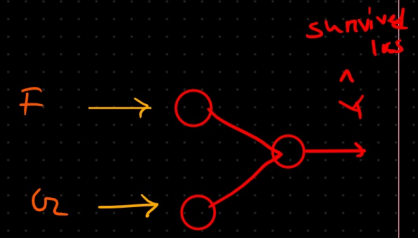
Agenda:

1. RNN – Recurrent Neural Networks
2. What is Sequential Data/ Learning?
3. Some Drawbacks in ANN and CNN
4. Types of RNN Networks
5. Practical Demo of RNN using Keras (Sentiment Analysis)

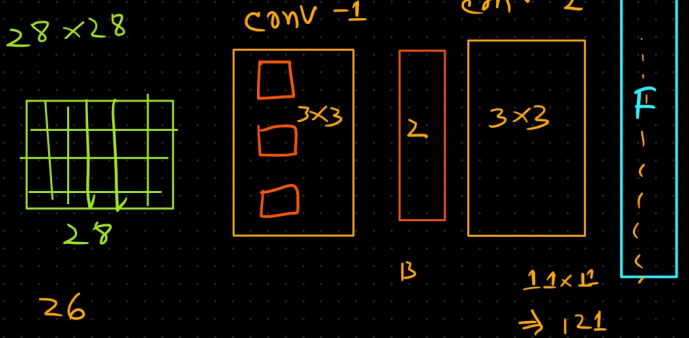
ANN → Tabular Data

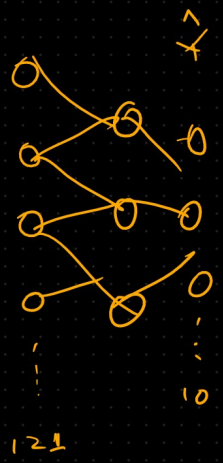
Gr	Fare	Survived
F	20K	1
M	5K	0
-	-	-
-	-	-

X Y

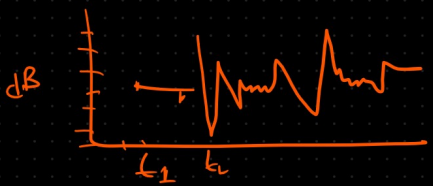


CNN → Image 28





1. My name is Buppy
   t_1 t_2 t_3 t_4
2. I love AI



→ Text
→ Time Series
→ Speech
→ DNA seq

* Why should use RNN?

5	Hi my name is Bappy	0
3	I love AI	1
4	Bappy is My name	0

Total : 12

Hi ⇒ [1, 0, 0, 0, ---- 12]

My ⇒ [0, 1, 0, 0, ---- 12]

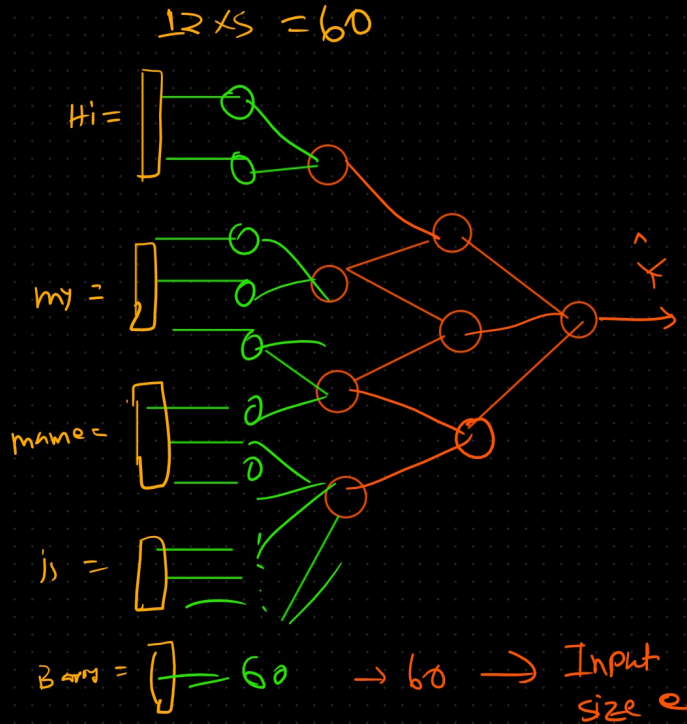
---

(2) [1, 0 0 0 .. 12)

[0, 1, 0 ---]

[0 0, 1 .. 12]

12 × 5 = 60



→ 60 → Input size error!

