### 第二十一次2020年5月11日-5月17日

LEARNING FINITE STATE REPRESENTATIONS OF RECURRENT POLICY NETWORKS

递归策略网络的学习有限状态表示

递归神经网络（RNN）是控制策略的有效表示形式，适用于各种强化和迁移学习问题。 但是，RNN策略由于使用连续值存储向量和观察特征而特别难以解释，理解和分析。在本文中，我们介绍了一种新技术“量化瓶颈插入”，以学习这些矢量和特征的有限表示形式。 结果是RNN的量化表示，可以对其进行分析以提高我们对内存使用和一般行为的理解。 我们在合成环境和六款Atari游戏中展示了该方法的结果。在某些情况下，使用有限的3个离散内存状态和10个观测值来获得完美的Pong策略，所得的有限表示形式令人惊讶地小。 我们还表明，这些有限的政策表示形式导致了更好的可解释性。

Abduction-based explanations for machine learning models

机器学习（ML）在众多设置中的应用范围不断扩大，激发了为预测做出小的解释的能力。 人们通常接受小小的解释，因为它们易于人类决策者理解。 关于计算解释的较早工作基于启发式方法，就此类解决方案与基数或最小子集解释的接近程度而言，不提供质量保证。本文开发了一种与约束无关的解决方案，用于计算任何ML模型的解释。所提出的解决方案利用了归纳推理，并提出了使用某些目标约束推理系统可以将ML模型表示为约束集的要求，对于这些目标约束推理系统，可以使用一些预言片来回答决策问题。 在众所周知的数据集上获得的实验结果验证了所提出方法的可扩展性以及所计算解决方案的质量。

总结：

如上所述，所提出的用于使用基本约束形式主义为ML模型计算子集和基数最小解释的方法可以为各种流行的数据集（例如，来自UCI机器学习存储库，宾夕法尼亚州机器学习基准以及MNIST数字。具体而言，实验结果表明，如果将该方法应用于神经网络分类器，则能够获得比现有技术合理的解释少的解释，并且因此，从人类决策者的角度来看，它可能是更可解释的首选。提议的想法的可伸缩性已通过使用SMT和MILP oracles进行了测试。如图所示，MILP总体上优于SMT同类产品。还显示出计算最小尺寸的解释在计算上是相当昂贵的，这使得子集最小的解释是更好的选择。此处的另一个重要因素是，最小子集的解释通常比基数最小的解释大一点。

相关工作：

我们简要概述了关于基于启发式的机器学习模型解释的两行研究。第一行着重于使用可解释的模型从整体上解释复杂的机器学习模型（Frosst and Hinton 2017; Zhang et al.2018）。这样的算法将一对ML模型，原始的复杂ML模型和目标可解释模型作为输入。例如，原始模型可能是经过训练的神经网络，而目标模型可能是决策树（Zhang等人2018）。这里的基本假设之一是，存在从原始模型到目标模型的转换，从而保留了原始模型的准确性。第二行工作重点是针对给定输入的ML模型的解释（Ribeiro，Singh和Guestrin 2016; 2018; Simonyan，Vedaldi和Zisserman 2013）。例如，给定汽车的图像，我们可能想要解释为什么神经网络将该图像分类为汽车。最突出的例子是LIME框架（Ribeiro，Singh和Guestrin 2016）。该方法的主要思想是利用输入的局部扰动来学习输入的重要特征。该算法观察ML模型在扰动的输入上的行为。根据此信息，将给定样本分类的重要特征与其余特征线性分离。 LIME方法与模型无关，因为它可以与任何ML模型一起使用。另一组方法基于显着性图（Simonyan，Vedaldi和Zisserman 2013）。这个想法是，最重要的输入功能可以使用ML模型的知识来获得，例如梯度。但是，这些方法都不能保证解释的质量。

结论：

ML预测的解释在许多应用中至关重要。过去有关ML模型的计算说明的工作主要针对启发式方法，这可能会产生质量较差的解决方案。本文提出了使用归纳推理（即（最短）素数蕴涵的计算）来寻找子集或基数最小的解释。尽管本文考虑了神经网络的MILP编码，但该方法不受约束，并且与所使用的ML模型无关。可以将类似的方法应用于任何其他ML模型，只要该模型可以用某种约束推理系统表示，并且可以使用专用的oracle来确定蕴含查询。实验结果证明了计算出的解释的质量，并强调了在子集最小的解释和基数最小的解释之间的重要折衷，子集最小的解释在计算上更容易找到，基数最小的解释可能很难找到，但是提供了最佳的质量。（度量为指定功能的数量）。可以设想许多工作。其中包括考虑其他ML模型，ML模型的其他编码，其他用于回答蕴含查询的方法，以及对其他基准套件的评估。

Understanding VSIDS Branching Heuristics in Conflict-Driven Clause-Learning SAT Solvers

冲突驱动子句学习（CDCL）SAT求解器的性能关键取决于可变状态独立衰变和（VSIDS）分支启发法。尽管VSIDS是在15年前提出的，并且此后又提出了许多其他的求解SAT的分支启发式方法，但VSIDS仍然是最有效的分支启发式方法之一。尽管它得到了广泛的使用和反复的尝试来理解它，但是这种累加的碰撞和乘法衰变分支启发法仍然是一个谜。在本文中，我们通过回答以下关键问题来加深对VSIDS的理解。我们提出的第一个问题是“ VSIDS选择累加碰撞的变量类别有何特别之处？”在回答这个问题时，我们表明VSIDS压倒性地拾取，颠簸和学习桥梁变量，桥梁变量定义为在SAT实例社区结构中连接不同社区的变量。这是令人惊讶的，因为在发现社区结构和SAT解算器性能之间的联系之前，发明了VSIDS已有十多年了。此外，我们表明，被视为排名函数的VSIDS与时间图中心度度量密切相关。综合这两个发现，我们得出结论，VSIDS选择了高中心桥梁变量。我们提出的第二个问题是“乘法衰减在使VSIDS如此有效方面起什么作用？”我们表明，乘法衰减的行为就像指数移动平均线（EMA），它比在间歇性发生的变量（噪声）更偏向于在冲突中持续发生的变量（信号）。我们提出的第三个问题是“ VSIDS是否在时间和空间上得到关注”。我们表明，VSIDS不成比例地从一些社区中选择了变量，这与随机分支启发法不同。我们将这些发现汇总在一起，发明了一种新的自适应VSIDS分支启发式算法，该算法比SAT Competition 2013基准中最知名的VSIDS变体之一可以解决更多实例。

Heuristics for Fast Exact Model Counting

快速精确模型计数的启发式

良好的变量分支启发式方法对于基于DPLL的SAT求解器的性能至关重要，并且，由于Cachet是基于DPLL的#SAT求解器，很自然地，其性能也取决于良好的变量分支启发式方法。 我们探索了许多可用的不同分支启发法，包括动态文字计数启发法，冲突驱动启发法和基于单元传播的启发法。 我们还开发了一种新的启发式VSADS，它似乎非常适合#SAT。 所有这些启发式方法当前都在Cachet中实现，可以通过命令行参数进行选择。 我们首先回顾一下这些启发式方法。

Score (VSADS) = p × score (VSIDS) + q × score (DLCS)