



(12) 发明专利申请

(10) 申请公布号 CN 119323237 A

(43) 申请公布日 2025. 01. 17

(21) 申请号 202411337511.2

(22) 申请日 2019.02.11

(30) 优先权数据

62/628,899 2018.02.09 US

(62) 分案原申请数据

201980009347.3 2019.02.11

(71) 申请人 渊慧科技有限公司

地址 英国伦敦

(72) 发明人 T·M·S·P·拉马尔霍

D·罗森鲍姆 M·加内洛

C·马迪森 S·M·埃斯拉米

Y·W·泰 D·J·雷赞德

(74) 专利代理机构 北京市柳沈律师事务所

11105

专利代理师 金玉洁

(51) Int.Cl.

G06N 3/0455 (2023.01)

G06N 3/084 (2023.01)

G06N 3/09 (2023.01)

G06N 3/092 (2023.01)

G06N 7/01 (2023.01)

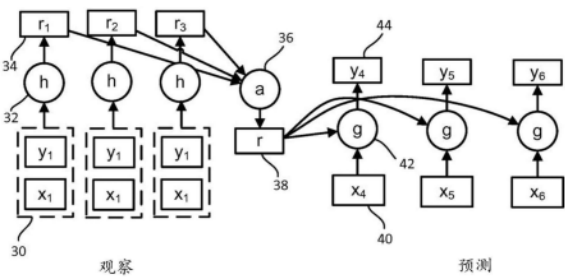
权利要求书2页 说明书16页 附图3页

(54) 发明名称

实现用于高效学习的条件神经过程的神经网络系统

(57) 摘要

根据第一方面,一种用于根据一个或多个学习的数据分布从输入数据值生成输出数据值的网络系统包括:输入,该输入接收观察的集合,每个观察包括第一变量的相应第一数据值和依赖于第一变量的第二变量的相应的第二数据值。该系统可以包括编码器神经网络系统,该编码器神经网络系统被配置为编码观察的集合中的每个观察,以提供对于每个观察的编码的输出。该系统可以进一步包括聚合器,该聚合器被配置为聚合对于观察的集合的编码的输出,并提供聚合的输出。该系统还可以包括解码器神经网络系统,该解码器神经网络系统被配置为接收聚合的输出和目标输入值的组合,并提供解码器输出。该目标输入值可以包括第一变量的值,并且解码器输出可以预测第二变量的对应值。



1. 一种神经网络系统,用于根据一个或多个学习的数据分布从输入数据值生成解码器输出数据值,所述系统包括:

输入,该输入接收观察的集合,每个观察包括第一变量的相应的第一数据值和依赖于所述第一变量的第二变量的相应的第二数据值;

编码器神经网络系统,被配置为编码所述观察的集合中的每个观察,以提供对于每个观察的编码的输出;

聚合器,被配置为聚合对于所述观察的集合的编码的输出,并提供聚合的输出;和

解码器神经网络系统,被配置为接收所述聚合的输出和目标输入值的组合,并提供解码器输出,其中,所述目标输入值包括所述第一变量的值,并且所述解码器输出预测所述第二变量的对应值。

2. 根据权利要求1所述的系统,其中,所述系统被配置为对于多个训练步骤中的每一个:

从观察中选择观察的子集以输入到所述编码器神经网络系统,并选择用于训练所述系统的目标的集合;

使用所述编码器神经网络系统编码所述观察的子集中的每个观察,以提供对于每个观察的编码的输出;

聚合所述编码的输出;

将所述编码的输出的聚合与来自所述目标的集合中的每个第一数据值进行组合,以提供组合的集合以进行解码;

使用所述解码器神经网络系统解码所述组合的集合中的每个组合,以提供解码器输出的集合;和

使用依赖于所述解码器输出的集合和所述目标的集合中的第二数据值的损失函数,通过反向传播来训练所述编码器神经网络系统和所述解码器神经网络系统。

3. 根据权利要求1或2所述的系统,其中,所述解码器输出限定解码器输出分布的参数,并且其中,所述系统被配置为根据所述解码器输出分布确定所述第二变量的所述对应值。

4. 根据权利要求3所述的系统,其中,所述解码器输出分布限定所述第二变量的所述对应值的均值和方差。

5. 根据权利要求3所述的系统,其中,所述第一数据值中的每一个包括限定数据项的向量,并且所述第二数据值中的每一个限定相应的数据项的对应分类,并且其中,所述解码器输出分布限定在分类上的概率分布。

6. 根据权利要求5所述的系统,其中,所述编码器神经网络系统被配置为通过编码每个观察的第一数据值来编码每个观察,并且其中,所述聚合器被配置为:

对于每个类,基于对于该类中的观察的编码的第一数据值来确定该类的聚合;和

通过组合每个类的聚合来确定聚合的输出。

7. 根据权利要求1至6中任一项所述的系统,其中,在所述观察的集合中的观察的置换下,所述聚合的输出是不变的。

8. 根据权利要求1至7中任一项所述的系统,其中,由所述解码器神经网络系统接收的组合还包括一组隐变量,所述隐变量一起表征所述目标输入值。

9. 根据权利要求8所述的系统,其中,所述编码器神经网络系统还被配置为确定所述一

组隐变量的分布的参数,并且其中,所述解码器神经网络系统被配置为从所述一组隐变量的分布进行采样以提供所述解码器输出。

10.根据从属于权利要求2时的权利要求9所述的系统,还包括先验神经网络和后验神经网络,所述先验神经网络根据所述观察的集合确定所述一组隐变量的先验分布的参数,所述后验神经网络根据所述目标的集合确定所述一组隐变量的后验分布的参数。

实现用于高效学习的条件神经过程的神经网络系统

[0001] 本申请是申请日为2019年02月11日、申请号为201980009347.3、发明名称为“实现用于高效学习的条件神经过程的神经网络系统”的发明专利申请的分案申请。

技术领域

[0002] 本说明书涉及生成输出数据值的神经网络系统。

背景技术

[0003] 神经网络是利用非线性单元的一个或多个层来预测接收的输入的输出的机器学习模型。一些神经网络除了输出层之外还包括一个或多个隐藏层。每个隐藏层的输出被用作到网络中的下一层(即,下一隐藏层或输出层)的输入。网络的每个层根据相应一组参数的当前值从接收的输入生成输出。

