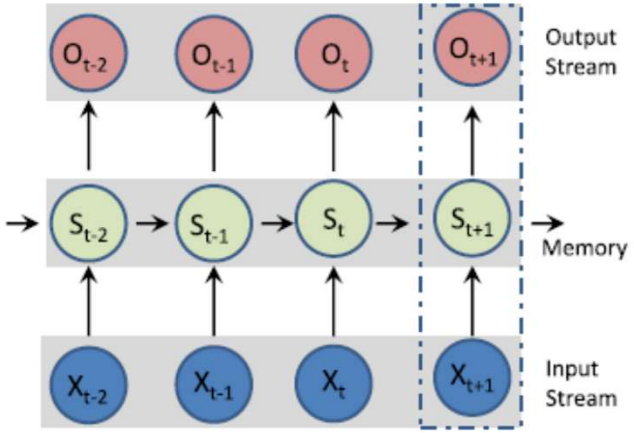
What is RNN?

RNN



Recurrent Neural Network

- Proposed in [13] is a NN capable of analyzing stream of data
- Useful in applications where the output depends on the previous computations
- Shares the same weights across all steps

Pros:

- Can memorize sequential events
- Can model time dependencies
- Has shown great success in many Natural Language Processing applications

Cons:

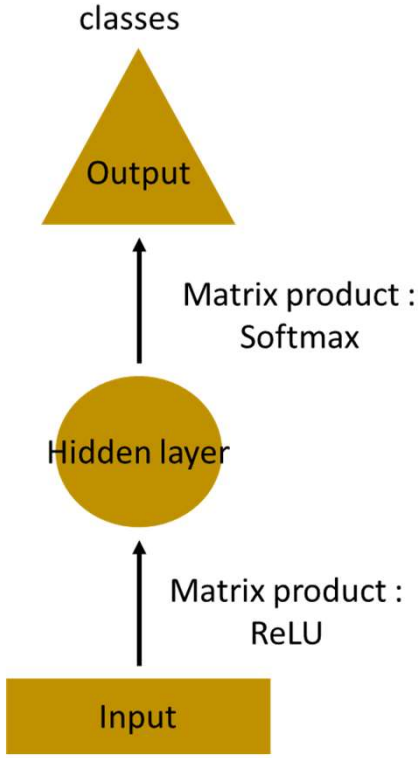
- Learning issues are frequent due to the vanishing gradient and exploding gradient problems

Recurrent Neural Network

- Need for a NN to deal with sequential information
  - Sentiment classification
  - Image captions
  - Language
- Use RNN to map semantic inputs of varying types and lengths

Multilayer NN

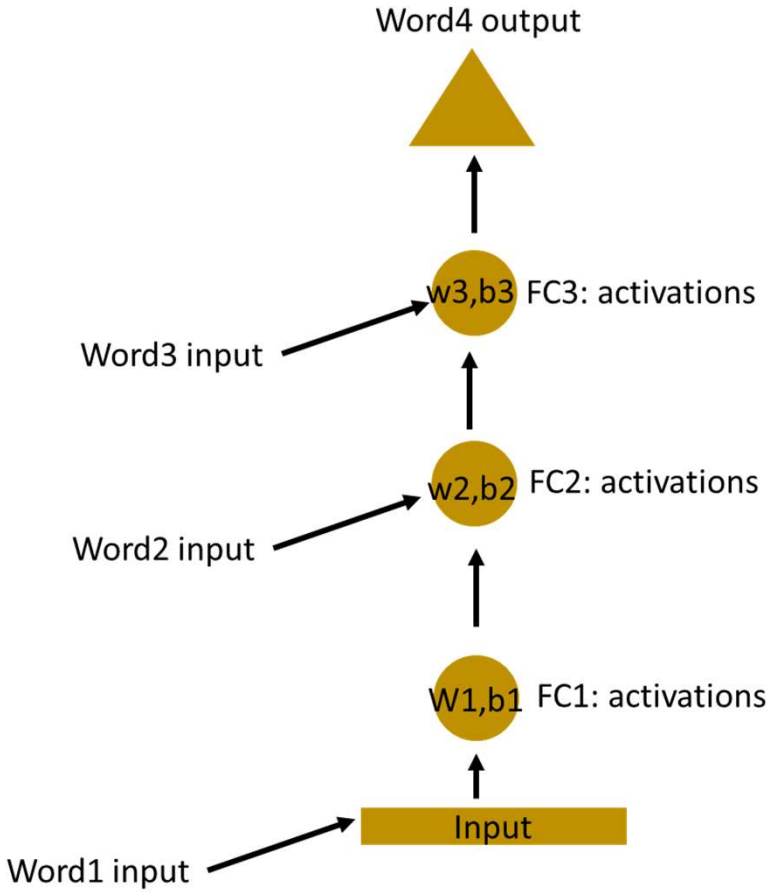
An input layer, a hidden layer, and an output layer to get a predicated output



