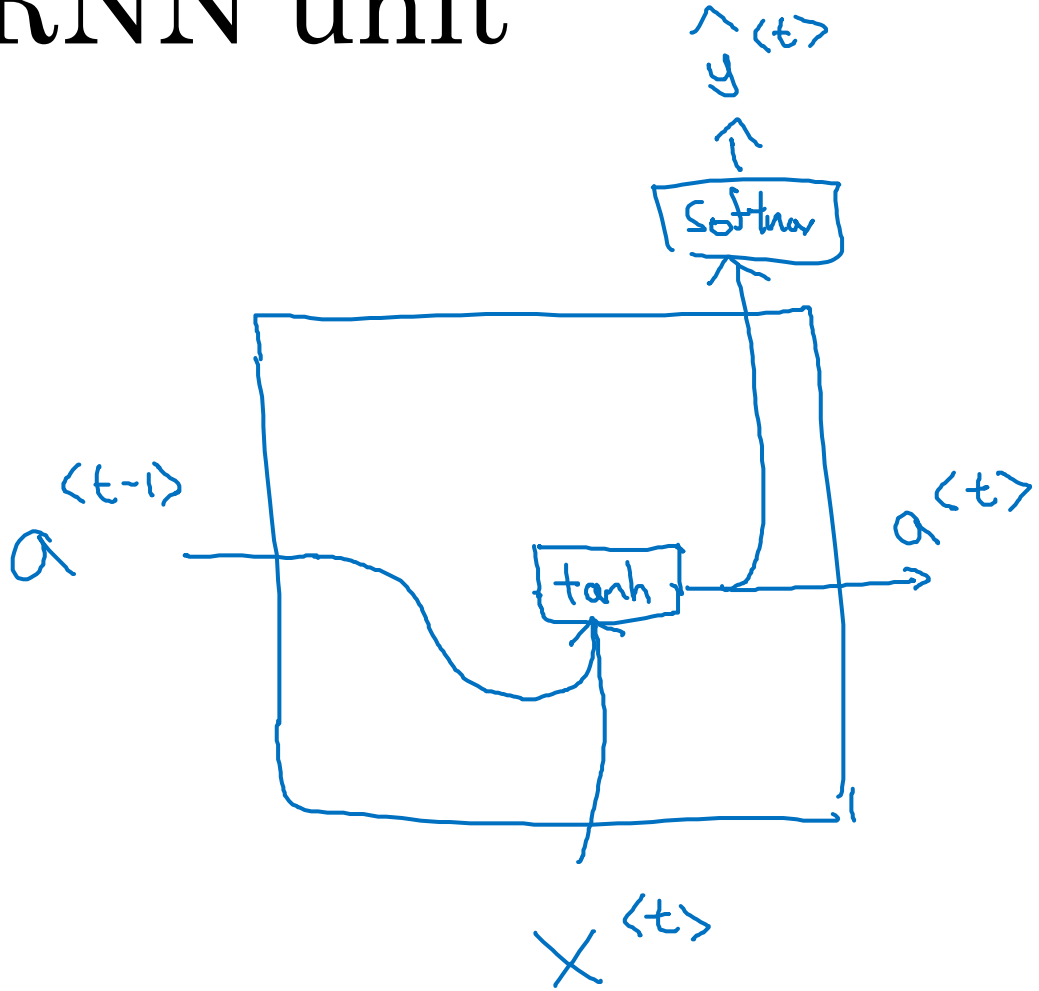
deeplearning.ai

Recurrent Neural Networks

Gated Recurrent Unit (GRU)

