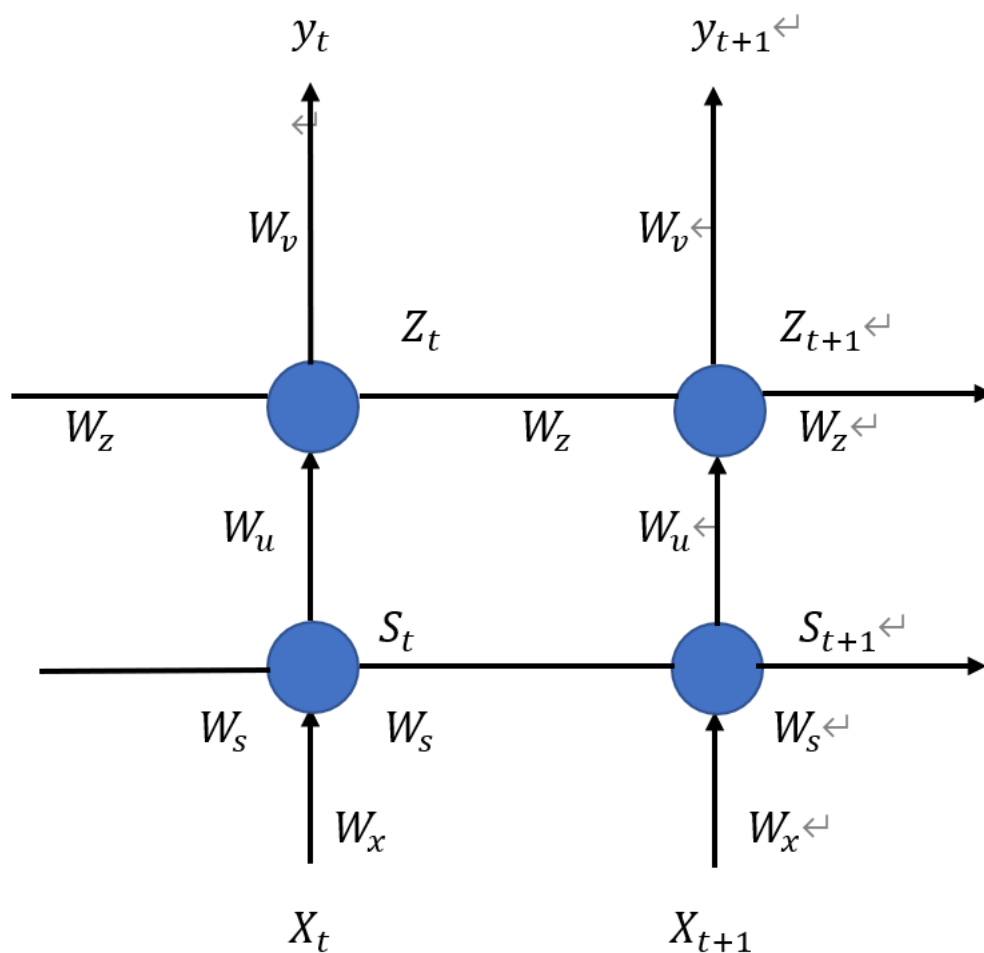


## 循环神经网络

这篇文章我们谈一下循环神经网络。之前我们使用的都是卷积神经网络，卷积神经网络可以帮助我们有效地处理输入的空间信息，CNN 擅长查找训练数据中的空间和视觉这类任务。而循环神经网络可以处理序列数据，它引入状态变量来存储过去的记忆，并且与当前输入共同决定当前的输出。这种处理方式非常重要，因为现实中很多时候我们面对的都是带有序列性质的数据，比方说一个句子，他们都是一系列单词组成。还有一段语音或者视频，都是基于一系列片段组成的。训练这类数据，我们不能将它们单独分离处理，必须得考虑其上下文关系。就比如一个动作，你看一张图片是判断不出是什么动作的，得给你一组图片你才能看出动作趋势。翻译句子也是这个道理，你不可能一个单词一个单词的翻译，你肯定要结合整句子的结构来处理，这些都是循环神经网络适合处理的。我们着重讲解下普通的循环神经单元 RNN 和改进的 LSTM。