[0004] 一些神经网络是循环神经网络(recurrent neural network)。循环神经网络是接收输入序列并根据输入序列生成输出序列的神经网络。特别地,循环神经网络可以将来自先前时间步的网络的一些或所有内部状态用于计算当前时间步处的输出。循环神经网络的示例是包括一个或多个长短期记忆(LSTM)块的LSTM神经网络。每个LSTM记忆块可以包括一个或多个细胞(cell),每个细胞包括输入门、遗忘门和输出门,该输入门、遗忘门和输出门允许细胞存储细胞的先前状态,例如,用于生成当前激活(activation),或被提供到LSTM神经网络的其他组件。

发明内容

[0005] 本说明书描述被实现为在一个或多个位置的一个或多个计算机上的计算机程序的系统,该系统生成输出数据值,例如,预测的观察。

[0006] 根据第一方面,一种用于根据一个或多个学习的数据分布从输入数据值生成输出数据值的网络系统包括:输入,该输入接收观察的集合,每个观察包括第一变量的相应第一数据值和依赖于第一变量的第二变量的相应的第二数据值。该系统可以包括编码器神经网络系统,该编码器神经网络系统被配置为编码观察的集合中的每个观察,以提供对于每个观察的编码的输出。该系统可以进一步包括聚合器(aggregator),该聚合器被配置为聚合对于观察的集合的编码的输出,并提供聚合的输出。该系统还可以包括解码器神经网络系统,该解码器神经网络系统被配置为接收聚合的输出和目标输入值的组合,并提供解码器输出。该目标输入值可以包括第一变量的值,并且解码器输出可以预测第二变量的对应值。

[0007] 神经网络系统的某些实现方式可以从观察的集合学习一般信息,对这些知识进行聚合,然后使用聚合的知识来促进进一步的学习。该系统可以例如通过嵌入观察并聚合嵌入,然后使数据生成器(即解码器神经网络系统)以聚合的嵌入为条件,来学习以输入观察的集合为条件的一个或多个分布。

[0008] 第一和第二数据值和目标输入值可以包括单个数据值或向量,例如,其限定图像

或音频段。对于这些数据/目标值,可能存在系统的单个物理输入。例如,通过对聚合器使用均值函数,在观察的集合中的观察的置换下,聚合的输出可以是不变的。置换(permutation)可以被认为是值的顺序的改变。因此,该聚合器可以是对称聚合器。聚合的参数可以具有预定义的大小。该大小可以与参数的位和/或维度的数量有关。

[0009] 该系统可以被配置为对于多个训练步骤中的每一个执行一组操作。所述操作可以包括从观察中选择观察的子集以输入到编码器神经网络系统,并选择用于训练所述系统的目标的集合。所述操作可以进一步包括使用编码器神经网络系统编码观察的子集中的每个观察,以提供对于每个观察的编码的输出。所述操作可以进一步包括聚合编码的输出。所述操作可以进一步包括将编码的输出的聚合与来自目标的集合中的每个第一数据值进行组合(例如串联),以提供组合的集合以进行解码。所述操作还可包括使用解码器神经网络系统解码组合的集合中的每个组合,以提供解码器输出的集合。所述操作可以进一步包括使用依赖于解码器输出的集合和目标的集合的第二数据值的损失函数,通过反向传播来训练编码器神经网络系统和解码器神经网络系统。

[0010] 编码器和解码器可以各自包括前馈神经网络,例如多层感知器(MLP),和/或编码器可以包括卷积神经网络,例如当输入数据包括图像时,和/或解码器可以包括反卷积神经网络。

[0011] 解码器输出可以限定解码器输出分布的参数。该系统然后可以被配置为根据解码器输出分布确定第二变量的对应值。例如,该系统可以从分布中采样或只是输出均值。例如,解码器输出分布可以限定第二变量的对应值的均值和方差。均值和方差可以是高斯分布的,高斯分布可以用于确定第二变量的对应值,以及可选地确定第二变量的不确定性。因此,该系统可以被配置为执行回归任务。

[0012] 回归任务可以是图像或声音补全任务。例如,第一数据值可以包括空间和/或时间索引(例如,指定样本(例如像素或音频值)在空间和/或时间中的位置)。然后,第二数据值可以包括与空间和/或时间索引对应的图像或声音数据值。然后,目标输入值可以限定将生成的图像或声音数据值的空间和/或时间索引。

[0013] 在一些其他实现方式中,该系统可以被配置为执行分类任务。例如,第一数据值可以包括图像或声音数据(例如限定包括数字化图像或声音数据的数据项的向量),并且第二数据值可以包括图像或声音数据的分类。解码器输出分布可以限定分类上的概率分布(例如,经由训练)。

[0014] 对于分类任务,对于每个类,编码的观察的聚合可以包括该类中的观察的编码的第一数据值的聚合(例如,经由置换不变聚合,例如均值计算)。然后,可以组合(例如,经由串联)每个类的聚合的值以形成输入到解码器神经网络系统中的聚合的输出(组合的聚合的输出)。

[0015] 在一些其他实现方式中,系统可以被配置为执行文本分类任务。第一数据值可以包括文本数据,并且第二数据值包括文本或声音数据的分类。

[0016] 在一些其他实现方式中,系统可以被配置为控制与环境交互的智能体。这可以由强化学习(reinforcement learning)来实现。广义上,强化学习系统是选择要由与环境交互的强化学习智能体执行的动作的系统。为了使智能体与环境交互,该系统接收表征环境的当前状态的数据,并响应于接收的数据选择要由智能体执行的动作。在本说明书中,将

表征环境的状态的数据称为观察 (observation)。可选地,在某时间步的观察可以包括来自前一个时间步的数据,例如,在前一个时间步执行的动作,在前一个时间步接收的奖励 (reward) 等等。

[0017] 在一些实现方式中,环境是现实世界环境,并且智能体是与现实世界环境交互的机电智能体。例如,智能体可以是与环境交互以完成特定任务 (例如,在环境中定位感兴趣的物体或将感兴趣的物体移动到环境中的指定位置或导航到环境中的指定目的地) 的机器人或其他静态或移动机器;或者智能体可以是导航通过环境的自主或半自主的陆地或空中或海上载具。

[0018] 在这些实现方式中,观察可以包括例如图像、物体位置数据和传感器数据中的一个或多个,以在智能体与环境交互时捕获观察,例如来自图像、距离或位置传感器的传感器数据或来自致动器的传感器数据。

