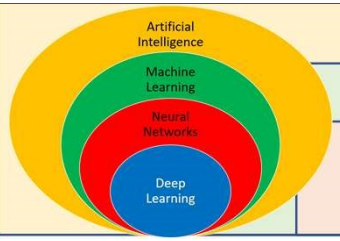


Artificial Intelligence (AI)

Machine Learning (ML)

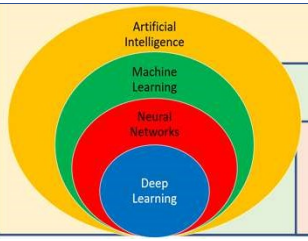
Neural Networks (NNs)

Deep Learning (DL)

Advanced Artificial Intelligence

Dr. Rastgoo

2022



Artificial Intelligence (AI)

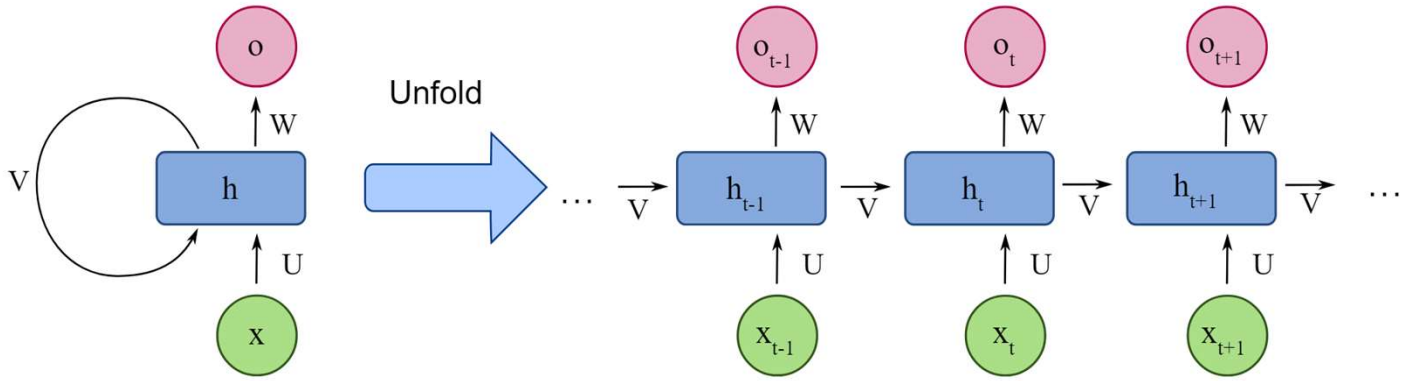
Machine Learning (ML)

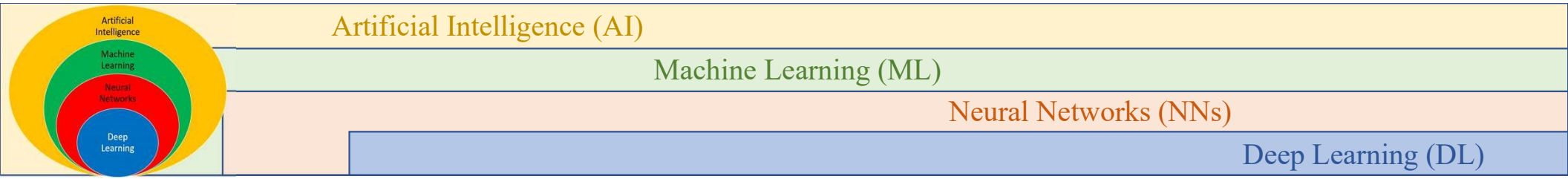
Neural Networks (NNs)

Deep Learning (DL)

Sequential Models

Part 1: Recurrent Neural Network (RNN)

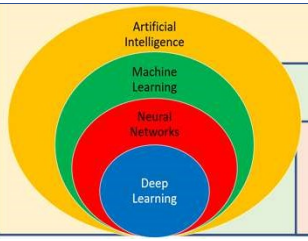




What is sequential data?

- If there is a particular order in which related things follow each other, we call it as a sequence.
- "I am a student" and "Am I a student".
- Do you think both sentences mean the same?

  NO! which means the position of words is very important!
- They are a sequence of words.



Artificial Intelligence (AI)

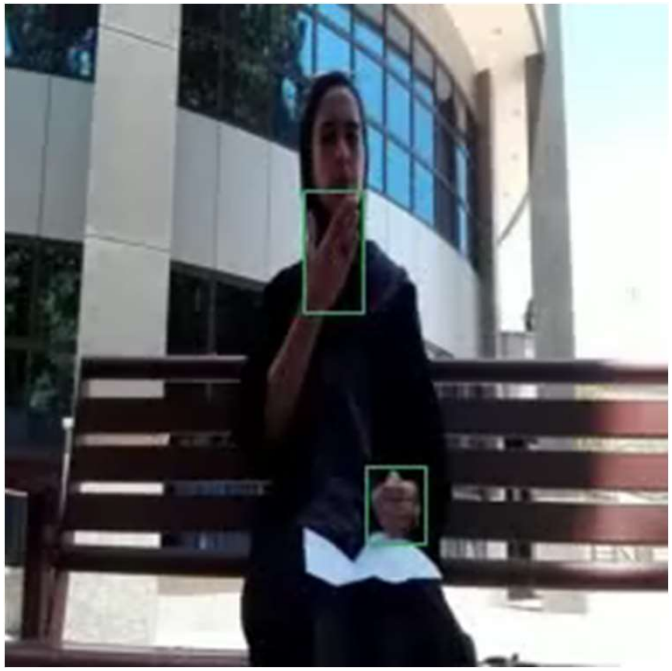
Machine Learning (ML)

Neural Networks (NNs)

Deep Learning (DL)

What is sequential data?

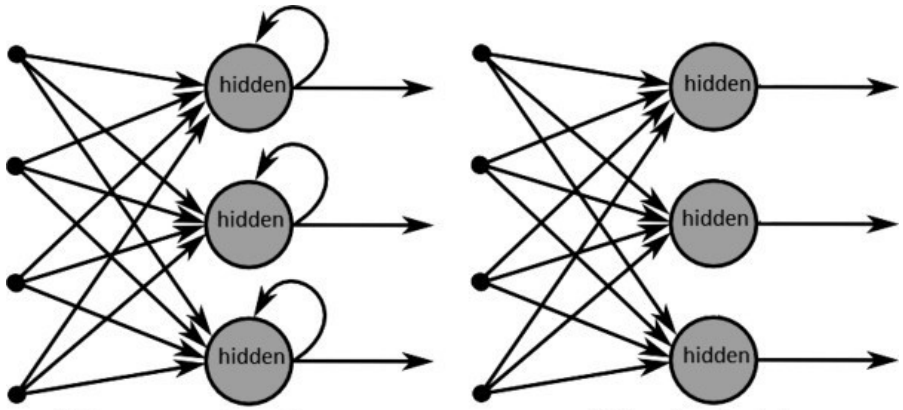
- Think of a video playing.
- You can easily predict the next scene if you have already watched that.
- But consider that you are sleepy, and you don't remember the position of frames(all jumbled frames in mind). Can you predict the next scene then??? Of course not!!!



