Recurrent Neural Networks(RNN)

Ch. Md. Rakin Haider

Introduction

- Understanding something doesn’t require to start from scratch

- Example:
  - Reading an essay doesn’t require us to understand each word from scratch.
  - Movie sequence.

Introduction

- Traditional neural networks fails to capture impact of sequence in understanding.

- Vanilla neural networks
  - Fixed-sized vector as input (e.g. an image)
  - Fixed-sized vector as output (e.g. probabilities of different classes)
  - Fixed amount of computational steps!!!!

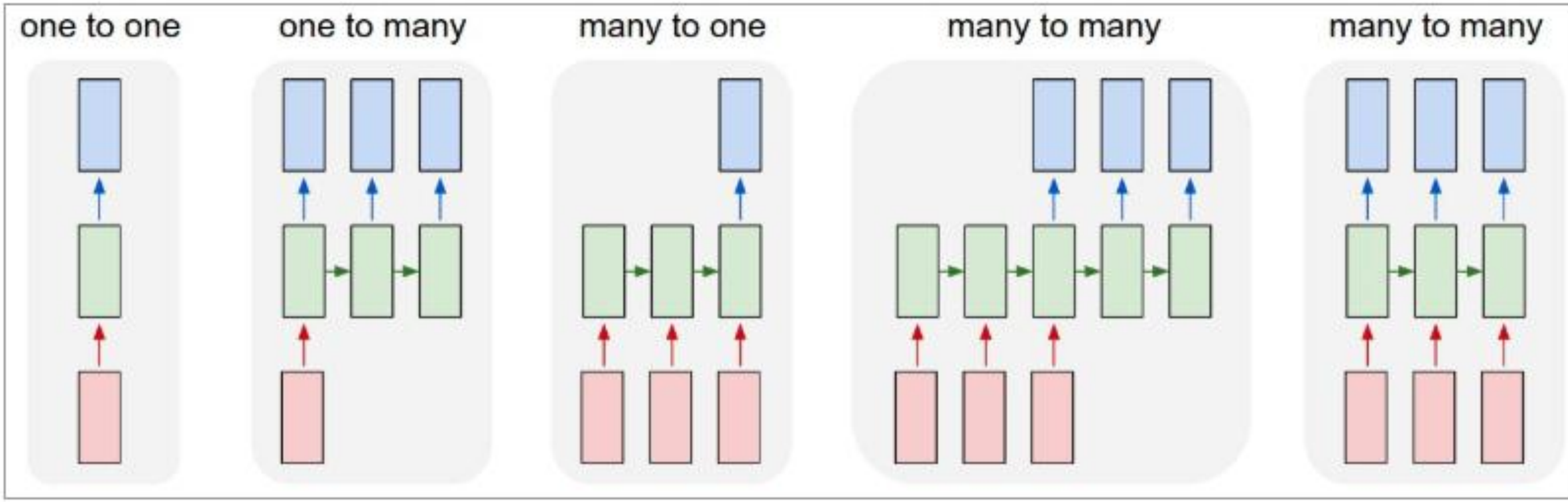
Introduction

- Does Vanilla NN works in every case?

- What if
  - Variable size input (e.g. text message)
  - Variable size output (e.g. Generate a caption)

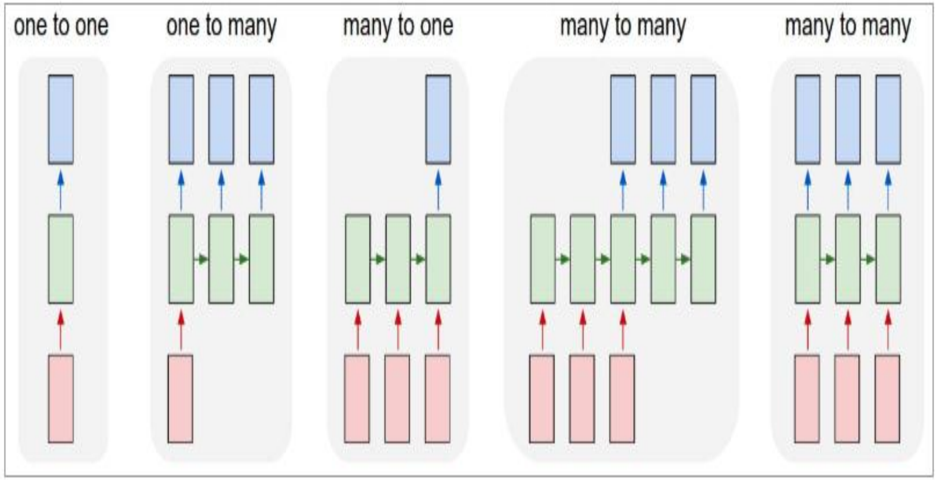
- Imagine you want to classify what kind of event is happening at every point in a movie.
  - Do you need information about previous events ?

