Lec 1-56

RNN

	review	sentiment
x_1 =	movie was good	1
x_2 =	movie was bad	0
x_3 =	movie was not good	0

↓

Total 5 unique words

movie was good bad not

[1 0 0 0 0] → "movie" word

[0 1 0 0 0] → "was" word

x_1 = [[1 0 0 0 0], [0 1 0 0 0], [0 0 1 0 0]]

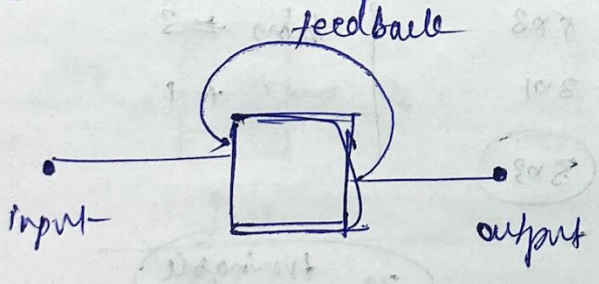
t=1 t=2 t=3

5×5 ← no. of features.

no. of times steps

x_2 = (3×5)

x_3 = (4×5)

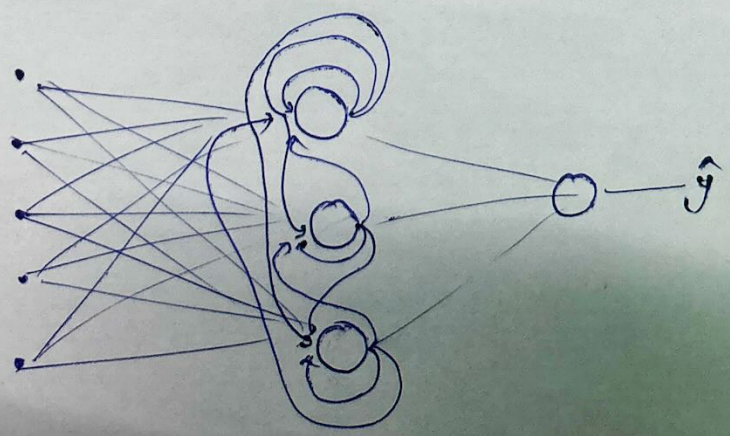


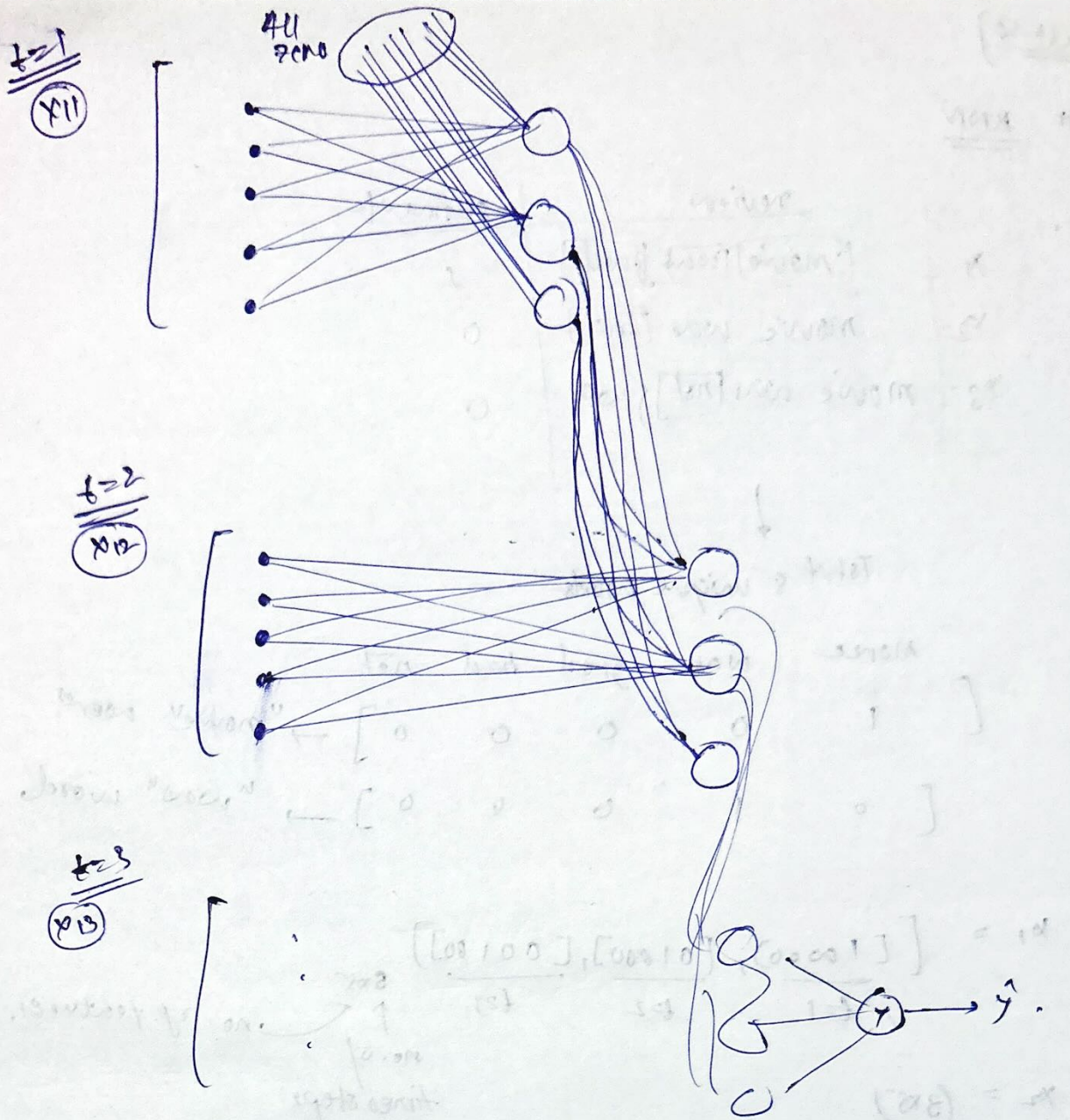
x_1

x_{11}, t_1

x_{12}, t_2

x_{13}, t_3





weights = 5×3 | biases = 3
3×1 | + 1
recurrent layer → 3×3.

31 trainable parameters

from keras.layers import Dense, SimpleRNN.

model.add(SimpleRNN(nodes, shape=(5,3)))

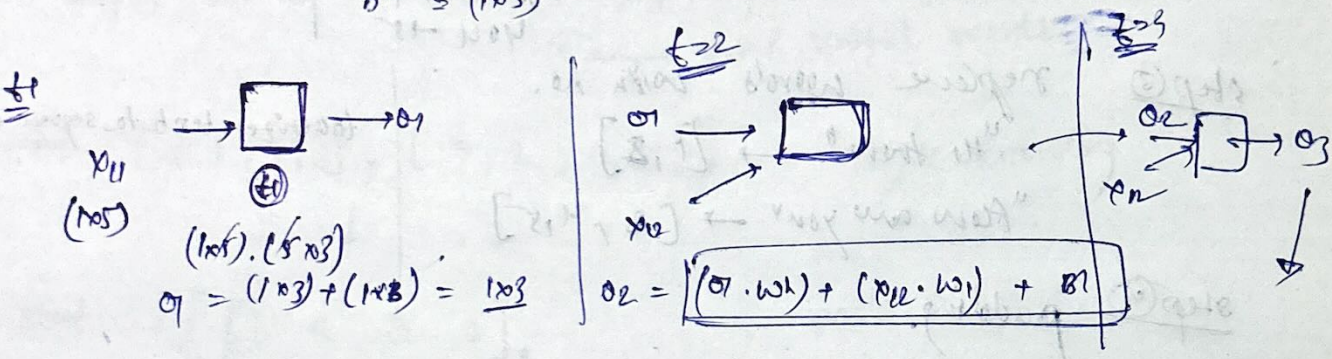
maximum time stamps. (eg longest sentence has 5 words, then 5)