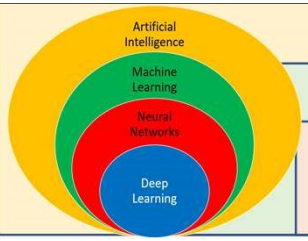
Recurrent Neural Network (RNN)

- When it comes to sequential or time series data, traditional feedforward networks cannot be used for learning and prediction.

- A mechanism is required that can retain past or historic information to forecast the future values.



(a) Recurrent neural network (b) Forward neural network



Artificial Intelligence (AI)

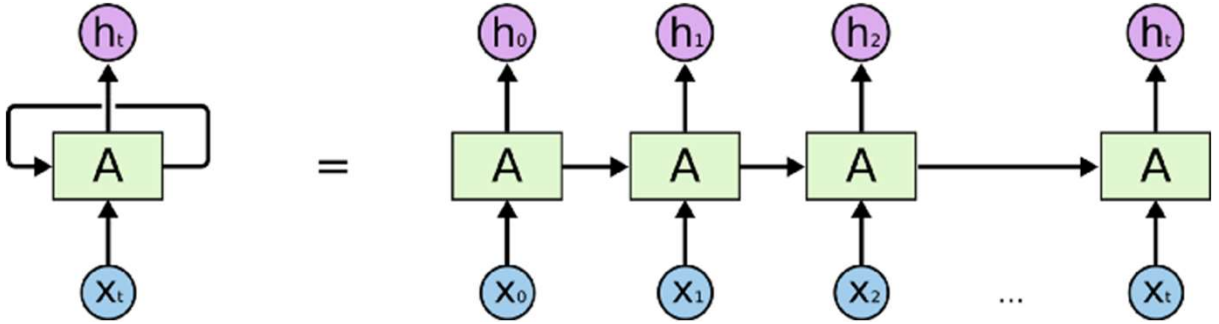
Machine Learning (ML)

Neural Networks (NNs)

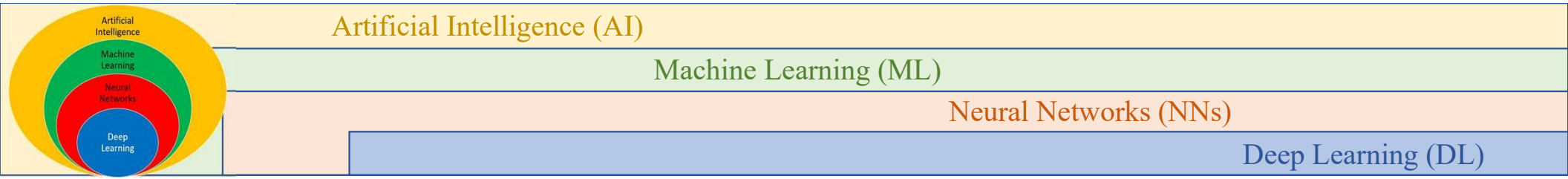
Deep Learning (DL)

Recurrent Neural Network (RNN)

- Recurrent neural networks or RNNs for short are a variant of the conventional feedforward artificial neural networks that can deal with sequential data and can be trained to hold the knowledge about the past.



The Green Box represents a Neural Network. The arrows indicate memory or simply feedback to the next input.



What is a Recurrent Neural Network?

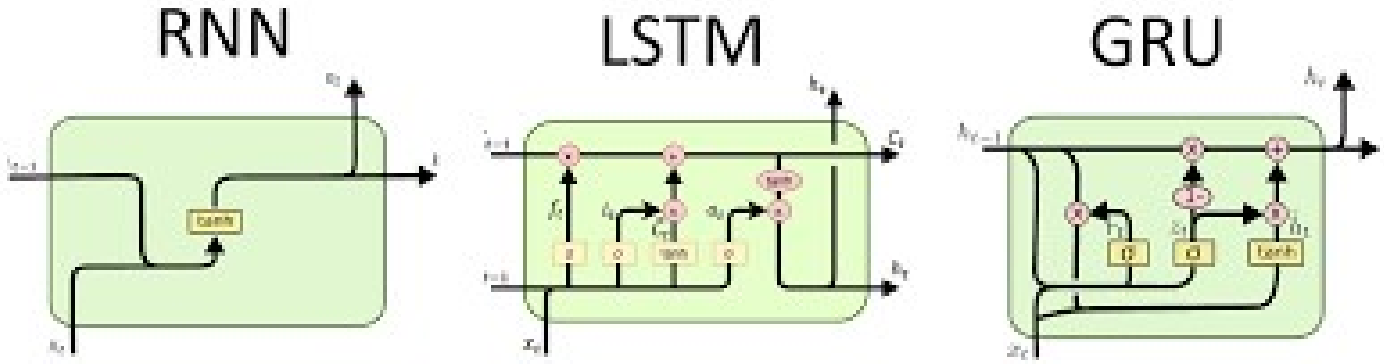
- A recurrent neural network (RNN) is a special type of an artificial neural network adapted to work for time series data or data that involves sequences.

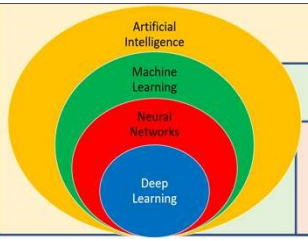
- Ordinary feed forward neural networks are only meant for data points, which are independent of each other.

- However, if we have data in a sequence such that one data point depends upon the previous data point, we need to modify the neural network to incorporate the dependencies between these data points.

What is a Recurrent Neural Network?

- RNNs have the concept of 'memory' that helps them store the states or information of previous inputs to generate the next output of the sequence.





