RNN Overview - 1

RNN, LSTM and GRU

2021. 4. 21

Dajin Han

Index

What is RNN?

Recurrent Neural Network

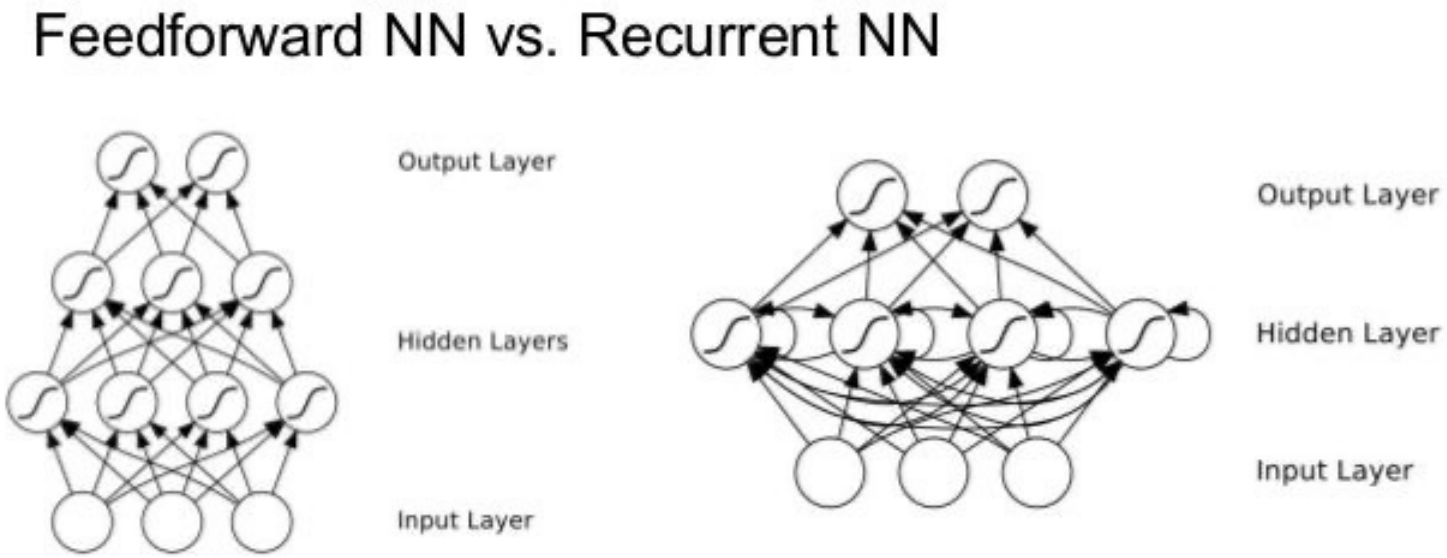
What is RNN?

- Recurrent Neural Network
  - Directed cycle

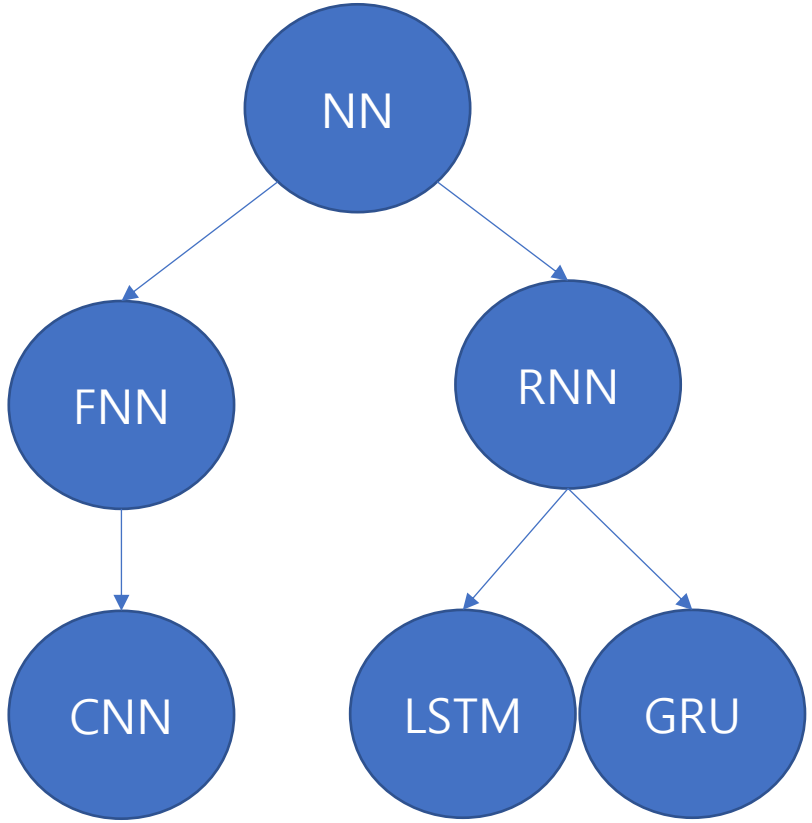
- Suitable for sequential data processing
  - Natural language processing(NLP)
  - Translation

