

# — Recurrent Neural Networks (RNNs)

# Agenda



Introduction to RNNs



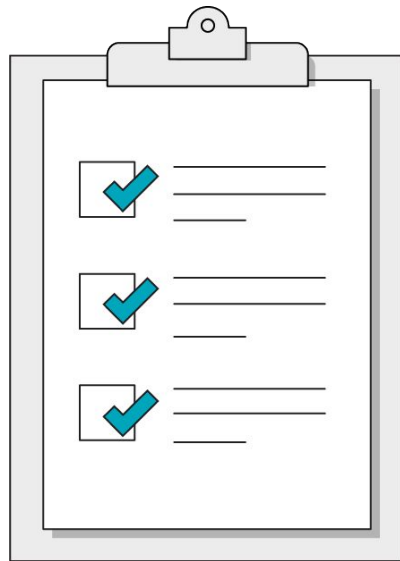
Case Study: Using RNNs to predict stock prices



RNN in Python

# Learning Objectives

- Preprocess sequence data for RNN modeling
- Design, train and evaluate an RNN model
- Create a model that predicts the price of Apple's (AAPL) stock



RNNs



# Introduction to RNNs



# Why do we need RNNs?

Let's start with an example. Suppose I asked you to tell me which direction this ball would move next?

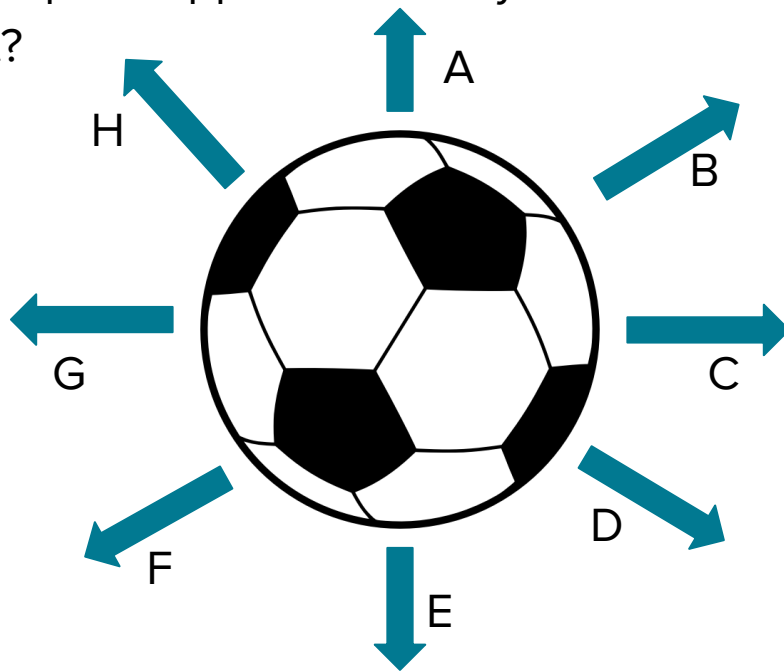
# Why do we need RNNs?

Let's start with an example. Suppose I asked you to tell me which direction this ball would move next?