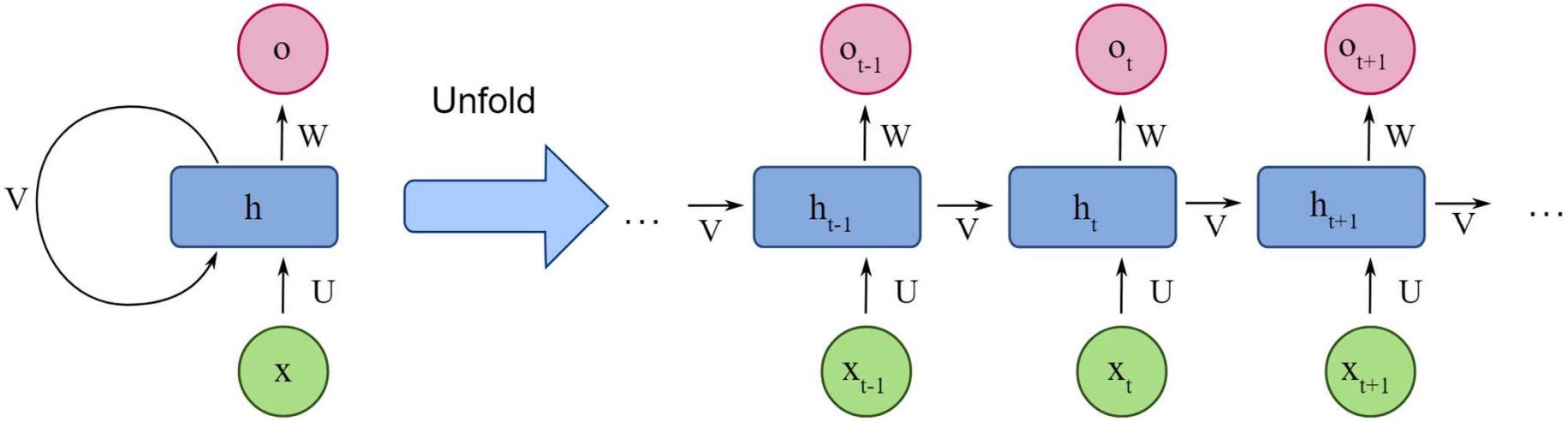
Artificial Intelligence (AI)

Machine Learning (ML)

Neural Networks (NNs)

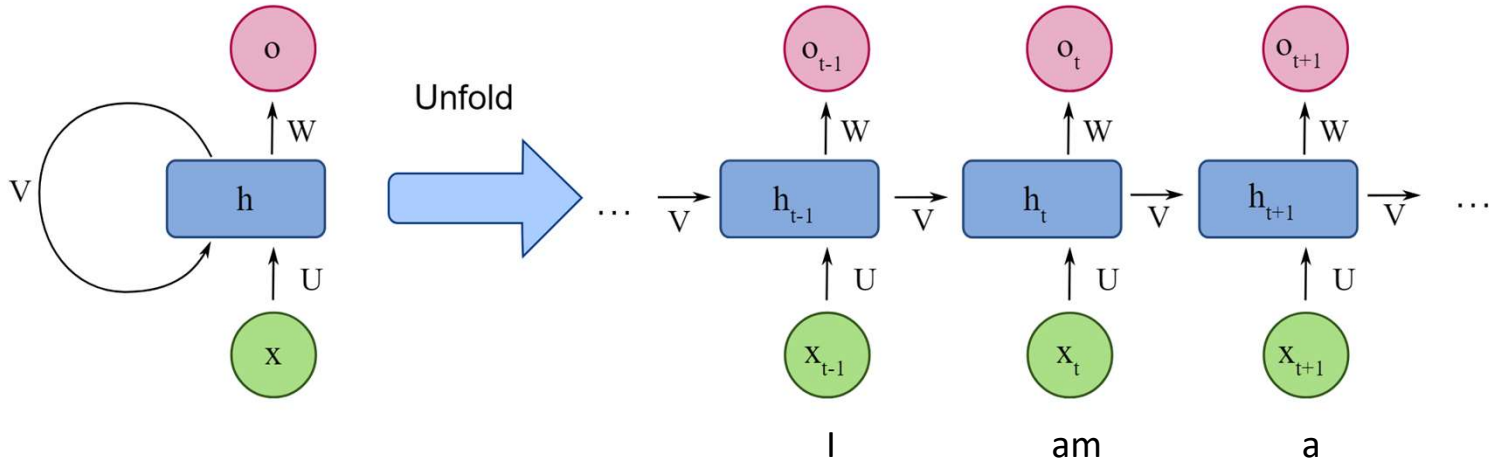
Deep Learning (DL)

What is a Recurrent Neural Network?



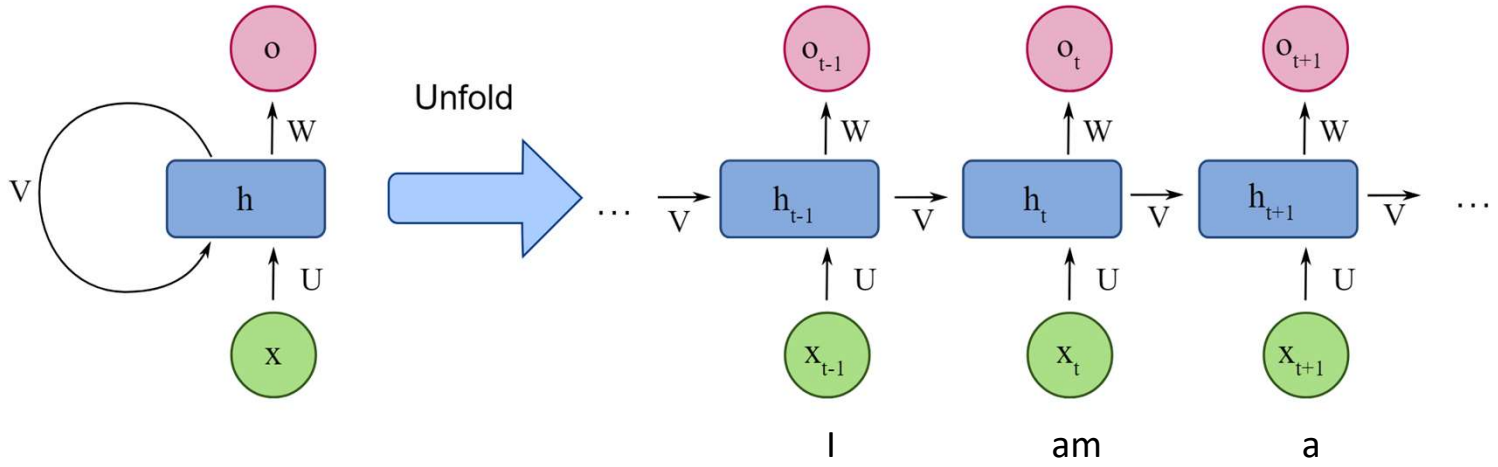
What is a Recurrent Neural Network?

- Consider a sequence "I am a student."



