# Recurrent Neural Networks

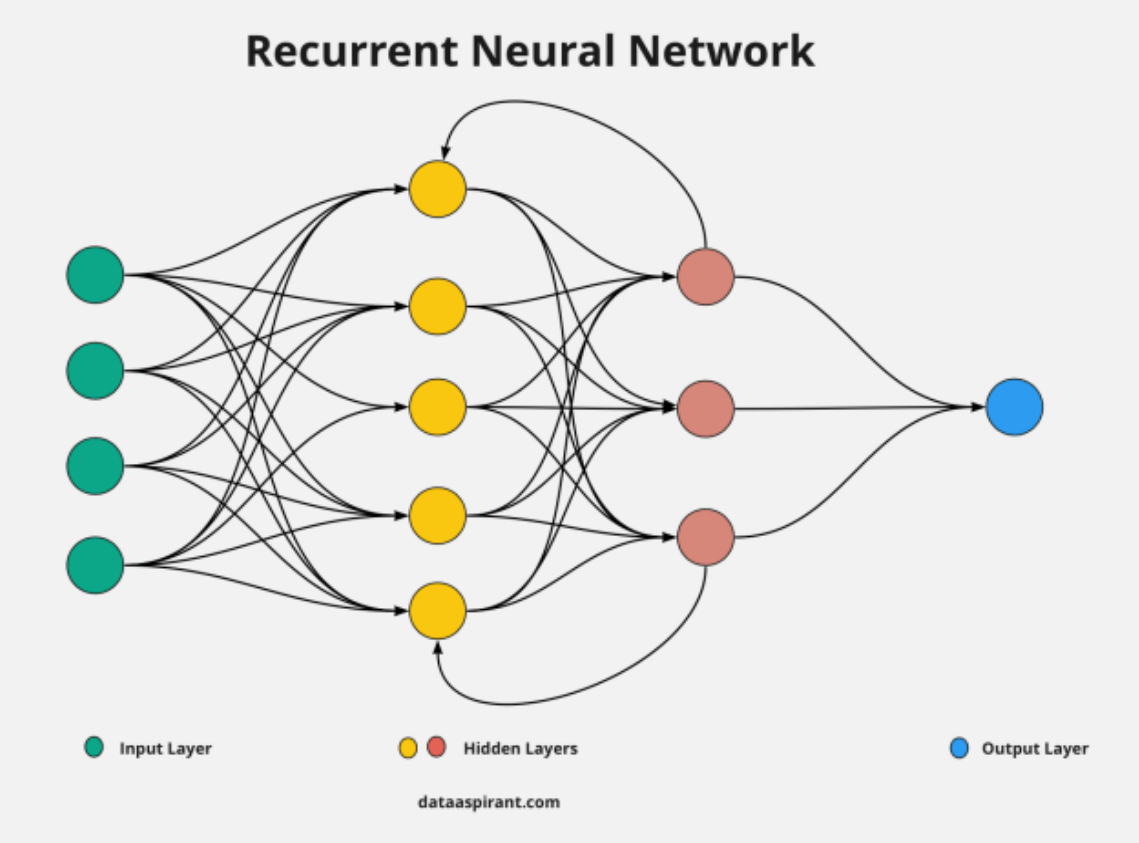
COSC 410: Applied Machine Learning

Spring 2022

Prof. Apthorpe

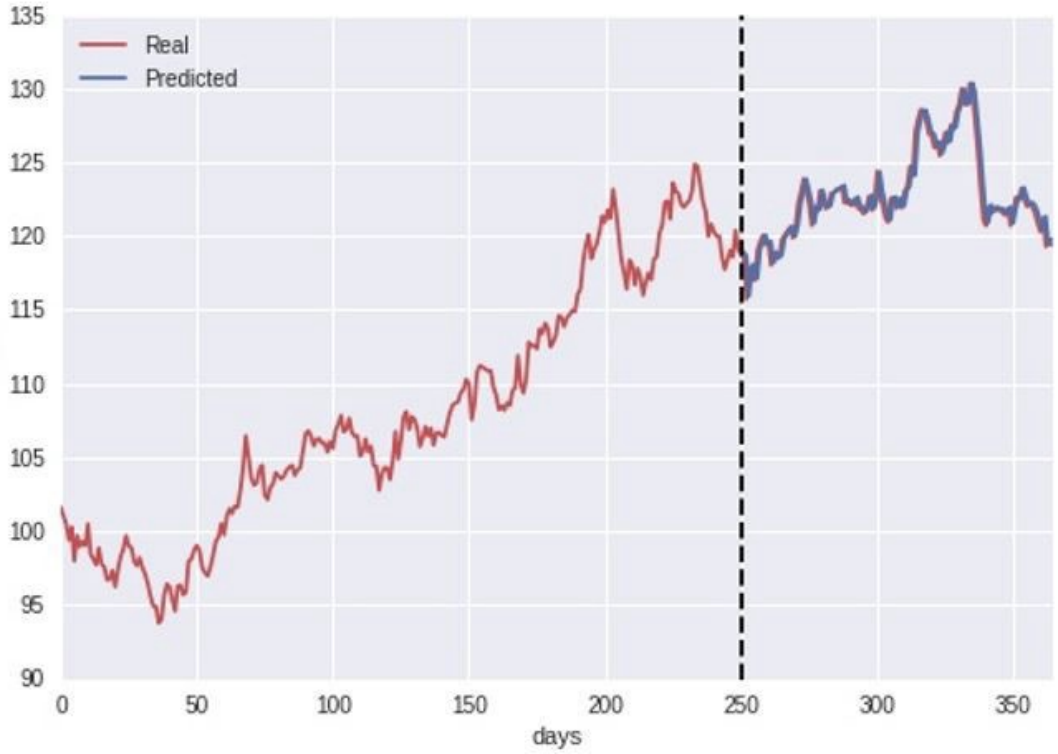
# Outline

- Motivating Tasks
- Challenges with FNNs & CNNs
- Recurrent Network Predictions
- Why Recurrent Connections?
- Training RNNs
- Deep RNNs
- RNN Memory



