

# **Recurrent Neural Network**

**Dr. Muhammad Safyan**

# Recurrent Neural Network

- Sequential Data
- Why Use RNN
- Applications of RNN
- Simple RNN
- Forward Propagation
- Types of RNN
- Back Propagation of RNN

# Sequential Data

- Sequential Data →
  - ANN, Tabular data
- CNN →
  - Grid Type Data
- RNN →
  - Sequential Data,
- Tabular data is not Positional Sensitive.
  - e.g. 

IQ	Gender	Marks
----	--------	-------
- **Sequential data is Positional Sensitive**
  - e.g. Hi, My Name is Safyan
- Time Series data( Stock Exchange)
  - Speech
  - DNA Sequence
- **RNN is special kind of data that is sequence specialist**

# RNN

- RNN is favorite of NLP Domain
  - Initially NLP uses ML, ANN
  - CNN → Image

# Why Use RNN

- e.g. we want classification

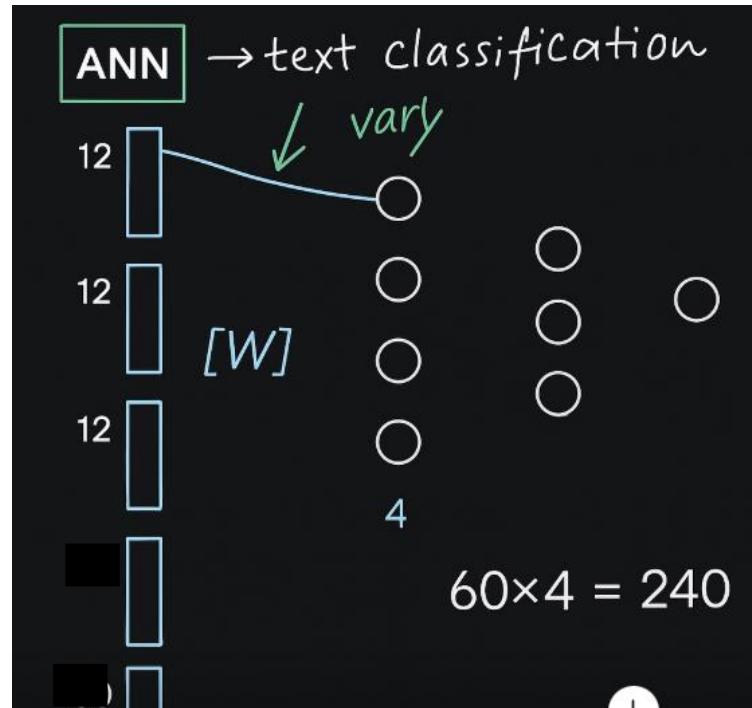
## Sentiment Classification Example

Input Text	Output
Hi my name is Safyan	0
I love GCU	0
Pakistan won the match	1

- vectorize it(one-hot-encoding, Tf-idf, BoW, W2V)  
change each word in to vector :ONE
- Hi[1,0,0,.....]
- My[0,1,0,0 .....

# Why Use RNN

- Variable Length input(Textual Data).
  - Take longest sentence as input length.
- Zero Padding.
  - What if the size is 10 k
  - Max length is 100 words(Zero Padding)
  - Sparse ( $10 \times 100$ ),
  - Imagine weight Matrix.
  - Too Much Weight to Process  
Unnecessary computation
- Prediction Problem  
take features independent of order
- Disregarding of Sequential Information
  - All word of Sentence go alongside in ANN.
  - Loose Semantic meaning.
- Need a New Architecture



# Application of RNN

- Sentiment Analysis – Free Online Demo
- <https://sentivisor.com/sentiment-analysis-free-online-demo/>
- Sentence Completion
  - Email/chrome Typing

Draft saved

Recipients

Subject

Hi, how| are you?

# Image Caption

**Generating image caption demo**

Input image (can drag-drop image file):

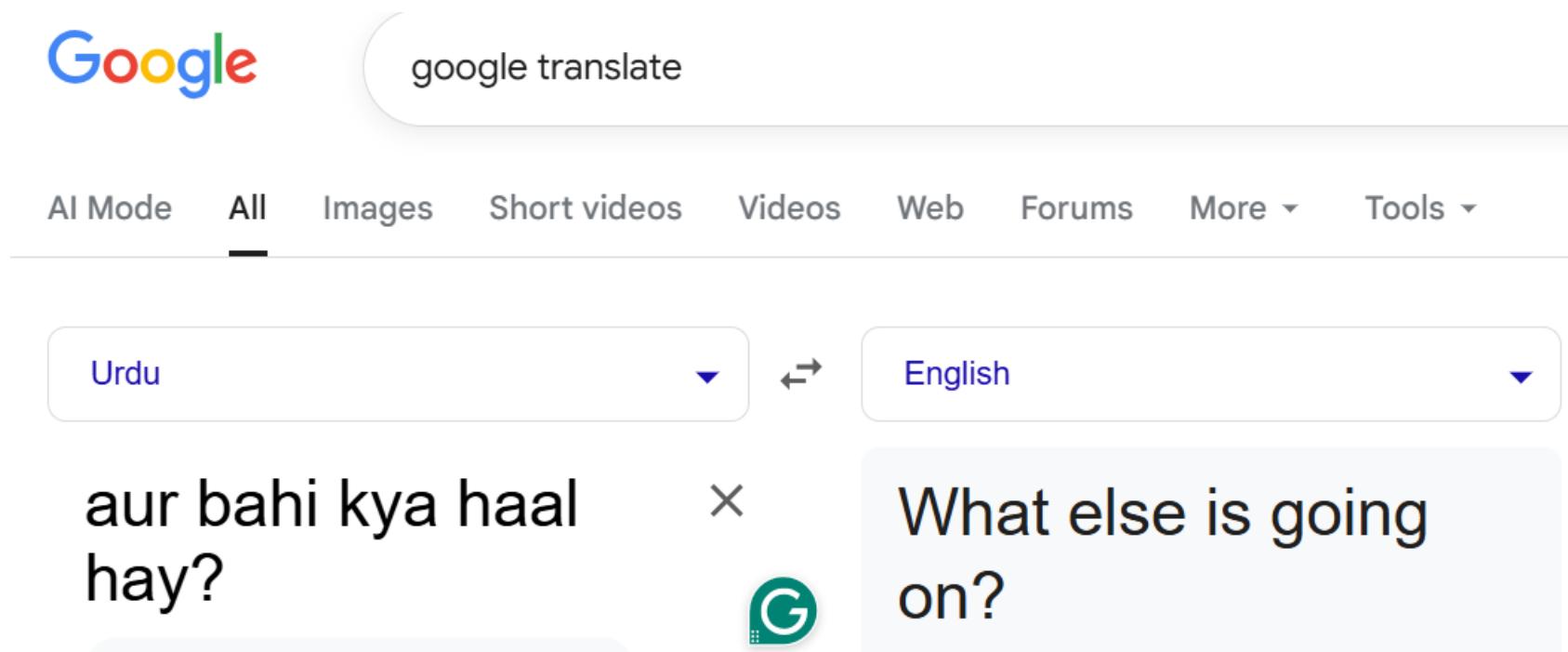
Choose File: mirzapur-7591.jpg  
Generate caption  
Load model > Analyze image > Choose best

- a group of people standing around a table
- a group of people sitting around a table
- a group of people sitting at a table
- a group of people standing next to each other
- a group of people standing in a kitchen
- a group of people standing next to each other in a room
- a group of people standing next to each other in a kitchen
- a group of people sitting at a table in a restaurant
- a group of people standing next to each other at a table
- a group of people standing next to each other in front of a building

Application: for Blind person

# Language Translation

- Google Translation



- Email Translation

# Question Answering

- **BERT Answer Questions?**
- <https://visbert.demo.datexis.com/>

## 💡 How Does BERT Answer Questions?

### What is this about?

Watch how BERT (fine-tuned on QA tasks) transforms tokens to get to the right answers. This demo shows how the token representations change throughout the layers of BERT. We observed that the transformations mostly pass four phases related to traditional Question Answering pipelines.

The tool demonstrates the findings from our paper: [Betty van Aken, Benjamin Winter, Alexander Löser and Felix Gers. How Does BERT Answer Questions? A Layer-Wise Analysis of Transformer Representations. CIKM 2019.](#)

### The 4 phases of BERT's transformations

1. **Topical / Word Clusters** Equal words and topics are clustered without current concern consideration.
2. **Connect Entities with Mentions and Attributes** Tokens are clustered based on their relation in the context.
3. **Match Question with Supporting Facts** Relevant context parts can be found close to question tokens.
4. **Answer Extraction** The answer tokens are separated from the rest. Semantic clusters are dissolved.

SQuAD [1]

HotpotQA [2]

bAbI QA [3]

Testset ID: 5733cf61d058e614000b62e9 >  Enter own example

Question: When was the French and Indian War?

Ground Truth Answer: 1754–1763

Predicted Answer: 1754–1763

Context: Seven Years' War. The war was fought between the colonies of British America and New France, with both sides supported by military units from their parent countries of Great Britain and France, as well as Native American allies. At the start of the war, the French North American colonies had a population of roughly 60,000 European settlers, compared with 2 million in the British North American colonies. The outnumbered French particularly depended on the Indians. Long in conflict, the metropole nations declared war on each other in 1756, escalating the war from a regional affair into an intercontinental conflict.

