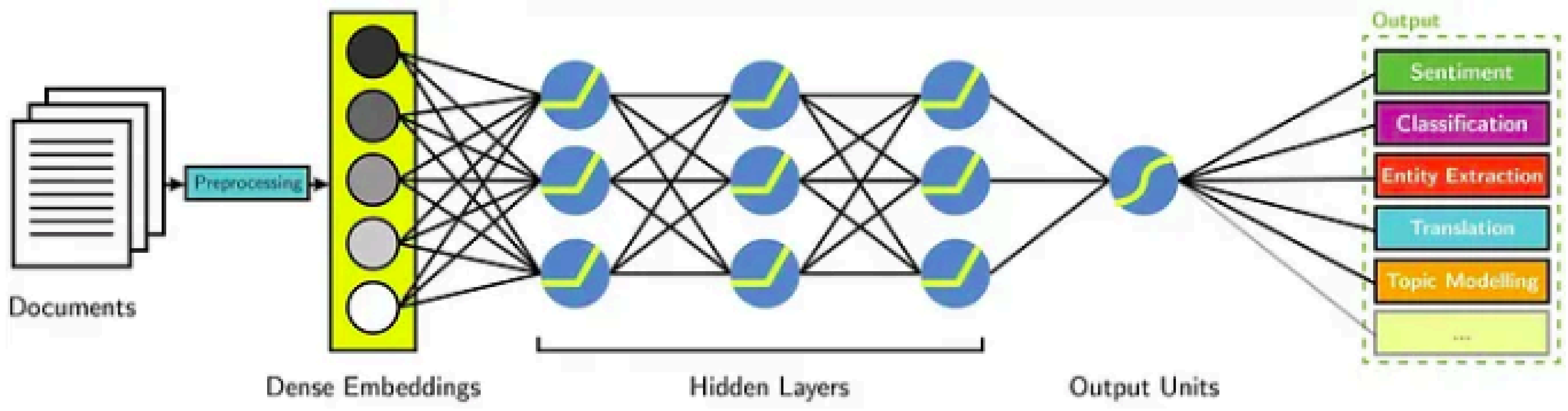
TUTORIAL – 2

NEXT WORD PREDICTION

- For example, if we have some text $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(T)}$, then the probability of this text (according to the Language Model) is:

$$\begin{aligned} P(\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(T)}) &= P(\boldsymbol{x}^{(1)}) \times P(\boldsymbol{x}^{(2)} \mid \boldsymbol{x}^{(1)}) \times \cdots \times P(\boldsymbol{x}^{(T)} \mid \boldsymbol{x}^{(T-1)}, \ldots, \boldsymbol{x}^{(1)}) \\ &= \prod_{t=1}^{T} \underbrace{P(\boldsymbol{x}^{(t)} \mid \boldsymbol{x}^{(t-1)}, \ldots, \boldsymbol{x}^{(1)})}_{\text{This is what our LM provides}} \end{aligned}$$

