Recurrent Neural Networks

Oliver Zhao

1 Basic Recurrent Neural Network

A basic recurrent neural network (RNN) is shown below, where we see that each input x_t provides an output h_t, that incorporates information from the previous time-point output h_{t-1}. The core idea is that this allows for the incorporation of time series data. However, this basic architecture does not perform well with long-term dependencies.

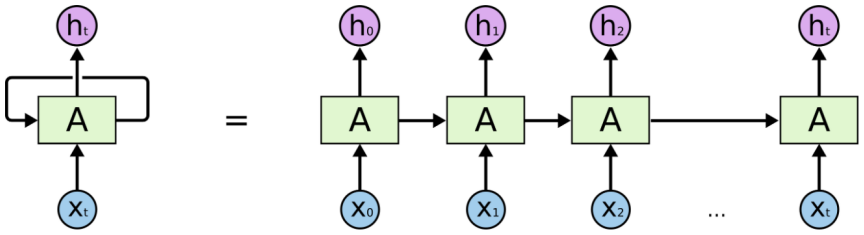


Figure 1: Basic RNN architecture.

2 Long Short-Term Memory (LSTM)

Compared to the basic fully recurrent neural network, LSTMs are designed to better capture long-term dependencies. It consists of repeating blocks shown below, with four major neural network layers. In addition to the outputs h_t, there is also "cell state" information that is used to help create these outputs h_t.

2.1 Discarding Old State Information

The forget gate layer is a sigmoidal layer that chooses which parts of the cell state C_{t-1} to "forget". In short, the output of the forget gate layer f_t with a

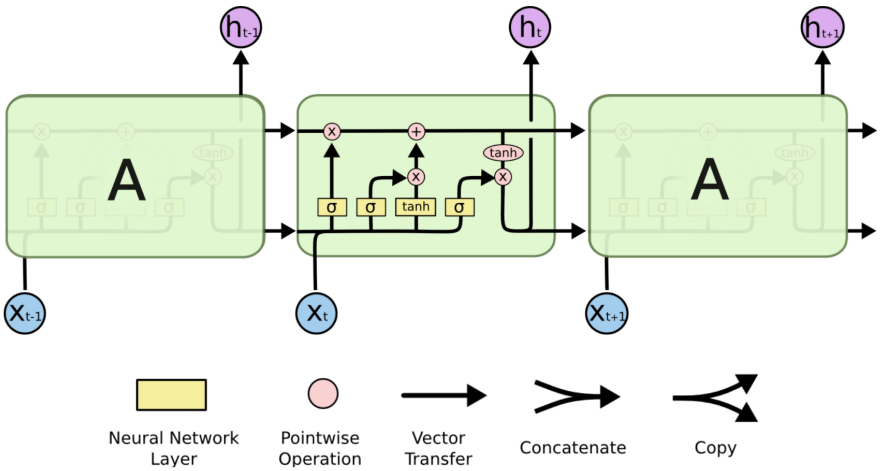
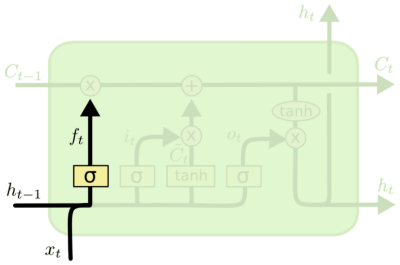


Figure 2: Standard LSTM architecture.

value of "0" indicates that the element is completely forgotten, while a value of "1" indicates that the element is completely retained. Mathematically,

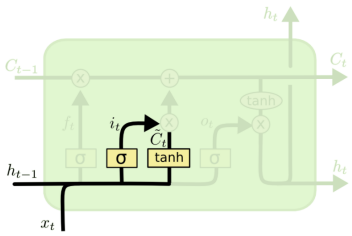


$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Figure 3: Forget gate layer.

2.2 Adding New State Information

An "input gate layer" generates the output i_t, which decides which values to update to make the new cell state C_t. Meanwhile, the tanh layer processes the actual inputs to choose how the values are updated. The outputs of this input gate layer are multiplied with outputs of the tanh layer $\tilde{C}_t$, before being added to update the new cell state C_t. Mathematically,

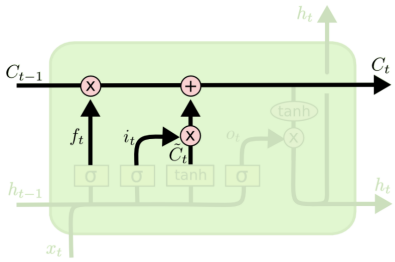


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Figure 4: Input gate layer.

2.3 Calculating the New State

In short, we see that the cell state C_t is calculated by removing some information from the past cell state C_{t-1} and adding new information. Mathematically,

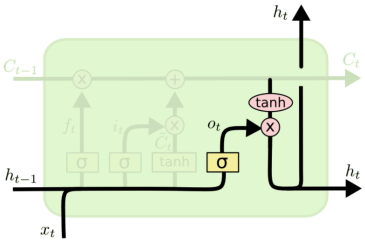


$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Figure 5: Cell state update.

2.4 Calculating the Output

Determining the output h_t comes from the current cell state h_t and additional input x_t. The input x_t is passed through a sigmoid layer to introduce nonlinearity, while the cell state C_t is passed through a tanh layer.

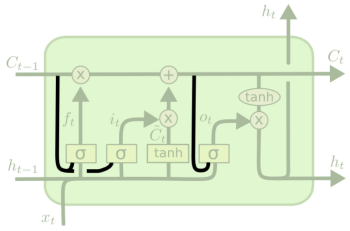


$$o_t = \sigma\left(W_o \; [\, h_{t-1}, x_t] \; + \; b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Figure 6: New state calculation.

2.5 Common LSTM Variants

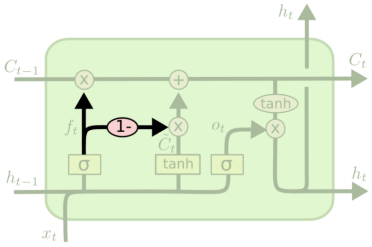
One common variant allows "peephole connections", in which the gate layers can see the cell state. Note that unlike the diagram provided, some of these variants only provide peepholes to some of the gates.



$$f_t = \sigma\left(W_f \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] \; + \; b_f\right)$$
$$i_t = \sigma\left(W_i \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] \; + \; b_i\right)$$
$$o_t = \sigma\left(W_o \cdot [\boldsymbol{C_t}, h_{t-1}, x_t] \; + \; b_o\right)$$

Figure 7: LSTM with peepholes.

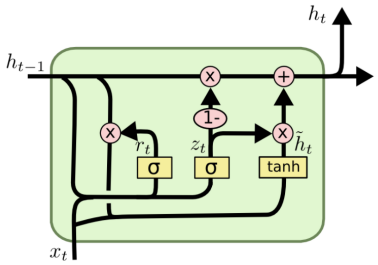
Another variation is to allow input and forget gates to make coupled decisions, instead of independent decisions, which is done with a simple inversion connection.



$$C_t = f_t * C_{t-1} + \boldsymbol{(1 - f_t)} * \tilde{C}_t$$

Figure 8: LSTM with coupled input and forget decisions.

Another variation is the Gated Recurrent Unit (GRU), which lacks an output gate and has fewer parameters. Notably, it also lacks a cell state C_t.



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Figure 9: Gated recurrent unit.