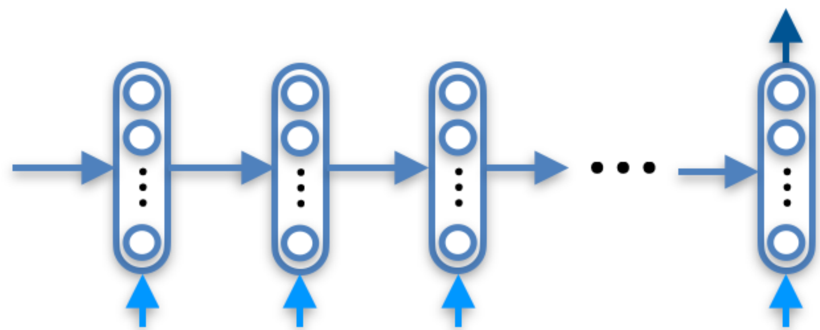
LSTM and GRU

Debapriyo Majumdar

debapriyo@isical.ac.in

Shortcomings of vanilla RNN



- The network cannot remember “enough” if the sequence is too long
- *Example*: Samuel spent his childhood in Spain. Then he lived in France, Germany and England. However, he can still speak Spanish.

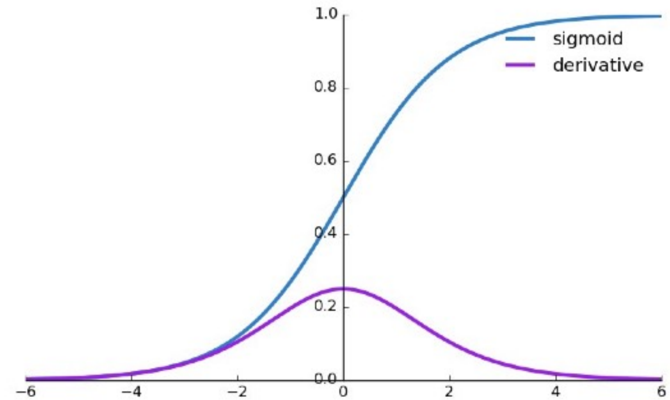
Shortcomings of vanilla RNN: vanishing gradient

Activation to the next cell →

$$a^{<t>} = g(W_a a^{<t-1>} + W_x x^{<t>} + b)$$

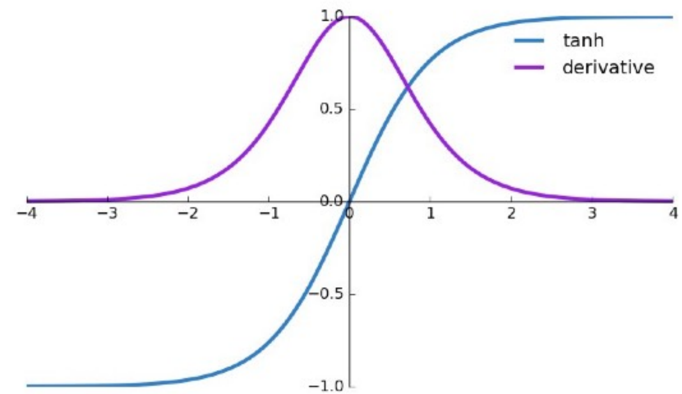
← Backpropagation from the next cell

$$\frac{\partial L}{\partial a^{<t-1>}} = g' \cdot W_a^T \frac{\partial L}{\partial a^{<t>}}$$



Vanishing gradient problem

- Long chain of *multiplication* with W_a^T and g' (derivative of the activation)
- Both $\tanh$ and sigmoid have small derivatives
- ReLU is not used in RNN because the values can explode

