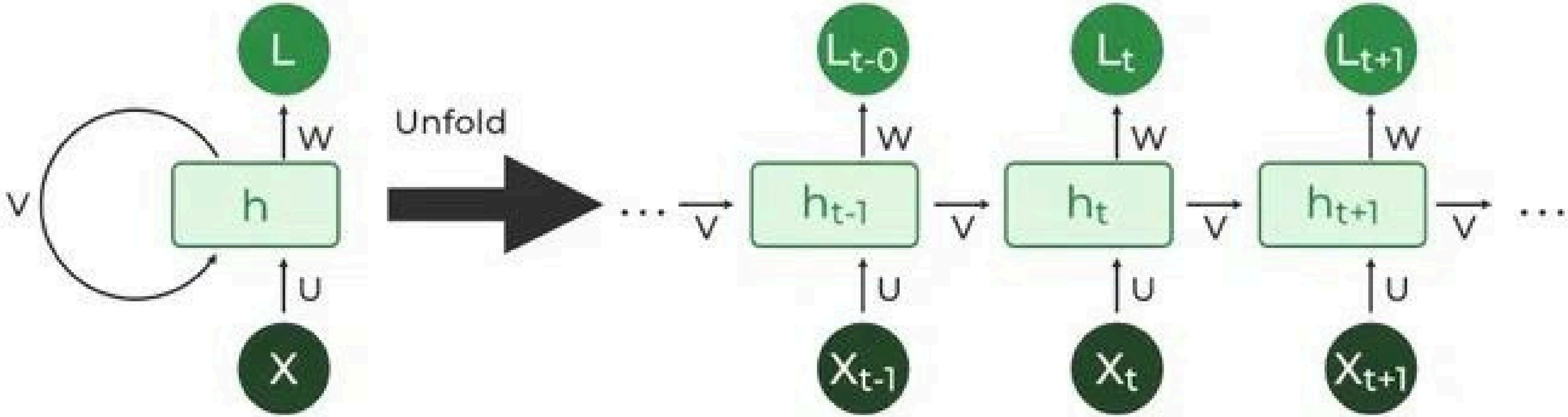
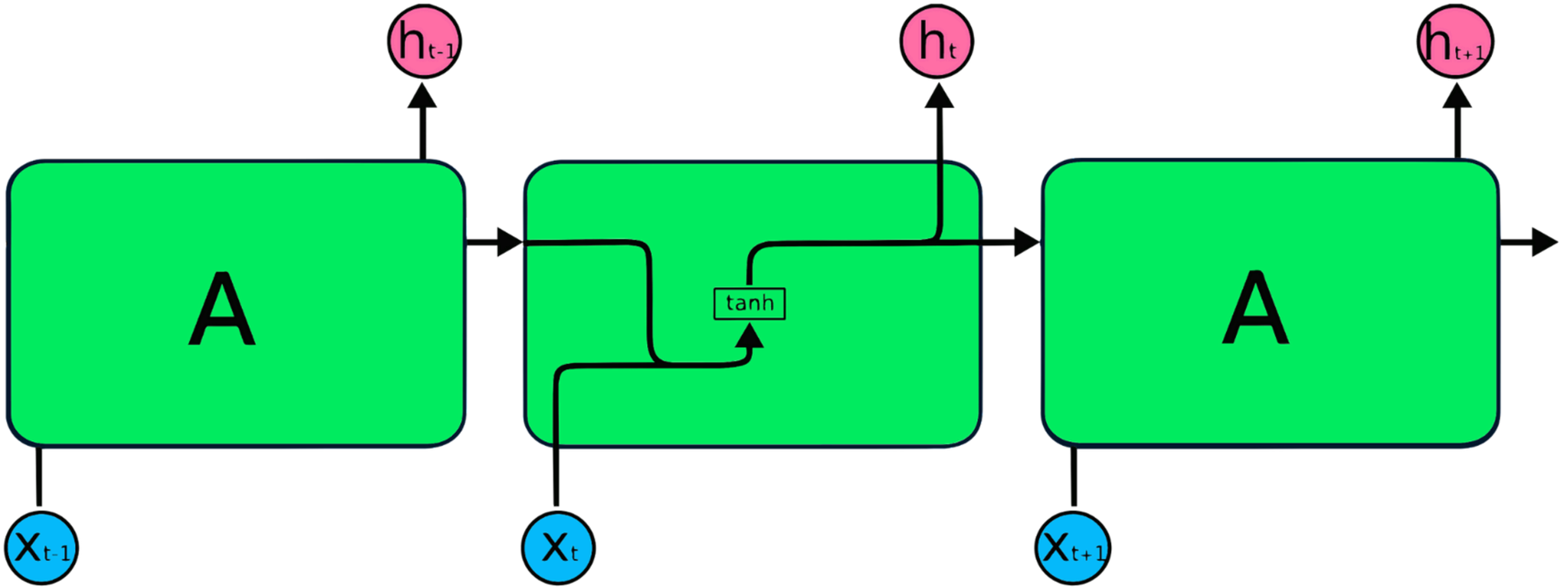