Predict & Visualize

# Simple RNN Architecture

- Sequence data can be of any length
  - Movie Reviews,
- Sequence Contains Some Meanings
  - ANN loose meaning and unable to grasp the context.
  - RNN have the capability of hold some memory

# RNN Data

- (Timsteps, Input Features)
- Vocabulary size=5
- Movie=[1,0,0,0,0]
- Was=[0,1,0,0,0]
- -----
- -----

Review	Sentiment
movie was good	1
movie bad was	0
movie was not good	0

review  
[1 0000], [010000], [0110000]

# RNN

- Send the word in RNN one by one
- T=1, T=2, T=3
- First Review is of (3 \* 5)
  - Where
    - 3 is the # of time steps
    - 5 is the # of Features
- 2<sup>nd</sup> Review (3\*5)
- 3<sup>rd</sup> Review (4\*5)
- Keras: batch size, time step, input features
  - (3,4,5) → 3 D Tensor → raw
  - 3 sentences, 4 words per sentence, 5 features per word

review  
[1 0000], [010000], [011000 0])

# Data Layout

$x_1$	$\underline{\underline{x_{11}}}$	$\overbrace{x_{12}}$	$\overbrace{x_{13}}$
	movie	was	good
$x_3$	$\underline{\underline{x_{21}}}$	$x_{22}$	$x_{33}$
	movie	was	bad
$x_3$	$x_{31}$	$x_{32}$	$x_{33}$
	movie	was	not good

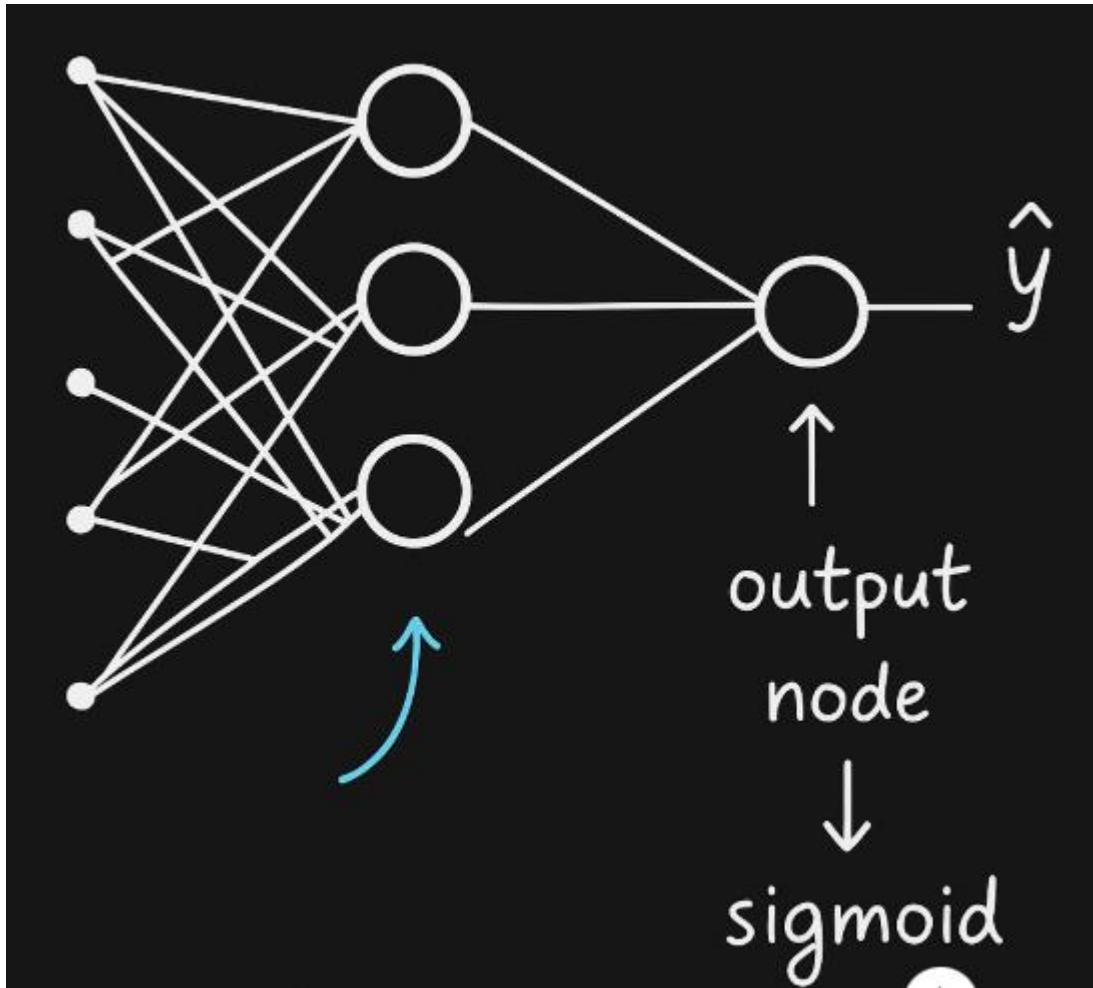
# Difference in Structure of RNN& ANN

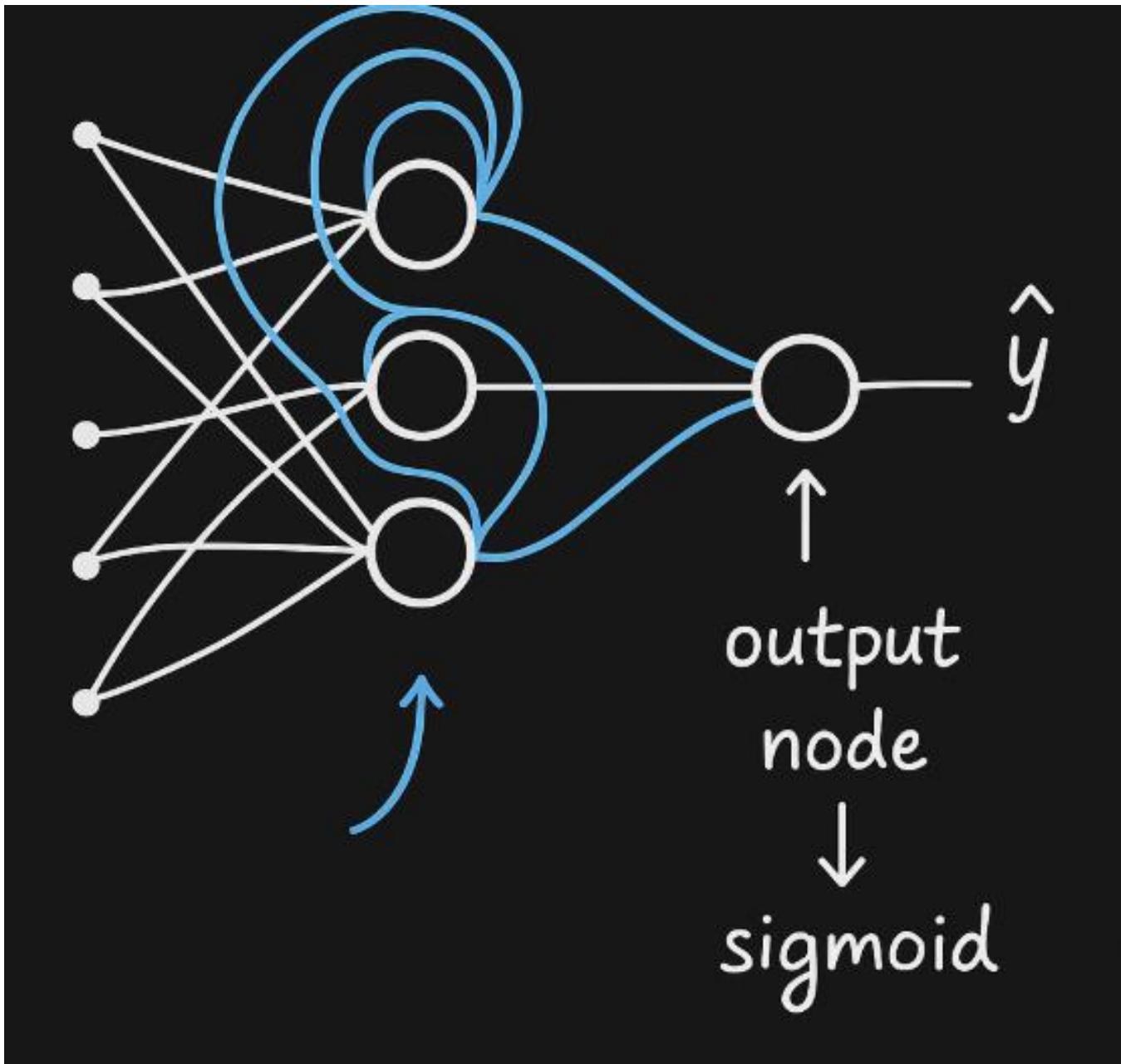
- ANN
  - Input layer → Hidden layers → Output Layer
- RNN
  - Input layer → Hidden layers → Output Layer
  - **1<sup>st</sup> Difference:**  
Pick your first review and pass on time basis  
 $t=1 \rightarrow x_1$   
 $t=2 \rightarrow x_2$   
 $t=3 \rightarrow x_3$
- **2<sup>nd</sup> Difference**
  - ANN is feed forward NN