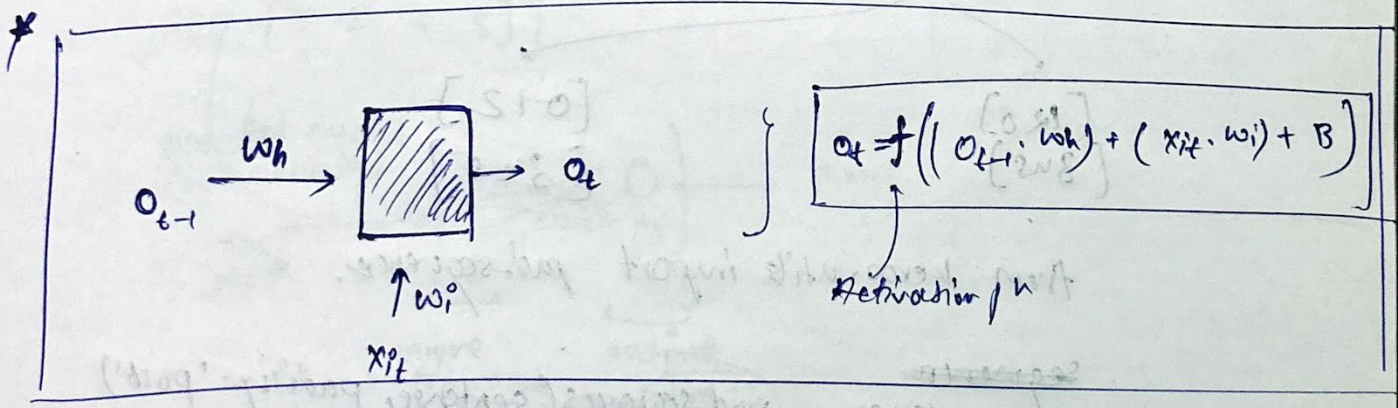
features (words) in one

$W_h = (3\times3)$

$W^I = (5\times3)$

$B^I = (1\times3)$





lect 57

text → Numbers

① Integer encoding

② embeddings -

* Integer Encoding

from keras.preprocessing.text import Tokenizer

tokenizer = Tokenizer(oov_token='<OOV>')

step ① create vocab from given inputs.

(let say there are total of N unique words).

step ② Rank those words, ex Hi → 1
there → 2
How → 3
Are → 4
You → 5

tokenizer.fit_on_texts(sentences)

step ③ replace words with no.

"Hi there" → [1, 2]

"How are you" → [3, 4, 5]

tokenizer.texts_to_sequences(sentences)

step ④ padding.

[1 2 0]
[3 4 5]

[0 1 2]
[3 4 5]

from keras.utils import pad_sequences.

sentences = pad_sequences(sentence, padding='post')
='pre'

Embeddings

step ① use ~~Integer Encoding~~

step ② from keras.layers import Embedding

model.add(Embedding(5, output_dim=2, input_length=4))

(each word is represented by 2 numbers)

5 ← no. of unique words in vocab

4 ← no. of words in each sentence.

Sent = ["How are you",
"you are welcome though"]

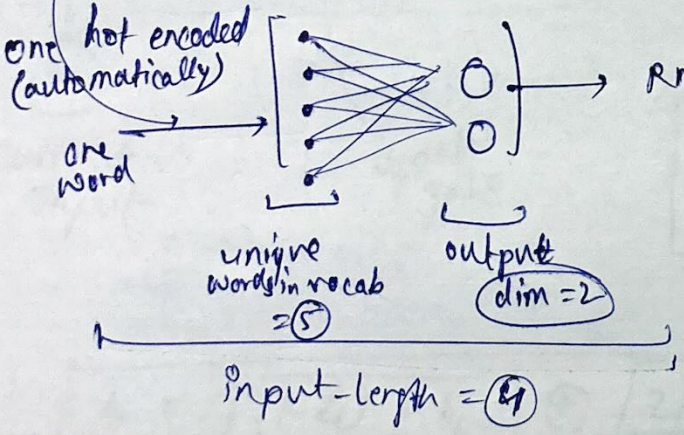
unique words → ① How, ② are, ③ you, ④ welcome, ⑤ though