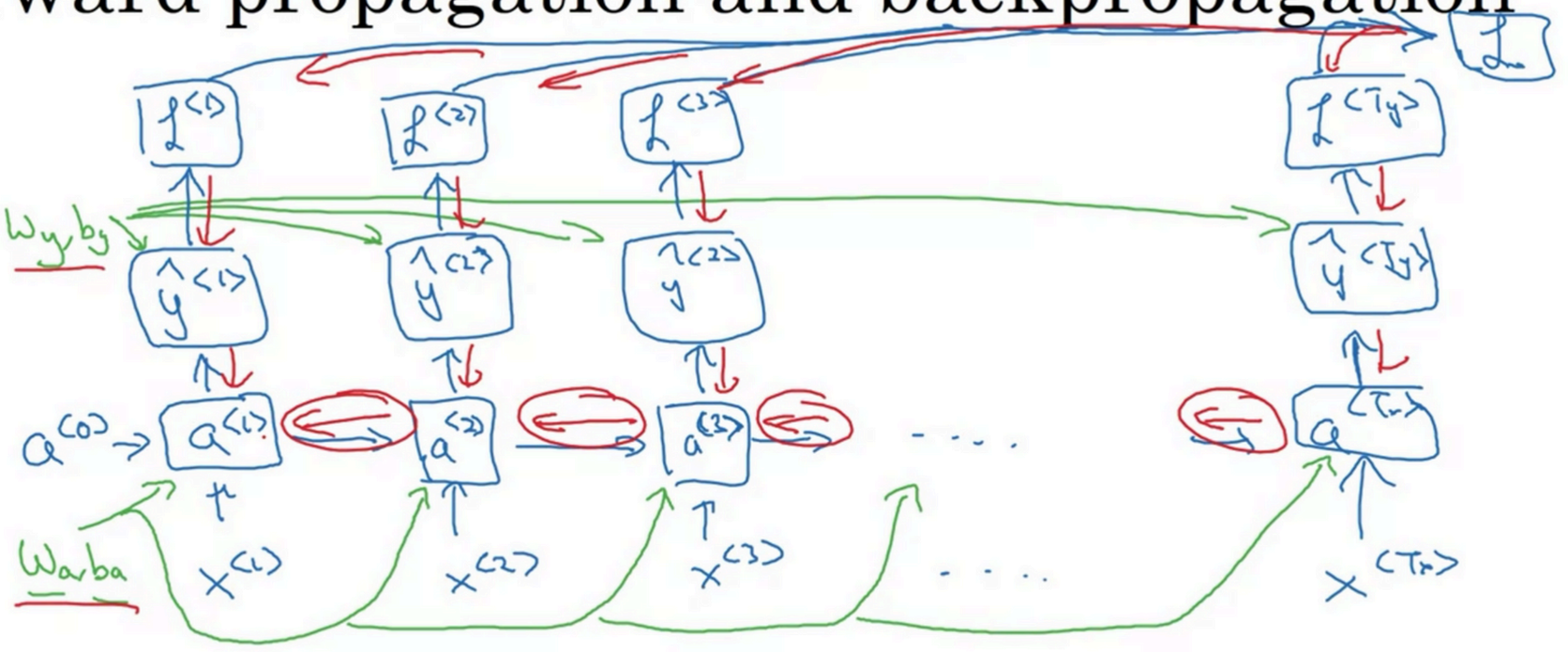
# LSTM

