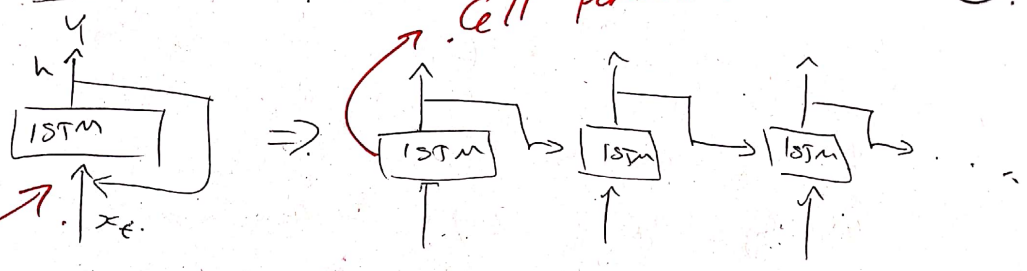
PSEUDO CODE



Make Base on nn.module.

Can separate into Cell + wrapper for multi layer networks.

→ need to put into encoder decoder format.

∴ 2 classes. LSTM Base unit, LSTM network.

LSTM: Basis

follow eqns from networking paper.

$$i_t = \sigma(W_{xi} * \mathcal{X}_t + W_{hi} * \mathcal{H}_{t-1} + W_{ci} \circ C_{t-1} + b_i)$$
$$f_t = \sigma(W_{xf} * \mathcal{X}_t + W_{hf} * \mathcal{H}_{t-1} + W_{cf} \circ C_{t-1} + b_f) \quad (1)$$
$$C_t = f_t \circ C_{t-1} + i_t \circ \tanh(W_{xc} * \mathcal{X}_t + W_{hc} * \mathcal{H}_{(t-1)} + b_c)$$
$$o_t = \sigma(W_{xo} * \mathcal{X}_t + W_{ho} * \mathcal{H}_{t-1} + W_{co} \circ C_t + b_o)$$
$$\mathcal{H}_t = o_t \circ \tanh(C_t)$$

will needed filters for. W....

→ per in picture. → use Conv 2D.

Class ConvLSTM Base.

def __init__ . . . . (Dims, channel in, hSize, kernel size)

Self.param = param for. Dims, Channel in each.

Self.padding = (filter // 2) ← to keep layers same size as in convolving

Self.Wxi..... = Conv2D(input, hidden, kernelSize, Stride=1, padding.)

TODO: Issue on the Biases

W_{ci}, W_{cf}, W_{co} ⇒ not Conv as are multiplicative

TODO: function how to initialize as zero variables.

Use. ModuleDict for Storage of Layers

def forward(Self, x, h, c):

↳ Implementation of eqns (1).

returns. h_t, C_t.

need function for sending arrays of 0s through first to initialize layers →

PSEUDO code.

LSTM Main

operates over LSTM unit.

class cLSTM Main (nn. module).

def ~~_init_~~ (self, shape, input chan, hidden chan, kernel, step ~~s~~

batch E.G.

self ~~.cell. lstm unit~~ params . . .

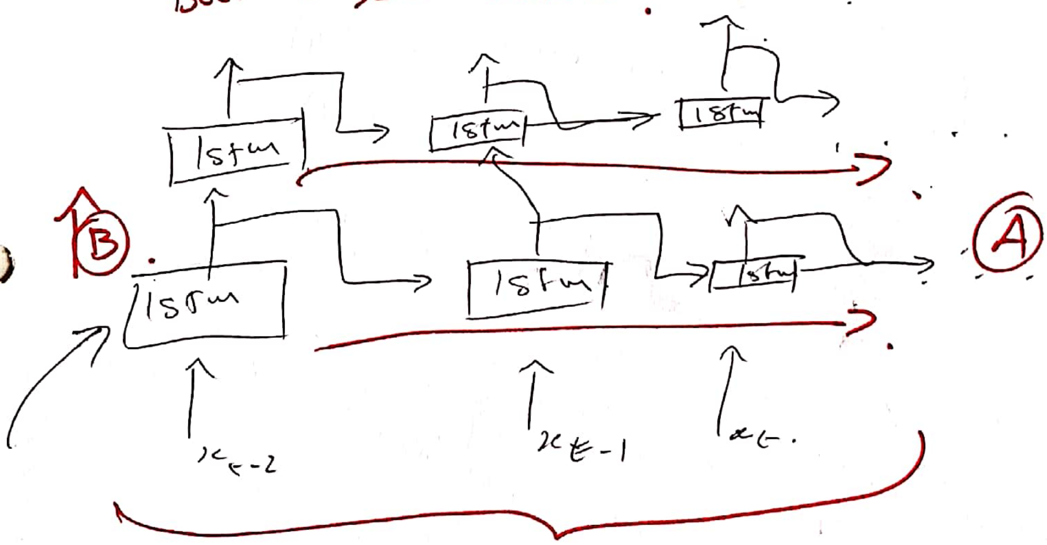
⟶ .

for self. cell ⟶ for multilayer. network.

self. unit list append (lstm unit . . . . . .).

self. unitlist = nn. modulelist (unit_list). ←.

Bool → save. hidden. ←



Bool to save all output.

Bottom row must be completed before.

∴ pertto for loop over B.

for loop over A.

→ ht into list.

→ feed ht from previous layer into input.