Recurrent Neural Network (RNN)

Gated Recurrent Unit (GRU)

Long Short Term Memory (LSTM)

By

Prof. Khaled Mostafa El Sayed

2019

[1] Recurrent Neural Network (RNN)

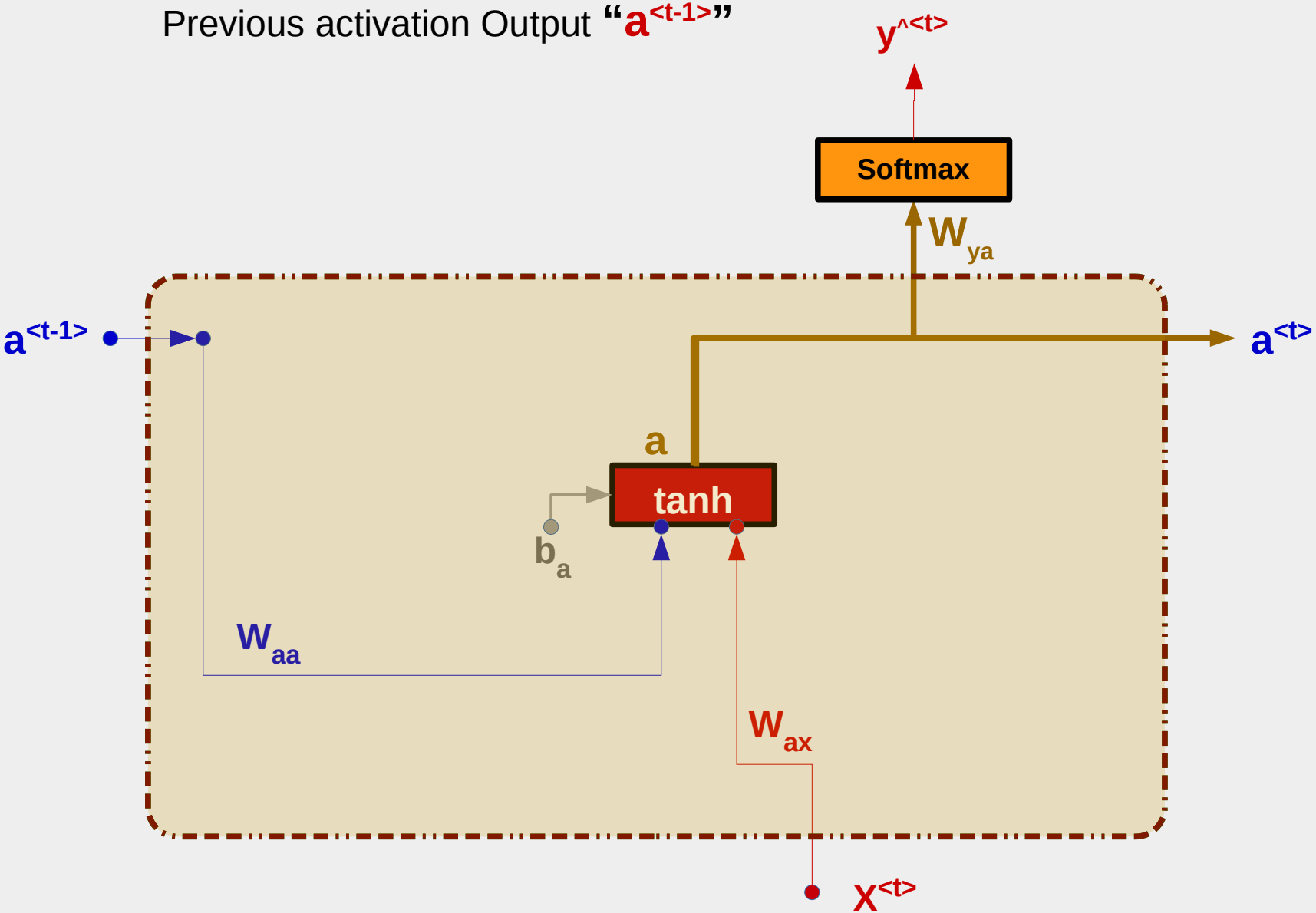
By Prof. Khaled M. El Sayed *Khaledms@fci-cu.edu.eg*

Recurrent Neural Network (RNN)

Output of Activation Function at time t; "$a^{<t>}$" depends on BOTH:-

Input "$X^{<t>}$" and

Previous activation Output "$a^{<t-1>}$"



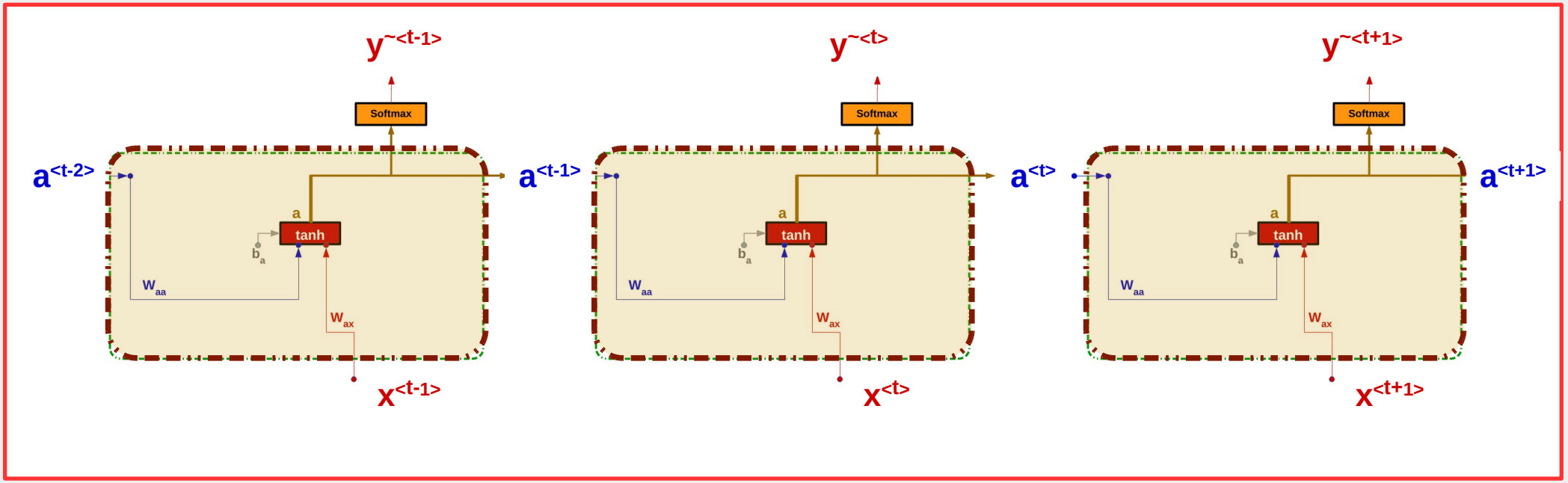
Recurrent Neural Network (RNN)

Output of Activation Function at time t; **"$a^{<t>}$"** depends on BOTH:-

Input **"$X^{<t>}$"** and

Previous activation Output **"$a^{<t-1>}$"**

Inputs at time t-1 , t , t+1



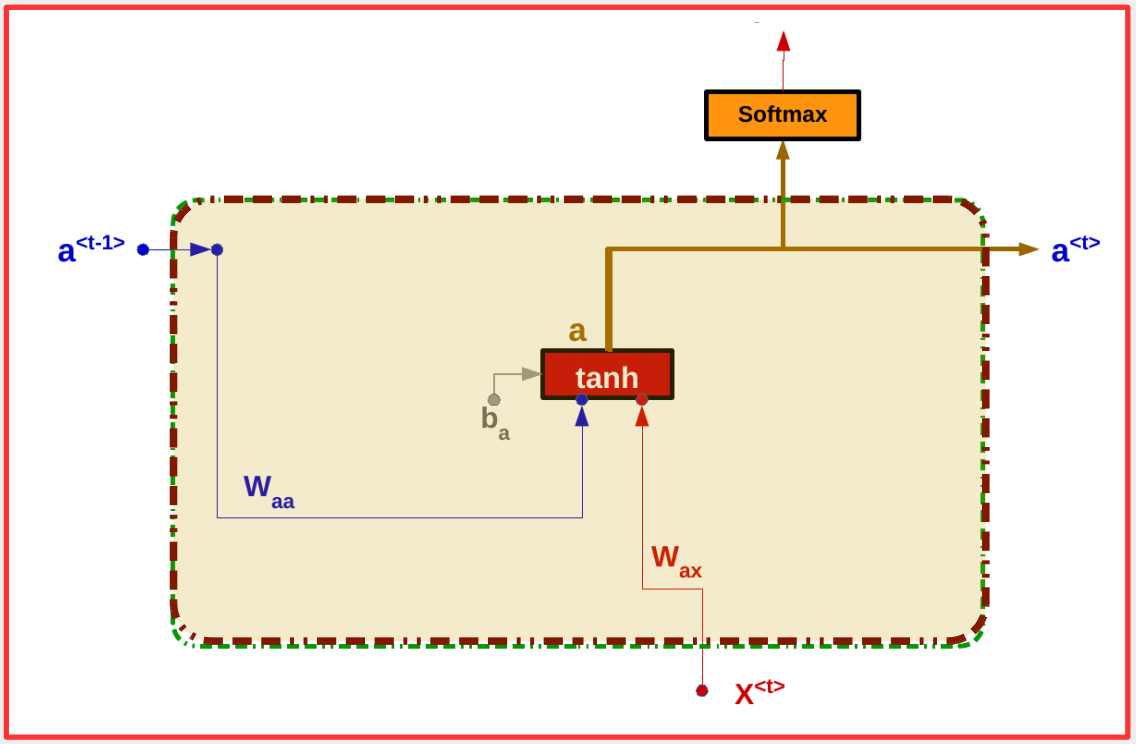
Recurrent Neural Network (RNN)

From Prev. Current I/P

$$a^{<1>} = F^{n}\left(W_{aa}a^{<0>} + W_{ax}x^{<1>} + b_{a}\right)$$

$$y\hat{}^{<1>} = F^{n}\left(W_{ya}\,a^{<1>} + b_{y}\right)$$

Non Linearity



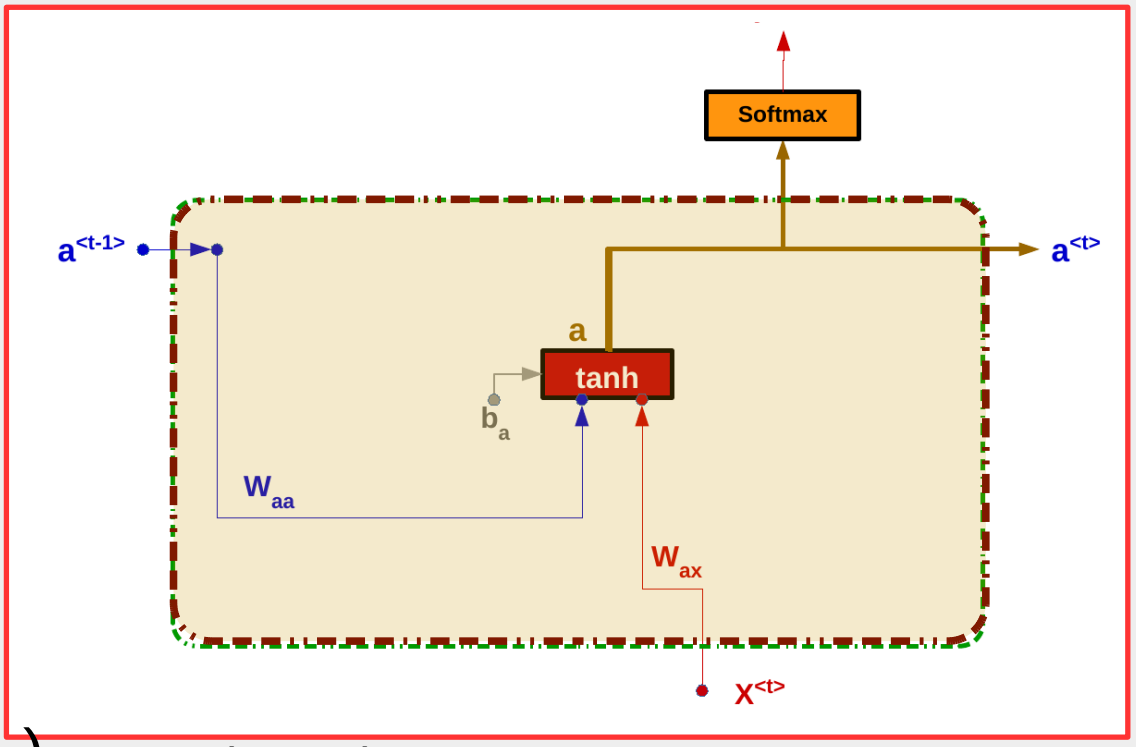
Recurrent Neural Network (RNN)

From Prev. Current I/P

$$a^{<1>} = F^{n}\left(W_{aa}a^{<0>} + W_{ax}x^{<1>} + b_a\right)$$

$$y\hat{}^{<1>} = F^{n}\left(W_{ya}\,a^{<1>} + b_y\right)$$

Non Linearity



$$a^{<1>} = \tanh\left(W_{aa}a^{<0>} + W_{ax}x^{<1>} + b_a\right)$$ May be tanh, ReLU, ...