5 → total unique words

sent = [[1, 2, 3], [3, 2, 4, 5]] Integer Encoding

↓

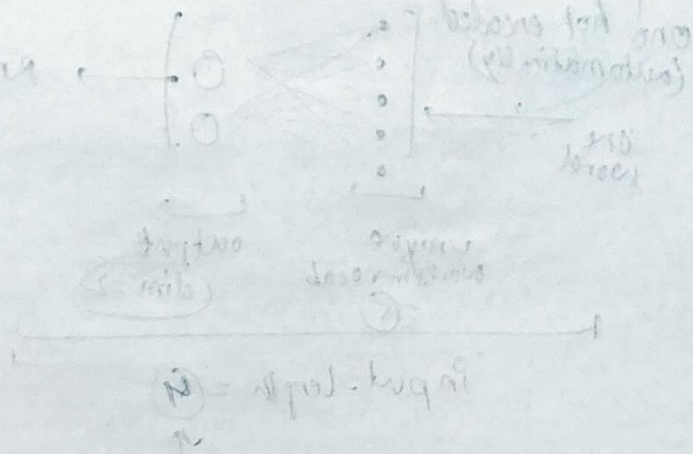
sent = [[1 2 3 0],
[3 2 4 5]] padded (post)



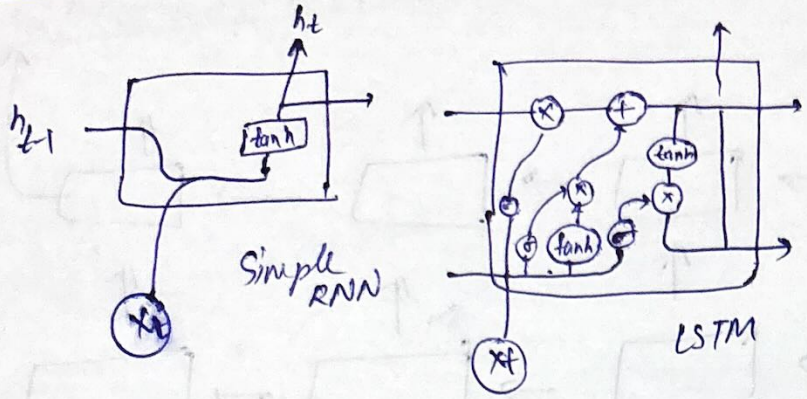
length of each sentence

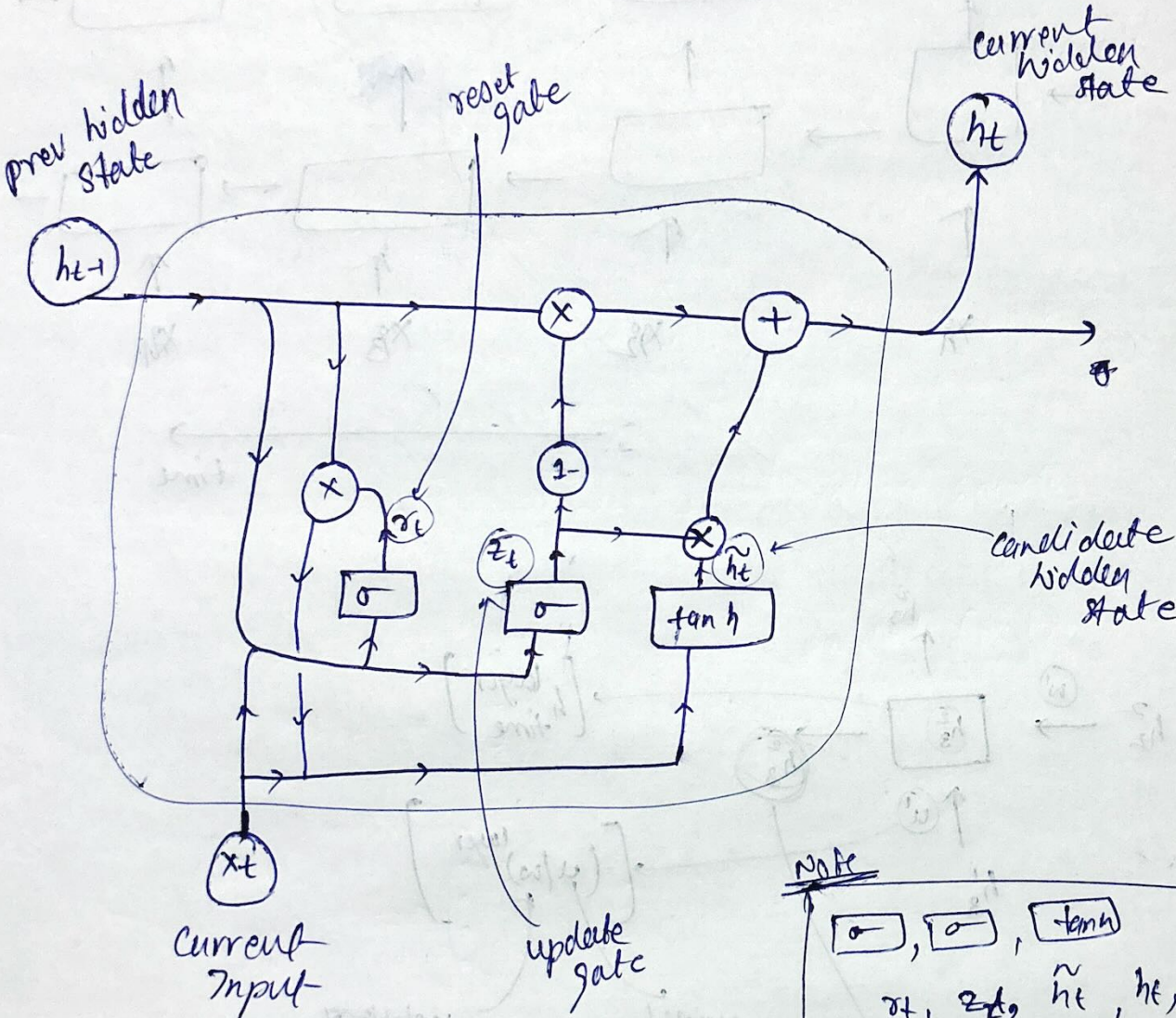
Ref 62]