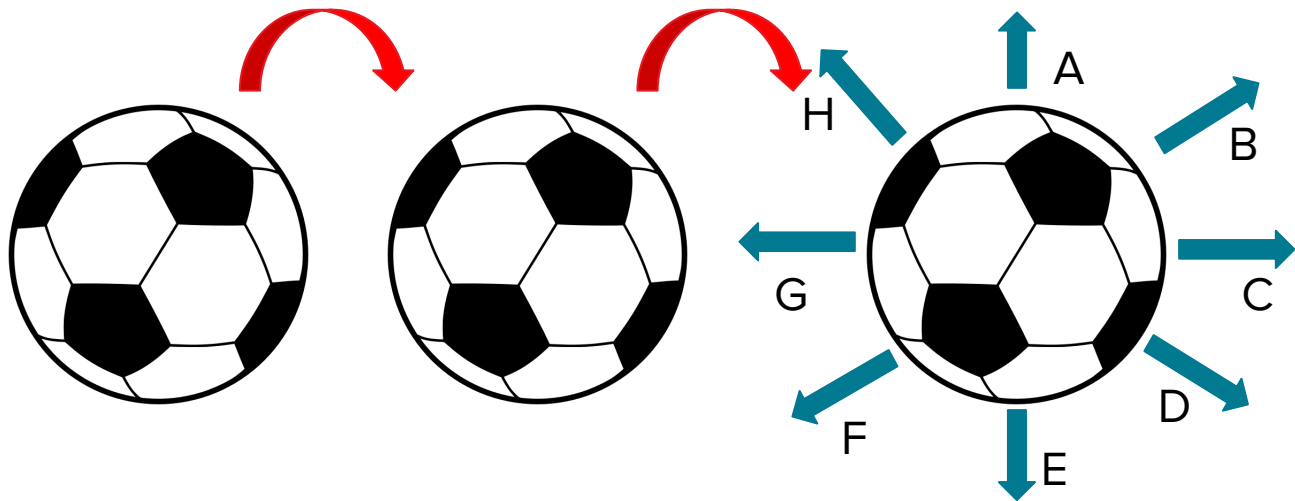
# Why do we need RNNs?

Let's start with an example. Suppose I asked you to tell me which direction this ball would move next?



# Why do we need RNNs?

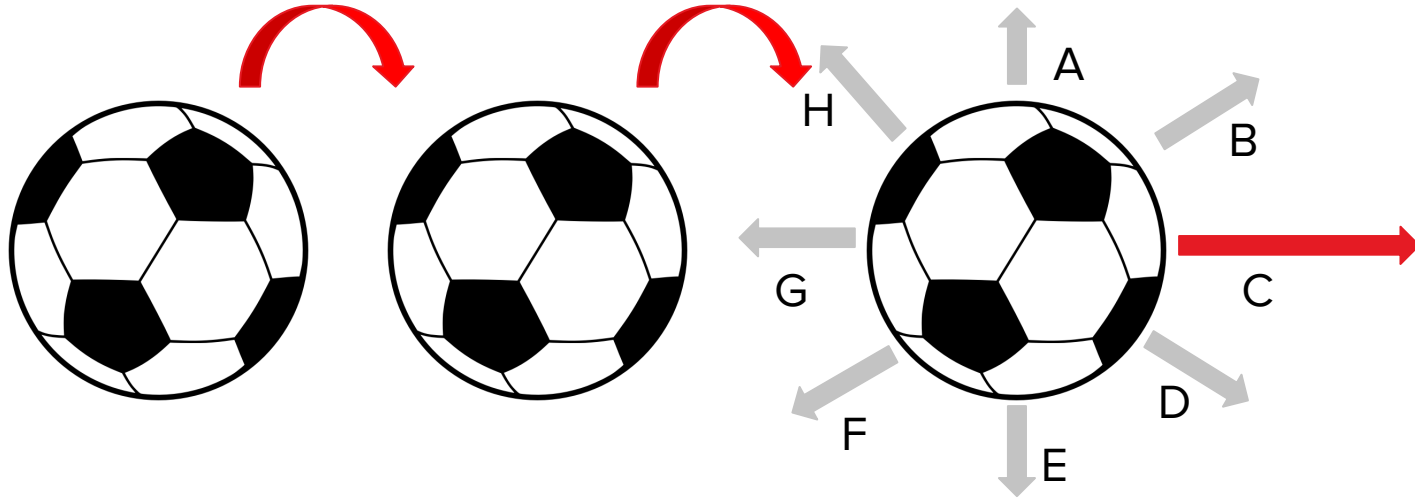
Let's start with an example. Suppose I asked you to tell me which direction this ball would move next?





