Machine Learning and Deep Learning
Lecture-11

By: Somnath Mazumdar
Assistant Professor
sma.digi@cbs.dk

Overview

- RNN Variants:
  - LSTM
  - GRU
  - Bidirectional LSTM

Long Short-Term Memory (LSTM)

Long Short-Term Memory (or LSTM)

- LSTM is a RNN variant.
  - More complicated memory cell than RNN.
  - Proposed in 1997 by Sepp Hochreiter and Jürgen Schmidhuber*.
  - Contain gates to control information flow within recurrent cell.
  - Efficient for capturing long-term dependencies across many time instants.
  - Can be used for faster training, long-term data dependency detection.

- Components of LSTM: Memory cell, Input gate, Forget gate, Output gate.

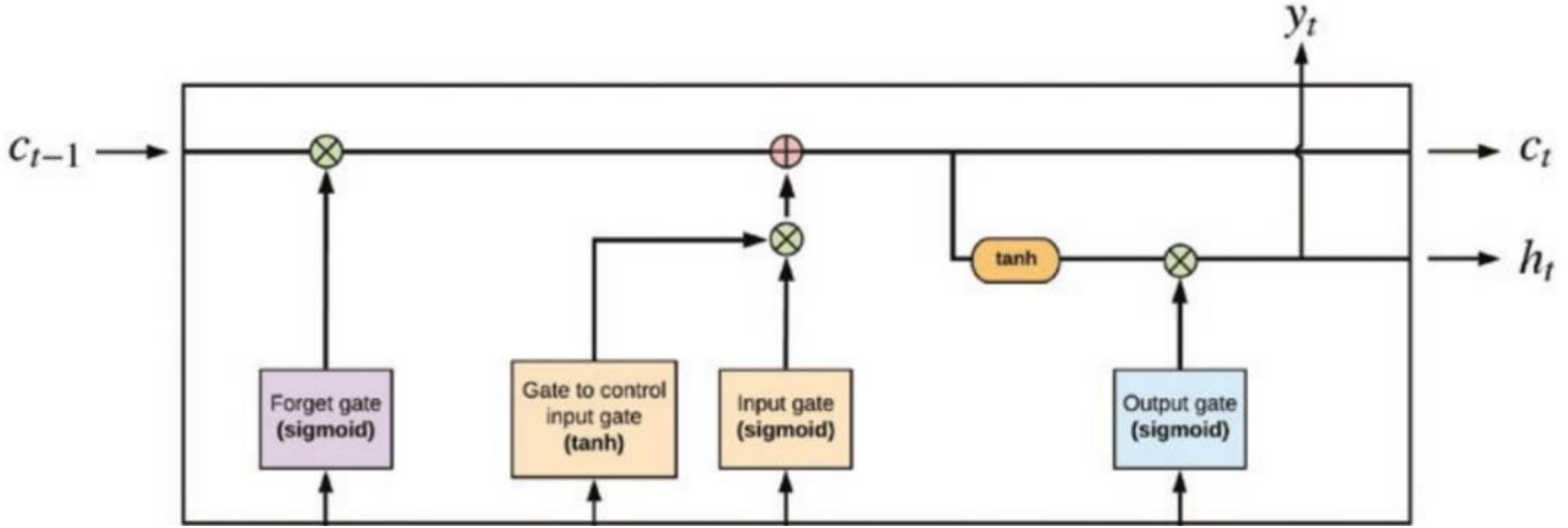
- Applications: Image captioning, stock market prediction, machine translation, and text classification (for sequence processing).

*Sepp Hochreiter and Jürgen Schmidhuber, "Long Short-Term Memory," Neural Computation 9, no. 8 (1997): 1735–1780.

LSTM Components

- Extra components enable RNN to remember and store important events from distant past.

- Input gate: controls what information gets stored in long-term state/memory cell (c).
  - There is another gate that regulates information flowing into input gate.
  - This gate analyzes current input to LSTM cell, x_t and previous short-term state (h_t-1).