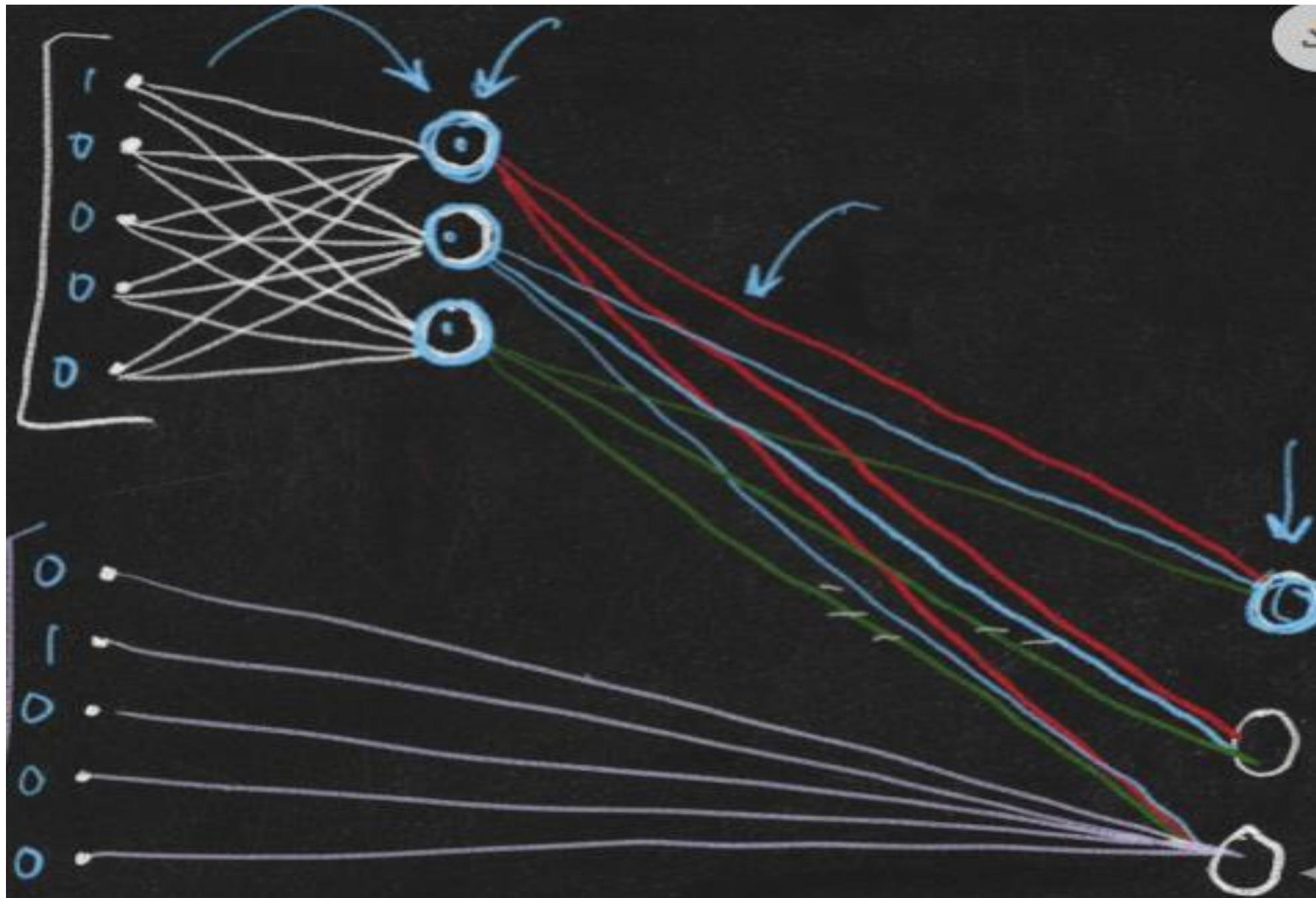
# Difference in Structure of RNN& ANN

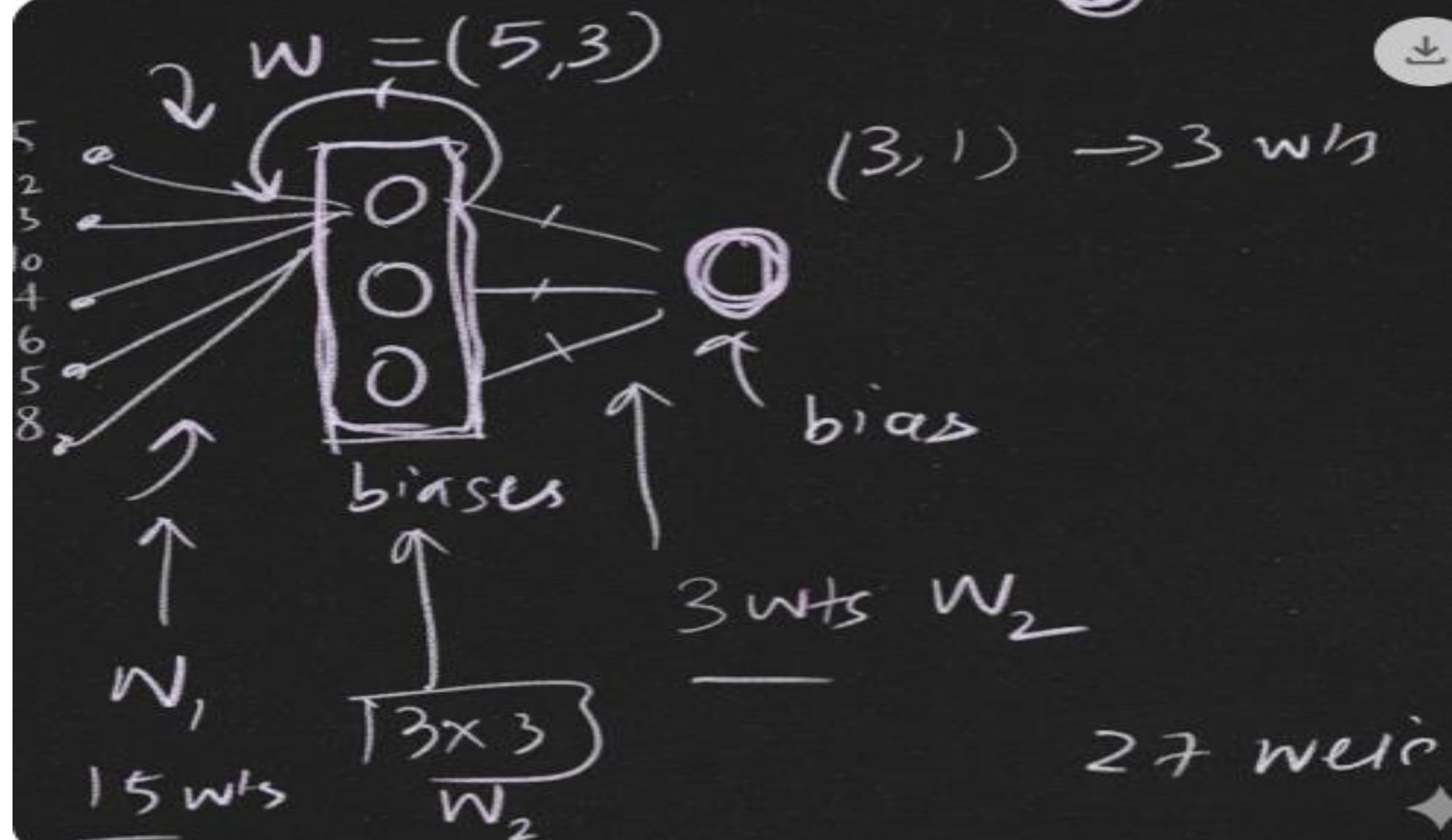
- RNN is not Feed forward
  - There is a concept of State
  - Hidden layer return back to its self that create the big difference