# Why do we need RNNs?

Let's start with an example. Suppose I asked you to tell me which direction this ball would move next?



# Sequence Data

RNNs are used on sequential data.

What are some examples of sequential data?



# Sequence Data

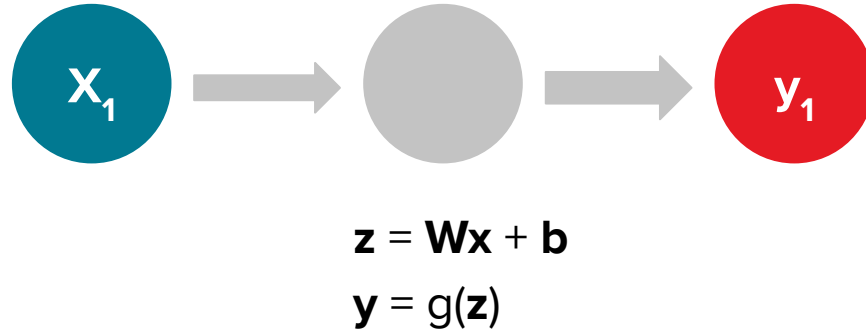
RNNs are used on sequential data.

What are some examples of sequential data?

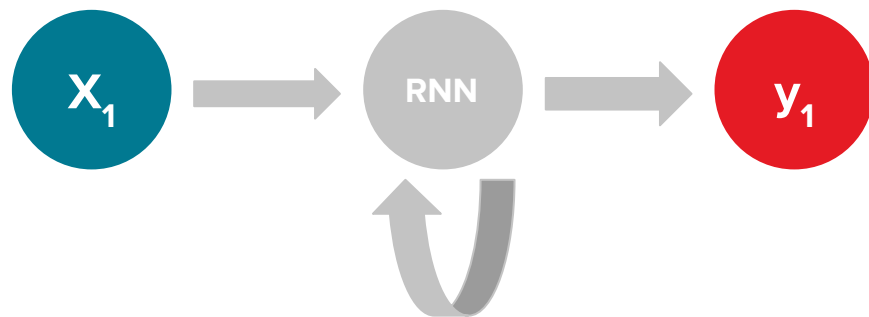
- Language/Text
- Timeseries
- Sound



# Reminder: Standard (Feed Forward) Neural Networks



# RNNs



$$z = \mathbf{W}_H \mathbf{h}_{t-1} + \mathbf{W}_x \mathbf{x} + \mathbf{b}$$
$$\mathbf{y} = g(\mathbf{z})$$

RNNs



# Case Study: using RNNs to predict stock prices



# Keras TimeseriesGenerator

	<b>Volume</b>	<b>Open</b>	<b>Close</b>	<b>Target</b>
2019-01-01	1,000,000	\$150.00	\$155.00	NaN
2019-01-02	2,000,000	\$160.00	\$165.00	1
2019-01-03	3,000,000	\$155.00	\$160.00	0
2019-01-04	4,000,000	\$180.00	\$185.00	1
2019-01-05	5,000,000	\$175.00	\$180.00	0
2019-01-06	6,000,000	\$170.00	\$175.00	0

## Keras TimeseriesGenerator: Length = 3

	Volume	Open	Close	Target
2019-01-01	1,000,000	\$150.00	\$155.00	NaN
2019-01-02	2,000,000	\$160.00	\$165.00	1
2019-01-03	3,000,000	\$155.00	\$160.00	0
2019-01-04	4,000,000	\$180.00	\$185.00	1
2019-01-05	5,000,000	\$175.00	\$180.00	0
2019-01-06	6,000,000	\$170.00	\$175.00	0



## Keras TimeseriesGenerator: Length = 3

	Volume	Open	Close	Target
2019-01-01	1,000,000	\$150.00	\$155.00	NaN
2019-01-02	2,000,000	\$160.00	\$165.00	1
2019-01-03	3,000,000	\$155.00	\$160.00	0
2019-01-04	4,000,000	\$180.00	\$185.00	1
2019-01-05	5,000,000	\$175.00	\$180.00	0
2019-01-06	6,000,000	\$170.00	\$175.00	0