Introduction

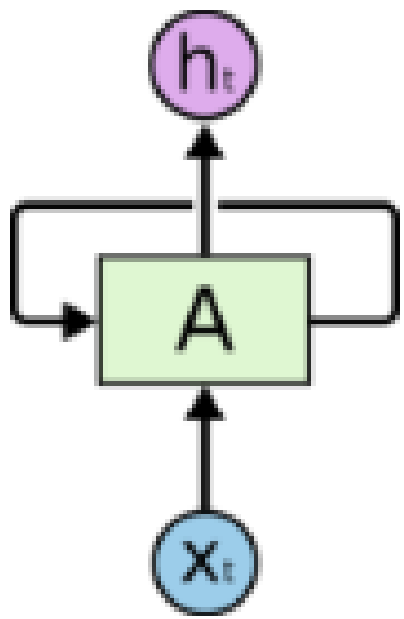


Introduction

- Image classification
- Image captioning
- Sentiment analysis
- Machine Translation
- Video classification



Recurrent Neural Networks

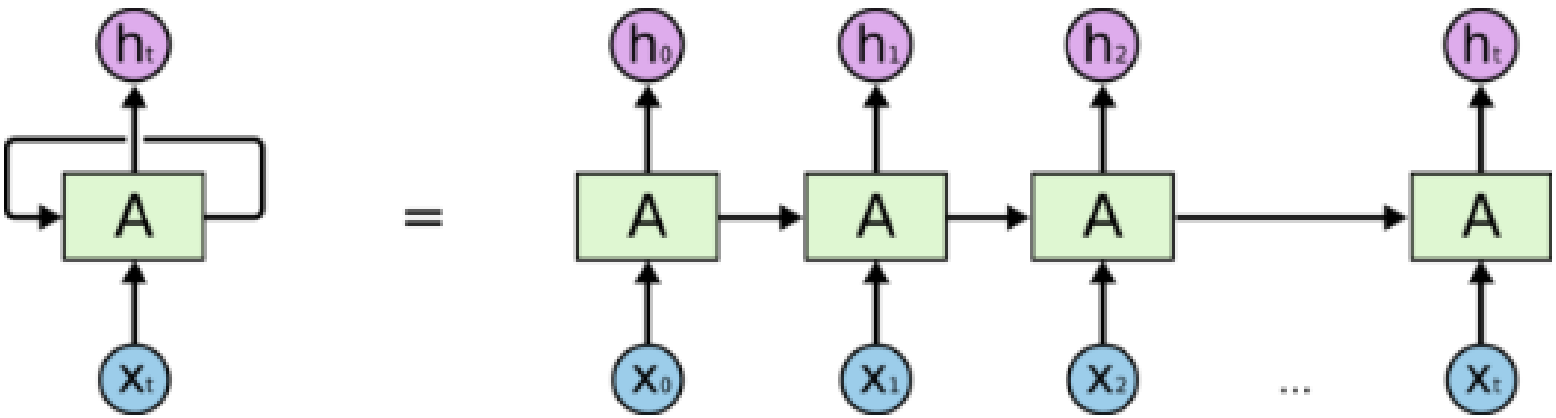


Recurrent Neural Networks have loops.

Recurrent Neural Networks

- Networks with loops
  - Allows information to persist
- Can be thought of as multiple copies of the same network
- Each copy passes a message to a successor.

