

- 5.0 前言
 - 5.0.1 课程目标
 - 5.0.2 课程概览
- 5.1 序列预测
- 5.2 序列预测的模型
 - 5.2.1 One-to-One模型
 - 5.2.2 One-to-Many模型
 - 5.2.3 Many-to-One模型
 - 5.2.4 Many-to-Many模型
- 5.3 将模型映射到应用
 - 5.3.1 时间序列
 - 5.3.2 自然语言处理
- 5.4 来自时间步长的基数（不是特征！）
- 5.6 两个经常的误解
 - 5.5.1 时间步长作为输入特征
 - 5.5.2 时间步长作为输出特征
- 5.6 扩展阅读
 - 5.6.1 论文
- 5.7 扩展
- 5.8 总结

5.0 前言

5.0.1 课程目标

本节课的目的是让大家了解4个序列预测模型以及如何在Keras中实现它们。完成这一课之后，你会知道：

- 序列预测的4中模型及其在Keras中的实现；
- 如何将4个序列预测模型映射到共同的和有趣的预测预测问题中；
- 初学者在序列预测问题中遇到的陷阱以及怎么样避免它们。

5.0.2 课程概览

本课程分成5个部分，它们是：

1. 序列预测；
2. 序列预测的模型；
3. 模型的映射应用；
4. 时间步长的基数；
5. 两个常见的错误。

让我们开始吧！

5.1 序列预测

LSTMs通过学习函数（ $f(\dots)$ ）来工作，该函数将输入序列值（ X ）映射到输出序列值（ y ）上。

$$y(t) = f(X(t))$$

表 5.1 一般LSTM模型的例子

所学到的映射函数是静态的，可以看做是一个接受输入变量和使用内部变量的程序。内部变量可以被表示为由网络所维持和建立或者在输入序列的每个值上所累积的内部状态。静态映射函数可以用一个不同数量的输入或者输出定义。本章节主要关注这个重要的细节。

5.2 序列预测的模型

在本章节中，将会看到4个序列预测的基本模型。我们将会用到下面的术语：

- x ：输入序列值；可能会被一个时间步限定限定，例如 $x(1)$ ；
- u ：隐藏状态值；可能会被一个时间步长所限制，例如 $u(1)$ ；
- y ：输出序列值；可能会被一个时间步长所限制，例如 $y(1)$ 。

每个模型将会用这些术语、图片和Keras中的代码例子进行解释。重点关注学习不同类型的序列预测问题如何映射到不同类型的模型。不要太关注Keras实例的细节，因为整个章节都致力于解释更复杂的模型类型。

5.2.1 One-to-One模型

一个one-to-one模型（ $f(\dots)$ ）为每一个输入值（ $x(t)$ ）产生一个输出（ $y(t)$ ）值。

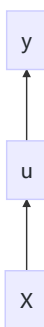


图 5.1 one-to-one序列预测问题

例如：

$$y(1) = f(X(1))$$

$$y(2) = f(X(2))$$

$$y(3) = f(X(3))$$

...

表 5.2 one-to-one序列预测模型的例子

第一个时间步长的内部状态是0,; 从这一点开始, 内部状态被累计在先前的时间步长上。

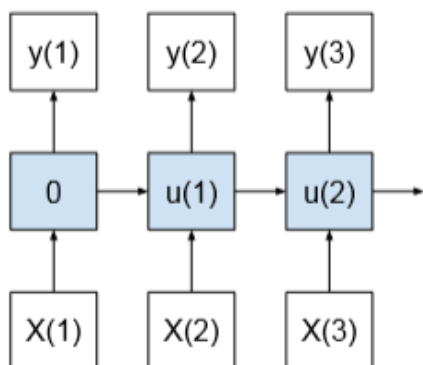


图 5.2 基于时间的one-to-one序列预测模型

这个模型对于序列预测问题是合适的, 这里我们希望给定1个时间步长作为输入的时候预测1个时间步长。例如:

- 时间系列中预测下一个真是的值;
- 在一个句子中预测下一个单词;

这是一个对LSTM不好的使用, 因为它不能学习输入或者输出的时间步长。该模型把所有的压力都放在了内部状态或者存储上。模型的结果可以用来建造一个模型, 该模型有输入或输出序列以便查看通过时间步长学习是否会对预测有积极的作用。我们可以通过定义一个1个时间步长输入以及在输出层预测1个时间步长的网络来将其在Keras中应用。

```
model = Sequential()
model.add(LSTM(..., input_shape=(1, ...)))
model.add(Dense(1))
```