我们先说下 RNN，它的结构类似如下，请看下图：



RNN 展开模型图

这是个两层的单向 RNN 结构。输入是  $X_t$ ，这是个序列数据，下标  $t$  代表的是某一时刻即是序列数据的索引。比方说一个句子 [I live football],  $X_0$  对应的就是 'I',  $X_1$  对应的是 'like',  $X_2$  就对应的是 'football'。  $S_t$  和  $Z_t$  则是我们之前说的状态变量，用来存储之前的记忆。  $W_s$  是连接

第一层输入层和第一层状态层的权重矩阵， $W_u$ 是连接第二层输入和第二层状态层的权重矩阵。 $W_v$ 是连接第二层状态层的输出到输出层的权重矩阵。 $W_s$ 和 $W_z$ 是第一层和第二层状态层中，表示连接之前时间步长状态到当前时间步长状态的权重矩阵。 $y$ 是最终输出结果，每个时刻都对应一个输出。接下来我可以给出 RNN 的前向传播过程。计算公式如下。

$$S_{t+1} = W_s S_t + X_{t+1}$$

$$Z_{t+1} = W_z Z_t + W_u S_{t+1}$$

$$y_t = W_v Z_t$$

$$y_{t+1} = W_v Z_{t+1}$$

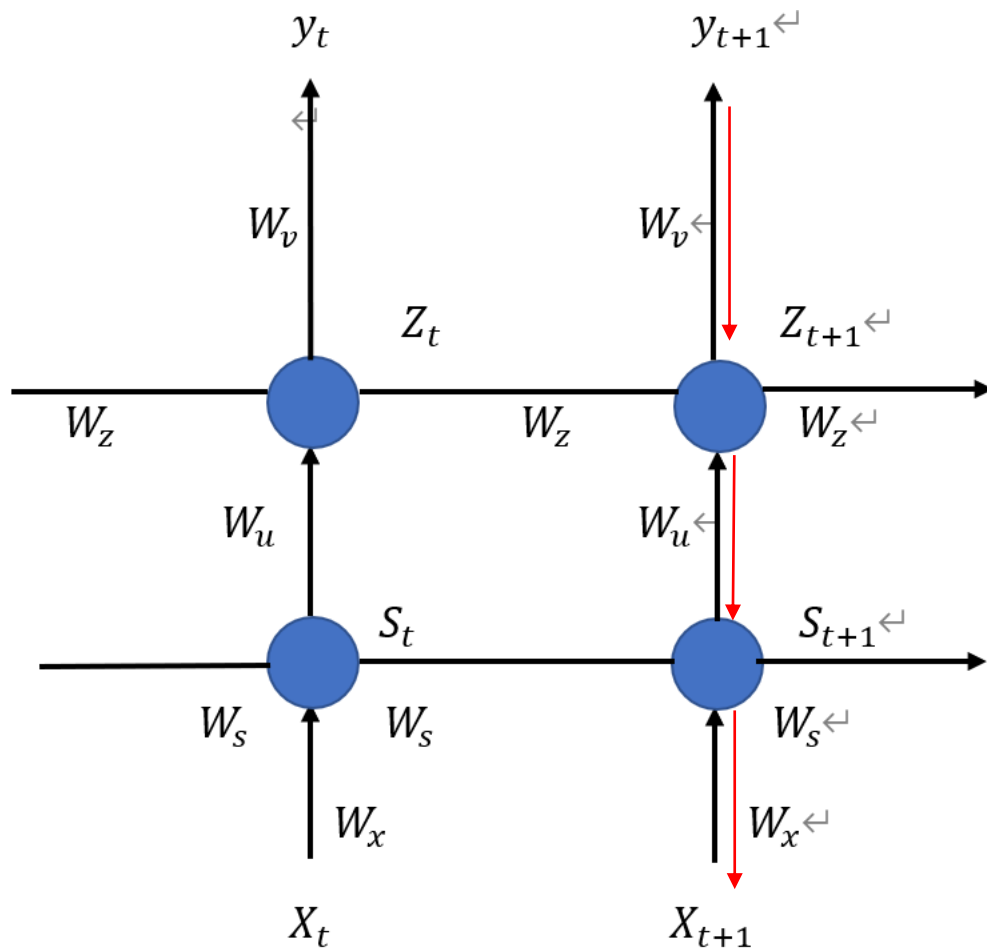
这便是两层循环神经网络的前向传播过程。显然就是前面一层的输出是后面一层的输入。和普通神经网络没有区别，仅仅是加入了序列的处理。通过之前存储的状态变量和当前输入共同决定当前的输出。这种方式使得我们预测当前的输出结果并不仅仅是通过当前的输入开决定，还会参考之前的信息。这样学习到的模式更加的抽象高级，序列其它位置如果也有类似的模式也会被识别出来。这样会更进一步提升我们预测的准确度。也确实符合我们处理序列类型数据的逻辑思考方式。