# RRN-Forward Propagation

$x_{11}(5,1)$

$W_i(5,3)$

At t=1

$F(x_{11}, w_1) = (1,3) \rightarrow O_1$

At t=2

$W_h = (3,3)$

$F((x_{12}, w_1) + (O_1, w_h))$

$x_{12} = (1,5)$

$W_i(5,3)$

Res=1,3

$O_1(1,3)$

$W_h = (3,3)$

res2=1,3

$O_2 = (1,3) + (1,3)$

At t=3

$x_{13}$

$O_3 = F((x_{13}, w_i) + (O_2, w_h))$

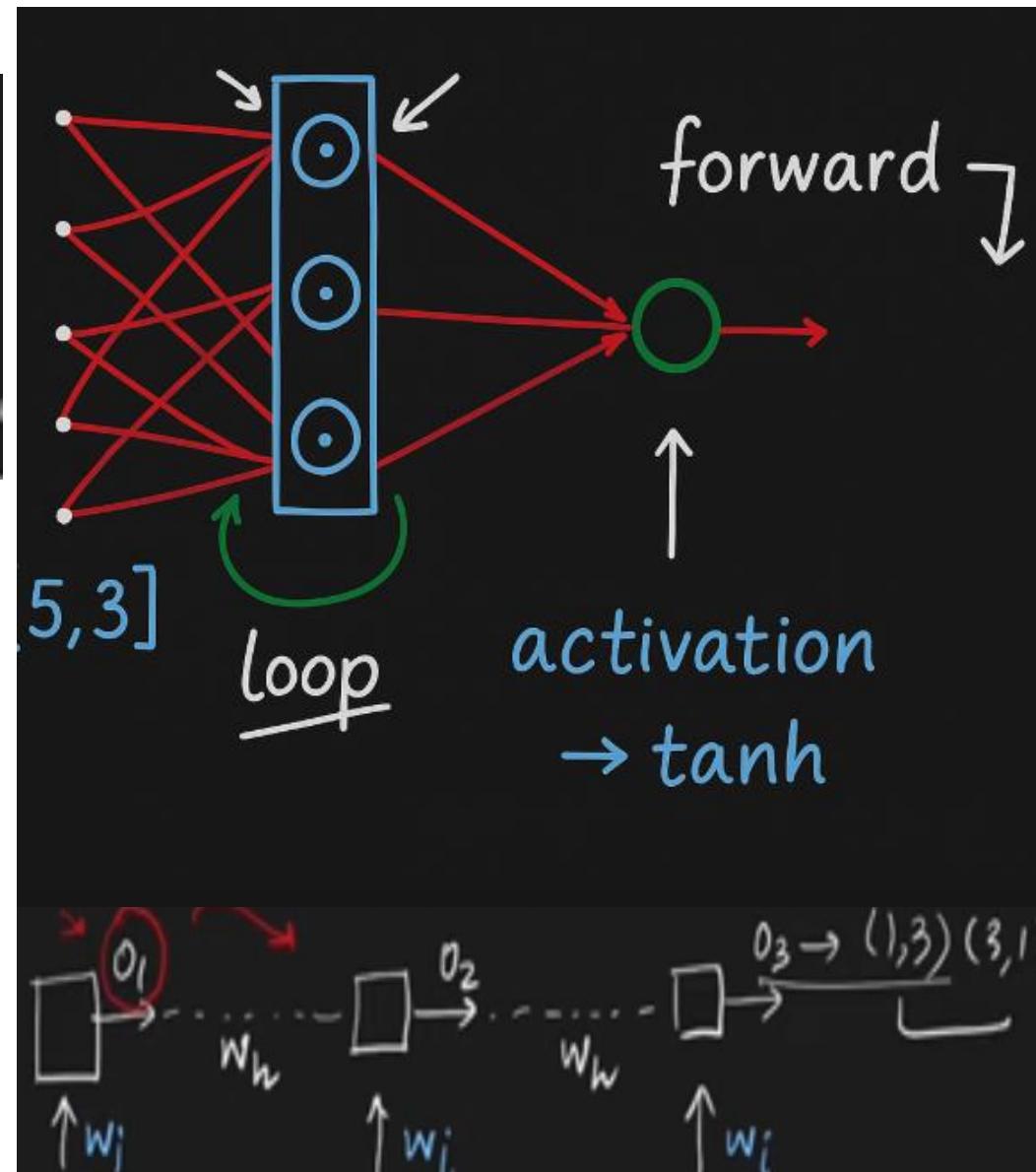
$O_3 = (1,3)$

Now

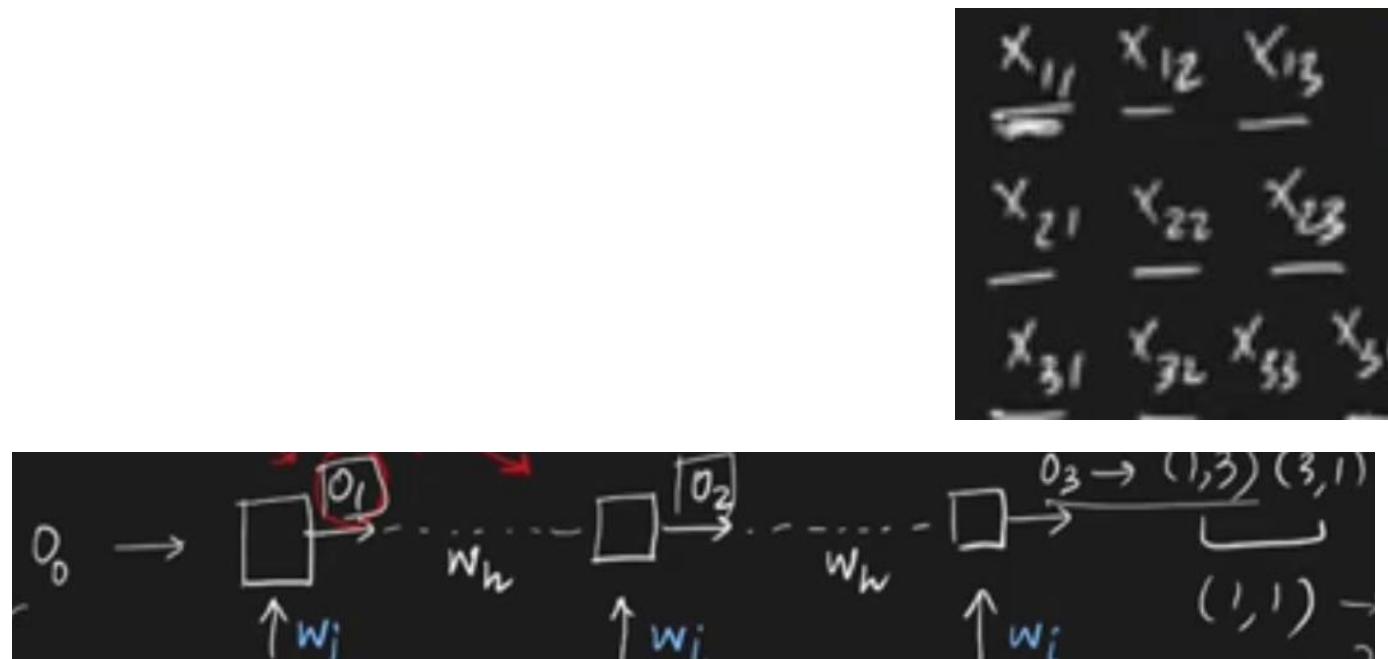
$W_o = (3,1)$

Final Output=Sigmoid(1,1)

$$\begin{array}{c} x_{11} \quad x_{12} \quad x_{13} \\ \hline \underline{x_{11}} \quad \underline{x_{12}} \quad \underline{x_{13}} \\ x_{21} \quad x_{22} \quad x_{23} \\ \hline \underline{x_{21}} \quad \underline{x_{22}} \quad \underline{x_{23}} \\ x_{31} \quad x_{32} \quad x_{33} \quad x_{34} \\ \hline \underline{x_{31}} \quad \underline{x_{32}} \quad \underline{x_{33}} \quad \underline{x_{34}} \end{array}$$



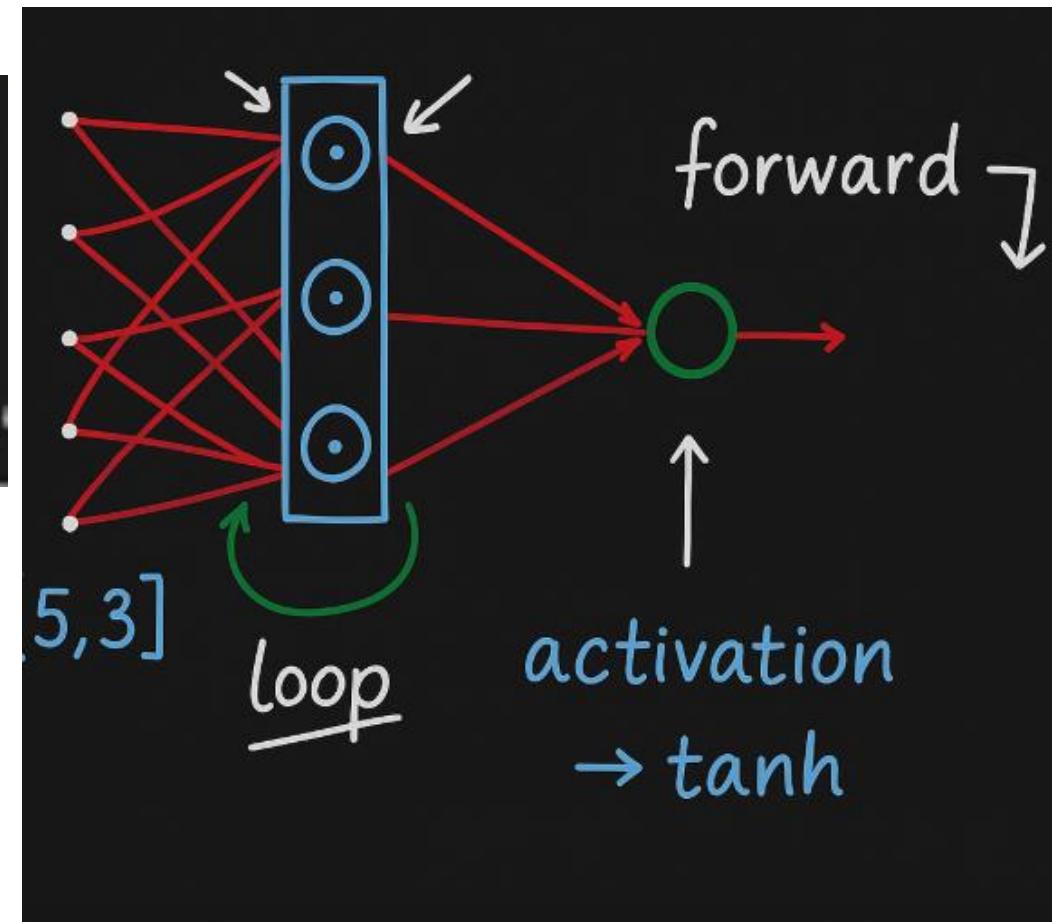
# RRN-Forward Propagation



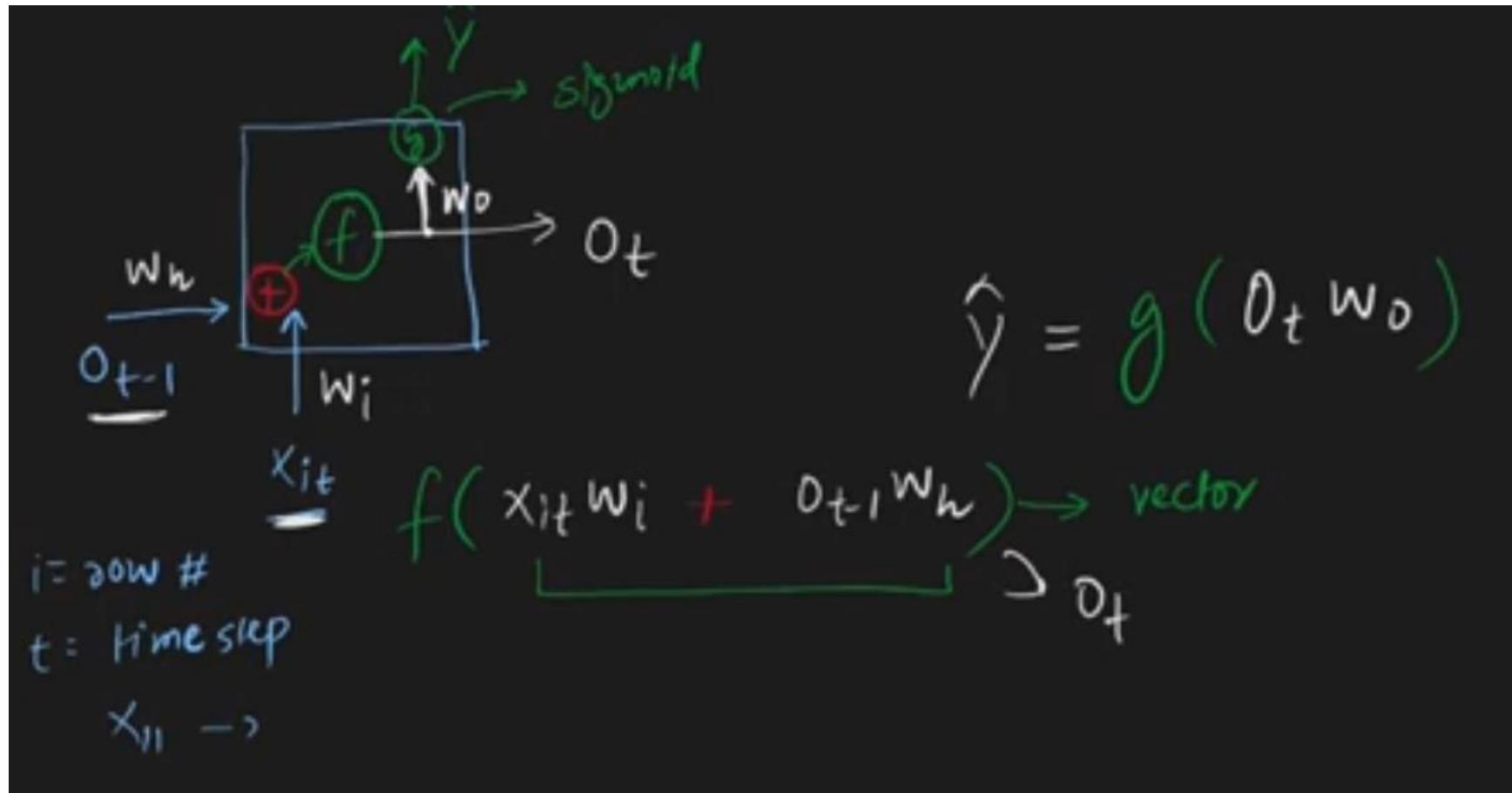
$$O_0 = [0, 0, 0]$$

Final Formula =  $F(x_{11} * w_i + O_0 * w_h)$

- Only One layer is used again and again → Recurrent NN.
- Almost reminder last ten steps
- Very good on sequence data