## Keras TimeseriesGenerator: Length = 3

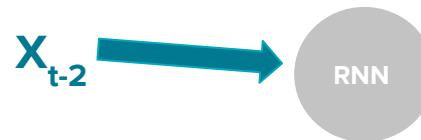
	Volume	Open	Close	Target
2019-01-01	1,000,000	\$150.00	\$155.00	NaN
2019-01-02	2,000,000	\$160.00	\$165.00	1
2019-01-03	3,000,000	\$155.00	\$160.00	0
2019-01-04	4,000,000	\$180.00	\$185.00	1
2019-01-05	5,000,000	\$175.00	\$180.00	0
2019-01-06	6,000,000	\$170.00	\$175.00	0

## Keras TimeseriesGenerator: Length = 4

	Volume	Open	Close	Target
2019-01-01	1,000,000	\$150.00	\$155.00	NaN
2019-01-02	2,000,000	\$160.00	\$165.00	1
2019-01-03	3,000,000	\$155.00	\$160.00	0
2019-01-04	4,000,000	\$180.00	\$185.00	1
2019-01-05	5,000,000	\$175.00	\$180.00	0
2019-01-06	6,000,000	\$170.00	\$175.00	0

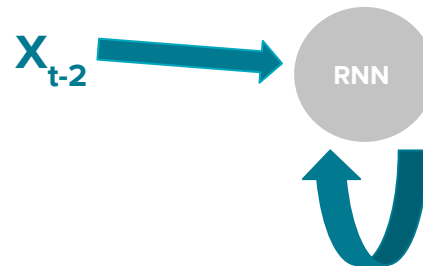
# RNN Architecture

	Volume	Open	Close
2019-01-01	1,000,000	\$150.00	\$155.00
2019-01-02	2,000,000	\$160.00	\$165.00
2019-01-03	3,000,000	\$155.00	\$160.00



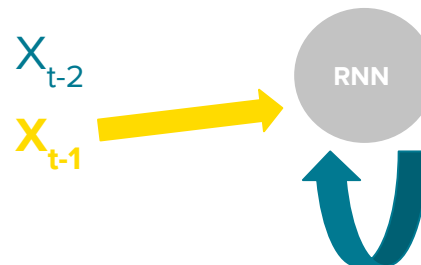
# RNN Architecture

	Volume	Open	Close
2019-01-01	1,000,000	\$150.00	\$155.00
2019-01-02	2,000,000	\$160.00	\$165.00
2019-01-03	3,000,000	\$155.00	\$160.00



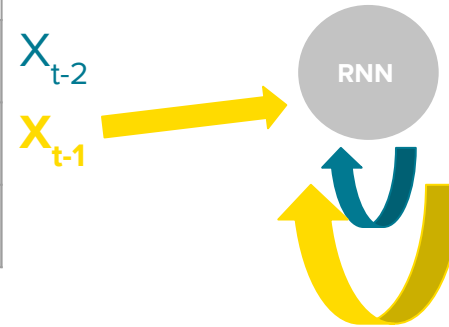
# RNN Architecture

	Volume	Open	Close
2019-01-01	1,000,000	\$150.00	\$155.00
2019-01-02	2,000,000	\$160.00	\$165.00
2019-01-03	3,000,000	\$155.00	\$160.00



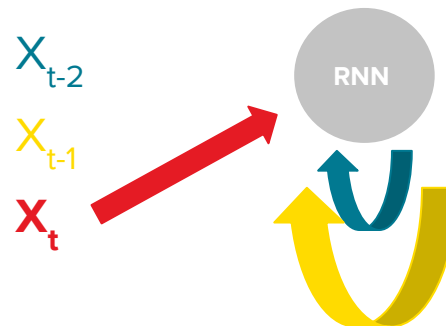
# RNN Architecture

	Volume	Open	Close
2019-01-01	1,000,000	\$150.00	\$155.00
2019-01-02	2,000,000	\$160.00	\$165.00
2019-01-03	3,000,000	\$155.00	\$160.00