说完了序列的前向传播，接下来我们自然就要讨论下循环神经网络的反向传播过程了。我们还用上面对出的两层神经网络来说明方向传播的过程。我们用定义 $E_t$ 来表示时刻  $t$  的输出误差。我们的损失误差函数定义如下：

$$E_{t+1} = (d_{t+1} - y_{t+1})^2$$

这里 $d_{t+1}$ 是实际标签的输出值。与预测值 $y_{t+1}$ 做比较。那么这个经历了两层循环神经网络的传递得到输出结果，其方向传播的公式是什么样子。我们以 $\frac{\partial E_{t+1}}{\partial W_x}$ 为例子来说明反向传播的过程。 $\frac{\partial E_{t+1}}{\partial W_x}$ 也是最复杂的一个，搞懂了这个其它都不是难题。

那么 $\frac{\partial E_{t+1}}{\partial W_x}$ 是怎么计算出来的呢？因为只需要观察时间  $t$  和时间  $t+1$ ，我们会在这两个时间步长展开模型。其实通过之前展示的 RNN 展开模型图，我们可以看到计算 $\frac{\partial E_{t+1}}{\partial W_x}$ ，涉及三个反向传播路径。以下三张图片可以帮你了解我们需要考虑的这三条反向传播路径。我们根据这些传播路径，使用链式法就可以解决了。我们先展示第一条路径：