- Various and flexible structures

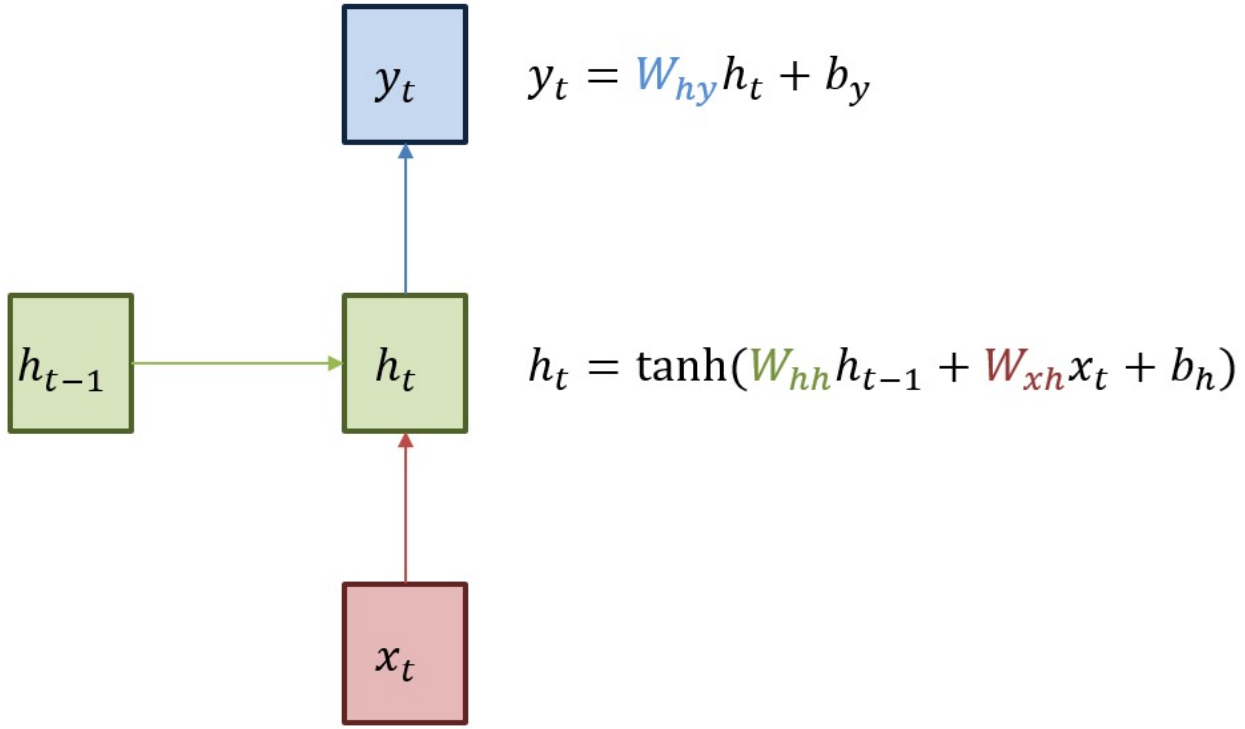
RNN vs CNN



Recurrent neural networks (RNNs) allow cyclical connections.



RNN Structure



$$y_t = W_{hy} h_t + b_y$$

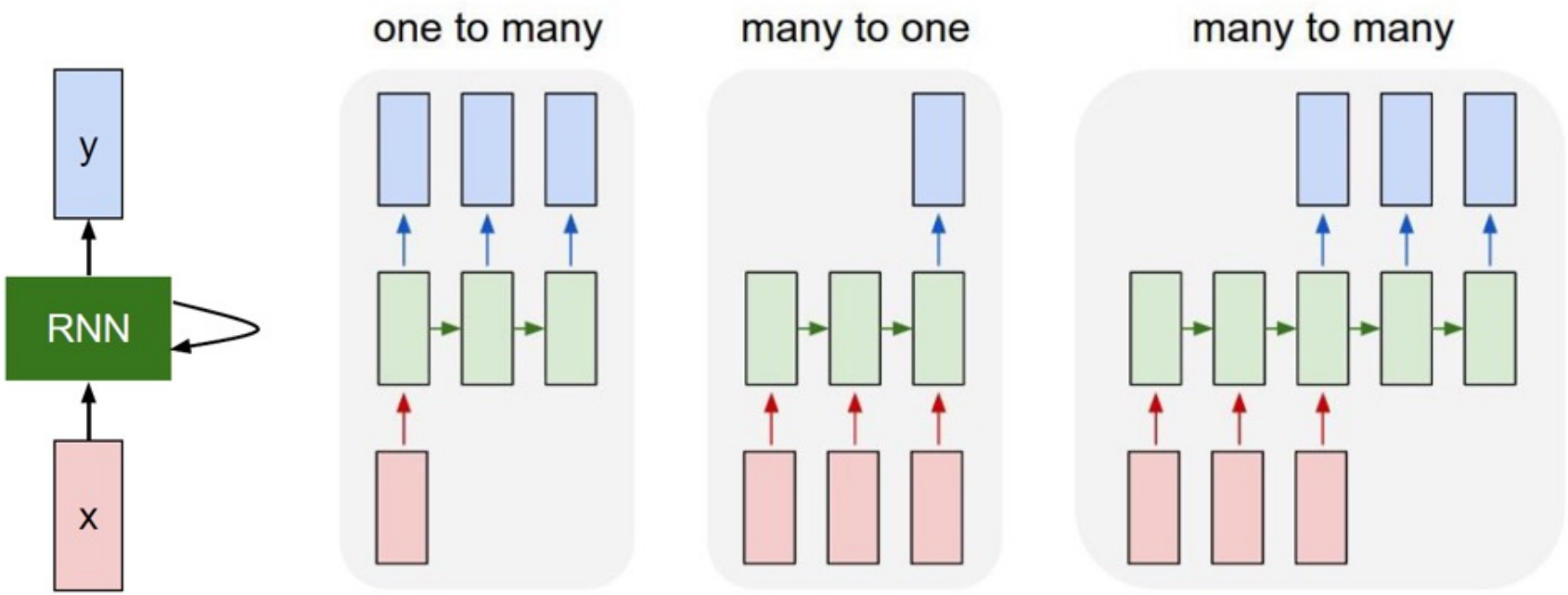
$$h_t = \tanh(W_{hh} h_{t-1} + W_{xh} x_t + b_h)$$