[0019] 在这些实现方式中,动作可以是控制机器人的控制输入,例如,机器人的关节的扭矩或更高级的控制命令;或控制自主或半自主的陆地或空中或海上载具的控制输入,例如,到载具的控制面或其他控制元件的扭矩或更高级的控制命令;或例如电机控制数据。换句话说,动作可以包括例如机器人的一个或多个关节或另一机械智能体的部件的位置、速度或力/扭矩/加速度数据。

[0020] 在一些实现方式中,环境是模拟环境,并且智能体被实现为与模拟环境交互的一个或多个计算机。

[0021] 例如,模拟环境可以是机器人或载具的模拟,并且可以在该模拟上训练强化学习系统。例如,模拟环境可以是运动模拟环境,例如驾驶模拟或飞行模拟,并且智能体是导航通过运动模拟的模拟载具。在这些实现方式中,动作可以是控制模拟用户或模拟载具的控制输入。模拟环境对于在现实环境中使用增强学习系统之前训练该系统非常有用。在另一示例中,模拟环境可以是视频游戏,并且智能体可以是玩视频游戏的模拟用户。通常,在模拟环境的情况下,观察可以包括一个或多个先前描述的观察或观察类型的模拟版本,并且动作可以包括一个或多个先前描述的动作或动作类型的模拟版本。

[0022] 因此,在一些实现方式中,第一数据值可以包括用于例如机器人或半自主或自主载具的机械设备的控制信号和/或传感器数据。第二数据值可以包括表征机械设备响应于控制信号和/或传感器数据的预测的行为的数据值。可以响应于预测的行为来控制机械设备。

[0023] 在一些实现方式中,聚合的输出和目标输入值的组合包括聚合的输出和目标输入值的串联。在一些实现方式中,聚合编码的输出包括计算编码的输出的均值。

[0024] 在一些实现方式中,由解码器神经网络系统接收的组合还包括一组隐变量,其共同表征目标输入值,例如,限定目标输入值所属的类。以这种方式,该系统可以用于限定预测的第二变量应当属于的目标类型,例如目标图像的类型或示例。以这种方式,可以通过固定该组隐变量并生成第二变量的值来生成目标类型的示例,例如,以生成目标类型的图像的一组像素或声音类型的声音的一组音频样本,或更一般而言,来自指定学习的分布的样本。该系统因此也可以用于从一种数据模式转换为另一种数据模式。例如,声音或图像类型可以由文本限定,因此该系统可以用于生成与文本对应的图像或声音,或者在自动转换的情况下,可以从另一声音样本中生成隐变量。

[0025] 该系统可以被配置为确定该组隐变量的分布的参数。然后,可以将解码器神经网络系统配置为从该组隐变量的分布中采样以提供解码器输出。采样的隐变量可以与聚合的输出组合(例如,串联到聚合的输出)并输入到解码器神经网络中。

[0026] 例如,该系统可以包括先验神经网络和后验神经网络,先验神经网络根据观察的集合确定一组隐变量的先验分布的参数,并且后验神经网络根据目标的集合确定一组隐变量的后验分布的参数。在训练期间,先验分布和后验分布可以变得越来越相似,使得基于目标和隐变量预测的值更加接近训练期间使用的观察的值。

[0027] 通常,预测的数据值可以与预测的观察对应,并且可以在现实或模拟环境中的控制任务中使用,以预测环境中的计划的动作的结果。这样的预测可以在强化学习系统中使用,例如以预测动作的结果。

[0028] 在一些实现方式中,该系统可以被用于查询数据库。例如,可以结合对数据库中的数据编码的聚合的输出或参数来向系统提供查询。然后,该系统可以基于目标输入值且基于存储在数据库中的值的先前编码的观察,提供查询数据库的预测或结果。

[0029] 可以从得自一个或多个传感器的观察来学习预测;预测的数据项可以包括好像在将来的时间从(一个或多个)传感器生成的数据。预测的数据项可以用于填充丢失的数据,或用于仅对于少量数据项可获得标记的半监督学习,或用于提供多模态数据输出,或用于从一个领域映射到另一领域。输入和/或预测的数据值可以包括音频数据的值,例如数字化的信号值,或静止或运动图像的值,例如像素颜色或亮度数据;限定物体的状态或运动的值;医学数据,例如生物标志物或患者健康数据;音乐数据;表征用户偏好或兴趣的推荐数据;与物理和/或化学过程有关的时空数据;等等。

[0030] 在一个实施例中,经由例如均值运算的交换运算来进行聚合。这提供了置换不变性。该系统可以被配置为在接收(一个或多个)新的观察时迭代地更新聚合输出。这可以通过对新的观察进行编码以提供新的观察的编码的输出(编码的表示),然后应用聚合函数将先前的聚合输出与新的观察的编码的输出进行聚合以产生更新的聚合输出来实现。该更新的聚合输出可以被输入解码器神经网络,并且被用于生成对于其他(一个或多个)目标输入值的预测。

[0031] 根据另一方面,提供了一种计算机实现的方法,用于根据一个或多个学习的数据分布从输入数据值生成解码器输出数据值。该方法包括:接收观察的集合,每个观察包括第一变量的相应第一数据值和依赖于所述第一变量的第二变量的相应的第二数据值;将观察的集合中的每个观察输入编码器神经网络系统,以编码每个观察,从而提供对于每个观察的编码的输出;将每个编码的输出输入聚合器,以聚合编码的输出并提供聚合的输出;并将聚合的输出和目标输入值的组合输入解码器神经网络系统以提供解码器输出,其中,目标输入值包括第一变量的值,并且解码器输出预测第二变量的对应值。

[0032] 根据另一方面,提供了一种用于根据一个或多个学习的数据分布从输入数据值生成输出数据值的方法。该方法可以包括接收与领域有关的目标输入值,该目标输入值用于第一变量。该方法可以进一步包括将目标输入值与聚合的参数组合,所述聚合的参数是来自与该领域相关的训练观察的集合的经由编码器神经网络编码的多个参数的聚合,每个训练观察包括第一变量的相应第一数据值和依赖于第一变量的第二变量的相应的第二数据值。该方法可以进一步包括将目标输入值和聚合的参数的组合输入解码器神经网络系

统,以提供解码器输出,该解码器输出预测与目标输入值对应的第二变量的值。

[0033] 通过利用聚合的参数,可以考虑该聚合的参数来对有关训练观察(涉及与目标输入值相同的领域)的信息进行编码,神经网络可以更有效地确定第二变量的预测。训练观察和目标输入值涉及相同的领域(例如,涉及相同的数据类型),因此它们共享相似的属性。