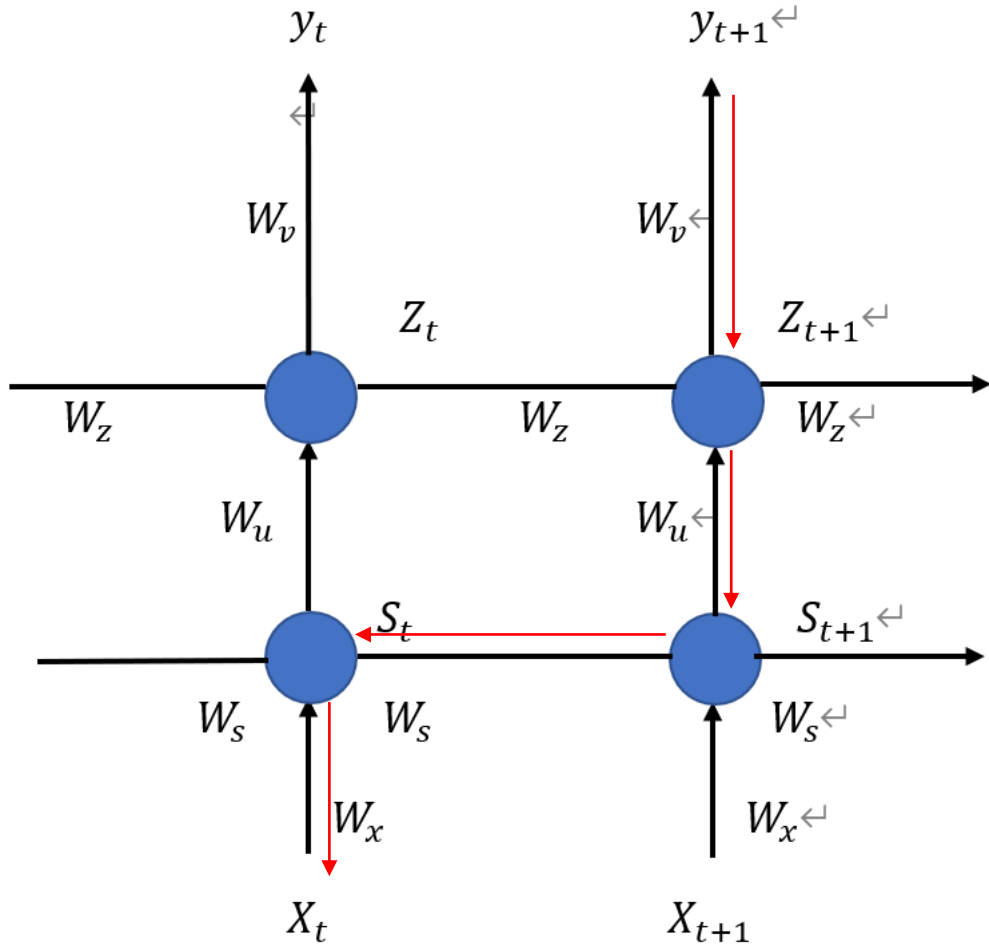


第一条反向传播路径

好了，以上就是我们的第一条方向传播路径展开的模型图。我们可以很容易看到传播的路径,看红色箭头方向。那么根据这条路径通过链式法则求导的结果如下：

$$\frac{\partial E_{t+1}}{\partial y_{t+1}} \frac{\partial y_{t+1}}{\partial Z_{t+1}} \frac{\partial Z_{t+1}}{\partial S_{t+1}} \frac{\partial S_{t+1}}{\partial W_x}$$

