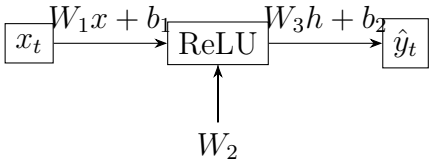
Worksheet 5: RNNs and LSTMs

Page 1: RNNs

We can do this with different amounts of data. Some weather stations might have different amounts.

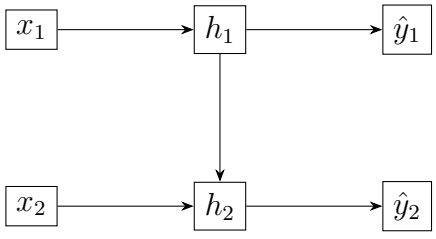


Example: Input $x = 0$, $W_1 = 1.8$, $b_1 = 0$, $W_2 = -0.5$, $b_2 = 0$, $W_3 = 1.1$.
Output for time step t:

$$h = 1.8 \cdot 0 + 0 = 0, \quad \hat{y} = 1.1 \cdot 0 + 0 = 0.$$

Output for time step $t + 1$ loops around. It is often clearer to *unroll* the network.

Page 2: Unrolling the Network



Now have 2 inputs and 1 output, but still very few parameters.
If $x_2 = 0$:

$$h_2 = W_1x_2 + W_2h_1 + b_1 = 0, \quad \hat{y}_2 = W_3h_2 + b_2 = 0.$$

As we expand over more days, the network grows. If W_2 is not close to one, then

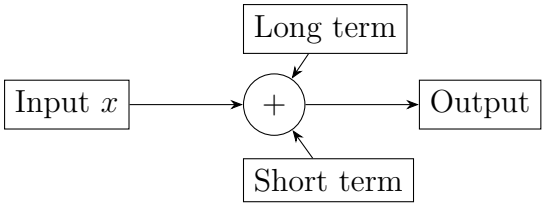
$$h_t \approx W_2^{t-1} h_1,$$

which can easily explode or vanish when training with gradient descent.

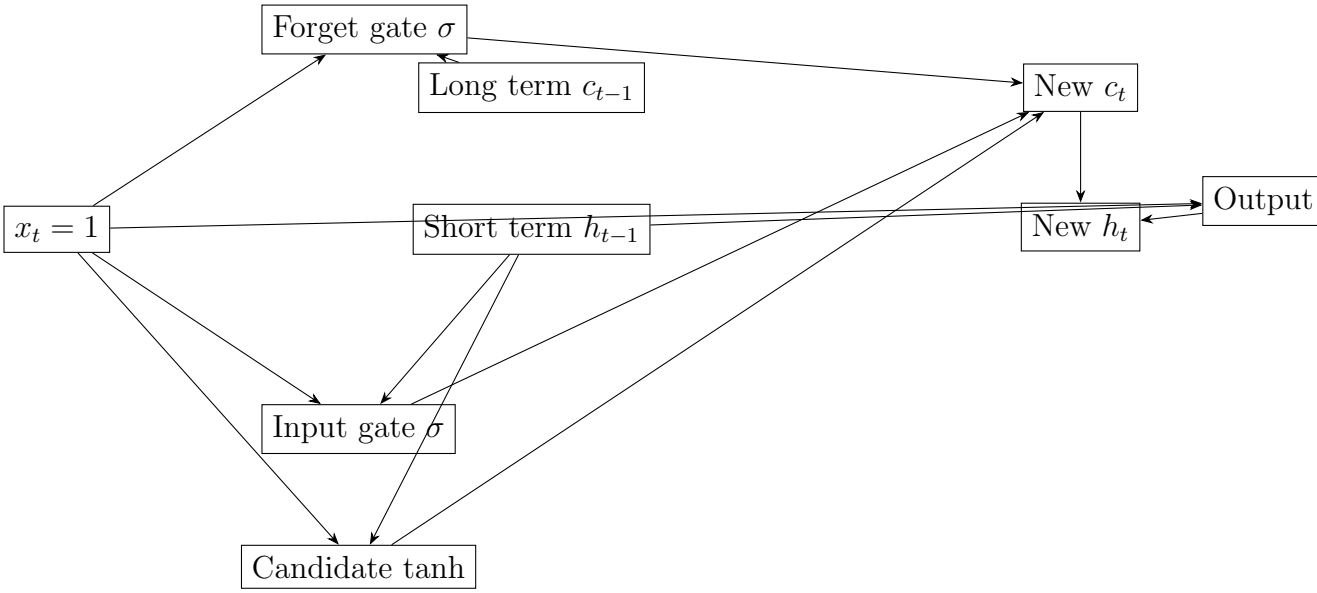
Page 3: LSTMs

Extension of vanilla RNN. Uses two separate paths: one for long memory, one for short memory. Activation functions: sigmoid (range $[0, 1]$) and tanh (range $[-1, 1]$).

This introduces the concept of the **forget gate**.



Page 4: LSTM Full Diagram



Page 5: LSTM Equations

$$\hat{y}_t = \text{prediction at time } t$$

$$h_t = \text{short-term state at time } t, \quad c_t = \text{long-term state.}$$

Equations:

$$f_t = \sigma(Wx_t + Uh_{t-1} + b), \quad \text{forget gate}$$

$$i_t = \sigma(Wx_t + Uh_{t-1} + b), \quad \text{input gate}$$

$$\tilde{c}_t = \tanh(Wx_t + Uh_{t-1} + b), \quad \text{candidate memory}$$

$$o_t = \sigma(Wx_t + Uh_{t-1} + b), \quad \text{output gate}$$

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

$$h_t = o_t \odot \tanh(c_t)$$

Here $\odot$ denotes element-wise multiplication.