深入理解RNN

机器学习实验室 2020-10-02

以下文章来源于狗熊会 ，作者louwill

狗熊会

狗熊会，统计学第二课堂！传播统计学知识，培养统计学人才，推动统计学在产业中的...

深度学习

Author：louwill

From：深度学习笔记

深度学习以处理非结构化数据而著称。除了常见的应用在图像领域的CNN之外，对于语音和文本等序列型的非结构化数据，CNN的效果并不好。本讲介绍一种在自然语言处理等领域应用非常广泛的一种序列网络模型——循环神经网络（Recurrent Neural Network, RNN）。

从语音识别到自然语言处理

CNN致力于解决如何让计算机理解图像的问题，但仅仅是视觉层面，还远远谈不上人工智能。人工智能除了要具备视觉能力之外，还得具备听力和读写能力。先看机器如何听的问题，也就是深度学习在语音识别方面的应用。语音识别应该是日常生活中比较常见的深度学习应用了，例如，苹果的siri，阿里的天猫精灵智能音箱等等，大家可以轻而易举的生成一段语音数据，siri收到你的语音信号后，通过内置的模型和算法将你的语音转化为文本，并根据你的语音指令给出反馈。那么语音识别这么高级的技术适用于深度学习方法吗？当然可以。相较于图像三维矩阵的存在形式，我们先来看看语音在计算机中是以何种形态呈现的。

语音通常是由音频信号构成的，而音频信号本身又是以声波的形式进行传递的，一段语音的波形通常是一种时序状态，也就是说音频是按照时间顺序播放的。一段语音信号外形如图1所示。

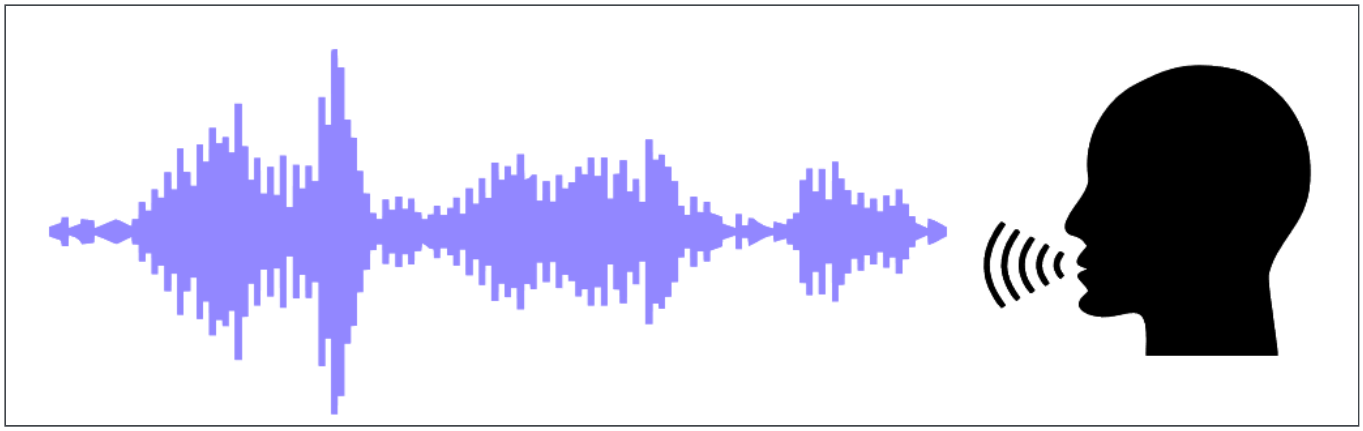

图1 语音信号

通过一些预处理和转换技术，我们可以将声波转换为更小的声音单元，即音频块。所以在语音识别的深度学习模型中，我们的输入就是原始的语音片段经过预处理之后的一个个音频块，这样的音频块是以序列形式存在的，所以输入是一个序列。那么输出呢？也就是咱们语音识别的结果是啥。语音识别的结果通常以一段文字的形式呈现，例如，siri会快速识别出你的语音指令并将识别结果以文字形式打印在手机屏幕上。这段文字也是一个按顺序排列的文本序列，所以我们的输出也是一个序列。那么如何建立由序列输入到序列输出之间的有监督机器学习模型呢？这便是RNN的要做的事。

