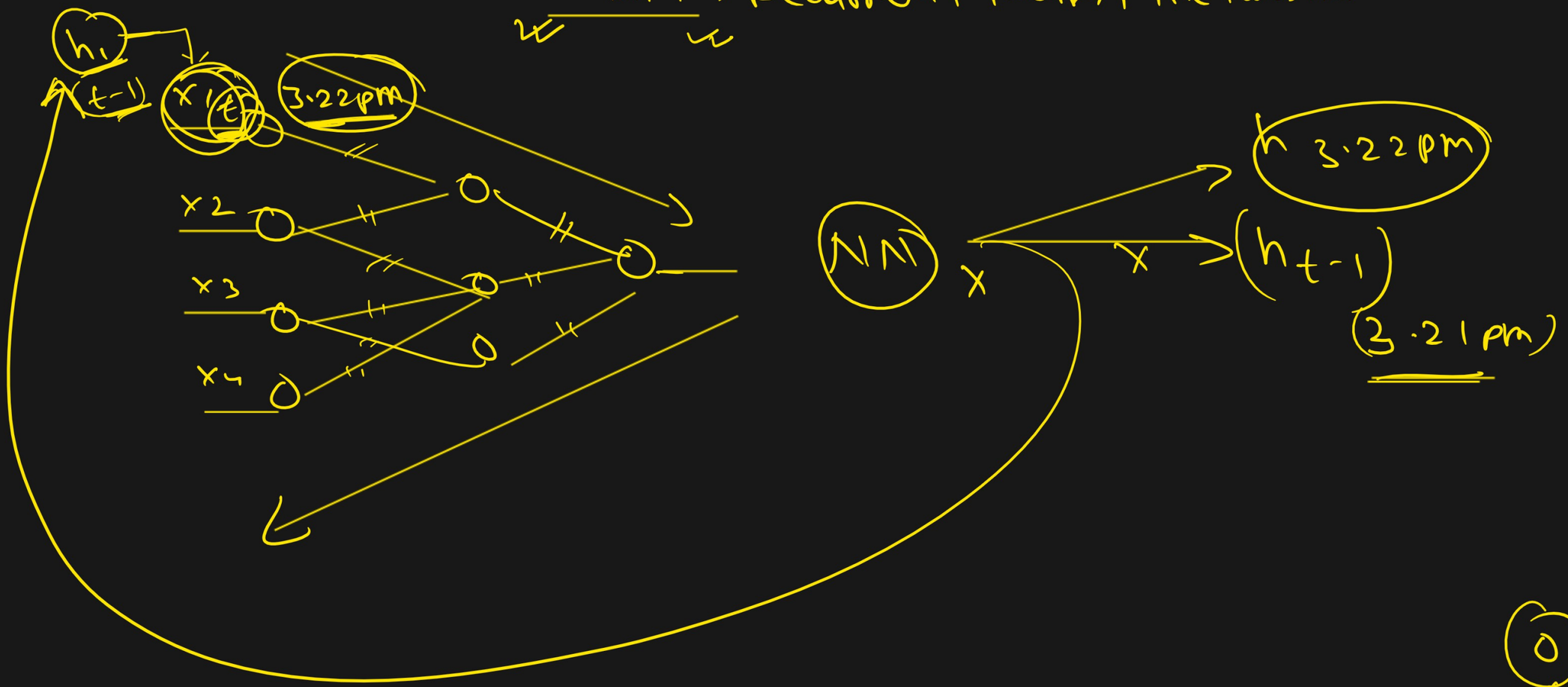
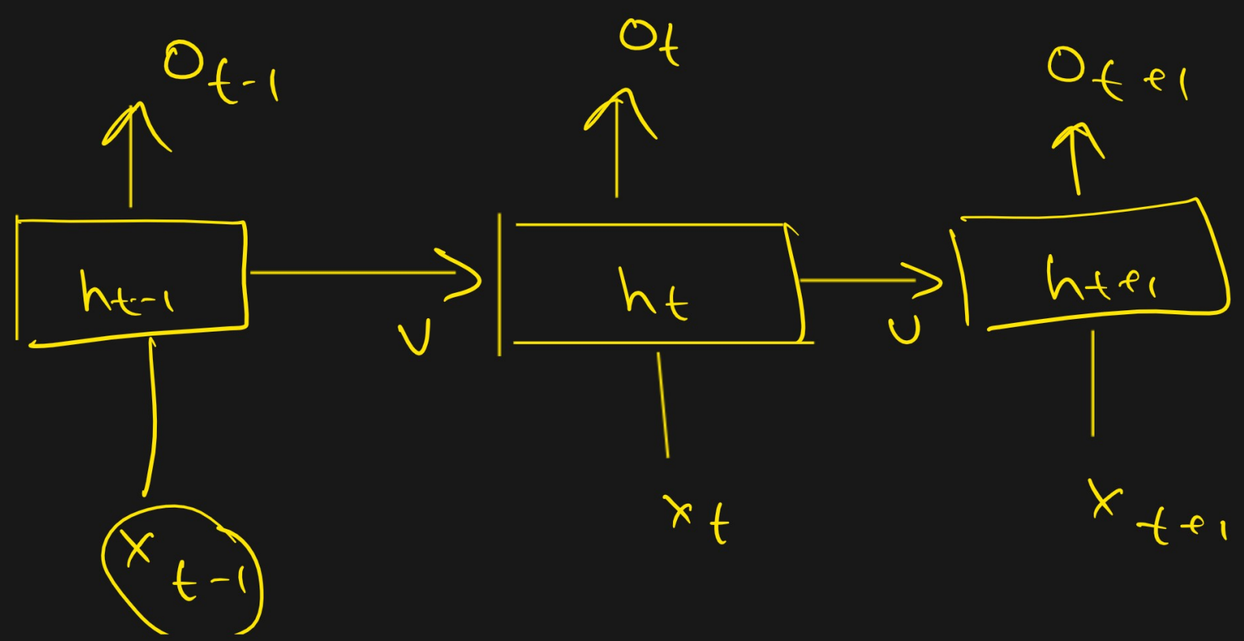
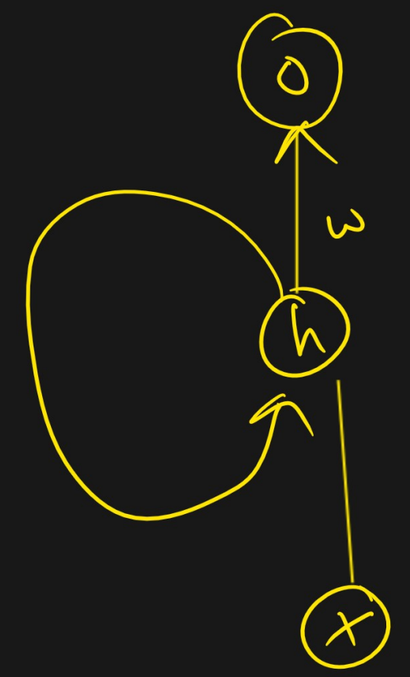
RNN → Recurrent Neural Network.