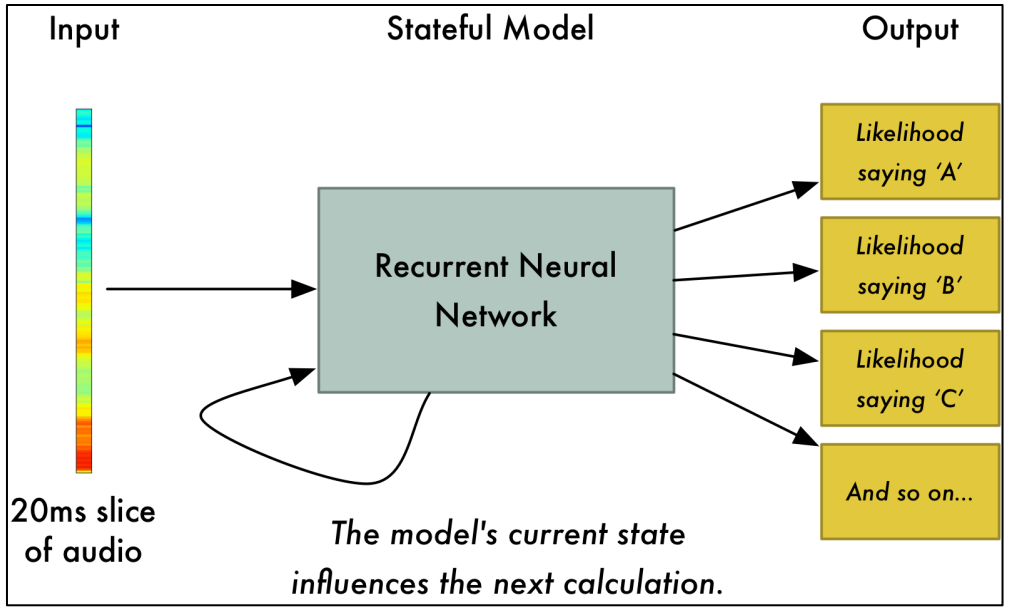
# Motivating Tasks

- **Predicting sequences**
  - Predict future elements (forecasting)
    - Commodity prices, vehicle trajectories, text generation, …



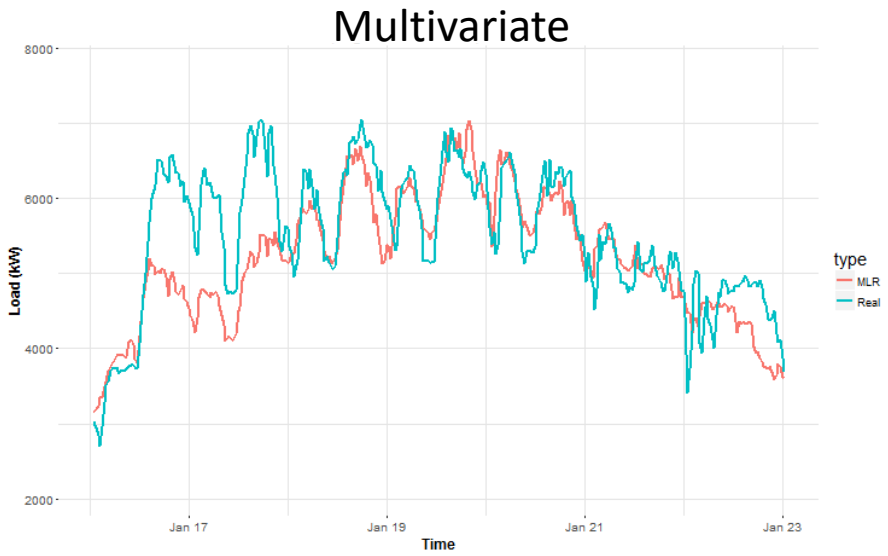
# Motivating Tasks

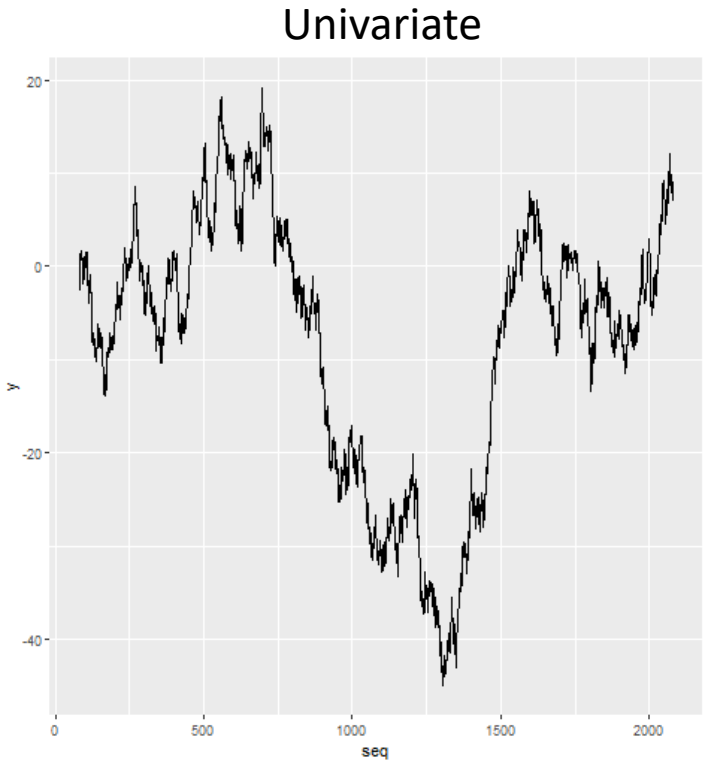
- **Predicting sequences**
  - Predict future elements (forecasting)
    - Commodity prices, vehicle trajectories, text generation, …
  - Label current state based on past
    - Speech recognition, dynamic hardware control, anomaly detection, …