$$y\hat{}^{<1>} = \text{Softmax}\left(W_{ya}\,a^{<1>} + b_y\right)$$ Segmoid for Binary O/P, Softmax for Multi-Class O/P

$$a^{<t>} = \tanh\left(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a\right)$$ May be tanh, ReLU, ...

$$y\hat{}^{<t>} = \text{Softmax}\left(W_{ya}\,a^{<t>} + b_y\right)$$ Segmoid for Binary O/P, Softmax for Multi-Class O/P

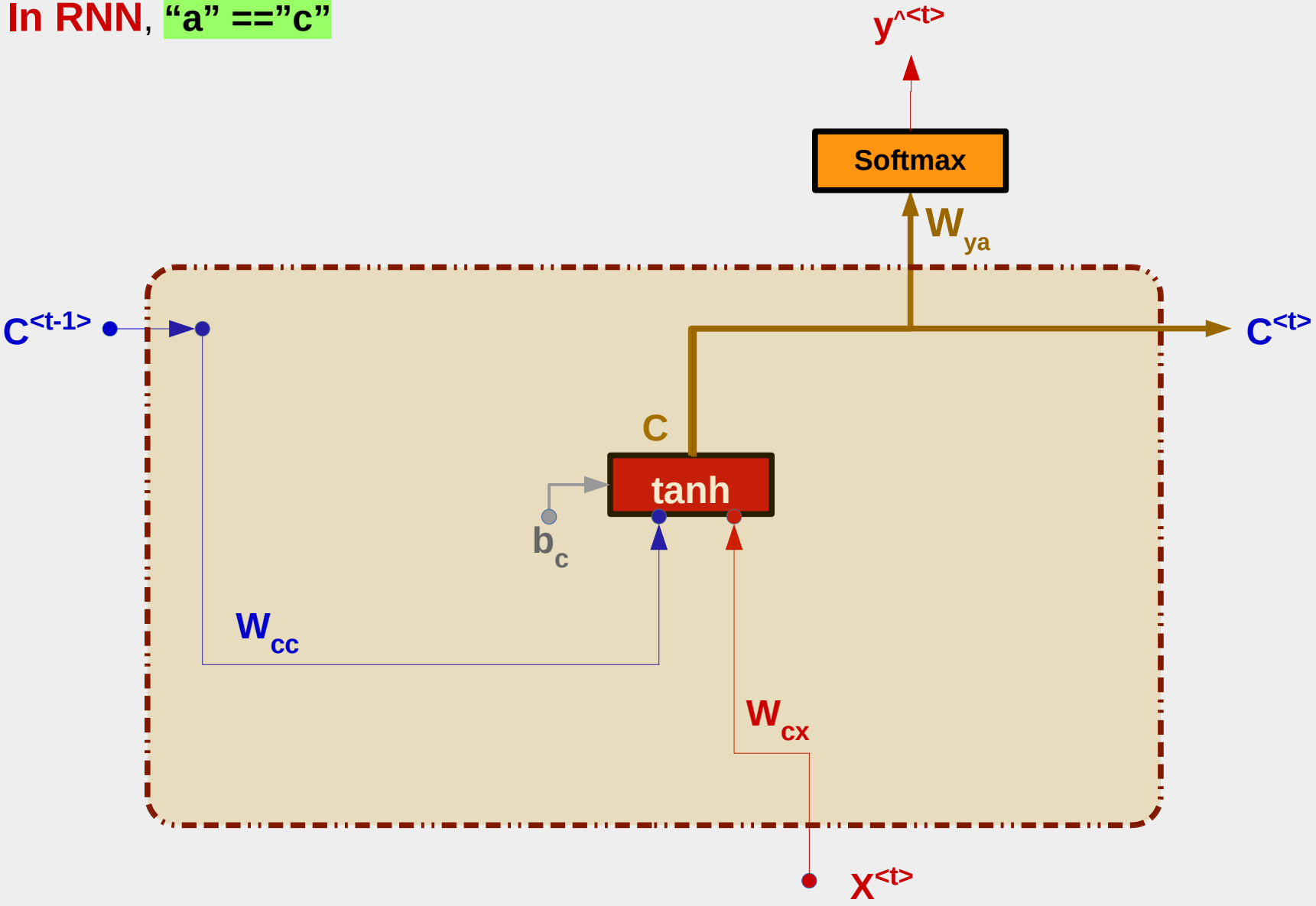
[2] Gated Recurrent Unit (GRU)

By Prof. Khaled M. El Sayed Khaledms@fci-cu.edu.eg

From RNN to GRU

Define **Memory** Cell "**C**" in addition to Output of **Activation** function "**a**".

In RNN, **"a" =="c"**



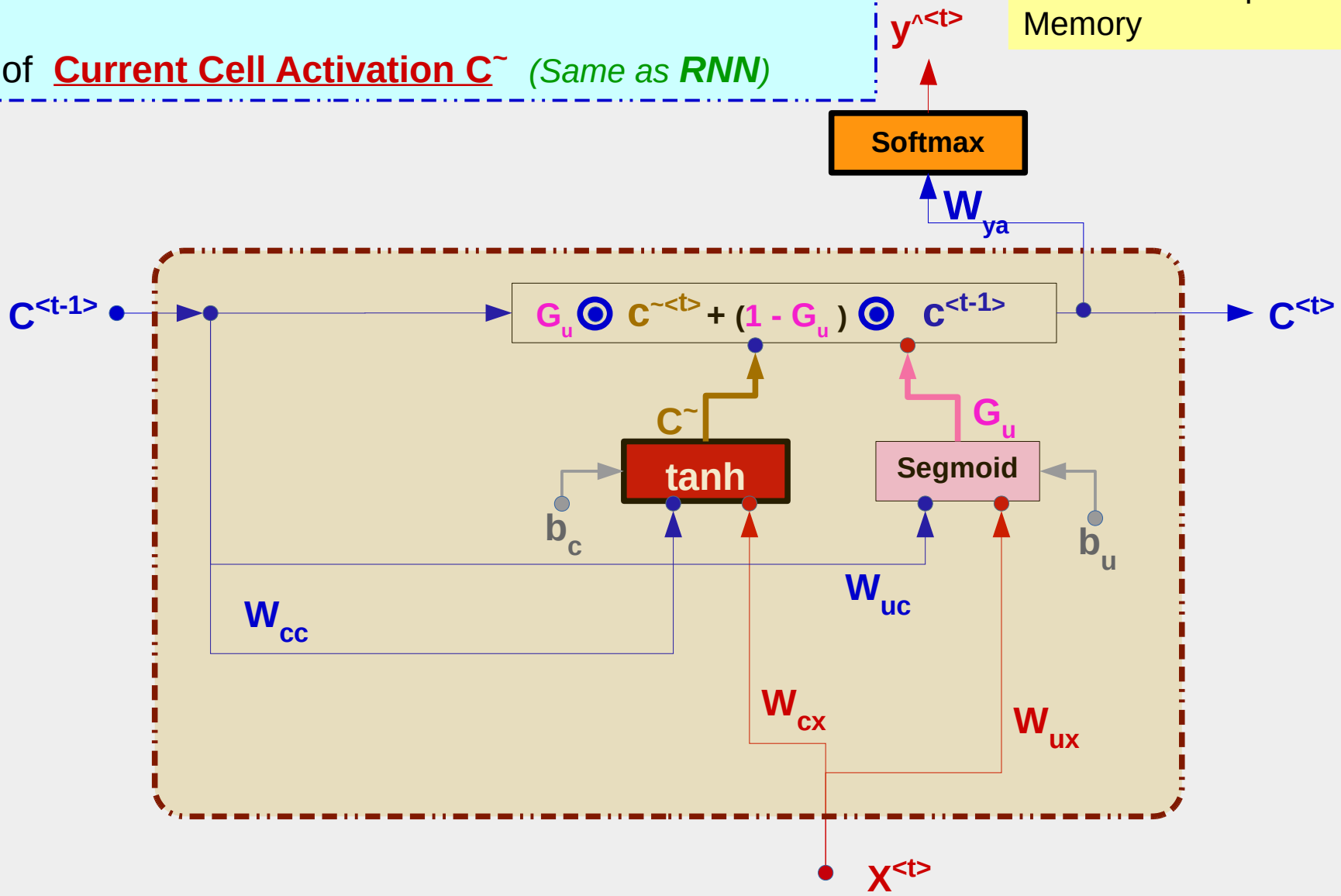
From RNN to GRU

[1] Adding Update Gate G_u

Value of CURRENT Memory Cell $C^{<t>}$=
Percentage of **Previous Memory Cell $C^{<t-1>}$** *(Not Affected by $X^{<t>}$)*
+
Percentage of **Current Cell Activation $C^{\sim}$** *(Same as **RNN**)*

$C^{\sim}$ Candidate Value of Updated Memory

C Value of Updated Memory



By Prof. Khaled M. El Sayed *Khaledms@fci-cu.edu.eg*

From RNN to GRU

[1] Adding Update Gate G_u

Value of G_u is based on:

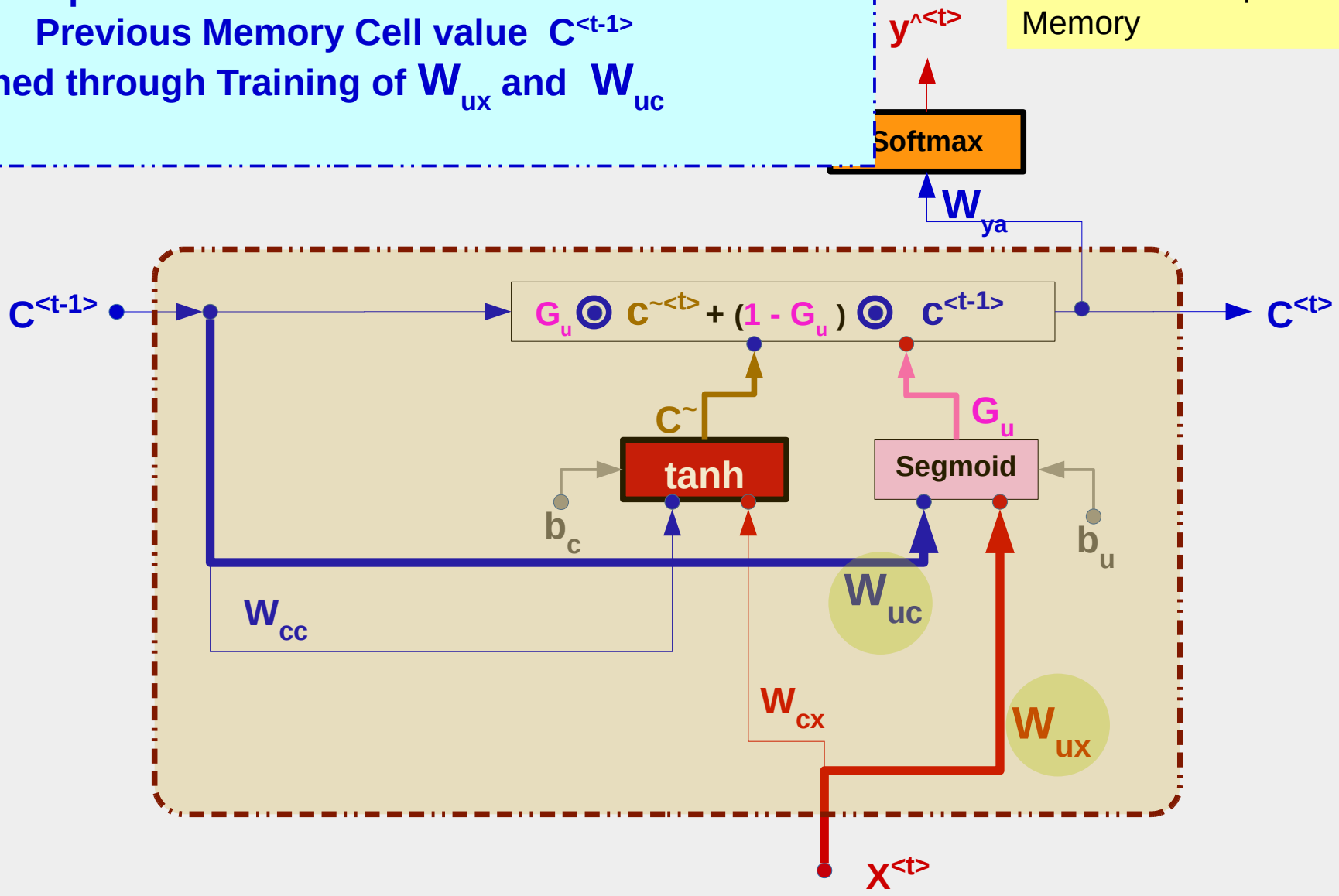
Input $X^{<t>}$ and

Previous Memory Cell value $C^{<t-1>}$

G_u is obtained through Training of W_{ux} and W_{uc}

$\tilde{C}$ Candidate Value of Updated Memory

C Value of Updated Memory



From RNN to GRU

[1] Adding Update Gate G_u

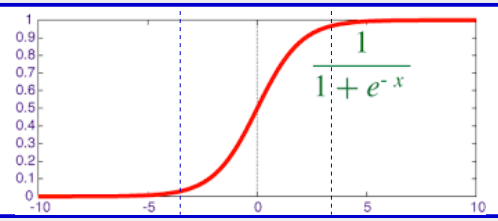
If G_u = 0 , **Keep** Memory Value "$C^{<t>}$" **Same as** Previous Value "$C^{<t-1>}$"

