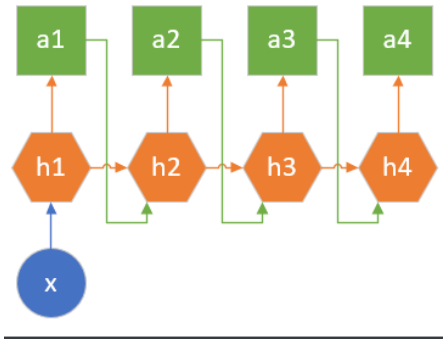
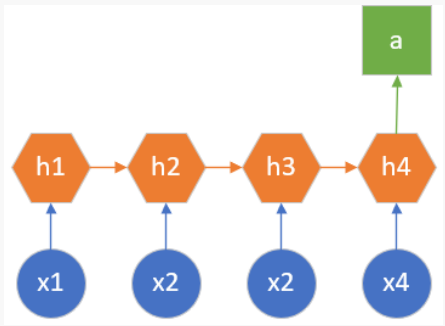
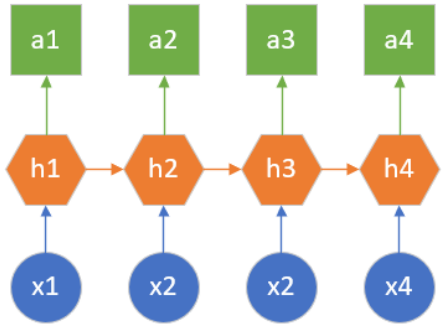
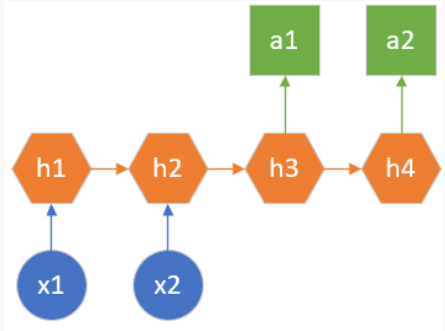
1. 循环神经网络

1.1 定义

循环神经网络实际上前馈全连接神经网络的一种扩展；如果说全连接网络是学习静态数据的非线性特征的，那么循环神经网络就是学习动态序列数据的非线性特征的。

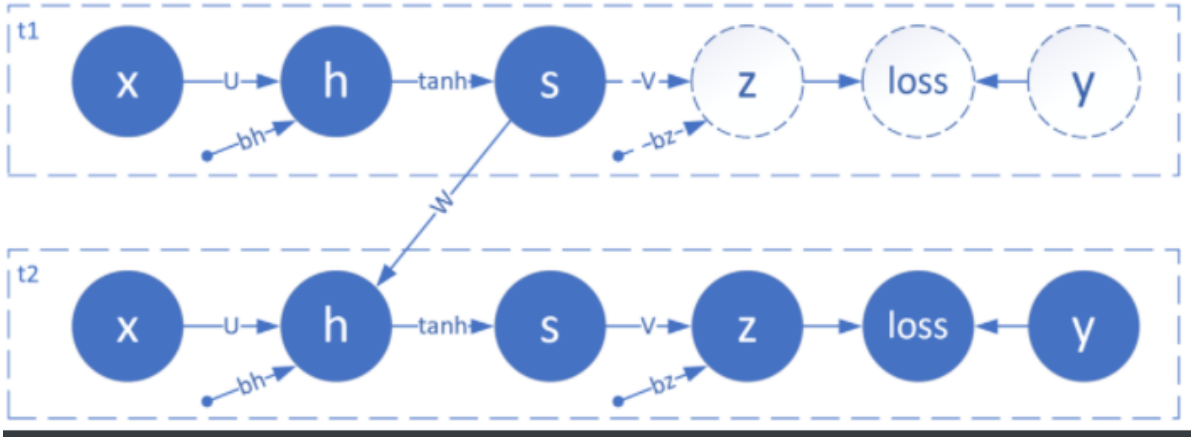
2.1 普通循环神经网络

2.1.1 循环神经网络的结构和典型用途

结构	图示	实例
一对多的结构		一个输入对应一个输出
多对一的结构		看完电影得到一段影评，或者几颗星的评价，或者预测股价
多对多（输入输出等量）		输入是“hell”四个字母，输出是“ello”四个字母的概率。
多对多（输入输出不等量）		机器翻译