SIMÓN MARTÍNEZ


L
w
h
v
u
X
Unfold
Lt-0
Lt
Lt+1
ht-1
ht
ht+1
Xt-1
Xt
Xt+1


$h_{t-1}$
$h_t$
$h_{t+1}$
A
tanh
A
$X_{t-1}$
$X_t$
$X_{t+1}$

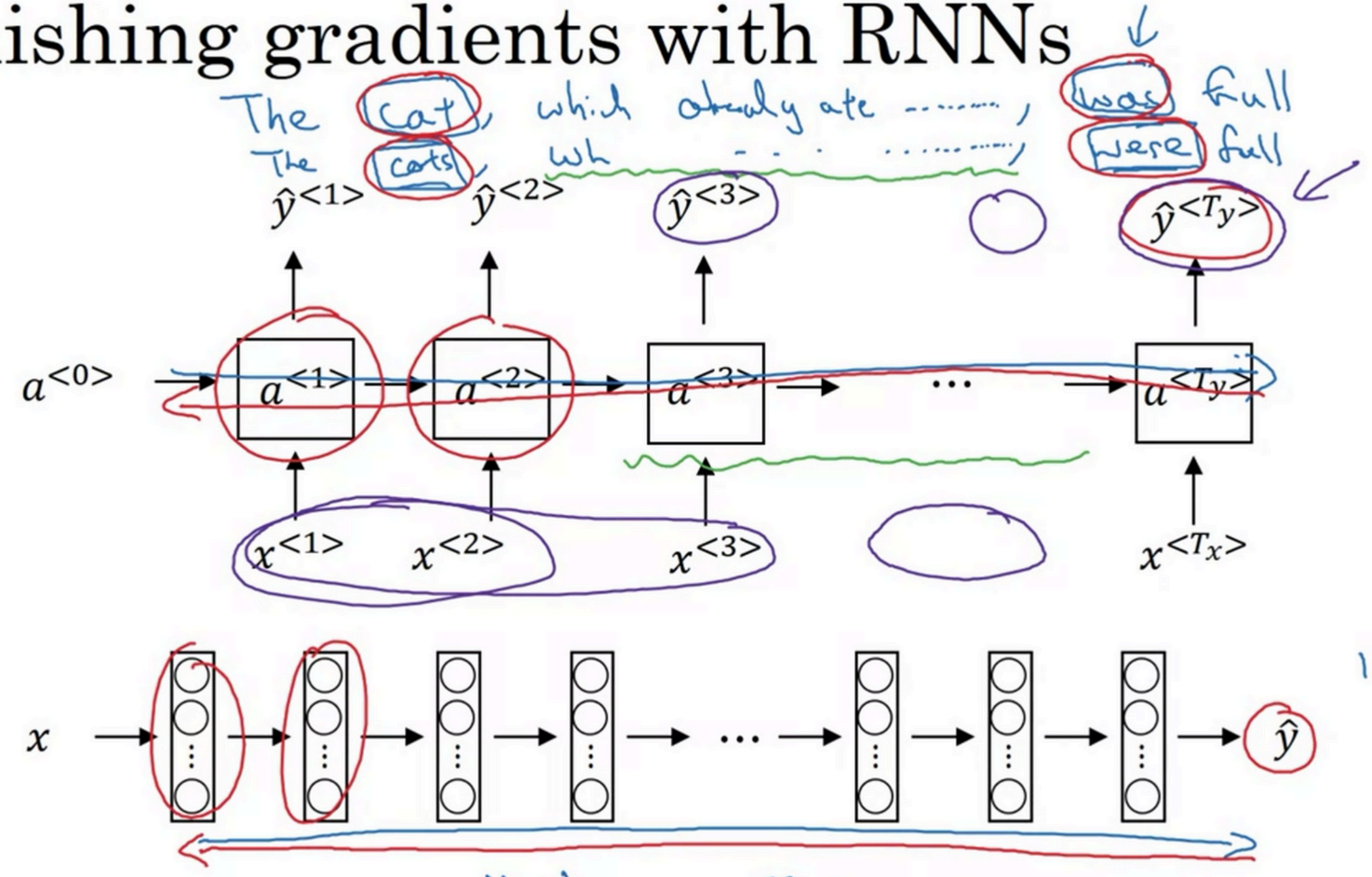
# Forward propagation and backpropagation




$$\mathcal{L}^{<t>}(\hat{y}^{<t>}, y^{<t>}) = -y^{(t)} \log \hat{y}^{(t)} - (1-y^{<t>}) \log (1-\hat{y}^{(t)})$$

$$\mathcal{L}(\hat{y}, y) = \sum_{t=1}^{T_y} \mathcal{L}^{<t>}(\hat{y}^{<t>}, y^{<t>}) \Leftarrow$$

