**学习进度**

本周学习任务：

学习吴恩达深度学习课程v5.72，关于循环序列模型RNN的内容。

1. 初识循环神经网络

**循环神经网络（Rerrent Neural Network, RNN）**作为一种典型的序列模型，擅长挖掘序列数据（序列数据是指符合时间、逻辑或其它顺序特性的数据）中的时序及语义信息。RNN及其变体，除了能够适应输入、输出数据长度的不一致性外，还能共享从序列不同位置上学到的特征。相较于传统神经网络模型，RNN能够结合输入序列的上下文来训练模型。因此，序列模型在语音识别、自然语言处理及推荐系统等领域，得到了广泛应用。

1. RNN网络架构

RNN的**网络架构**，如图2.1所示。若摒弃图中的W，则输入向量x、输入到隐藏层权重矩阵U、隐藏层向量s、隐藏到输出层权重矩阵V及输出向量o，构成了一个传统的全连接神经网络架构。若将RNN网络架构按照时间线展开，可知W是指每个时间点间的权重矩阵，分别指输入序列中不同位置上的词向量，为隐藏层第t时间步的状态，为第t时间步的输出（）。为结合前文信息处理序列数据，记忆单元根据当前输入层的输出与上一步隐藏层的状态进行计算，公式为：。此外，由于每次计算都需要前一时间步的状态值，因此计算时，一般将不存在的置为零向量。值得注意的是，基于序列顺序的每次时间步计算都可以共享参数U、V、W，从而极大地减少了网络可学习参数的数量。

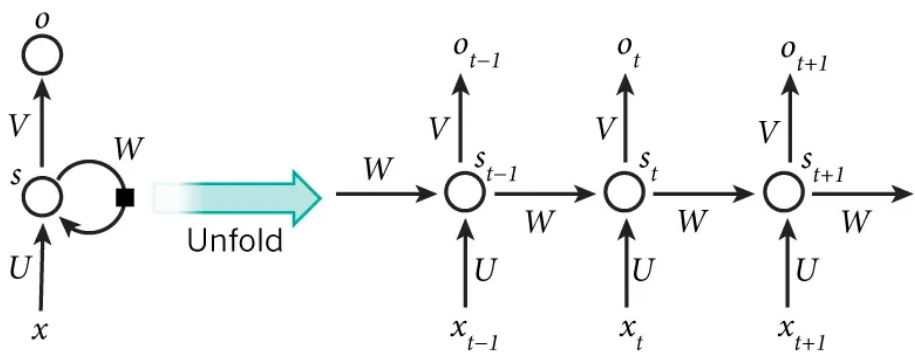


图2.1 RNN网络结构

问题是：为什么**可以**共享参数U、V、W？参数U、V、W相当于句子的语法结构或一般规律，参数作为一般规律的具象体现，在语言中也自然是一致且共享的。值得注意的是，很多深度学习算法利用参数共享，极大地减少了参数数量。比如：CNN在空间上共享参数，RNN在序列顺序上共享参数。然而，RNN是否共享参数或者共享哪些参数，往往取决于具体的设计及任务需求，共享参数U、V、W并不是必要条件。

1. 前、反向传播过程

如前所述，**RNN前向传播**是一个迭代的过程。基于输入序列的每个时间步，都计算当前时间步的隐藏状态，然后依据隐藏状态生成输出，同时将隐藏状态传递到下一时间步，直至处理完整个序列。隐藏状态在每个时间步中依赖于当前时间步的输入和历史隐藏状态，使得RNN能够捕捉序列数据中的上文信息和依赖关系。