If G_u = 1 , **Forget** Previous Memory Value "$C^{<t-1>}$"

$\tilde{C}$ Candidate Value of Updated Memory

C Value of Updated Memory

Sigmoid



From RNN to GRU

$G_u = \text{Sigmoid}\ (W_{uc} c^{<t-1>} + W_{ux} x^{<t>} + b_u)$

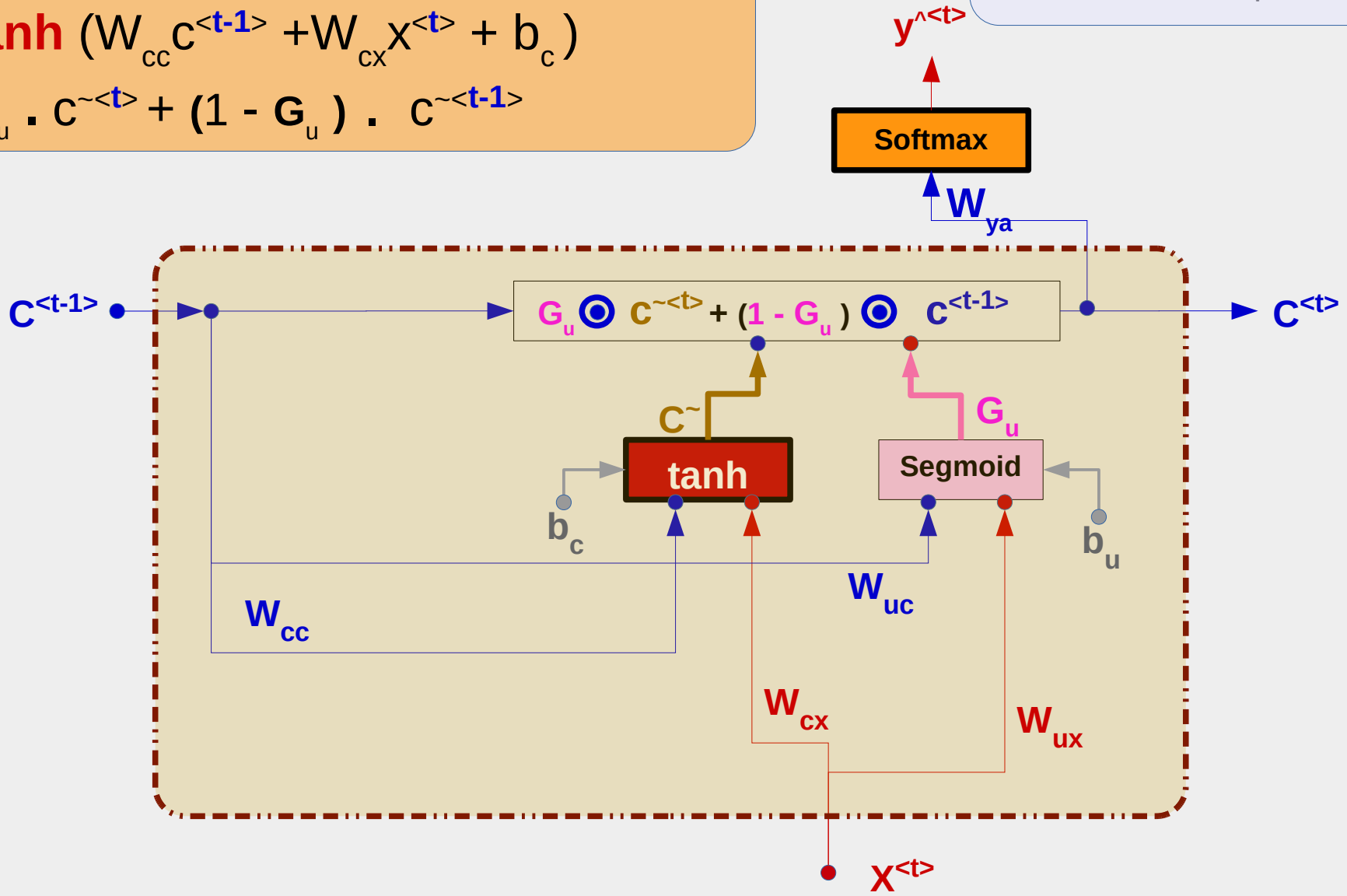
$c^{\sim<t>} = \text{tanh}\ (W_{cc} c^{<t-1>} + W_{cx} x^{<t>} + b_c)$

$c^{<t>} = G_u \,.\, c^{\sim<t>} + (1 - G_u) \,.\, c^{\sim<t-1>}$

$C^{\sim}$ is the Candidate Update

G_u is the Update Gate

C is the Actual Update



By Prof. Khaled M. El Sayed

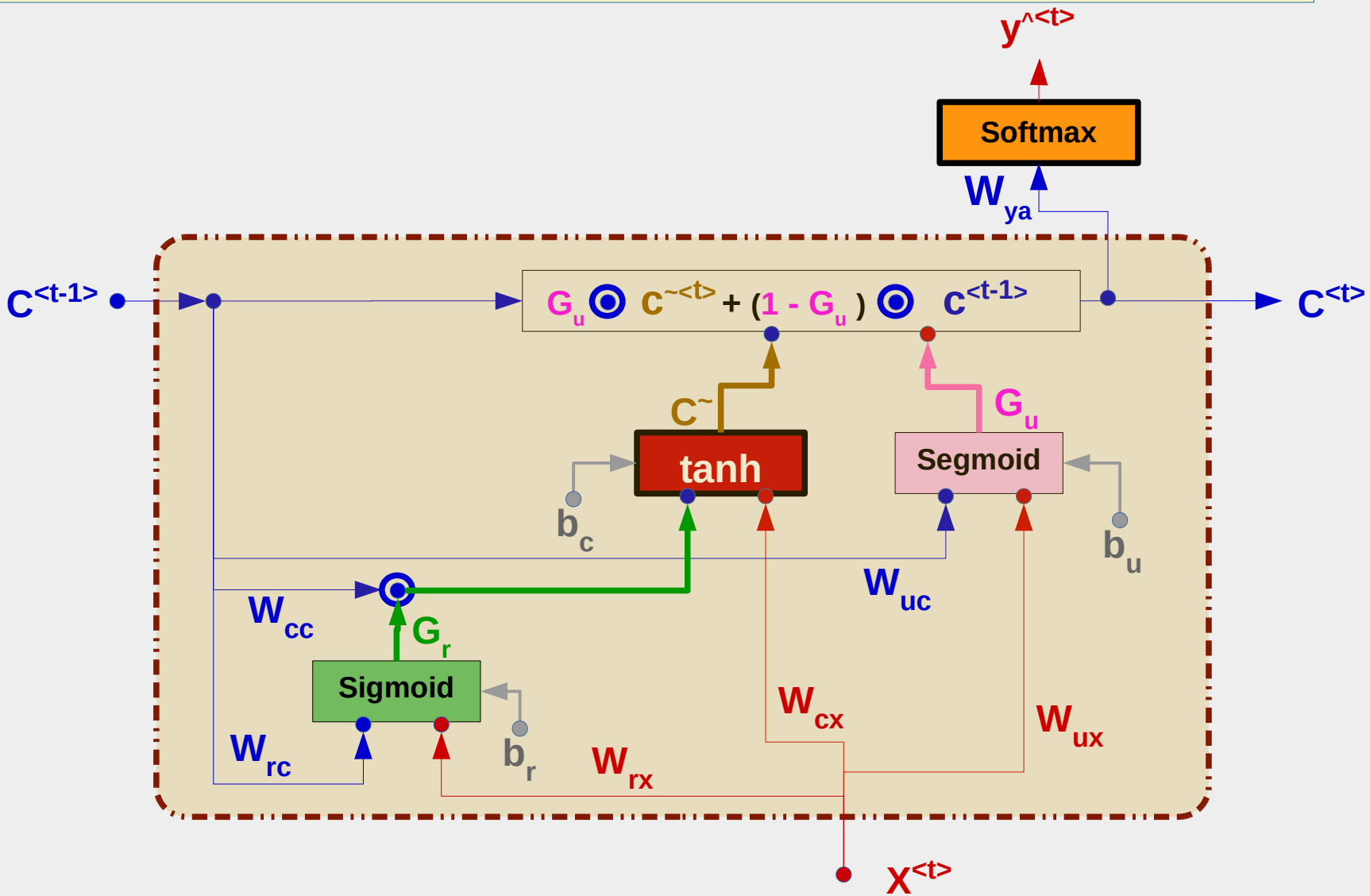
Khaledms@fci-cu.edu.eg

From RNN to GRU

[2] Adding Relevance Gate G_r

If $G_r = 1$, $C^{<t-1>}$ is **Relevant** to update Candidate Memory cell value "$C^{\sim}$"

If $G_r = 0$, $C^{<t-1>}$ is **IrRelevant** to update Candidate Memory cell value "$C^{\sim}$"



From RNN to GRU

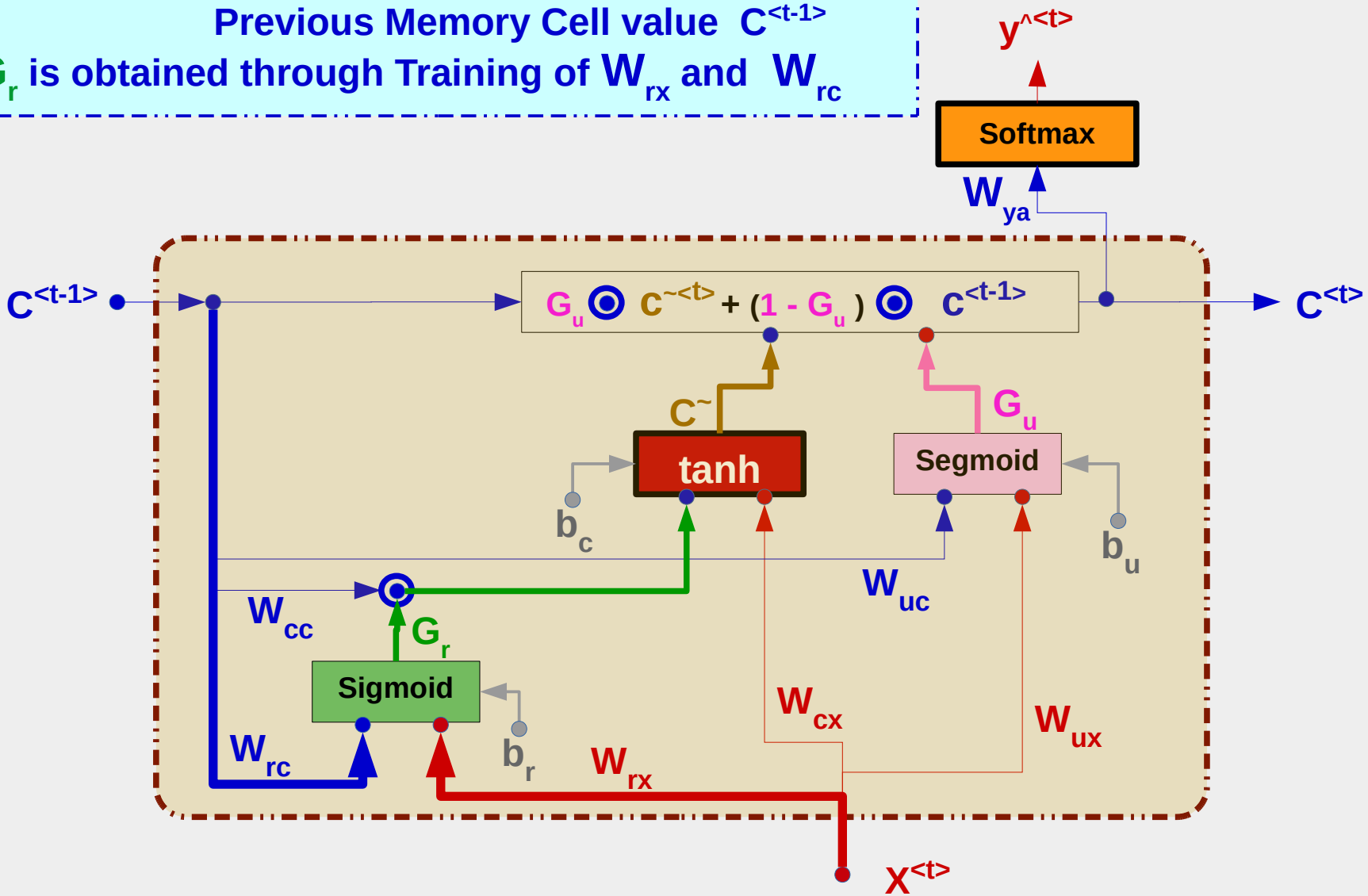
[2] Adding Relevance Gate G_r

Value of G_r is based on:

Input $X^{<t>}$ and

Previous Memory Cell value $C^{<t-1>}$

G_r is obtained through Training of W_{rx} and W_{rc}



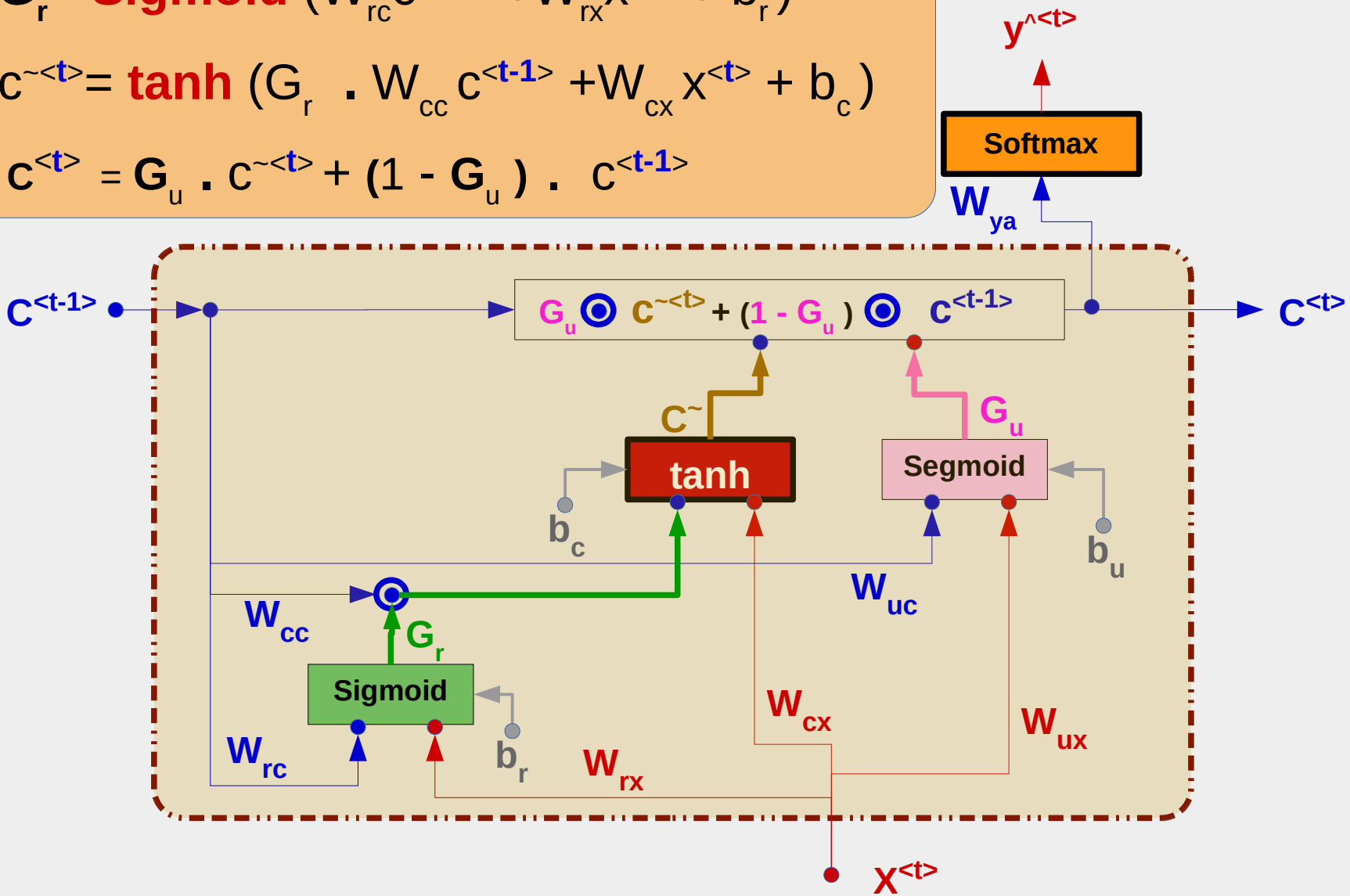
From RNN to GRU

$$G_u = \text{Sigmoid}\,(W_{uc} c^{<t-1>} + W_{ux} x^{<t>} + b_u)$$

$$G_r = \text{Sigmoid}\,(W_{rc} c^{<t-1>} + W_{rx} x^{<t>} + b_r)$$

$$c^{\sim<t>} = \tanh\,(G_r \,.\, W_{cc} c^{<t-1>} + W_{cx} x^{<t>} + b_c)$$

$$c^{<t>} = G_u \,.\, c^{\sim<t>} + (1 - G_u) \,.\, c^{<t-1>}$$



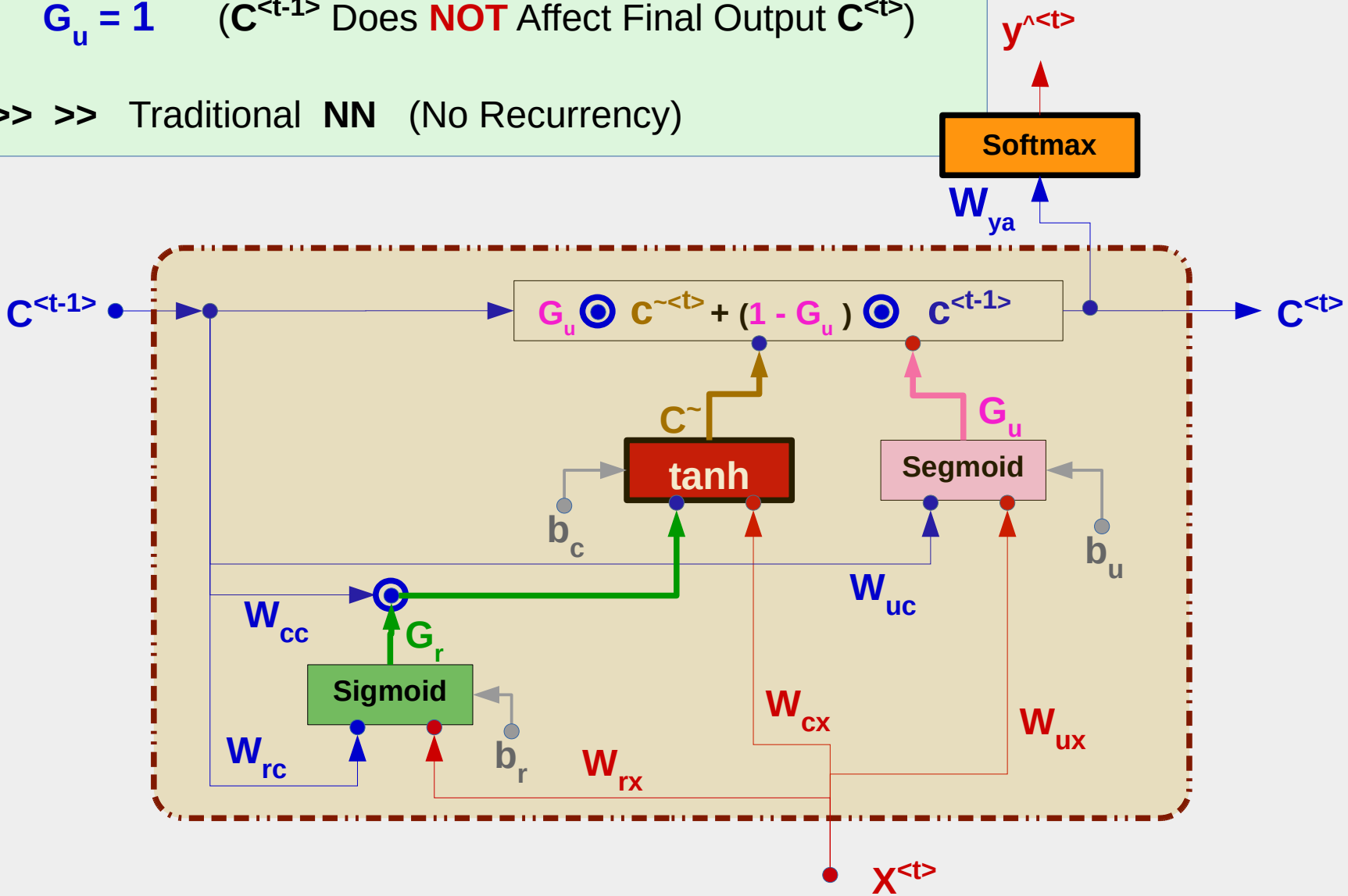
From RNN to GRU

IF

$G_r = 0$ ($C^{<t-1>}$ Does **NOT** Affect tanh Output $C^{\sim}$)

$G_u = 1$ ($C^{<t-1>}$ Does **NOT** Affect Final Output $C^{<t>}$)

>> >> Traditional **NN** (No Recurrency)



By Prof. Khaled M. El Sayed *Khaledms@fci-cu.edu.eg*

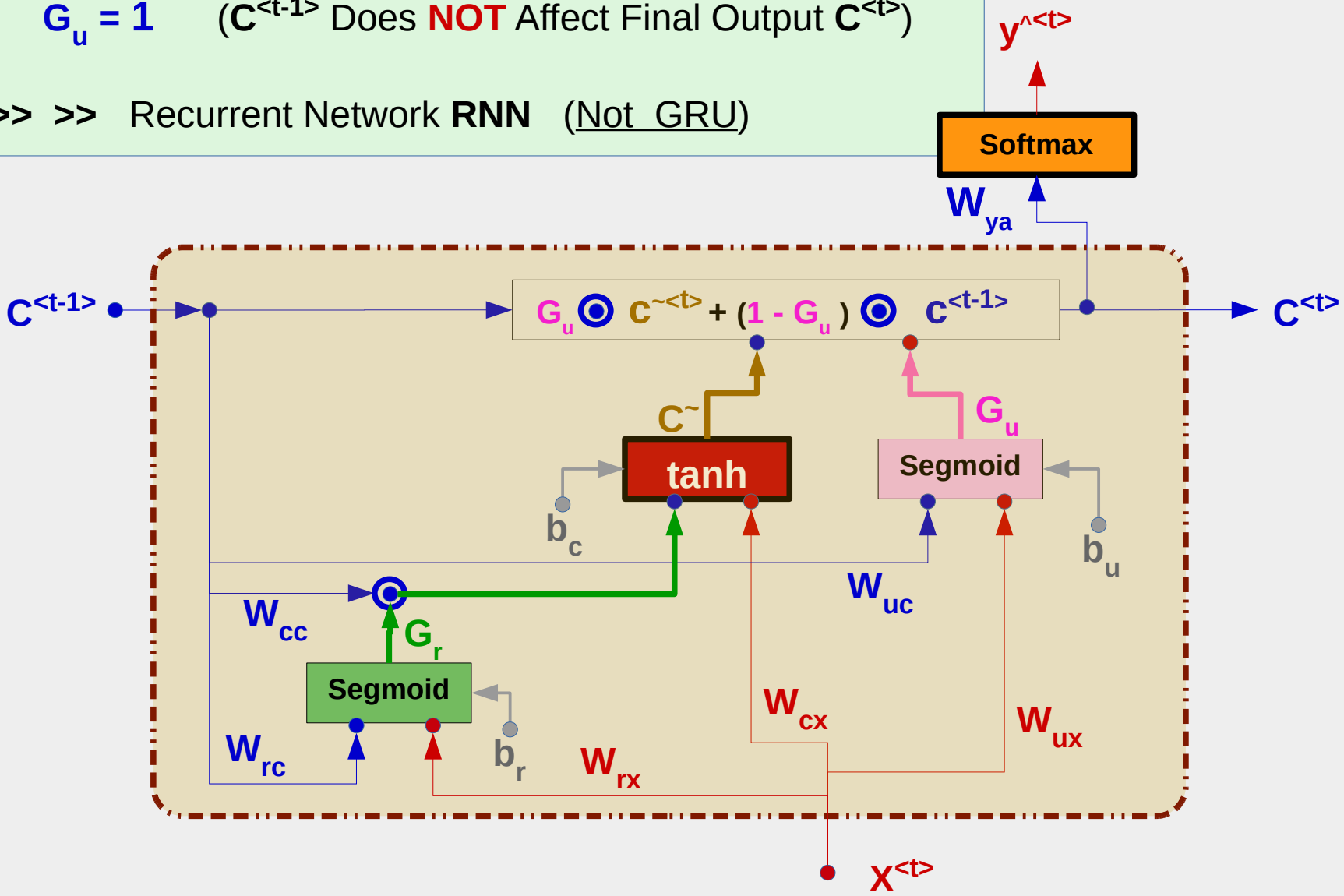
From RNN to GRU

IF

$G_r = 1$ ($C^{<t-1>}$ **Affects** tanh Output $C^{\sim}$)

$G_u = 1$ ($C^{<t-1>}$ Does **NOT** Affect Final Output $C^{<t>}$)

>> >> Recurrent Network **RNN** (Not GRU)