# Vanishing gradients with RNNs



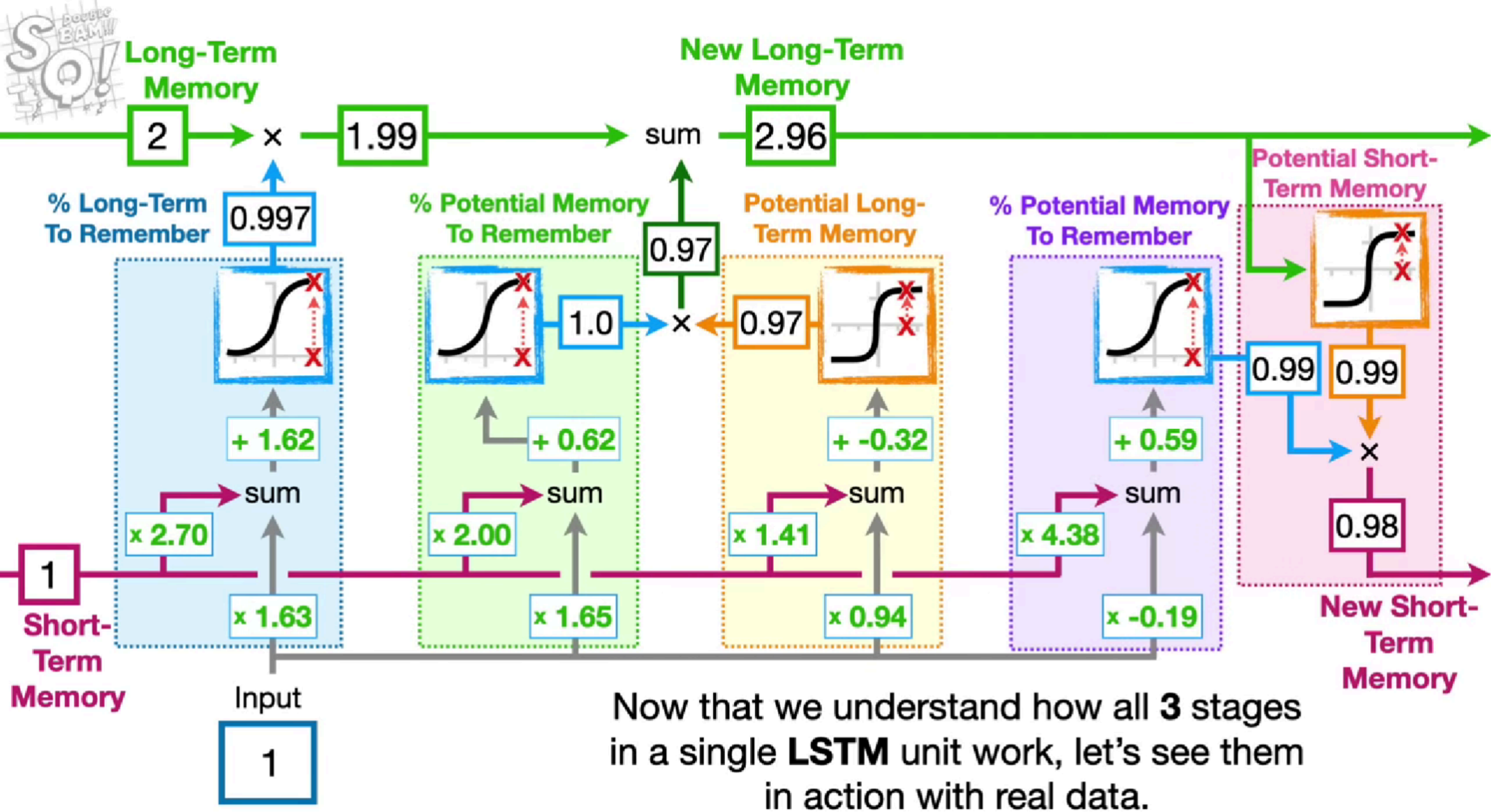

Exploding gradients.

# RNN

El problema. Con el retropropagación convencional a través del tiempo (BPTT; Williams y Zipser, 1992; Werbos, 1988) o el aprendizaje recurrente en tiempo real (RTRL; Robinson y Fallside, 1987), las señales de error que fluyen hacia atrás en el tiempo tienden a (1) explotar o (2) desvanecerse [...]