Recurrent Neural Networks



An unrolled recurrent neural network.

Recurrent Neural Networks

- Chain-like nature

- Reveals that recurrent neural networks are intimately related to sequences and lists

- Applied to variety of problems
  - speech recognition, language modeling, translation, image captioning

Understanding RNN

- Problem:
  - Given a sequence of letters predict the next letter from vocabulary.
  - Here, vocabulary = {h,e,l,o}
  - Given “hell” need to predict o.

Understanding RNN

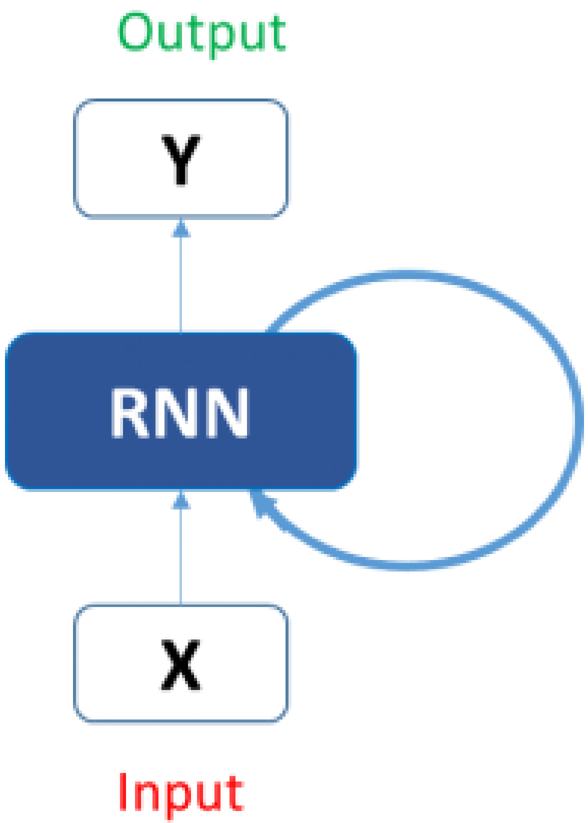
- Real World Problem:
  - Given a sequence of words predict the next word from vocabulary to make e meaningful sentence.
  - A bit complex to start with
  - Let’s consider a smaller problem

Understanding RNN

- Problem:
  - Given a sequence of letters predict the next letter from vocabulary.
  - Here, vocabulary = {h,e,l,o} (Very small for simplicity)
  - Given “hell” need to predict o.

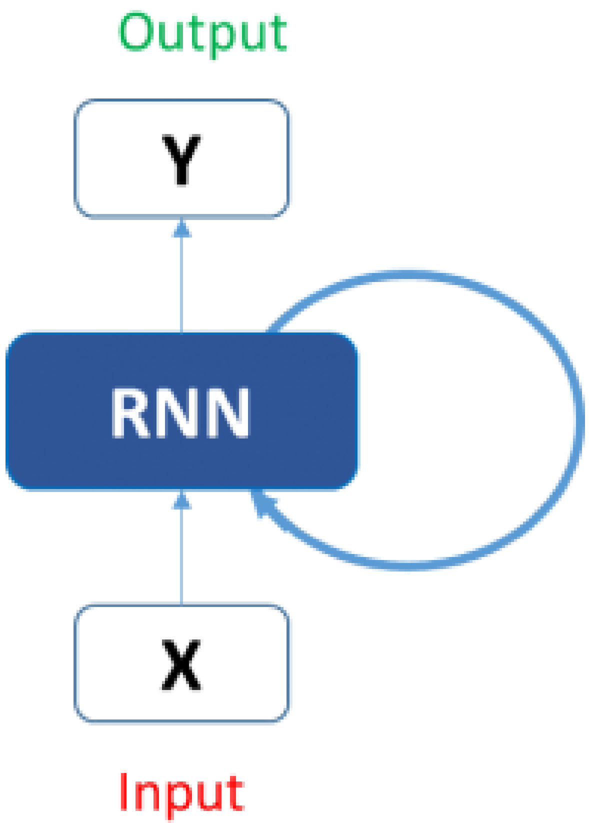
Understanding RNN

- Structure of network



Understanding RNN

- RNN block applies recurrence formula to the input vector and its previous state.