Y: Output

H: Hidden Layer

X: Input

RNN Structure



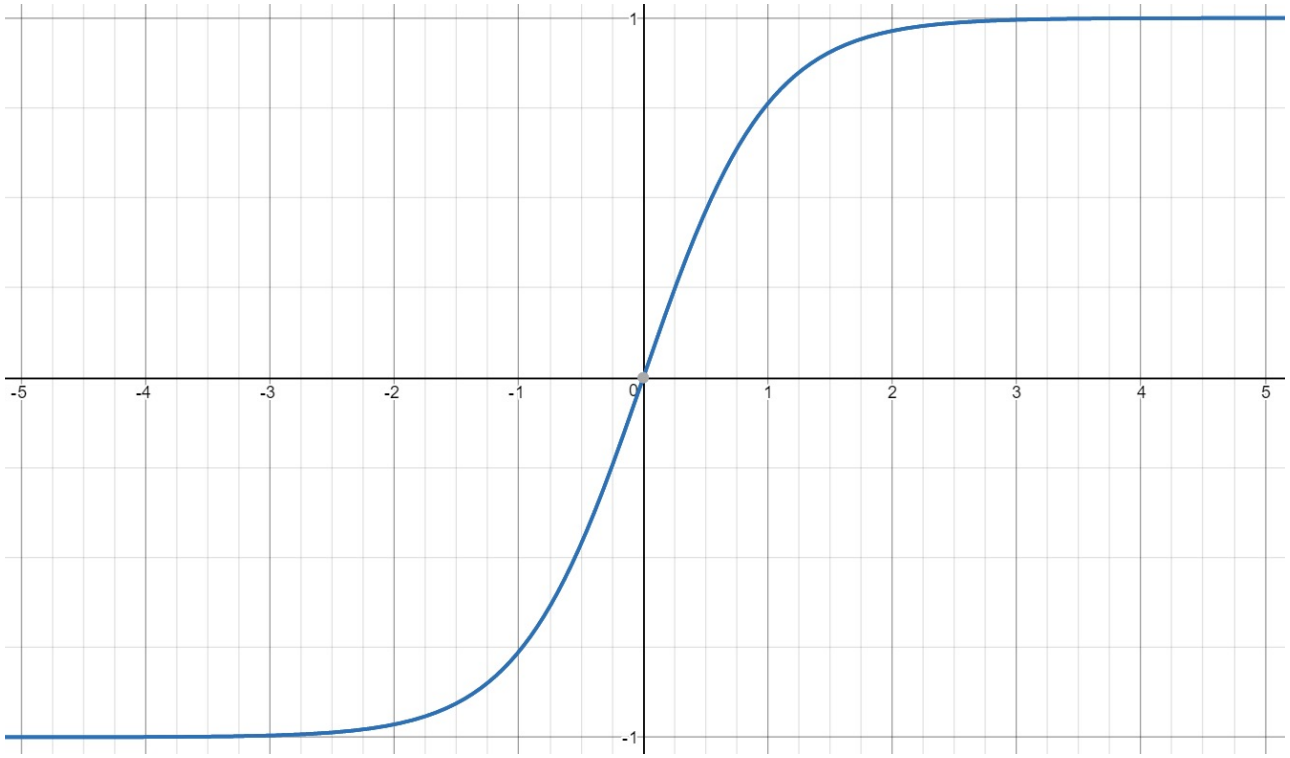
<기본구조>

<사진설명 붙이기>
사진 → 단어들

<감성분석>
단어들 → 감성점수

<번역>
단어들 → 단어들

Activation function



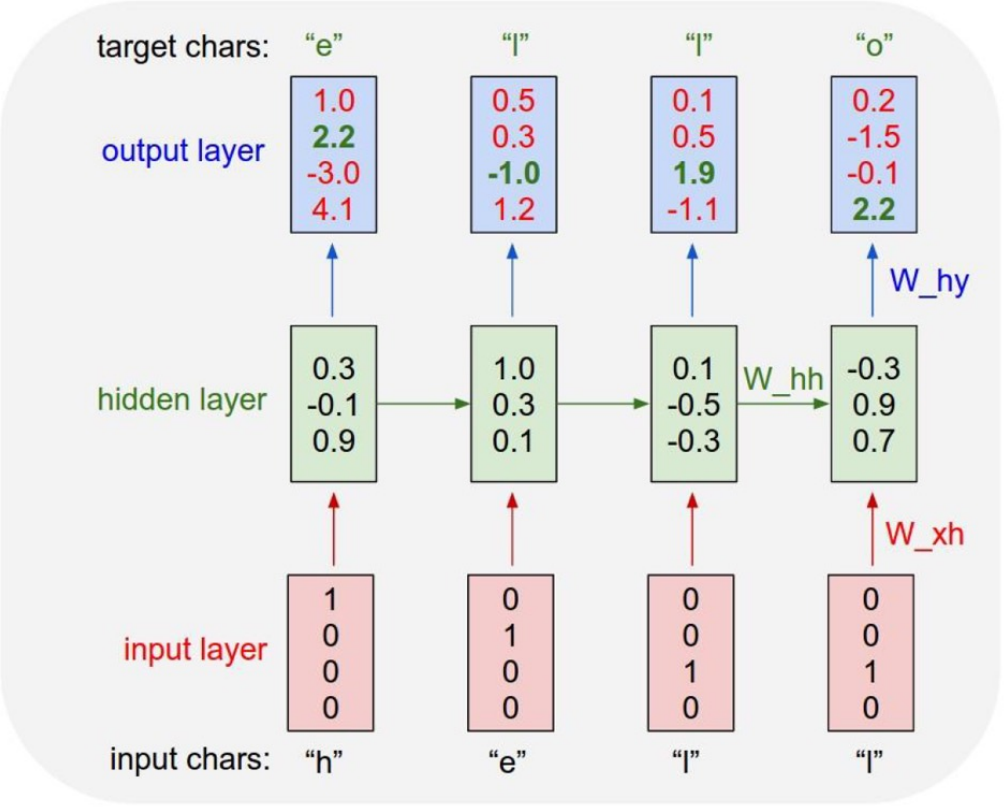
- tanh
  - Hyperbolic tangent function
  - Rescaled and shifted **sigmoid function**
  - Faster convergence

  - [-1. 1]
  - [-5, 5]

