Recurrent Neural Networks I

COSC 410: Applied Machine Learning
Fall 2025
Prof. Forrest Davis

October 23, 2025

Warm-up

1. It's spooky season. Discuss with your neighbor whether ghosts exist.
2. Consider the neural network in Figure 1. What is the value for $\frac{\partial z}{\partial x_1}$?

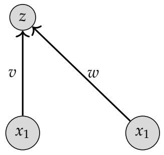

Figure 1: Sample neural network with multiple paths from a variable

Logistics

- Codelet 4 on Feed-Forward Neural Networks out, due Friday Oct 31
- Prof. Vijay is observing my course today and Tuesday

Learning Objectives

- Describe the key features of a sequential task
- Evaluate potential architectures for properties
- Apply a recurrent neural network
- Describe how learning works with recurrent neural networks (and the limitations)

Summary: We consider the problem of sequential tasks and the limitations feed-forward neural networks have in handling them. After evaluating candidate architectures, we settle on recurrent neural networks. We conclude by considering learning dynamics with recurrent neural networks and touch on modern tweaks.

The Structure of a Sequential Task

Reconsider the task of predicting the sunshine hours expected tomorrow based on the last three days of weather metrics. A sample approach using feed-forward neural networks that we settled yesterday looks like the graph in Figure 2.

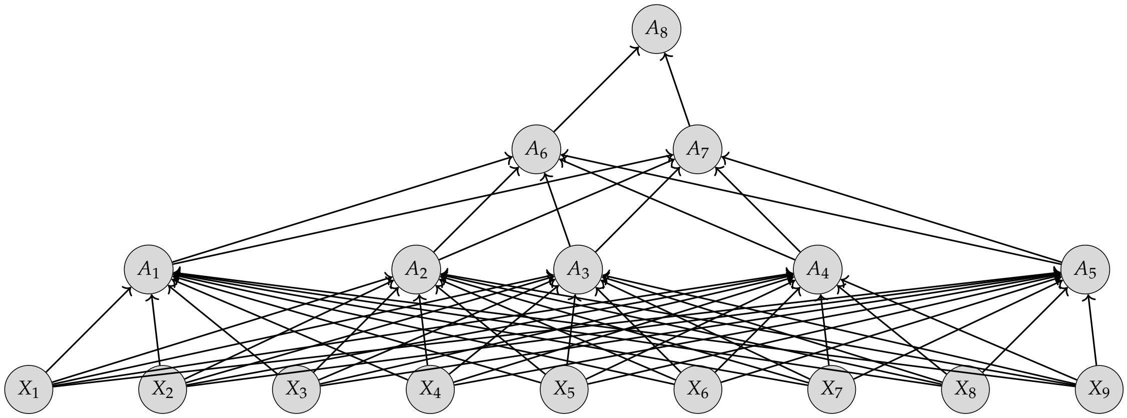

Figure 2: Three-layer neural network for predicting sunshine from three features per day. For a period of time, like Monday, Tuesday, Wednesday, x_1, x_2, x_3 would be features for Monday (e.g., humidity, max temperature, min temperature), x_4, x_5, x_6 for Tuesday, and x_7, x_8, x_9 for Wednesday.

In thinking about the task, we probably have in mind at least three facts/assumptions about the problem:

1. The humidity feature for two days ago and today is measuring the same property (even if the humidity value changes)
2. Some amount of the past is likely helpful for predicting sunshine, but we are unsure of the exact amount
    - It's probably more than one day. We saw in lab that just yesterday's sunshine is not enough to reliably predict today's sunshine.
3. More recent information is more relevant than earlier information.
    - Information from 6 months ago is less helpful than 1 week ago. Winter is quite different from summer (especially here in Hamilton)

The feed-forward network is not set up to naturally address any of these points.

Question

Give the expression for calculating A_1 and A_2 for the network given in Figure 3 with a ReLU activation function.

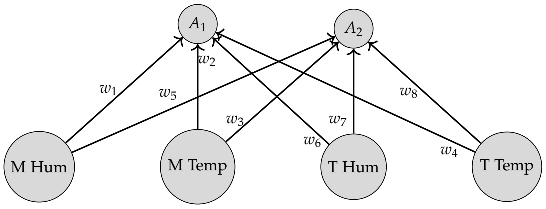

Figure 3: Feed-forward neural network basic prototype

Candidate Network Architectures

WE CAN PULL OUT OF OUR DISCUSSION ABOVE a few desiderata for a neural network designed for sequential data.

1. Uniform treatment of the same features across time
2. Retention of past information
3. A notion that time is unfolding[1]

Let's consider sample architectures and compare them to our desiderata.

Practice Problems

A standard recurrent neural network for a single feature input is given in Figure 4. Calculate the values of A_1, A_2, A_3, and A_4 given the input (assume that each node uses a ReLU activation function).