# Sequence Data

- Input sequences varieties
  - **Univariate:** one value per step
  - **Multivariate:** multiple values per step

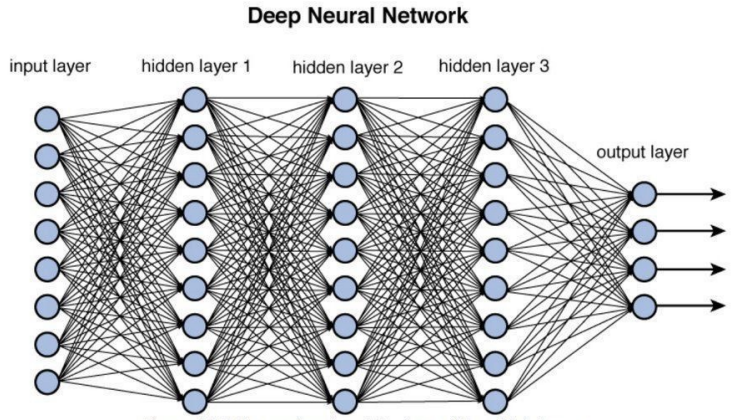


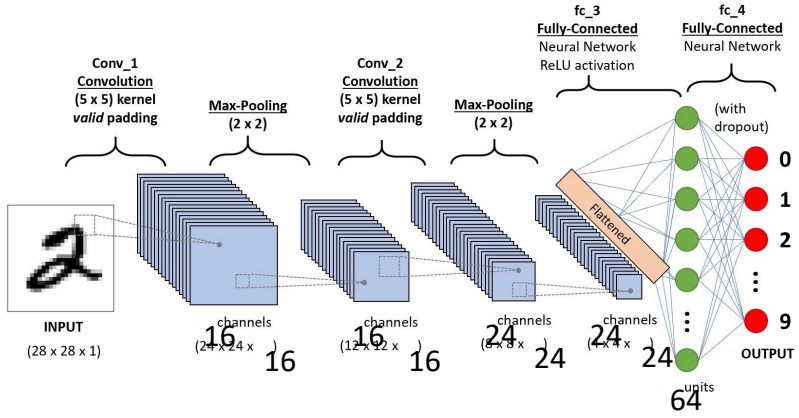

What are some examples of each?

# Challenges with FNNs & CNNs

- **Expect fixed-size inputs**
  - All examples must have same number of features

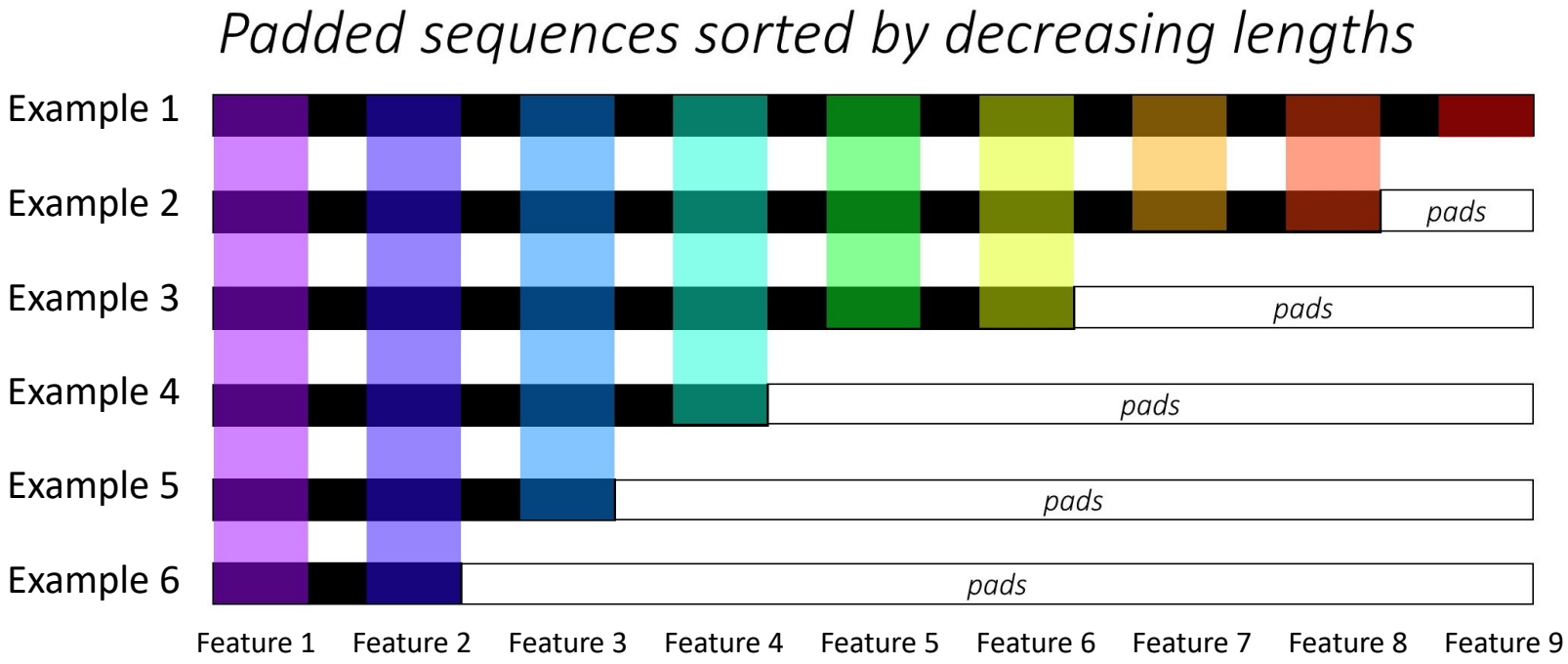



- BUT…real-world sequences can be **arbitrary lengths**
  - Natural language text documents vary in word count…
  - One user may say “Alexa” faster or slower than others…



# Challenges with FNNs & CNNs

- *Option:* Pad all sequences to same length
  - Pros & cons? When might this be good idea? When might this be a bad idea?