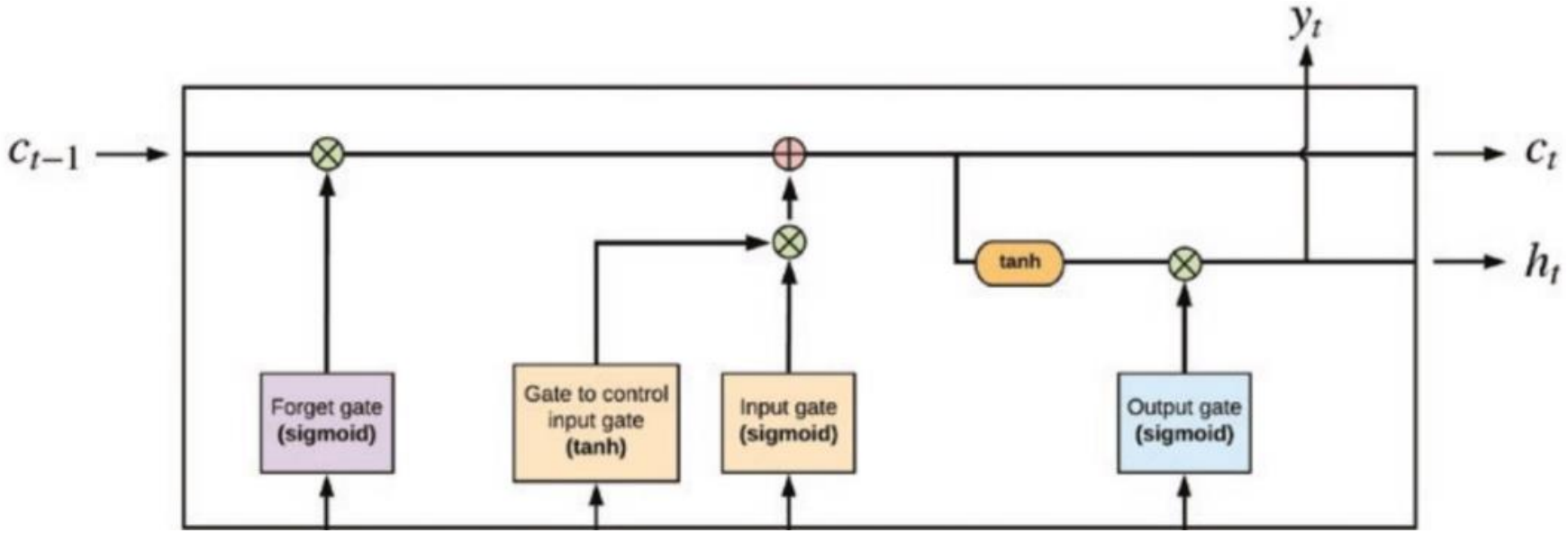
Notations: Previous cell state (c_t-1), Previous hidden state (h_t-1), Current input x_t.

