



图 6-7 仅关注 RNN 层的矩阵乘积时的反向传播的梯度

在图 6-7 中，假定从上游传来梯度 dh ，此时 MatMul 节点的反向传播通过矩阵乘积 $dh W_h^T$ 计算梯度。之后，根据时序数据的时间步长，将这个计算重复相应次数。这里需要注意的是，每一次矩阵乘积计算都使用相同的权重 W_h 。

那么，反向传播时梯度的值通过 MatMul 节点时会如何变化呢？一旦有了疑问，最好的方法就是做实验！让我们通过下面的代码，来观察梯度大小的变化（[⇒ ch06/rnn_gradient_graph.py](#)）。

```
import numpy as np
import matplotlib.pyplot as plt

N = 2 # mini-batch 的大小
H = 3 # 隐藏状态向量的维数
T = 20 # 时序数据的长度

dh = np.ones((N, H))
np.random.seed(3) # 为了复现，固定随机数种子
Wh = np.random.randn(H, H)

norm_list = []
for t in range(T):
    dh = np.dot(dh, Wh.T)
    norm = np.sqrt(np.sum(dh**2)) / N
    norm_list.append(norm)
```

这里用 `np.ones()` 初始化 dh (`np.ones()` 是所有元素均为 1 的矩阵)。然后，根据反向传播的 MatMul 节点的数量更新 dh 相应次数，并将各步的 dh