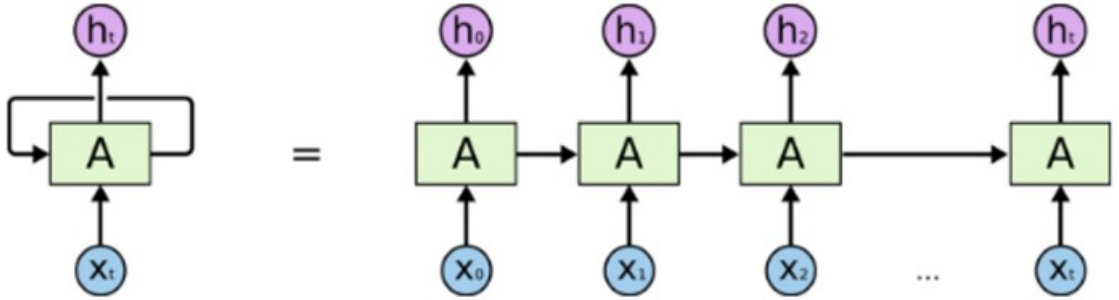
RNN Example

Text Estimation

Character Estimation



- Predicting a next character



An unrolled recurrent neural network.

Example

- http://localhost:8888/notebooks/DeepLearning_Study/tensorflow_rnn/tf_rnn_example.ipynb

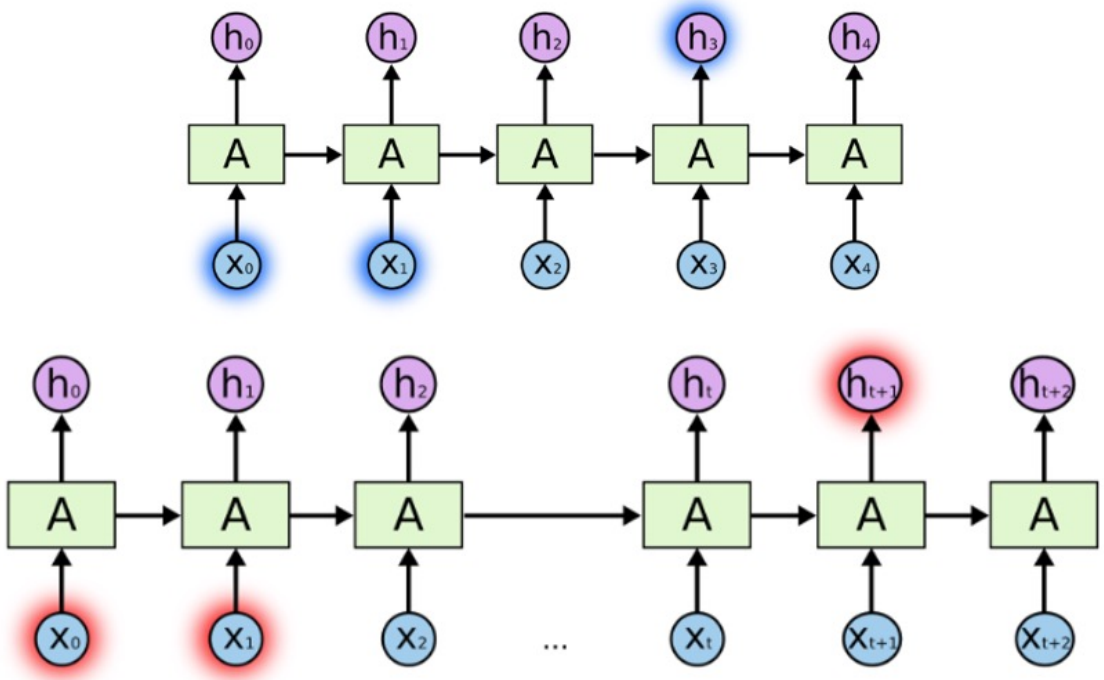
- https://www.tensorflow.org/tutorials/text/text_generation

LSTM

Long Short Term Memory

Limit of RNN

- **Periodic** information
- Gradient Vanishing/Exploding



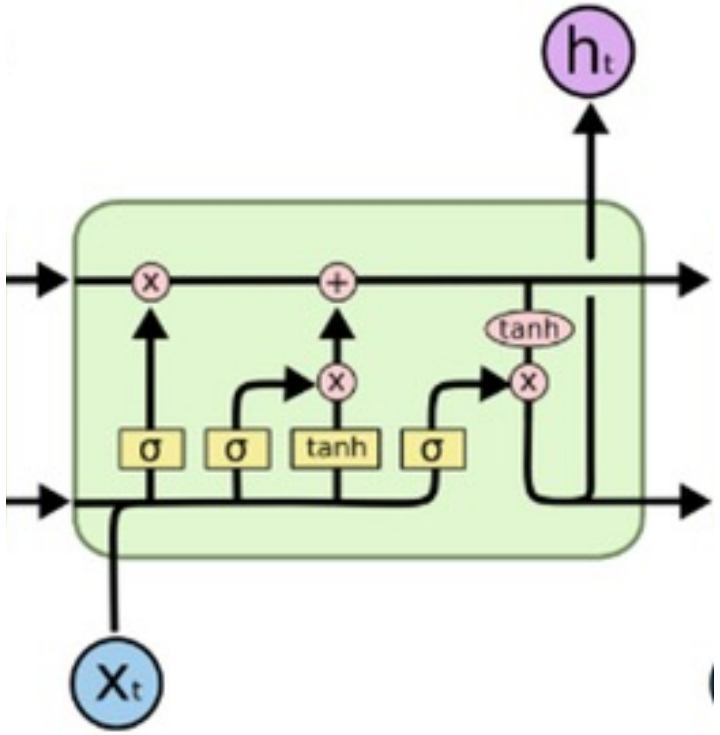
<관련 정보와 그 정보를 사용하는 지점 사이 거리가 멀 경우 RNN 학습능력 저하>