12 × 3 ⇒ 36

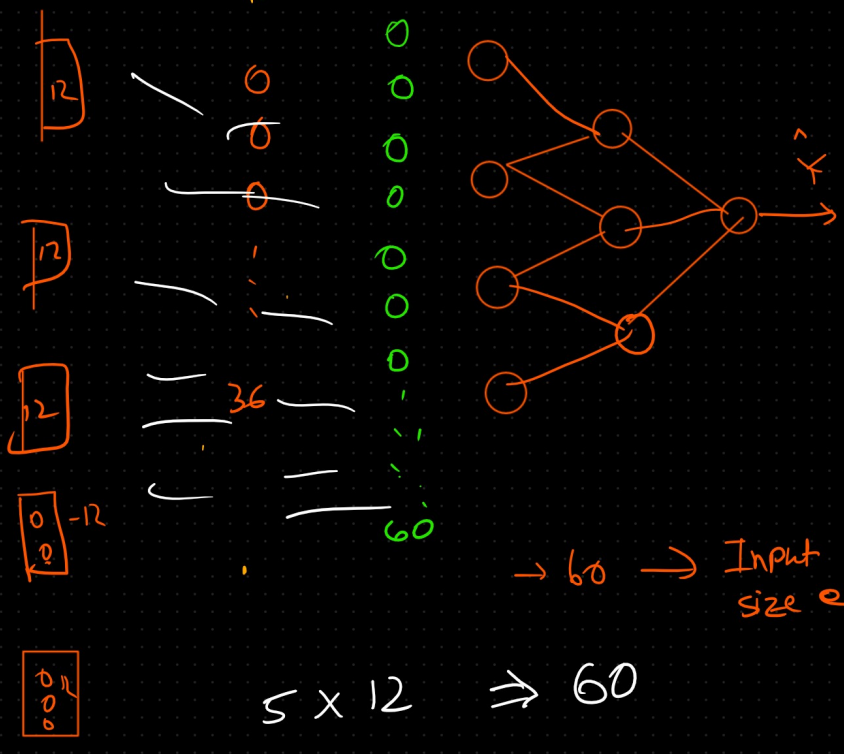
12 × 4 ⇒ 48

Hand coded Solution:

28 × 28

32 × 2

32 × 32



→ 60 → Input size error!

5 × 12 ⇒ 60

* Note all problems:

① Text input → varying size

② zero padding → sparse Matrix → Unecssary computation

③ loosing Sequential Info (context meaning)