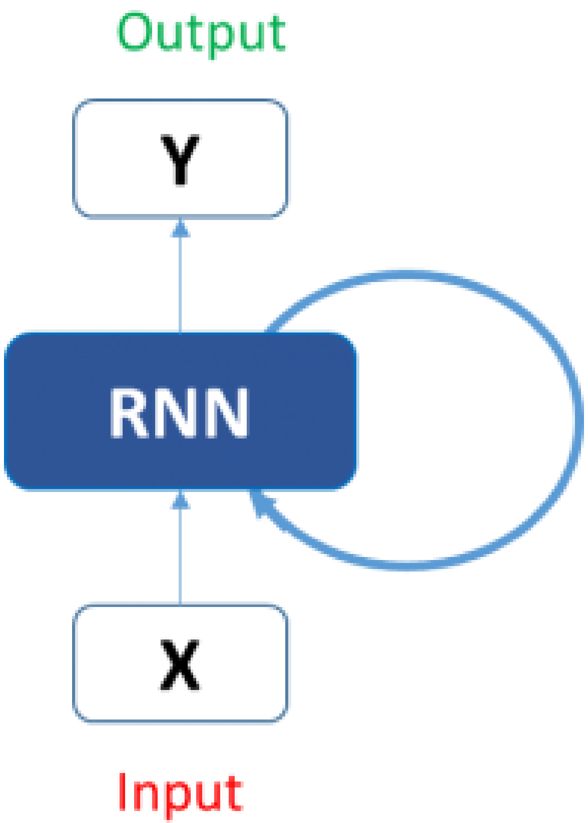


Understanding RNN

- Each input letter corresponds to a time steps of the input.
- For example, if at time t, the input is “e”, at time t-1, the input was “h”.
- The recurrence formula is applied to e and h both. and we get a new state.

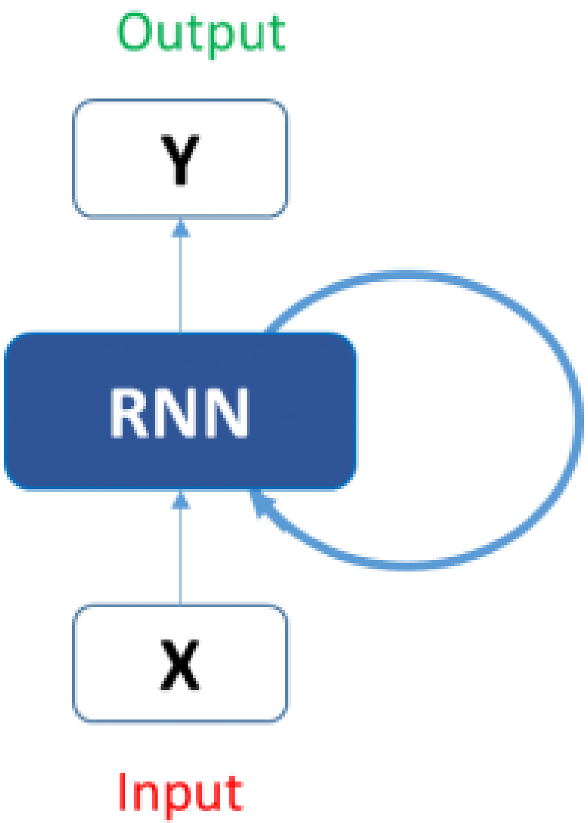
Understanding RNN

- This RNN's parameters are the three matrices -
  - W_hh : Matrix based on the Previous Hidden State
  - W_xh : Matrix based on the Current Input
  - W_hy : Matrix based between hidden state and output