LSTM

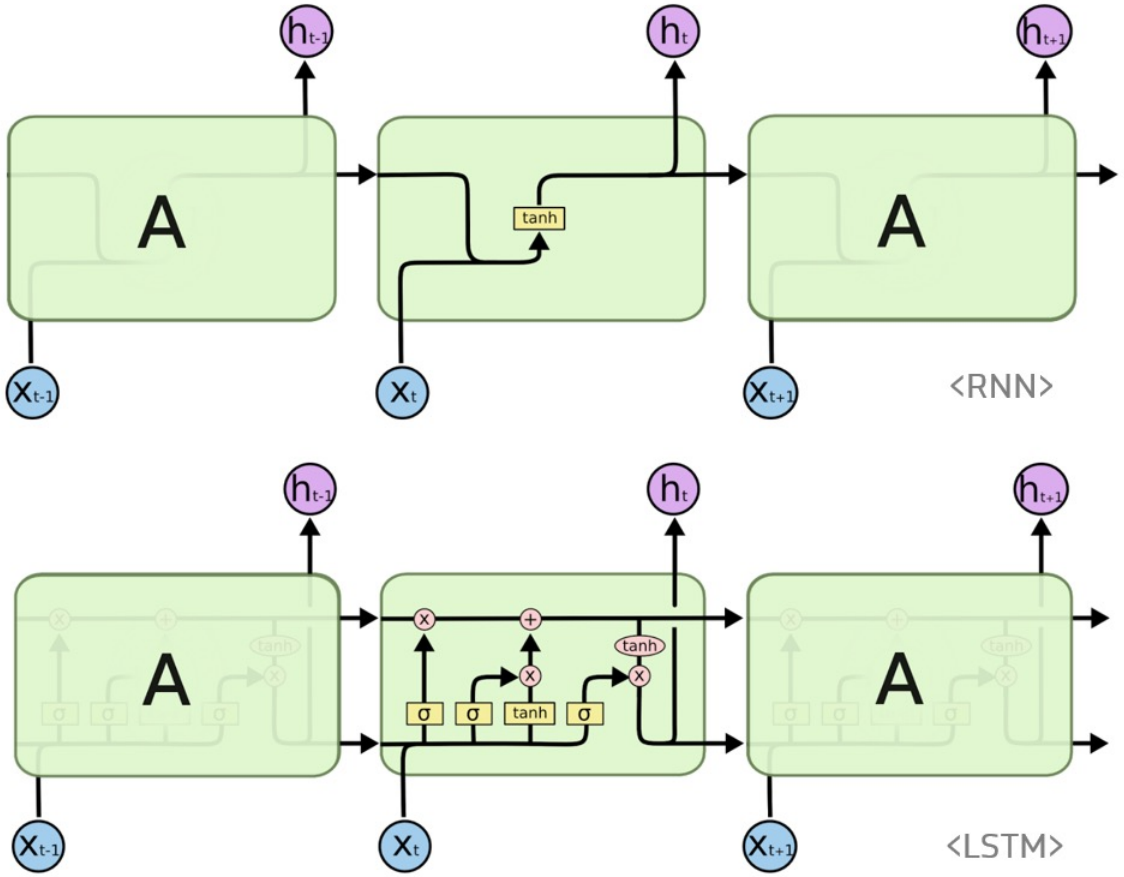
- Long Short Term Memory

- **Cell State = Memory**
  - Forget gate
  - Input gate
  - Output gate

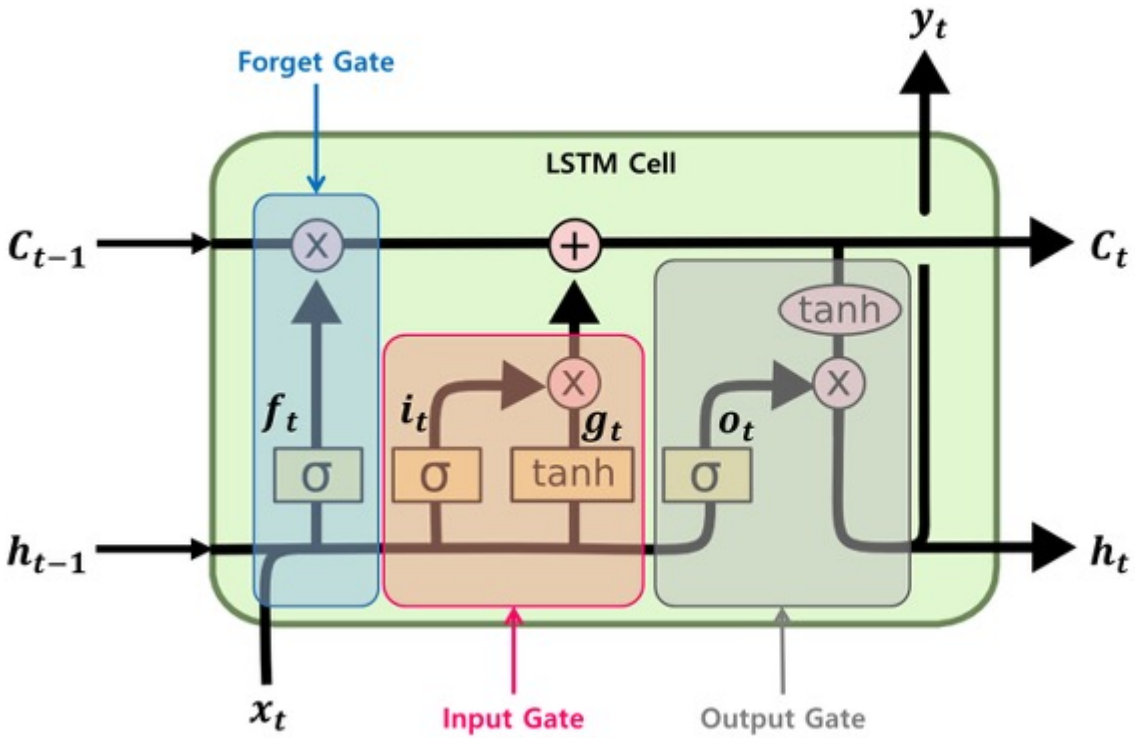
- **Hidden Layer** assistor



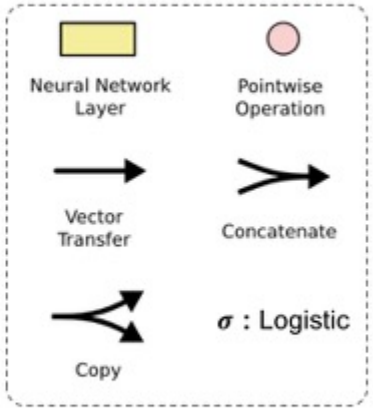
LSTM


h_{t-1}
h_t
h_{t+1}
A
tanh
A
x_{t-1}
x_t
x_{t+1}
<RNN>
h_{t-1}
h_t
h_{t+1}
A
σ
σ
tanh
σ
tanh
A
x_{t-1}
x_t
x_{t+1}
<LSTM>

LSTM



$$f_t = \sigma(W_{xh\_f}x_t + W_{hh\_f}h_{t-1} + b_{h\_f})$$
$$i_t = \sigma(W_{xh\_i}x_t + W_{hh\_i}h_{t-1} + b_{h\_i})$$