Mediante un algoritmo eficiente basado en gradientes, se garantiza un flujo constante de error (por lo tanto, ni explotador ni desvaneciente) a través de los estados internos de unidades especiales (siempre que el cálculo del gradiente se trunque en ciertos puntos específicos de la arquitectura; esto no afecta el flujo de error a largo plazo).

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory.


Long-Term Memory
2
x
1.99
sum
New Long-Term Memory
2.96
% Long-Term To Remember
0.997
% Potential Memory To Remember
0.97
Potential Long-Term Memory
% Potential Memory To Remember
Potential Short-Term Memory
1.0
0.97
0.99
0.99
+ 1.62
+ 0.62
+ -0.32
+ 0.59
sum
sum
sum
sum
x 2.70
x 2.00
x 1.41
x 4.38
0.98
1
x 1.63
x 1.65
x 0.94
x -0.19
Short-Term Memory
New Short-Term Memory
Input
1
Now that we understand how all 3 stages in a single LSTM unit work, let's see them in action with real data.

EJEMPLO PLURAL GATOS

# LSTM in pictures

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

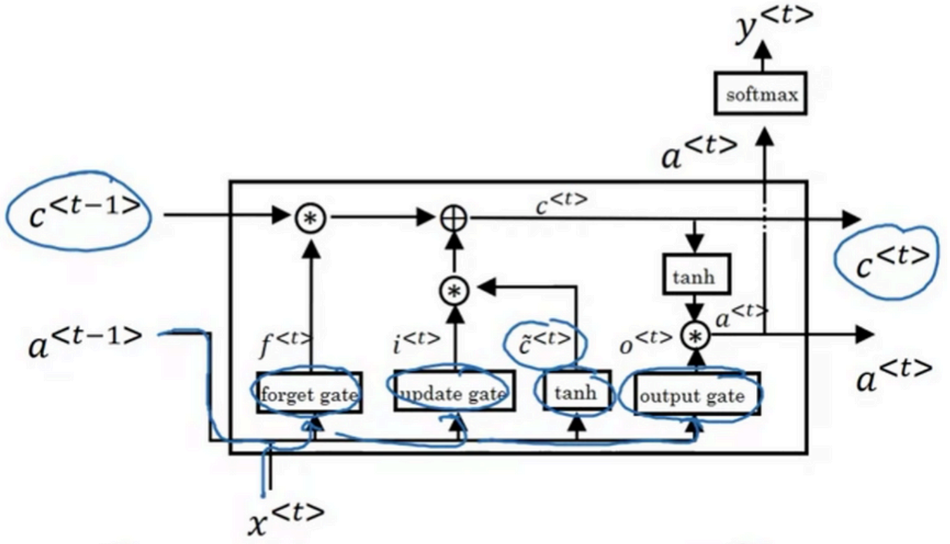
$$\Gamma_u = \sigma(W_u[a^{<t-1>}, x^{<t>}] + b_u)$$

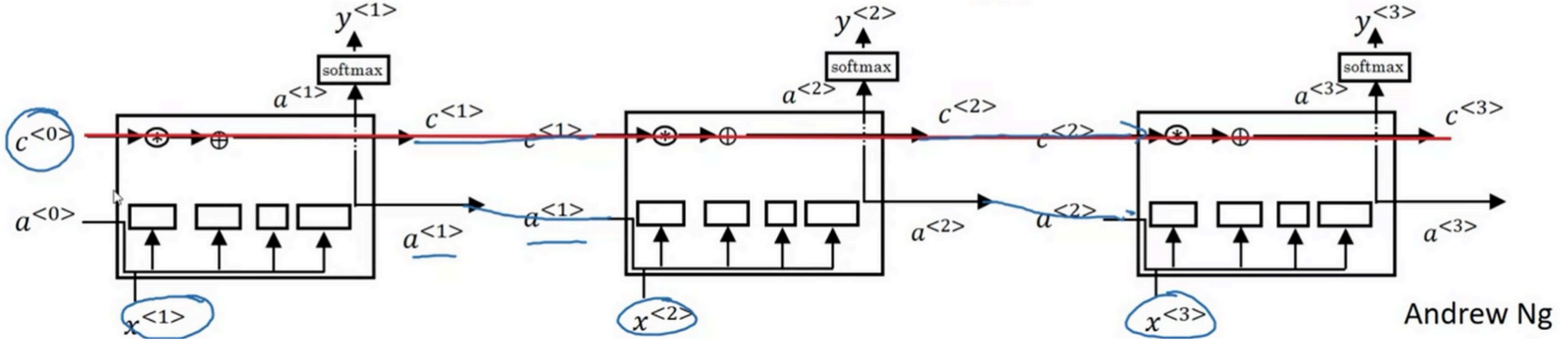
$$\Gamma_f = \sigma(W_f[a^{<t-1>}, x^{<t>}] + b_f)$$

