

- 14.0 预览
 - 14.0.1 课程目标
 - 14.0.2 课程概览
- 14.1 新数据是什么样的？
 - 14.1.1 模型是否仍然有技能？
 - 14.1.2 我们可以恢复模型技能吗？
 - 14.1.3 我们能够提升模型技能吗？
- 14.2 什么是LSTM模型更新？
- 14.3 更新LSTM模型的5步过程
 - 14.3.1 收集新数据
 - 14.3.2 评估已经存在的模型
 - 14.3.3 开发更新模型
 - 14.3.4 评估候选模型
 - 14.3.5 替换模型
- 14.4 扩展
- 14.5 总结

14.0 预览

14.0.1 课程目标

本课程的目标是学习在新数据可用后怎么样更新LSTM模型。完成了本课程的学习之后，你将会知道：

- 对新数据的监控、恢复技能和提升模型学习能力的兴趣；
- 用新的数据更新一个新的LSTM模型的5步过程；
- 当开发具有新数据的更新了的LSTM模型时需要考虑的4个关键方法。

14.0.2 课程概览

本课程被分为3个部分，它们是：

1. 新数据怎么样？
2. 什么是LSTM模型更新？
3. 更新LSTM模型的5步过程。

让我们开始吧！

14.1 新数据是什么样的？

一旦你完成了你的LSTM模型，你就可以用它来做预测。但这并不是故事的结尾。几个月或几年之后，你将开始积累新数据的语料库。这将提出一些重要的问题。其中的第一个方面是：

14.1.1 模型是否仍然有技能？

重要的是要把这个问题作为你模型的持续维护的一部分来处理。一旦你监控模型技能，也许你会注意到，随着时间的推移，预测的技能正在下降。

14.1.2 我们可以恢复模型技能吗？

LSTM模型所解决的序列预测问题的性质可能随着时间而改变。一个模型只不过和用来训练它的数据一样好。如果用于训练那你的模型的数据也是一年前的，也许新的数据收集到一天会导致一个不同和更有技能的母校。也许你的模型预测和你第一次发开模型时一样有技能。

14.1.3 我们能够提升模型技能吗？

通过使用新数据提升你模型的技能是可能的。

14.2 什么是LSTM模型更新？

更新LSTM模型指的是充分利用新数据的技术来评估和提升一个已经存在和已经完成了的LSTM模型的技术。评估的目标是去评估新的或者更新了的候选模型是不是比以前完成了的模型更具有技能。这不意味着在老的训练数据集的现有模型上调优。它明确地指出如何最好地将新的数据合并到现有模型的更新中，该模型可能涉及或者不涉及模型参数的调优。

像所有的建模一样，当更新LSTM模型的时候系统化是很重要的。

14.3 更新LSTM模型的5步过程

更新一个已经存在的LSTM模型的包括5个关键步骤。它们是：

1. 收集新数据；
2. 评估已经存在的模型；
3. 开发更新模型；
4. 评估更新模型；
5. 替换模型。

让我们挨个来仔细看下它们。

14.3.1 收集新数据

如果没有新数据，则无法更新模型。这意味着在序列回归或者完全输入序列的情况下，在序列分类的情况下，完成完整和高质量的输入序列。

如果序列预测问题没有被改变，数据应该与原始模型开发有着相同的格式并以相同的方式被准备。理想情况下，访问用于训练现有模型的数据也将是有价值的。

你可能有数月或者数年的数据，也许比你能处理的更多。如果是这种情况，考虑重用你用来选择数据的方法来训练和评估现有模型。考虑选择序列的子样本，也许是最近的，也是每个时间间隔的一部分。访问大量数据（甚至太多）可能是有用的。我们可能或者不能不决定使用一些更新模型（一个缓慢的过程）。但是我们可以使用大部分或者全部来评估现有的和新的候选模型（一个很快的过程）。

14.3.2 评估已经存在的模型

监控你的模型预测的技巧是至关重要的。与评估候选模型一样，评估最终模型的性能需要访问预测的以及观察的真实值。具体来说，这意味着：

1. 预测值：你必须将一些或者所有由完成模型所作出的预测存储起来。
2. 观察值：你必须聚集和存储一些或者全部真的或者实际观察值。

给定模型预测值和观察值，可以评估最近数据的完成模型的技能，并通过适当的时间间隔，例如小时、日、周或者月来量化它。在很长一段时间内对模型的技能有一个认识是有用的，甚至回到模型的开始。这可能需要对模型进行重新测试，并重新生成未存储的任何预测。

随着时间的推移绘制模型技能将帮助你回答关于模型技能是否保持稳定或者退化的问题。它反过来告诉你你的项目是否持久化来恢复模型技能或者开发来提升模型技能。

- **降低模型技能。**你的目标是利用新的数据恢复模型技能回到历史水平。
- **稳定模型技能。**你的目标是通过使用新的数据来提升模型技能在稳定的水平上。
- **提高模型技能。**好！你的目标可能是调查新数据与旧模型中原始数据相比发生了什么变化。

