In [1]:

```
import torch
import torch.nn as nn

class TinyRNN(nn.Module):
  def __init__(self, wh_init=1):
    super().__init__()
    self.h_init = nn.Parameter(torch.tensor(1.0))
    self.wh = nn.Parameter(torch.tensor(wh_init).float())
    self.wx = torch.tensor(0.0)
    self.wy = nn.Parameter(torch.tensor(1.0))

  def forward(self, x_seq):
    # Single RNN step
    def _single_step(x, h):
      # For simplicty, we use the relu activation,
      #  instead of tanh when computing the new h
      new_h = torch.relu(self.wh * h + self.wx * x)
      y = self.wy * new_h
      return y, new_h

    # Loop through sequence
    h = self.h_init
    h_list = [h]
    y_list = []
    for i, x in enumerate(x_seq):
      y, h = _single_step(x, h)
      y_list.append(y)
      h_list.append(h)

    return torch.stack(y_list), torch.stack(h_list)

# Run through some examples with different
#  initializations and sequence lengths
criterion = torch.nn.MSELoss()
for wh_init in [0.1, 1.0, 10.0]:
  print(f'wh_init = {wh_init}')

  for L in [1, 2, 3, 4, 5, 10]:
    # Initialize model and run
    rnn = TinyRNN(wh_init=wh_init)
    x_seq = torch.ones(L)
    y_seq_pred, h_seq = rnn(x_seq)

    # Compute loss and gradient
    y_true = y_seq_pred[-1].detach() - 1 # Fake target to ensure MSE is 1
    loss = criterion(y_seq_pred[-1], y_true)
    loss.backward()

    print(f'  L={L:2d}  ' + ', '.join([f'{name}.grad = {p.grad:2.1e}' for
```

```
wh_init = 0.1
  L= 1  h_init.grad = 2.0e-01, wh.grad = 2.0e+00, wy.grad = 2.0e-01
  L= 2  h_init.grad = 2.0e-02, wh.grad = 4.0e-01, wy.grad = 2.0e-02
  L= 3  h_init.grad = 2.0e-03, wh.grad = 6.0e-02, wy.grad = 2.0e-03
  L= 4  h_init.grad = 2.0e-04, wh.grad = 8.0e-03, wy.grad = 2.0e-04
  L= 5  h_init.grad = 2.0e-05, wh.grad = 1.0e-03, wy.grad = 2.0e-05
  L=10  h_init.grad = 2.0e-10, wh.grad = 2.0e-08, wy.grad = 2.0e-10
wh_init = 1.0
  L= 1  h_init.grad = 2.0e+00, wh.grad = 2.0e+00, wy.grad = 2.0e+00
  L= 2  h_init.grad = 2.0e+00, wh.grad = 4.0e+00, wy.grad = 2.0e+00
  L= 3  h_init.grad = 2.0e+00, wh.grad = 6.0e+00, wy.grad = 2.0e+00
  L= 4  h_init.grad = 2.0e+00, wh.grad = 8.0e+00, wy.grad = 2.0e+00
  L= 5  h_init.grad = 2.0e+00, wh.grad = 1.0e+01, wy.grad = 2.0e+00
  L=10  h_init.grad = 2.0e+00, wh.grad = 2.0e+01, wy.grad = 2.0e+00
wh_init = 10.0
  L= 1  h_init.grad = 2.0e+01, wh.grad = 2.0e+00, wy.grad = 2.0e+01
  L= 2  h_init.grad = 2.0e+02, wh.grad = 4.0e+01, wy.grad = 2.0e+02
  L= 3  h_init.grad = 2.0e+03, wh.grad = 6.0e+02, wy.grad = 2.0e+03
  L= 4  h_init.grad = 2.0e+04, wh.grad = 8.0e+03, wy.grad = 2.0e+04
  L= 5  h_init.grad = 2.0e+05, wh.grad = 1.0e+05, wy.grad = 2.0e+05
  L=10  h_init.grad = 0.0e+00, wh.grad = 0.0e+00, wy.grad = 0.0e+00
```