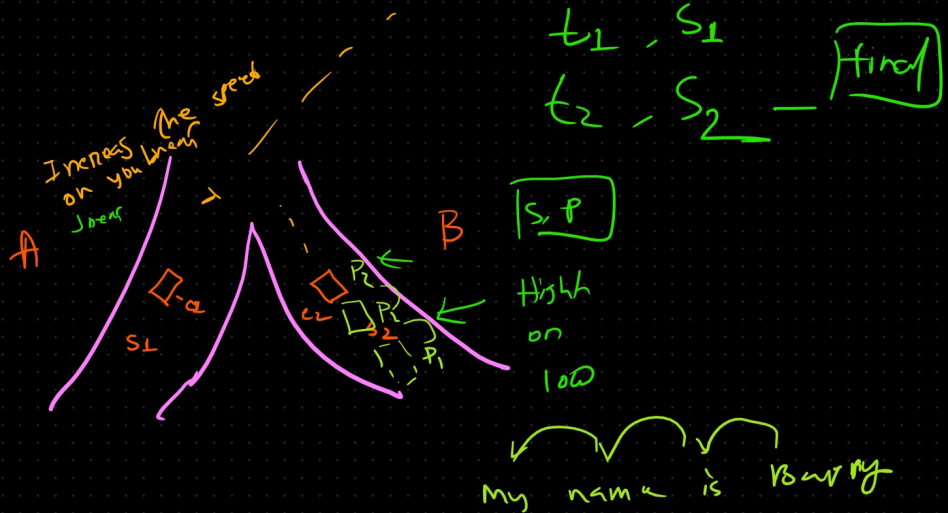
RNN – Recurrent Neural Network

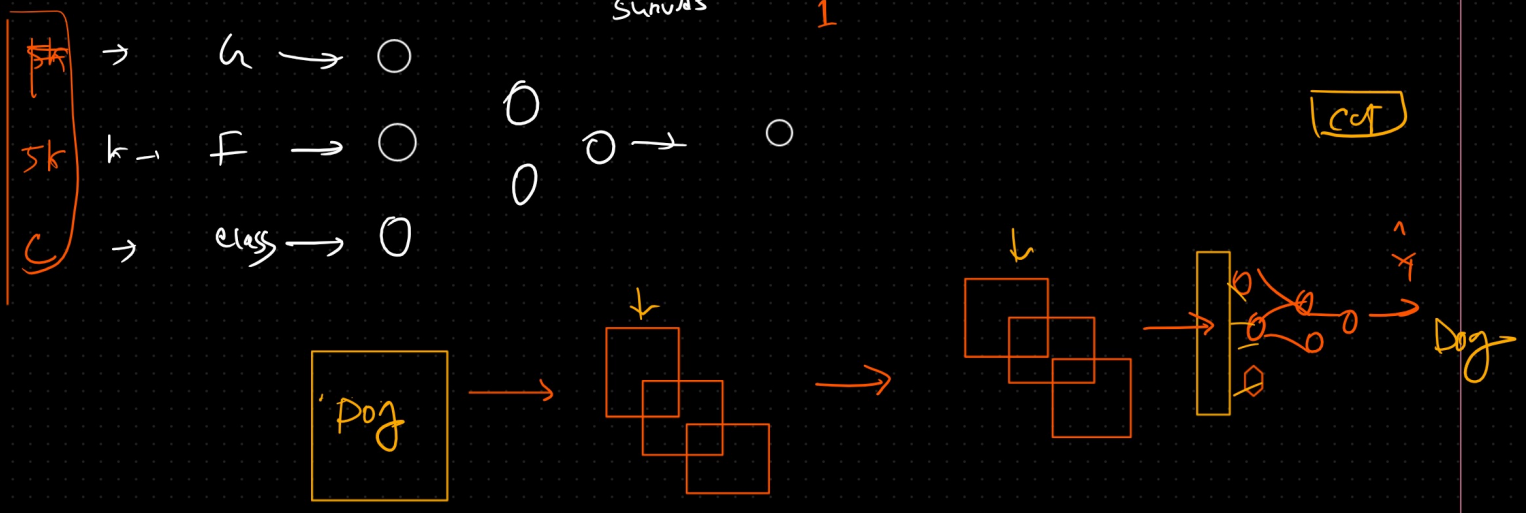
→ 1980

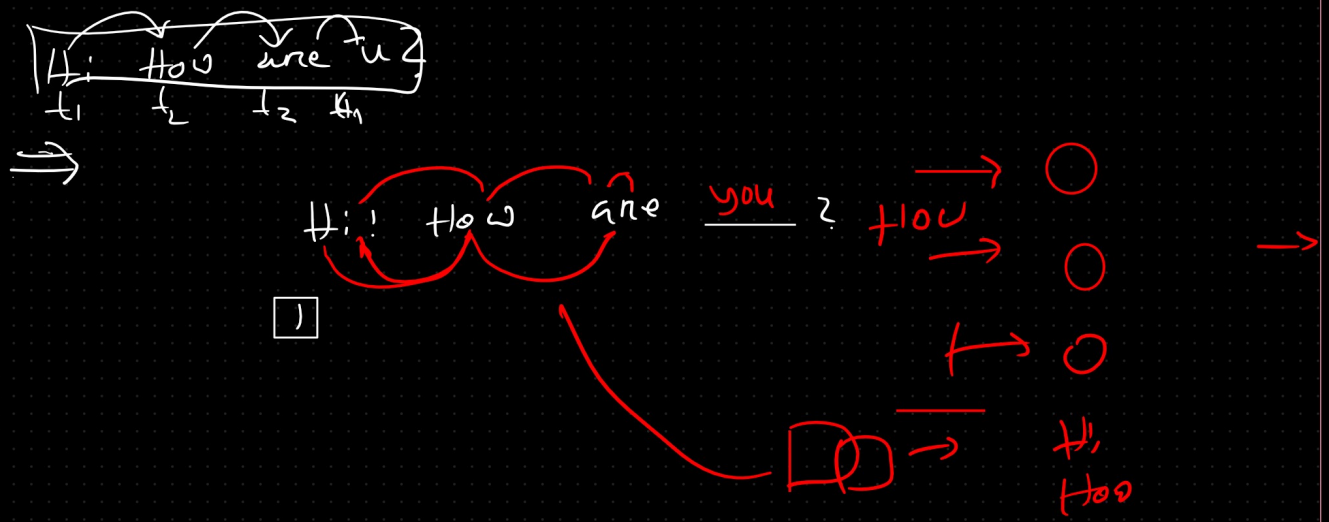
Demo:

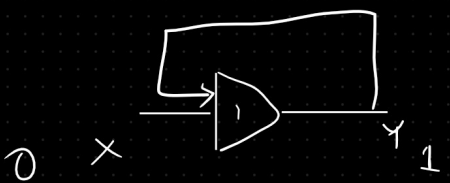
Self driven →

RL →









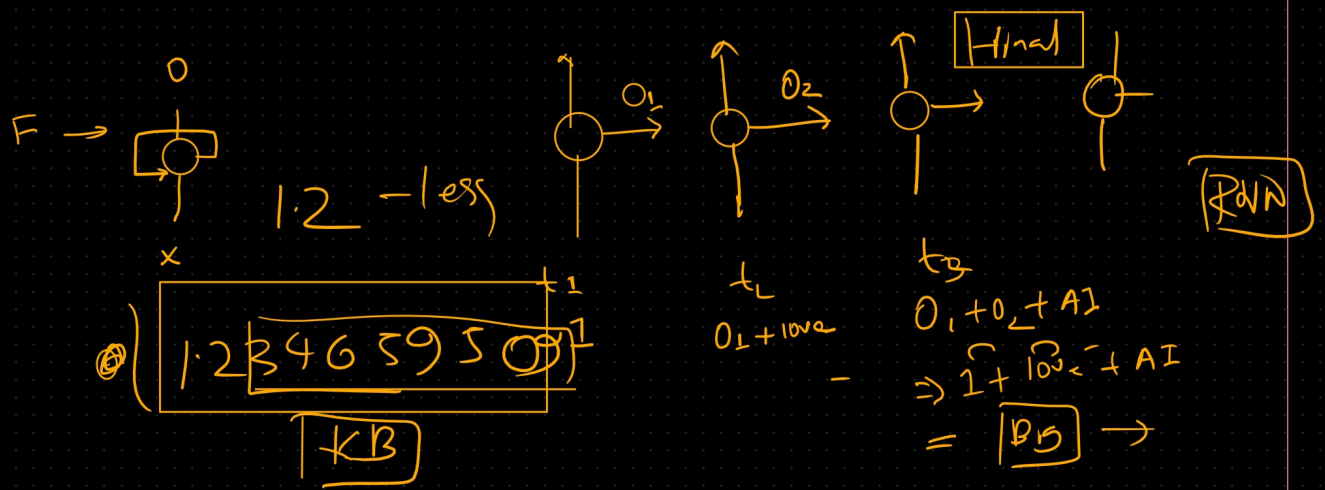
RNN → Recurrent Neural Network

Recursion

DSA

[I love AI]