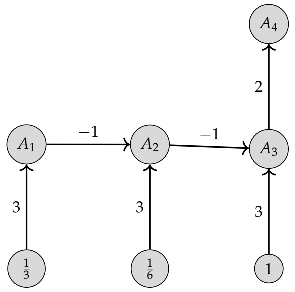

Figure 4: Sample recurrent neural network with three time steps of a one feature input to generate one output (A_4)

Backpropagation Through Time and The Limits of Context

LET'S CONSIDER HOW the weights of a recurrent neural network are updated, focusing on the gradient variables (as we did last class for feed-forward neural networks).

Question

Consider the neural network in Figure 5 used for a regression task and fit with mean-squared error (MSE). Ignore activation functions for now.

1. What is the expression for $\frac{\partial \text{MSE}}{\partial v_1}$?
2. What is the expression for $\frac{\partial \text{MSE}}{\partial w_3}$?
3. What is the expression for $\frac{\partial \text{MSE}}{\partial w_1}$?

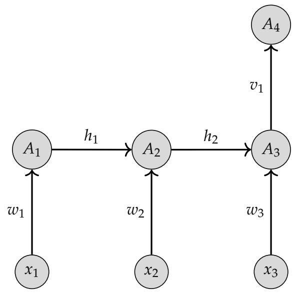

Figure 5: A neural network with three time steps of a one feature input to generate one output (A_4)

[1] This could include biases (in a technical sense) like a greater weight to more recent information, but that isn't necessarily desirable in all contexts. What is essential is capturing the temporal dynamics of sequential data.

Gradients with RNNs

Consider the more general expression of a recurrent neural network in Figure 6.

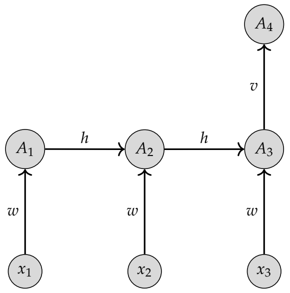

Figure 6: A recurrent neural network with three time steps of a one feature input to generate one output (A_4)

Question

What is different about the gradient calculation here compared to Figure 5?

Identifying a Problem with RNN Gradients

Notice, we can apply the recurrent neural network, recursively for any amount of input. We can diagram this fact using what's called the **rolled** representation of a recurrent neural network, given in Figure 7.

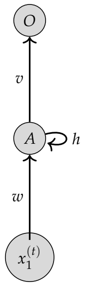

Figure 7: A rolled one-layer recurrent neural network

Question

If we ran our neural network over 100 inputs and calculated the loss for the final time step, what might happen to our gradient? Hint: imagine two cases 1. each gradient in our chain is < 1 2. each gradient in our chain is > 1.

Time is a River. We Need a Dam

There are a few strategies to deal with gradient issues with RNNs:

- Gradient clipping
- Truncated Backpropagation Through Time (Truncated BPTT)

Modern Architectures for Sequence Prediction

MODERN RECURRENT NEURAL NETWORKS uses one of two different tweaks to better handle long-distance dependencies. One is called Long short-term memory (LSTM) cells.[2] The other is called gated recurrent units (GRUs).

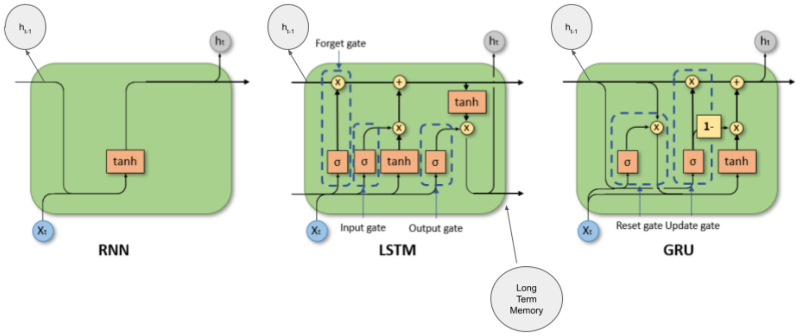

Figure 8: Diagram of different recurrent neural network cells. On the left, the plain version, in the middle LSTM, and on the right GRU.

Practice Problems

1. Draw an unrolled RNN for input over three time steps with each input having two features. The RNN should have one hidden layer and should give a one-dimensional output at the end of time step 3.

2. Provide weights that, in general, sum the values in the second input dimension and return that sum as the output.

   For example, for the input $x^{(1)} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ $x^{(2)} = \begin{bmatrix} -6 \\ 4 \end{bmatrix}$ $x^{(3)} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$ would return 7 as the output.

Before Next Class

- Reading and pre-class quiz
- Work on Codelet 4

[2] A name bordering on an oxymoron.