Exercise 1: Building RNNs: From Rolling Balls to Hidden States

Step 1.1 Imagine a ball rolling on a surface with some friction. The ball's momentum (hidden state) changes based on:

1. Previous momentum (how fast it was already rolling)
2. New external force (input)

For each scenario below, shade the boxes to show the ball's momentum (darker = faster). For simplicity, assume the relationship between momentum and time is linear, e.g., if the ball is moving at 1 m/s at t_1 and the external force is 1 m/s^2, it will be moving at 2 m/s at t_2.

Constant push force [1, 1, 1, 1]:

Momentum				
	t_1	t_2	t_3	t_4

Step 1.2 Now let's understand how friction (weight on previous state) affects motion. A high weight ($w = 0.9$) means low friction, while a low weight ($w = 0.1$) means high friction. The friction is applied to the previous momentum by multiplying it, e.g., $h_{\text{new}} = w \times h_{\text{old}}$.

Shade these boxes showing momentum for different friction levels, with force [2, 2, 0, 0]:

Momentum				
	t_1	t_2	t_3	t_4

> *Think:* How does friction (weight) affect how long the ball "remembers" previous pushes?

Step 1.3 Let's turn this physical intuition into an RNN. The new hidden state is:

$$h_{\text{new}} = \text{tanh}(w \times h_{\text{old}} + w_x \times \text{force})$$

where:

1. h_{old} is previous momentum
2. force is input
3. w relates to friction (how much previous momentum is preserved)
4. w_x relates to how effectively force changes momentum
5. tanh keeps momentum from growing infinitely

For these force sequences, shade the predicted momentum, with $w = 1.0$ and $w_x = 1$:

[3,0,0,0]				[1,1,1,1]				[0,3,0,0]			
☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

> *Think:* How does each force sequence affect momentum differently? Which sequence would be hardest for the network to "learn"?

Step 1.4 Design your own RNN weights! If you wanted to: (Design 1) Remember past inputs longer, (Design 2) Respond more quickly to new forces, (Design 3) Have a maximum speed limit. Shade these weight matrices to achieve each goal:

Design 1		Design 2		Design 3	
w	w_x	w	w_x	w	w_x

Step 1.5 Predict what would happen with these "physically impossible" weights:

1. $w > 1$ (momentum grows from previous state)
2. No activation function
3. Negative weights

Shade the momentum evolution for input sequence [1, 1, 1, 1]:

$w = 1.2$				
No tanh				
$w = -0.5$				

Think: Why are these scenarios "impossible" physically but possible in an RNN? What problems might they cause or solve?