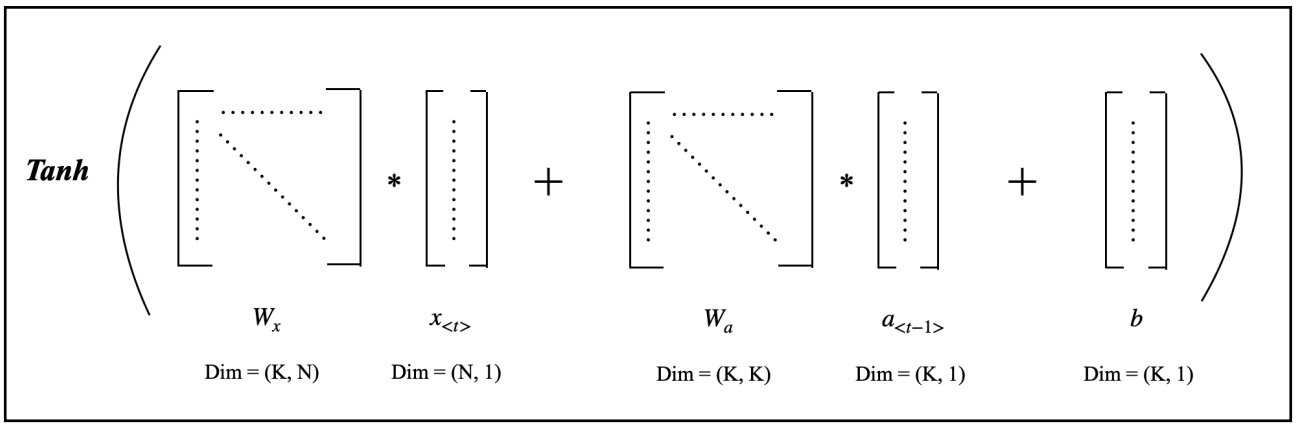

Figure 2: Linear Operations inside Simple RNN Cell