2.1.2 两个时间步的循环神经网络

使用前馈神经网络的概念来做正向和反向推导，但是通过t1、t2两个时序的衔接，图示如下：



接收到两个序列的数值时，返回第一个序列的数值。

2.1.3 四个时间步的循环神经网络

在加减法运算中，总会遇到进位或者退位的问题，我们以二进制为例，比如13-6=7这个十进制的减法，变成二进制后如下所示：

```
13 - 6 = 7
====================
  x1: [1, 1, 0, 1]
- x2: [0, 1, 1, 0]
------------------
  y:  [0, 1, 1, 1]
====================
```

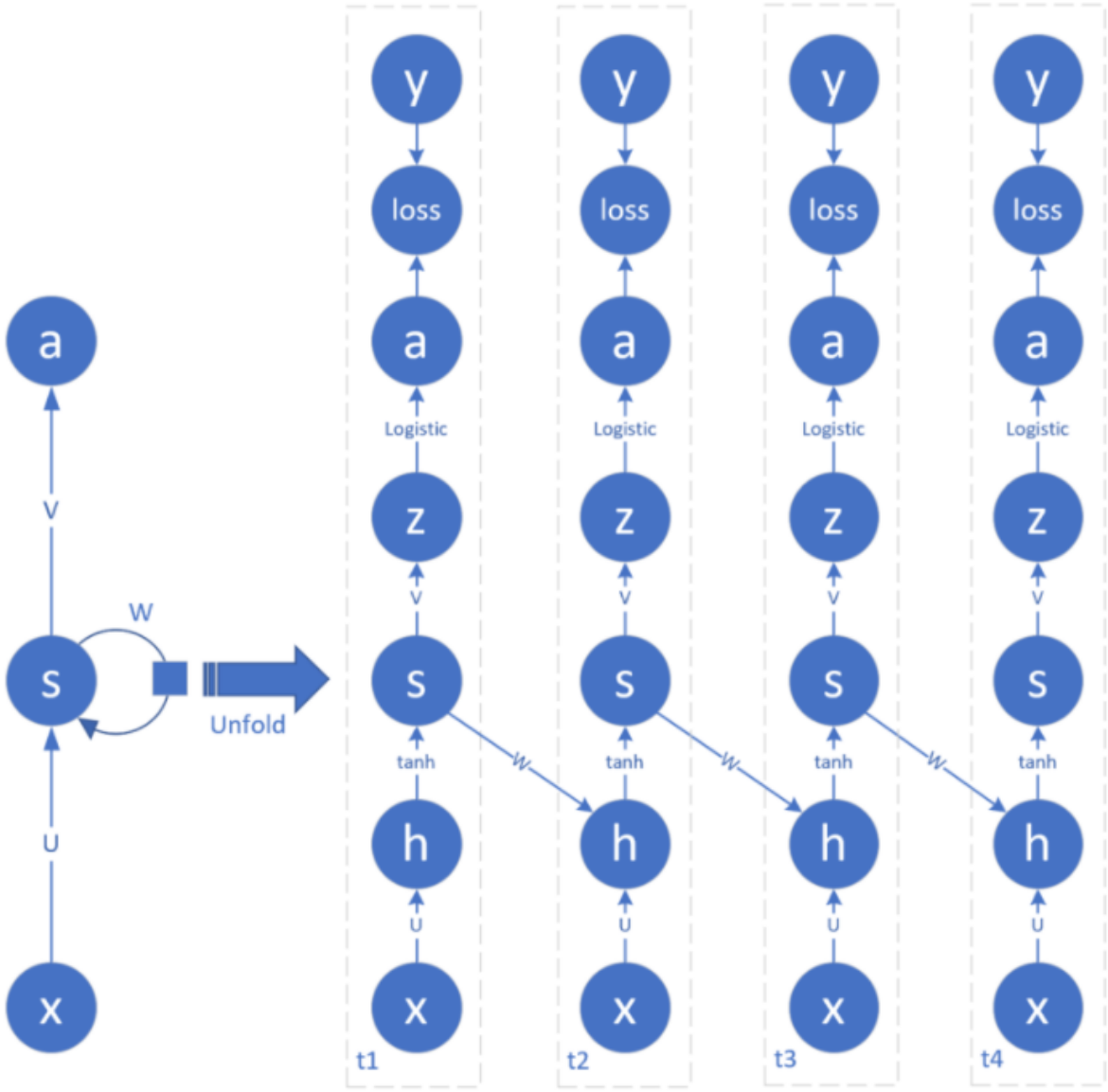
也就是说，在减法过程中，后面的计算会影响前面的值，所以必须逐位计算，这也就是时间步的概念，所以可以用循环神经网络的技术来解决。