LSTM Components

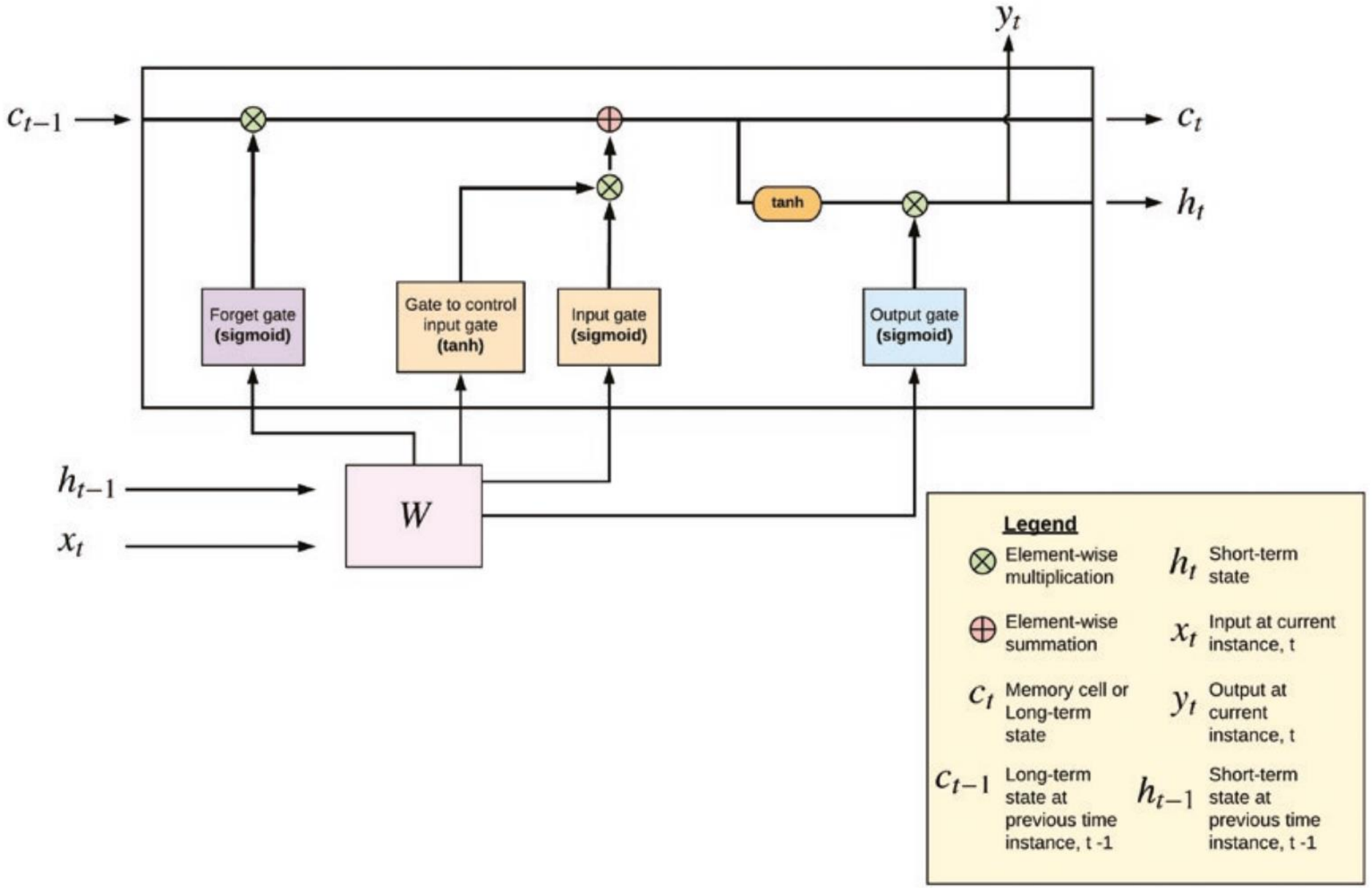
- Forget gate: Regulate how much of information in long-term state is persisted across time instants.
- Note: Components of LSTM cells are fully connected NNs.
  - Variants of RNs with memory cells are peephole connections and gated recurrent units.

- Output gate: controls how much information to output from cell at a time instant.
  - Control value of h_t (short-term state) and y_t (output at time t).

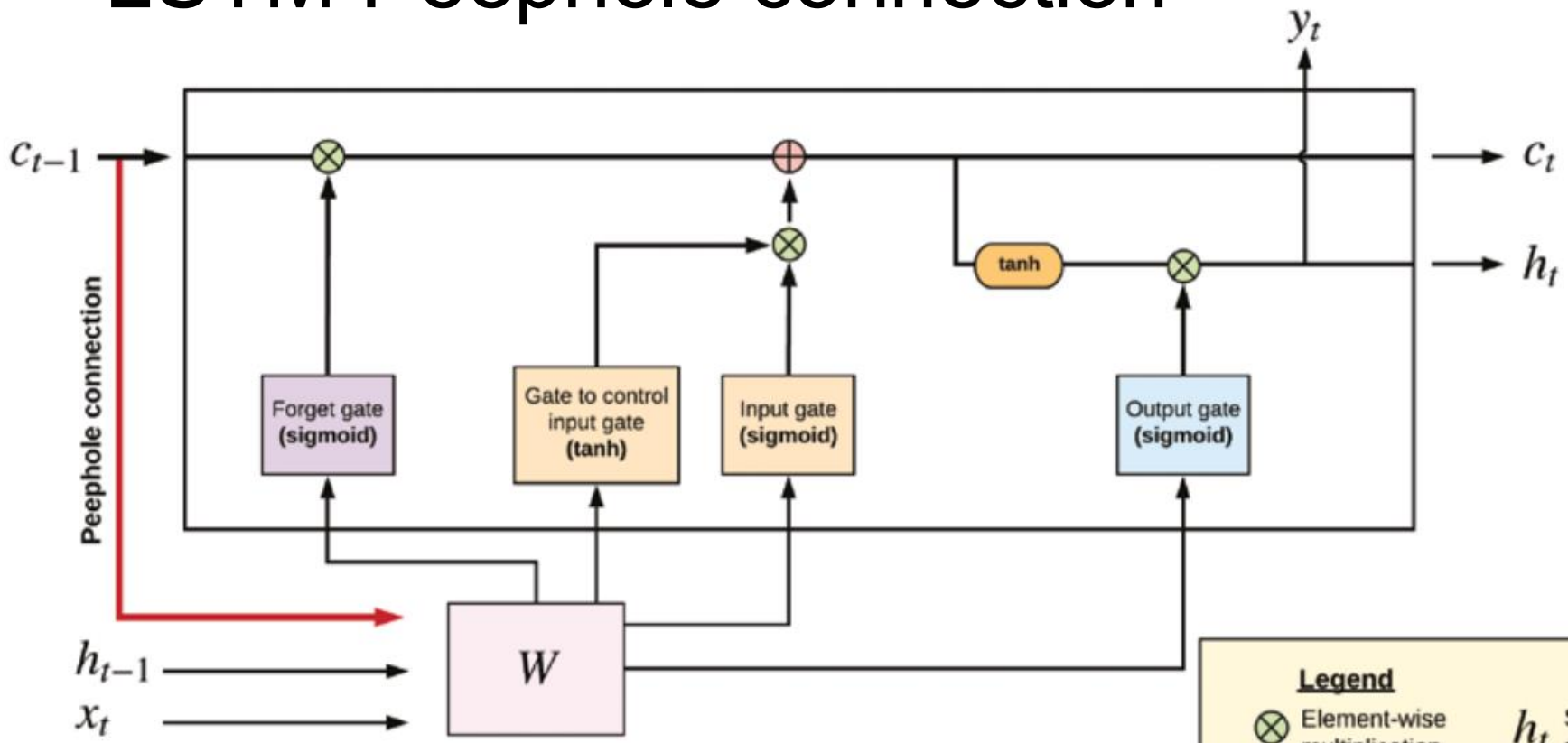
Notations:
- Previous cell state (c_t-1)
- Previous hidden state (h_t-1)
- Current input x_t.