[0034] 在一个实现方式中,该方法还包括:对于多个训练步骤中的每个步骤:接收观察的训练集合,每个观察包括第一变量的相应的第一数据值和第二变量的相应的第一数据值;将聚合的参数与来自观察的训练集合的每个第一数据值组合,以提供组合的集合以进行解码;使用解码器神经网络系统解码组合的集合中的每个组合,以提供解码器输出的结合;以及使用依赖于解码器输出的集合和目标的集合的第二数据值的损失函数,通过反向传播来训练解码器神经网络系统。

[0035] 通过利用聚合的参数,编码器神经网络可以更快、更高效地被训练。

[0036] 本文描述的实现方式可以实现在一种系统中,该系统包括一个或多个计算机以及一个或多个存储设备,该存储设备存储指令,该指令在由一个或多个计算机执行时,使该一个或多个计算机执行本文描述的方法的操作。

[0037] 此外,实现方式可以实现在一个或多个计算机可读存储介质中,该存储介质存储指令,该指令在由一个或多个计算机执行时,使该一个或多个计算机实现本文描述的系统或方法。

[0038] 本文中对值的集合的任何引用均应解释为表示一个或多个值的集合。因此,观察的集合可以是一个或多个观察。同样,目标的集合可以是一个或多个目标。

[0039] 可以在特定实施例中实现本说明书中描述的主题,以实现以下优点中的一个或多个。

[0040] 与常规系统相比,神经网络系统的某些实现方式可以更快地学习并且使用较少的数据,因为它们能够依赖于从训练数据学习的分布,并能够利用此知识更快地学习相似的分布。因此,例如,可以在几个不同的机器人臂的观察上训练系统,并且总体上学习机器人臂的行为。然后,该系统可能更快地学习特定的新机器人臂的行为。因此,该系统可能学习更快、更准确地做出预测,消耗较少的存储器并使用较少的计算资源。

[0041] 由于可以在任务之间共享学习,因此本文描述的方法和系统为在各种相关任务和数据集上的应用提供了灵活性,并减少了训练。此外,本文描述的机制通过在训练期间学习先验知识并以用于条件化解码器神经网络系统的聚合的参数形式对其进行编码,避免了如同在随机过程中一般对预先指定的先验的需要。

[0042] 神经网络系统的某些实现方式可扩展到大数据集和复杂分布。通过利用编码的输出聚合,本文描述的神经网络系统比例如高斯过程的替代方法更高效,并且更有效地扩展。例如,尽管高斯过程能够适应不同的数据集,但它们的扩展性较差,随着数据集或维度的增加而很快变得难以计算。相反,本文描述的方法和系统可以更高效地扩展。对于 n 个观察的集合和 m 个目标的集合,实施例的运行时间复杂度为 $\mathcal{O}(n + m)$ 量级,因为 n 个观察被聚合为单个表示。这比运行时间复杂度可为 $\mathcal{O}(nm)$ 量级的替代方法更为高效。因此,可以用比常规技术相对少的资源从大数据集学习分布。

[0043] 本说明书中描述的主题的一个或多个实施例的细节在附图和下面的说明书中阐

述。根据以下描述、附图和权利要求书,所述主题的其他特征、方面和优点将变得明了。

附图说明

- [0044] 图1A示出了用于监督学习问题的一般框架的示意图;
[0045] 图1B示出了一般监督学习训练方案的示例的示意图;
[0046] 图1C示出了根据实施例的用于预测输出的条件神经过程 (conditional neural process, CNP) 的示意图;
[0047] 图2示出了根据实施例的使用条件神经过程 (CNP) 来预测解码器输出的示例性过程;和
[0048] 图3示出了根据实施例的用于训练条件神经过程 (CNP) 系统的示例性过程。

具体实施方式

[0049] 神经网络擅长于函数近似 (function approximation), 但通常要从头开始针对每个新函数训练它们。另一方面, 贝叶斯方法, 例如高斯过程 (Gaussian Processes, GP), 利用先验知识来在测试时快速推断新函数的形状。然而, GP的计算量很大, 并且难以设计适当的先验。

[0050] 本申请提出了神经模型的新的族, 即条件神经过程 (CNP), 其结合了神经网络和贝叶斯方法的优点。CNP受例如GP的随机过程的灵活性的启发, 但被构造为神经网络并经由梯度下降进行训练。CNP仅在观察少量训练数据点之后即可做出准确的预测, 但可以扩展到复杂函数和大数据集。

[0051] 鉴于上述情况, 本文描述的实施例提供了能够在较少的数据集上被准确地训练的改进的预测神经网络。这确保了准确性, 同时通过避免处理大量标记数据集的需要而提高了系统的计算效率。CNP是可扩展的, 对于用 n 个观察进行 m 个预测, 实现的运行时间复杂度为 $\mathcal{O}(n + m)$ 。这比运行时间复杂度为 $\mathcal{O}(nm)$ 的等效系统显著提高了效率。

[0052] CNP的性能和通用性确保了可以将它们应用于一系列典型的机器学习任务, 包括回归 (regression)、分类和图像补全。

[0053] 通常可以将监督学习问题界定为给定观察的有限集合的函数近似。分类、回归、动力学建模和图像生成都可以在这一框架中进行。

[0054] 图1A示出了用于监督学习问题的一般框架的示意图。观察的集合包括输入10和输出14。输入经由对应的隐藏函数12 (隐藏函数和参数由虚线指示) 映射到输出。目标16的集合包括输入, 但具有未知输出18。

[0055] 观察是 n 个输入 $x_i \in X$ 和输出 $y_i \in Y$ 的数据集 $\{(x_i, y_i)\}_{i=0}^{n-1}$ 。假定这些表示某个未知函数 $f: X \rightarrow Y$ 的评估 (evaluation) $y_i = f(x_i)$, 该未知函数可以是固定的, 或者可以是某个随机函数的实现。

[0056] 图1B示出了一般监督学习训练方案的示例的示意图。包括输入和输出的已知观察的集合20用于确定隐藏函数 f 的近似22 g 。然后, 该近似函数22可以被用于通过提供从输入到输出的映射来预测对于输入24的集合的输出26。

[0057] 更正式地说, 监督学习算法返回近似函数 $g: X \rightarrow Y$ 或这样的函数上的分布。目的是

在整个空间 X 上使 f 与 g 之间的损失最小化,但是在实践中,该例程是在被留存(held-out)的观察的有限集合(使得它们实际上未标记)上进行评估的。这些未标记的数据点被称为目标。通过最小化 f 和 g 之间的损失,可以将近似函数 g 拟合到观察到的数据,以便凭经验确定一函数,该函数准确预测对于给定输入的输出。