能让机器会看懂图像、听懂语音还不够，最好还得能理解人类语言。所谓自然语言处理（Natural Language Processing，NLP），就是让计算机具备处理、理解和运用人类语言的能力。实际上，NLP的任务难度要远大于计算机视觉。人类语言的多样性、复杂性和歧义性，即使是一个国家、一个省份甚至一个地区大家说的语言都差之千里，我们自己都谈不上能充分理解人类语言，更何况去让机器理解？但虽说如此，但在基于深度学习的自然语言处理上，目前确实能够做到一定程度的让机器理解人类语言。

没有语言，我们的思维就无从谈起，那么对于机器来说，没有语言，人工智能永远都不够智能。所以从这个角度来说，自然语言处理代表了深度学习最高任务境界。虽说是最高境界，但也脱离不了监督机器学习基本范式。以NLP的一个应用案例——机器翻译来分析一下，看看基于深度学习自然语言处理问题是如何被规范为一个从输入到输出的有监督机器学习问题的。

相信不少朋友都用过机器翻译，谷歌翻译、百度翻译、有道翻译，可供选择的工具就有很多，将大段的英文复制粘贴到谷歌翻译中直接机翻的经历应该很多人都有过。在这样的一个问题里，模型输入毫无疑问就是一段待翻译的中文、英文或者是任意国家的文字，总的来说输入是由一个个单词或者文字组成的序列文本。那么作为翻译的结果，输出也是一个个单词或者文字

组成的序列文本，只不过换了一种语言，所以在机器翻译这样一个自然语言处理问题中，研究的关键在于如何构建一个深度学习模型来将输入语言转化为输出语言。可以看到，这个问题跟前面语音识别的例子很像，它们的输入输出形式都是序列化的。针对这样的序列建模问题，深度学习给出的网络方案和语音识别一样，都是循环神经网络。图2给出的是谷歌机器翻译的例子。