FFNN

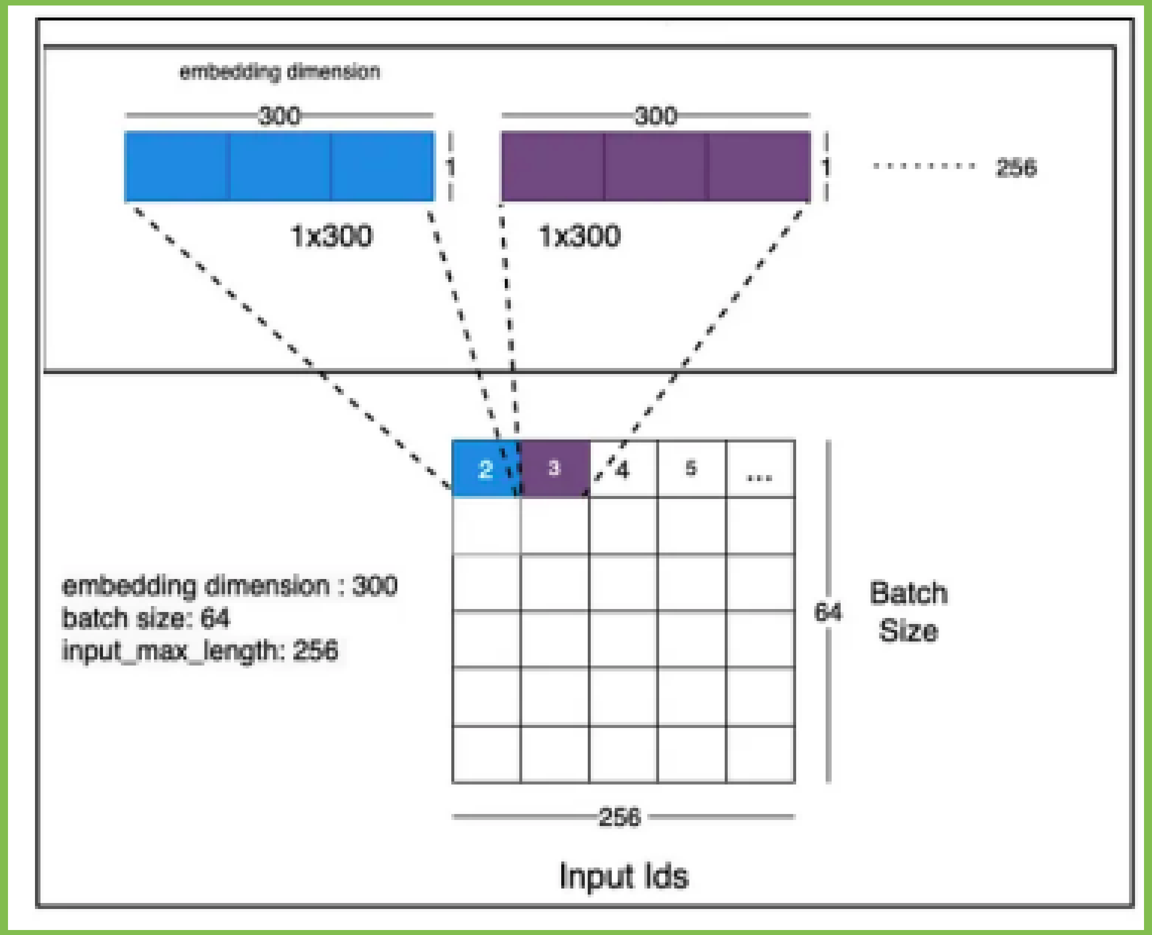


Source: Aylien

Input data: We need a Mathematical representation of the textual data for computational purposes.
Simple Idea????
Simple Idea: One hot encoding.
Can we do better? Can we learn the embeddings of the words as we train our model. We can use pytorch for embeddings

Input data: We need a Mathematical representation of the textual data for computational purposes.
Simple Idea????
Simple Idea: One hot encoding.
Can we do better? Can we learn the embeddings of the words as we train our model.

Embedding layer is also a Matrix, and what is the Matrix dimension? PyTorch provides a Module for Embedding layer, which we can initialize with appropriate dimensions



$$h_t = \sigma(\mathbf{W}_1 \mathbf{x}_t + \mathbf{b}_1)$$

$$\hat{y}_t = \mathrm{softmax}(\mathbf{W}_2 \mathbf{h}_t + \mathbf{b}_2)$$

W_1 and b_1 are the weight matrix and bias vector for the hidden layer, respectively. They are learned during the training process during back propogation

Cross-Entropy Loss:

