- RNN is an extension of a conventional feedforward neural network, which is able to handle a variable-length sequence input.

- The Long Short-Term Memory (LSTM) unit was initially proposed by Hochreiter and Schmidhuber [1997].

- Unlike the recurrent unit which simply computes a weighted sum of the input signal and applies a nonlinear function, each j-th LSTM unit maintains a memory cj at time t. The output hj, or the activation of the LSTM unit is then:

$$h_t^j = o_t^j \tanh\left(c_t^j\right),$$

Where oj is an output gate that modulates the amount of memory content exposure. The output gate (oj) is computed by:

$$o_t^j = \sigma\left(W_o \mathbf{x}_t + U_o \mathbf{h}_{t-1} + V_o \mathbf{c}_t\right)^j,$$

- The memory cell is updated by partially forgetting the existing memory and adding a new memory:

$$c_t^j = f_t^j c_{t-1}^j + i_t^j \tilde{c}_t^j,$$

- The extent to which the existing memory is forgotten is modulated by a forget gate fj , and the degree to which the new memory content is added to the memory cell is modulated by an input gate ij . Gates are computed by:

$$f_t^j = \sigma\left(W_f \mathbf{x}_t + U_f \mathbf{h}_{t-1} + V_f \mathbf{c}_{t-1}\right)^j,$$
$$i_t^j = \sigma\left(W_i \mathbf{x}_t + U_i \mathbf{h}_{t-1} + V_i \mathbf{c}_{t-1}\right)^j.$$

- Unlike to the traditional recurrent unit which overwrites its content at each time-step an LSTM unit is able to decide whether to keep the existing memory via the introduced gates.Intuitively, if the LSTM unit detects an important feature from an input sequence at early stage, it easily carries this information (the existence of the feature) over a long distance, hence, capturing potential long-distance dependencies.

- A gated recurrent unit (GRU) was proposed by Cho et al. [2014] to make each recurrent unit adaptively capture dependencies of different time scales. Similarly to the LSTM unit, the GRU has gating units that modulate the flow of information inside the unit, however, without having separate memory cells.

- The activation of the GRU at time t is a linear interpolation between the previous activation and the candidate activation:

$$h_t^j = (1 - z_t^j)h_{t-1}^j + z_t^j \tilde{h}_t^j,$$

- Where an update gate (zj) decides how much the unit updates its activation, or content. The update gate is computed by:

$$z_t^j = \sigma\left(W_z \mathbf{x}_t + U_z \mathbf{h}_{t-1}\right)^j .$$

- This procedure of taking a linear sum between the existing state and the newly computed state is similar to the LSTM unit. The GRU, however, does not have any mechanism to control the degree to which its state is exposed, but exposes the whole state each time.

- The candidate activation~hjt is computed similarly to that of the traditional recurrent unit and as in [Bahdanau et al., 2014],

- Where rt is a set of reset gates and is an element-wise multiplication. 1 When off (rjt close to 0), the reset gate effectively makes the unit act as if it is reading the first symbol of an input sequence, allowing it to forget the previously computed state.

- The reset gate rjt is computed similarly to the update gate:

$$r_t^j = \sigma\left(W_r \mathbf{x}_t + U_r \mathbf{h}_{t-1}\right)^j .$$

- The most prominent feature shared between these units (LSTM & GRU) is the additive component of their update from t to t + 1, which is lacking in the traditional recurrent unit. The traditional recurrent unit always replaces the activation, or the content of a unit with a new value computed from the current input and the previous hidden state. On the other hand, both LSTM unit and GRU keep the existing content and add the new content on top of it

- This additive nature has two advantages. First, it is easy for each unit to remember the existence of a specific feature in the input stream for a long series of steps. Any important feature, decided by either the forget gate of the LSTM unit or the update gate of the GRU, will not be overwritten but be maintained as it is.

- Second, and perhaps more importantly, this addition effectively creates shortcut paths that bypass multiple temporal steps. These shortcuts allow the error to be back-propagated easily without too quickly vanishing (if the gating unit is nearly saturated at 1) as a result of passing through multiple, bounded nonlinearities, thus reducing the difficulty due to vanishing gradients.

- These two units however have a number of differences as well. One feature of the LSTM unit that is missing from the GRU is the controlled exposure of the memory content. In the LSTM unit, the amount of the memory content that is seen, or used by other units in the network is controlled by the output gate. On the other hand the GRU exposes its full content without any control.

- Another difference is in the location of the input gate, or the corresponding reset gate. The LSTM unit computes the new memory content without any separate control of the amount of information flowing from the previous time step. Rather, the LSTM unit controls the amount of the new memory content being added to the memory cell independently from the forget gate. On the other hand, the GRU controls the information flow from the previous activation when computing the new candidate activation, but does not independently control the amount of the candidate activation being added (the control is tied via the update gate).

- From these similarities and differences alone, it is difficult to conclude which types of gating units would perform better in general. Although Bahdanau et al. [2014] reported

that these two units performed comparably to each other according to their preliminary experiments on machine translation, it is unclear whether this applies as well to tasks other than machine translation.