如下：标签值为一组4位二进制数。三组二进制数都是倒序。

时间步	特征值1	特征值2	标签值
1（最低位）	1	0	1
2	0	1	1
3	1	1	1
4（最高位）	1	0	0

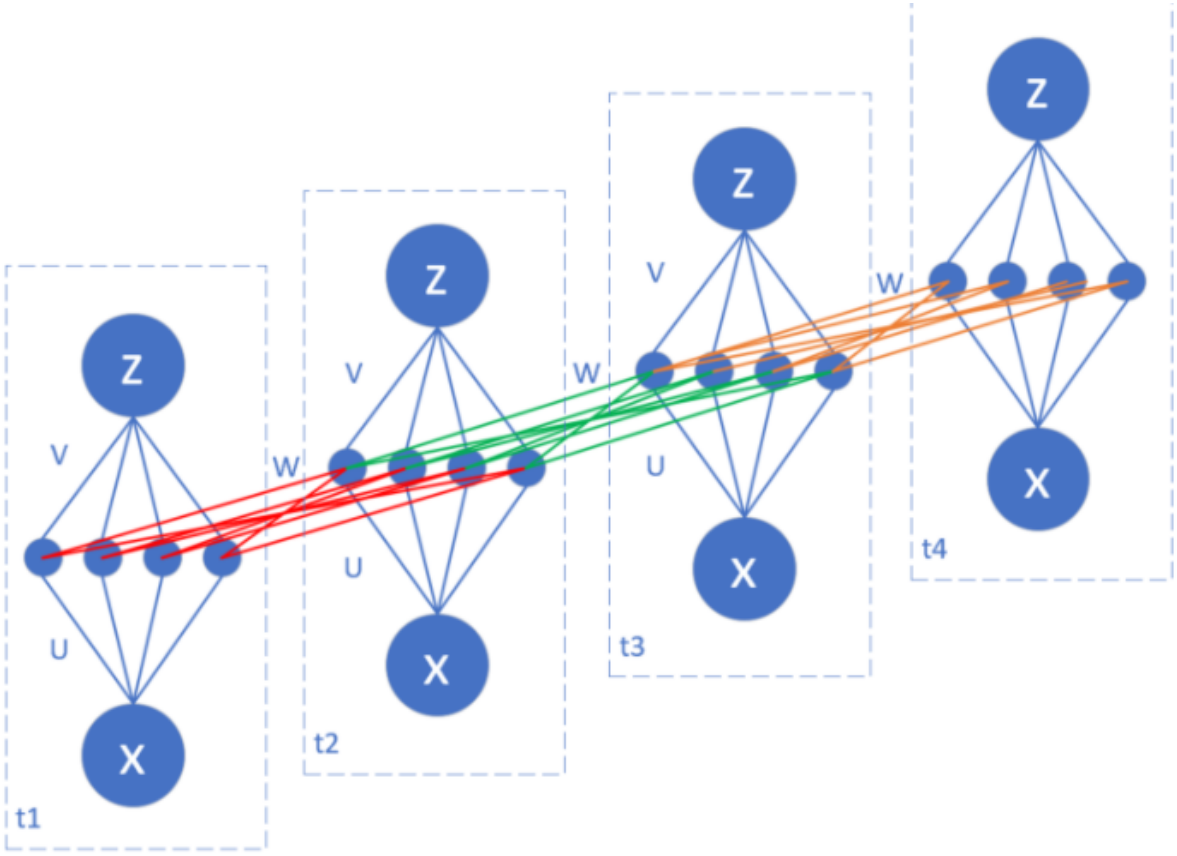
所以，单个样本是一个二维数组，而多个样本就是三维数组，第一维是样本，第二维是时间步，第三维是特征值。

得到的结构如下：



在每个时间步的结构中，多出来一个a，是从z经过二分类函数生成的。这是为什么呢？因为在本例中，模拟二进制数的减法，所以结果应该是0或1，于是我们把它看作是二分类问题，z的值是一个浮点数，用二分类函数后，使得a的值尽量向两端（0或1）靠近，但是并不能真正地达到0或1，只要大于0.5就认为是1，否则就认为是0。