Understanding RNN

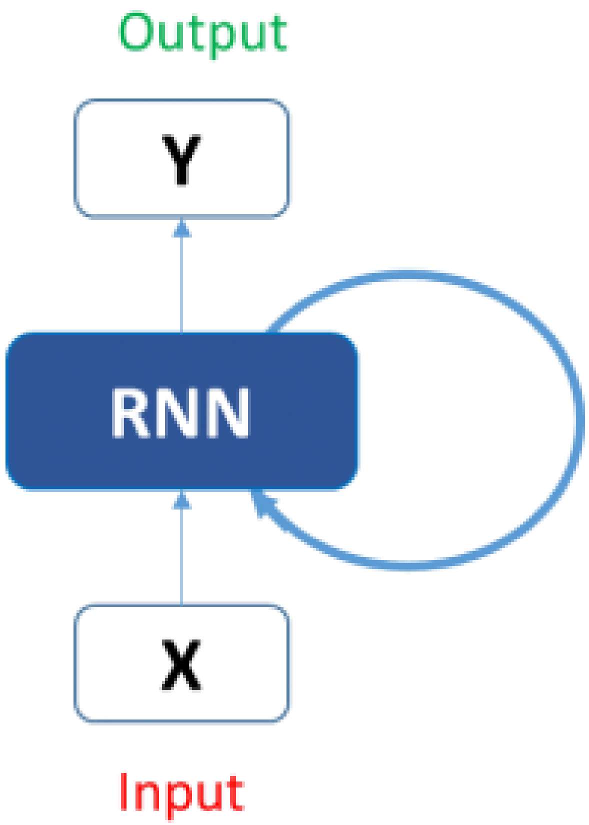
- This RNN's parameters are the three matrices -
  - W_hh : Matrix based on the Previous Hidden State
  - W_xh : Matrix based on the Current Input
  - W_hy : Matrix between hidden state and output