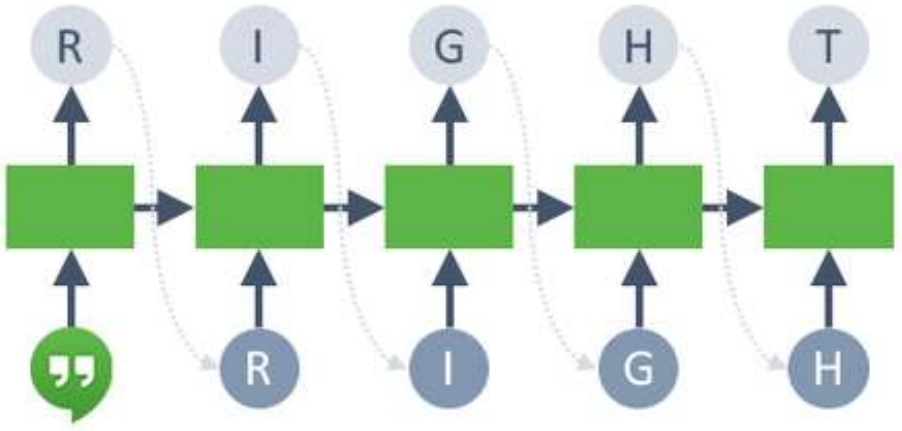
What is a Recurrent Neural Network?

- In a Feed Forward Neural Network, the Network is forward propagated only once per sample. But in RNN, the network is forward propagated equal to the number of time steps per sample.



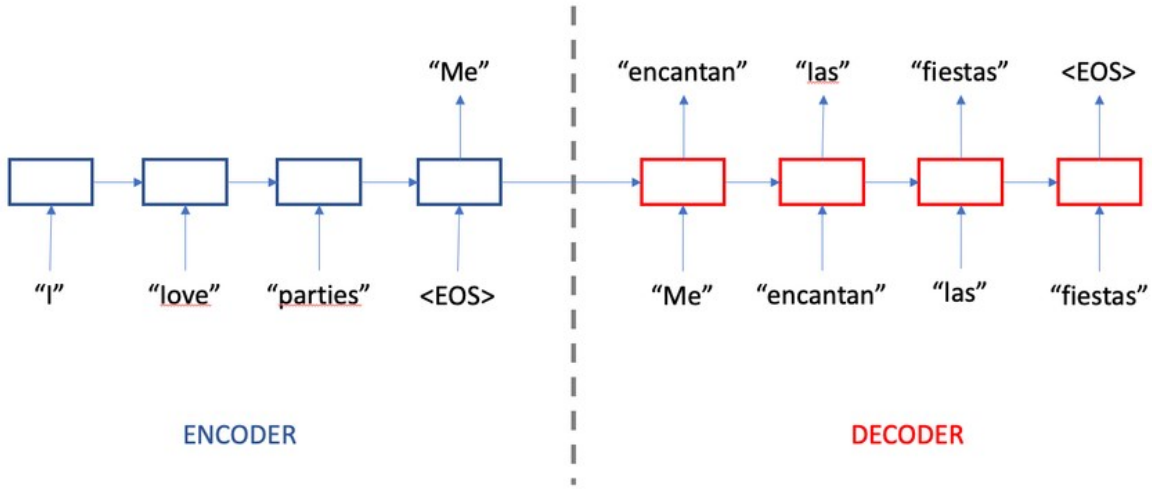
Different Problems solved with RNN

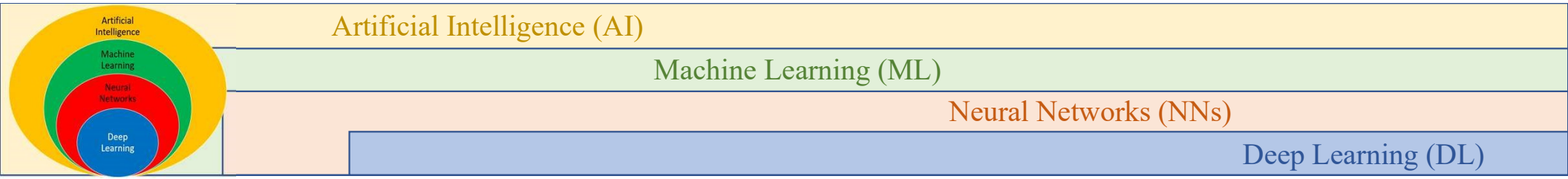
- **Generating Text**: Given a sequence of words we want to predict the probability of each word given the previous words.



Different Problems solved with RNN

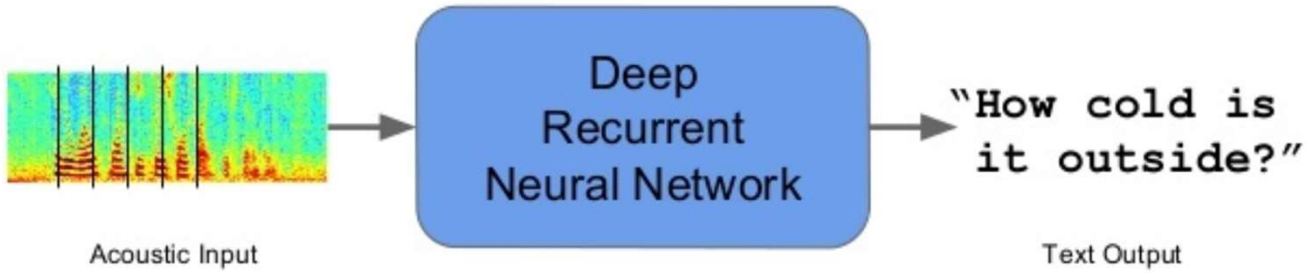
- **Machine Translation:** Machine Translation is similar to language modeling in that our input is a sequence of words in our source language (e.g. German). We want to output a sequence of words in our target language (e.g. English).





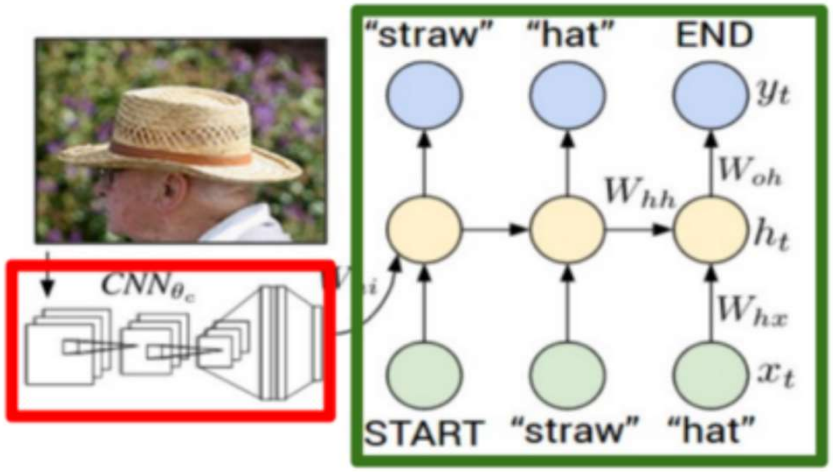
Different Problems solved with RNN

- **Speech Recognition:** Given an input sequence of acoustic signals from a sound wave, we can predict a sequence of phonetic segments together with their probabilities.