# RNN Architecture

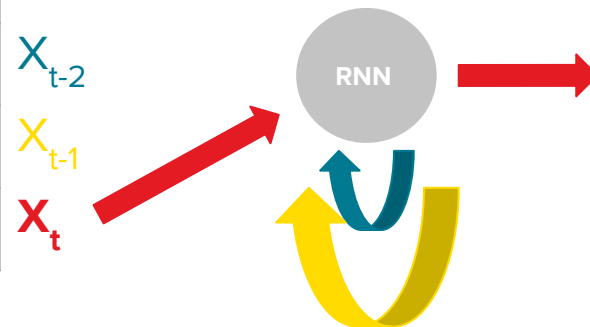
	Volume	Open	Close
2019-01-01	1,000,000	\$150.00	\$155.00
2019-01-02	2,000,000	\$160.00	\$165.00
2019-01-03	3,000,000	\$155.00	\$160.00





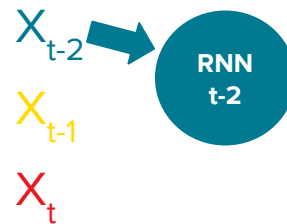
# RNN Architecture

	Volume	Open	Close
2019-01-01	1,000,000	\$150.00	\$155.00
2019-01-02	2,000,000	\$160.00	\$165.00
2019-01-03	3,000,000	\$155.00	\$160.00



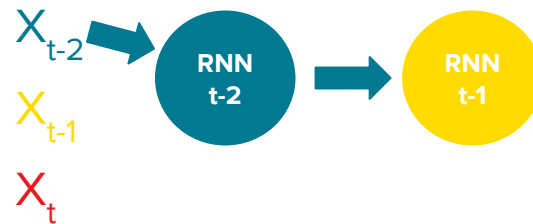
# RNN Architecture: Unfolded

	Volume	Open	Close
2019-01-01	1,000,000	\$150.00	\$155.00
2019-01-02	2,000,000	\$160.00	\$165.00
2019-01-03	3,000,000	\$155.00	\$160.00



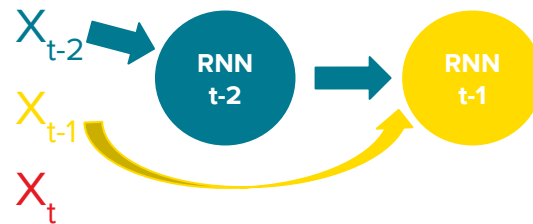
# RNN Architecture: Unfolded

	Volume	Open	Close
2019-01-01	1,000,000	\$150.00	\$155.00
2019-01-02	2,000,000	\$160.00	\$165.00
2019-01-03	3,000,000	\$155.00	\$160.00



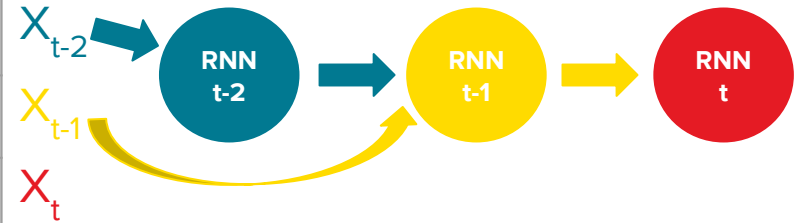
# RNN Architecture: Unfolded

	Volume	Open	Close
2019-01-01	1,000,000	\$150.00	\$155.00
2019-01-02	2,000,000	\$160.00	\$165.00
2019-01-03	3,000,000	\$155.00	\$160.00