表 5.3 在Keras中定义一个one-to-one序列预测模型的例子

在第11章中介绍了one-to-oneLSTM模型用于序列的生成。

5.2.2 One-to-Many模型

一个one-to-many模型 ($f(\dots)$) 对于输出值 ($x(t)$) 产生多个输出值 ($y(t)$, $y(t+1)$, ...)。例如:

$$y(1), y(2) = f(X(1))$$

表 5.4 one-to-many序列预测模型的例子

考虑时间对于输入和输出的单独计算可能会有帮助。

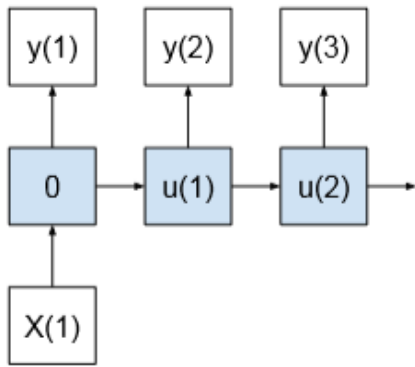


图 5.3 one-to-many序列预测模型

当输出序列中的每个值被产生的时候，内部状态就被积累。该模型适合于序列预测问题，我们希望为每个输入时间步长产生序列输出。例如：

- 从一个单一的图片中预测一个序列的词；
- 从一个单一事件中预测一系列的观察值；

该模型一个很好的例子是为图片生成一个文字的标题。在Keras中，这需要一个卷积神经网络来从图像中抽取特征，然后接着由LSTM模型来一次性输出词的序列。输出层每个输出时间步长预测一个观察值，并且为了使用相同的输出层多次输出所需要数量的时间步长，该输出层被包裹在 `TimeDistributed` 包裹层中。

```
model = Sequential()
model.add(Conv2D(...))
...
model.add(LSTM(...))
model.add(TimeDistributed(Dense(1)))
```

表 5.5 Keras中定义one-to-many序列预测模型例子

第八章中的序列分类例子可以使用one-to-many的模型。例如，为了描述线从图像的上、下、左和右运动。

5.2.3 Many-to-One模型

一个many-to-one模型（ $f(\dots)$ ）在收到多个输入值（ $x(t)$ ， $x(t+1)$ ，...）产生一个输出（ $y(t)$ ）值。例如：

$$y(1) = f(x(1), x(2))$$

表 5.6 many-to-one序列预测模型例子

同样，在输入序列和输出序列中考虑时间被单独计算是有帮助的。

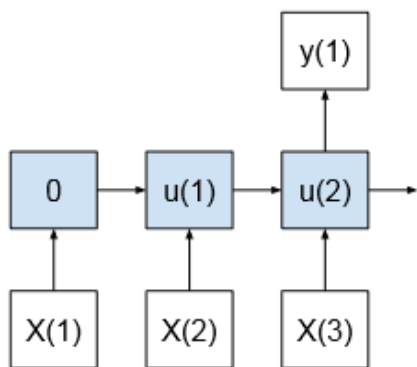


图 5.4 many-to-one序列预测模型

在产生一个输出值之前，内部状态与每个输入值一起累积。该模型适用于序列预测问题，其中需要多个输入时间步长，以便进行进一步预测。例如：

- 在给定一个输入观察值序列时，在时间系列中预测下一个真实值；
- 为词输入序列预测分类标签，例如情感分析。

这个模型就像one-to-one模型一样可以在Keras中实现，输入时间步长所期望的数目可以被变化为问题所需要的样子。

```
model = Sequential()
model.add(LSTM(..., input_shape = (steps, ...)))
model.add(Dense(1))
```

表 5.7 在Keras模型中定义一个many-to-one序列预测模型的例子

通常，当实践者实施一对一模型时，他们实际上打算实现多对一模型，例如当在时间序列预测的情况下。在第六章中一个many-to-one模型开发来用于序列预测问题，第七章中用来预测一个单向量输出，以及第八章中用来做序列分类。