RNN





Forget gate

input gate

Candidate memory

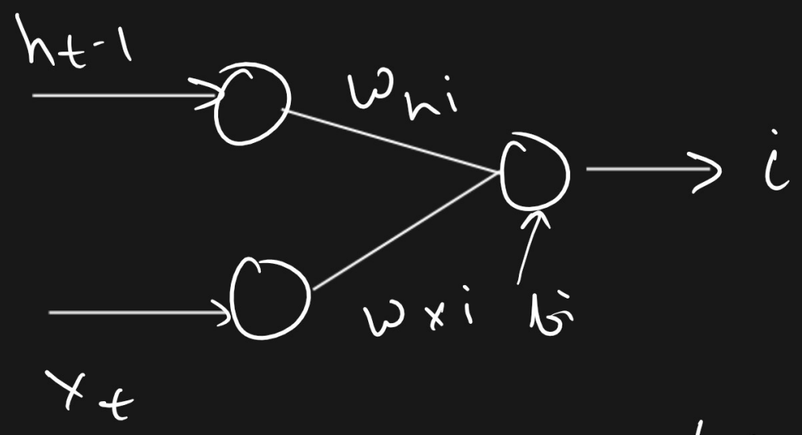
Output gate

tanh

tanh

f_t = Decide what to forget from the previous cell

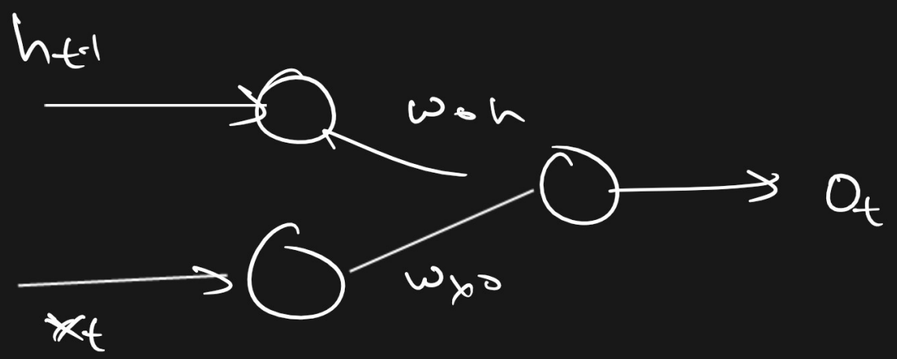
$$f_t = \sigma\left(\omega_f \cdot x_t + h_{t-1}\omega_h + b\right)$$



$$i = \sigma\left(h_{t-1}\omega_{hi} + x_t \cdot \omega_{xi} + b_i\right)$$

$$\tilde{C}_t = \tanh\left(x_t\omega_c + h_{t-1}\omega_c + b_c\right)$$

→ Decide what new information I should to store



$$o_t = \sigma\left(h_{t-1} * W_{ho} + x_t W_{xo} + b_o\right)$$

→ Decide what to output as a new hidden state

$$\left(\frac{dL^{(t)}}{dW_f}, \frac{dL}{dW_i}, \frac{dL}{dC_c}, \frac{dL}{dW_o}\right)$$

$$\left(W_t = W_{t-1} - \eta \frac{\partial L}{\partial W_x}\right)$$

c = help

h e l p ?

c = ['h' , 'e' , 'l' , 'p']
0 1 2 3

input size = 4

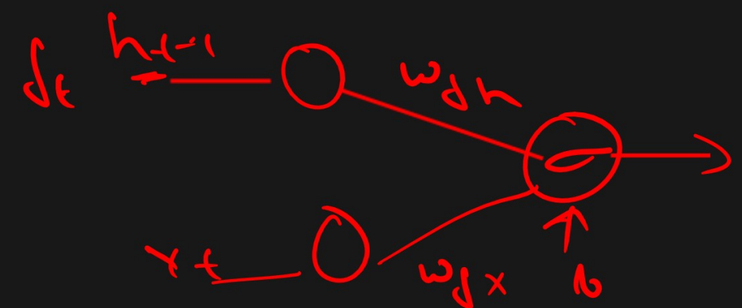
$x_t = \underline{h} = [1, 0, 0, 0]$

$h_{t-1} = [0, 0]$

$c_{t-1} = [0, 0]$

w_f , w_i , w_c , w_o

→	h	[1, 0, 0, 0]
=	e	[0, 1, 0, 0]
	l	[0, 0, 1, 0]
	p	[0, 0, 0, 1]



$f_t = \sigma(h_{t-1} \cdot w_{fh} + x_t \cdot w_{fx})$

$= [0.8, 0.3]$

forget Gate → [0.8, 0.3]

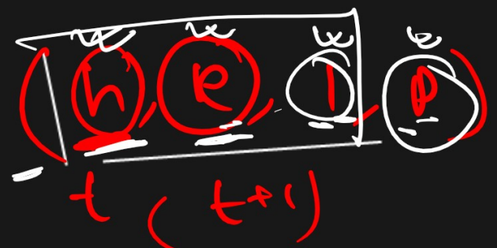
input Gate → [0.5, 0.7]

Candidate mem → [0.2, -0.1]

C_t (update cell state) → [0.1, -0.07]

o_t (output gate) → [0.3, 0.6]

H_t (hidden state) → [0.013, -0.041]



$h \rightarrow y_t$

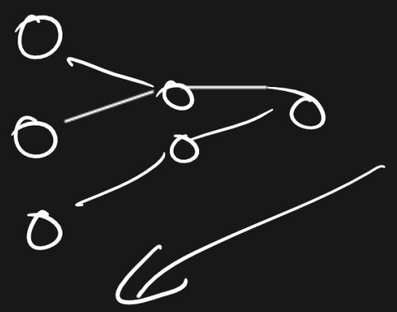
y_t = Softmax $(w \cdot h_t + b)$ → $\hat{y} = ?$