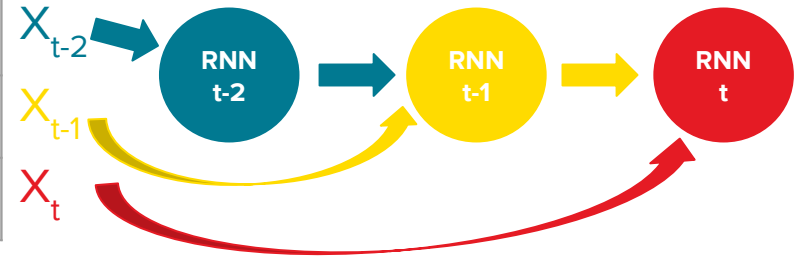
# RNN Architecture: Unfolded

	Volume	Open	Close
2019-01-01	1,000,000	\$150.00	\$155.00
2019-01-02	2,000,000	\$160.00	\$165.00
2019-01-03	3,000,000	\$155.00	\$160.00



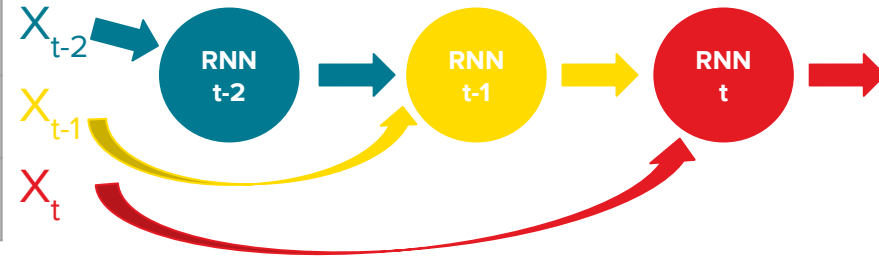
# RNN Architecture: Unfolded

	Volume	Open	Close
2019-01-01	1,000,000	\$150.00	\$155.00
2019-01-02	2,000,000	\$160.00	\$165.00
2019-01-03	3,000,000	\$155.00	\$160.00



# RNN Architecture: Unfolded

	Volume	Open	Close
2019-01-01	1,000,000	\$150.00	\$155.00
2019-01-02	2,000,000	\$160.00	\$165.00
2019-01-03	3,000,000	\$155.00	\$160.00





# A problem with RNNs: Vanishing Gradients





# A problem with ~~RNNs~~-NNs: Vanishing Gradients



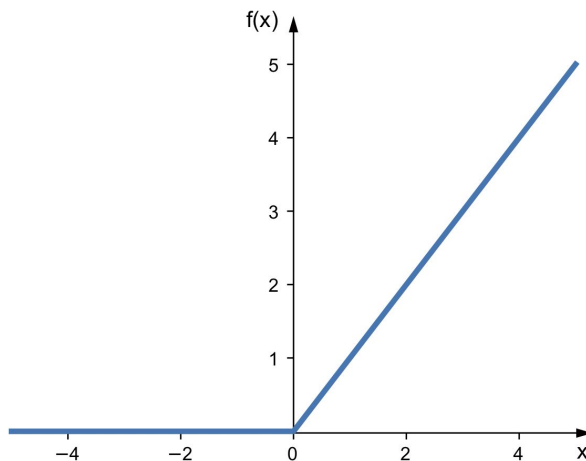
# A problem with ~~RNNs~~-NNs: Vanishing Gradients

- When the gradients are too small - when you multiply a bunch of small gradients together you just end up with smaller and smaller gradients as you backpropagate
- The weights and biases of the earlier layers will not be effectively updated and can lead to an inaccurate model



# A solution for NNs: ReLU

- One of the reasons ReLU is so effective is that it doesn't cause a small derivative
  - Convince yourself of this: what can the derivative of the ReLU function be?



# A solution for RNNs: LSTMs and GRUs

- Long Short-Term Memory (LSTM)
- Gated Recurrent Unit (GRU)

---

# Let's try it out!