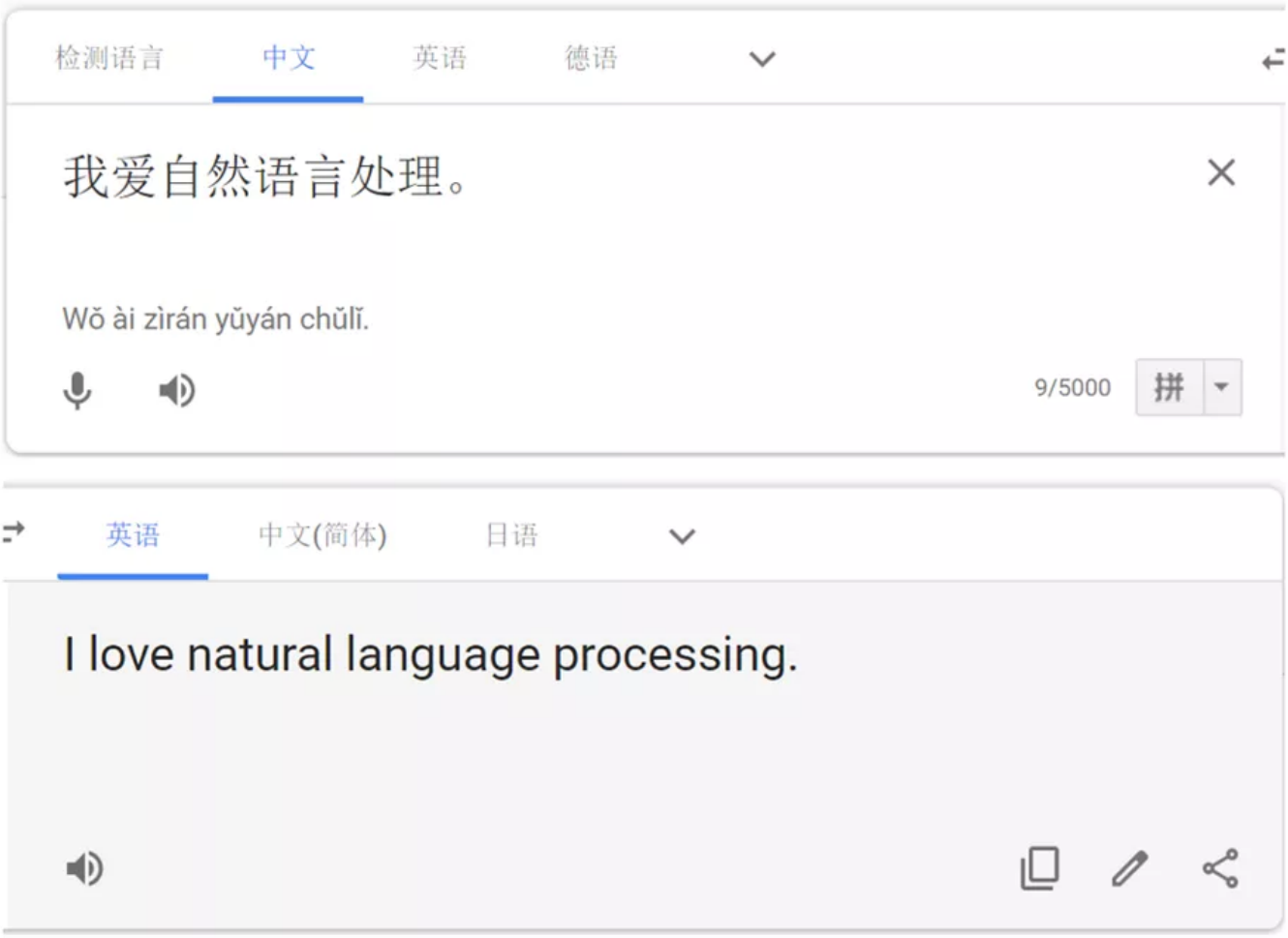

图2 谷歌翻译

对于博大精深的自然语言处理来说，机器翻译还仅仅是一个小的方向，除此之外，自然语言处理还包括很多有趣的研究与应用方向：句法语义分析、文本挖掘、信息检索、问答系统等等。但是不管是哪个方向的应用，只要它是属于监督机器学习性质的深度学习问题，我们都可以将其归纳为一个从输入到输出的有监督机器学习问题。

RNN：网络架构与技术

相较于DNN和CNN，RNN网络结构有什么特别之处？它与前两者又有哪些不一样的结构设计？在对RNN的结构进行深入了解之前，我们先对RNN的应用场景进行梳理。假设我们在进行语音识别时，给定了一个输入音频片段X，要求我们输出一个文本片段Y，其中输入X是一个按照时间播放的音频片段，Y是一个按照顺序排列的单词组成的一句话，所以在RNN中我们的

输入输出都是序列性质的。针对这样的输入输出（X,Y）的有监督学习，最适合的神经网络结构就是循环神经网络。为什么循环神经网络就最适用这种场景？

假设我们现在需要对输入的一段话识别其中每个单词是否是人名，即输入是一段文本序列，输出是一个每个单词是否是人名的序列。假设这段话有9个单词，我们将其转化为9个one-hot向量输入到标准神经网络中去，经过一些隐藏层和激活函数得到最终9个值为0/1 的输出。但这样做的问题有两个。

一是输入输出的长度是否相等以及输入大小不固定的问题。在语音识别问题中，输入音频序列和输出文本序列很少情况下是长度相等的，普通网络难以处理这种问题。

二是普通神经网络结构不能共享从文本不同位置上学到的特征，简单来说就是如果神经网络已经从位置1学到了louwill是一个人名，那么如果louwill出现在其他位置，神经网络就可以自动识别到它就是已经学习过的人名，这种共享可以减少训练参数和提高网络效率，普通网络不能达到这样的目的。