# Simplified Representation

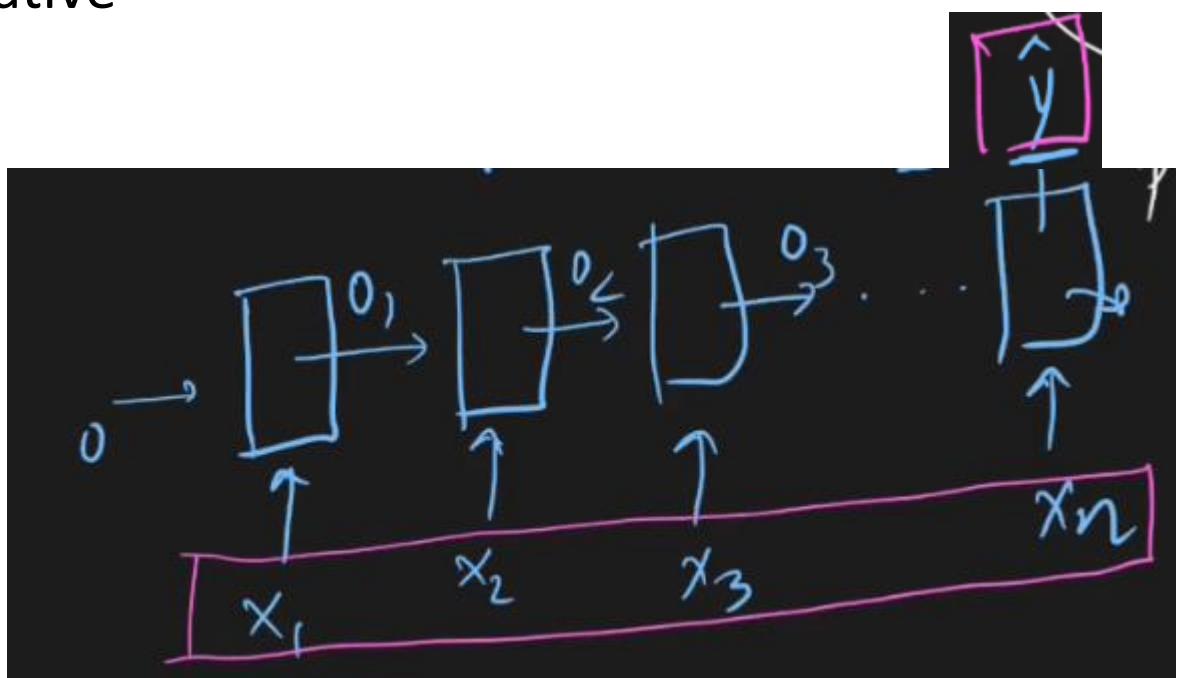
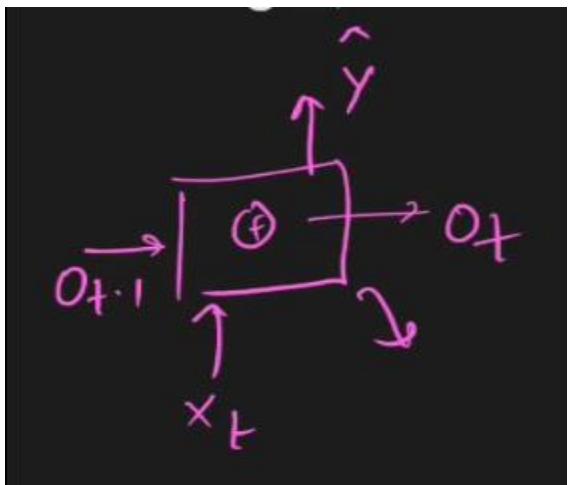


# Types of RNNs

- Back propagation varies on different types of RNN.
- 4 Types of RNN
  - Many to one
  - One to many
  - Many to many
  - One to one

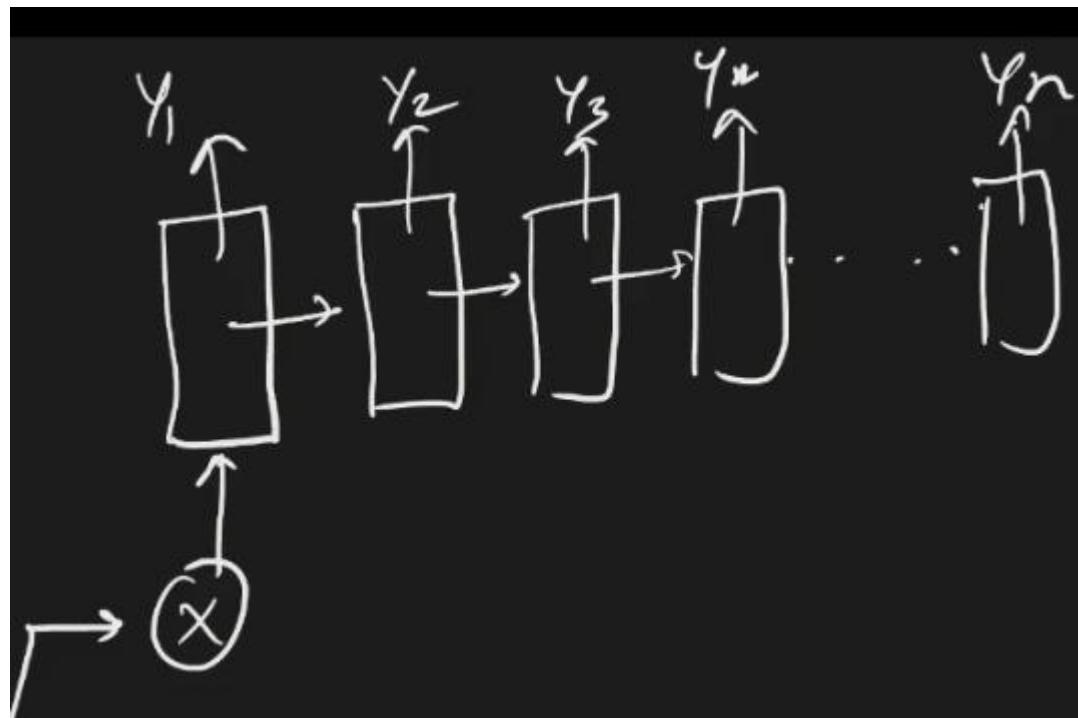
# Many to one

- Input =sequence
- Output= Non sequence(integer, scalar)  
e.g.
  - Sentiment Analysis → Positive/Negative
  - Rating Prediction-> \*\*\*



# One-to-Many

- Input=non sequential(number , image)  
output= Sequential



# One-to-Many

- Input=non sequential(number , image)  
output= Sequential
  - Image Captioning
- Input= image of a person play cricket
- Output=“Man is playing cricket”
- Another example:  
music generation

# Many-to-Many

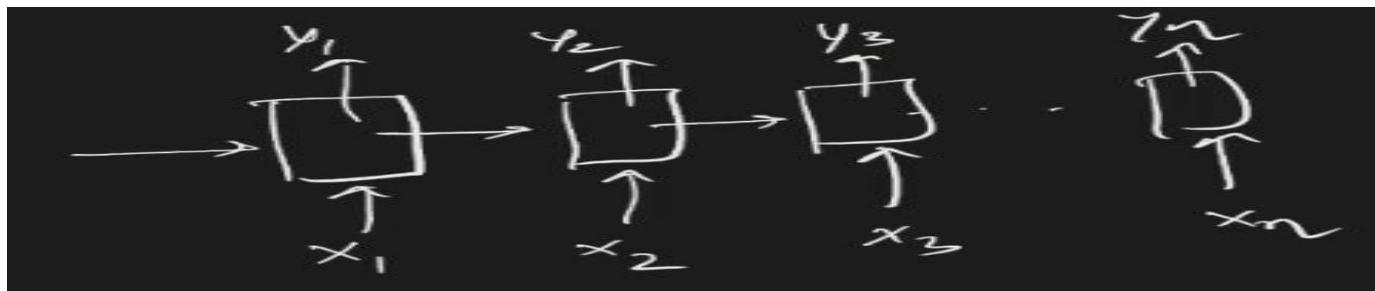
- Input =sequential data
- Output =sequential data
- Also called Seq2Seq
- Two Types:

## Same Length

- Input size=equal to output size

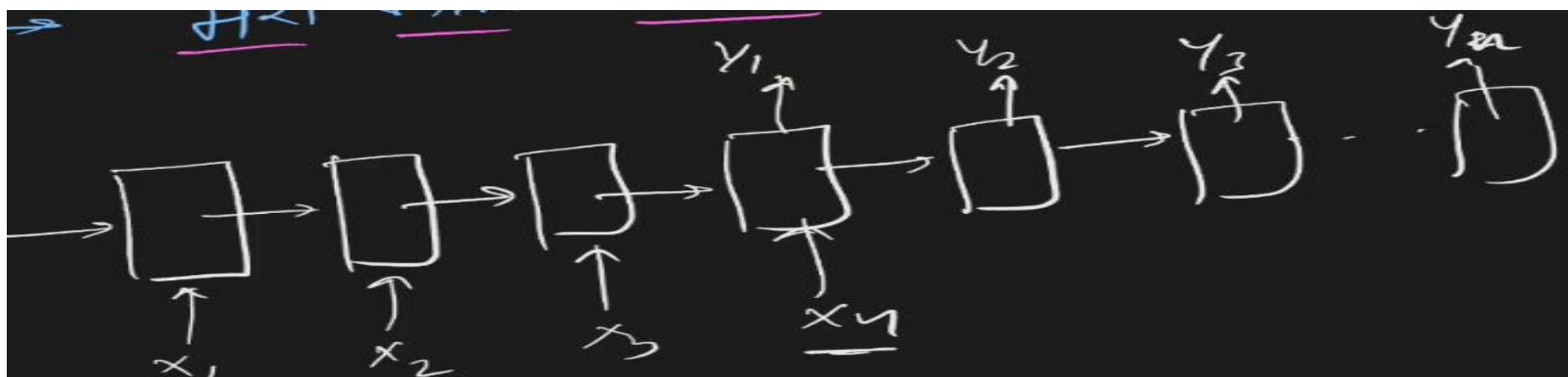
My	Name	Is	Safyan
Pronoun	Noun	Verb	Noun

- Name Entity Recognition



# Many-to-Many

- Variable Length  
input size!=output size
- Language Translation
- The boy is playing with a red ball.
- لڑکا ایک سرخ گینڈ کے ساتھ کھیل رہا ہے
- Encoder/Decoder



# One-to-One

- Input=Non Sequential
- Output=non Sequential
  - Image Classification

# RNN-Back Propagation

- Learning is attained with back propagation.
- Back Propagation Through Time(BPTT).
- Took an example of Many-to-One.
- Text---→ 1/0