LSTM


y_t
c_{t-1}
c_t
h_t
tanh
Forget gate
(sigmoid)
Gate to control
input gate
(tanh)
Input gate
(sigmoid)
Output gate
(sigmoid)
h_{t-1}
x_t
W
Legend
Element-wise multiplication
Element-wise summation
c_t Memory cell or Long-term state
c_{t-1} Long-term state at previous time instance, t -1
h_t Short-term state
x_t Input at current instance, t
y_t Output at current instance, t
h_{t-1} Short-term state at previous time instance, t -1

LSTM Peephole connection



- LSTM variant with extra connections called peephole connections.
- Previous long-term state is added as an input to controllers of forget gate and input gate.
- Often improves performance, not always!!

LSTM Code Snippet

LSTM cell can recognize an important input, store it in long-term state, preserve it for if it is needed and extract it whenever it is needed.

```
model = keras.models.Sequential([
keras.layers.LSTM(20, return_sequences=True,
input_shape=[None, 1]),
keras.layers.LSTM(20, return_sequences=True),
keras.layers.TimeDistributed(keras.layers.Dense(10)) ])
```

Using LSTM layer NOT SimpleRNN layer

```
model = keras.models.Sequential([
keras.layers.RNN(keras.layers.LSTMCell(20),
return_sequences=True,
input_shape=[None, 1]),
keras.layers.RNN(keras.layers.LSTMCell(20),
return_sequences=True),
keras.layers.TimeDistributed(keras.layers.Dense(10))
])
```