所以直观上看，普通神经网络和循环神经网络的区别如图3所示。

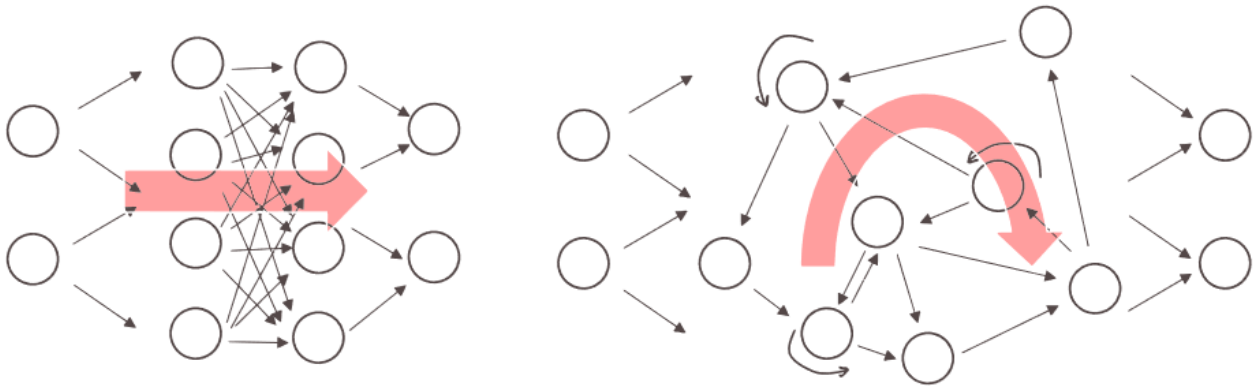

图3 普通网络和RNN在结构上的区别

那么RNN的具体结构是怎样的？

假设我们将一个句子输入RNN，第一个输入的单词就是，我们将输入到神经网络，经过隐状态得到输出判断其是否为人名，即输出为。同时网络初始化隐状态激活值，并在隐状态中结合输入进行激活计算传入到下一个时间步（Time Step）。当输入第二个单词的时候，除了使用预测输出之外，当前时间步的激活函数会基于上一个时间步的进行激活计算，即第二个时间步利用了第一个时间步的信息。这便是循环（Recurrent）的含义。如此下去，一直到网络在最后

一个时间步输出和激活值。所以在每一个时间步中，RNN传递一个激活值到下一个时间步中用于计算。

图4所示是循环神经网络的基本结构。左边是一个统一的表现形式，右边则是左边的展开图解。在这样的循环神经网络中，当我们在预测时，不仅要使用的信息，还要使用的信息，因为在横轴路径上的隐状态激活信息得以帮助我们预测。