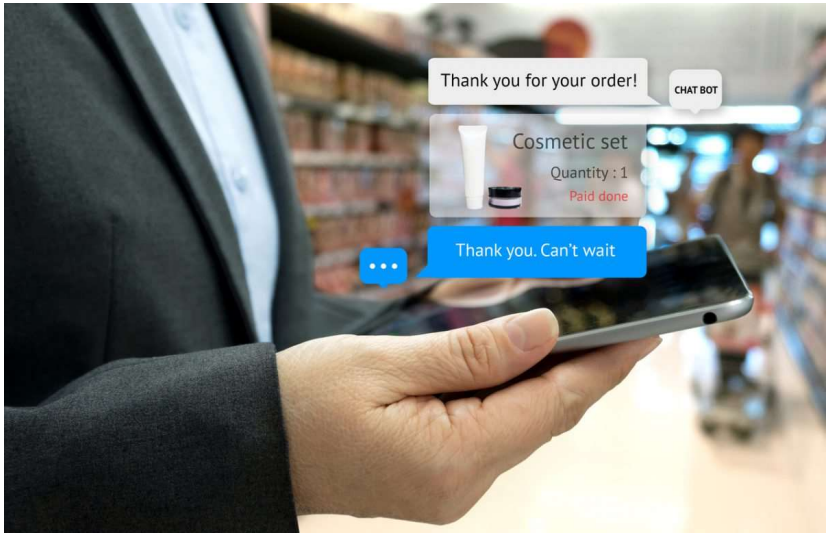
Different Problems solved with RNN

- **Generating Image Descriptions:** Together with convolutional Neural Networks, RNNs have been used as part of a model to generate descriptions for unlabeled images.

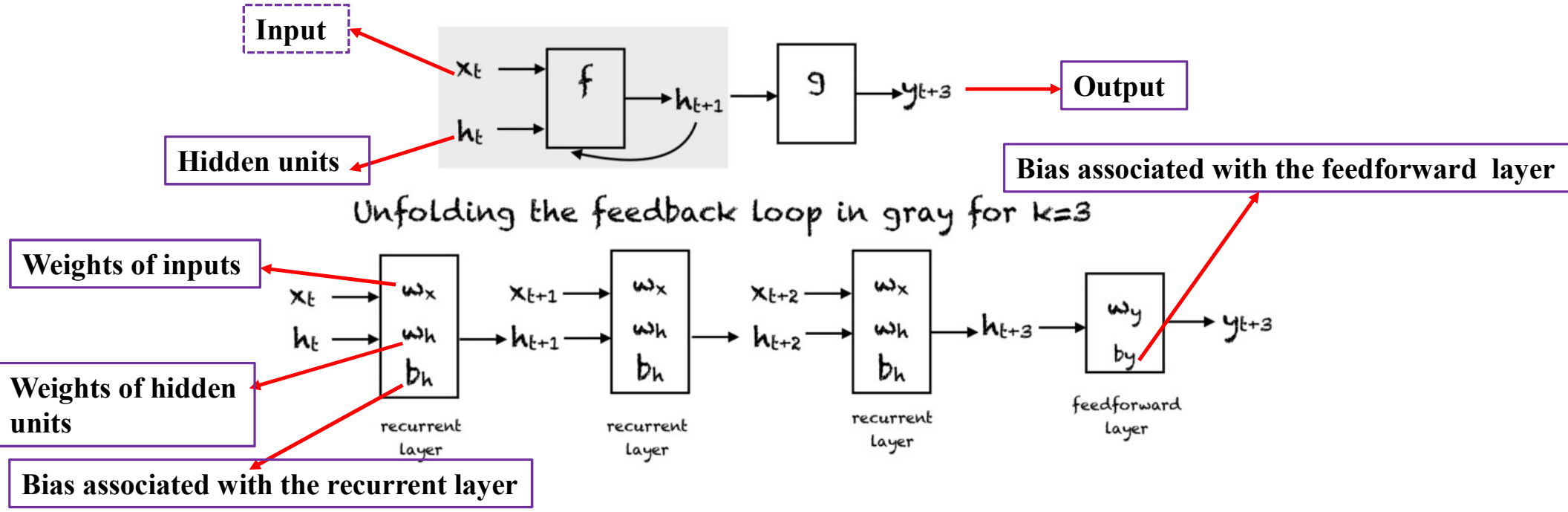


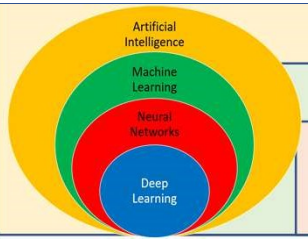
Different Problems solved with RNN

- **Chatbots:** Chatbots can give reply to your queries. When a sequence of words is given as the input, sequence of words will be generated at the output.



Unfolding a Recurrent Neural Network





Artificial Intelligence (AI)

Machine Learning (ML)

Neural Networks (NNs)

Deep Learning (DL)

Unfolding a Recurrent Neural Network

- At every time step, we can unfold the network for **K** time steps to get the output at time step **K+1**.
- The unfolded network is very similar to the feedforward neural network.

$$h_{t+1} = f(x_t, h_t, w_x, w_h, b_h) = f(w_x x_t + w_h h_t + b_h)$$

- The output y at time t is computed as:

$$y_t = f(h_t, w_y) = f(w_y \cdot h_t + b_y)$$

$\cdot$ is the dot product.

