



















代码中dx表示dL/dx，dw表示dL/dw 。

设： y = f(xw), dx = f’(x)w = dy\*w, dw = x f’(x) = x\*dy, 即： 

因为：

1. dh = np.dot(Why.T, dy) + dhnext——是在计算隐藏层的梯度。

dh：隐藏层的梯度。

np.dot(Why.T, dy)：这是输出权重矩阵Why的转置与误差向量dy的点积。这个操作实际上是在将误差反向传播到隐藏层。

dhnext：下一个时间步的隐藏状态的梯度，这是从未来步骤反向传播回来的梯度。

所以，dh = np.dot(Why.T, dy) + dhnext 这行代码的意思是，当前时间步的隐藏状态的梯度是由两部分组成的：一部分是当前时间步输出误差反向传播回来的梯度，另一部分是未来步骤反向传播回来的梯度。

1. dhraw = (1 - hs[t] \* hs[t]) \* dh

dhraw ：未激活隐藏状态的梯度， 即h\*。

原始（未激活）隐藏状态的梯度dhraw是由tanh激活函数的导数和隐藏层的梯度的乘积得到的。

1. dWxh += np.dot(dhraw, xs[t].T)

dWhh += np.dot(dhraw, hs[t-1].T)

dhnext = np.dot(Whh.T, dhraw) 即： dhnext 就是 dht-1， 下一次循环再加。

1和2一起，实现递推。