由于隐层神经元数量为4，所以U是一个 1x4 的参数矩阵，V是一个 4x1 的参数矩阵，而W就是一个 4x4 的参数矩阵。把它们展开画成图如下：



W是一个连接相邻时序的参数矩阵，并且共享相同的参数值（**注意是共享**）

2.1.4 通用的循环神经网络模型

不同场景下的循环神经网络参数

	回波检测问题	**二进制减法问题**	**PM2.5预测问题**
时间步	2	4	用户指定参数
网络输出类别	回归	二分类	多分类
分类函数	无	Logistic函数	Softmax函数
损失函数	均方差	二分类交叉熵	多分类交叉熵
时间步输出	最后一个	每一个	最后一个
批大小	1	1	用户指定参数
有无偏移值	有	无	有

“比较通用”是什么意思呢？那就是应该满足以下条件：

1. 既可以支持分类网络（二分类和多分类），也可以支持回归网络；
2. 每一个时间步可以有输出并且有监督学习信号，也可以只在最后一个时间步有输出；
3. 第一个时间步的前向计算中不包含前一个时间步的隐层状态值（因为前面没有时间步）；
4. 最后一个时间步的反向传播中不包含下一个时间步的回传误差（因为后面没有时间步）；
5. 可以指定超参数进行网络训练，如：学习率、批大小、最大训练次数、输入层尺寸、隐层神经元数量、输出层尺寸等等；
6. 可以保存训练结果并可以在以后加载参数，避免重新训练。

2.1.5 不定长时序的循环神经网络

典型例子：各个国家的人都有自己习惯的一些名字，下面列举出了几个个国家/语种的典型名字

```
Guan    Chinese
Rong    Chinese
Bond    English
Stone   English
Pierre  French
Vipond  French
Metz    German
Neuman  German
Aggio   Italian
Falco   Italian
Akimoto Japanese
Hitomi  Japanese
```

如果两个样本的时间步总数不同，是不能做为一个批量一起喂给网络的，比如一个名字是Rong，另一个名字是Aggio，这两个名字不能做为一批计算。由于名字的长度不同，所以不同长度的两个名字，是不能放在一个batch里做批量运算的。但是如果一个一个地训练样本，将会花费很长的时间

所以需要我们对本例中的数据做一个特殊的处理：

1. 先按字母个数（名字的长度）把所有数据分开，由于最短的名字是2个字母，最长的是19个字母，所以一共应该有18组数据（实际上只有15组，中间有些长度的名字不存在）。
2. 使用OneHot编码把名字转换成向量，比如：名字为“Duan”，变成小写字母“duan”，则OneHot编码是：

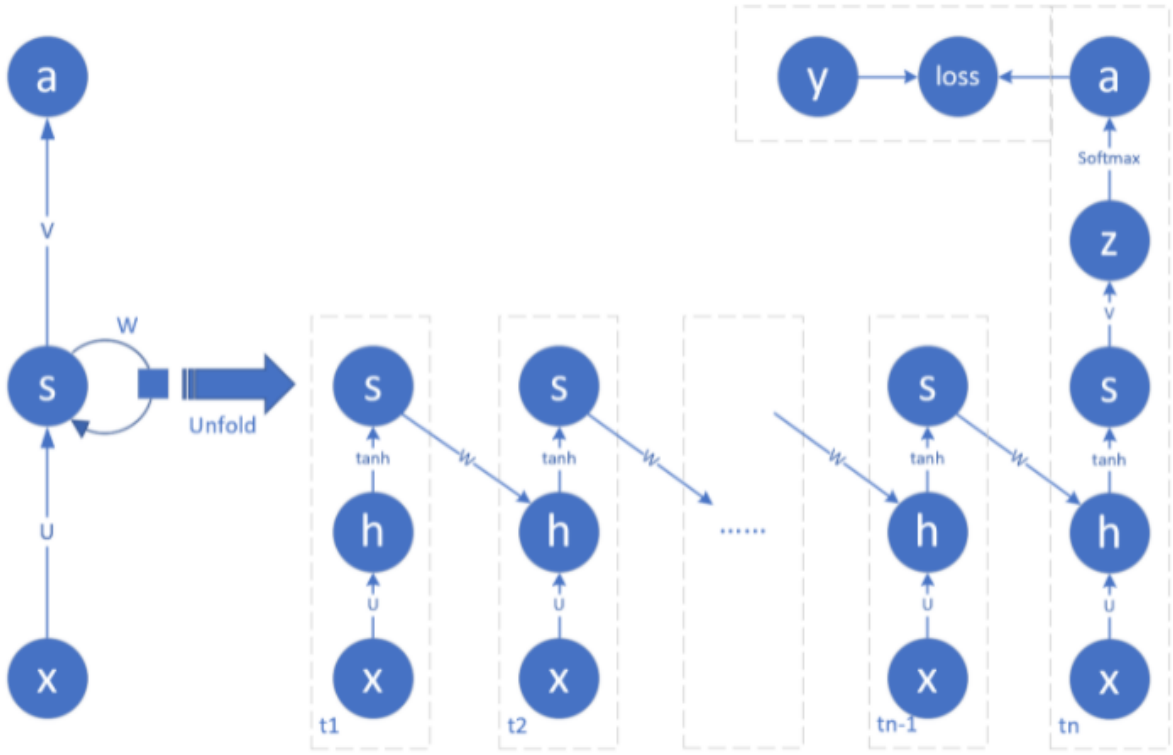
```
[[0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],  # d
 [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0],  # u
 [1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],  # a
 [0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0]]  # n
```