5.2.4 Many-to-Many模型

一个many-to-many模型（ $f(\dots)$ ）在接收到多个输入值（ $x(t)$ ， $x(t+1)$ ，...）产生多个输出（ $y(t)$ ， $y(t+1)$ ，...）。例如：

$$y(1), y(2) = f(x(1), x(2))$$

表 5.8 many-to-many序列预测模型的例子

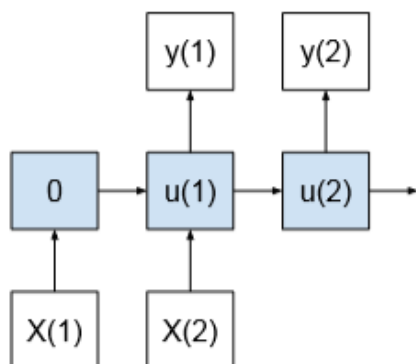


图 5.5 many-to-many序列预测模型

与多对一的情况一样，状态被积累到第一次输出被创建。但是在这种情况下，输出多个时间步长。重要的是，输入时间步长的数目不必与输出时间步长的数目相匹配。该模型适用于序列预测，其中需要多个输入时间步长以预测输出时间步长序列。这些通常被称为序列到序列，或seq2seq，类型问题，并且最近可能是主要使用LSTMs进行学习的问题。例如：

- 将一个文档的词汇总结到一个短序列词语中；
- 将一个序列的音频数据分类到一个序列的词语中。

```
model = Sequential()
model.add(LSTM(..., input_shape=(steps, ...), return_sequences = True))
model.add(TimeDistributed(Dense(1)))
```

表 5.9 用相同长度的输入和输出序列在Keras中定义一个many-to-many序列预测模型的例子

如果输入和输出时间步长的数目不同，则可以使用Encoder-Decoder结构。将输入的时间步长映射到序列中的一个固定大小的内部表示，然后使用该向量作为输入来产生输出序列的每个时间步长。

```
model = Sequential()
model.add(LSTM(..., input_shape = (in_steps, ...)))
model.add(RepeatVector(out_steps))
model.add(LSTM(..., return_sequences = True))
model.add(TimeDistributed(Dense(1)))
```

表 5.10 通过不同长度的输入和输出序列在Keras中定义一个many-to-many序列预测问题

这可能是最为复杂的序列预测模型，具有许多变化和优化的探索。第九章中开发了一个many-to-many模型，其输入和输出序列有不同数目的时间步长（例如序列长度）。第10章中也开发了一个many-to-many模型，其输入和输出序列具有相同数量的时间步长。

5.3 将模型映射到应用

我真的希望你理解这些模型，以及如何把你的问题作为上述4种类型之一。为此，本节列出了10种不同的序列预测模型问题，并指出了哪些模型可以用来解决这些问题。在每一个解释中，我给出了一个可以用来解决问题的模型的例子，但是如果重新需要预测序列预测问题，则可以使用其他的模型。把这些当做最好的建议，而不是牢不可破的规则。

5.3.1 时间序列

- **单变量预测。**这是一个具有多个输入时间步长的系列，并希望预测超出输入序列的一个时间步长。这可以被多对一模型实现。
- **多变量预测。**这是多个具有多个时间步长的系列，并希望在一个或者多个输入时间序列的基础上预测一个时间步长。这可以实现many-to-one模型。每个系列就是另一个输入特征。
- **多步长时间序列预测。**这是一个或者多个具有时间步长的系列，并希望在一个或者多个输入序列上预测多个时间步长。这可以被many-to-many的模型实现。
- **时间序列分类。**这是一个或者多个具有多个时间步长的输入的系列，并希望输出一个分类标签。这个可以被many-to-one模型实现。

5.3.2 自然语言处理

- **图像标题**。这是你具有一张图片，并且希望生成一个文本描述。这可以被one-to-many模型实现。
- **视频描述**。这是你具有视频中一个系列的图片，并且希望生成一个文本的描述。这可以被many-to-many模型实现。
- **情感分析**。这里你具有一系列的文本作为输入，并希望生成一个分类标签。这可以被many-to-one模型实现。
- **语音识别**。这里你具有一个序列的语音数据作为输入，并希望生成他们所说内容的文本描述。这可以被many-to-many模型实现。
- **文本翻译**。这里你具有一种语言的一个序列的单词作为输入，希望生成另一种语言的一个序列的输出。这可以被many-to-many模型实现。
- **文本摘要**。这里你具有一个文档的文本作为输入，并希望产生该文档的短文本摘要作为输出。这可以被many-to-many模型实现。

