# DeepMind提出关系RNN:记忆模块RMC解决关系 推理难题

不久之前,DeepMind 和伦敦大学学院 CoMPLEX 的研究者提出一种关系循环神经网络,该网络利用一种新型记忆模块 RMC 解决标准记忆架构难以执行关系推理任务的问题。该方法在强化学习领域(如 Mini PacMan)、程序评估和语言建模上获得了很大进步,在WikiText-103、Project Gutenberg 和 GigaWord 数据集上获得了当前最优的结果。

论文: Relational recurrent neural networks

# Relational recurrent neural networks

Adam Santoro\* $^{\alpha}$ , Ryan Faulkner\* $^{\alpha}$ , David Raposo\* $^{\alpha}$ , Jack Rae $^{\alpha\beta}$ , Mike Chrzanowski $^{\alpha}$ , Théophane Weber $^{\alpha}$ , Daan Wierstra $^{\alpha}$ , Oriol Vinyals $^{\alpha}$ , Razvan Pascanu $^{\alpha}$ , Timothy Lillicrap $^{\alpha\beta}$ 

\*Equal Contribution

<sup>α</sup>DeepMind London, United Kingdom

<sup>β</sup>CoMPLEX, Computer Science, University College London London, United Kingdom

{adamsantoro; rfaulk; draposo; jwrae; chrzanowskim; theophane; weirstra; vinyals; razp; countzero}@google.com

论文链接: https://arxiv.org/abs/1806.01822

摘要:基于记忆的神经网络通过长期记忆信息来建模时序数据。但是,目前尚不清楚它们是否具备对记忆信息执行复杂关系推理的能力。在本论文中,我们首先确认了标准记忆架构在执行需要深入理解实体连接方式的任务(即涉及关系推理的任务)时可能会比较困难。然后我们利用新的记忆模块 Relational Memory Core (RMC) 改进这些缺陷,RMC 使用 Multi-head点积注意力令记忆相互影响。最后,我们在一系列任务上对 RMC 进行测试,这些任务可从跨序列信息的更强大关系推理中受益,测试结果表明在强化学习领域(如 Mini PacMan)、程序评估和语言建模上获得了很大进步,在 WikiText-103、 Project Gutenberg 和 GigaWord数据集上获得了当前最优的结果。

## 1引言

人类使用复杂的记忆系统来获取和推理重要信息,而无需过问信息最初被感知的时间 [1, 2]。 在神经网络研究中,建模序列数据的成功方法也使用记忆系统,如 LSTM [3] 和记忆增强神经 网络 [4-7]。凭借增强记忆容量、随时间有界的计算开销和处理梯度消失的能力,这些网络学 会关联不同时间的事件,从而精通于存储和检索信息。

这里我们提出:考虑记忆交互与信息存储和检索会有很大收获。尽管当前模型可以学会分割和关联分布式、向量化记忆,但它们并不擅长显性地完成这些过程。我们假设擅长这么做的模型可能会更好地理解记忆的关联,从而获得对时序数据进行关系推理的更强能力。我们首先通过一个强调序列信息的关系推理的演示任务展示了当前模型确实在这方面比较困难。而使用Multi-head 点积注意力的新型 RMC 可使记忆交互,我们解决并分析了这个演示任务。之后我们应用 RMC 处理一系列任务(这些任务可能从更显著的记忆交互中受益),从而得到了潜在增长的记忆容量,可处理随时间的关系推理:在 Wikitext-103、Project Gutenberg、GigaWord 数据集上的部分可观测强化学习任务、程序评估和语言建模任务。

### 3 模型

我们的主导设计原则是提供架构主干网络,使模型可学习分割信息,并计算分割后信息之间的交互。为此我们结合了 LSTM 构造块、记忆增强神经网络和非局部网络(具体来说是 Transformer seq2seq 模型 [19]) 以实现主体网络。与记忆增强架构类似,我们考虑使用固定的记忆单元集合,但是我们利用注意力机制进行记忆单元之间的交互。如前所述,我们的方法与之前的研究不同,我们在单个时间步上对记忆应用注意力机制,而且不跨越从所有之前的观测中计算出的所有之前表征。