- https://www.analyticsvidhya.com/blog/2017/12/introduction-to-recurrent-neural-networks/

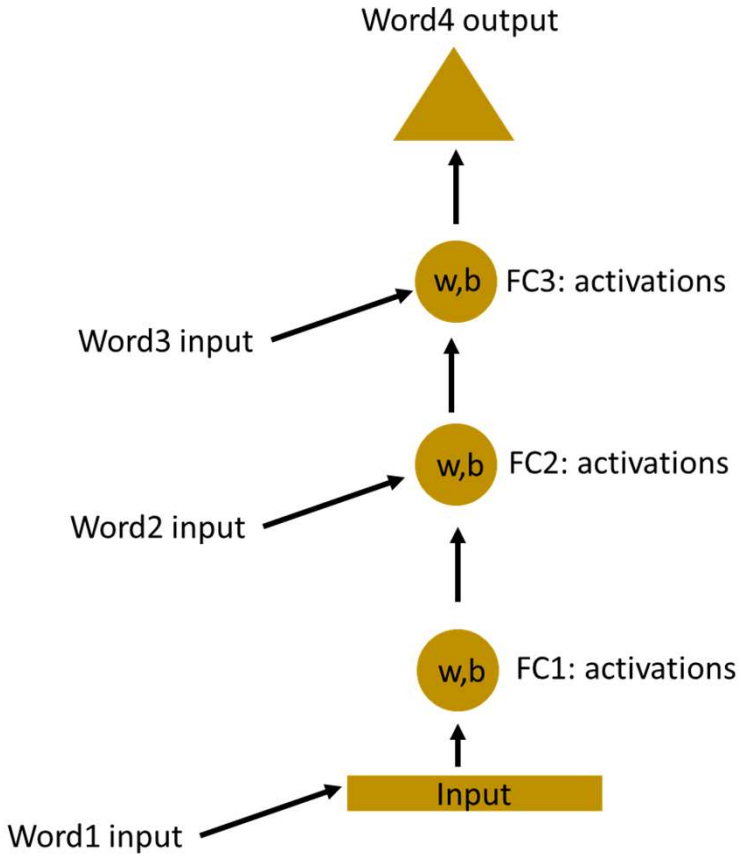
MNN for Multiple Hidden Layers and Inputs

- Now multiple hidden layers, but inputs to hidden layers for semantics to create relationship between successive inputs
- Note different weights and bias.
- Each hidden layer has its own weights and activations and behaves independently



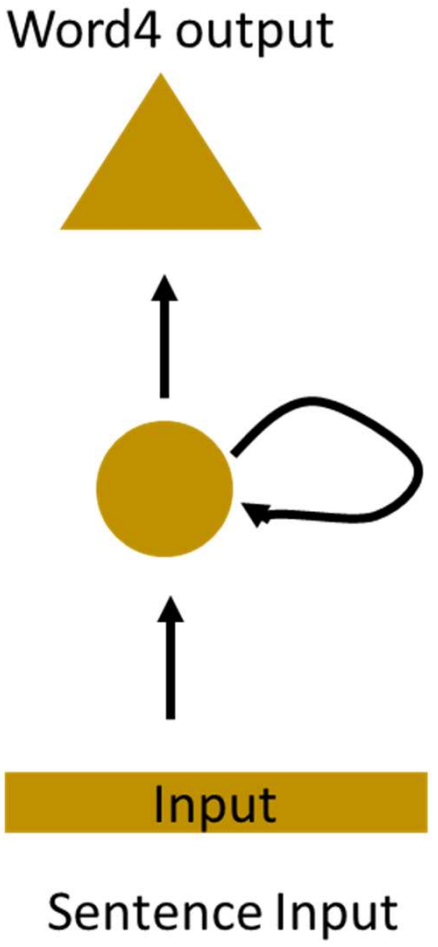
RNN Unfolded

- Now make the weights and bias same for the hidden layers
- To combine the hidden layers together