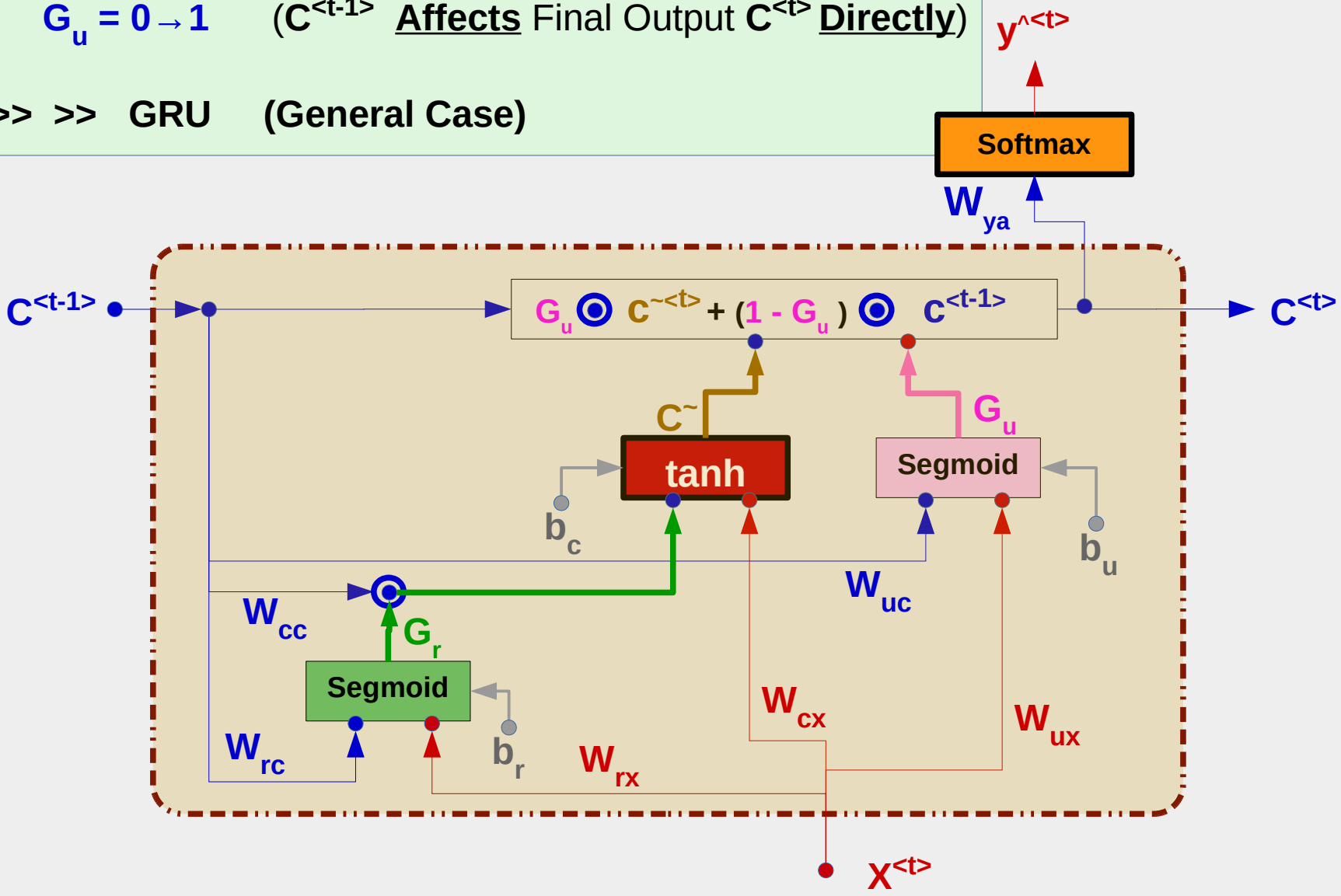
From RNN to GRU

IF

$G_r = 0 \rightarrow 1$ ($C^{<t-1>}$ **Affects** tanh Output $C^{\sim}$)

$G_u = 0 \rightarrow 1$ ($C^{<t-1>}$ **Affects** Final Output $C^{<t>}$ **Directly**)

>> >> GRU (General Case)



[3] Long Short Term Memory (LSTM)

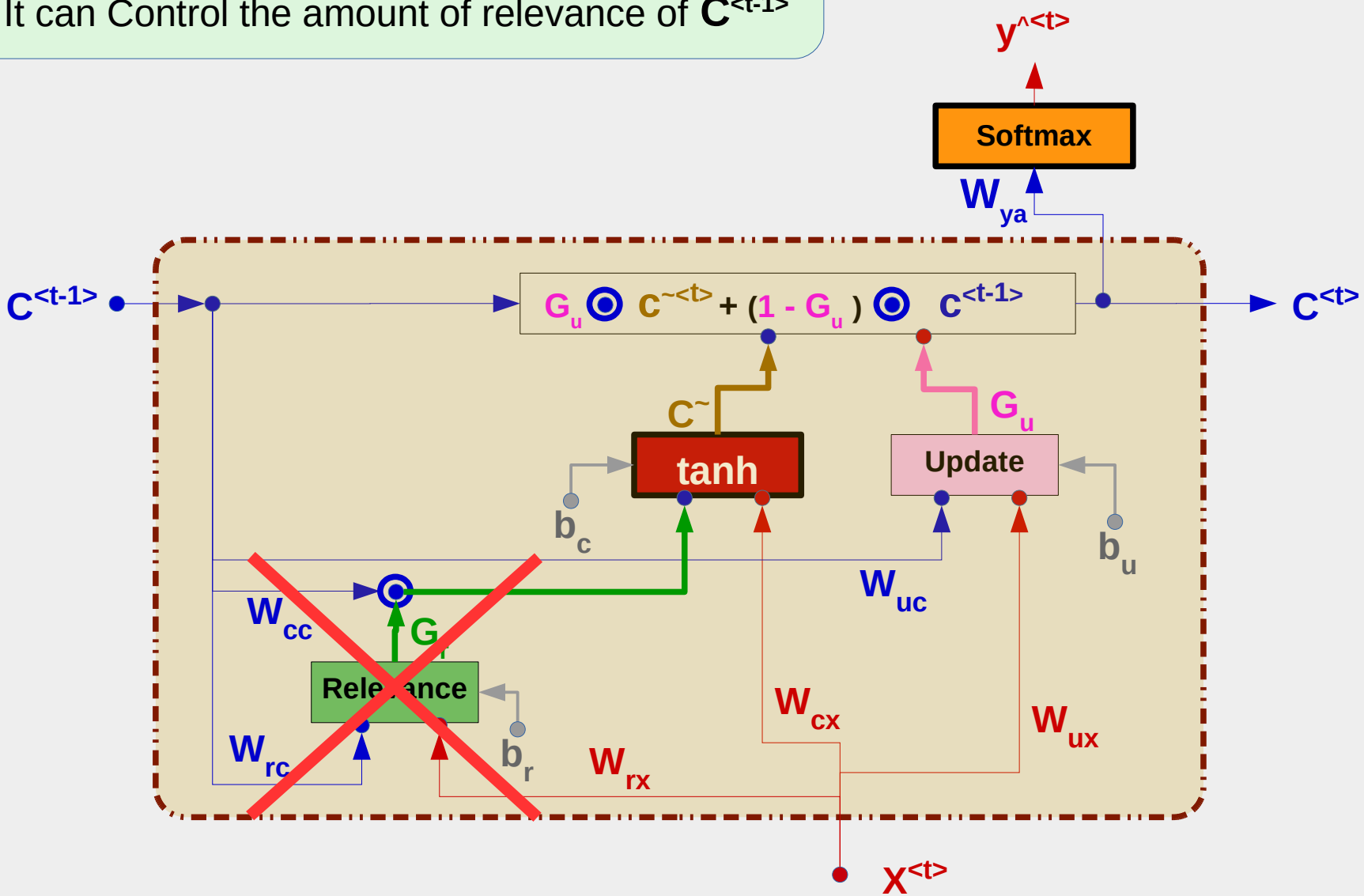
By Prof. Khaled M. El Sayed *Khaledms@fci-cu.edu.eg*

From GRU to LSTM

[1] Removing **Relevance** Gate G_r

W_{uc} can Do the same functionality

It can Control the amount of relevance of $C^{<t-1>}$



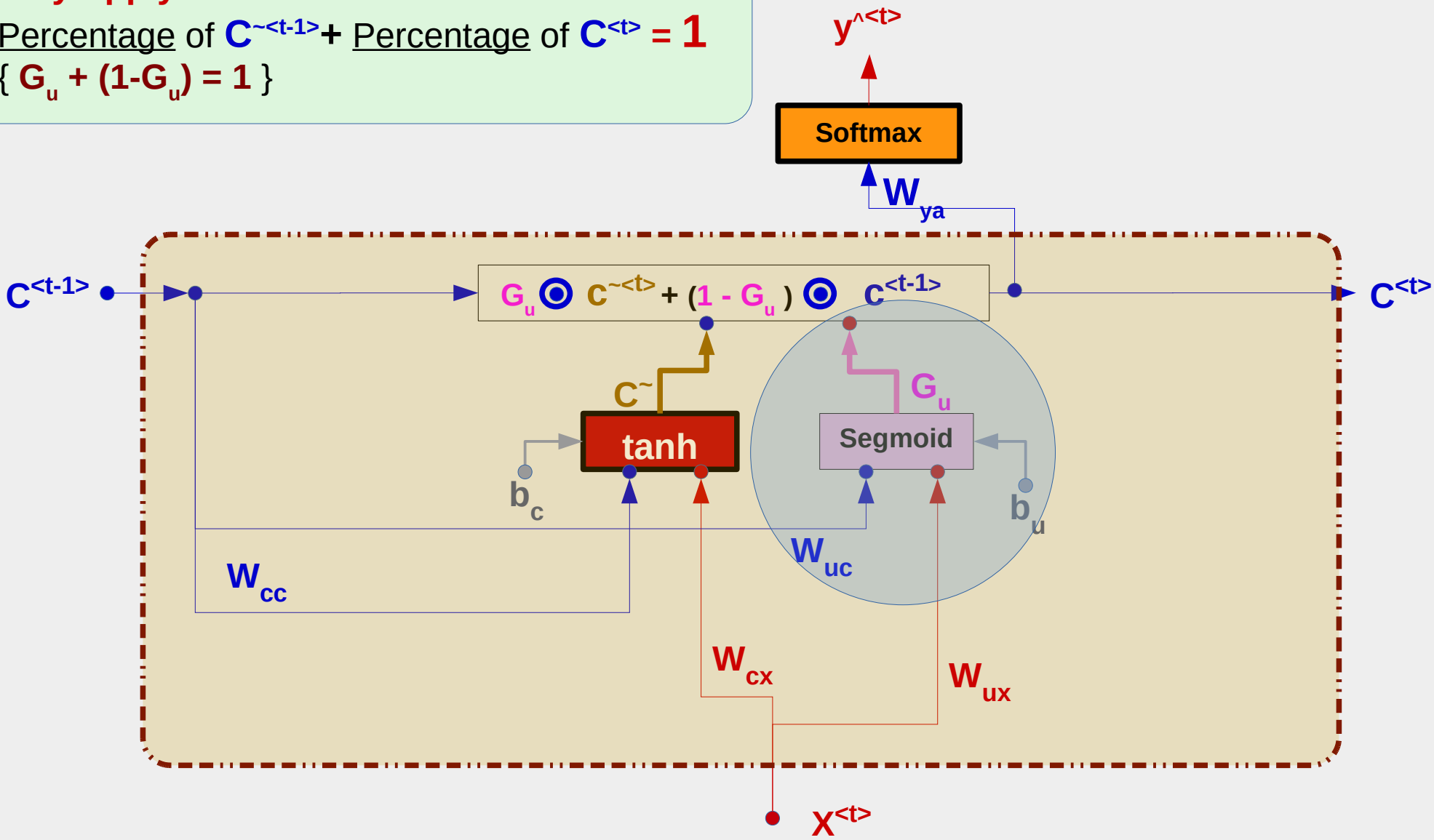
From GRU to LSTM

[2] ??????

Why Apply Constrain:

Percentage of $C^{\sim<t-1>}$ + Percentage of $C^{<t>} = 1$

$\{ G_u + (1-G_u) = 1 \}$



From GRU to LSTM

[2] ??????