$$o_t = \sigma(W_{xh\_o}x_t + W_{hh\_o}h_{t-1} + b_{h\_o})$$
$$g_t = \tanh(W_{xh\_g}x_t + W_{hh\_g}h_{t-1} + b_{h\_g})$$
$$c_t = f_t \odot c_{t-1} + i_t \odot g_t$$
$$h_t = o_t \odot \tanh(c_t)$$



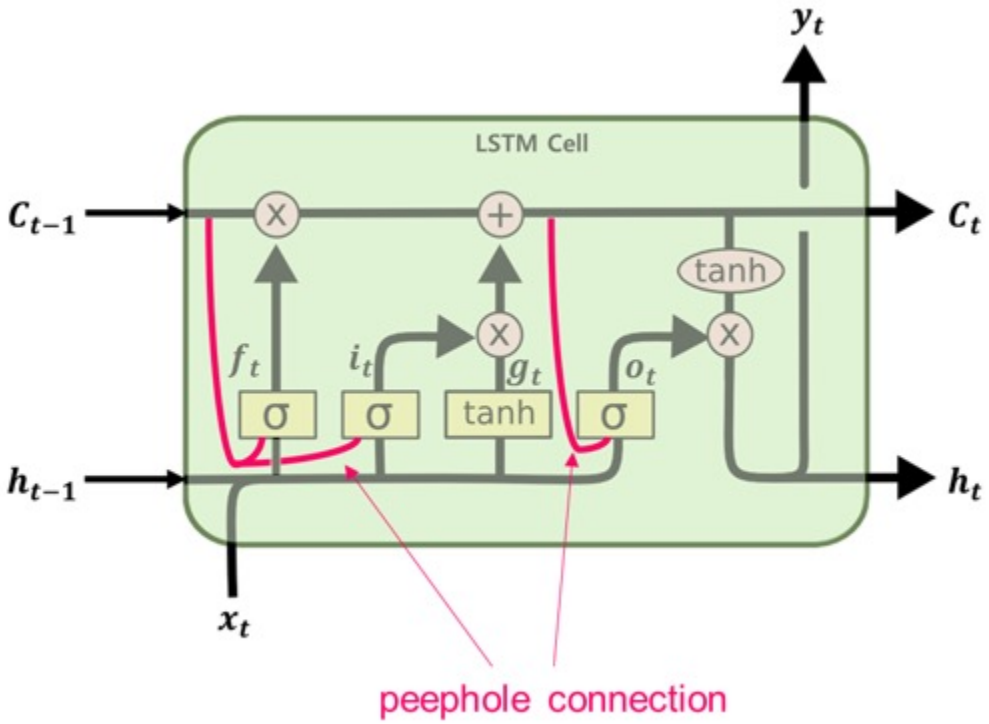
C : Long-term State
H : Short-term State

Peephole Connection

- Paper - Recurrent Nets that and Count
- Keypoints
  - Modified LSTM
  - obtaining **contextual information**

- Standard LSTM
  - **Short-term state changes Long-term state**
  - Long-term state makes output