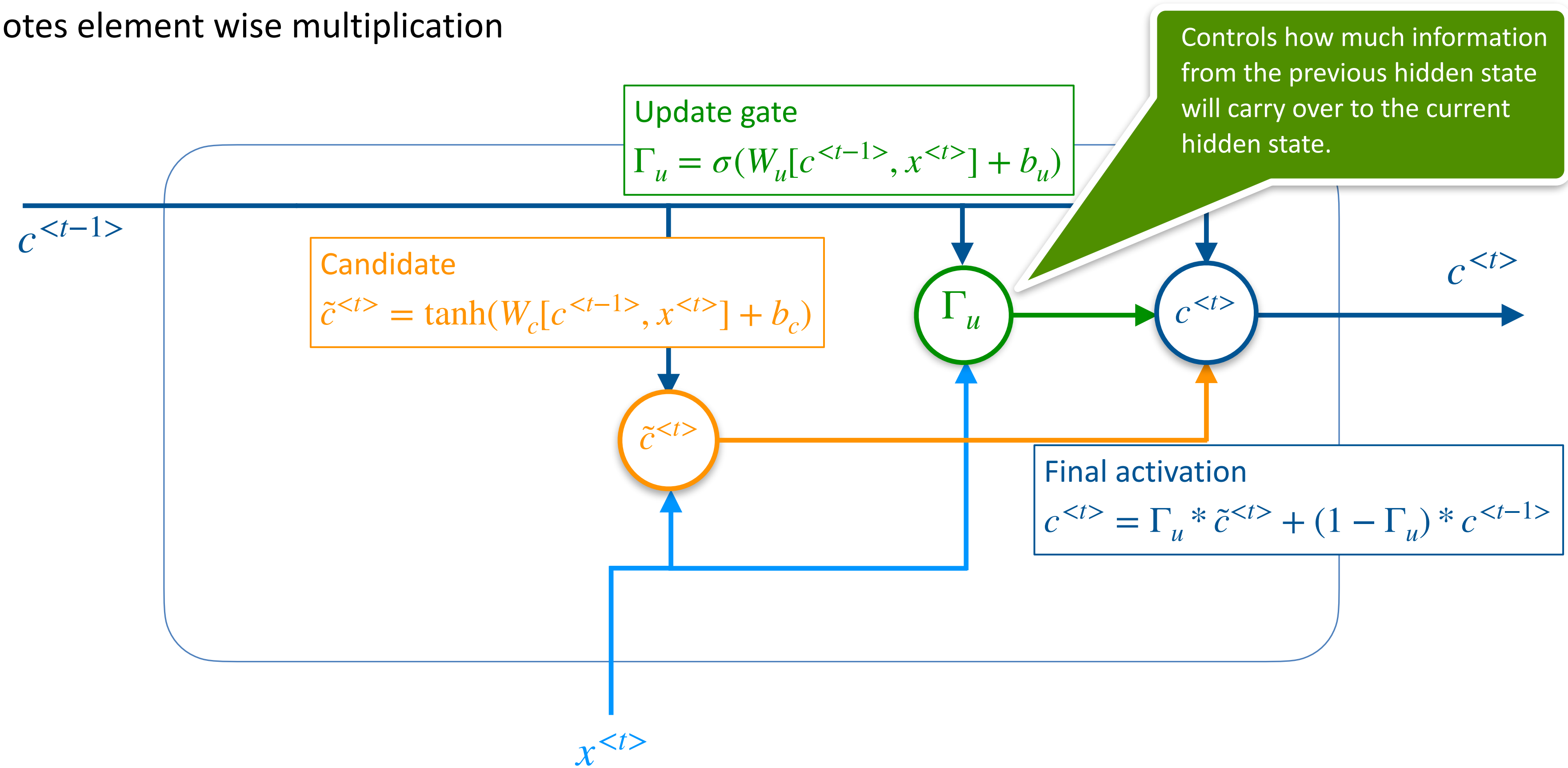


Gated Recurrent Unit (GRU): Simplified

Notation: memory cell $c^{<t>}$

For GRU, $c^{<t>}$ is same as activation $a^{<t>}$ as before

* denotes element wise multiplication

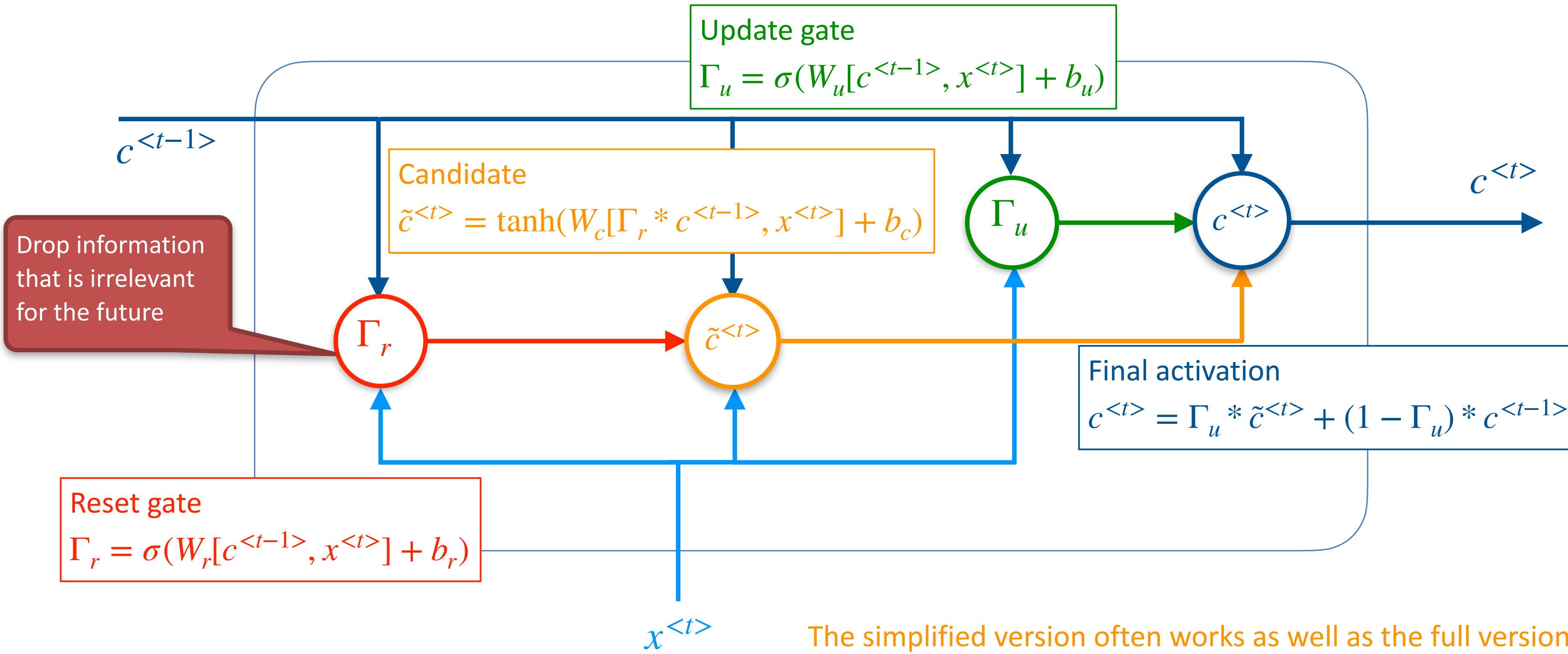


Gated Recurrent Unit (GRU)

Notation: memory cell $c^{<t>}$