同理我们再看第二条反向传播路径，如下如所示：

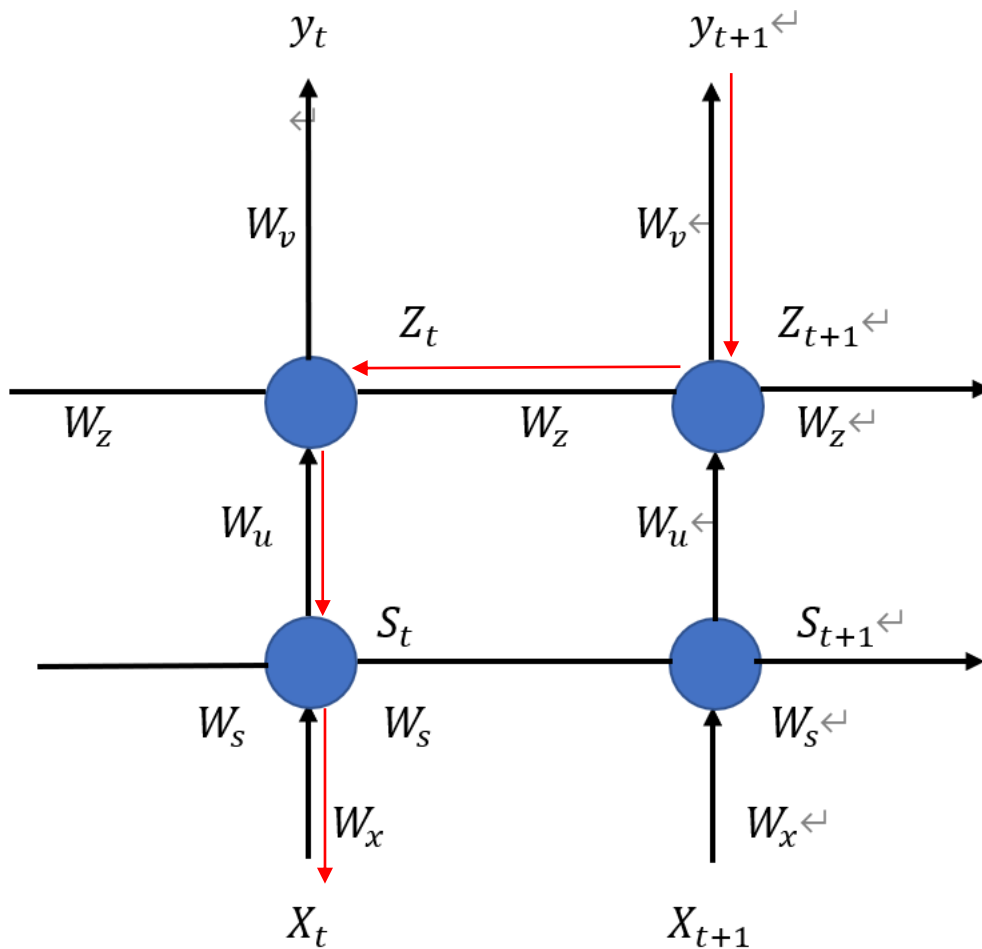


第二条反向传播路径

我们根据红色箭头方向根据链式法则写出第二条反向传播路径的求导公式如下：

$$\frac{\partial E_{t+1}}{\partial y_{t+1}} \frac{\partial y_{t+1}}{\partial Z_{t+1}} \frac{\partial Z_{t+1}}{\partial S_{t+1}} \frac{\partial S_{t+1}}{\partial S_t} \frac{\partial S_t}{\partial W_x}$$

最后我们再看下第三条反向传播路径的模型展开示意图：



第三条反向传播路径

根据红色箭头方向，让我们写出这条传播路径的求导过程吧。

$$\frac{\partial E_{t+1}}{\partial y_{t+1}} \frac{\partial y_{t+1}}{\partial Z_{t+1}} \frac{\partial Z_{t+1}}{\partial Z_t} \frac{\partial Z_t}{\partial S_t} \frac{\partial S_t}{\partial W_x}$$

好了涉及到 t+1 时刻的三条方向传播路径我们都写出来了。大家可以仔细看看，看是否还有其它路径。那肯定是没有了。我们通过这种找出所有相关的反向传播路径，再根据链式法则就可以写出相应的求导公式了，这种方法使我们可以很容易的写出循环神经网络的反向传播求导公式来。大家仔细体会一下。最后我们把这三条路径的求导公式加起来就是我们的需要的

的最终结果，即  $\frac{\partial E_{t+1}}{\partial W_x}$ 。

$$\begin{aligned} \frac{\partial E_{t+1}}{\partial W_x} = & \frac{\partial E_{t+1}}{\partial y_{t+1}} \frac{\partial y_{t+1}}{\partial Z_{t+1}} \frac{\partial Z_{t+1}}{\partial S_{t+1}} \frac{\partial S_{t+1}}{\partial W_x} + \frac{\partial E_{t+1}}{\partial y_{t+1}} \frac{\partial y_{t+1}}{\partial Z_{t+1}} \frac{\partial Z_{t+1}}{\partial S_{t+1}} \frac{\partial S_{t+1}}{\partial S_t} \frac{\partial S_t}{\partial W_x} \\ & + \frac{\partial E_{t+1}}{\partial y_{t+1}} \frac{\partial y_{t+1}}{\partial Z_{t+1}} \frac{\partial Z_{t+1}}{\partial Z_t} \frac{\partial Z_t}{\partial S_t} \frac{\partial S_t}{\partial W_x} \end{aligned}$$