LSTMCell as an argument

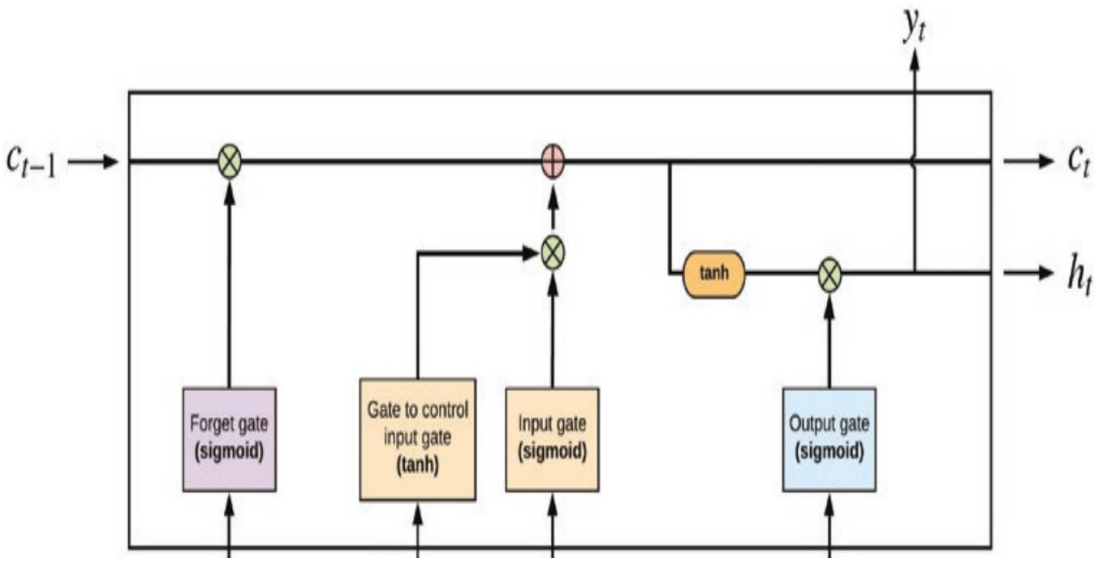
Gated Recurrent Unit (GRU)

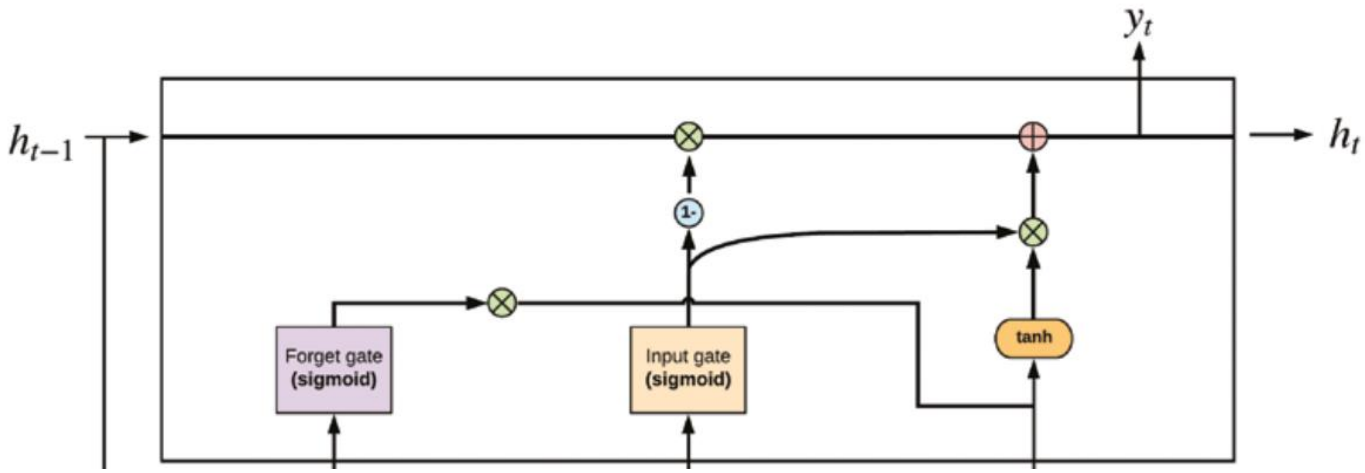
- Variants of LSTM cell exists such as GRU (Gated Recurrent Unit (GRU) cell*.
- LSTM and GRU cells are reasons behind success of RNNs.
- GRU cell is a simplified version of LSTM cell.
- Can handle much longer sequences than simple RNNs.

- GRUs combine forget and input gates to decide on what information should be committed to long-term memory (memory cell) and be left out.
  - Combines long and short-term state into a single state vector (h_t)
  - Removes output gate and returns state vector h_t at each time instant.
  - Implemented in 'tf.keras.layers.GRU()'.