For GRU, $c^{<t>}$ is same as activation $a^{<t>}$ as before

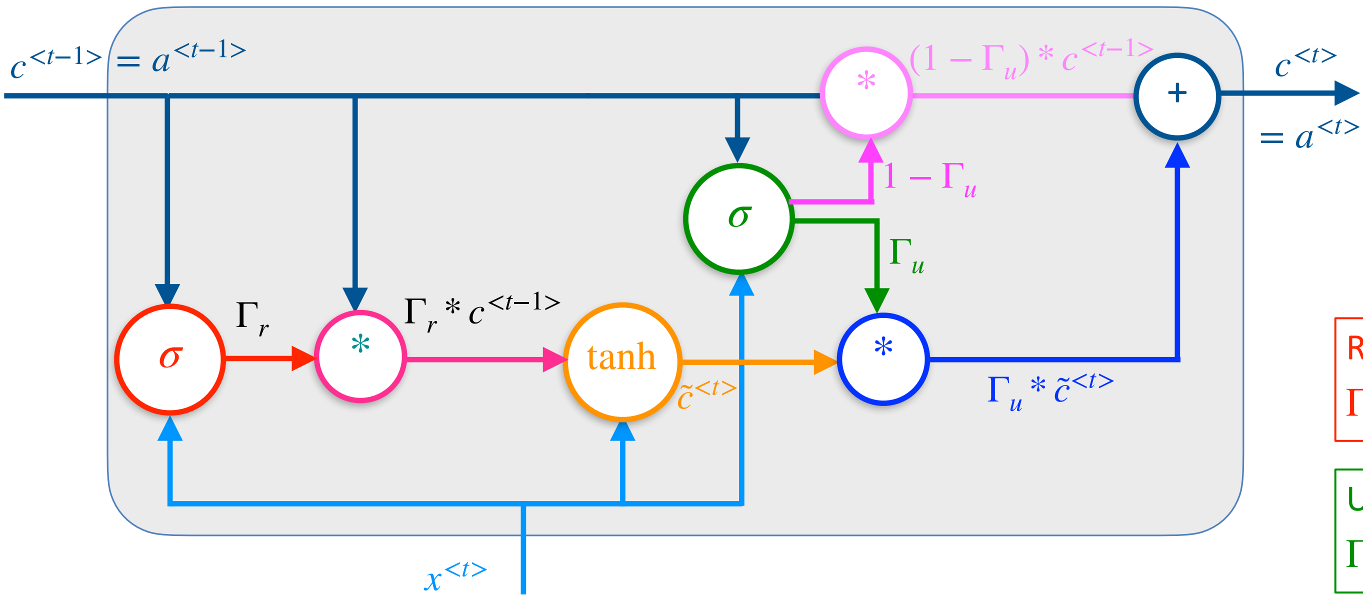
* denotes element wise multiplication



The simplified version often works as well as the full version

Gated Recurrent Unit (GRU)

Notation: memory cell $c^{<t>}$ (for GRU, $c^{<t>}$ = activation $a^{<t>}$), * denotes element wise multiplication