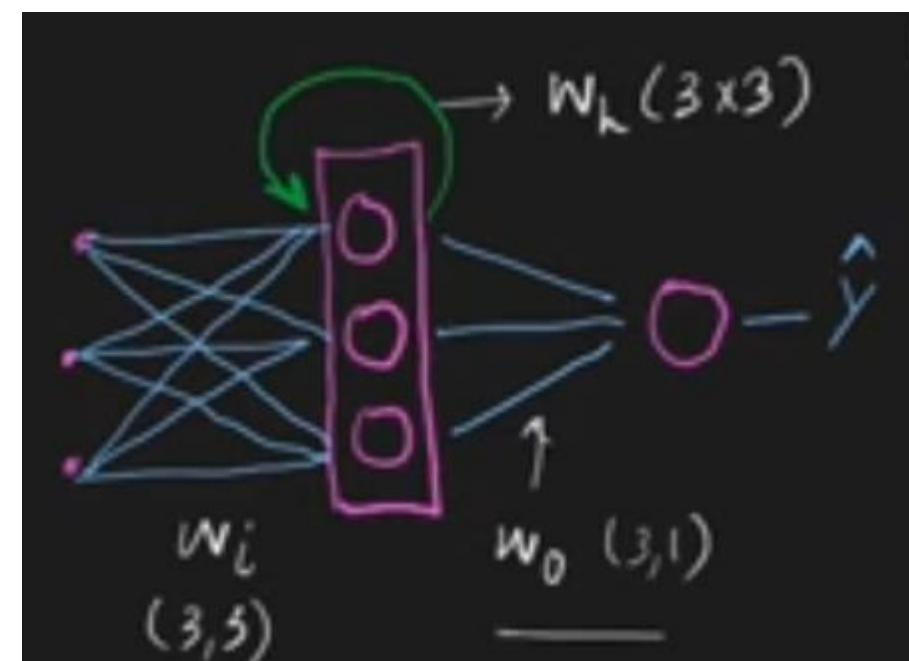
F1	F2	F3	Output
Cat	mat	rat	1
rat	rat	mat	1
mat	mat	cat	0

- First step: Convert text into embedding
  - Total words=3

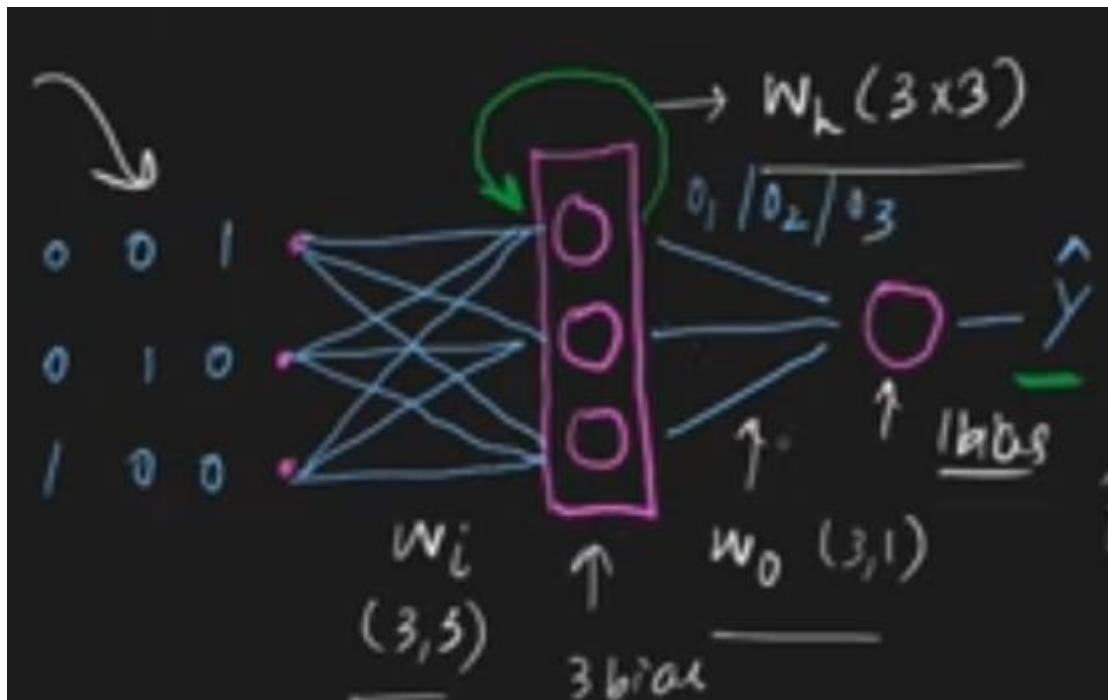
# RNN-Back Propagation

- Cat=[1,0,0], Mat=[0,1,0] Rat=[0,0,1]

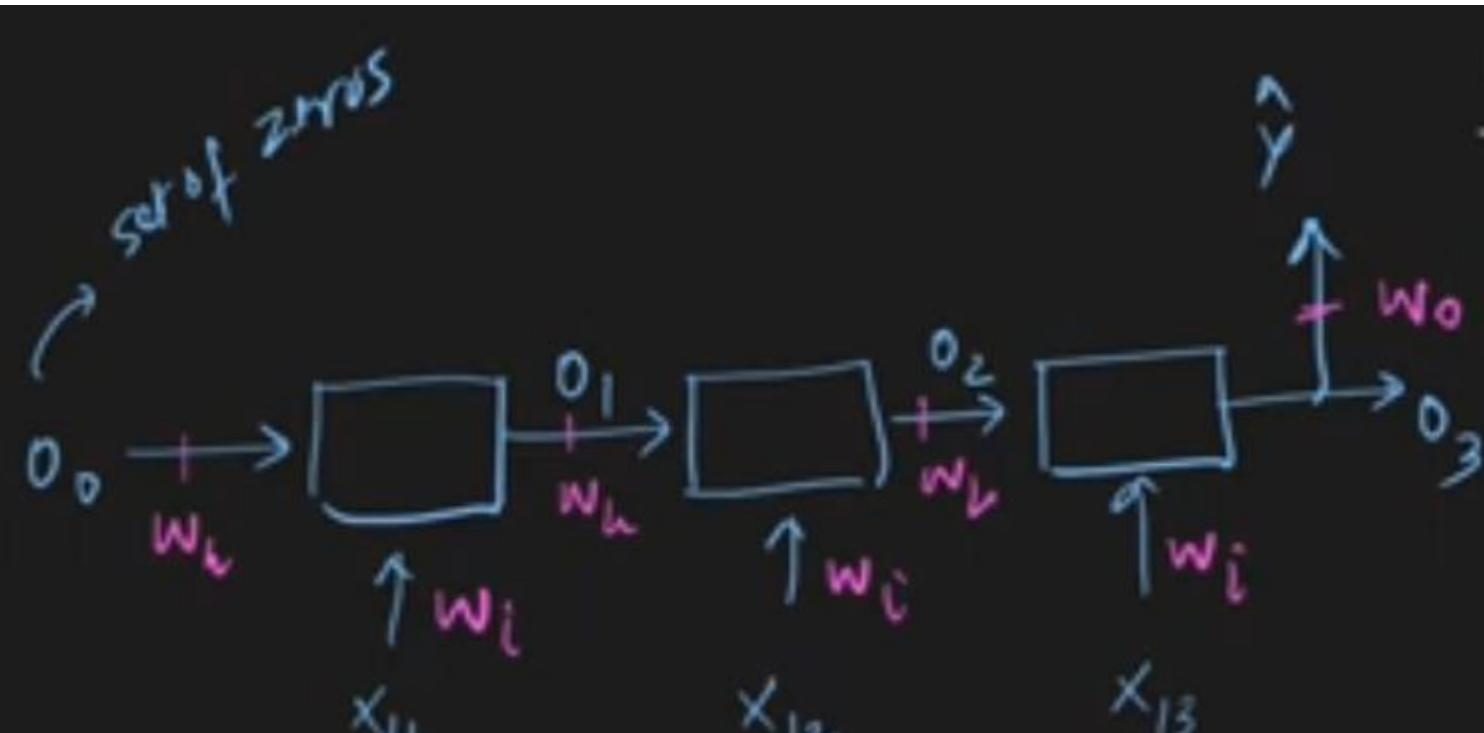
$$\begin{array}{l} x_1 \quad [1 \ 0 \ 0] \quad [0 \ 1 \ 0] \quad [0 \ 0 \ 1] \quad y \\ x_2 \quad [0 \ 0 \ 1] \quad [0 \ 0 \ 1] \quad [0 \ 1 \ 0] \quad 1 \\ x_3 \quad [0 \ 1 \ 0] \quad [0 \ 1 \ 0] \quad [1 \ 0 \ 0] \quad 0 \end{array}$$



# RNN-Back Propagation



# RNN-Back Propagation



$$\begin{aligned} y &\rightarrow \hat{y} = \sigma(o_3 w_o) \\ L &= -y_i \log \hat{y}_i - (1-y_i) \log(1-\hat{y}_i) \end{aligned}$$

Final Formula= $F(x_{11}*Wi + O0 *Wh)$

- **Objective Function:**

$\text{Loss}_{\min}$  = by adjusting/tuning weights of  $w_i, w_h, w_o$

Possible with Gradient Descent

# Gradient Descent

$$w_i = w_i - \eta \frac{\partial L}{\partial w_i}$$

$$w_h = w_h - \eta \frac{\partial L}{\partial w_h}$$

$$w_o = w_o - \eta \frac{\partial L}{\partial w_o}$$

$$O_1 = f(x_{i1}w_i + O_0w_h)$$

$$O_2 = f(x_{i2}w_i + O_1w_h)$$

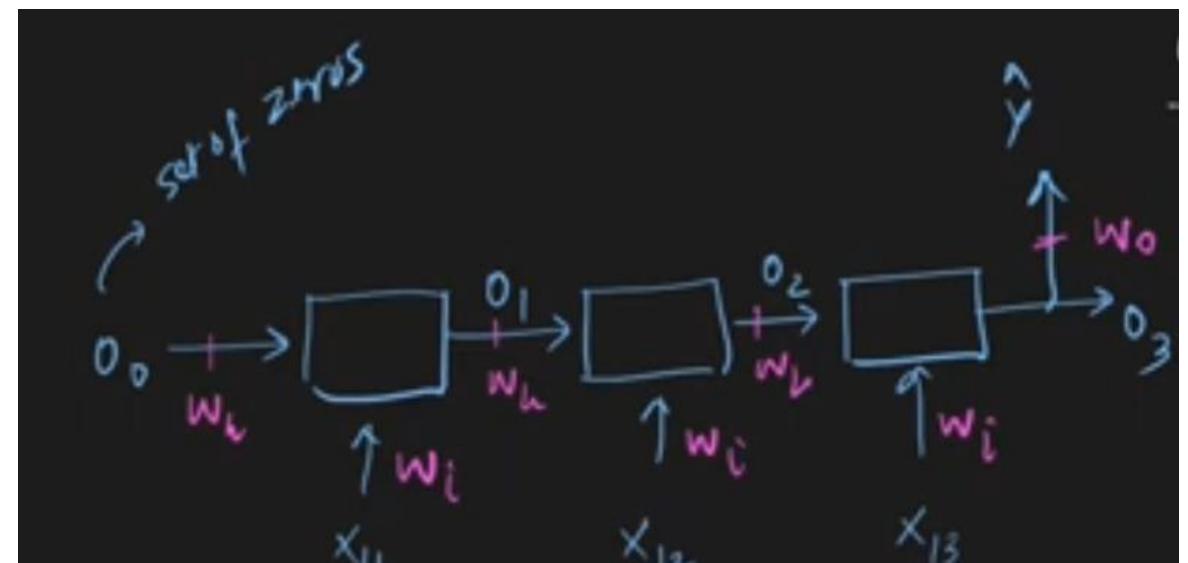
$$O_3 = f(x_{i3}w_i + O_2w_h)$$

$$\hat{y} = \sigma(O_3w_o)$$

- $O_t$ : The **output** or hidden state at time step  $t$ .
- $f$ : An **activation function** (e.g., tanh, ReLU).
- $x_{it}$ : The **input** at time step  $t$ .
- $w_i, w_h, w_o$ : **Weight matrices** for input, hidden (recurrent), and output layers, respectively.
- $O_0$ : The initial hidden state or previous hidden state.
- $\hat{y}$ : The **final prediction** or output.
- $\sigma$ : A final activation function (e.g., **sigmoid** or **softmax**).