Understanding RNN

- Recurrence Formula

$$h_t = f(h_{t-1}, x_t)$$

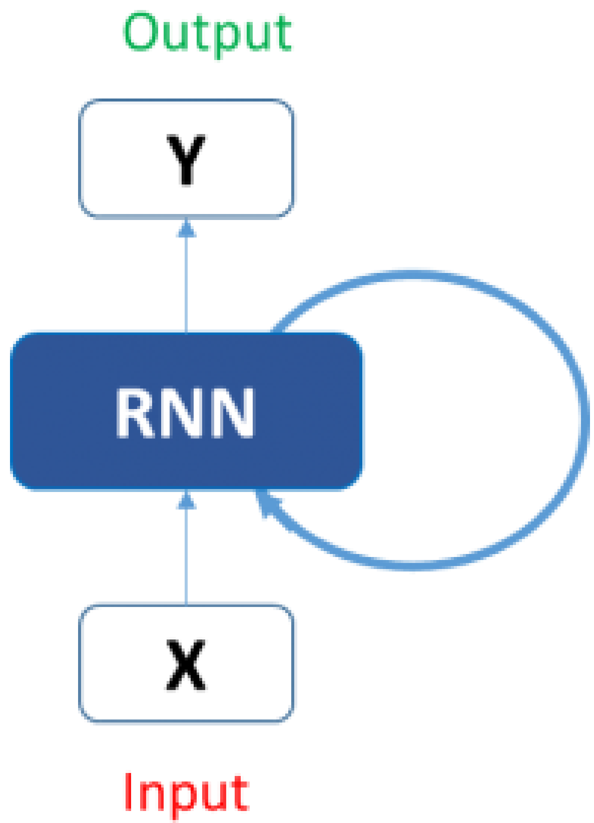


Understanding RNN

- Recurrence Formula

$$h_t = f(h_{t-1}, x_t)$$

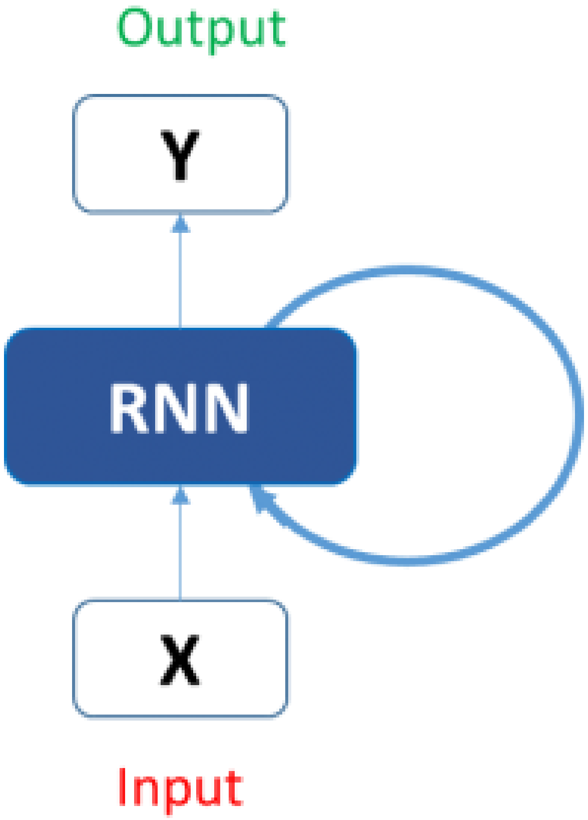
  - Ht is new state
  - H(t-e) is old state
  - Xt is the current intput



Understanding RNN

- Recurrence Formula

$$h_t = \tanh\left(W_{hh} h_{t-1} + W_{xh} x_t\right)$$

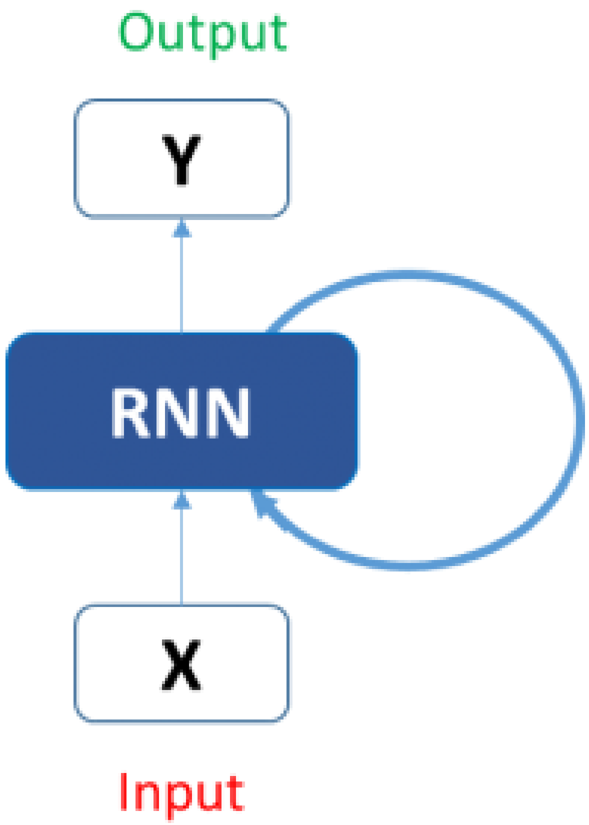


Understanding RNN

- Recurrence Formula

$$h_t = \tanh\left(W_{hh}h_{t-1} + W_{xh}x_t\right)$$

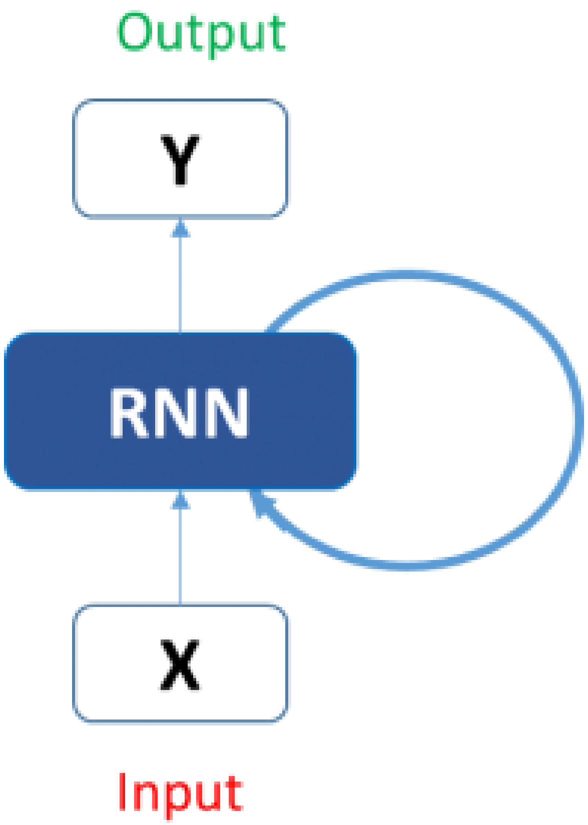
  - Just taking the immediate previous state into consideration
  - For longer sequences the equation can involve multiple such states.