Reset gate

$\Gamma_r = \sigma(W_r[c^{<t-1>}, x^{<t>}] + b_r)$

Update gate

$\Gamma_u = \sigma(W_u[c^{<t-1>}, x^{<t>}] + b_u)$

Candidate

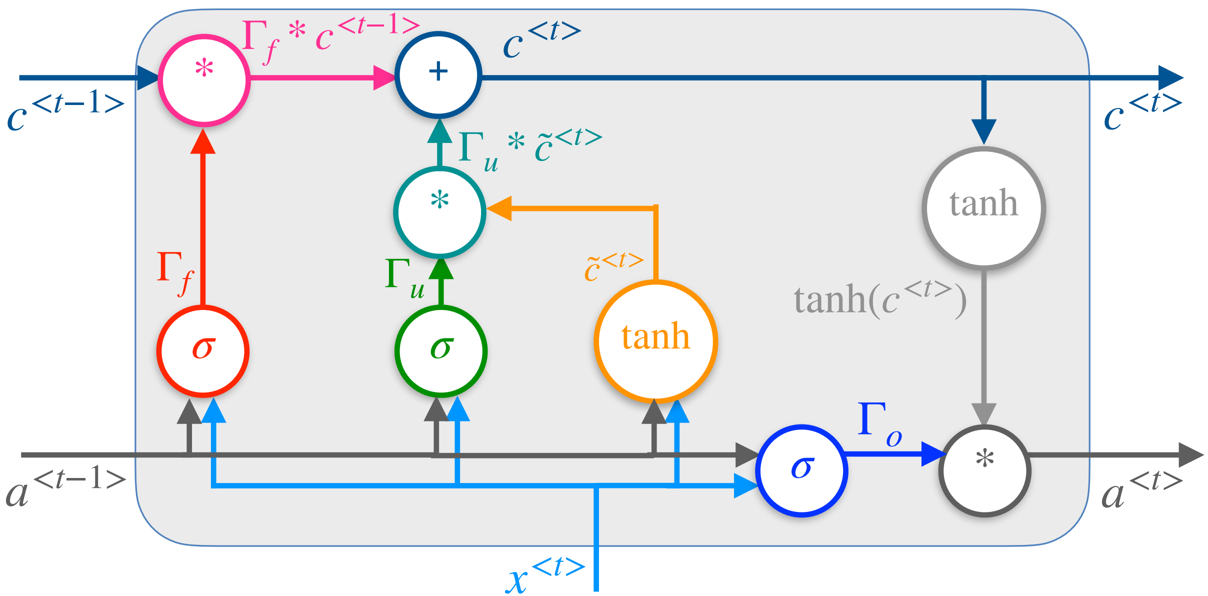
$\tilde{c}^{<t>} = \tanh(W_c[\Gamma_r * c^{<t-1>}, x^{<t>}] + b_c)$

Final activation to the next cell

$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + (1-\Gamma_u) * c^{<t-1>}$

Long Short-Term Memory (LSTM)

Notation: memory cell $c^{<t>}$, activation $a^{<t>}$, * denotes element wise multiplication



Forget gate

$\Gamma_f = \sigma(W_f[a^{<t-1>}, x^{<t>}] + b_f)$

Update gate

$\Gamma_u = \sigma(W_u[a^{<t-1>}, x^{<t>}] + b_u)$

Candidate activation

$\tilde{c}^{<t>} = \tanh(W_c[a^{<t-1>}, x^{<t>}] + b_c)$

Memory to the next cell

$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + \Gamma_f * c^{<t-1>}$

Output gate

$\Gamma_o = \sigma(W_o[a^{<t-1>}, x^{<t>}] + b_o)$

Activation to the next cell

$a^{<t>} = \Gamma_o * \tanh(c^{<t>})$

- Memory from the previous cell pass through the gate Γ_f

LSTM vs GRU

LSTM	GRU
More widely used	Recently gained popularity
More powerful	Simpler, but works almost as well as LSTM
Computationally expensive	Computationally cheaper, so deeper networks can be trained

References

- Andrew Ng’s lectures on *Sequence Models*: www.coursera.org/learn/nlp-sequence-models
- Chris Manning, Abigail See and other TAs. *Natural Language Processing with Deep Learning.* Stanford University Course (CS224n), Winter 2019. web.stanford.edu/class/archive/cs/cs224n/cs224n.1194/
- Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016. www.deeplearningbook.org
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. Neural Comput. 9, 8 (November 15, 1997), 1735–1780. DOI:https://doi.org/10.1162/neco.1997.9.8.1735
- Nir Arbel. *How LSTM networks solve the problem of vanishing gradients.* 2018. medium.com/datadriveninvestor/how-do-lstm-networks-solve-the-problem-of-vanishing-gradients-a6784971a577
- Cho, Kyunghyun, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." *arXiv preprint arXiv:1406.1078* (2014).