[0058] 监督问题的一种处理方法是对于每个新任务重新随机地初始化参数函数 g ,并将大量计算花费在昂贵的拟合阶段上。经由 g 的架构、损失函数或训练细节来指定实践者可能具有的有关 f 的先验信息。这种方法涵盖了深度监督学习的大部分。由于可以以这种方式表达的先验知识的程度相对有限,并且无法在不同任务之间共享学习,因此所需的训练量很大,并且当训练数据不富足时,深度学习方法往往会失败。

[0059] 另一种方法是采取概率立场并指定函数上的分布,这称为随机过程;高斯过程(GP)就是一个示例。从这一观点来看,实践者关于 f 的先验知识被捕获在关于先验过程的分布假设中,并且学习对应于以观察到的值为条件的函数空间上的贝叶斯推断。在GP示例中,经由参数核函数先验地捕获 f 的平滑度的假设,而 g 被视为根据预测后验分布而分布的随机函数。不幸的是,随着数据集或维数的增加,这样的贝叶斯方法很快在计算上变得棘手。

[0060] 本申请提出了表示监督问题的解决方案的新的模型族;以及学习它们的端到端的训练方法,其将神经网络与和高斯过程相称的特征结合。该族模型称为条件神经过程(CNP),因为它们限定了给定观察集合的情况下函数的条件分布。CNP对观察的依赖性由神经网络参数化,该神经网络在其输入的置换下是不变的。本文描述的实施例包括放大为 $O(n + m)$ 的架构,其中, n 、 m 分别是观察和目标的数量。

[0061] 在其最基本的形式中,CNP嵌入每个观察,并用对称聚合器将这些嵌入聚合到固定维度的进一步嵌入中,并使函数 g 以聚合嵌入为条件。

[0062] 图1C示出了根据实施例的用于预测输出的条件神经过程(CNP)的示意图。 n 个观察30的集合 $\{(x_i, y_i)\}_{i=0}^{n-1}$ 被输入到神经网络编码器 h_0 32中,该编码器 h_0 根据参数 θ 将每个观察映射到对应的编码的观察 r_i 34。

[0063] 然后,将编码的观察34的集合输入聚合器36中,该聚合器36基于聚合操作 a 聚合所有编码的观察以产生聚合 r 38。

[0064] 然后,在预测针对目标输入40的输出44时,利用聚合38来条件化神经网络解码器 g_0 42。也就是说,神经网络解码器42将聚合 r 和目标输入 x_i 作为输入,并基于参数 θ 将其映射到输出 y_i 。

[0065] 可以通过采样随机数据集并遵循梯度步骤来训练CNP,以在给定随机观察集合的情况下最大化目标的随机子集的条件似然。这促使CNP在各种设置(即 $n < m$ 或 $n > m$)上都表现良好。

[0066] 为了界定该问题,考虑输入 $x_i \in X$ 和输出 $y_i \in Y$ 的配对的集合 $O =$

[0067] $\{(x_i, y_i)\}_{i=0}^{n-1} \subset X \times Y$ 和未标记点的另一集合 $T = \{(x_i)\}_{i=n}^{n+m-1} \subset X$ 。这些分别称为观察集合和目标集合。假设输出是以下过程的实现;设 P 为函数 $f: X \rightarrow Y$ 上的概率分布,正式称为随机过程,则对于 $f \sim P$,设置 $y_i = f(x_i)$ 。 P 限定了随机变量 $\{f(x_i)\}_{i=0}^{n+m-1}$ 上的联合分布,因此限定了条件分布 $P(f(T) | O, T)$;系统的任务是给定 O ,预测对于每个 $x \in T$ 的输出值 $f(x)$ 。也就是说,给定有标记观察的集合,系统预测对于每个目标输入值的输出值。

[0068] 作为启发性示例,考虑在实线(即 $X := \mathbb{R}, Y := \mathbb{R}$)上限定的随机一维函数 $f \sim P$ 。0将在实线上构成在不同位置 x_i 处的 f 的值 y_i 的 n 个观察。给定这些观察,目的是在实线上预测新位置处的 f 的值。对 P 进行的经典假设是 f 的函数评估的所有有限集合都是联合高斯分布的。这类随机函数称为高斯过程(GP)。在这种情况下,预测分布 $P(f(T) | O, T)$ 具有简单的解析形式,该解析形式由配对相关结构(经由核函数指定)上的先验假设限定。随机过程的框架很有吸引力,因为贝叶斯规则允许人们在一组概率假设下通过观察 O 而进行对 f 上的预测分布的一致推理。这允许模型是数据高效的,这是大多数深度学习模型中不常见的特性。但是,在实践中,很难设计适当的先验,并且随机过程的最令人感兴趣的示例在计算上代价很高,并且对于 n 和 m 放大得很大。这包括放大为 $\mathcal{O}((n + m)^3)$ 的GP。

[0069] 为了克服这个问题,本文描述的实施例利用一种新颖的机制来对输出进行建模,称为条件神经过程(CNP)。这些条件神经过程比替代方法(例如GP)高效得多,放大为 $\mathcal{O}(n + m)$ 。

[0070] CNP直接参数化条件随机过程,而不对某些先验过程强加一致性。给定固定维度的 O 的分布式表示,CNP参数化 $f(T)$ 上的分布。通过这样做,模型牺牲了与随机过程相关的数学保证,将其交换为功能灵活性和可扩展性。具体来说,给定观察的集合 O ,CNP是条件随机过程 Q_θ ,它限定了对于输入 $x \in T$ 的在 $f(x)$ 上的分布。 θ 是限定 Q 的所有参数的实向量。继承了随机过程的性质,假设 Q_θ 对于 O 和 T 的置换是不变的。如果 O', T' 分别是 O 和 T 的置换,则 $Q_\theta(f(T) | O, T) = Q_\theta(f(T') | O, T') = Q_\theta(f(T) | O', T)$ 。在本实施例中,通常通过采用因式结构(factored structure)来相对于 T 强制置换不变性。具体地,利用了 Q_θ ,因式 $Q_\theta(f(T) | O, T) = \prod_{x \in T} Q_\theta(f(x) | O, x)$ 。在没有关于输出空间 Y 的假设的情况下,这是确保有效随机过程的最简单方法。尽管如此,该框架仍可以扩展到非因式分布(non-factored distribution)。下面描述利用这样的模型的实施例。