# Recurrent Network (RNN) Approach

- Process sequence data **one step at a time**

- Make one prediction per step

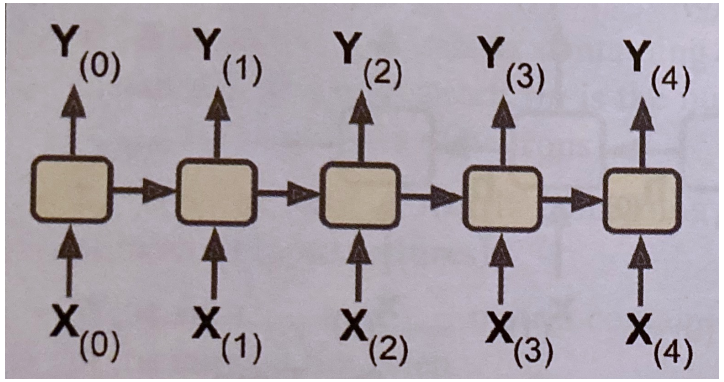
```
for step in sequence:
    ...
```

  - Rather than one prediction per example
  - If you only need one prediction per example, use the one from the last step

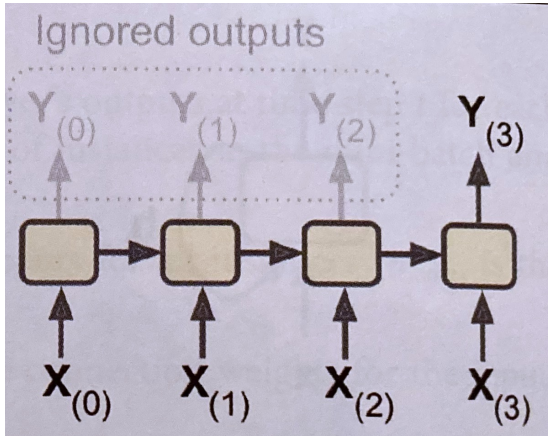
- Works for examples with any number of steps