Why Apply Constrain:

Percentage of $C^{\sim<t-1>}$ + Percentage of $C^{<t>} = 1$

{ $G_u + (1-G_u) = 1$ }

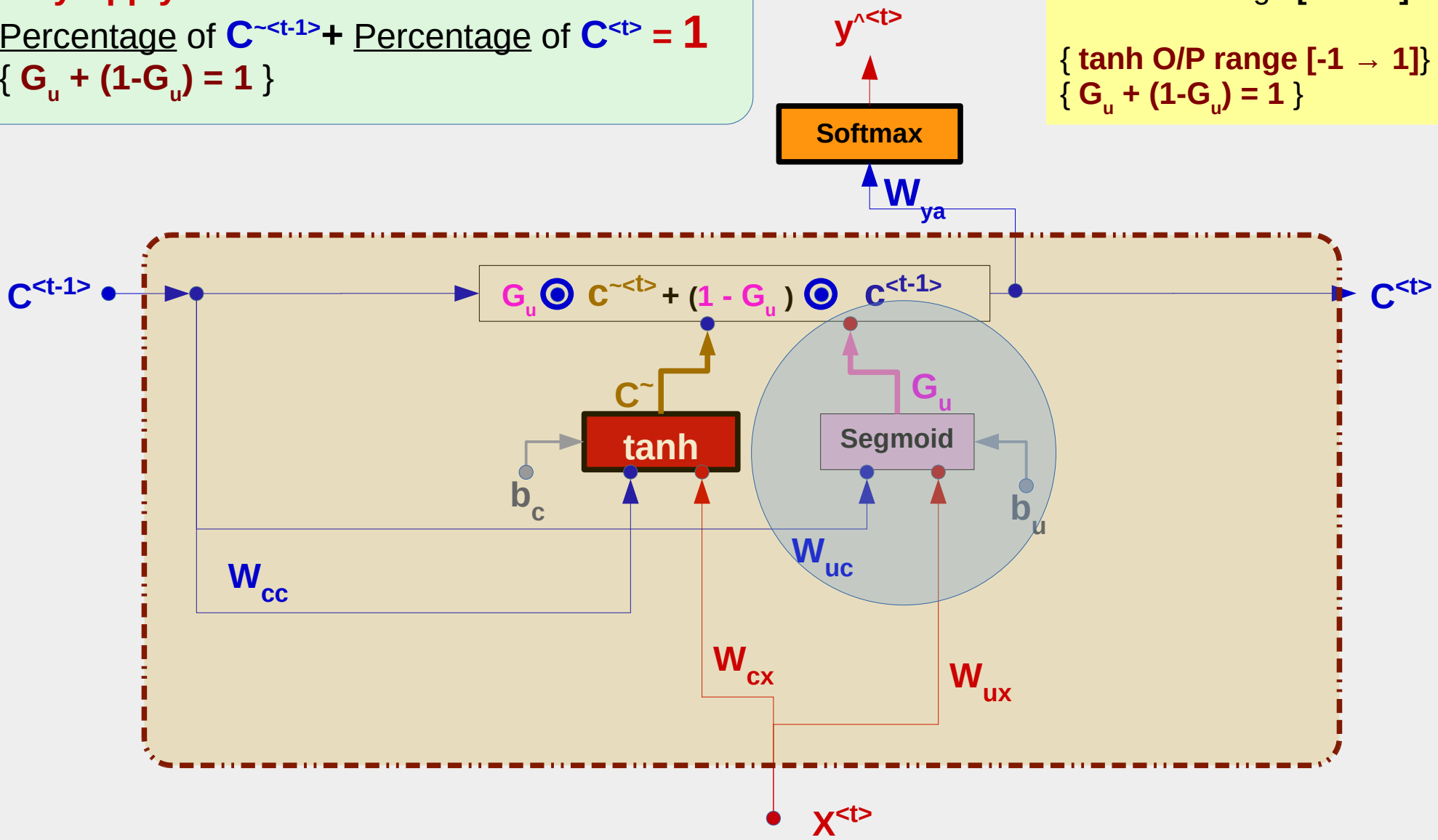
Answer

$C^{\sim}$ value range **[-1 → 1]**

C value range **[-1 → 1]**

{ **tanh O/P range [-1 → 1]**}

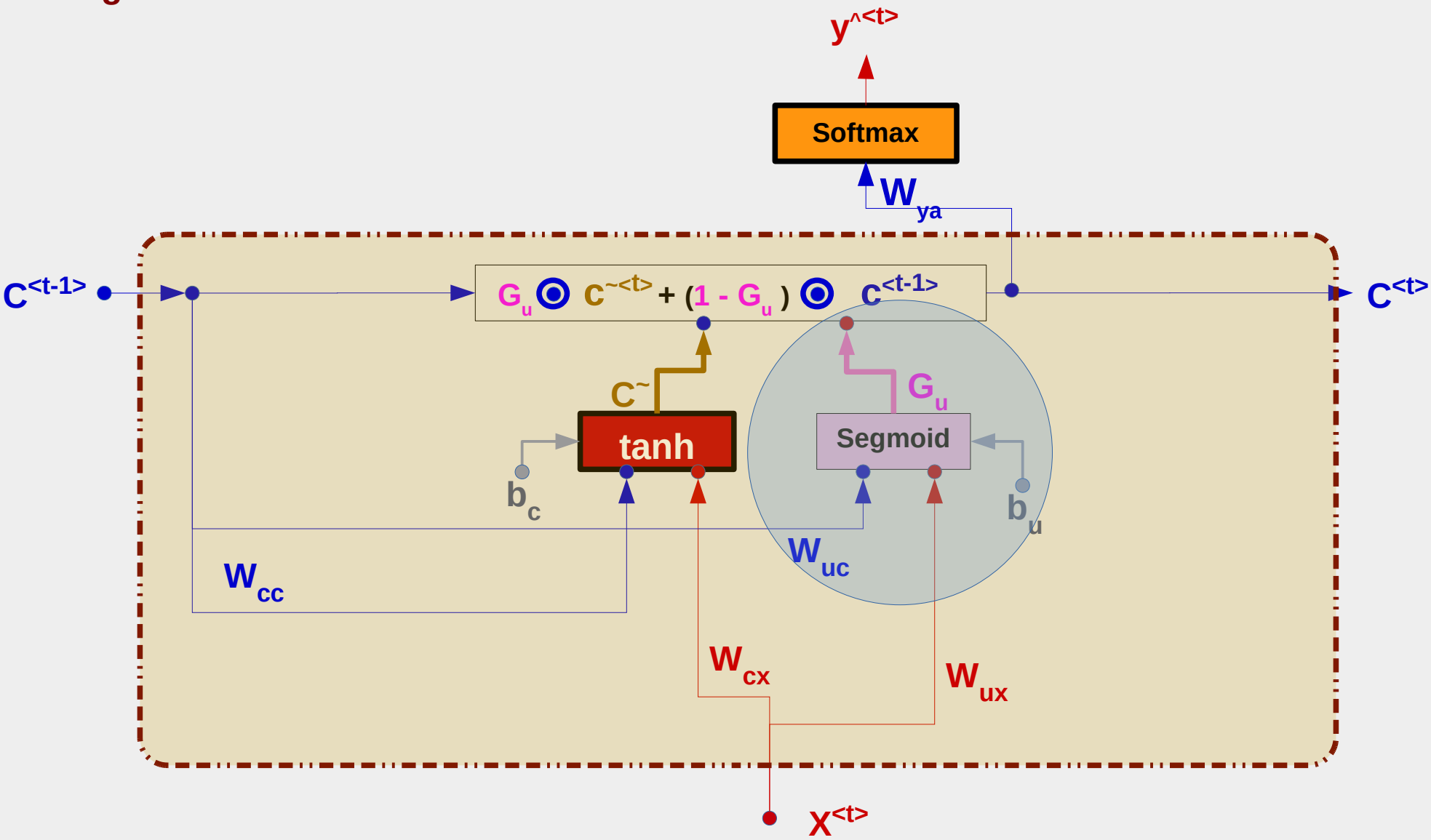
{ $G_u + (1-G_u) = 1$ }



By Prof. Khaled M. El Sayed *Khaledms@fci-cu.edu.eg*

From GRU to LSTM

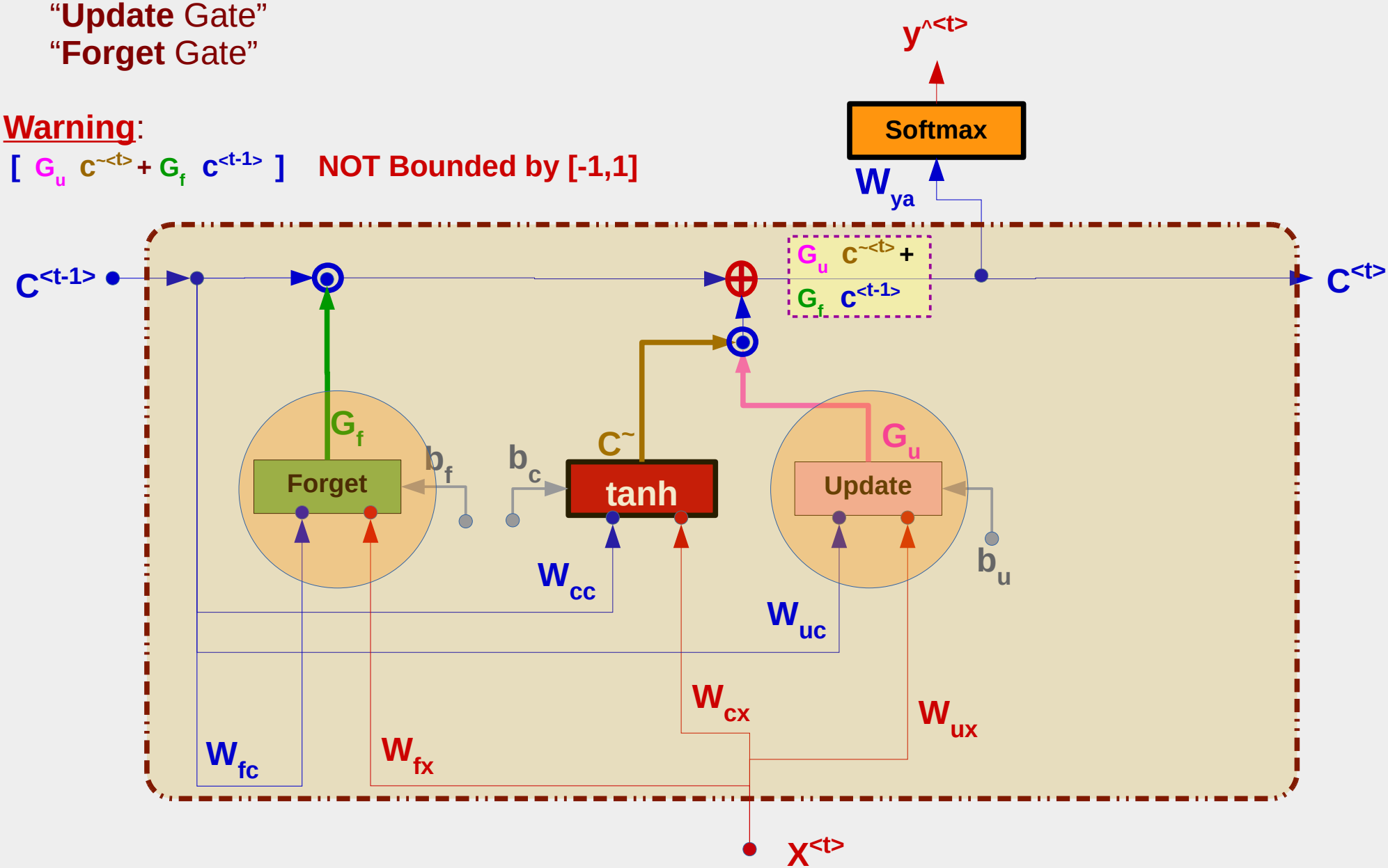
[2] Split "**Update** Gate" into two gates:
"**Update** Gate"
"**Forget** Gate"



From GRU to LSTM

[2] Split "**Update** Gate" into two gates:
"**Update** Gate"
"**Forget** Gate"

Warning:
$[\ G_u\ c^{\sim<t>} + G_f\ c^{<t-1>}\ ]$ **NOT Bounded by [-1,1]**

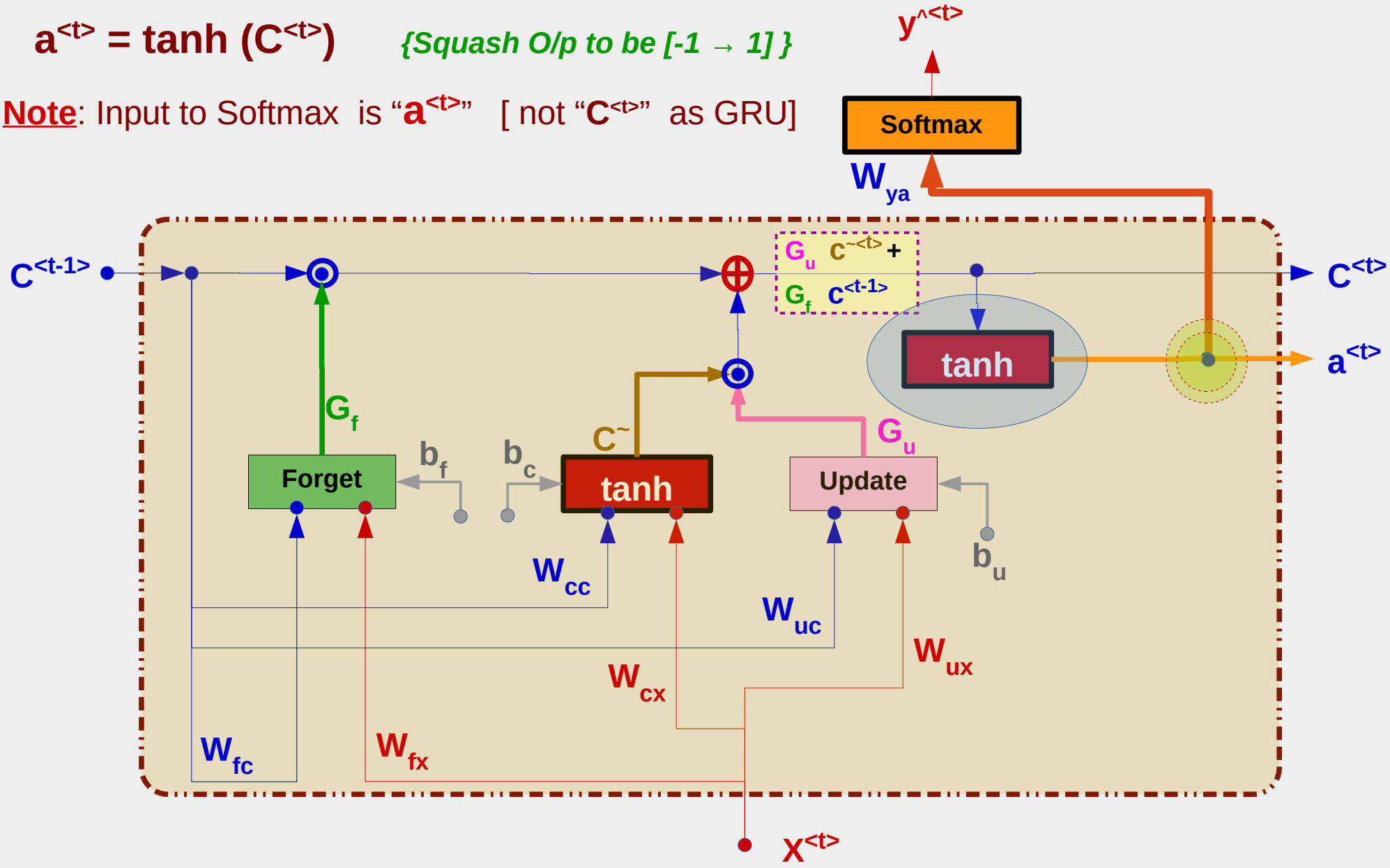


From GRU to LSTM

[3] Differentiate between Cell Memory value $C^{<t>}$ and Cell Output $a^{<t>}$

$a^{<t>} = \tanh(C^{<t>})$ *{Squash O/p to be [-1 → 1] }*

Note: Input to Softmax is "$a^{<t>}$" [not "$C^{<t>}$" as GRU]