RNN unit



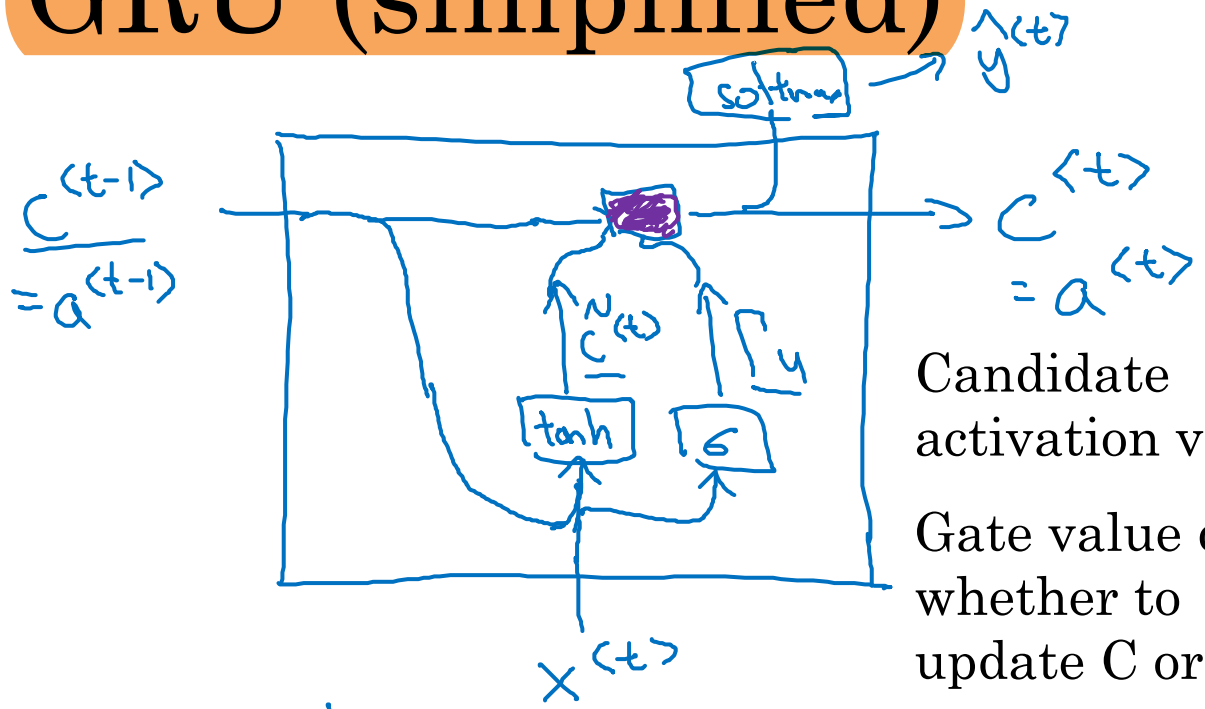
RNN unit without Gated Recurrent Unit

Activation at time step t

tanh

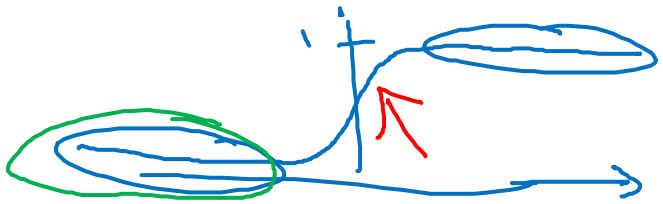
$$a^{<t>} = g(W_a[a^{<t-1>}, x^{<t>}] + b_a)$$

GRU (simplified)



Candidate activation value

Gate value decide whether to update C or not



c = memory cell

$c^{\langle t \rangle} = a^{\langle t \rangle}$

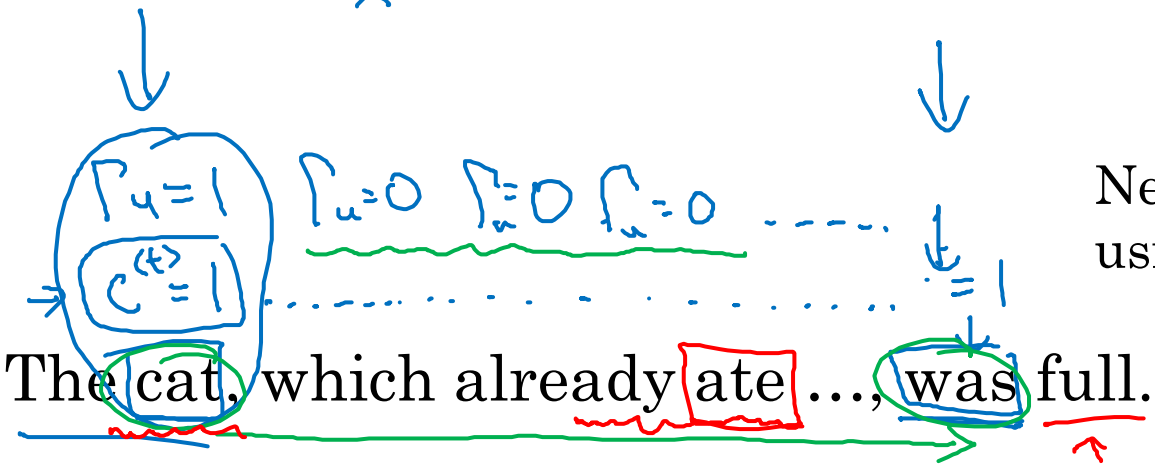
$\tilde{c}^{\langle t \rangle} = \tanh(W_c [c^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_c)$

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

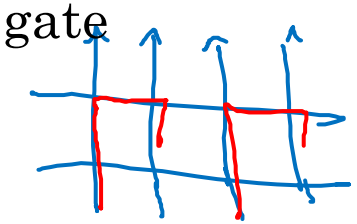
"update"

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

New C value using the gate



The cat, which already ate …, was full.



element-wise

Gate

$\Gamma_u = 0.000001$

This value is very small, hence prevents the vanishing gradient problem

[Cho et al., 2014. On the properties of neural machine translation: Encoder-decoder approaches]

[Chung et al., 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling]

Full GRU

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

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

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

Gate u controls whether to update C or not

Gate r controls the relevance between C and C~

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

The cat, which ate already, was full.

Andrew Ng