[0071] CNP的限定特性是它经由固定维度的嵌入而以 O 为条件。更详细地说,利用以下架构,

$$[0072] \quad r_i = h_\theta(x_i, y_i) \quad \forall (x_i, y_i) \in O \quad (1)$$

$$[0073] \quad r = r_1 \oplus r_2 \oplus \dots \oplus r_{n-1} \oplus r_n \quad (2)$$

$$[0074] \quad \phi_i = g_\theta(x_i, r) \quad \forall (x_i) \in T \quad (3)$$

[0075] 其中, $h_\theta : X \times Y \rightarrow \mathbb{R}^d$ 是神经网络编码器; $g_\theta : X \times \mathbb{R}^d \rightarrow \mathbb{R}^e$ 是神经网络解码器; $\oplus$ 是交换运算,其取 $\mathbb{R}^d$ 中的元素并将它们映射到 $\mathbb{R}^d$ 的单个元素;并且 ϕ_i 是 $Q_\theta(f(x_i) | O, x_i) = Q(f(x_i) | \phi_i)$ 的参数。

[0076] 因此,CNP输出用于概率分布的参数。取决于任务,模型学习参数化不同的输出分布。对于回归任务, ϕ_i 可用于对于每个 $x_i \in T$ 参数化高斯分布 $\mathcal{N}(\mu_i, \sigma_i^2)$ 的均值和方差 $\phi_i = (\mu_i, \sigma_i^2)$ 。对于分类任务, ϕ_i 可用于参数化经典分布的 c 类别上的类概率 p_c 。在特定实施例中,可以对类概率的对数进行参数化(例如,经由对数层),使得该概率扩展为 $-\infty$ 至 $+\infty$ 之间的实数。在特定实施例中,将聚合器函数 $a_1 \oplus \dots \oplus a_n$ 当做平均运算 $(a_1 + \dots + a_n)/n$,但

是可以使用替代的聚合器函数。

[0077] 图2示出了根据实施例的用于使用条件神经过程 (CNP) 来预测解码器输出的示例性过程。该方法利用图1C所示的CNP布置。

[0078] 首先,系统接收观察的集合 O ,每个观察包括对于第一变量的相应的数据值 x_i 和对于依赖于第一变量的第二变量的相应的第二数据值 y_i 201。这些观察可以是表示各种形式的内容的计算机可读数据,例如视频数据、数值数据、图片数据、传感器数据、分类数据等。这些观察是用于条件化解码器的有标记的观察。

[0079] 然后,将每个观察输入编码器神经网络,以对每个观察进行编码,从而提供对于每个观察的编码的输出202。编码器神经网络可以根据输入和输出所需的维度而被配置的多层感知器 (multi-layer perceptron, MLP)。在一个实施例中,编码器是三层MLP,但是可以使用其他类型的编码器和其他层数。编码器根据编码器神经网络的参数 θ 将观察映射到隐空间 (latent space) (如参考式 (1) 所述)。以这种方式,编码器根据学习的函数 $h_\theta(x_i)$ 产生编码的输出 (编码的观察) 的集合 $\{r_i\}_{i=1}^n$ 。

[0080] 然后,聚合编码的输出 $\{r_i\}_{i=1}^n$ 以提供聚合的输出 r 203。根据聚合函数来聚合编码的输出 (如参考式 (2) 所述)。聚合函数是置换不变的交换运算。聚合函数可以包括取编码的输出的均值。

[0081] 然后,将聚合的输出 r 与目标输入值 x_t 组合,并将其传递到解码器神经网络系统,该解码器神经网络系统将组合作为输入并提供解码器输出204。解码器输出 y_t 是以编码的观察的聚合 r 为条件的目标输入值 x_t 的输出的预测。聚合的输出 r 和目标输入值 x_t 可以通过串联 (concatenation) 来组合。

[0082] 解码器神经网络可以根据输入和输出所需的维度而被配置的多层感知器 (MLP)。在一个实施例中,编码器是五层MLP,但是可以使用其他类型的解码器和层数。解码器根据解码器神经网络的参数 θ 将聚合输出 r 和目标输入值 x_t 的组合映射到输出空间 (具有等于 y_i 的维度) 上 (如参考式 (3) 所述)。以这种方式,解码器可以根据学习的函数 $g_\theta(x_t)$ 产生对于每个输入目标值 x_t 的预测。

[0083] 上面描述的架构确保了条件预测的置换不变性和 $O(n + m)$ 放大。由于解码器以编码的观察的集合上的聚合为条件,因此与训练标准神经网络时通常所需的观察相比,可以用较少的观察将系统训练到高等级的准确性和精度。

[0084] 要注意,由于可以从 $r_1 \oplus \dots \oplus r_{n-1}$ 以 $O(1)$ 计算 $r_1 \oplus \dots \oplus r_n$,因此该架构以最小的开销支持流式 (streaming) 观察。也就是说,每次接收到新的观察 r_i ,可以通过编码该新的观察并将新的编码的观察与先前的聚合进行聚合来更新聚合。在这种情况下,更新后的聚合 r' 将为:

[0085]
$$r' = r \oplus r_i = r_1 \oplus \dots \oplus r_{i-1} \oplus r_i$$

[0086] 这允许每次接收到新的观察时仅使用该新的观察和先前的聚合作为输入,高效地更新聚合。因此,无需将原始观察或其对应的编码表示存储在存储器中以供将来计算,因为聚合可以在没有它们的情况下被更新。

[0087] 编码器神经网络 $h_\theta(x_i, y_i)$ 和解码器神经网络 $g_\theta(x_i)$ 的参数 θ 可以通过反向传播进

行训练,使损失函数最小化(例如,经由随机梯度下降)。在一个实施例中,通过让神经网络系统 Q_θ 预测以0的随机选择的子集为条件的0来训练神经网络系统 Q_θ 。这对模型给出了在给定观察的集合的情况下分布P中固有的在空间X上的不确定性的信号。更准确地说,设 $f \sim P$, $O = \{(x_i, y_i)\}_{i=0}^{n-1}$ 为观察的集合,则 $N \sim \text{uniform}[0, \dots, n-1]$ 。系统以子集 $O_N = \{(x_i, y_i)\}_{i=0}^N \subset O$ (0的前N个元素) 为条件。然后,通过最小化负条件对数概率来训练系统

$$[0088] \quad \mathcal{L}(\theta) = -\mathbb{E}_{f \sim P} \left[\mathbb{E}_N [\log Q_\theta(\{y_i\}_{i=0}^{n-1} | O_N, \{x_i\}_{i=0}^{n-1})] \right] \quad (4)$$

