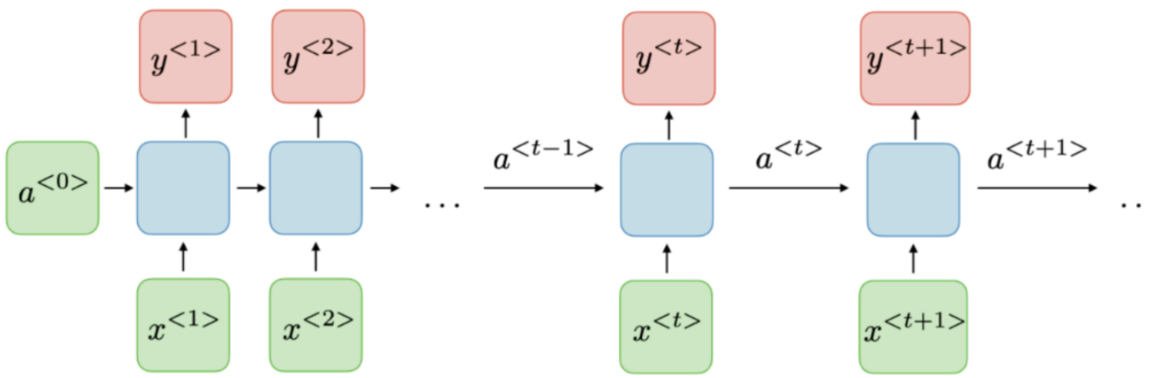
# Lecture 6: Vanilla RNN

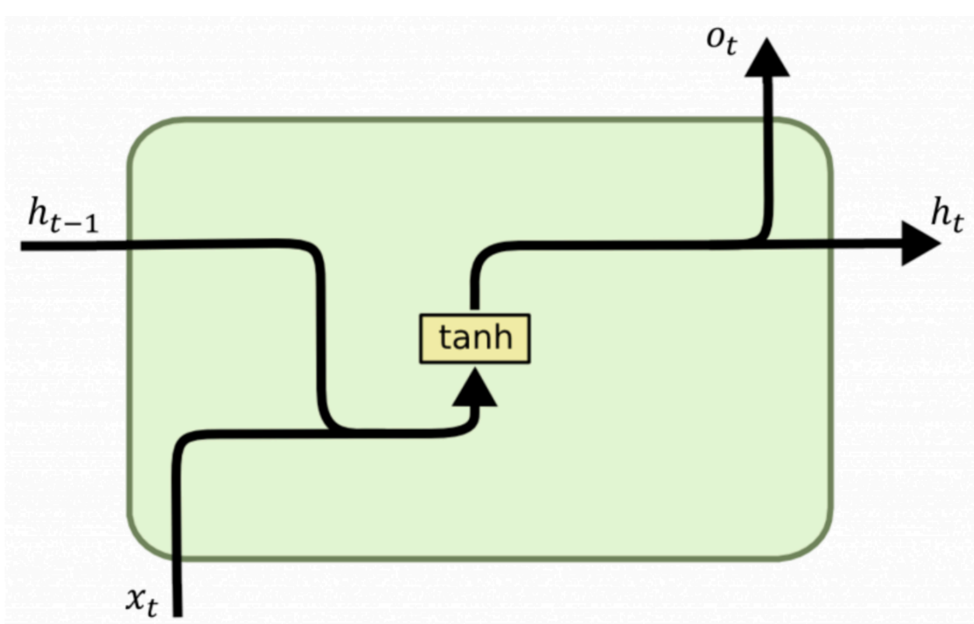
## RNN

Recurrent neural networks, also known as RNNs, are a class of neural networks that allow previous outputs to be used as inputs while having hidden states. They are typically as follows:



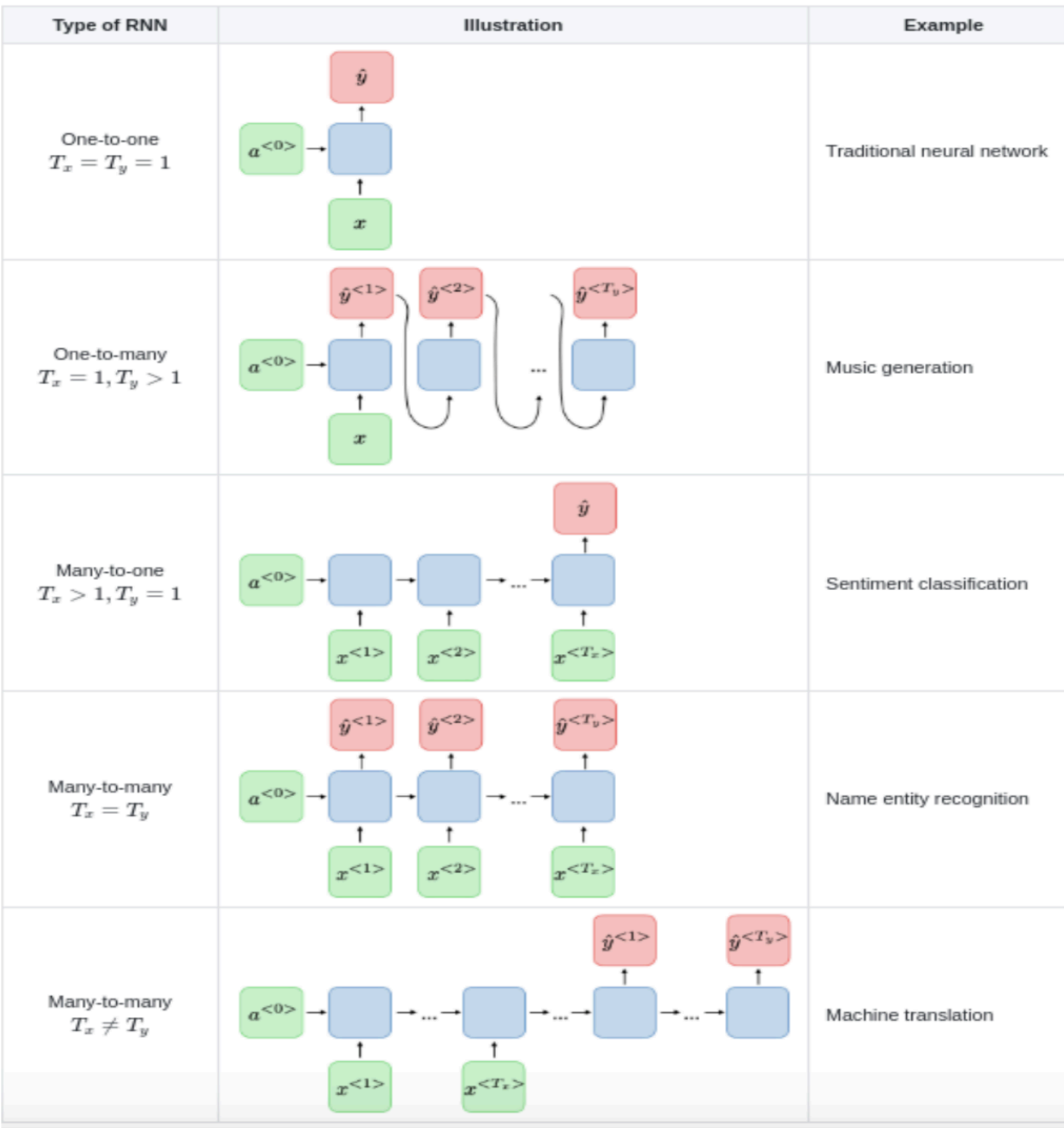

For each timestep *t* , the input x^t the activation a^t and the output y^t .

## What are the different components of the RNN unit?



- **Xt**: Xt is the input at time step t.
- **ht-1**: Hidden state from the previous time step which is a vector representation of the information from previous time steps.
- **tanh**: The input from the current time step **t** . In the case of Vanilla RNN it is the **tanh.**
- **ht**: Hidden state from the current time step **t.**
- **Ot**: Ot is the output of RNN.