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

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

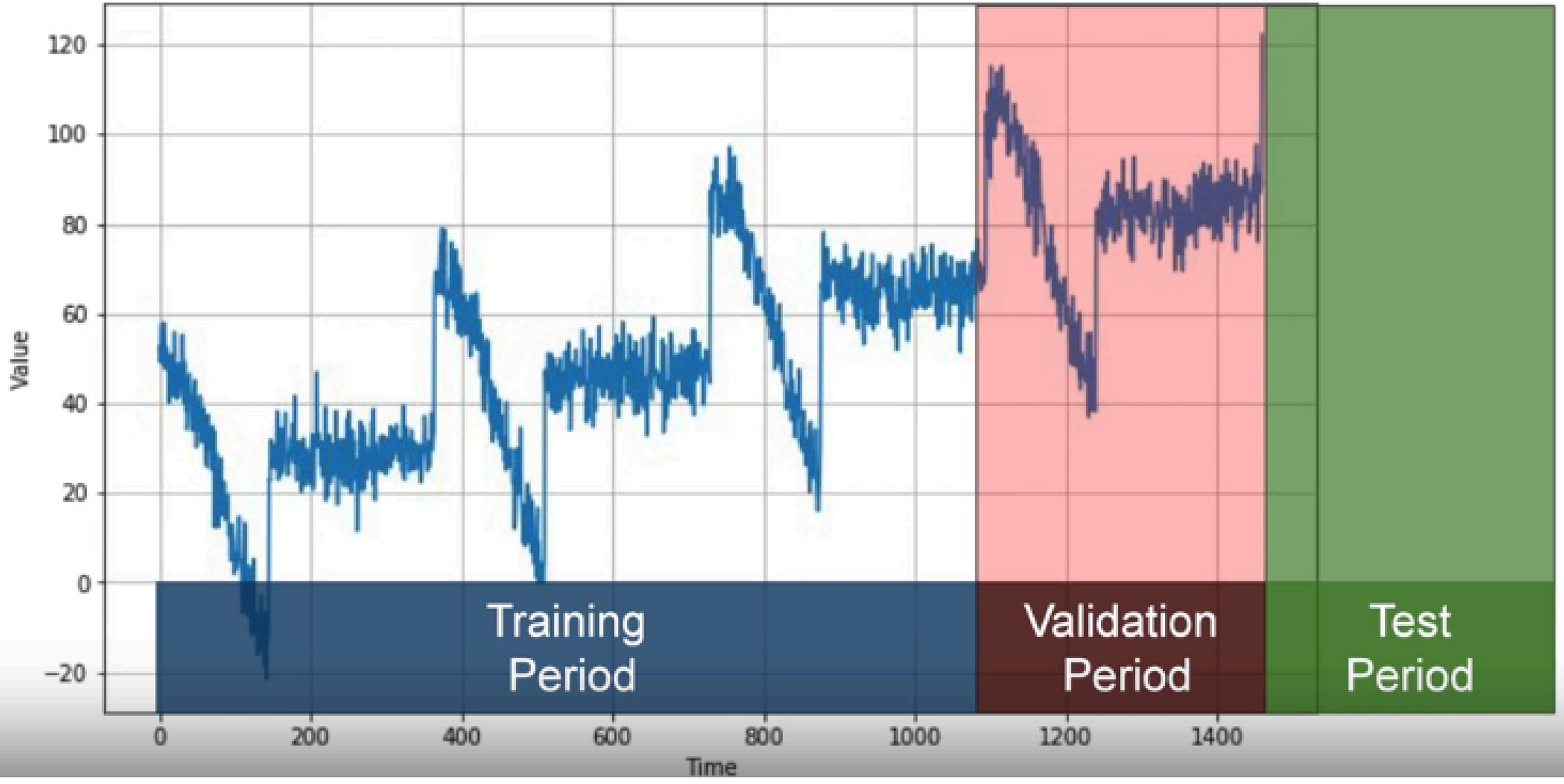
$$a^{<t>} = \Gamma_o * \tanh c^{<t>}$$