- Improved LSTM
  - **Short-term state and Long-term state make output**



GRU

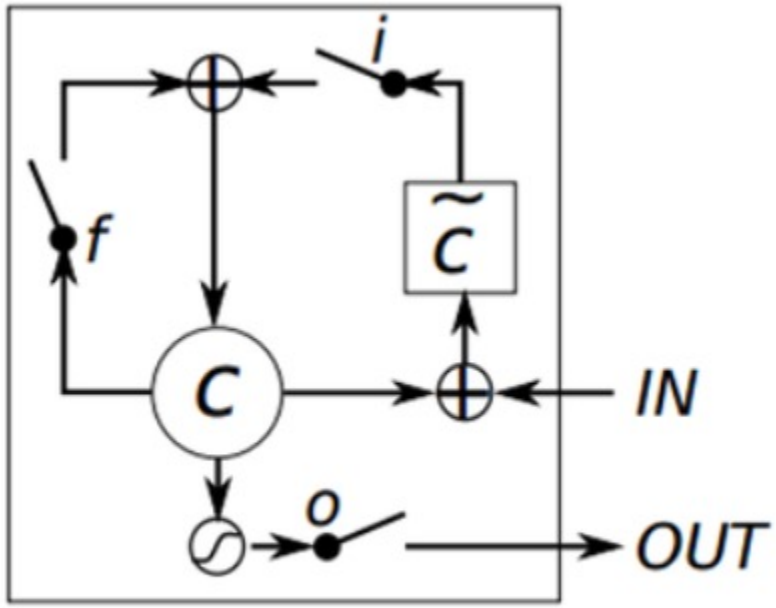
Gated Recurrent Unit

GRU

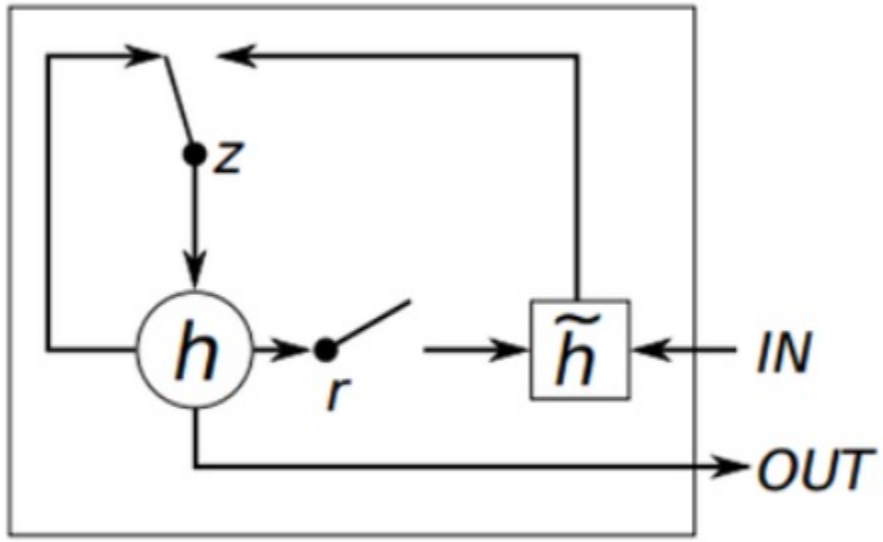
- Paper - Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. KyungHyun Cho