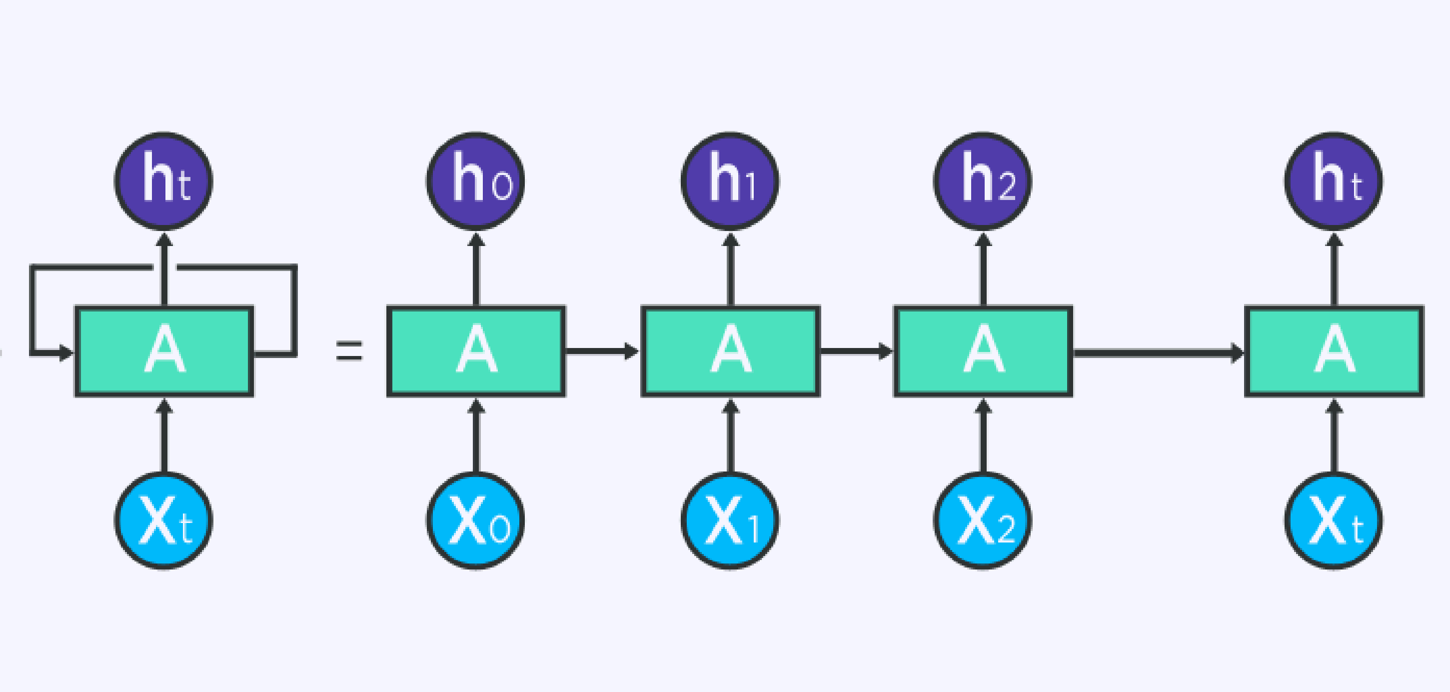
$$\text{Loss} = -\sum_{i=1}^{N} y_i \cdot \log(\hat{y}_i)$$

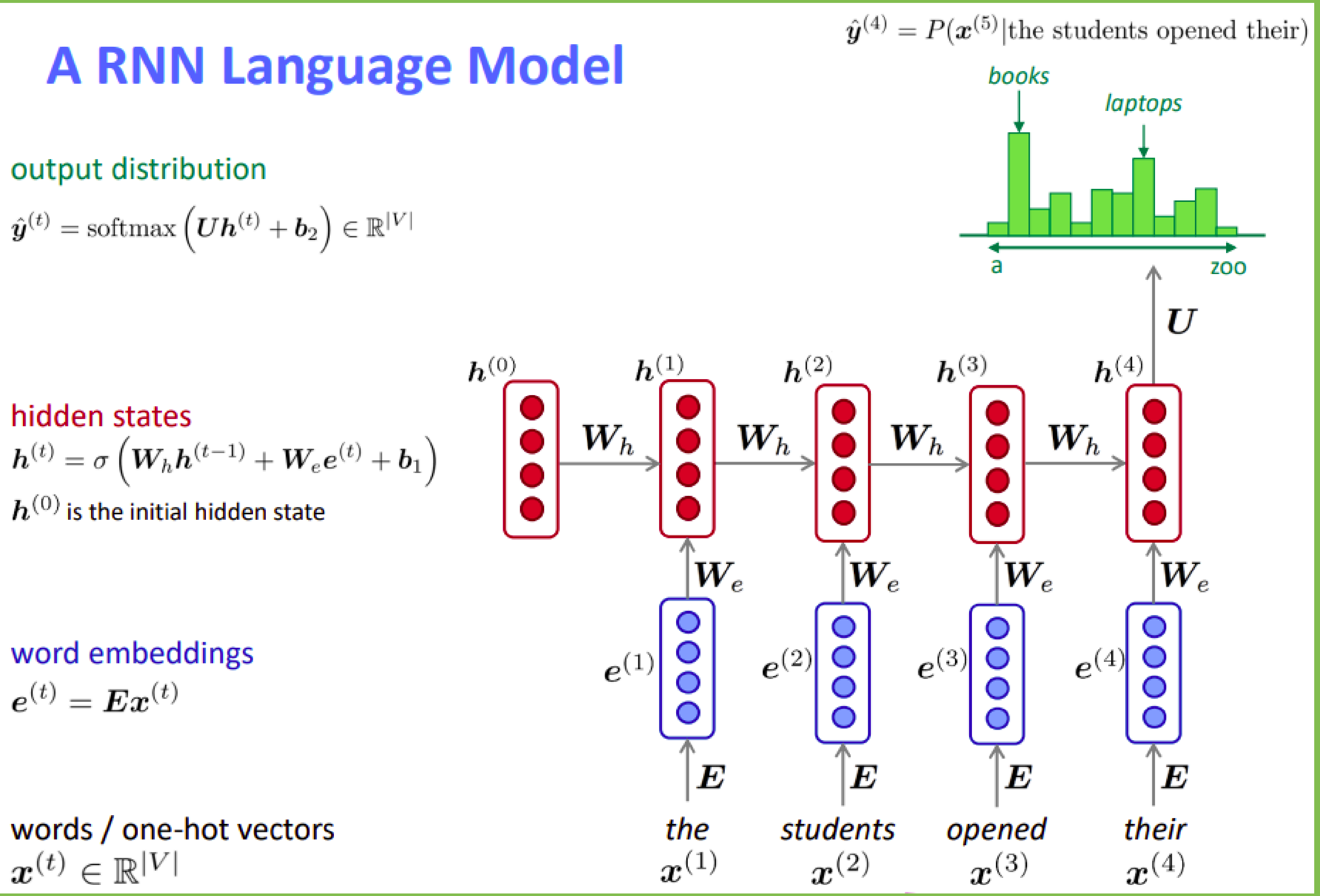
CE Loss and Adam optimizer can be used for back propogating the loss