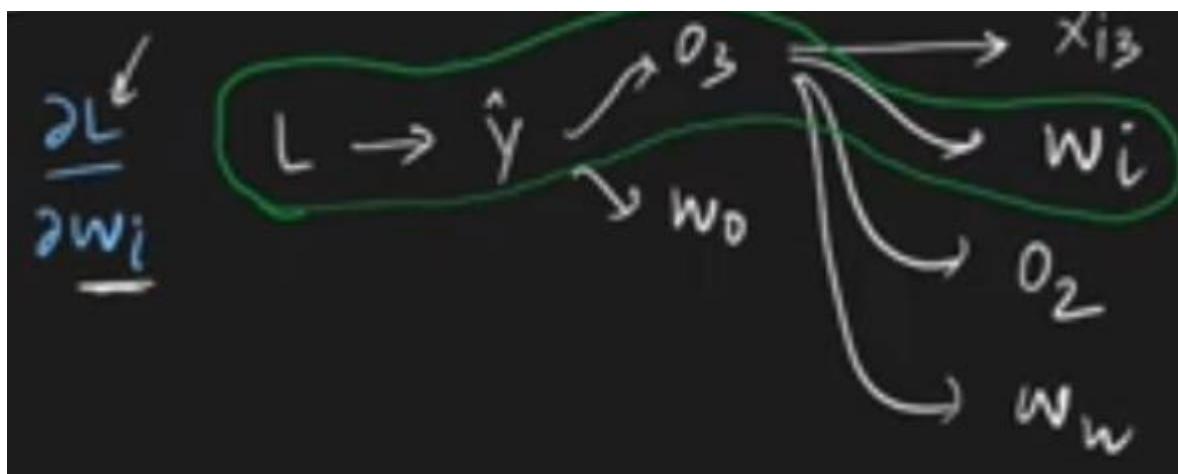
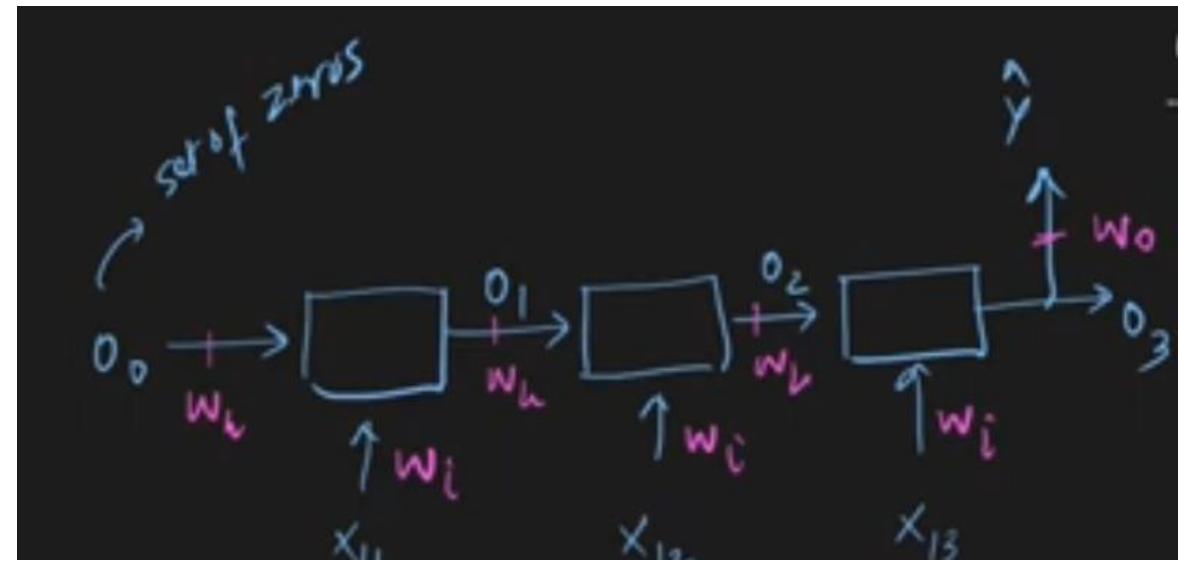
# Gradient Descent

$$\frac{\partial L}{\partial w_o} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial w_o}$$

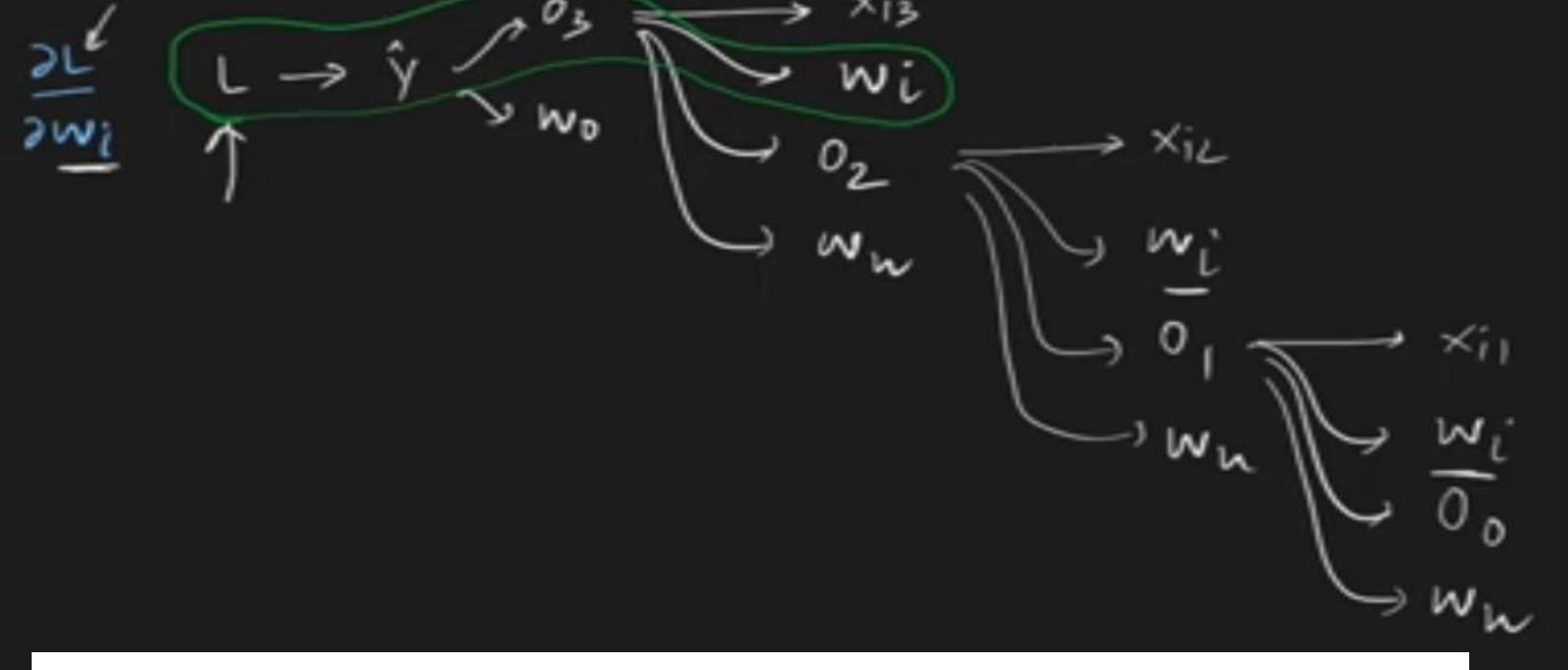


# Gradient Descent

$$\frac{\partial L}{\partial w_i} =$$



$$o_1 = f(x_{i1}w_i + o_0w_n)$$
$$o_2 = f(x_{i2}w_i + o_1w_n)$$
$$o_3 = f(x_{i3}w_i + o_2w_n)$$
$$\hat{y} = \sigma(o_3w_0)$$



The total gradient is:

$$\frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial O_3} \frac{\partial O_3}{\partial w_i} \quad (\text{Path through } O_3 \text{ only})$$

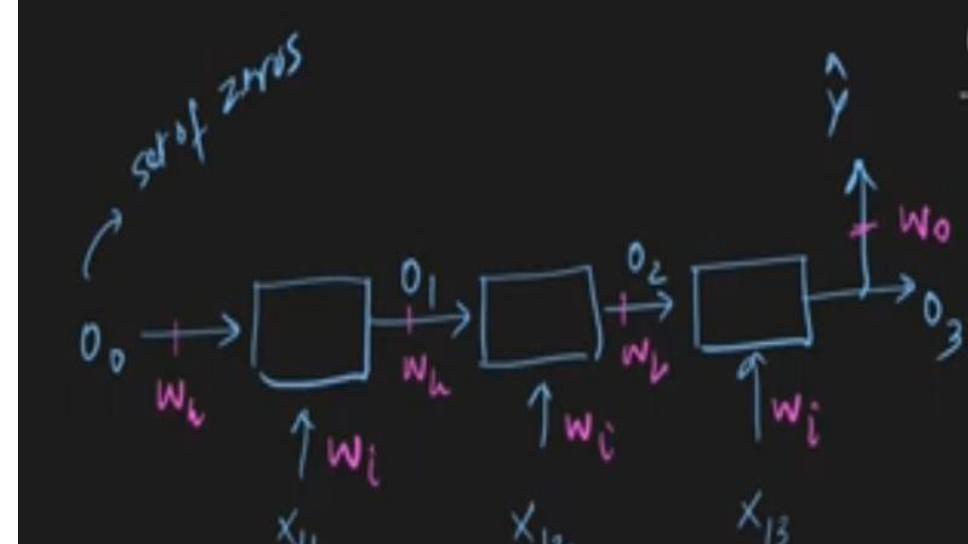
$$+ \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial O_3} \frac{\partial O_3}{\partial O_2} \frac{\partial O_2}{\partial w_i} \quad (\text{Path through } O_2 \text{ and } O_3)$$

$$+ \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial O_3} \frac{\partial O_3}{\partial O_2} \frac{\partial O_2}{\partial O_1} \frac{\partial O_1}{\partial w_i} \quad (\text{Path through } O_1, O_2, \text{ and } O_3)$$

