RNN (Recurrent Neural Network)

Core Architecture

x_t → h_t ↻ h_{t+1}

Hidden State Update Formula

$$h_t = \tanh(W_{hh} \times h_{t-1} + W_{xh} \times x_t)$$

💾 **Hidden state** maintains sequence memory

🔄 **Weight sharing:** same weights at each timestep

↔️ **Bidirectional RNN:** forward & backward processing

⏱️ **Sequential processing:** models temporal dependencies

⚠️ Training Challenges

- Vanishing/exploding gradients
- Difficult to train for long sequences

📊 Detailed Trend Calculation Example

Setup: Hidden dim = 2, Input dim = 3

W_{xh} = [[0.5, 0.3], [0.2, 0.4], [0.1, 0.6]] (3×2), W_{hh} = [[0.7, 0.2], [0.3, 0.8]] (2×2)

Input sequence: x_0 = [1.0, 0.5, 0.8], x_1 = [1.2, 0.6, 0.9]

Step 1 (t=0):

- h_0 = [0, 0] (initial hidden state)
- $W_{hh} \times h_0$ = [[0.7, 0.2], [0.3, 0.8]] × [0, 0] = **[0, 0]**
- $W_{xh} \times x_0$ = [[0.5, 0.3], [0.2, 0.4], [0.1, 0.6]] × [1.0, 0.5, 0.8] = **[0.68, 0.88]**
- h_1 = tanh([0, 0] + [0.68, 0.88]) = tanh([0.68, 0.88]) = **[0.59, 0.71]**

Step 2 (t=1):

- h_1 = [0.59, 0.71]
- $W_{hh} \times h_1$ = [[0.7, 0.2], [0.3, 0.8]] × [0.59, 0.71] = **[0.55, 0.75]**
- $W_{xh} \times x_1$ = [[0.5, 0.3], [0.2, 0.4], [0.1, 0.6]] × [1.2, 0.6, 0.9] = **[0.78, 1.02]**
- h_2 = tanh([0.55, 0.75] + [0.78, 1.02]) = tanh([1.33, 1.77]) = **[0.87, 0.95]**

Trend detected: Hidden states evolve [0, 0] → [0.59, 0.71] → [0.87, 0.95], capturing increasing trend through accumulated memory in both dimensions