Implementing Recurrent Neural Networks with Keras

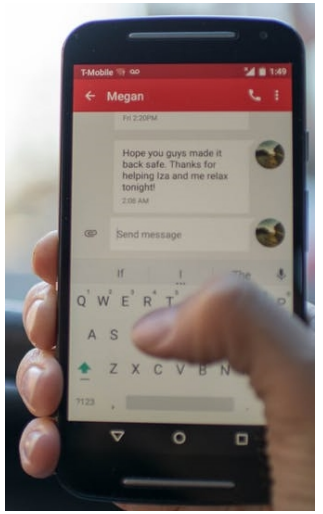
Jerry Kurata
CONSULTANT

@jerrykur

Sequential Dependence

Stock Prices

Weather



Text Input

Traditional NN Issues

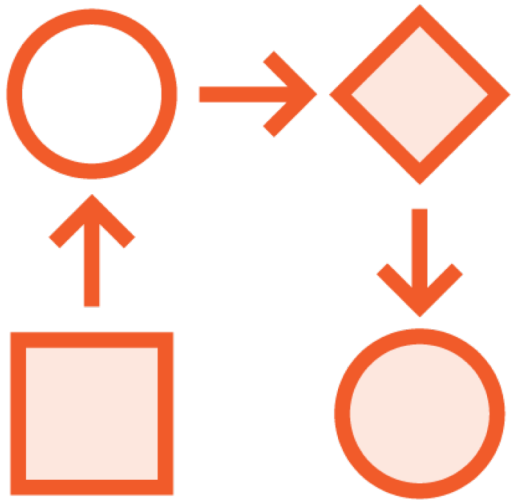