LSTM



lect 64] # GRU's





Note

σ, σ, tanh

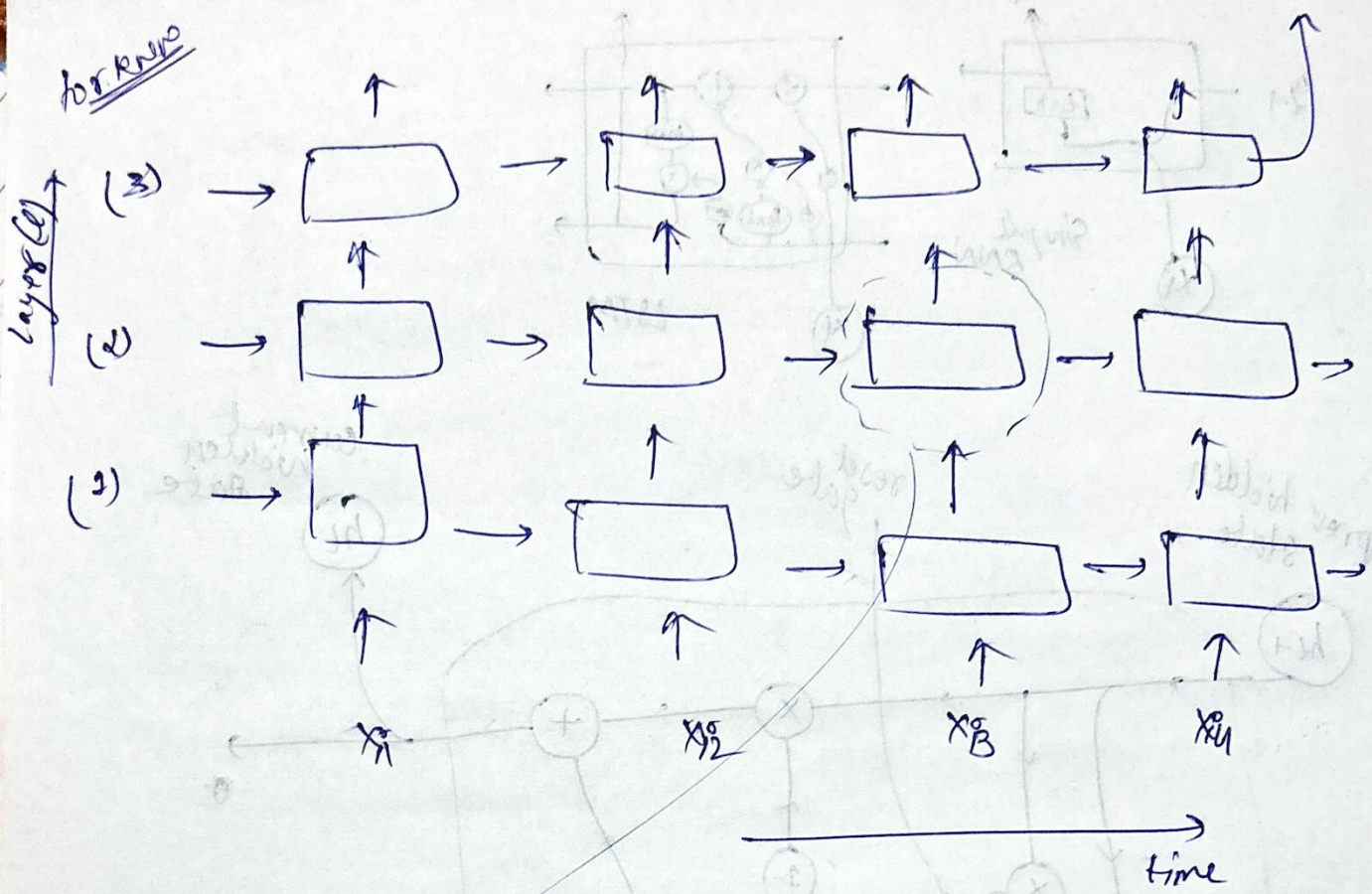
r_t, z_t, $\tilde{h}_t$, h_t, h_{t-1}

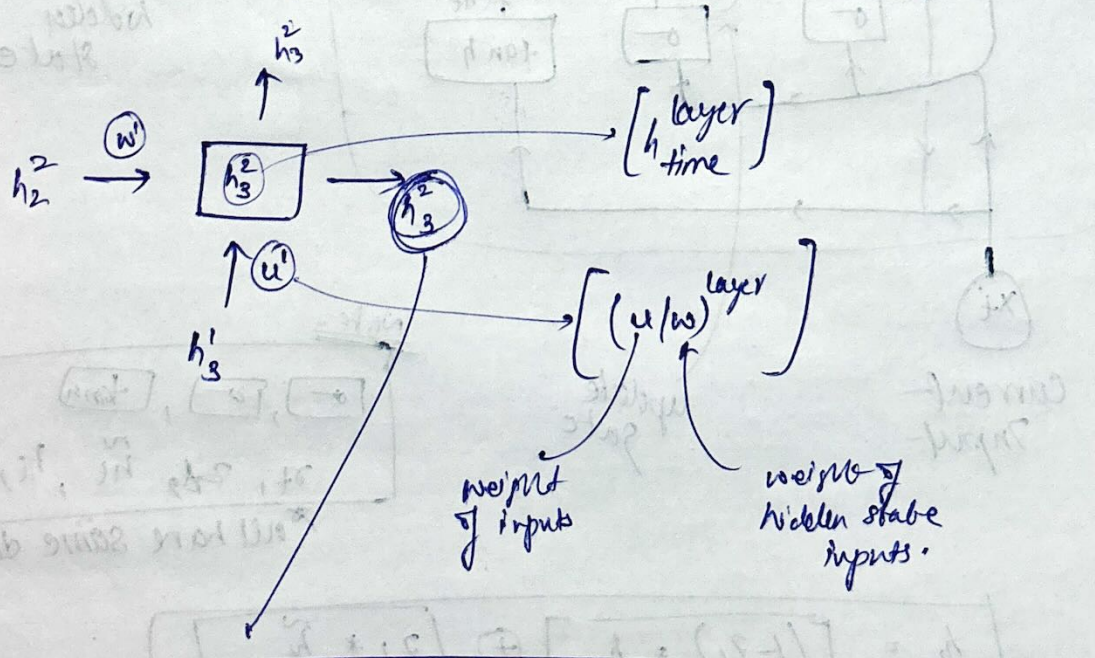
*all have same dim

$$h_t = [(1-z_t) * h_{t-1}] \oplus [z_t * \tilde{h}_t]$$

Refs] # Deep RNN's / LSTM / GRU

for RNN's





RNN

$$h_t^l = h_3^2 = \tanh\left(w^l \cdot h_{t-1}^l + u^l h_t^{l-1} + b \right)$$

Lec66]

Bi-directional RNN/LSTM/GRU