14.3.3 开发更新模型

使用新数据有很多方法。下面的4个选项供你考虑：

- **在老的和新的数据上更新模型。**现在模型被加载和训练在附件的周期中，使用原始数据的混合来拟合模型（旧数据和新数据）。
- **在新的数据上开发新的模型。**一种新的模型是从头开始开发的，只适合于新的数据。
- **在新的和老的数据上开发新的模型。**在新旧数据上建立的一个新的模型。

你选择的具体方法可能取决于你的序列预测问题、你的具体实现，或者最终是候选模型的技巧。更新模型就像加载模型和运行额外的训练周期一样简单。在内部，模型是由一个结构（如何挂在一起）和权重（数组的数字）来定义的。通常，更新考虑到找到一个好的网络结构的工作已经得到解决，并且我们主要对网络权重的重新处理过程感兴趣。例如，下面是Keras如何更新权重的片段：

```
# Load model from file
model = ...
```

```
# access new data
newX, newY = ...
# fit model on new data
model.fit(newX, newY, ...)
```

表 14.1 更新已经存在的LSTM模型的例子

调优更新模型

在更新模型时，可能需要考虑调整选项。例如：

- **学习率。**可能需要一个小的学习率来对权重进行小的调整，而不是大的跳跃。
- **周期。**可能需要对新序列进行少量的迭代来将权重拨入新数据。
- **样本。**可能只需要最近一天、周、月或者年的最新样本才能在模型中进行拨号。

考虑这些搜索和其他关注点的网格搜索，重点是将权重调整到新数据，而不是用新的和非常不同的权重来批量替换它们。

新模型调优

如果你决定从开头开始开发一个新的模型，那么整个模型选择和超参数调优会对你开放。这可能是令人畏惧的，特别是对于第一次更新项目。考虑将模型结构固定，并着重于利用新数据开发一组新的权重。

也许集中精力在哪一个和多少个数据样本来拟合模型。我建议探索模型技能的敏感性分析与最近用于训练的数据范围（例如，模型拟合上个月的数据，在过去3个月的拟合，等等）。

14.3.4 评估候选模型

更新模型实际上是一个用于替换已有模型的新的候选模型。因此，必须严格评估这些模型并与现有模型进行比较。

评估模型

这意味着对用于评估现有模型和解释技能得分的相同数据进行预测，以查看候选模型如何执行。这个技能不需要一直比较，事实上，它可能不应该。我建议根据最近的一个月，比如上个月、3个月或6个月的时间，关注问题和数据的可用性来评估和比较模型技能。

鲁棒性评估 得分技能是相对的。现有的模型是基准线，改进基准线是目标。这意味着结果必须是稳健的。在一个大的测试数据语料库上重复试验，以控制随机数据中的方差。重复多次试验以控制随机算法中的方差。考虑使用统计测试来告诉你两个结果群体之间的差异是否显著以及显著到了什么程度。决定是否用新模型替换现有模型必须是一个强有力的防御决策，而唯一的防御是鲁棒的结果。

公平基准线

候选模型必须远远胜过现有的模型。现有的模型是基准线。但是允许现有模型提供一个公平的比较点是很重要的。如果你已经开发了用新的数据在n个额外周期更新现有权重的模型，那么考虑包括一个候选模型，该模型在旧的数据上更新n个额外的周期的现有模型。这个和类似的比较点将帮助你弄清楚模型技能的任何变化是由于额外的周期还是归因于新的数据。

展示结果

这个决定可能不是你自己的，例如，其他利益的相关者。考虑使用图表，如箱线图，让你在视觉上比较结果的分布，包括平均、中位数和其他百分位的结果。还考虑将结果呈现在模型技能（例如损失或者准确性）方面，但是技能的改变对用户、经验、成本或者其他业务关注的影响。

14.3.5 替换模型

一旦评估了候选模型，就可以由一个新的模型来替换现有的模型。鉴于利益相关者的签署，这应该是一个简单的过程。我将以将模型的权重和模型结构存储在单独的配置文件中。这使您可以在更新模型时单独更新模型权重，这是一个较小的且风险较小的更改。

对新模型和旧模型的并行性进行一段时间的监控是非常重要的。这种对当前和先前模型的持续监测是至关重要的。

- 如果一切都好的话，它可以让你维护更新模块的决定；
- 如果所有情况都不好，它会给你证明以前的模型是否应该被转换回来；
- 它有助于向利益相关者报告系统的健康和持续改进。

14.4 扩展

你想更深入地学习更新一个拟合LSTM模型吗？本章节列出了本课程中一些具有挑战性的扩展：

- 列出5个预测问题

14.5 总结

在本课程中，你学习到了怎么样在更新你的最后LSTM模型中充分利用新数据。特别地，你学到了：

- 对新数据的监控、恢复技能和提升模型学习能力的兴趣；
- 用新的数据更新一个新的LSTM模型的5步过程；
- 当开发具有新数据的更新了的LSTM模型时需要考虑的4个关键方法。

这是最后的课程，做得好！