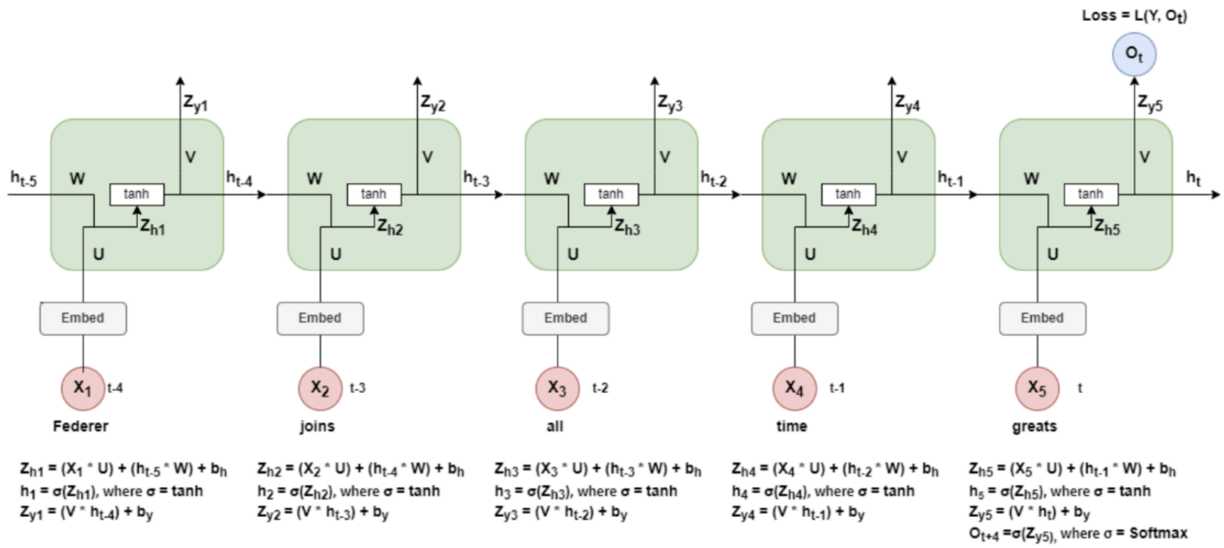
## Types of RNN



| Type of RNN | Illustration | Example |
|---|---|---|
| One-to-one $T_x = T_y = 1$ | | Traditional neural network |
| One-to-many $T_x = 1, T_y > 1$ | | Music generation |
| Many-to-one $T_x > 1, T_y = 1$ | | Sentiment classification |
| Many-to-many $T_x = T_y$ | | Name entity recognition |
| Many-to-many $T_x \neq T_y$ | | Machine translation |

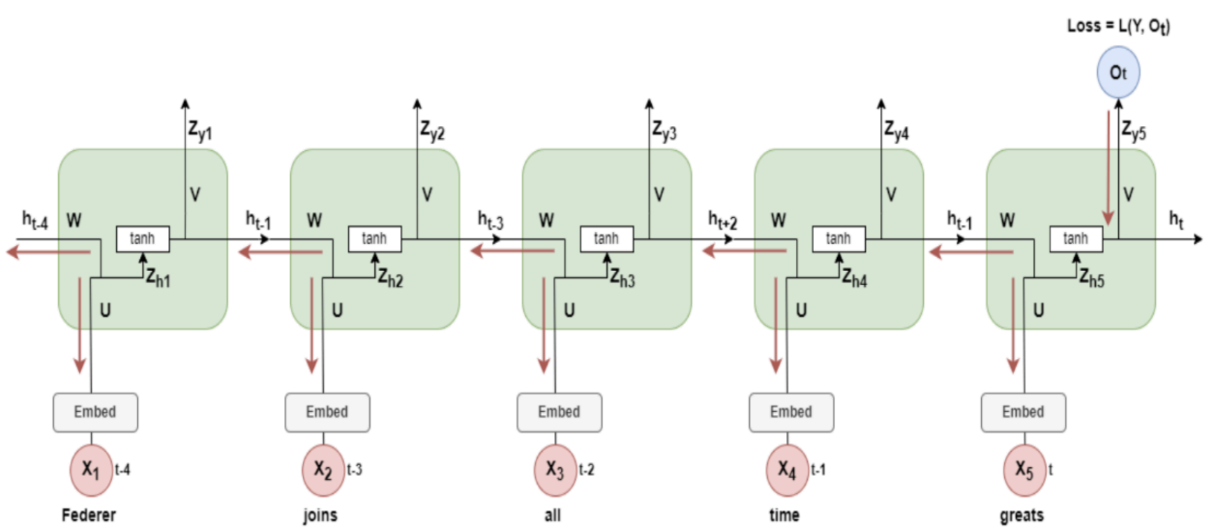
# Propagation in RNN

## Forward Propagation:




- **Whx = U**: The weight matrix at the input
- **Whh = V**: The weight matrix at the hidden state
- **Why = W**: The weight matrix at the output
- **Zyt, Zht =** Intermediate results
- **σ =** Activation function
- **Y =** Actual
- **L =** Loss function

## Back Propagation:



**Steps involved in updating parameter weights:**

- Calculate the gradients of the loss with respect to the parameters
- Multiply it with the Learning rate
- Update the new weights

## Loss function:

In the case of a recurrent neural network, the loss function ***L*** of all time steps is defined based on the loss at every time step as follows:

$$\boxed{\mathcal{L}(\hat{y}, y) = \sum_{t=1}^{T_y} \mathcal{L}(\hat{y}^{<t>}, y^{<t>})}$$

## Back propagation through time:

Backpropagation is done at each point in time. At timestep ***T*** , the derivative of the loss ***L*** with respect to weight matrix ***W*** is expressed as follows:

$$\boxed{\frac{\partial \mathcal{L}^{(T)}}{\partial W} = \sum_{t=1}^{T} \left.\frac{\partial \mathcal{L}^{(T)}}{\partial W}\right|_{(t)}}$$

## Bidirectional RNN