

自动编程：可微分抽象演算规则解释器

周鹏 武延军 赵琛

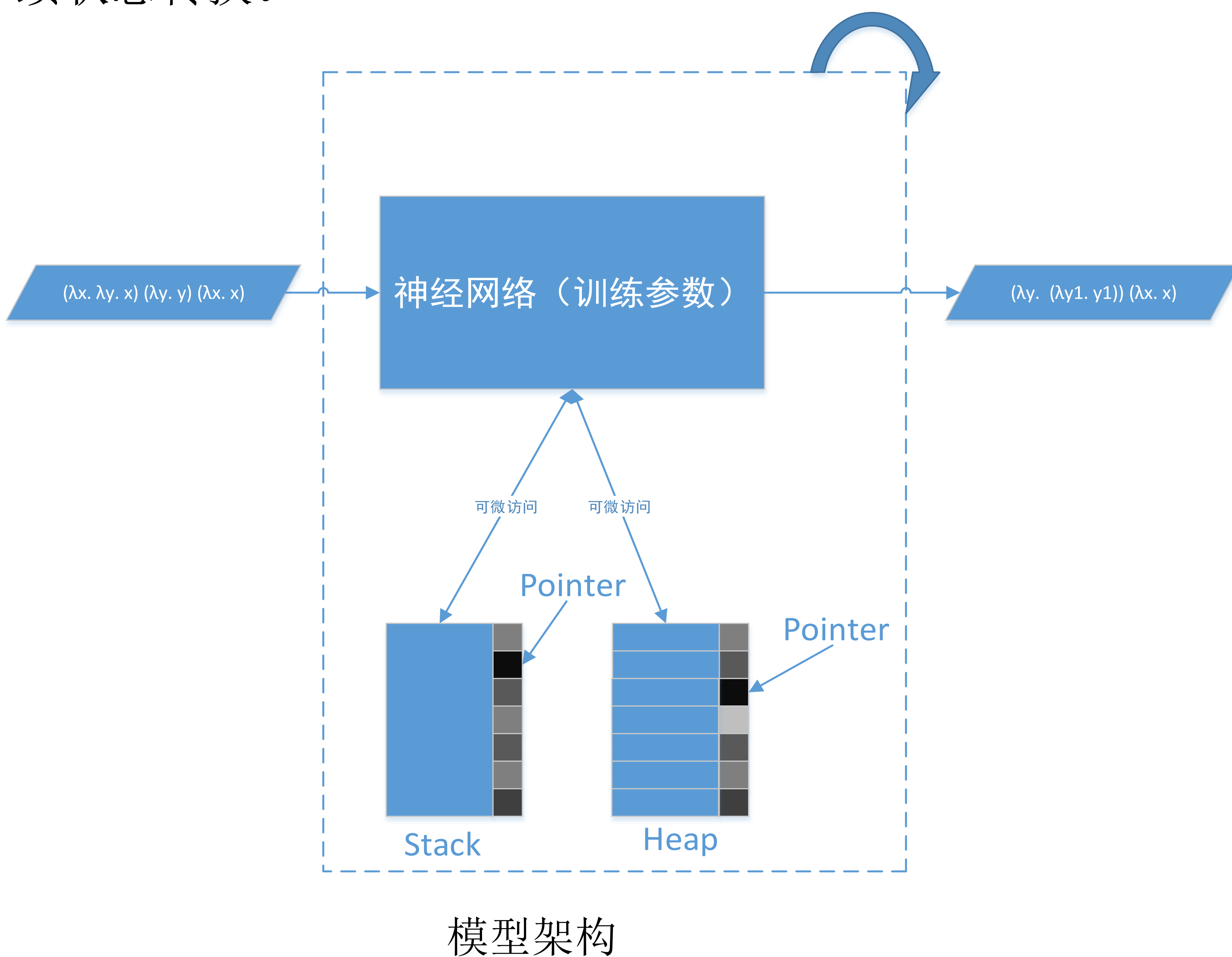
联系人：周鹏 13701282442 zhoupeng@iscas.ac.cn

系统简介：

本系统是软件所在自动编程上的研究成果之一。自动编程是智能软件的核心研究问题，当前自动编程相关研究过多关注程序执行底层细节（如复杂的编程语言规则、内存访问、编程者逻辑等），过多的细节给神经网络引入了非必需的额外学习复杂度，导致学习模型存在跨任务的泛化能力差、程序覆盖面窄、通用性不足等问题，这些问题明显制约了自动编程技术的实际应用能力。因此，我们的研究工作努力避免编程者视角，在更具抽象的层次，从机器或纯演算角度选取基于推演式计算语言，构造基本操作连续可微分的演算结构，实现演算推导规则的可训练学习，以期学习模型具备更好的跨任务泛化能力。

实现思路：

本系统选择一种应用广泛的推演式演算语言，将演算语言的执行环境表示为神经网络和辅助缓存组成的状态机，以attention机制与矩阵运算表达式将缓存（栈、堆）的寻址和读写访问实现为连续函数，每步归纳执行实现为状态机单步状态转换，循环神经网络在训练阶段学习预测函数（训练参数），在推导阶段基于状态上下文预测对缓存的操作动作类型和序列，整个推演执行过程表示为状态机的连续状态转换。



典型技术对比

方法	对比
谷歌 NPI	1) 特定于一个过程化实验用小指令集 2) 需要给出特定执行轨迹，每个算法单独训练，一个训练模型不能解决另一个相似算法
微软 DeepCoder	1) 特定于一个自定义的DSL语言 2) 需要大量输入输出数据，只能预测指令空间的每条指令在每行的呈现概率 3) 一个训练模型只能解决一个算法不能给其他类似任务带来帮助
本系统	1) 基于一个成熟的图灵完备演算语言 2) 基于一些基本演算样例训练模型，实现对演算动作的逐步预测学习 3) 具备跨任务的泛化能力

创新点：

- ✓ 首次初步实现了一种应用广泛的图灵完备演算语言的可微分解释器模型；
- ✓ 实现了通过以基本演算样例训练学习，可以自动解释执行其他未见过的算法演算，实现跨任务的泛化能力；
- ✓ 对于自动编程，首次提出并验证了从机器或纯计算角度以更抽象层次构造神经网络学习模型的可行性以及带来的好处。