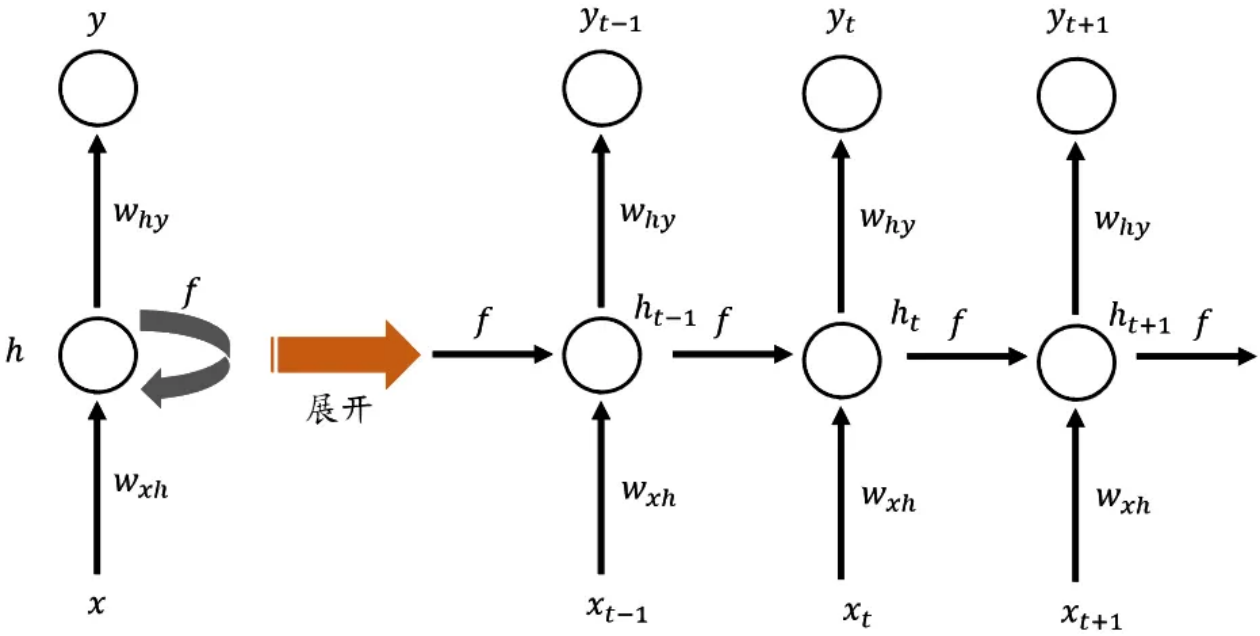

图4 RNN基本结构

所以，RNN单元结构通常需要两次激活运算，一次是结合上一个时间步的隐状态值和输入的计算，另一个是基于当前隐状态值的输出计算。一个RNN单元和两次计算如图5所示。

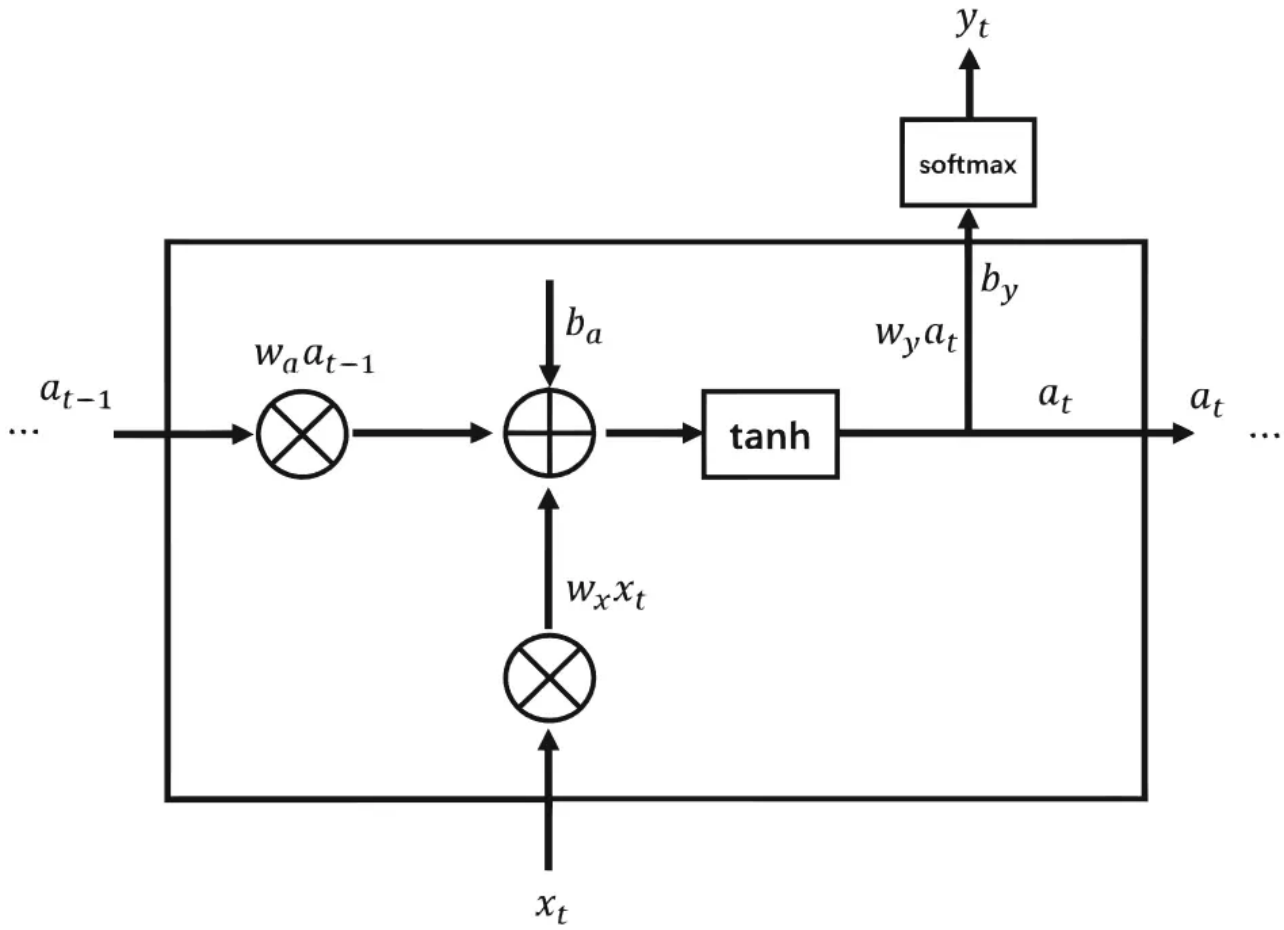

图5 RNN单元

两次激活运算公式如下：

$$a_t = \tanh\left(w_x x_t + w_a a_{t-1} + b_a\right)$$
$$y_t = \mathrm{softmax}\left(w_y a_t + b_y\right)$$

其中隐藏层的激活函数一般采用tanh，而输入输出的激活函数一般使用sigmoid或者softmax函数。当多个这样的RNN单元组合到一起便是RNN结构。一个RNN结构如图6所示。