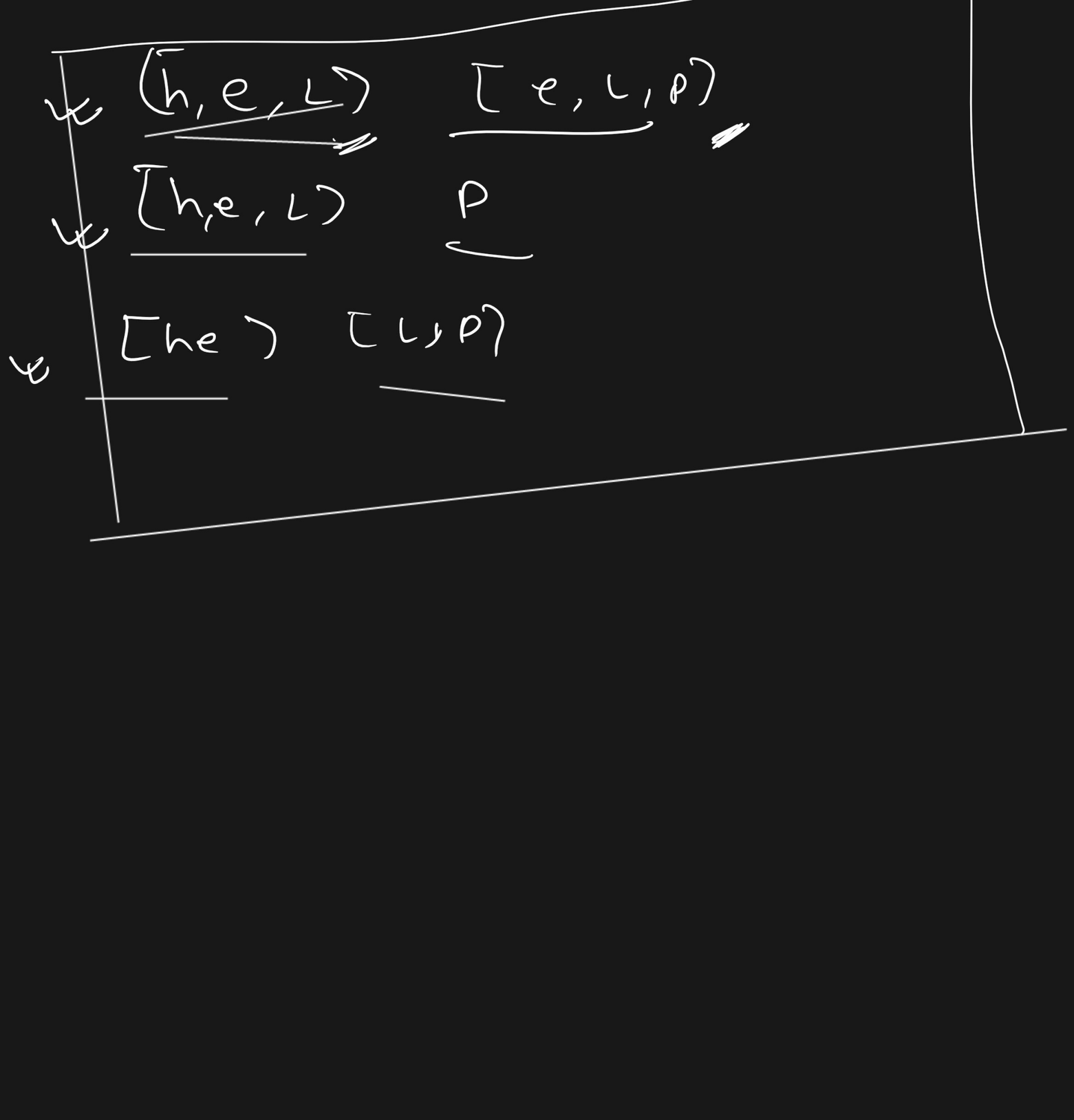
$x_t = [0, 1, 0, 0]$

$h_{t-1} = [0.013, -0.041]$

$C_{t-1} = [0.1, -0.07]$

(dL)




(h, e, L) [e, L, p)
[h, e, L) p
[he) [L, p)