[0089] 也就是说,系统最小化与基准真相标记(ground truth label) ($\{x_i\}_{i=0}^{n-1}$) 匹配的解码器的输出 ($\{y_i\}_{i=0}^{n-1}$) 的概率的负值。这等效于最大化与基准真相标记 ($\{x_i\}_{i=0}^{n-1}$) 匹配的解码器的输出 ($\{y_i\}_{i=0}^{n-1}$) 的概率;但是,鉴于许多优化器实现损失函数,因此可以利用概率的负值的最小化来保持一致性。

[0090] 因此,在其上对模型 Q_θ 评分的目标包括观察的值和未观察的值两者。在特定实施例中,损失函数的梯度的蒙特卡洛估计(经由概率或蒙特卡洛模拟)通过对f和N进行采样来获取。

[0091] 这种方法将施加先验知识的负担从解析先验转移到经验数据。这具有使实践者从必须指定先验的解析形式解放出来的优点,先验的解析形式最终旨在归纳其经验体验。仍然需要强调的是, Q_θ 不一定是所有观察集合的一组一致条件(conditionals),并且训练例程不能保证这一点。

[0092] 图3示出了根据实施例的用于训练条件神经过程(CNP)系统的示例性过程。

[0093] 该过程从接收301n个观察的集合 $\{O\}_{i=0}^{n-1}$ 开始。观察是标记的观察,包括n个输入 $\{x_i\}_{i=0}^{n-1}$ 和n个输出 $\{y_i\}_{i=0}^{n-1}$ 。然后,从0中选择N个观察的子集 O_N 302。对该子集 O_N 中的每个观察进行编码303,以确定对应的编码的观察的集合 $\{r_i\}_{i=0}^{n-1}$ 。然后,聚合204编码的观察以供聚合r。然后,通过将聚合r与原始观察集合0(包括 O_N 和 O_N 中未选择的所有其他观察)中的每个输入进行组合(例如串联)并将每个组合输入解码器以确定对应的预测输出,来确定305对于每个输入的预测的输出。预测的输出可以采用目标输出 $\hat{y}_t$ 的高斯均值和方差的形式。

[0094] 然后将预测的输出与基准真相值(来自0的 $\{y_i\}_{i=0}^{n-1}$)进行比较,并更新编码器和解码器的参数 θ 以减少损失函数306。然后,可以用更新的参数 θ 重复该方法,直到参数收敛到使损失函数最小化的一组最佳的参数上,或者直到达到最大迭代次数为止。可以训练参数以最大化目标点的对数似然。这可以例如使用随机梯度下降来实现,但是也可以利用各种不同优化方法。在一个实施例中,使用自适应矩估计(adaptive moment estimation, Adam)优化方法来优化参数。

[0095] 应当注意,尽管图3的方法确定了对于被用于生成聚合的相同观察 O_N 的预测,但是在替代实施例中,可以将0的不同子集用于聚合的生成和预测的输出的计算。

[0096] 总而言之,在本文描述的实施例中:

[0097] 1.CNP是经训练以对函数 $f \sim P$ 的经验条件分布建模的函数上的条件分布。

[0098] 2.CNP在0和T中是置换不变的。

[0099] 3.CNP是可扩展的,对于用 n 个观察进行 m 个预测实现的运行时间复杂度为 $\mathcal{O}(n + m)$ 。

[0100] 可以针对例如分类、图像补全和预测以及函数回归的各种用例实现本文描述的方法。

[0101] 为了测试函数回归,使用由具有指数核的高斯过程(GP)生成的曲线来训练系统。对曲线进行采样,选择曲线上 n 个点的子集作为观察,并选择其他点的子集作为目标点。该模型如图1所示,具有三层MLP编码器 h ,其具有128维输出表示 r_i ;以及五层MLP解码器,其配置为输出每个目标输出 $\hat{y}_i$ 的均值和方差。方差允许系统量化预测中的误差。

[0102] 当将经训练的系统与由具有正确的超参数的GP(其构成了性能的上限)生成的预测进行比较时,与GP所需的相比,该系统能够以更少的上下文点(用于生成聚合的观察)进行回归。尽管由生成的预测,但是随着上下文点数量的增加,模型的准确性提高,并且模型的不确定性降低。至关重要的是,模型学会非常准确地估计在给定观察下其自身的不确定性。尽管如此,它提供了良好的近似,随着上下文点数量的增加,其准确性也随之增加。

[0103] 此外,当向模型提供由其中核的参数发生改变的GP生成的一组上下文点时,该模型实现类似良好的性能。这种类型的回归任务对于GP而言并非无关紧要,相反,在本文的实施例中,可以通过仅改变用于训练的数据集来有效地进行操作。

[0104] 可以使用本文描述的实施例,作为针对灰度图像的 $f: [0, 1]^2 \rightarrow [0, 1]$ 的或针对RGB图像的 $f: [0, 1]^2 \rightarrow [0, 1]^3$ 的函数上的回归任务来执行图像补全。输入 x 是归一化到 $[0, 1]^2$ 的2D像素坐标,输出 y 是对应像素的灰度强度或RGB强度的向量。对于此补全任务,可以使用与1D函数回归相同的模型架构(除了针对RGB将最后一层设为3维之外)。

[0105] 训练过程涉及在每个步骤从数据集中选择图像并挑选像素的子集作为观察。以这些为条件,对模型进行训练以预测图像中所有像素(包括已经以其为条件的像素)的值。像在1D回归中一样,该模型输出每个像素的高斯均值和方差,并针对基准真相图像的对数似然进行优化。

[0106] 与1D回归任务一样,即使用少量上下文点,系统也可以进行准确的预测。至关重要的是,当仅以一个无信息上下文点(例如,图像边缘上的黑像素)为条件时,模型的预测对应于所有训练图像上的平均。随着上下文点数量的增加,预测变得与底层基准真相更加相似。这证明了模型提取数据集特定的先验知识的能力。值得一提的是,即使用完整的观察集合,该模型也不实现像素完美的重构,因为在表示级别上存在瓶颈。