Understanding RNN

- Once the current state is calculate the output can be calculated

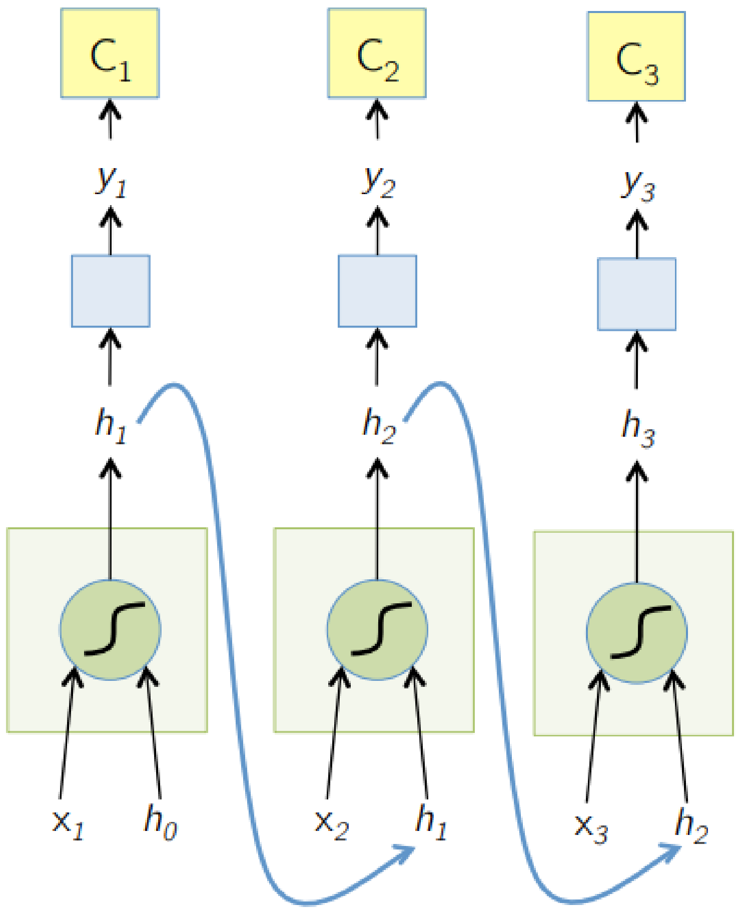
$$y_t = W_{hy}h_t$$



Understanding RNN

- Summary
  - A single time step is supplied to the network
  - Calculate current state
  - Current state becomes previous state for the next time step
  - Go as many time steps as required
  - After all time step compute the final output
  - Compare with actual output and compute error
  - Backpropagate error.

Visualizing RNN



Backpropagating RNN

- Backpropagation Through Time (BPTT)
  - Unroll the network
  - In case of an RNN, if yt is the predicted value ȳt is the actual value, the error is calculated as a cross entropy loss –

$$E_t(\bar{y}_t, y_t) = -\bar{y}_t \log(y_t)$$

$$E(\bar{y}, y) = -\sum \bar{y}_t \log(y_t)$$

Backpropagating RNN

- Backpropagation Through Time (BPTT)
  - Cross entropy error is computed using the current output and the actual output
  - Network is unrolled for all the time steps
  - The gradient is calculated for each time step with respect to the weight parameter
  - Now that the weight is the same for all the time steps the gradients can be combined together for all time steps
  - The weights are then updated for both recurrent neuron and the dense layers

Problems with RNN

- Understanding long range dependencies

- Vanishing Gradient

$$\frac{\partial C_t}{\partial h_1} = \left(\frac{\partial C_t}{\partial y_t}\right)\left(\frac{\partial y_t}{\partial h_1}\right)$$