The Activation Function

- We can use any activation function we like in the recurrent neural network. Common choices are:

Sigmoid function

$$\frac{1}{1+e^{-x}}$$

Tanh function

$$\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$$

ReLU function

$$\max(0,x)$$

The Activation Function

- We can use any activation function we like in the recurrent neural network. Common choices are:

Sigmoid function

$$\frac{1}{1+e^{-x}}$$

Tanh function

$$\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$$

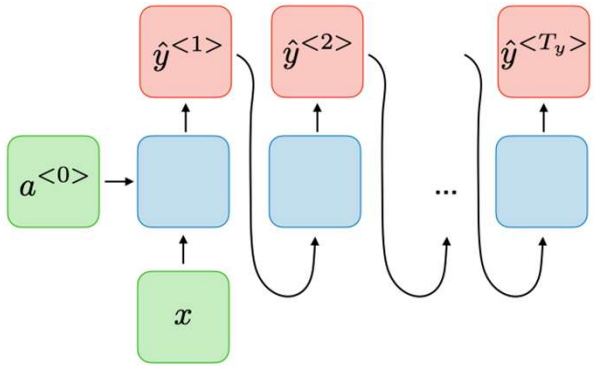
ReLU function

$$\max(0,x)$$

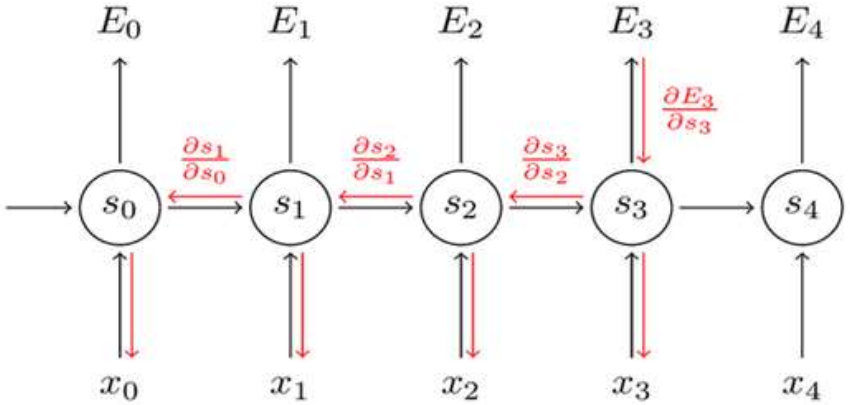
Training a Recurrent Neural Network

- The backpropagation algorithm of an artificial neural network is modified to include the unfolding in time to train the weights of the network.

- This algorithm is based on computing the gradient vector and is called backpropagation in time or BPTT algorithm for short.

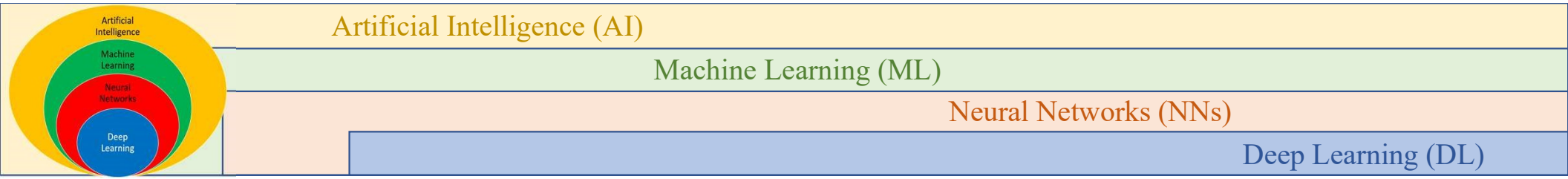


Training a Recurrent Neural Network



- The pseudo-code for training is given below:

1. Repeat till the stopping criterion is met:
    1. Set all h to zero.
    2. Repeat for t = 0 to n-k
        1. Forward propagate the network over the unfolded network for k time steps to compute all h and y
        2. Compute the error as: $e = y_{t+k} - p_{t+k}$
        3. Backpropagate the error across the unfolded network and update the weights

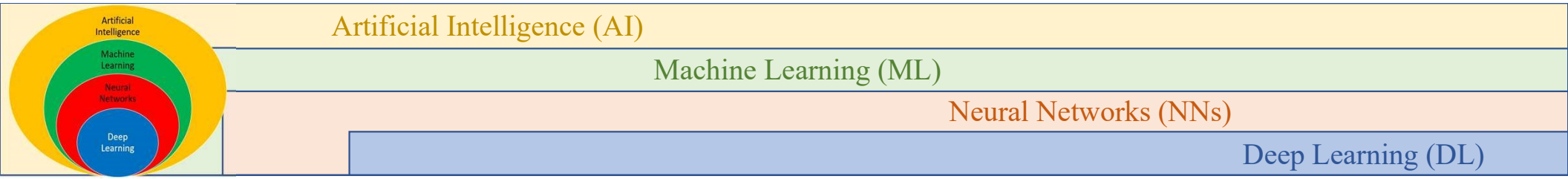


Vanishing and Exploding Gradients

- Let's first understand what is gradient?

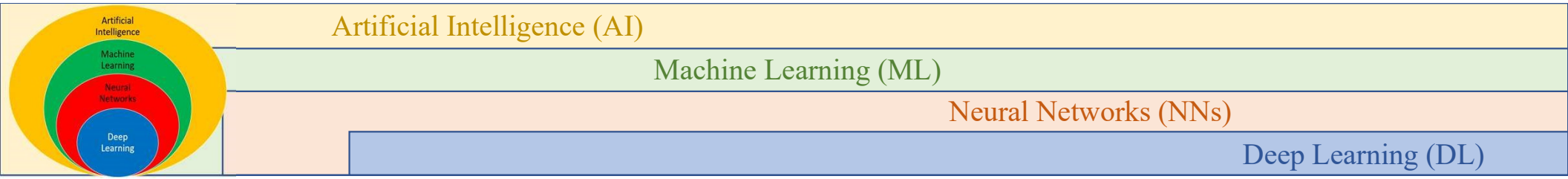
- **Gradient**: A gradient is a partial derivative with respect to its inputs.

- A gradient measures how much the output of a function changes, if you change the inputs a little bit.



Vanishing and Exploding Gradients

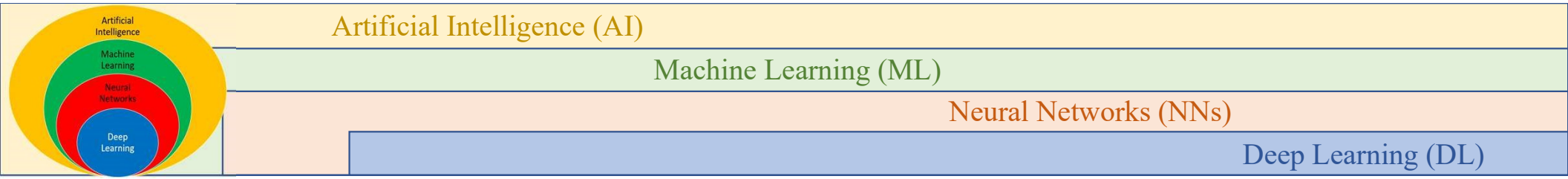
- You can also think of a gradient as the slope of a function.
- Higher the gradient, steeper the slope and the faster a model can learn.
- If the slope is almost zero, the model stops to learn.
- **A gradient simply measures the change in all weights with regard to the change in error.**