No tracking of data sequences

Do not remember previous values

RNNs address these issues

RNN Features

Sequential information



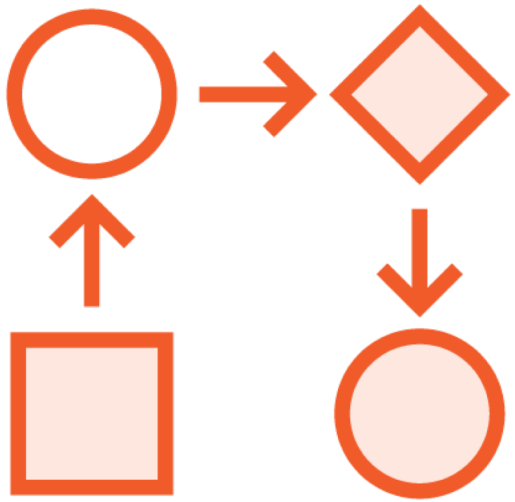
Same computations

Date	Price
1/15	145.25

Current

RNN Features

Sequential information

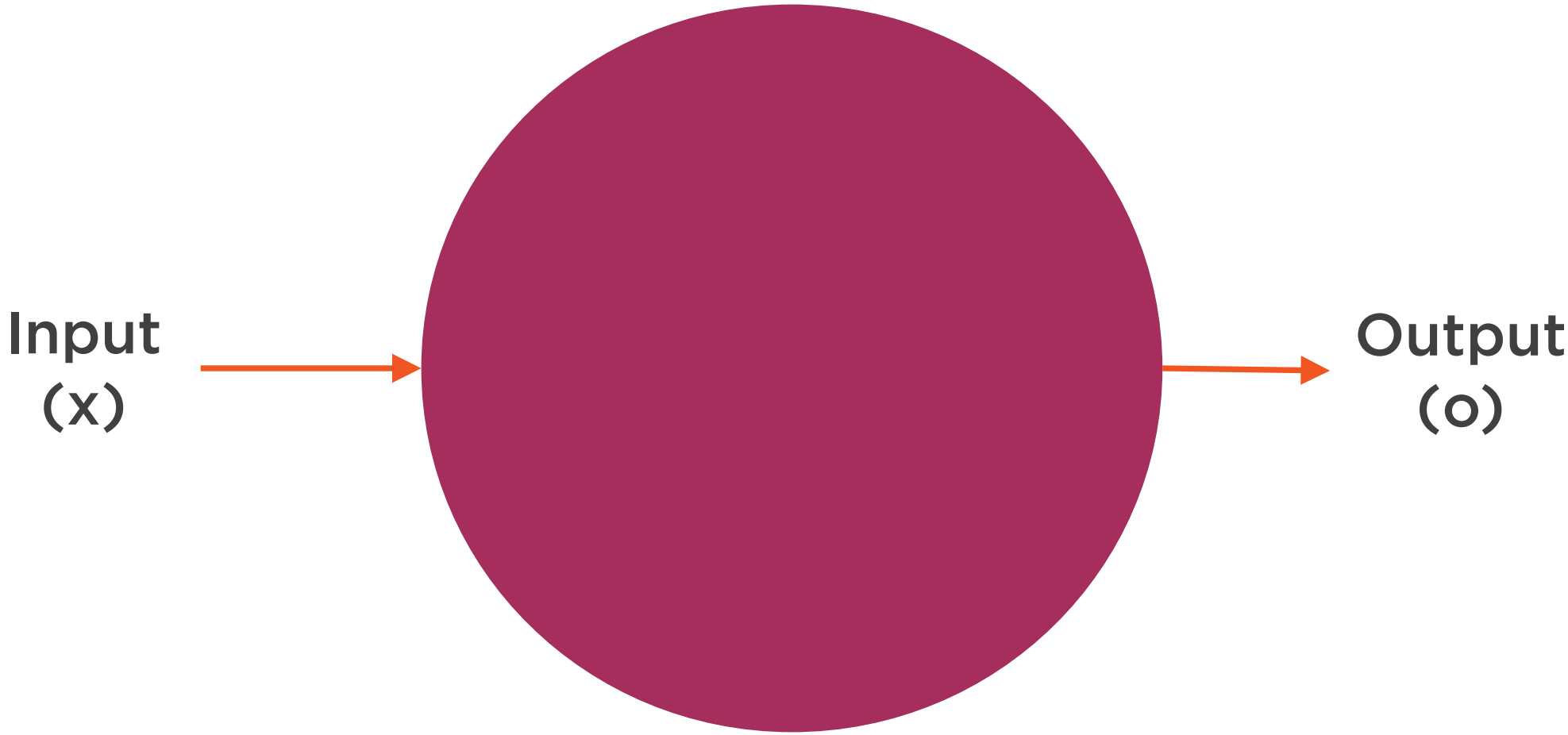


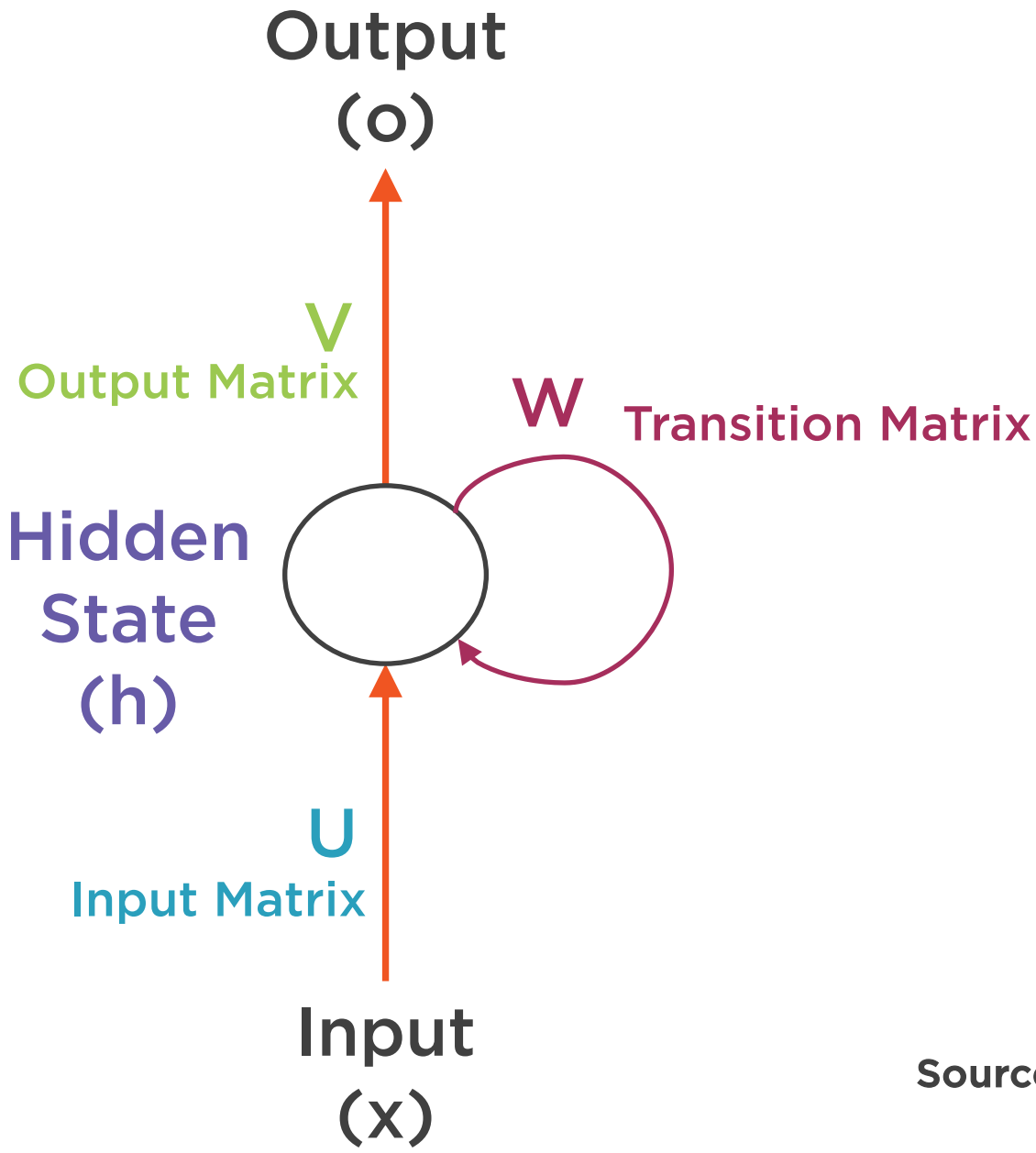
Same computations

Date	Price
1/15	145.25

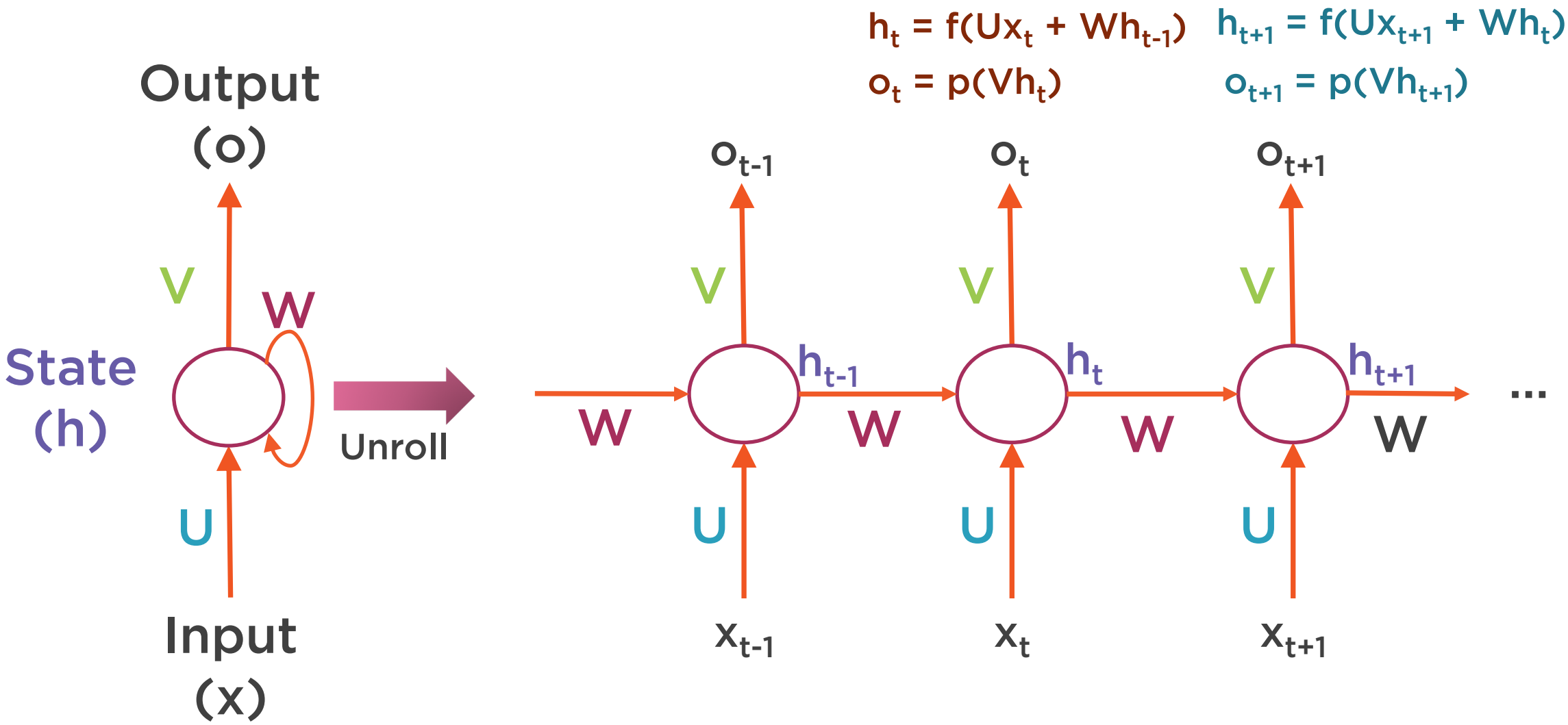
Date	Price
1/12	138.50
1/13	152.00
1/14	147.50