(My name is Kumar)



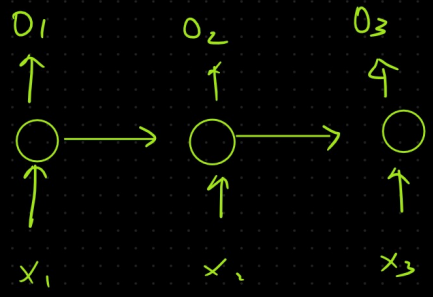
1.20345657 → Right KB

1.2 → less → KB

RNN → ① LSTM ② GRU → memory cell

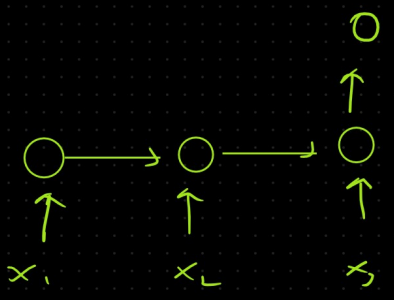
Types of RNN / Sequential Architecture:

Many to many: (seq to seq)



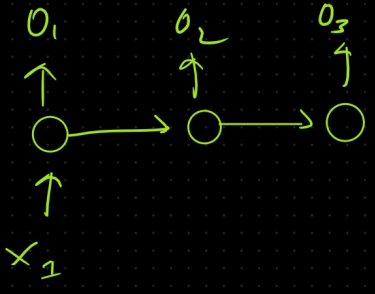
⇒ Translator

Many to one (seq to vec)