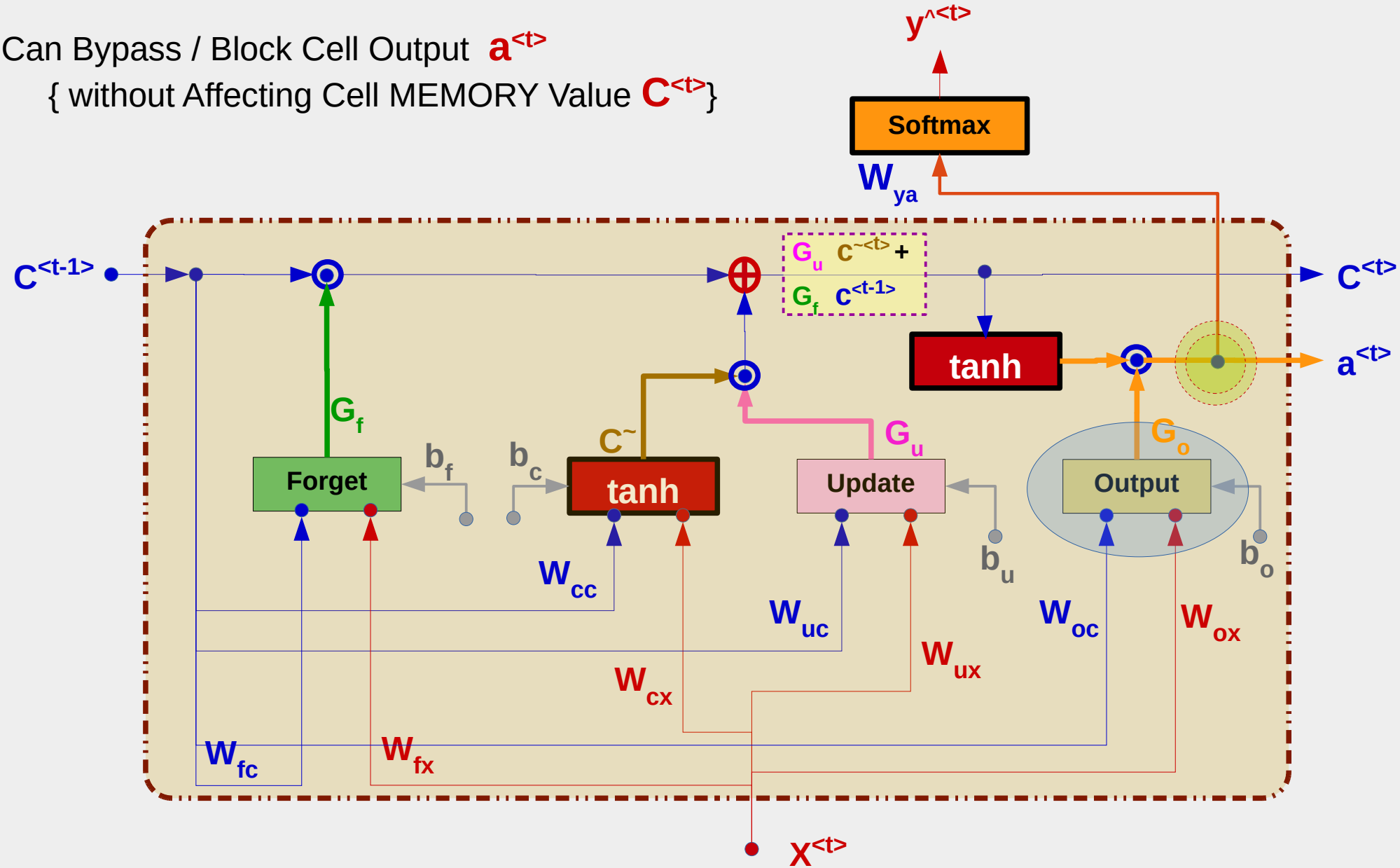


From GRU to LSTM

[4] Add "**Output** Control Gate"

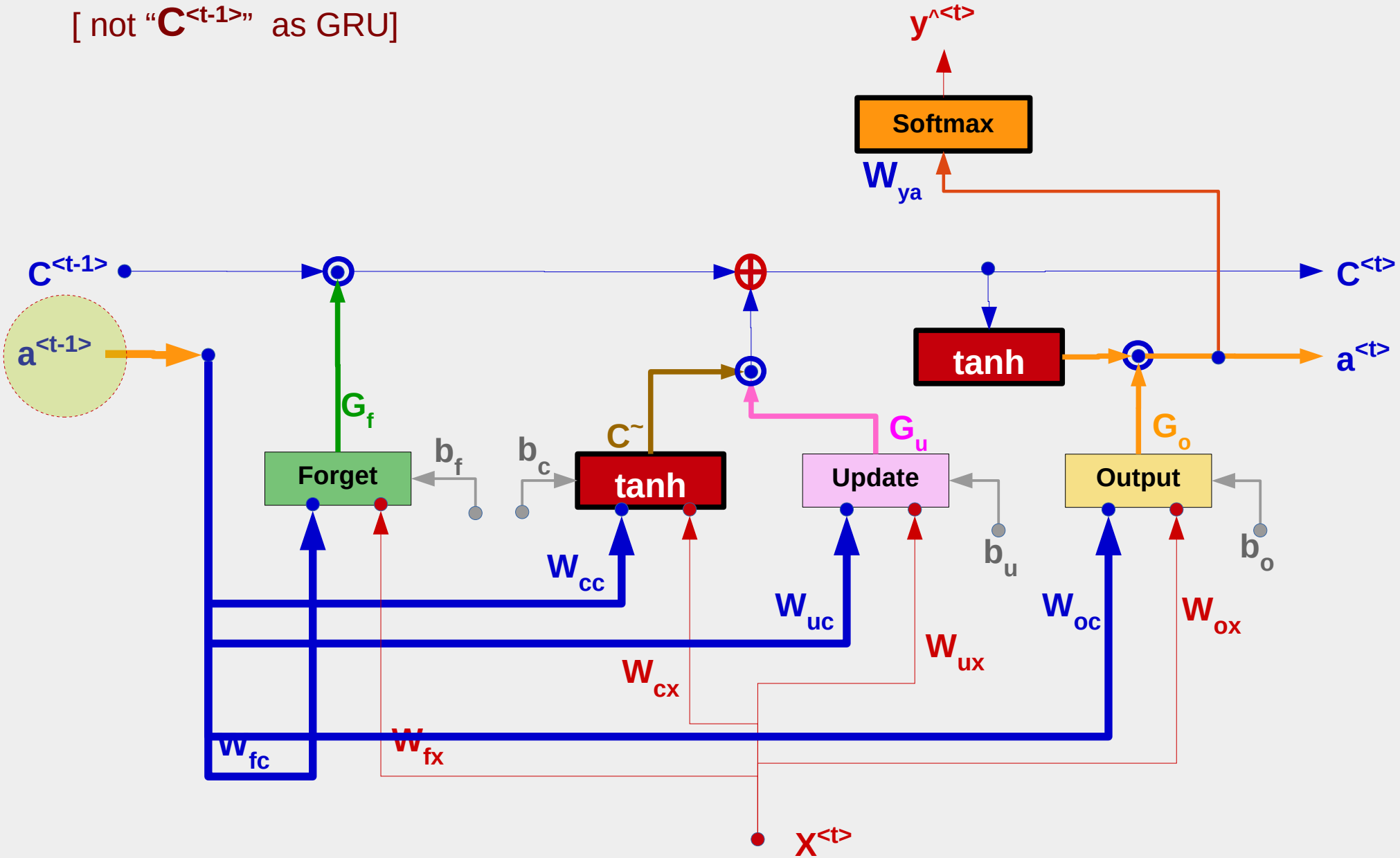
Can Bypass / Block Cell Output $a^{<t>}$

{ without Affecting Cell MEMORY Value $C^{<t>}$}



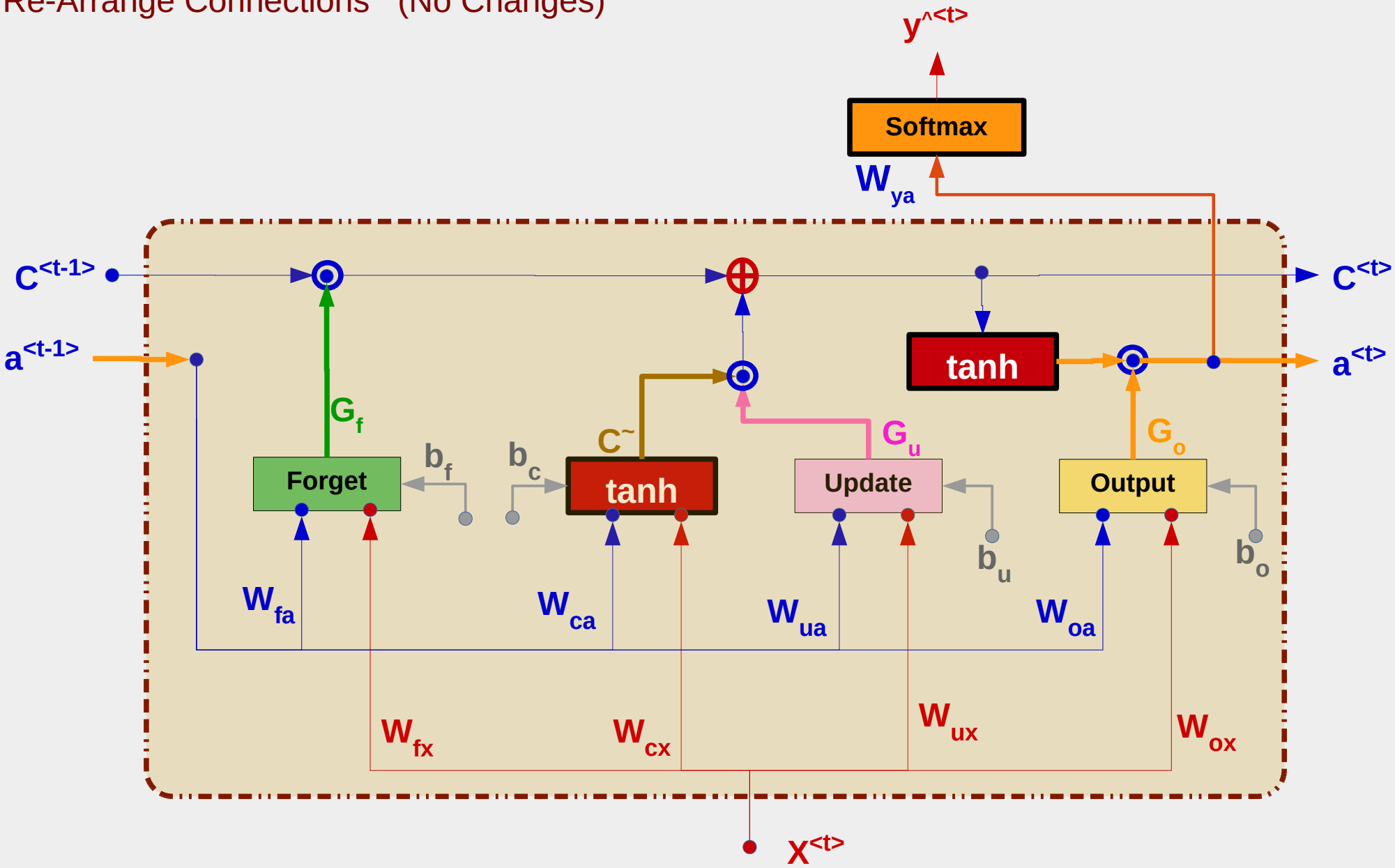
From GRU to LSTM

[5] **Input to Control Gates** is "$a^{<t-1>}$"
[not "$C^{<t-1>}$" as GRU]