5.4 来自时间步长的基数（不是特征！）

混淆的一个共同点是上述序列映射模型与多个输入和输入特征相混淆。序列可以由单个值组成，每个时间步长一个值。

或者序列可以被简单地表达为时间步长上多个观察值的一个向量（例如，在给定时间步长，根据 $[X1, X2]$ 来预测 $[y1, y2]$ ）。向量在时间步长中的每个item可以被认为是它自己的单独的时间序列。它没有对上述模型描述。例如，输入温度（ $X1$ ）和压力（ $X2$ ）的一个时间步长的模型并预测温度（ $y1$ ）和压力（ $y2$ ）的一个时间步长是one-to-one的模型，而不是many-to-many模型。

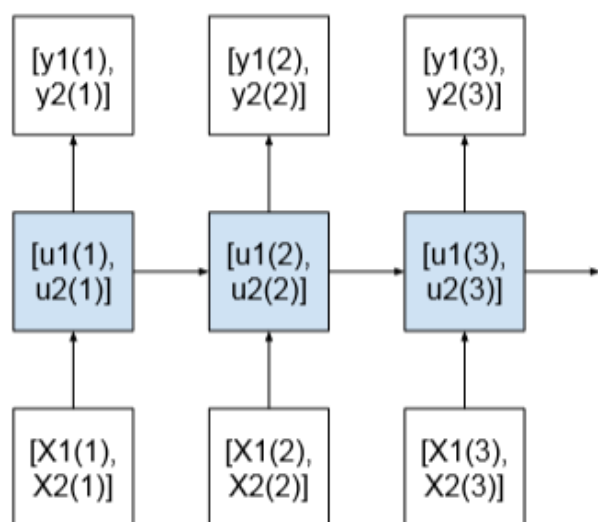


图 5.6 多特征序列预测模型

模型可以接受两个值作为输入，具有两个分别的内部状态，并且预测两个值，并且预测两个值，但是只有一个序列的时间步长表示输入，并作为输出预测。上述序列预测模型的基数是指时间步长，而不是特征。

5.6 两个经常的误解

特征vs时间步长的混淆导致了实践者在实现LSTMs模型的时候的两个主要的误解。

5.5.1 时间步长作为输入特征

在以前的时间步长滞后观察值被界定为模型的输入特征。这是感知机输入序列预测问题中经典的fixed-window-based方法。相反地，该序列应该作为多输入时间步长（例如many-to-one模型）呈现给模型。这种混乱可能会导致你认为自己已经实现了one-to-many或者many-to-many模型，实际上你只有一个单一的向量输入一个时间步长。

5.5.2 时间步长作为输出特征

在多个未来时间步长的预测被界定为模型的输出特征。这是感知机和其它机器学习算法中用于多步预测的经典的基于固定窗口的方法。相反，应该一次生成一个时间步长的模型。这种混乱可能会导致你认为你已经实现了one-to-many或many-to-many的序列预测模型，实际上你只有一个单一的向量输出一个时间步长（例如，seq2vec不是seq2seq）。

注意：帧时间步长作为序列预测问题的特征是一种有效的策略，并且及时在使用循环神经网络时也可以提高性能（试试它！）。这里重要的一点是了解常见的陷阱，而不是在自己预测问题的时候欺骗自己。

5.6 扩展阅读

这个章节提供了一些用于扩展阅读的资源。

5.6.1 论文

- [The Unreasonable Effectiveness of Recurrent Neural Networks, 2015.](#)

5.7 扩展

你是否想更加深入地了解序列预测模型？本节列出了本课程一些您可能需要考虑的具有挑战性的扩展：

- 总结在你自己的备忘录上总结这4个模型；
- 累出多变量时间序列可以被构造的所有方式和可以使用的二次预测模型；
- 为该模型选择一个模型一个应用程序，并用示例输入和输出序列绘制图表；
- 列出每个模型的序列预测问题的2个新实例；
- 在arXiv.org上找到5篇近期的关于LSTM的论文，列出标题并说明它们讨论了哪些序列预测模型。

将你的扩展发布在网上，并把链接分享给我。我很想知道你是怎么样想的。

5.8 总结

在本课程中，你发现了4个序列预测的模型，以及怎样在Keras中实现它们。具体来说，你学习到了：

- 序列预测的4个模型以及怎么样在Keras中实现它们；
- 序列预测问题的例子以及它们映射到哪4个模型；
- 在应用序列预测模型中初学者可能会遇到的陷阱以及怎么样避免它们；

在下一课中，我们会学习到Vanilla LSTM结构，你额可以将其用于绝大多数的序列预测问题。