Gradient issues in RNN

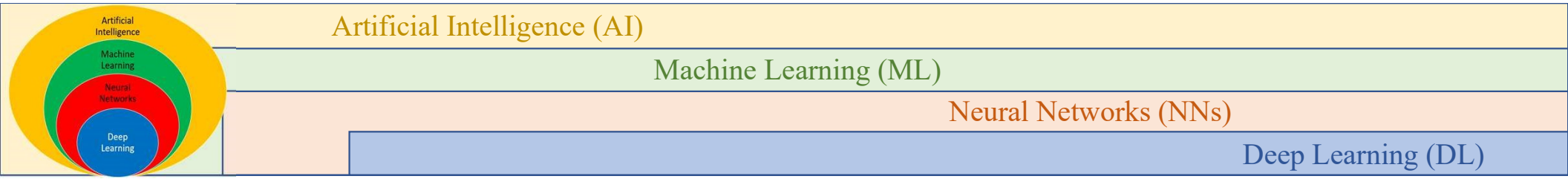
- While training an RNN algorithm, sometimes gradient can become too small or too large.

- So, the training of an RNN algorithm becomes very difficult in this situation. Due to this, following issues occur:
  - Poor Performance,
  - Low Accuracy,
  - Long Training Period



Exploding Gradient

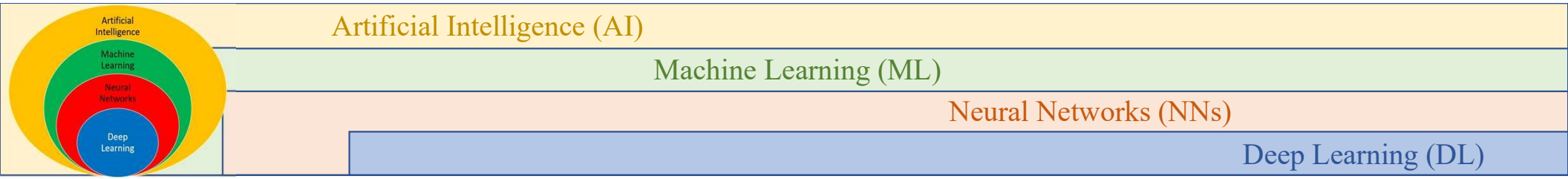
- When we assign high importance to the weights, exploding gradient issue occurs.
- In this case, values of a gradient become too large and slope tends to grow exponentially.
- This can be solved using following methods:
  - Identity Initialization,
  - Truncated Back-propagation,
  - Gradient Clipping.



Vanishing Gradient

- This issue occurs when the values of a gradient are too small and the model stops learning or takes way too long because of that.

- This can be solved using following methods:
  - Weight Initialization
  - Choosing the right Activation Function
  - LSTM (Long Short-Term Memory) Best way to solve the vanishing gradient issue is the use of LSTM (Long Short-Term Memory).

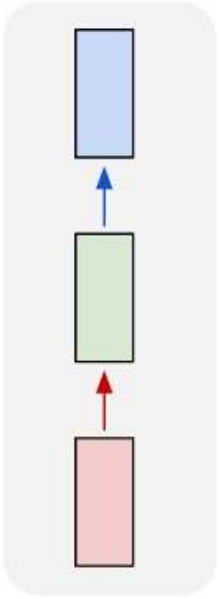


Types of RNN Architectures

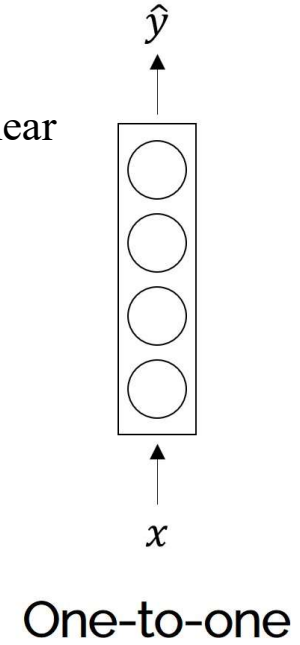
- The common architectures which are used for sequence learning are:
  - One to one
  - One to many
  - Many to one
  - Many to many

One to one

one to one

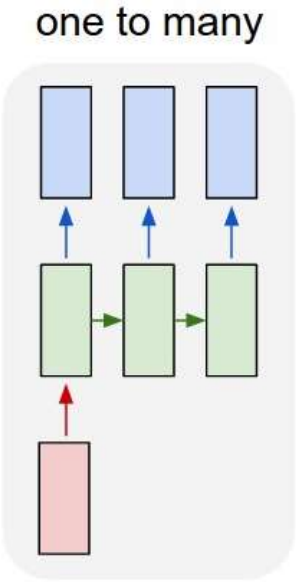


- This model is similar to a single layer neural network as it only provides linear predictions.
- It is mostly used fixed-size input ‘x’ and fixed-size output ‘y’ (example: image classification)

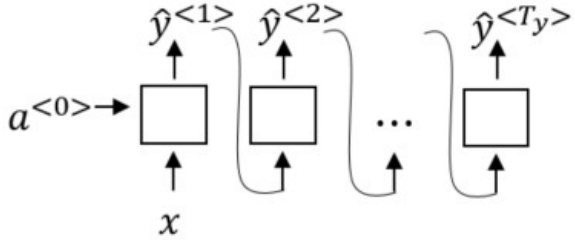


One-to-one

One to many

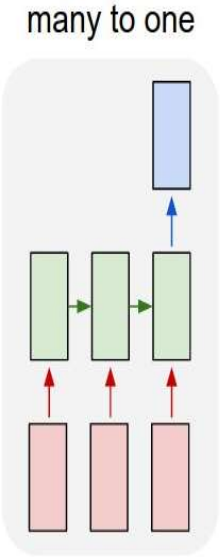


- This consist of a single input ‘x’, activation ‘a’, and multiple outputs ‘y’.
- Example: generating an audio stream. It takes a single audio stream as input and generates new tones or new music based on that stream.
- In some cases, it propagates the output ‘y’ to the next RNN units.

