



# Sequence Models & Recurrent Neural Networks (RNNs)

Understanding Deep Recurrent Neural Networks (RNNs)



Santhoopa Jayawardhana

Jul 27, 2020 · 5 min read ★



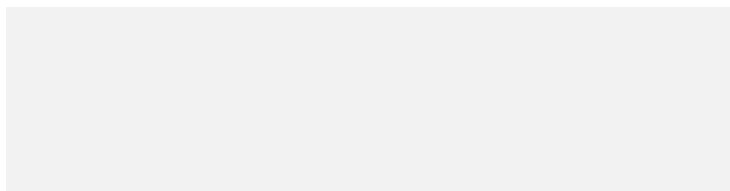
## Sequence Models

Sequence models are the machine learning models that input or output sequences of data. Sequential data includes text streams, audio clips, video clips, time-series data and etc.

Recurrent Neural Networks (RNNs) is a popular algorithm used in sequence models.

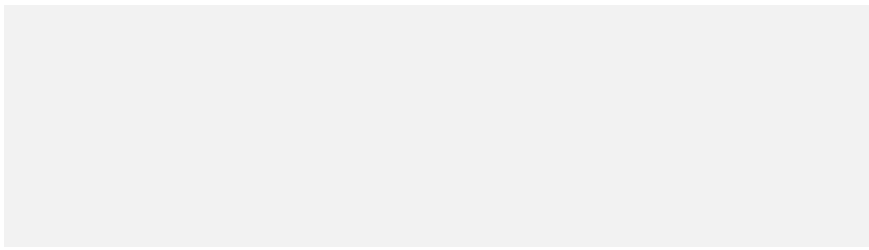
### Applications of Sequence Models

1. **Speech recognition:** In speech recognition, an audio clip is given as an input and then the model has to generate its text transcript. Here both the input and output are sequences of data.



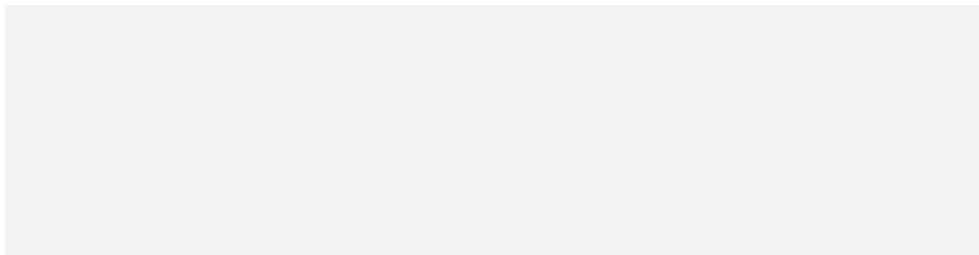
Speech recognition (Source: Author)

2. **Sentiment Classification:** In sentiment classification opinions expressed in a piece of text is categorized. Here the input is a sequence of words.



Sentiment Classification (Source: Author)

**3. Video Activity Recognition:** In video activity recognition, the model needs to identify the activity in a video clip. A video clip is a sequence of video frames, therefore in case of video activity recognition input is a sequence of data.

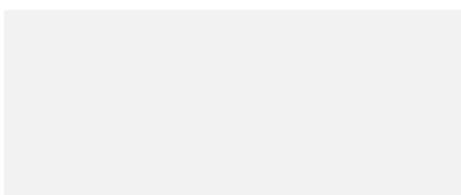


Video Activity Recognition (Source: Author)

These examples show that there are different applications of sequence models. Sometimes both the input and output are sequences, in some either the input or the output is a sequence. Recurrent neural network (RNN) is a popular sequence model that has shown efficient performance for sequential data.

## Recurrent Neural Networks (RNNs)

Recurrent Neural Network (RNN) is a Deep learning algorithm and it is a type of Artificial Neural Network architecture that is specialized for processing sequential data. RNNs are mostly used in the field of Natural Language Processing (NLP). RNN maintains internal memory, due to this they are very efficient for machine learning problems that involve sequential data. RNNs are also used in time series predictions as well.

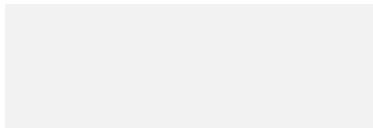


Additional RNN architecture (Source: Stanford.edu)

The main advantage of using RNNs instead of standard neural networks is that the features are not shared in standard neural networks. Weights are shared across time in RNN.

RNNs can remember its previous inputs but Standard Neural

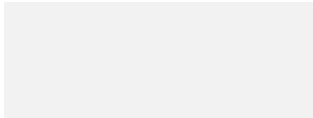
Networks are not capable of remembering previous inputs. RNN takes historical information for computation.



Loss function

In RNN loss function is defined based on the loss at each time step.

In RNN backpropagation is done at each point in time



The derivative of the Loss with respect to Weight

## RNN Architectures

There are several RNN architectures based on the

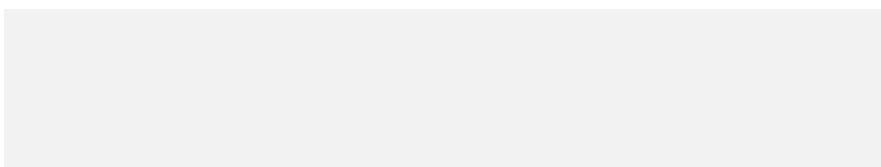
number of inputs and outputs,

1. One to Many Architecture: Image captioning is one good example of this architecture. In image captioning, it takes one image and then outputs a sequence of words. Here there is only one input but many outputs.

2. Many to One Architecture: Sentiment classification is one good example of this architecture. In sentiment classification, a given sentence is classified as positive or negative. In this case, the input is a sequence of words and output is a binary classification.