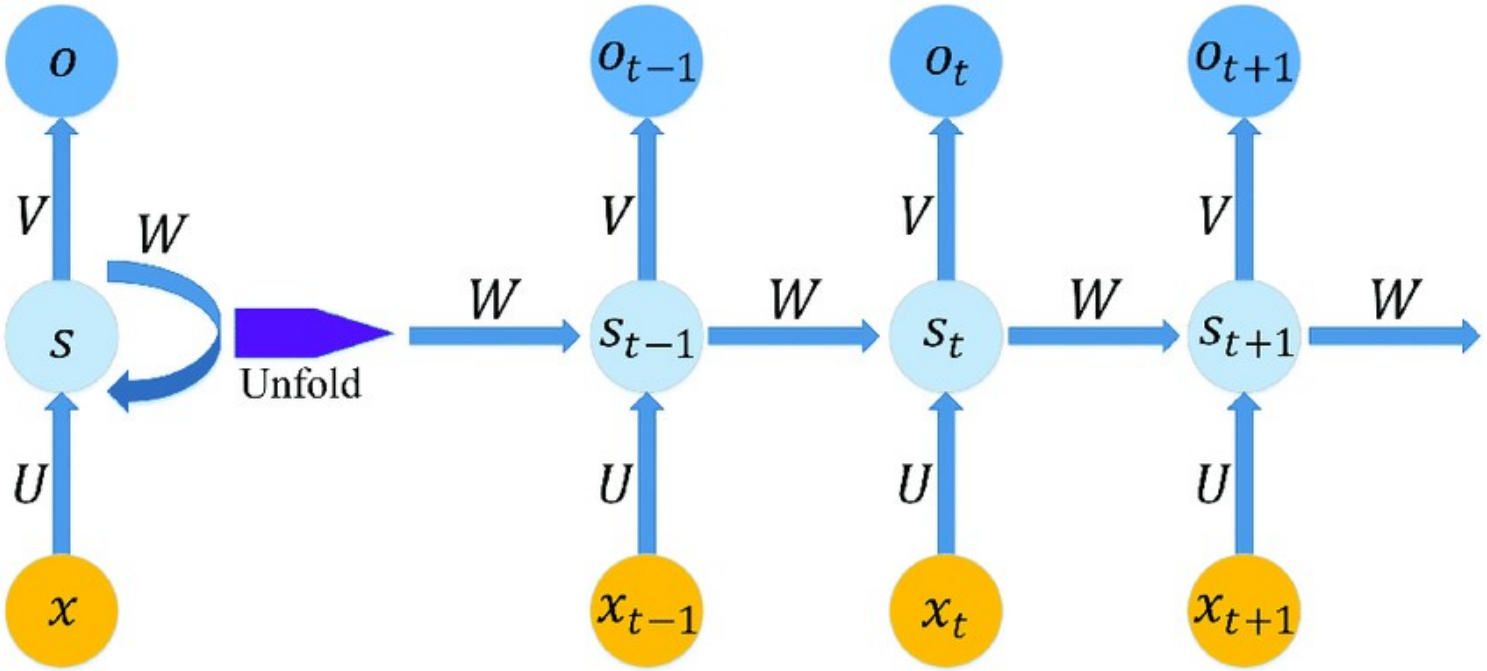


RNN Folded

- Combine the hidden layers, since the weights and bias of them are same.
- All these hidden layers can be rolled in a single recurrent layer.
- Recurrent neuron stores the states of a previous input and combines with the current input, thereby preserving some relationship of the current input with the previous input.



A RNN and its Unfolding Structure



- A recurrent neural network and the unfolding architecture. U, V and W are the weights of the hidden layer, the output layer and the hidden state, respectively.
- x_t and o_t are the input vector and output result at time t, respectively

RNN Ex.

- In a case scenario of natural language processing
- Teach a word, “Hello”
- Input “H-E-L-L”
- Predict output of “O”