$\hat{y}_t$ is the predicted probability distribution over all the possible output words at time step t.

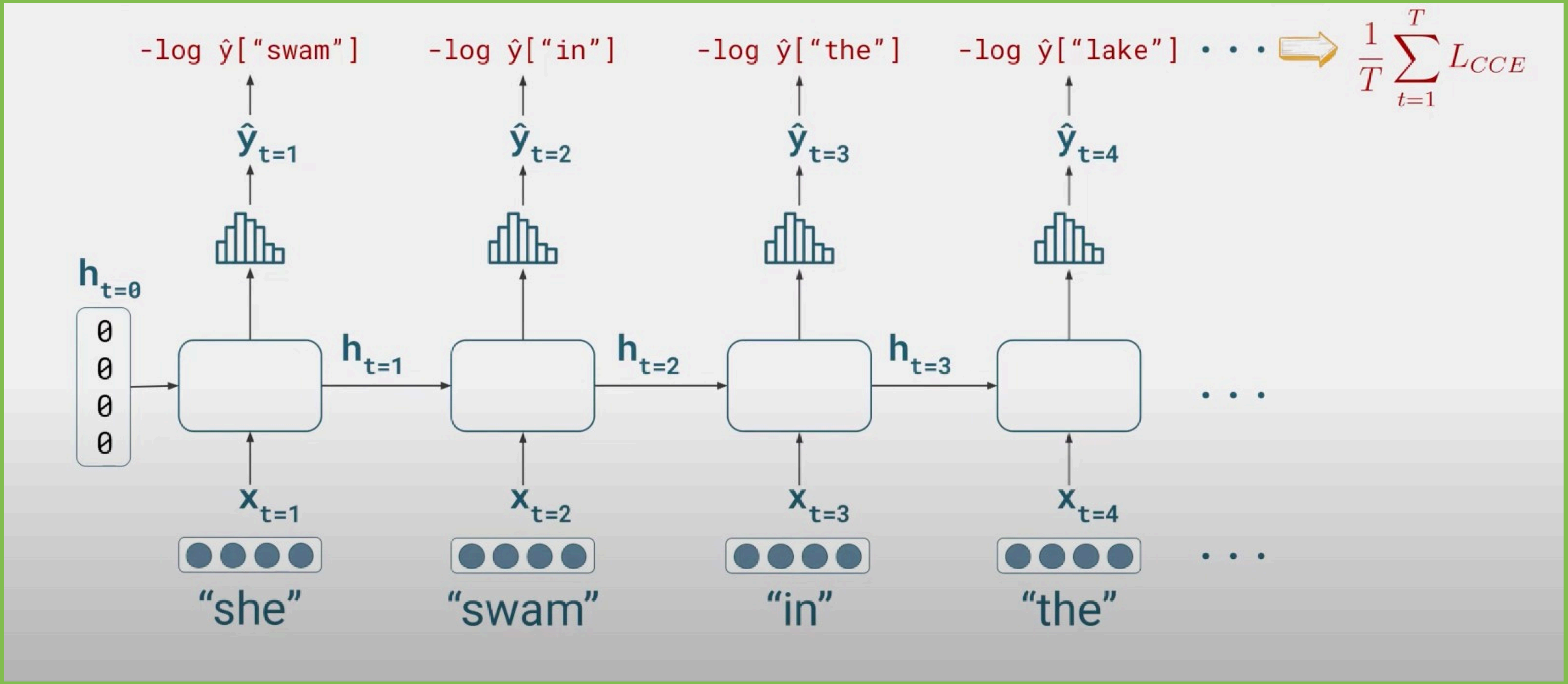
RNN

concept of memory - introduces a hidden state that carries information from previous inputs.