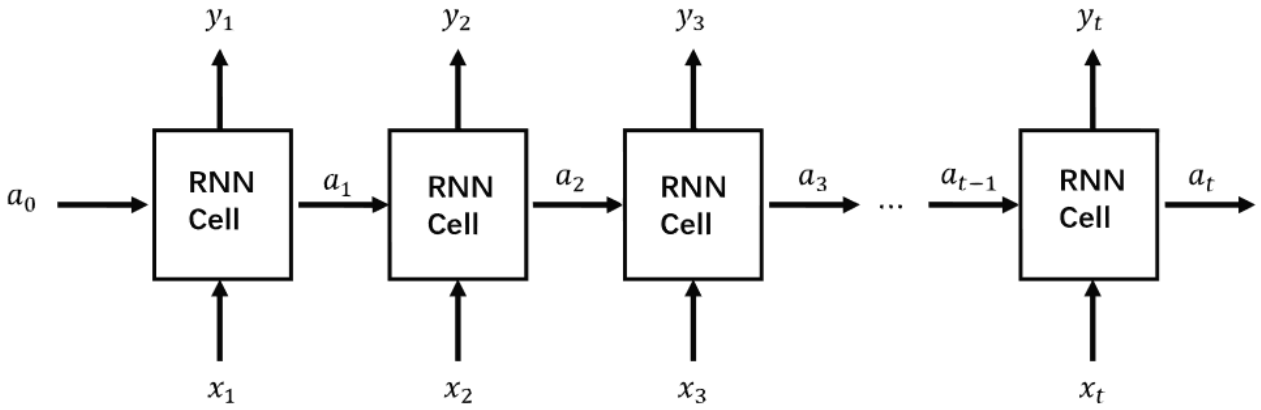

图6 RNN结构

这样的带有时间和记忆属性的神经网络模型使得深度学习可以胜任语音识别和自然语言处理等序列建模问题。

四种RNN结构

以上是RNN最基本的结构形式，但NLP等序列建模问题多样且复杂，基础的RNN结构并不够用，在初始RNN结构的基础上，针对多种不同的任务类型，RNN可以分为表1中的4种类型。

表1 4种RNN结构

结构类型	简称	适用任务场景
一对多	1 VS N	根据类型生成对应音乐或图像等
多对一	N VS 1	情感分析、文本分类等
多对多（等长）	N VS N	视频每帧分类等
多对多（不等长）	N VS M	机器翻译、语言识别等

下面就简单来介绍一下这4种RNN结构。

首先是一对多结构。所谓一对多就是指RNN只有一个输出，但却有多个输出的情形，也即输入为单一值，输出为一个序列，其结构如图7所示。一对多结构在音乐生成、图像生成或视频生成等方面有着广泛的应用。指定一种类型，要神经网络生成这个类型的音乐，这是一种较为常见的应用。

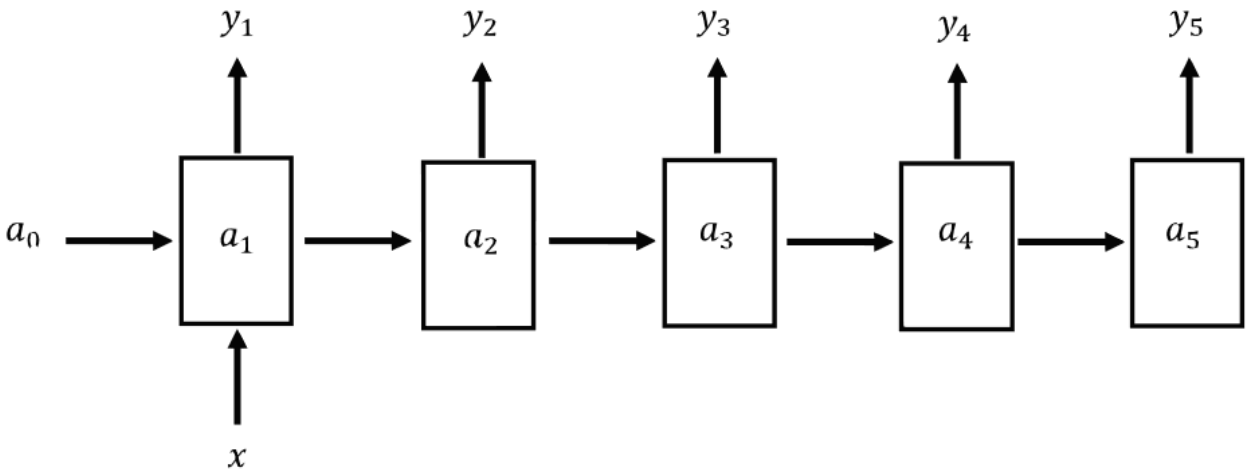

图7 1 VS N结构

与一对多相对应的则是多对一结构，多对一正好跟一对多输入输出调换了一下，即有多个输入，但仅有一个输出。这种结构也有广泛的应用场景，例如，对电影评论的情感分析，就是一个简单的文本分类问题，输入有多个，但输出只有一个类别标签。多对一结构如图8所示。