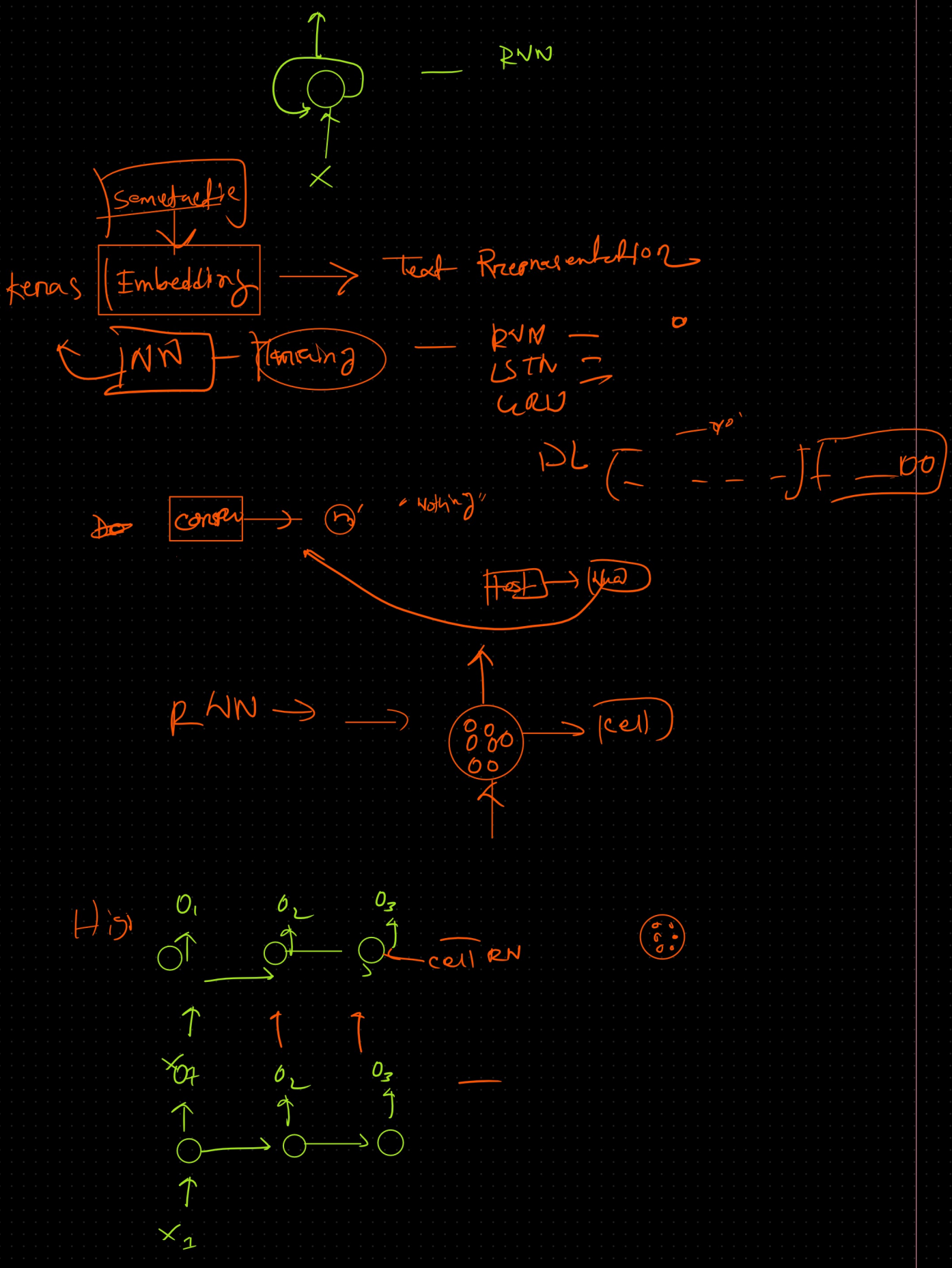
⇒ Text classification

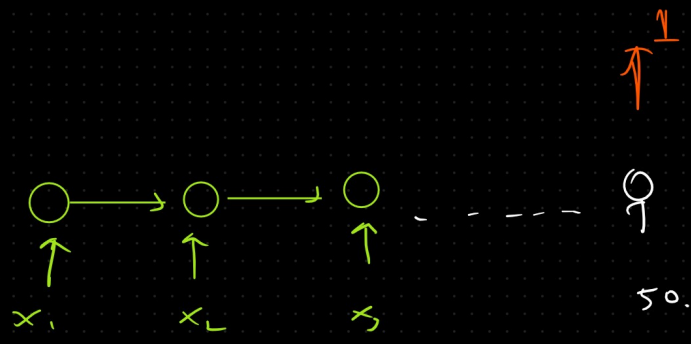
One to many (vec to seq):

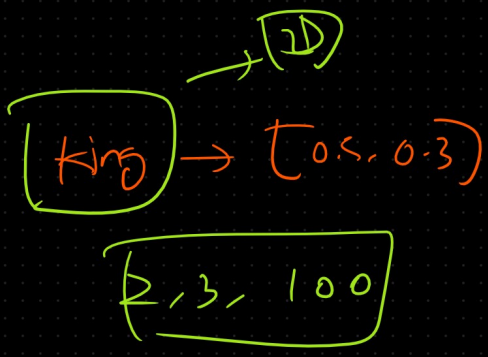


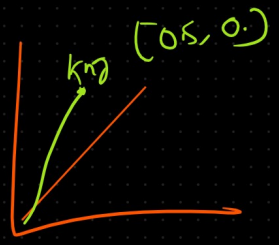
⇒ Text generation

One to one (vec to vec):




1
x_1
x_2
x_3
50.


king
[0.5, 0.3)
2, 3, 100


king
[0.5, 0.)