**RNN反向传播**过程，是为了计算损失函数关于网络参数（、、b）的梯度，从而依据梯度下降等优化算法来更新可学习参数以最小化损失。

其中，损失函数可选用交叉熵损失函数（Cross Entropy Loss），公式如下：



则，目标函数可定义为：



反向传播过程如图3.1所示：

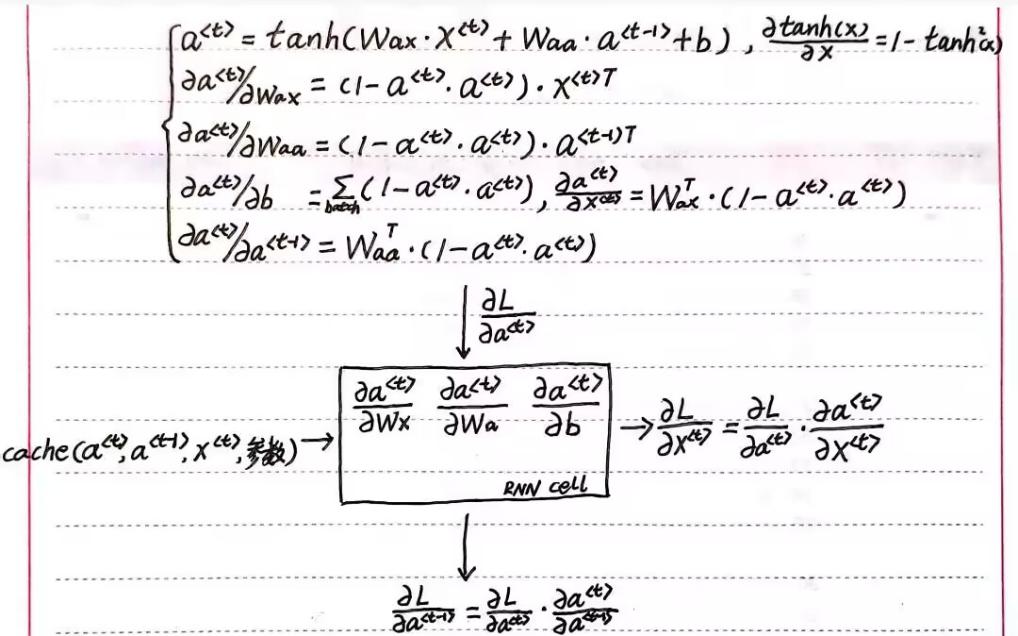


图3.1 反向传播过程

1. RNN变体

RNN的常见变体，如图4.1所示。对于**One to one**，若去掉则为标准的单个神经网络。**One to many**把单一输入x转化为序列输出，实现方式有两种：将x只输入到第一个时间步中、将x输入到每个时间步中，常见应用是音乐生成。**Many to one**则接收一个输入序列并返回一个单独的值，常见应用是电影分类。而**Many to many（）**输入输出的序列等长，是前述中的默认情景。

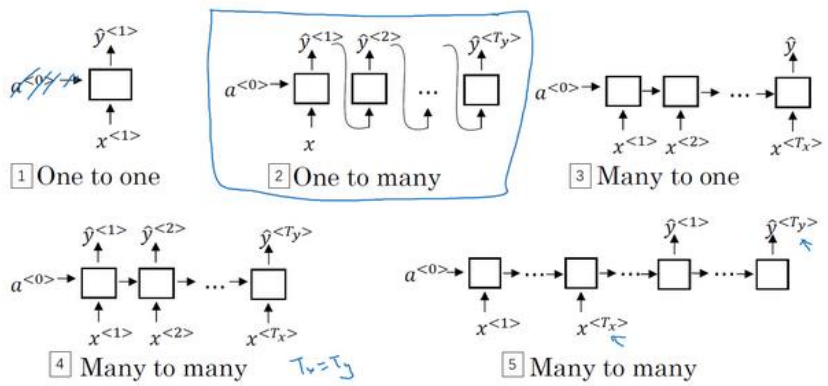


图4.1 不同类型的RNN

然而，在诸如机器翻译、文本摘要及对话生成等应用场景中，输入输出序列是不等长的。如图4.2所示，**Many to many（）**作为一种Encoder-Decoder结构的RNN变体，一般先将输入序列集成编码成一个上下文向量c，然后通过上下文向量c输出预测序列。

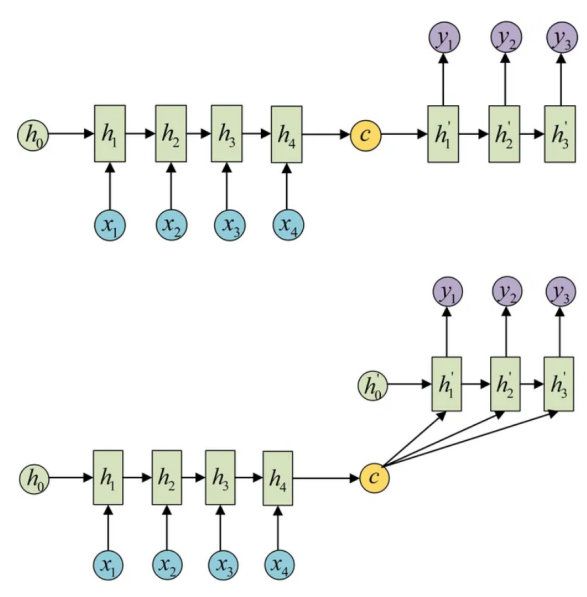


图4.2 输入输出序列不等长时的RNN变体

1. 长期依赖问题

当输入序列相当长时，很难通过反向传播将影响传递到足够远的位置，极易发生梯度衰减现象。因此传统RNN模型，不擅长捕获长期依赖效应。为解决长期依赖问题，GRU和LSTM应运而生...