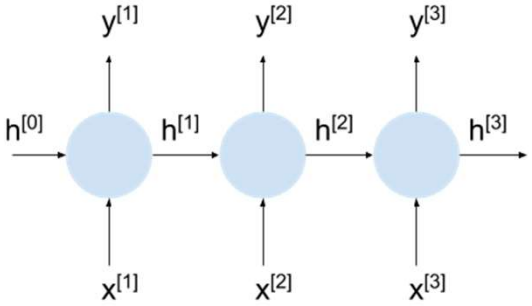


One to many

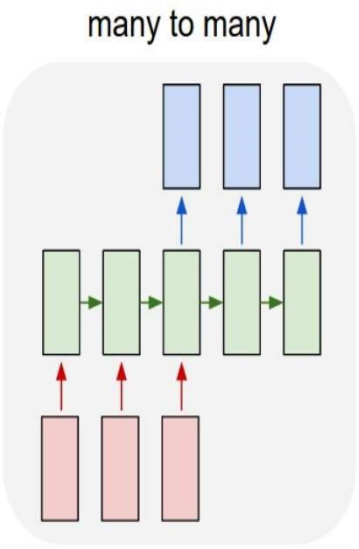
Many to one



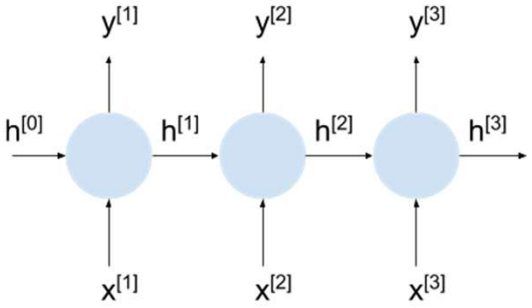
- This consist of multiple inputs ‘x’ (such as words or sentences), activation ‘a’ and produce a single output ‘y’ at the end.
- This type of architecture is mostly used to perform sentiment analysis as it processes the entire input (collection of words sentences) to produce a single output (positive, negative, or neutral sentiment)

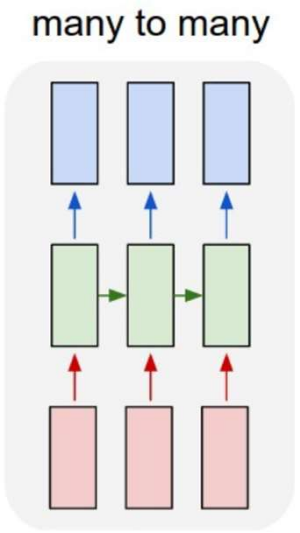


Many to many



- In this, a single frame is taken as input for each RNN unit. A-frame represents multiple inputs ‘x’, activations ‘a’ which are propagated through the network to produce output ‘y’ which are the classification result for each frame.
- It used mostly in video classification, where we try to classify each frame of the video

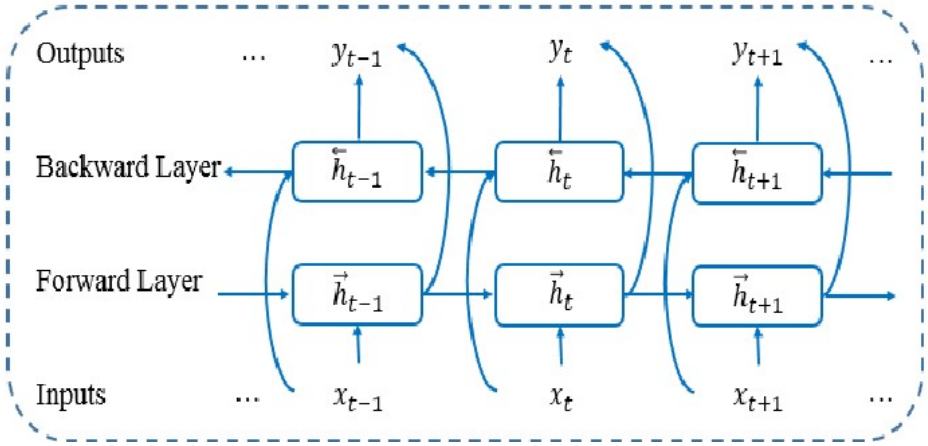




Bi- directional RNNs

- In this neural network, 2 hidden layers running in the opposite direction are connected to produce a single output.

- These layers allow the neural network to received information from both past as well as a future state.



Bi- directional RNNs

- For example, given a word sequence: 'I like programming'. The forward layer will input the sequence as it is while the backward layer will feed the sequence in the reverse order 'programming like I'.

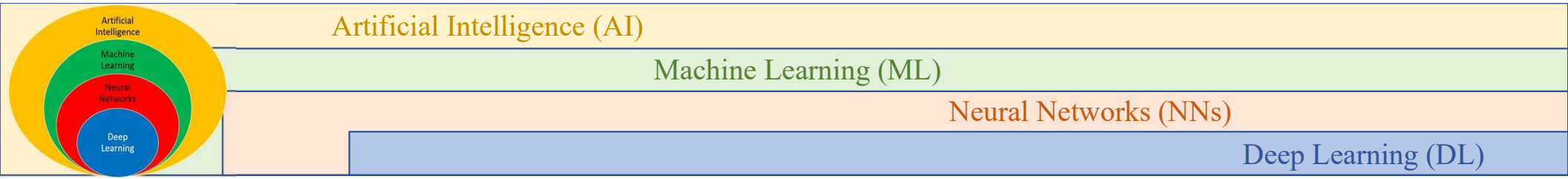
- The output for this will be calculated by concatenating the word sequence at each time step and generating the weight

Forward

'I like programming'.

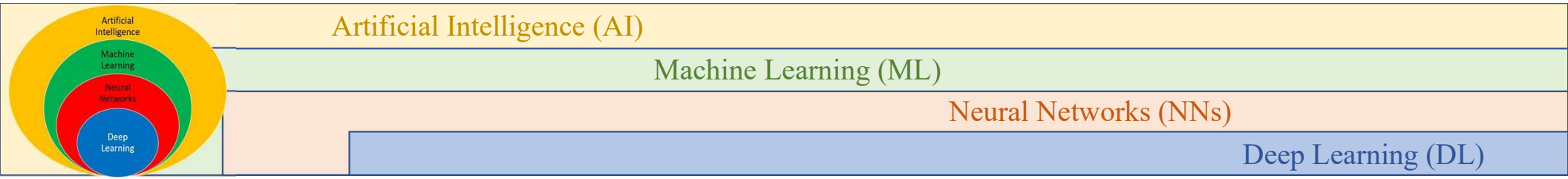
'programming like I'.

Backward



Notes

- RNNs remember each and every piece of information through timestamp.
- The memory state which stores information of all the state is useful for tasks such as sentence generation and time series prediction.
- RNNs can handle inputs and outputs of arbitrary length.

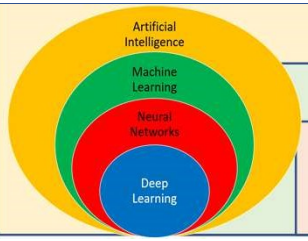


Notes

- RNNs share the same parameters across different time steps which means fewer parameters to train and computation cost.

- RNNs can not process very long sequences.

- RNNs face vanishing and exploding gradient problem.



Artificial Intelligence (AI)

Machine Learning (ML)

Neural Networks (NNs)

Deep Learning (DL)