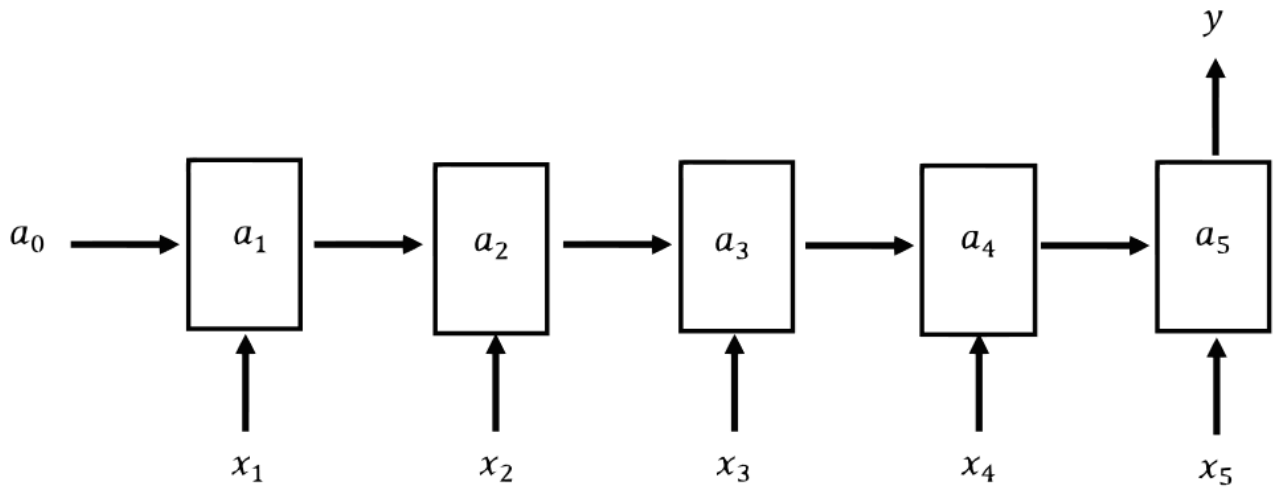

图8 多对一结构

最后是多对多结构，即输入输出都是多个的情形。但多对多又可以分为输入输出等长和不等长两种情形。等长的多对多也就是我们前面提到的经典的RNN结构，有多少输入就是多少输出，这个限制使得等长RNN结构在实际应用中并不广泛，但也有一些应用场景是多对多的，例如，对视频进行逐帧分类，每一帧都打一个标签，这就是一种等长的多对多结构。等长多对多结构如图9所示。