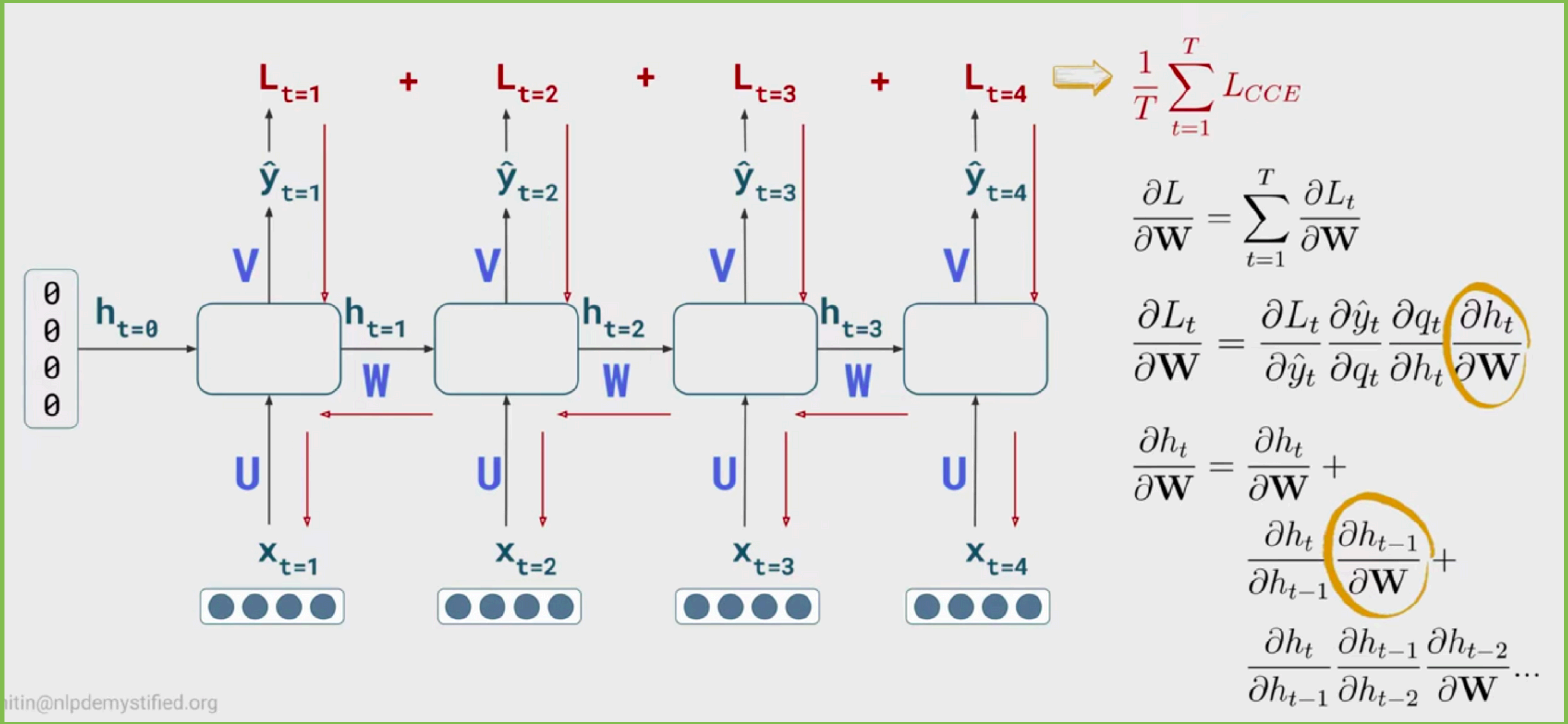




TRAINING RNN



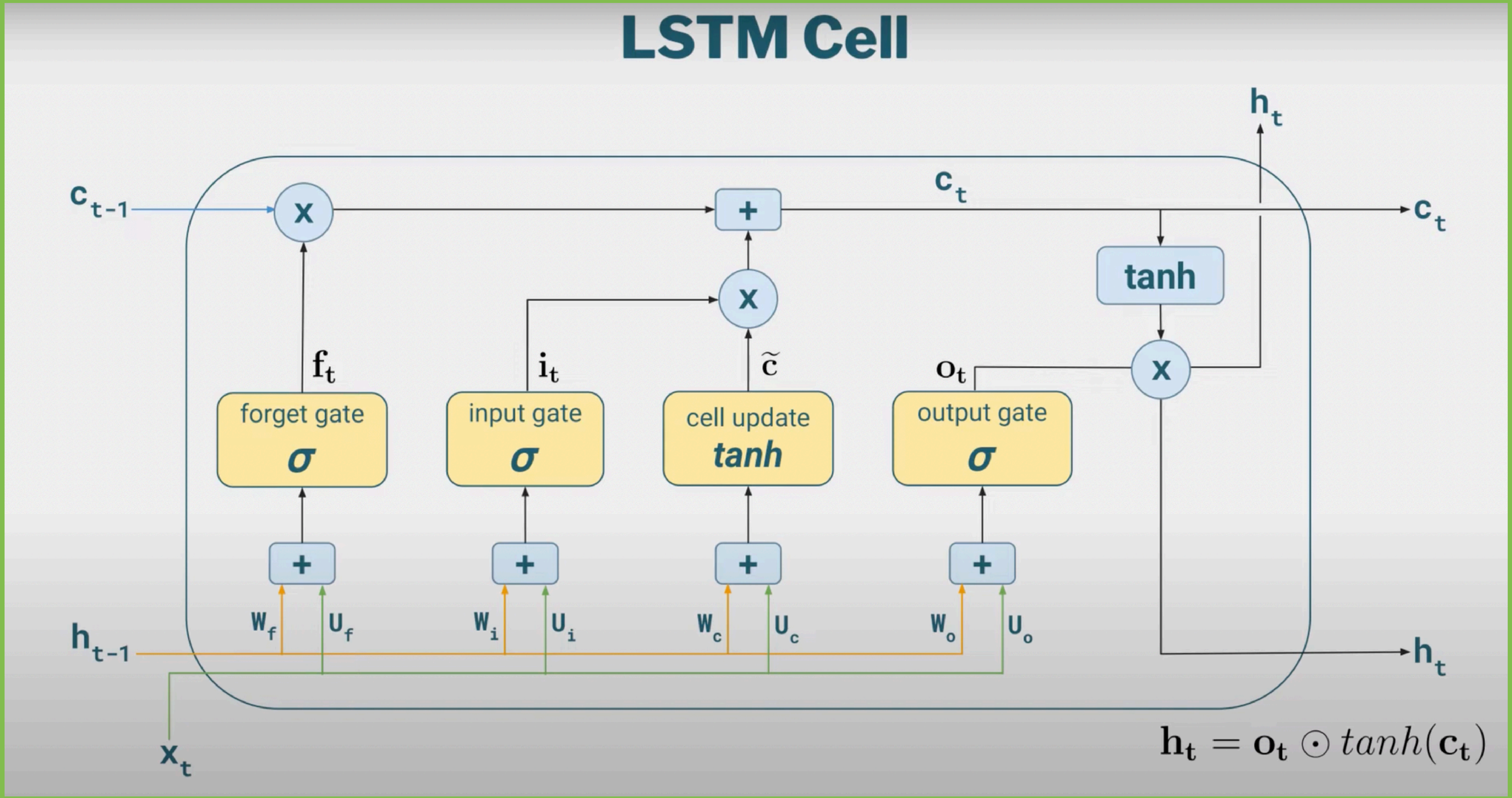
BACKPROPOGATION



DISADVANTAGE

He was supposed to meet his friends at the show but something came up at work, so he called them

LSTM


LSTM Cell
c_{t-1}
h_{t-1}
x_t
c_t
h_t
x
+
tanh
f_t
i_t
$\tilde{c}$
o_t
forget gate σ
input gate σ
cell update tanh
output gate σ
W_f
U_f
W_i
U_i
W_c
U_c
W_o
U_o
$\mathbf{h}_t = \mathbf{o}_t \odot tanh(\mathbf{c}_t)$

DOUBTS?