3. Many to Many Architecture: There are two cases in many to many architectures,

- *The first type* is when the input length equals to the output length. Name entity recognition is one good example where the number of words in the input sequence is equal to the number of words in the output sequence.
- *The second type* of many to many architecture is when input length does not equal to the output length. Machine translation is one good scenario for this architecture. In machine translation, RNN reads a sentence in one language and then converts it to another language. Here input length and output length are different.





RNN architectures (Source: <https://calvinfeng.gitbook.io>)

## Long Short-Term Memory (LSTM)

Traditional RNNs are not good at capturing long-range dependencies. This is mainly due to the vanishing gradient problem. When training very deep network gradients or the derivatives decreases exponentially as it propagates down the layers. This is known as Vanishing Gradient Problem. These gradients are used to update the weights of neural networks. When the gradients vanish then the weights will not be updated. Sometimes it will completely stop the neural network from training. This vanishing gradient problem is a common issue in very deep neural networks.

To overcome this vanishing gradient problem in RNNs, Long Short-Term Memory was introduced by Sepp Hochreiter and Juergen Schmidhuber. LSTM is a modification to the RNN hidden layer. LSTM has enabled RNNs to remember its inputs over a long period of time. In LSTM in addition to the hidden state, a cell state is passed to the next time step.



Internal structure of basic RNN and LSTM unit (Source: stanford.edu)

LSTM can capture long-range dependencies. It can have memory about previous inputs for extended time durations. There are 3 gates in an LSTM cell. Memory manipulations in LSTM are done using these gates. Long short-term memory (LSTM) utilizes gates to control the gradient propagation in the recurrent network's memory.

- Forget Gate: Forget gate removes the information that is no longer useful in the cell state
- Input Gate: Additional useful information to the cell state is added by input gate
- Output Gate: Additional useful information to the cell state is added by output gate

This gating mechanism of LSTM has allowed the network to learn the conditions for when to forget, ignore, or keep information in the memory cell.

LSTM is a very popular deep learning algorithm for sequence models. Apple's Siri and Google's voice search are some real-world examples that have used the LSTM algorithm and it is behind the success of those applications. Recent research has shown how the LSTM algorithm can improve the performance of the machine learning model. LSTM is also used for time-series predictions and text classification tasks as well.

## References

[https://www.researchgate.net/publication/13853244\\_Long\\_Short-term\\_Memory](https://www.researchgate.net/publication/13853244_Long_Short-term_Memory)

<https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks>



22



Artificial  
Intelligence

Deep Learning

Recurrent Neural  
Network

Long Short Term Memory

NLP

## More from Towards Data Science

Follow

Your home for data science. A Medium publication sharing concepts, ideas and codes.

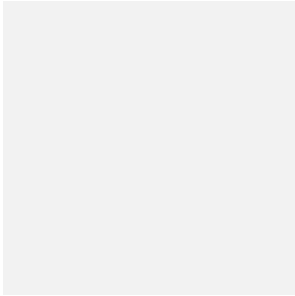
Read more from Towards Data Science

## More From Medium

---

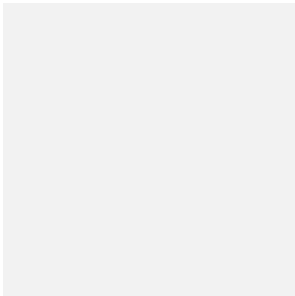
### Event driven bars

Denny Joseph



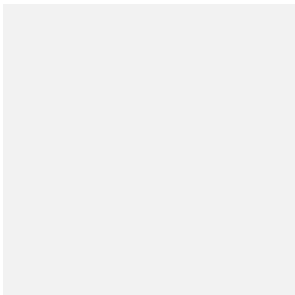
### My intro with Image Recognition at Couture.ai

Nandan Meda in couture-ai



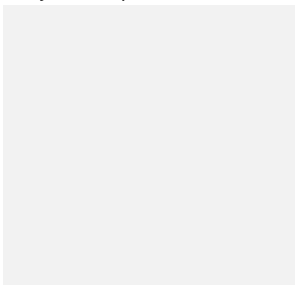
### Principal Component Analysis for Logistic Regression with scikit-learn

Александра Пухова



### CycleGan for Season Transfer

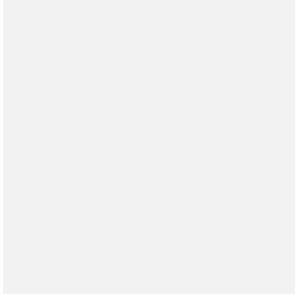
Viswajith Karapoondi Nott



---

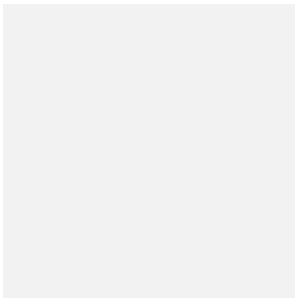
## Using Forward-search algorithms to solve AI Planning Problems

Debby Nirwan in Artificial Intelligence in Plain English



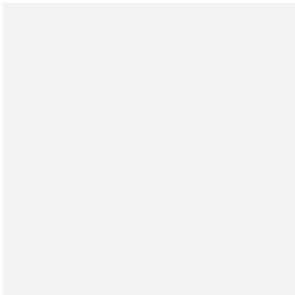
## Introduction to Weights&Biases

Marco Varrone in Polimi Data Scientists



## AI-Based Visual Inspection For Defect Detection

MobiDev



## Using FastAI's ULMFiT to make a state-of-the-art multi-class text classifier

Aditya Chinchure in Technonerds