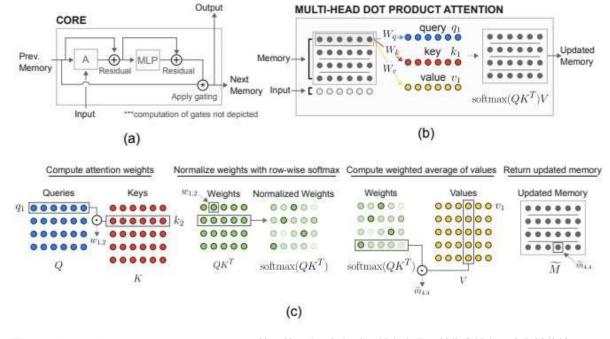


图 1: Relational Memory Core。 (a) RMC 接受前一个记忆矩阵和输入向量,并作为输入,它们被传输至MHDPA 模块(A)。 (b) 利用 Query 逐行共享的权重 W^q、Key 逐行共享的权重 W^k 和 Value 逐行共享的权重 W^v,计算每个记忆单元的线性投影。 (c) 将 Query、key 和 Value 编译成矩阵,计算 softmax(QK^T) V。该计算的输出是一个新的记忆,其中的信息根据记忆的注意力权重进行混合。MLP 被逐行应用于 MHDPA 模块的输出(a),得到的记忆矩阵是门控矩阵,作为核心输出或下一个记忆状态。

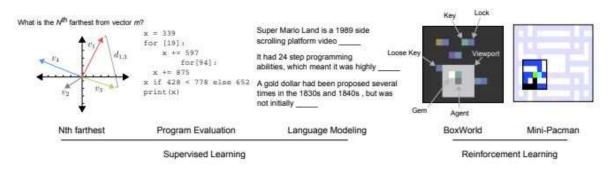
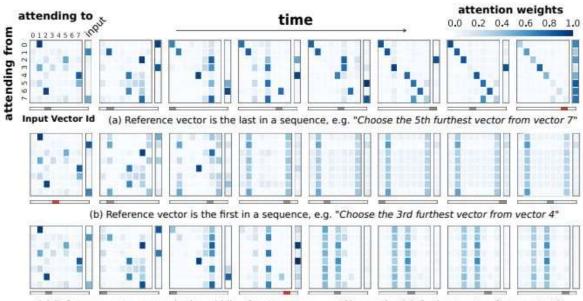


图 2: 任务。我们在一系列监督和强化学习任务上对 RMC 进行测试。Nth Farthest 演示任务和语言建模任务值得注意。前者中解决方案需要显性的关系推理,因为该模型必须把向量之间的距离关系进行分类,而不是对向量本身进行分类。后者基于大量自然数据测试模型,使得我们可以进行与精心调整的模型之间的性能对比。

#### 5 结果



(c) Reference vector comes in the middle of a sequence, e.g. "Choose the 6th furthest vector from vector 6"

图 3:模型分析。每行描述了特定序列在每个时间步上的注意力矩阵。下方的文本即该序列的特定任务,序列被编码,并作为模型输入。我们把任务中引用的向量标红:即如果模型选择离向量 7 第 2 远的向量,则标红的是向量7 中被输入到模型的时间点。单个注意力矩阵展示了从一个特定记忆单元(y 轴)到另一个记忆单元(列)的注意力权重,或者输入(offset 列),数字表示记忆单元,「input」表示输入词嵌入。

Table 1: Test per character Accuracy on Program Evaluation and Memorization tasks.

Model	Add	Control	Program	Copy	Reverse	Double
LSTM [3, 34]	99.8	97.4	66.1	99.8	99.7	99.7
EntNet [35]	98.4	98.0	73.4	91.8	100.0	62.3
DNC [5]	99.4	83.8	69.5	100.0	100.0	100.0
Relational Memory Core	99.9	99.6	79.0	100.0	100.0	99.8

表 1: 在程序评估和记忆任务上的每字符测试准确率。

Table 2: Validation and test perplexities on WikiText-103, Project Gutenberg, and GigaWord v5.

	WikiText-103		Gutenberg		GigaWord	
	Valid.	Test	Valid	Test	Test	
LSTM [37]	(2)	48.7		(4)	34	
Temporal CNN [38]	27	45.2	12		12	
Gated CNN [39]		37.2			17	
LSTM [29]	34.1	34.3	41.8	45.5	43.7	
Quasi-RNN [40]	32	33	2		-	
Relational Memory Core	30.8	31.6	39.2	42.0	38.3	

表 2: 在 WikiText-103、Project Gutenberg 和 GigaWord v5 数据集上的验证困惑度和测试困惑度。