# RNN Prediction Modes

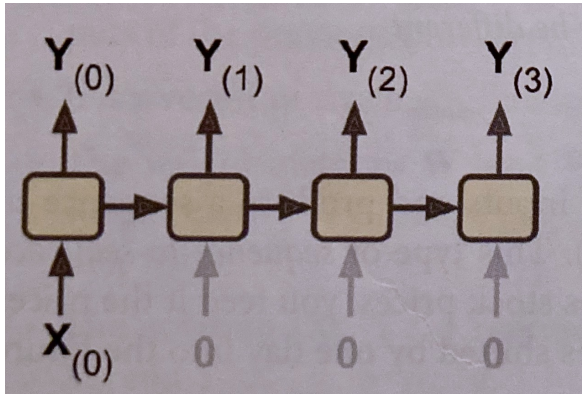
Sequence to Sequence




Sequence to Vector




Vector to Sequence

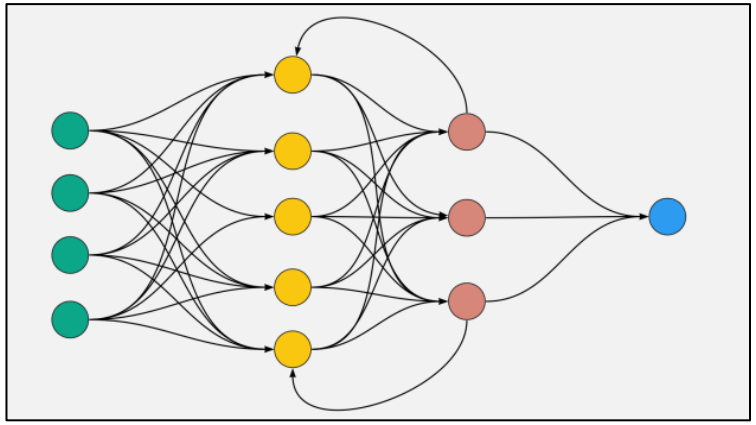


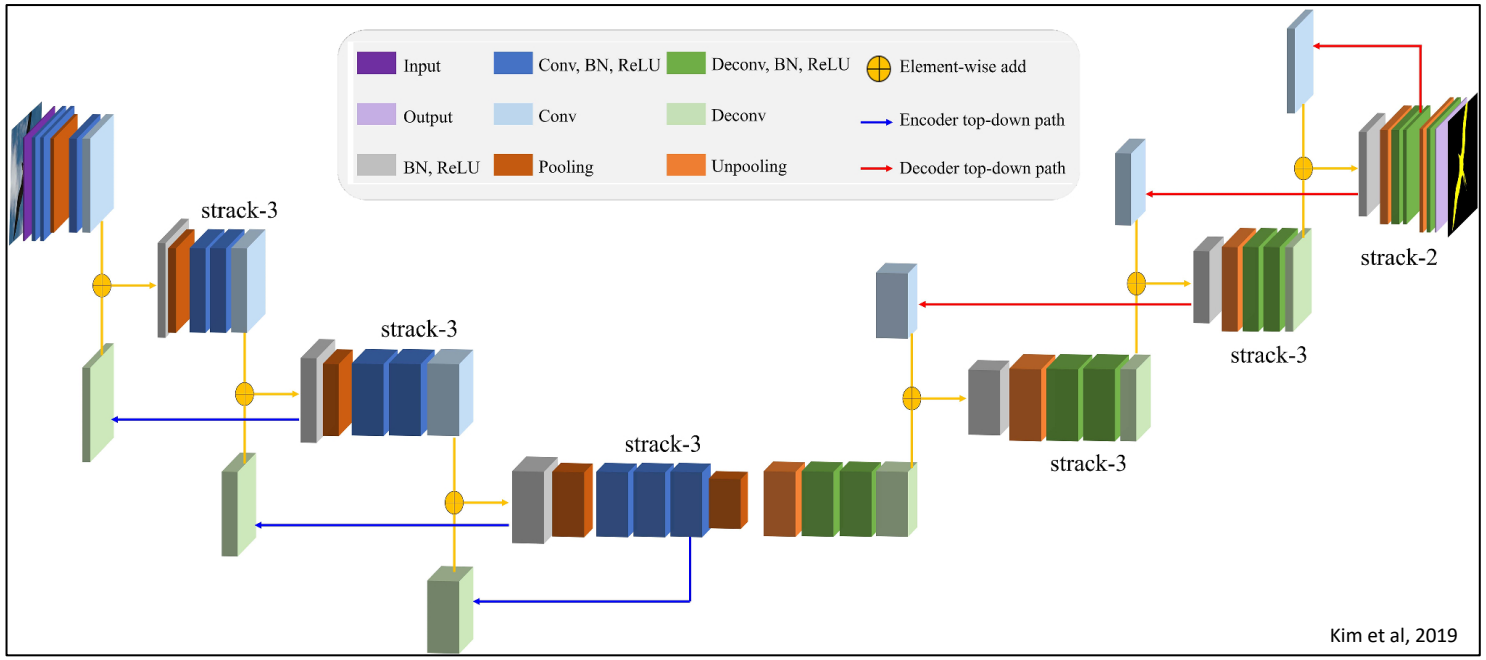

What are some examples of each?

# Recurrent Connections

- FNN and CNN connections go **forward** through the network

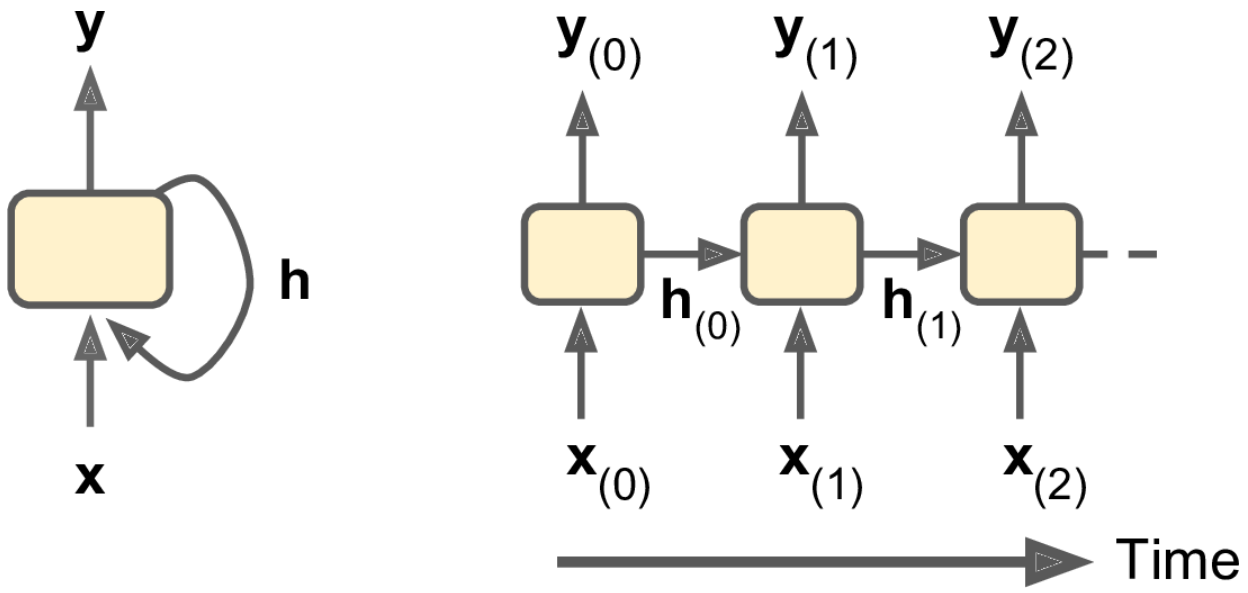
★RNNs have connections **backward** to the same or earlier layers





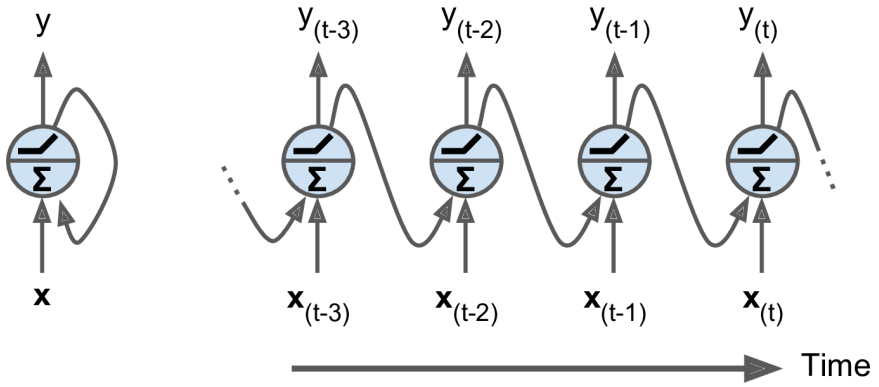
# Why Recurrent Connections

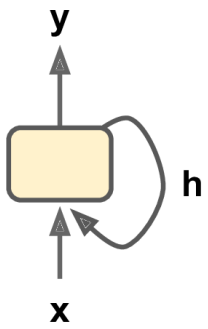
- Recurrent connections provide a network with **memory**
  - The output at step *t* depends on the input at times [*t*-*n*, *t*]
  - Why?