LSTM Unit

Re-Arrange Connections (No Changes)



By Prof. Khaled M. El Sayed

Khaledms@fci-cu.edu.eg

LSTM Unit

Gates

$G_u = \text{Sigmoid}\ (W_{ua} a^{<t-1>} + W_{ux} x^{<t>} + b_u)$

$G_f = \text{Sigmoid}\ (W_{fa} a^{<t-1>} + W_{fx} x^{<t>} + b_f)$

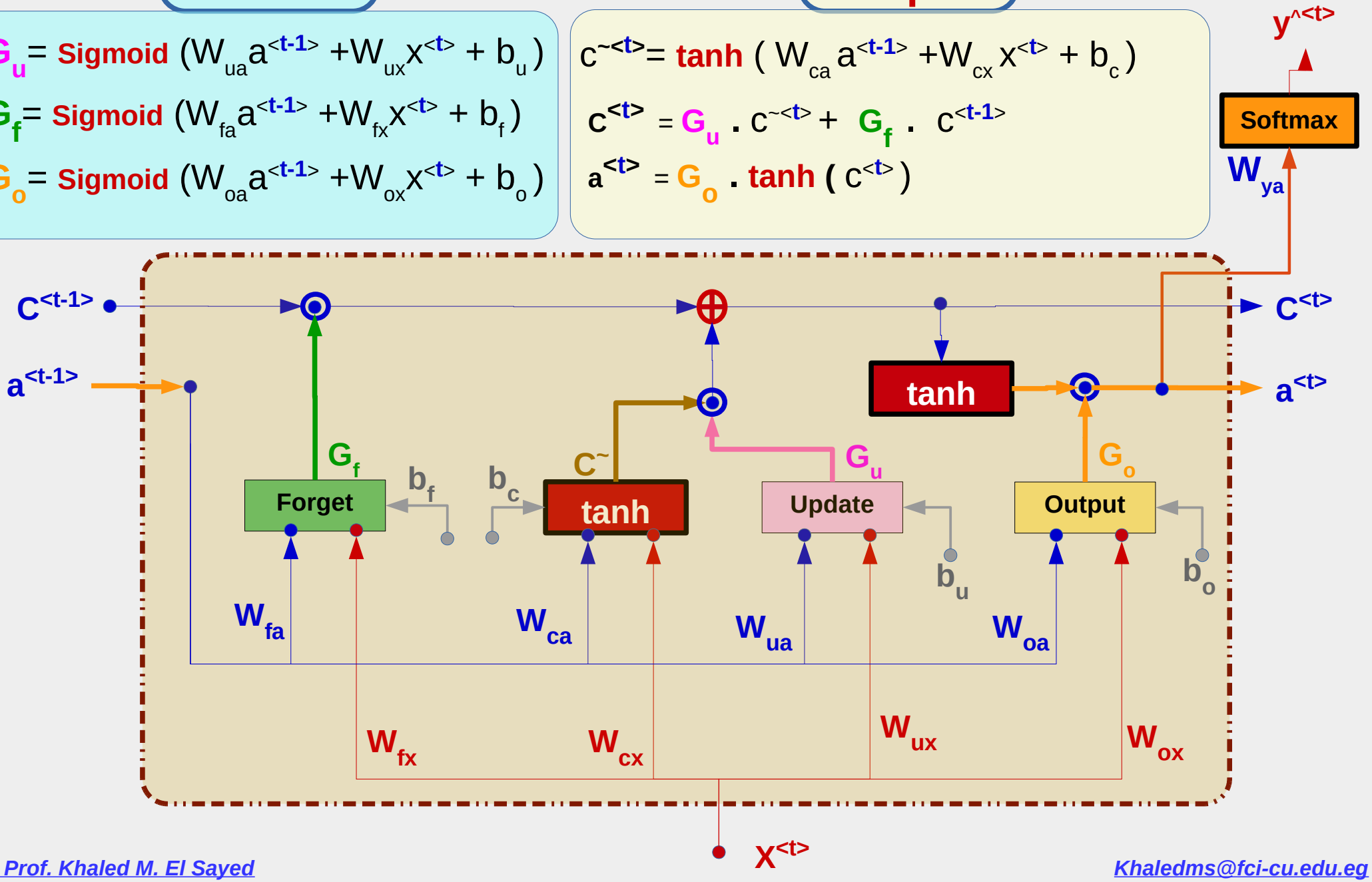
$G_o = \text{Sigmoid}\ (W_{oa} a^{<t-1>} + W_{ox} x^{<t>} + b_o)$

Outputs

$c^{\sim<t>} = \tanh\ (W_{ca} a^{<t-1>} + W_{cx} x^{<t>} + b_c)$

$c^{<t>} = G_u \cdot c^{\sim<t>} + G_f \cdot c^{<t-1>}$

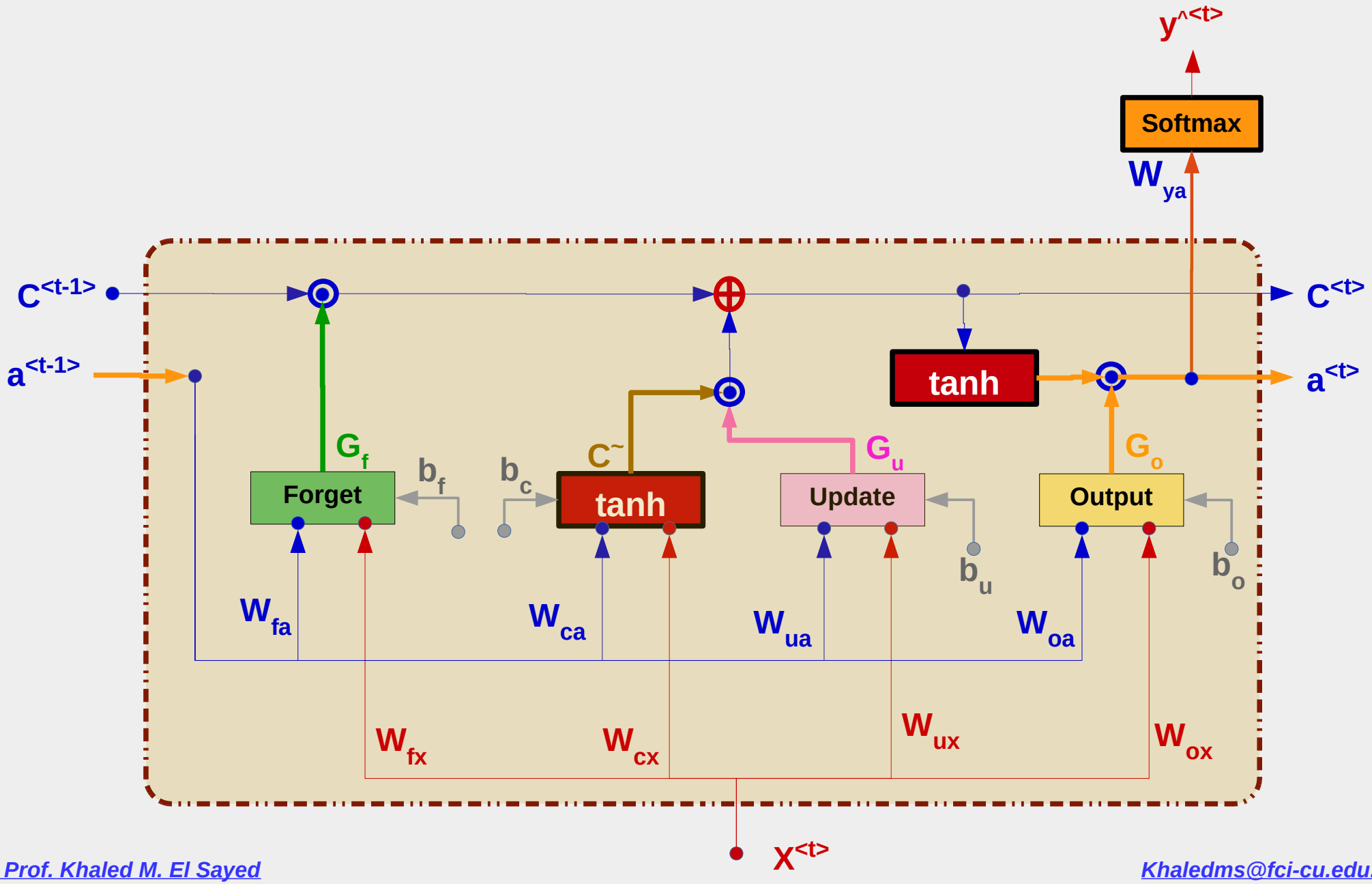
$a^{<t>} = G_o \cdot \tanh\ (c^{<t>})$



By Prof. Khaled M. El Sayed

Khaledms@fci-cu.edu.eg

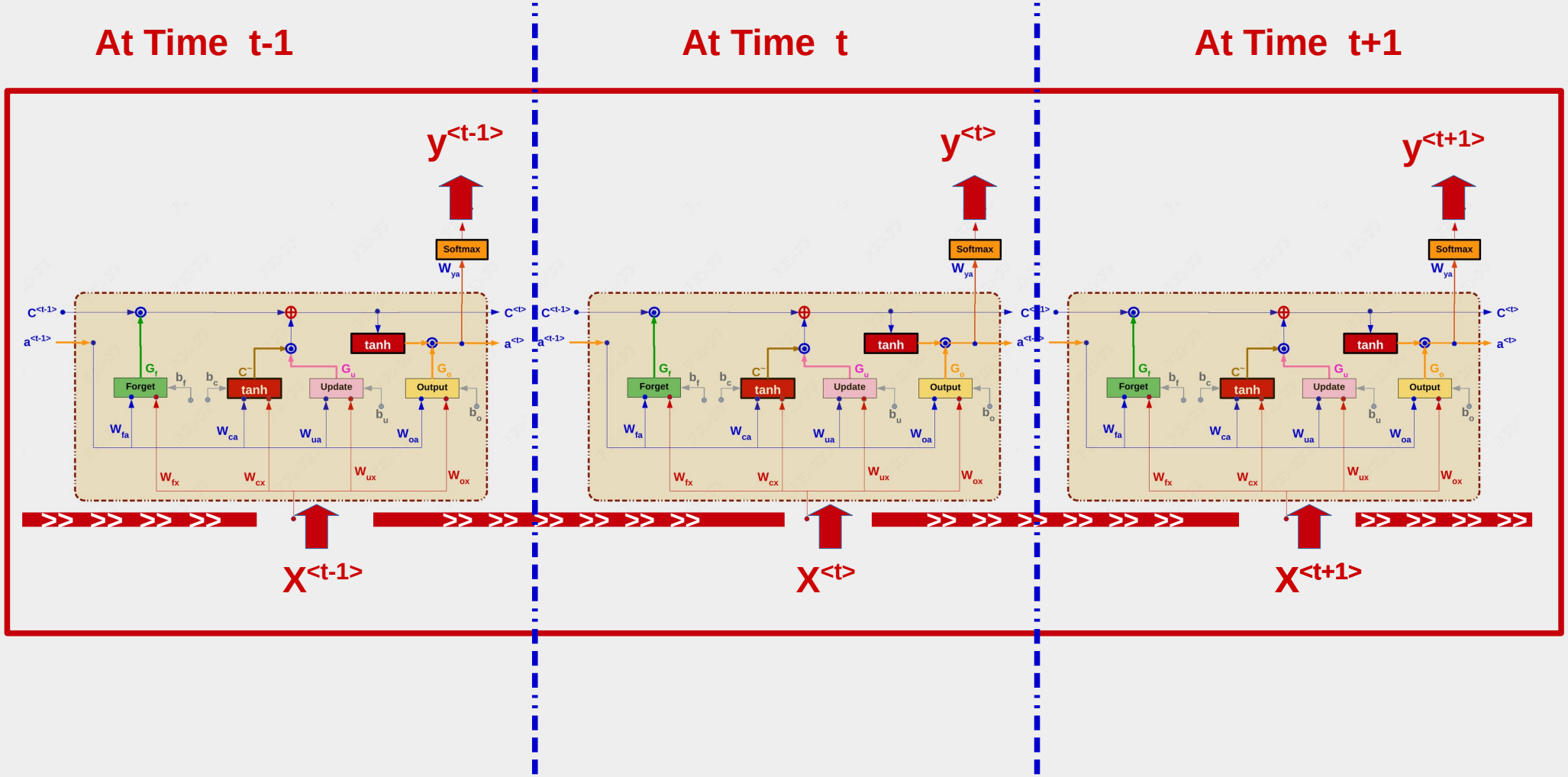
LSTM Unit



By Prof. Khaled M. El Sayed

Khaledms@fci-cu.edu.eg

Input Sequence to LSTM



By Prof. Khaled M. El Sayed
Khaledms@fci-cu.edu.eg