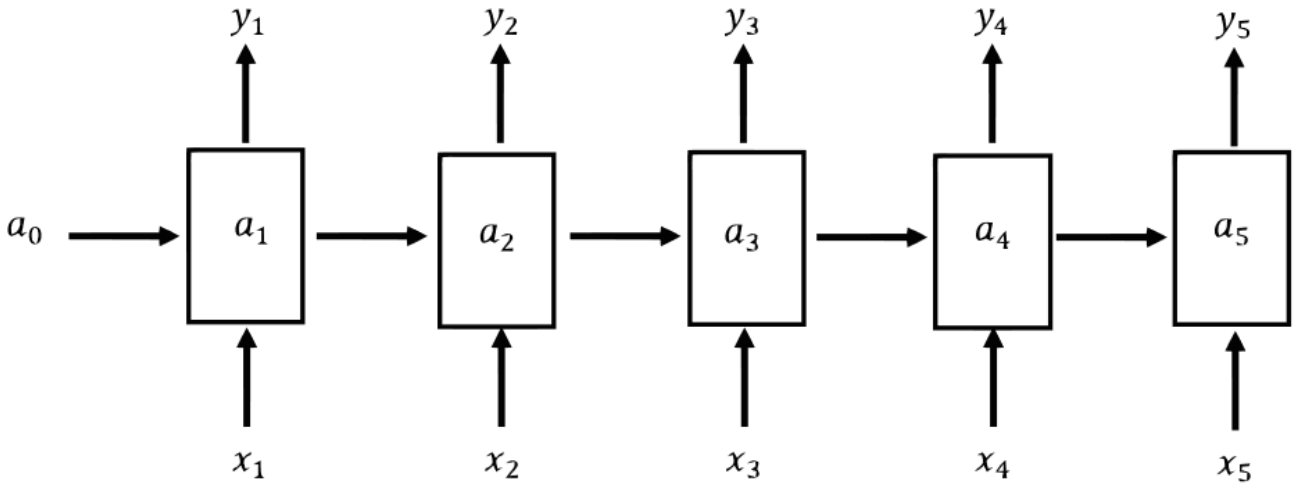

图9 等长多对多结构

最常见的是不等长的多对多结构，即输入输出虽然都是多个，但并不相等。这种不等长的输入输出模型也叫作seq2seq（序列对序列）模型，不等长的多对多结构符合实际序列建模的大多数情况，很多时候我们的输入输出序列并不等长，例如，我们进行汉译英的机器翻译，输入汉

语句子和输出英文句子基本不会等长。对于这种情况，RNN的做法通常是先将输入序列编码成一个上下文向量C，如图10所示。具体的编码过程到本书的第15讲再进行详述，本讲了解编码结构即可。

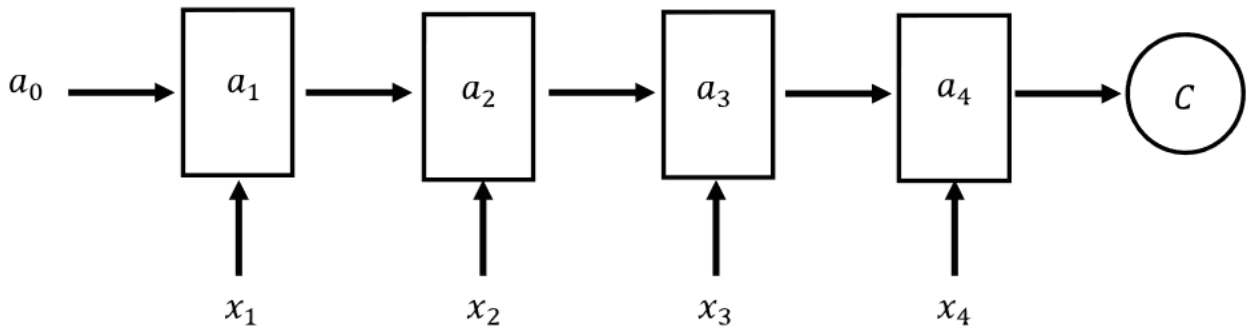

图10 多对多编码结构

编码完成后我们再用一个RNN对C的结果进行解码，简而言之就是将C作为初始状态的隐变量输入到解码网络，如图11所示。

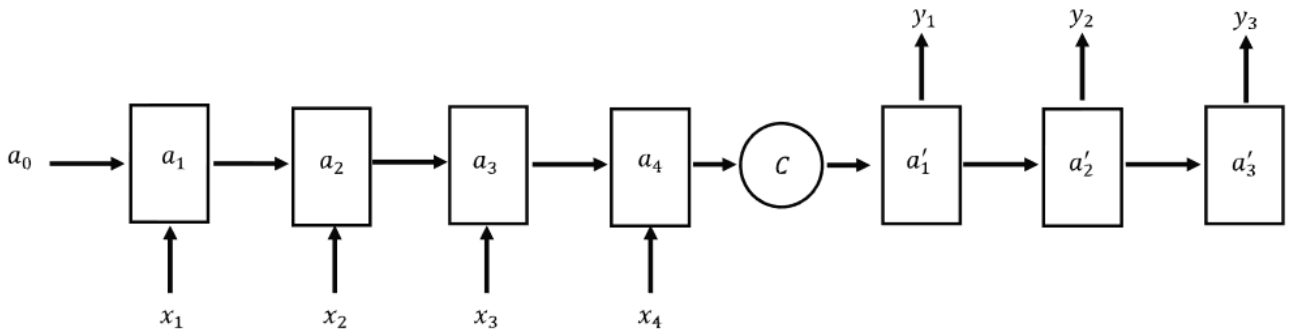

图11 编码+解码结构

不等长的多对多结构因为对输入输出长度没有限制，因而有着特别广泛的应用，主要包括语音识别、机器翻译、文本摘要生成和阅读理解等。

往期精彩：

【原创首发】机器学习公式推导与代码实现30讲.pdf

【原创首发】深度学习语义分割理论与实战指南.pdf