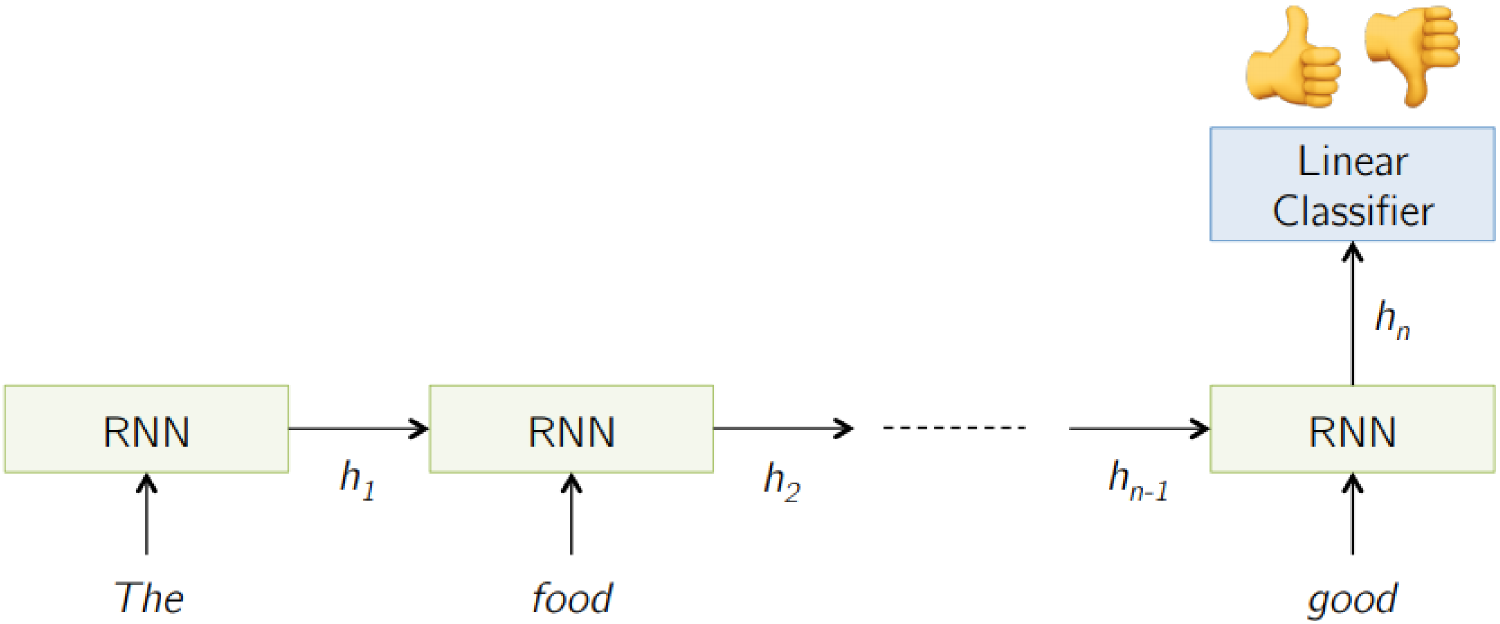
$$= \left(\frac{\partial C_t}{\partial y_t}\right)\left(\frac{\partial y_t}{\partial h_t}\right)\left(\frac{\partial h_t}{\partial h_{t-1}}\right)\cdots\left(\frac{\partial h_2}{\partial h_1}\right)$$

- Exploding Gradient

Applications

- Sentiment Classification
  - restaurant review from Yelp! OR
  - movie review from IMDB OR
  - Classify as positive or negative
- Inputs: Multiple words, one or more sentences
- Outputs: Positive / Negative classification

Applications



Reference

- https://medium.com/explore-artificial-intelligence/an-introduction-to-recurrent-neural-networks-72c97bf0912
- https://www.analyticsvidhya.com/blog/2017/12/introduction-to-recurrent-neural-networks/
- http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture10.pdf

Reference

- http://slazebni.cs.illinois.edu/spring17/lec02_rnn.pdf