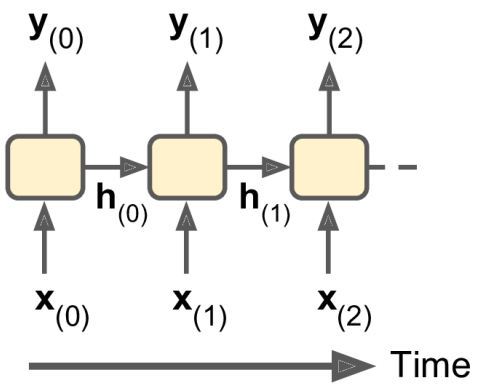


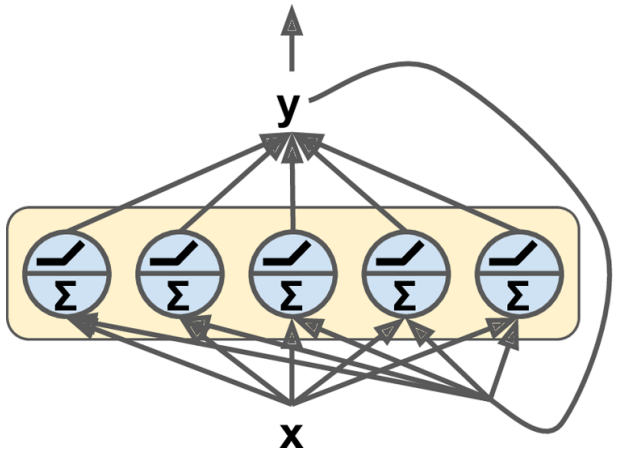
# Training RNNs

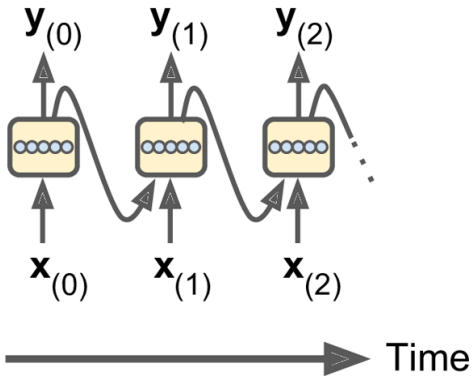
- Recurrent connections can be **unrolled in time**