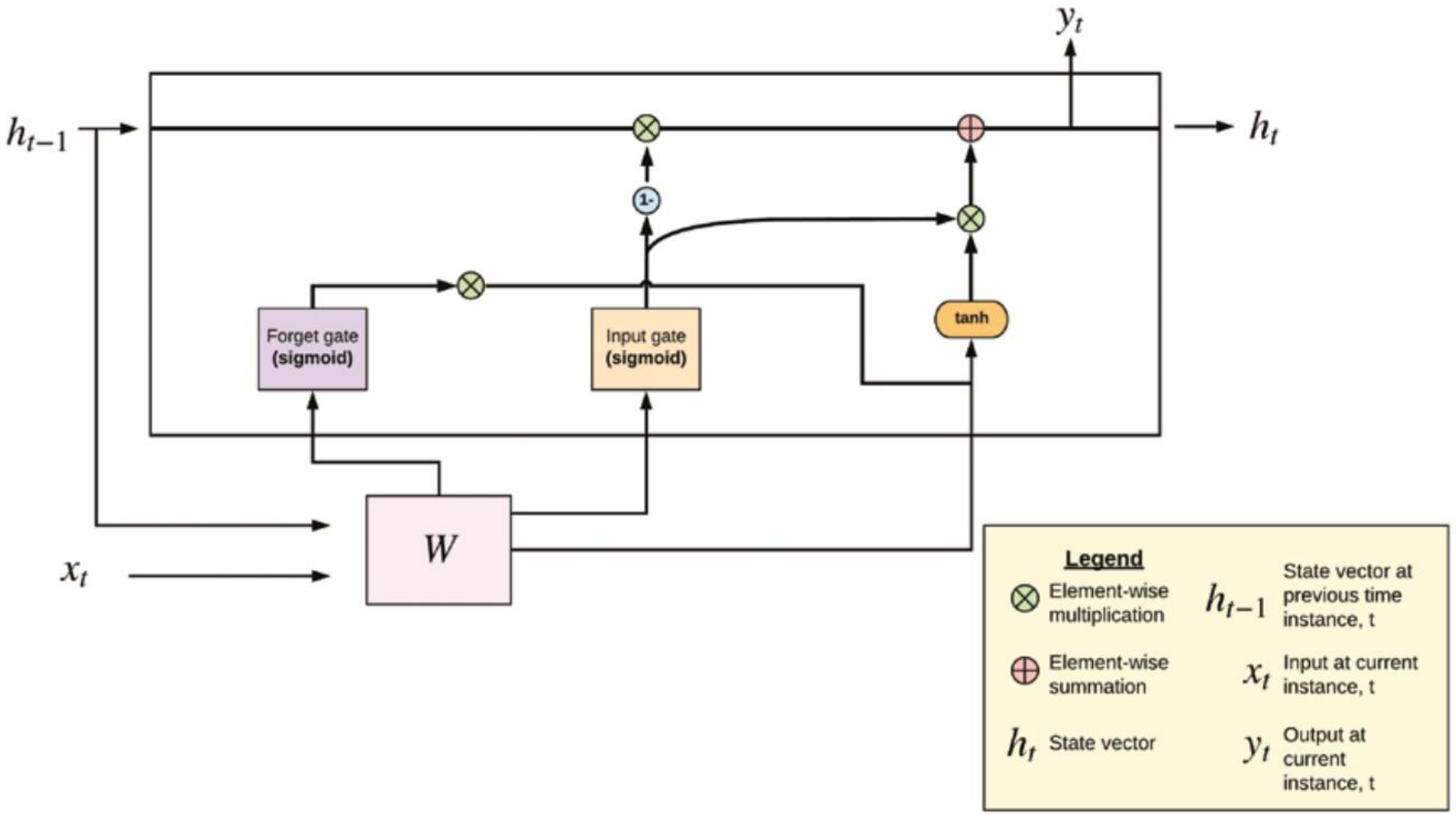
Kyunghyun Cho et al., "Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation," Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (2014): 1724–1734.

LSTM VS GRU


y_t
c_{t-1}
c_t
h_t
tanh
Forget gate
(sigmoid)
Gate to control
input gate
(tanh)
Input gate
(sigmoid)
Output gate
(sigmoid)


y_t
h_{t-1}
h_t
1-
Forget gate
(sigmoid)
Input gate
(sigmoid)
tanh

Gated Recurrent Unit (GRU)