3. 把所有相同长度的名字的OneHot编码都堆放在一个矩阵中，形成批量，这样就是成为了一个三维矩阵：
    - 第一维是名字的数量，假设一共有230个4个字母的名字，175个5个字母的名字，等等；
    - 第二维是4或者5或者其它值，即字母个数，也是时间步的个数；
    - 第三维是26，即a~z的小写字母的个数，相应的位为1，其它位为0。

2.1.5.1 搭建不定长时序的网络

搭建网络

为什么是不定长时序的网络呢？因为名字的单词中的字母个数不是固定的，最少的两个字母，最多的有19个字母。



并不是所有的时序都需要做分类输出，而是只有最后一个时间步需要。比如当名字是“guan”时，需要在第4个时序做分类输出，并加监督信号做反向传播，而前面3个时序不需要。但是当名字是“baevsky”时，需要在第7个时间步做分类输出。所以n值并不是固定的。

对于最后一个时间步，展开成前馈神经网络中的标准Softmax多分类。

前向计算

分类函数使用Softmax，损失函数使用多分类交叉熵函数：

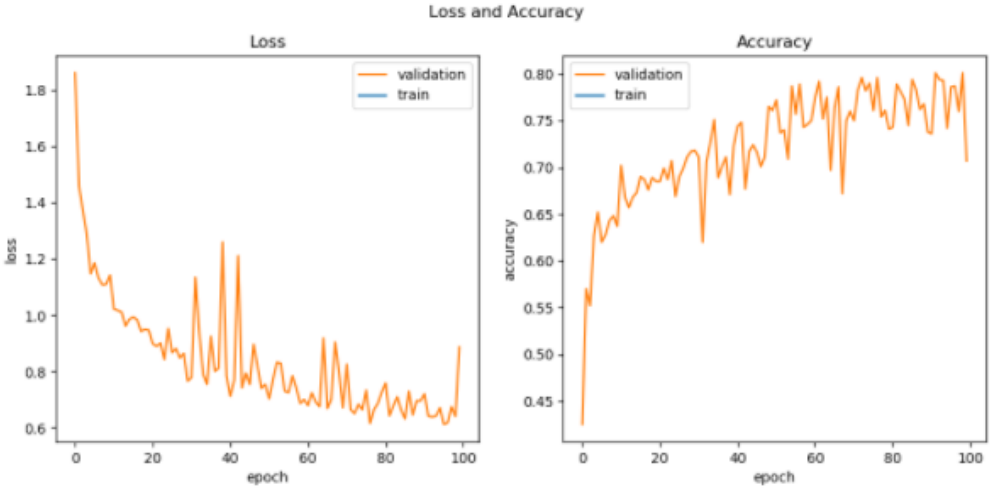
$$a = Softmax(z) \tag{1}$$

$$Loss = loss_\tau = -y \odot \ln a \tag{2}$$

反向传播

Softmax接多分类交叉熵损失函数

训练的结果为：可以看到两条曲线的抖动都比较厉害，此时可以适当地降低学习率来使曲线平滑，收敛趋势稳定。



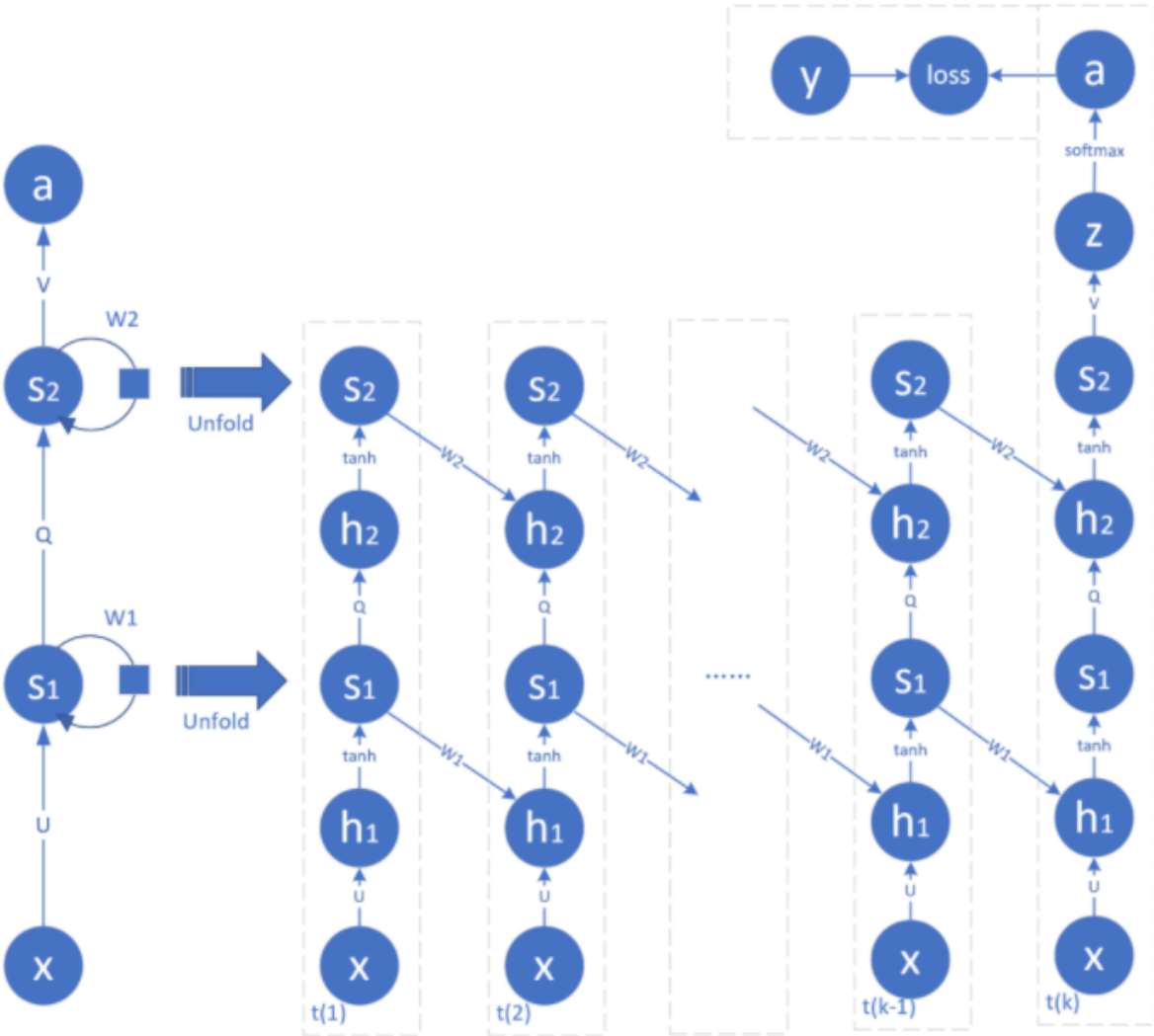
由于是多分类任务，我们可以用混淆矩阵来分析结果：**对角线上的方块越亮，表示识别越准确。**

最后的效果	最好的效果
准确率为67.9%的混淆矩阵	准确率为73.9%的混淆矩阵

2.1.6 深度循环神经网络

前面的几个例子中，单独看每一时刻的网络结构，其实都是由“输入层->隐层->输出层”所组成的，这与前馈神经网络中的单隐层的知识一样，由于输入层不算做网络的一层，输出层是必须具备的，所以网络只有一个隐层。但是单隐层的能力是有限的，所以人们会使用更深（更多隐层）的网络来解决复杂的问题。