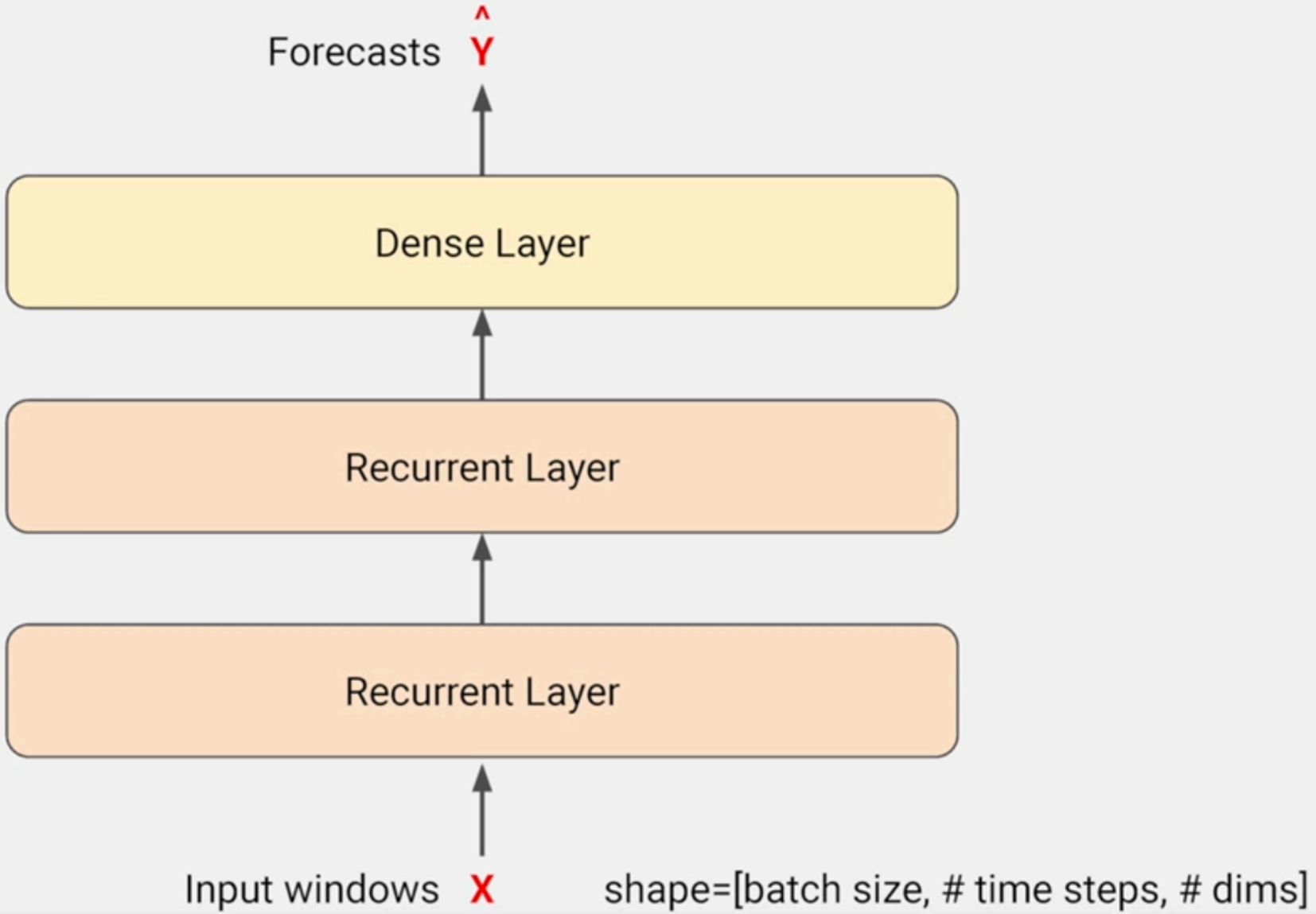




Andrew Ng

# Fixed Partitioning



# Recurrent Neural Network



TU LOGO
AQUÍ

# CONTÁCTANOS

(55) 1234-5678

hola@sitioincreible.com

@sitioincreible

Calle Cualquiera 123, Cualquier Lugar