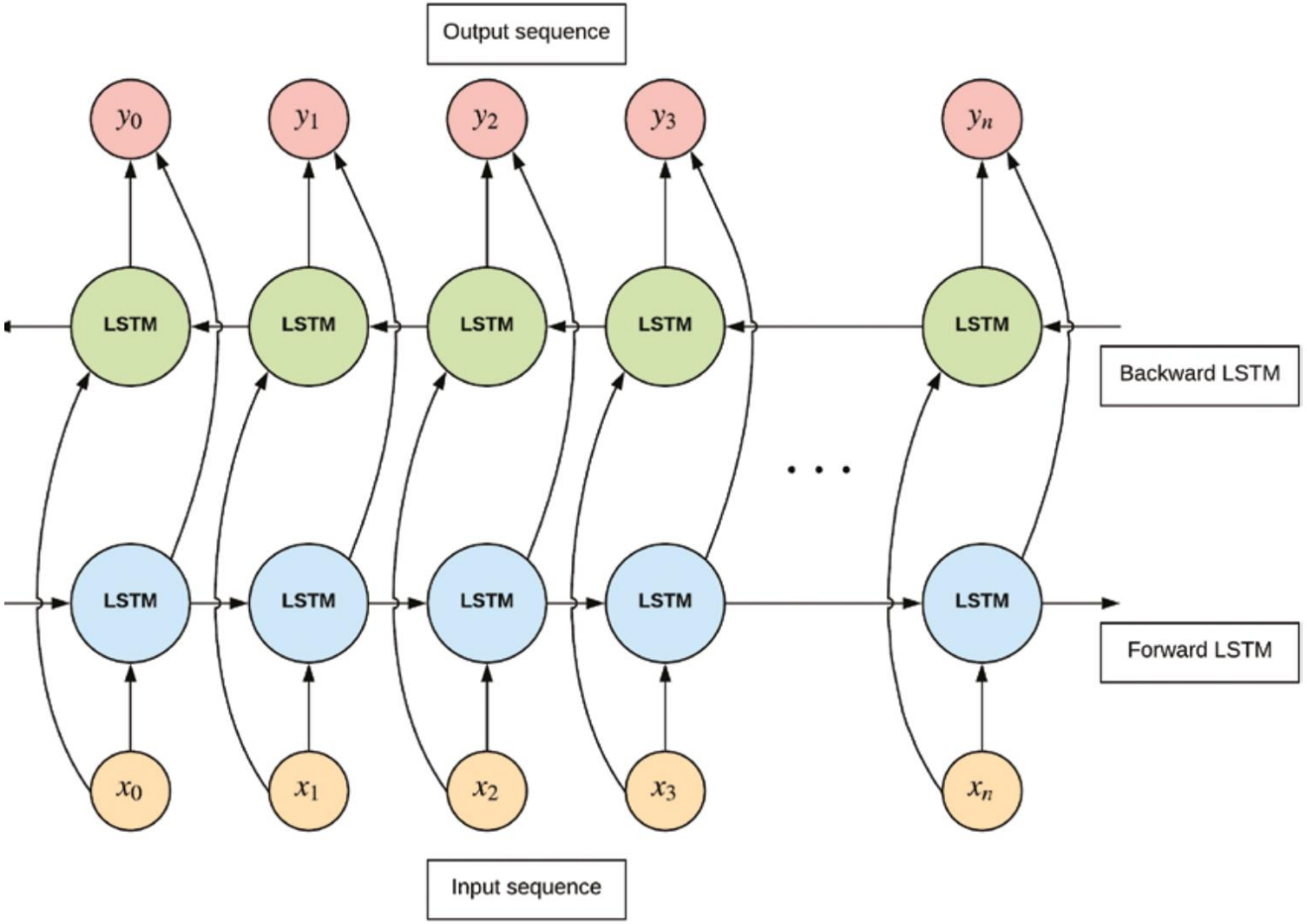
Bidirectional LSTM (BLSTM)

- Bidirectional LSTM a type of RNN
  - Built to improve unidirectional LSTM.
    - Designed for speech recognition.
  - Combines outputs from side-by-side networks to predict next time step of a sequence having both past and future related information.

- Usage Scenarios
  - When a coherent set of information is present (learning problems).
  - Works well for sequence prediction.

Bi-LSTM

- Forward LSTM: Involves placing RN layers beside each other where one layer works to learn long-term dependencies from past.
- Backward LSTM: Input is reversed and fed into the network; network learns long-term dependencies from future.



Recap: Compare

	LSTM	GRU	Bi-LSTM
Architecture	Three gates: input, output, and forget	Two gates	Consist of two LSTM layers
Advantage	1. Capture long-term dependencies in sequential data 2. Handle complex sequences	1. Faster to train 2. Perform similarly to LSTMs, especially with less training data*	1. Offers comprehensive understanding 2. Better performance
Disadvantage	Complex structure	Cannot capture long-term dependencies	Computational complexity and training time