两个隐层的循环神经网络的图示如下：

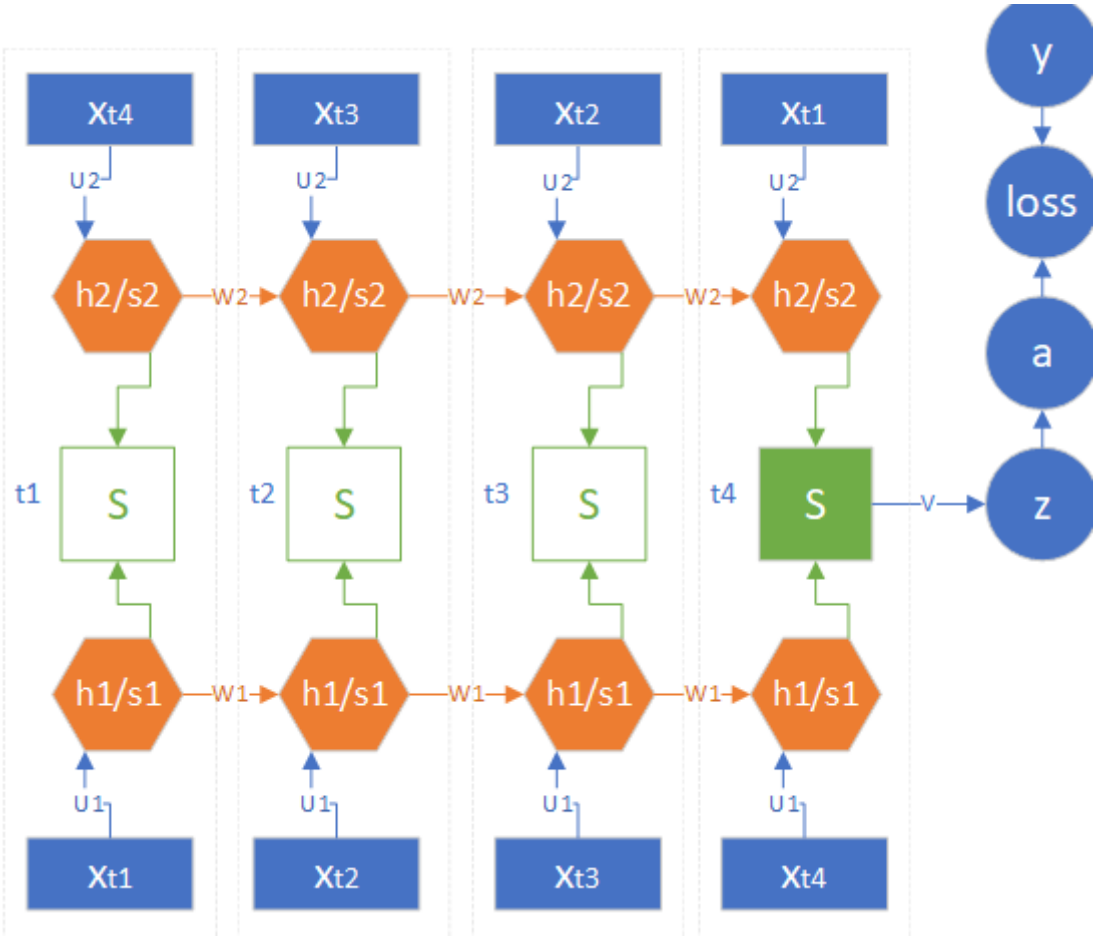


经过验证：双层的循环神经网络在参数少的情况下，取得了比单层循环神经网络好的效果。

2.1.7 双向循环神经网络

之前，学习的都是因为“过去”的时间步的状态对“未来”的时间步的状态有影响，但是存在很多都是双向影响的结构，比如：比如在一个语音识别的模型中，可能前面的一个词听上去比较模糊，会产生多个猜测，但是后面的词都很清晰，于是可以用后面的词来为前面的词提供一个最有把握（概率最大）的猜测。再比如，在手写识别应用中，前面的笔划与后面的笔划是相互影响的，特别是后面的笔划对整个字的识别有较大的影响。

图示如下：



用$h1/s1$表示正向循环的隐层状态，$U1$、$W1$表示权重矩阵；用$h2/s2$表示逆向循环的隐层状态，$U2$、$W2$表示权重矩阵。s 是 h 的激活函数结果。