Uses current and previous data


Input
(x)
Output
(o)



Source: Nature



Source: Nature

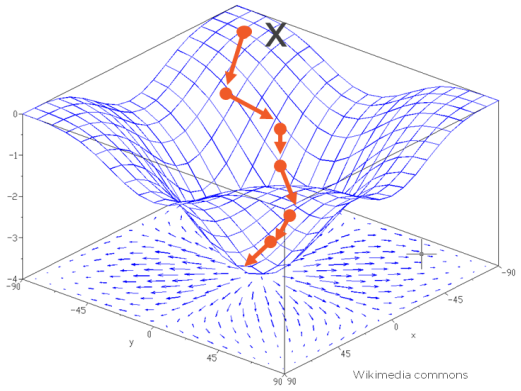
Simple RNN

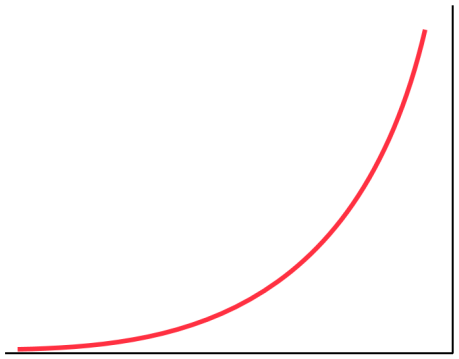
Keras simpleRNN layer

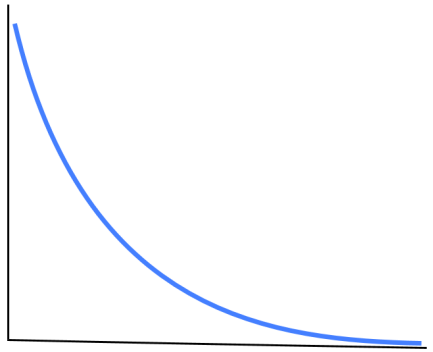
Implements forwarding of information

Sometimes hard to train

RNN Training Issues







LSTM

GRU

Back Propagation issues

Exploding Gradient

Vanishing Gradient

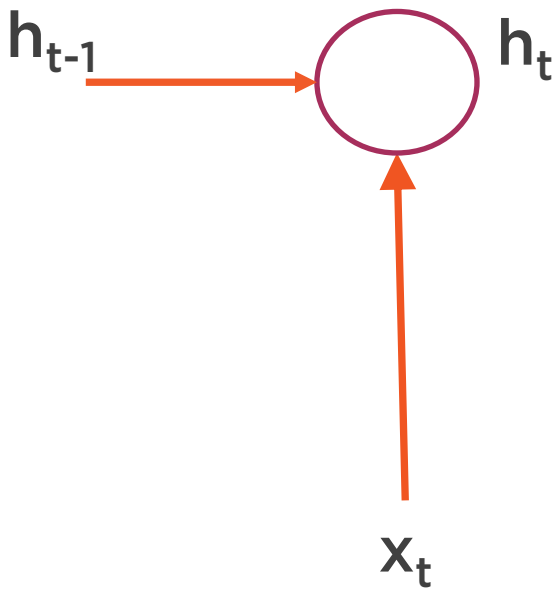
Fix training issues

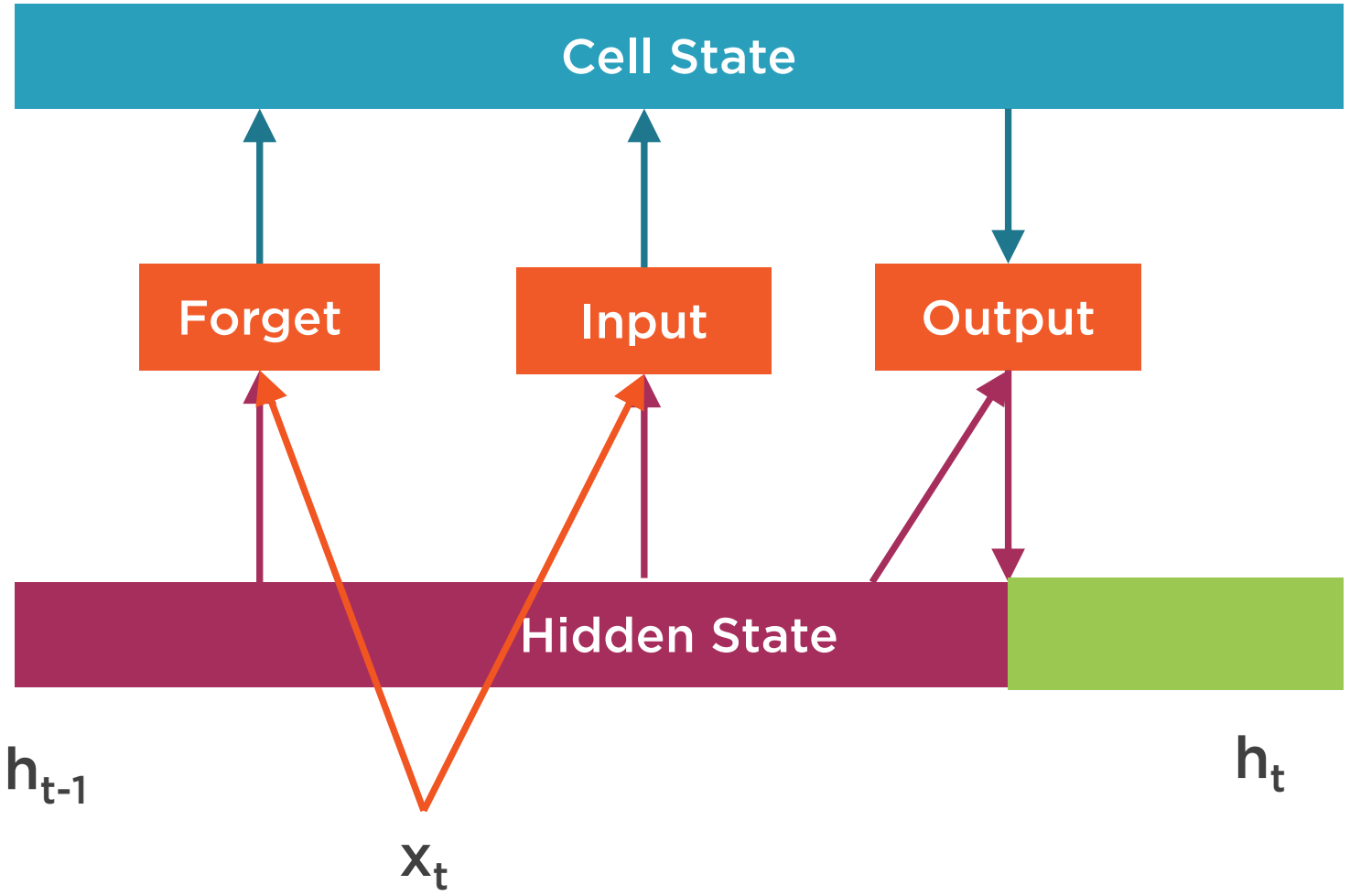
LSTMs and GRUs

Cell State memory

Accessed via Gates

Cell State





For more information see:
http://colah.github.io/posts/2015-08-Understanding-LSTMs

Gated Recurrent Unit (GRU)

Simplifies LSTM

Forget + Input = Update

Merges Cell State and Hidden State

Keras Recurrent Neural Network Layers

RNN Type	Keras Layer
Standard RNN	SimpleRNN
Long Short Term Memory(LSTM)	LSTM
Gated Recurrent Unit (GRU)	GRU
CNN + LSTM	ConvLSTM2D
CUDA DNN Support	CuDNNGRU
	CuDNNLSTM

Textual Data

Text is everywhere

Determining meaning is hard

Restaurant Review

"I really love this place. The food was passible and the service was glacial. But the lights are pretty and tables cloths impressive."

Customer 72

Demo

Sentiment Classifier

IMDB review data

Summary

- Sequential data depends on previous data
- RNNs handle sequential dependencies
- Classic RNNs have issues
- LSTM/GRU handles issues
- Performed Sentiment Analysis with LSTM