- Gated Recurrent Unit

LSTM vs GRU

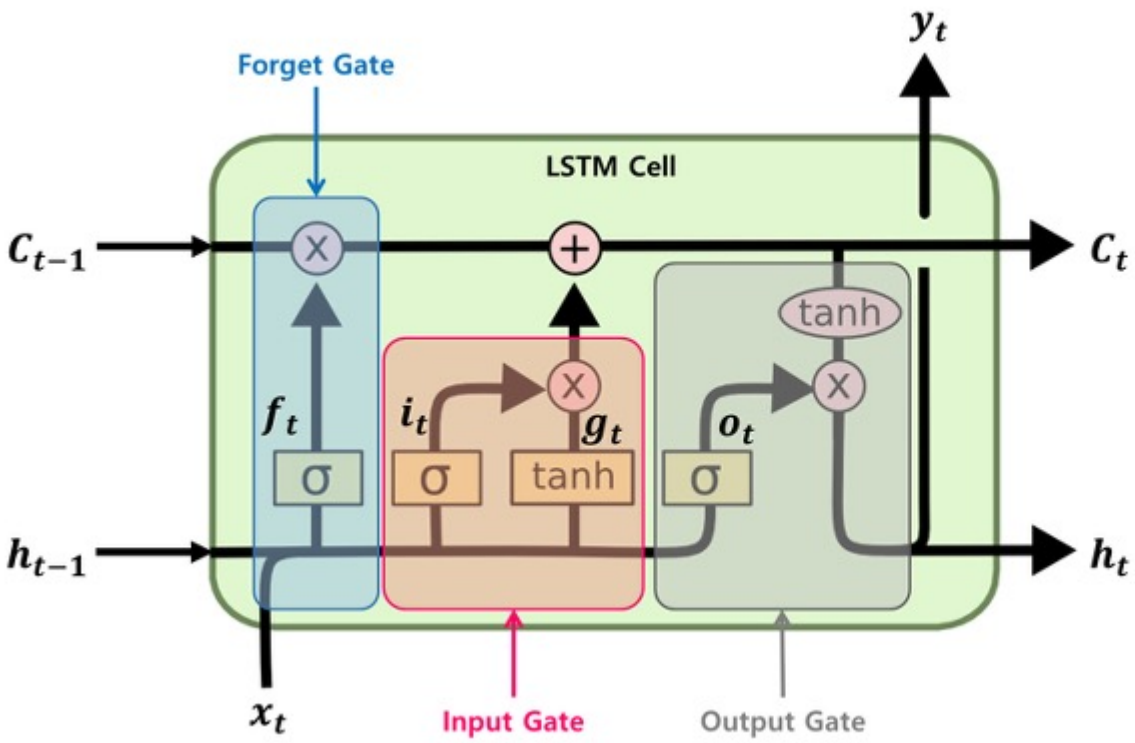


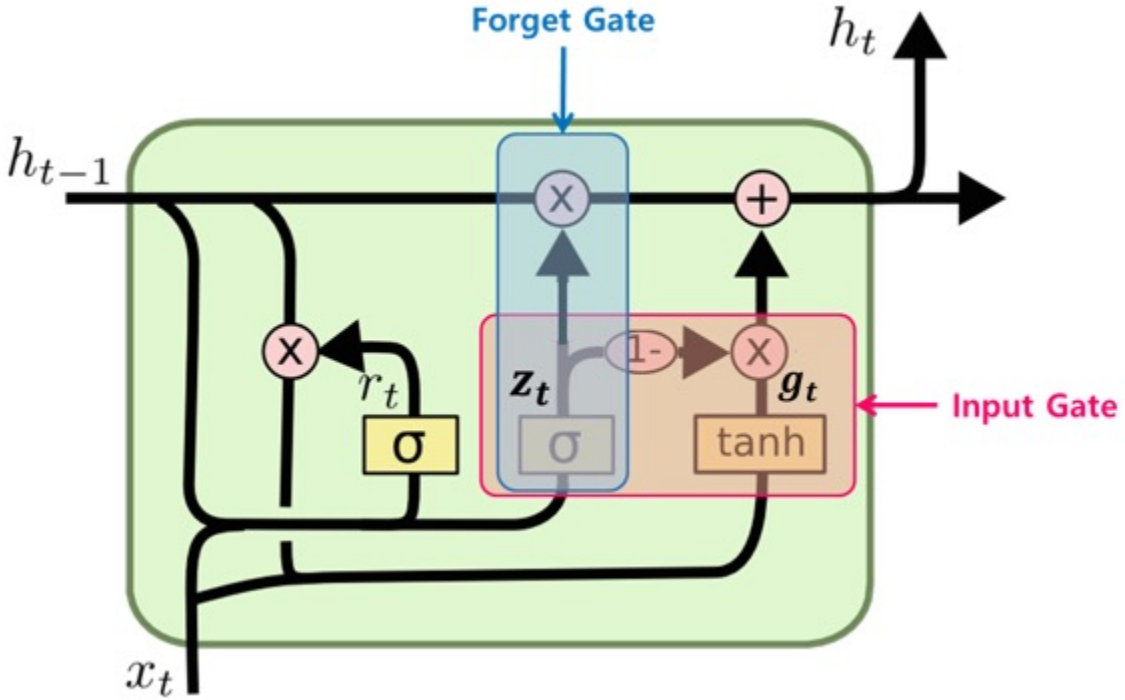
(a) Long Short-Term Memory



(b) Gated Recurrent Unit

LSTM vs GRU

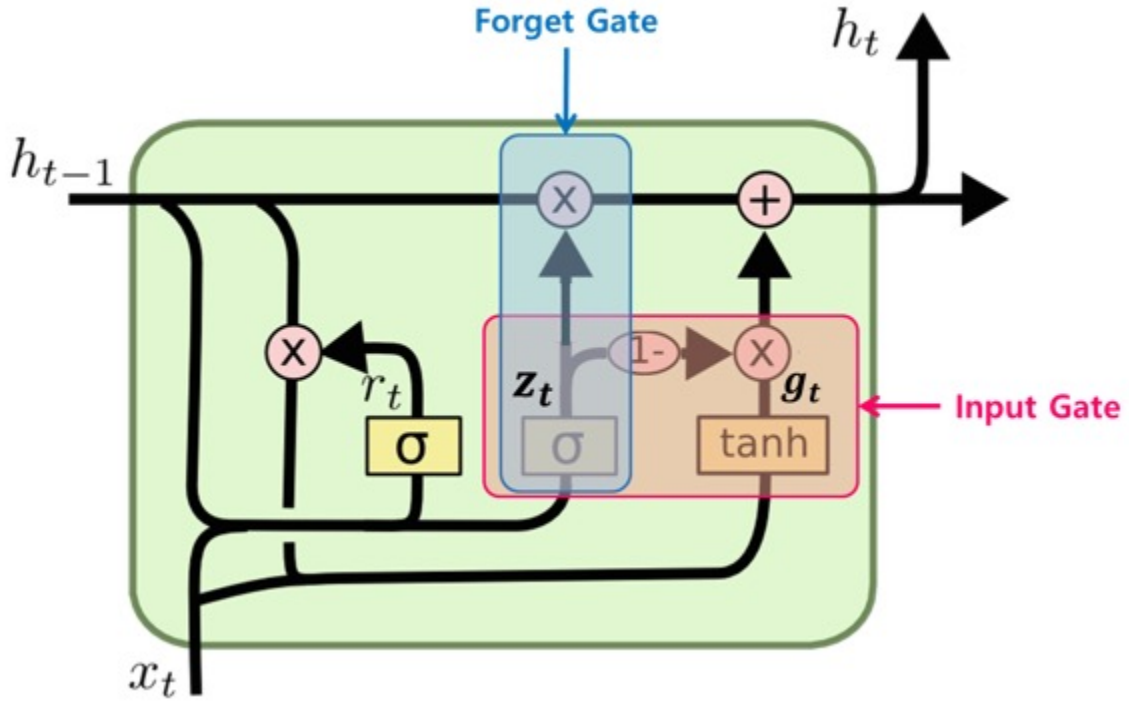

Forget Gate
LSTM Cell
y_t
C_{t-1}
C_t
f_t
i_t
g_t
o_t
σ
tanh
h_{t-1}
h_t
x_t
Input Gate
Output Gate


Forget Gate
h_t
h_{t-1}
r_t
z_t
1-
g_t
σ
tanh
Input Gate
x_t

GRU

- Simple
- Compliant Performance
- R: What to use
- Z: What to Remember

$$\mathbf{r}_t = \sigma\left(\mathbf{W}_{xr}^T \cdot \mathbf{x}_t + \mathbf{W}_{hr}^T \cdot \mathbf{h}_{t-1} + \mathbf{b}_r\right)$$
$$\mathbf{z}_t = \sigma\left(\mathbf{W}_{xz}^T \cdot \mathbf{x}_t + \mathbf{W}_{hz}^T \cdot \mathbf{h}_{t-1} + \mathbf{b}_z\right)$$
$$\mathbf{g}_t = \tanh\left(\mathbf{W}_{xg}^T \cdot \mathbf{x}_t + \mathbf{W}_{hg}^T \cdot (\mathbf{r}_t \otimes \mathbf{h}_{t-1}) + \mathbf{b}_g\right)$$
$$\mathbf{h}_t = \mathbf{z}_t \otimes \mathbf{h}_{t-1} + (1 - \mathbf{z}_t) \otimes \mathbf{g}_t$$



Next Presentation

- Encoder and Decoder
- Attention
- Transformer

Next Presentation

- Encoder and Decoder
  - Seq2seq
  - Paper - Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. KyungHyun Cho
- Attention
- Transformer

References

References

- DL Functions and Techniques
  - https://ratsgo.github.io/deep%20learning/2017/04/22/NNtricks/
- DL Propagation Concepts
  - https://ratsgo.github.io/deep%20learning/2017/05/14/backprop/
- RNN, LSTM and GRU
  - https://ratsgo.github.io/natural%20language%20processing/2017/03/09/rnnlstm/
  - https://excelsior-cjh.tistory.com/185