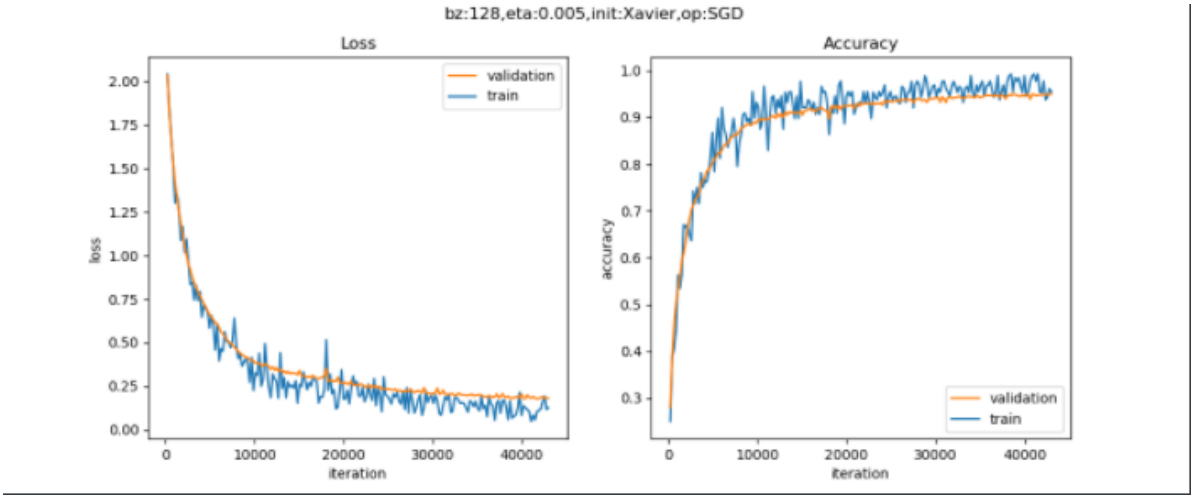
请注意上下两组x_{t1}至x_{t4}的顺序是相反的：

- 对于正向循环的最后一个时间步来说，x_{t4} 作为输入，$s1_{t1}$是最后一个时间步的隐层值；
- 对于逆向循环的最后一个时间步来说，x_{t1} 作为输入，$s2_{t4}$是最后一个时间步的隐层值；
- 然后 $s1_{t4}$ 和 $s2_{t4}$ 拼接得到 s_{t4}，再通过与权重矩阵 V 相乘得出 Z。

在超参数设置为：

```
eta = 0.01
max_epoch = 100
batch_size = 128
num_step = 28
num_input = 28
num_hidden1 = 20          # 正向循环隐层神经元20个
num_hidden2 = 20          # 逆向循环隐层神经元20个
num_output = 10
```

得到的结果如下：



最好的时间点的权重矩阵参数得到的准确率为95.59%，损失函数值为0.153259。

2.1.8 高级循环神经网络

2.1.8.1 传统循环神经网络的不足

但传统的循环神经网络也有自身的缺陷，由于容易产生梯度爆炸和梯度消失的问题，导致很难处理长距离的依赖。传统神经网络模型，不论是一对多、多对一、多对多，都很难处理不确定序列输出的问题，一般需要输出序列为1，或与输入相同。在机器翻译等问题上产生了局限性。

2.1.8.2 长短时记忆网络（LSTM）

长短时记忆网络（LSTM）是最先提出的改进算法，由于门控单元的引入，从根本上解决了梯度爆炸和消失的问题，使网络可以处理长距离依赖。

2.1.8.3 门控循环单元网络（GRU）

LSTM网络结构中有三个门控单元和两个状态，参数较多，实现复杂。为此，针对LSTM提出了许多变体，其中门控循环单元网络是最流行的一种，它将三个门减少为两个，状态也只保留一个，和普通循环神经网络保持一致。

2.1.8.4 序列到序列网络（Sequence-to-Sequence）

LSTM与其变体很好地解决了网络中梯度爆炸和消失的问题。但LSTM有一个缺陷，无法处理输入和输出序列不等长的问题，为此提出了序列到序列（Sequence-to-Sequence, 简称Seq2Seq）模型，引入和编码解码机制（Encoder-Decoder），在机器翻译等领域取得了很大的成果，进一步提升了循环神经网络的处理范围。