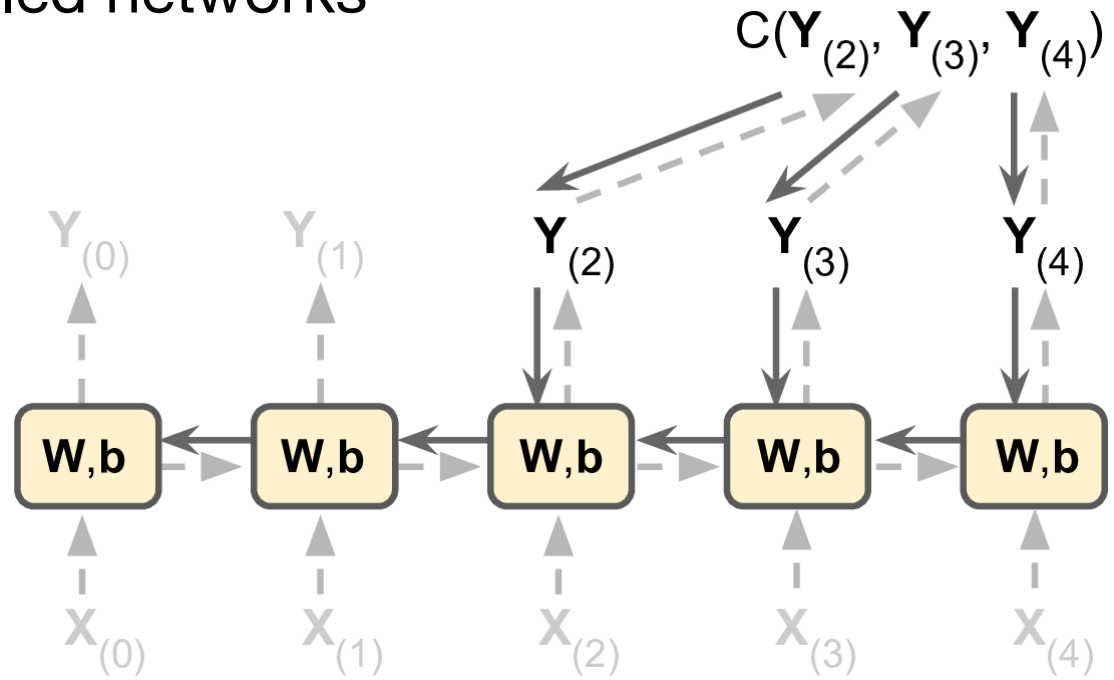



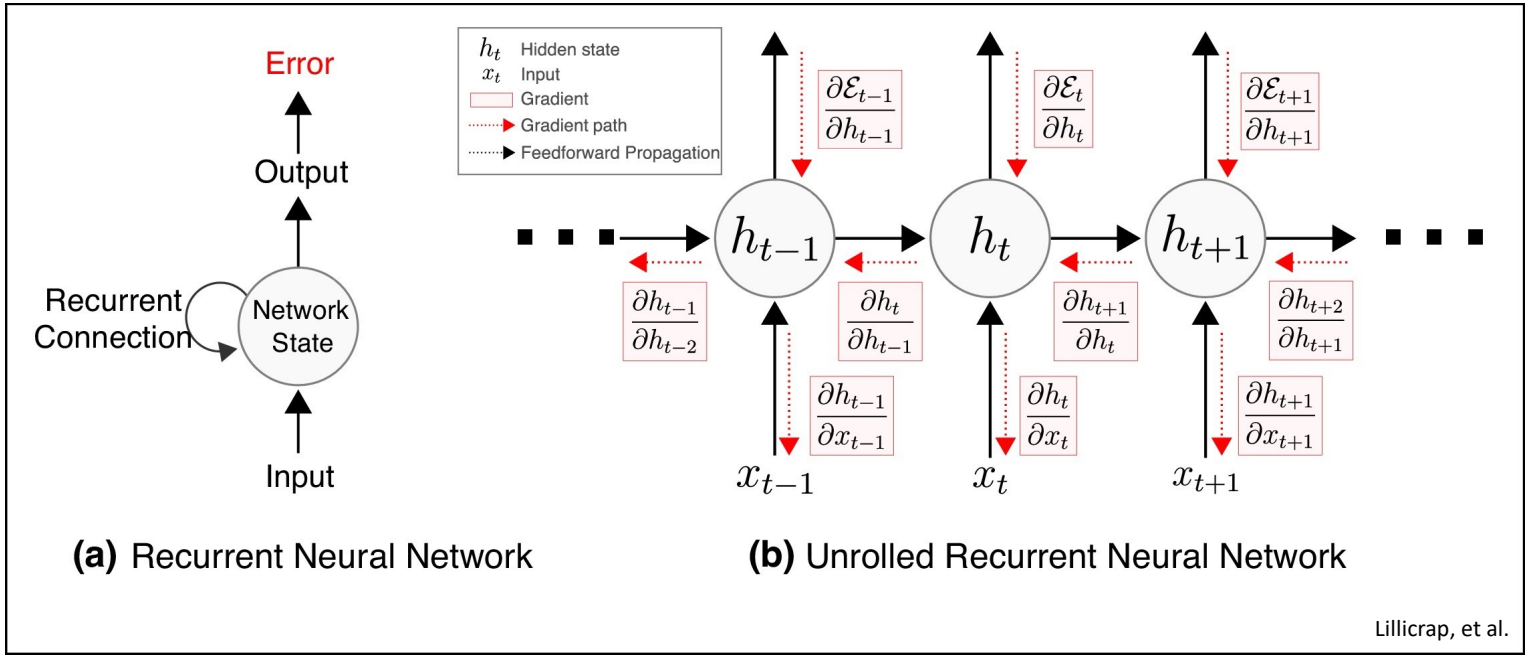
# Training RNNs

- **Backpropagation through time**
  - Compute weight & bias gradients by propagating errors backward through unrolled networks



