🌱 一、先定义一个最小 RNN

我们构造一个极小的循环神经网络：

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

损失函数定义为：

$$L = \frac{1}{2}(y_2 - \hat{y}_2)^2$$

也就是：我们只在第 2 个时间步有损失。

参数：

- W_{xh}：输入权重
- W_{hh}：隐藏层的循环权重（关键）

输入序列：

- $x_1 = 1.0, x_2 = 2.0$
- 初始隐藏状态 $h_0 = 0$
- 目标输出（ground truth）：$\hat{y}_2 = 1.0$

二、前向传播

逐步算出每个时间步的值。

Step 1: 时间步 t=1

$$h_1 = \tanh(W_{xh}x_1 + W_{hh}h_0)$$

假设：$W_{xh} = 1.0$, $W_{hh} = 0.5$

所以：

$$h_1 = \tanh(1.0 * 1.0 + 0.5 * 0) = \tanh(1.0) = 0.7616$$

Step 2: 时间步 t=2

$$h_2 = \tanh(W_{xh}x_2 + W_{hh}h_1)$$
$$h_2 = \tanh(1.0 * 2.0 + 0.5 * 0.7616) = \tanh(2.3808) = 0.9831$$

输出 $y_2 = h_2 = 0.9831$

损失：

$$L = \frac{1}{2}(0.9831 - 1.0)^2 = \frac{1}{2}(-0.0169)^2 = 0.000143$$

三、反向传播 Through Time

我们要算：

$$\frac{\partial L}{\partial W_{hh}}$$

这相当于问："若循环权重 W_{hh}稍微变动，最终损失 L 会怎样变化？"

第一步：从最后一步开始反传

因为 L只依赖 $y_2 = h_2$，

所以：

$$\frac{\partial L}{\partial h_2} = (h_2 - \hat{y}_2) = (0.9831 - 1.0) = -0.0169$$

第二步：L 对 W_{hh}的直接影响（在 t=2）

$h_2 = \tanh(W_{xh}x_2 + W_{hh}h_1)$，

所以：

$$\frac{\partial h_2}{\partial W_{hh}} = (1 - h_2^2) \cdot h_1$$

（使用 $\frac{d\tanh(z)}{dz} = 1 - \tanh^2(z)$）

代入数值：

$$(1 - 0.9831^2) \cdot 0.7616 = (1 - 0.9665) * 0.7616 = 0.0335 * 0.7616 = 0.0255$$

因此：

$$\text{直接梯度（第 2 步）} = \frac{\partial L}{\partial h_2} \cdot \frac{\partial h_2}{\partial W_{hh}} = (-0.0169) * 0.0255 = -0.00043$$

第三步：来自前一时刻（t=1）的间接影响

还要加上 W_{hh}通过 $h_1 \to h_2 \to L$的间接贡献。

链式法则：

$$\frac{\partial L}{\partial W_{hh}}^{\text{(间接)}} = \frac{\partial L}{\partial h_2} \cdot \frac{\partial h_2}{\partial h_1} \cdot \frac{\partial h_1}{\partial W_{hh}}$$

分别计算：

1. $\frac{\partial h_2}{\partial h_1} = (1 - h_2^2) \cdot W_{hh} = 0.0335 * 0.5 = 0.01675$
2. $\frac{\partial h_1}{\partial W_{hh}} = (1 - h_1^2) \cdot h_0 = (1 - 0.7616^2) * 0 = 0$

因为 $h_0 = 0$，所以这项为 **0**。

✅ 所以总梯度：

$$\frac{\partial L}{\partial W_{hh}} = (-0.00043) + 0 = -0.00043$$