### Mathematical Expression (Summation Form)

The expression is:

$$\frac{\partial L}{\partial w_i} = \sum_{j=1}^3 \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial O_j} \frac{\partial O_j}{\partial w_i}$$



$$O_1 = f(x_{i1}w_i + O_0w_h)$$

$$O_2 = f(x_{i2}w_i + O_1w_h)$$

$$O_3 = f(x_{i3}w_i + O_2w_h)$$

$$\hat{y} = \sigma(O_3w_o)$$

### Mathematical Expression (Generalized BPTT)

The expression is:

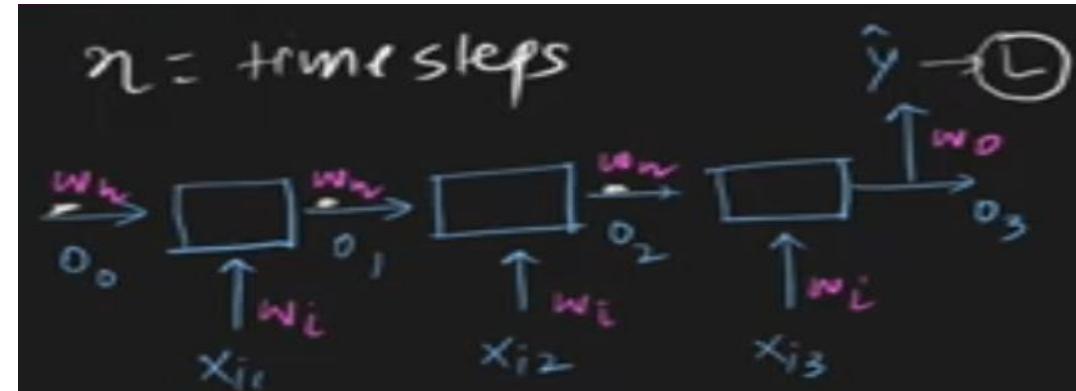
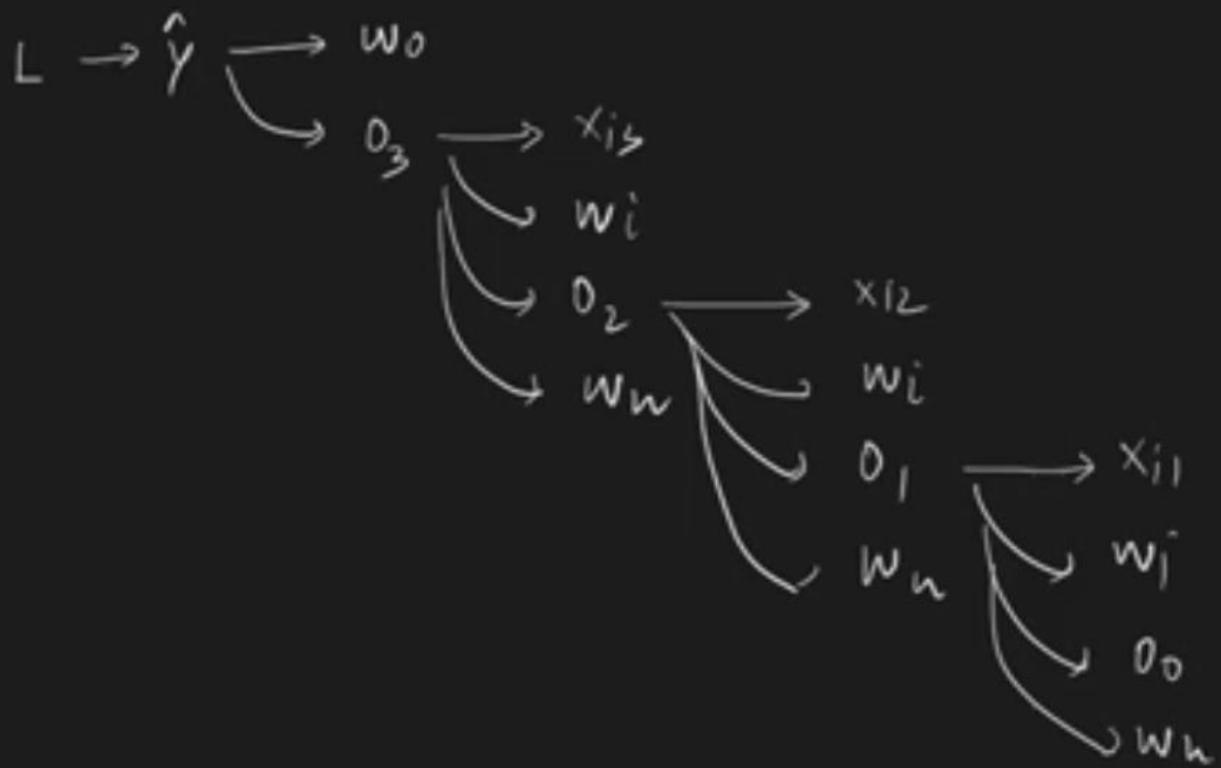
$$\frac{\partial L}{\partial w_i} = \sum_{j=1}^n \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial O_j} \frac{\partial O_j}{\partial w_i}$$

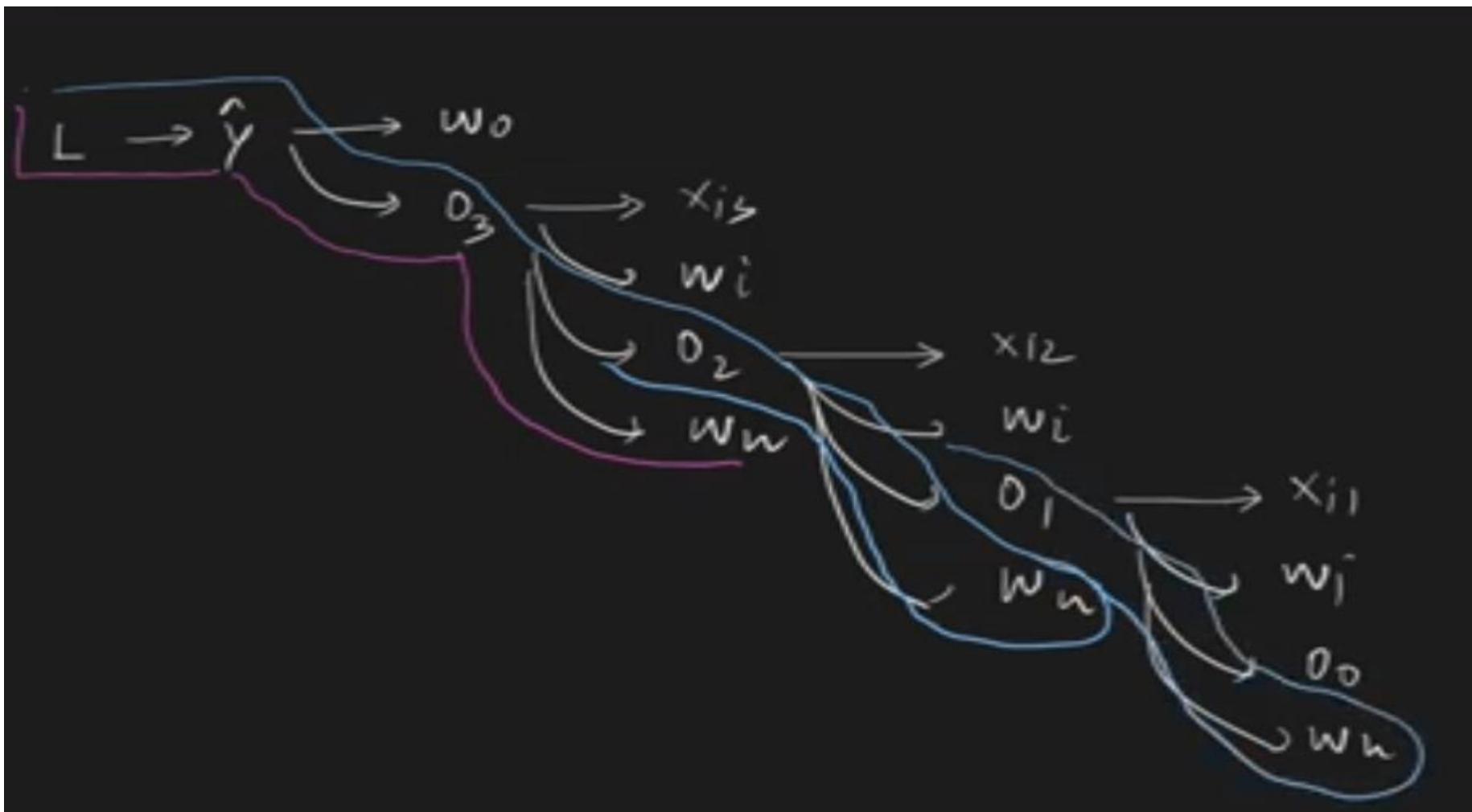
The image also provides the definition:

*n* = time steps

$$\frac{\partial L}{\partial w_h}$$

- - -





The total gradient  $\frac{\partial L}{\partial w_h}$  is given by the sum of three terms (representing three time steps,  $O_1, O_2, O_3$ ):

$$\frac{\partial L}{\partial w_h} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial O_3} \frac{\partial O_3}{\partial w_h} \quad (\text{Path through } O_3 \text{ only})$$

$$+ \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial O_3} \frac{\partial O_3}{\partial O_2} \frac{\partial O_2}{\partial w_h} \quad (\text{Path through } O_2 \text{ and } O_3)$$

$$+ \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial O_3} \frac{\partial O_3}{\partial O_2} \frac{\partial O_2}{\partial O_1} \frac{\partial O_1}{\partial w_h} \quad (\text{Path through } O_1, O_2, \text{ and } O_3)$$

$$\frac{\partial L}{\partial w_h} = \sum_{j=1}^n \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial O_j} \frac{\partial O_j}{\partial w_h}$$