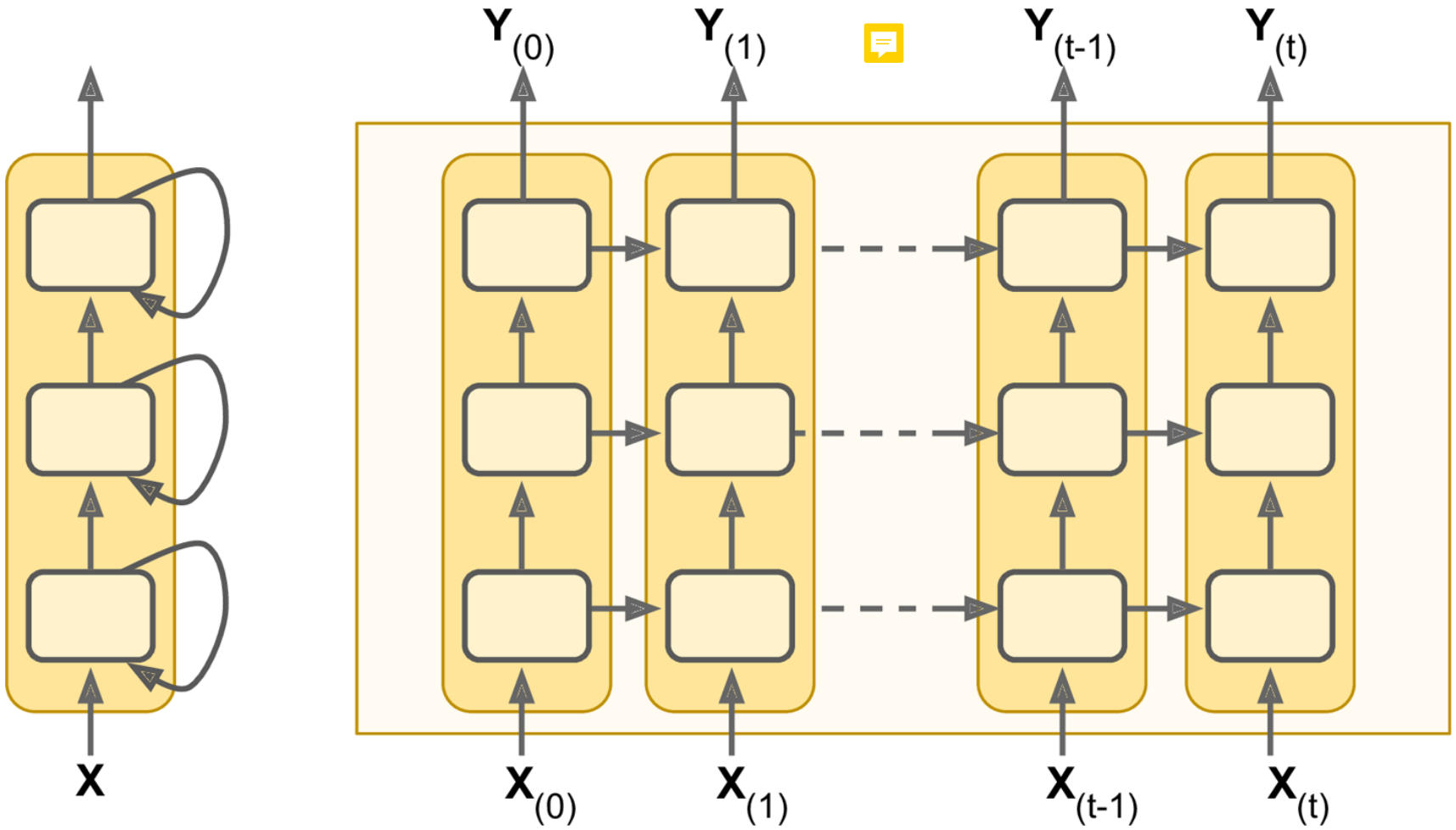
# Training RNNs

- **Backpropagation through time**
  - Compute weight & bias gradients by propagating errors backward through unrolled networks

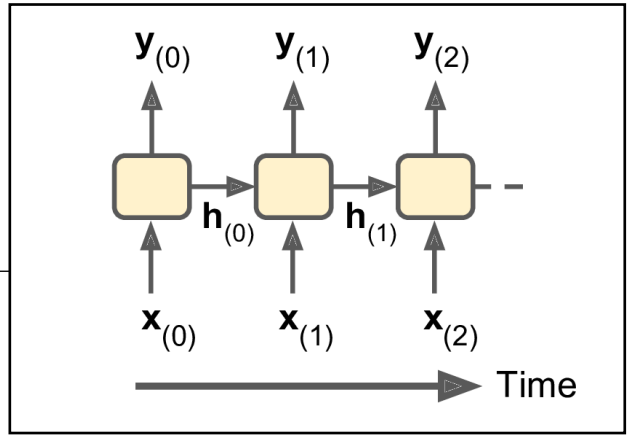



Lillicrap, et al.

# Deep RNNs

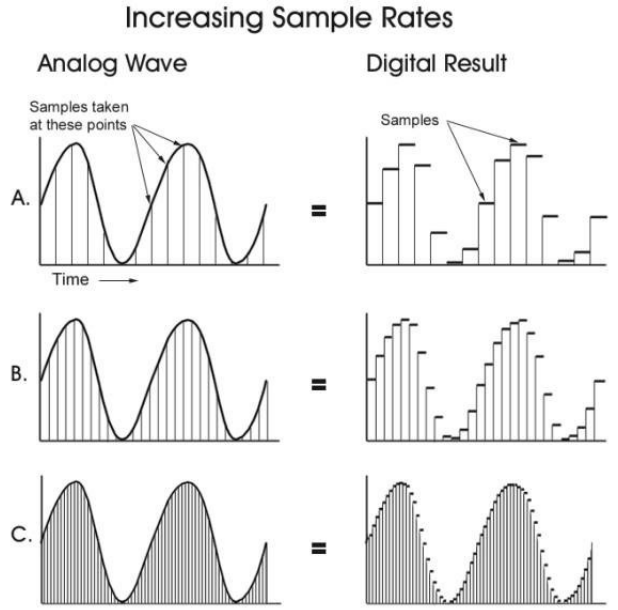


# Challenges of “Simple” RNNs



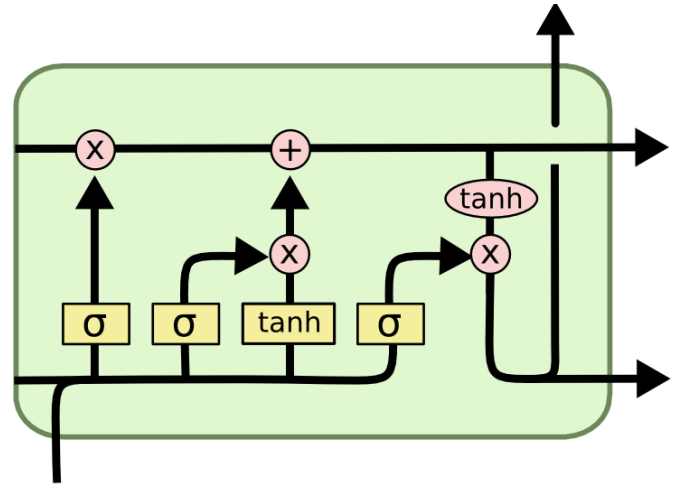

- **Limited memory**
  - Input at step *t* only affects output up to step *t+n* **for small *n***
  - Influence of previous inputs **decays** over successive steps

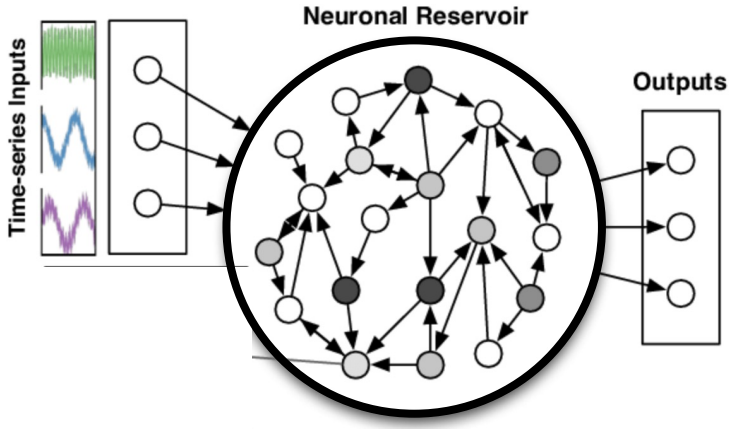
- Long sequences (with many steps) need networks with longer memory to model
  - High-fidelity audio, long text documents, etc.



# Challenges of “Simple” RNNs

- Techniques to increase RNN memory
  - **Hand-designed recurrent “cells”**
    - Next class!
  - **“Reservoir” networks**



# Questions?