[0107] 由于该实施例返回因式的输出(factored outputs),因此给定有限的上下文信息,它可以产生的最佳预测是在与该上下文一致的所有可能的预测上求平均。这样一种替代是在模型中添加隐变量(latent variable),使得可以以该上下文为条件对它们进行采样,以在数据分布中以高概率产生预测。下面将讨论该实施例。

[0108] 该模型的一个重要方面是其估计预测的不确定性的能力。随着更多观察被添加,方差从几乎均匀地散布在图像上转移到局部化到特定于底层图像的区域周围的位置,例如

图像内的区别特征的边缘。能够对给定某个上下文的不确定性进行建模可能对许多任务有帮助。一个示例是主动探索,其中,模型可以选择要观察的位置。可以通过将根据不确定性选择观察(即每个步骤方差最高的像素)时CNP的预测与随机像素进行比较,以对此进行测试。这种方法是进行主动探索的一种非常简单的方法,但是与随机选择条件点相比,它已经产生了更好的预测结果。

[0109] 本文描述的实施例的一个重要方面是其灵活性不仅在于接收的观察和目标的数量,而且还在于输入值方面。第一类型的灵活性是,在测试期间,系统可以以在训练期间未遇到的子集为条件。例如,模型可以以图像的一半为条件。这迫使模型不仅根据图像的某些固定(stationary)平滑度属性来预测像素值,而且还根据全局空间属性(例如,面部不同部位的对称性以及相对位置)来预测像素值。本文描述的实施例能够捕获这些属性。与此相比,具有固定核的GP无法捕获该属性,并且在没有观察的情况下会恢复为其均值(均值本身可以是非固定的,但通常这不足以捕获感兴趣的属性)。

[0110] 另外,本文描述的实施例在目标输入值方面是灵活的。这意味着,例如,可以以训练期间未观察到的分辨率查询模型。在测试时,可以使用仅使用特定分辨率的像素坐标训练的模型来预测原始坐标之间的目标的子像素值。只需要一次前向传递来以不同的分辨率查询系统。尽管GP也表现出这种类型的灵活性,但对于经训练的生成模型却并非如此,经训练的生成模型只能预测它们在其上被训练的像素坐标的值。在这个意义上,本文描述的方法在两个方面都占了上风-它们在条件化和预测任务方面很灵活,并且具有从训练集中提取领域知识的能力。

[0111] 在以上实施例中,仍然存在一些方面可以被修改以适合特定需求。例如,h的切实实现可以适应于数据类型。对于低维数据,编码器可以实现为多层感知器(MLP),而对于具有较大维度和空间相关性的输入,其还可以包括卷积。最后,在上述实施例中,该系统不能产生任何相干样本,因为它学习的是仅对均值和方差的因式预测进行建模,而不管目标点之间的协方差。这是该模型的这种特定实现的结果。可以获得相干样本的一种方法是通过引入可以从其进行采样的隐变量。

[0112] 上述主要实施例利用了预测目标输出的均值和方差的因式模型(factored model)。尽管均值本身是一种有用的预测,并且方差是捕获不确定性的一种好方法,但该因式模型无法获得多个目标上的相干样本。

[0113] 考虑上述参考以少量观察为条件的图像补全的实施例。可以实现此实施例以补全数字图像。不是仅预测所有数字的均值,有时需要可用于以观察为条件对所有可能的数字的不同相干图像进行采样的系统。

[0114] GP可以这样做,因为它们包含预测观察和目标中所有点之间的协方差的参数核。这形成了可以用于相干地抽取样本的多元高斯分布。为了在经训练模型中保持此属性,一种方法是训练模型以预测GP核。然而,困难在于需要通过采样反向传播,这涉及大的矩阵求逆(或其某种近似)。

[0115] 与此相比,根据一个实施例,提供了一种如上所述的系统,但是向解码器g增加了隐变量z。这允许模型捕获全局不确定性。为了生成相干样本,根据观察计算表示r,该表示参数化隐变量z上的高斯分布。然后,可以对隐变量z进行采样,其中,相同的值被用于生成所有目标(在图像的情况下为所有像素)的样本。为了获得不同的相干样本,可以从隐变量z

抽取新样本,并针对所有目标再次将其输入到解码器中。与标准VAE相似,可以使用以观察为条件的条件高斯先验 $p(z|0)$ 和也以目标点为条件的高斯后验 $p(z|0,T)$,通过优化对数似然的变分下界来训练该模型。

[0116] 我们使用与以前可以使用的模型相同的模型,但是将表示 r 转而串联到隐变量 z 的向量。对于图像补全(例如,具有维度为128的 r), z 可以为64。对于先验模型和后验模型两者,可以使用三层MLP并且可以取它们的输出的均值(对于观察编码器)。要强调的是,先验和后验之间的区别在于先验仅看到观察到的像素,而后验既看到了观察到的像素又看到了目标像素。当用少量观察到的像素从该模型进行采样时,将获得相干样本并捕获数据集的变异性(variability)。随着模型以越来越多的观察为条件,样本的变异性下降,并且它们最终收敛到单一可能性。

[0117] 另外,本文描述的方法可以应用于分类(例如,图像分类)。在这种情况下,通过使用输入标记中提供的信息,仅在来自同一类的观察上产生聚合。换句话说,基于观察的标记(其基准真相分类)将观察划分到它们对应的类中。对每个类的观察进行编码,然后进行聚合以形成类特定的聚合。然后,组合(例如,串联)类特定的聚合(聚合的类特定的表示)以形成最终表示。然后,将其用作输入,以条件化解码器从而对目标输入进行分类。解码器对于每个潜在类输出目标输入属于该类的概率。

[0118] 假设类特定的表示的大小和类的数量都是恒定的,则最终表示的大小仍然是恒定的,因此 $\mathcal{O}(n + m)$ 运行时间仍然成立。

[0119] 表1中示出了CNP分类实施例在Omniglot数据集上相对于例如MANN(Santoro等人,2016)和匹配网络(matching networks,MN)(Vinyals等人,2016)的替代系统的性能。

[0120]		5 路径 (5-way)		20 路径 (20-way)		运行时间
		准确性		准确性		
		1 样本 (1-shot)	5 样本 (5-shot)	1 样本	5 样本	
	MANN	82.8%	94.9%	-	-	$\mathcal{O}(n\ m)$
	MN	98.1%	98.9%	93.8%	98.5%	$\mathcal{O}(n\ m)$
	CNP	95.3%	98.5%	89.9%	96.8%	$\mathcal{O}(n\ +\ m)$

[0121] 表1:CNP实施例在Omniglot上相对于MANN和MN的分类结果

[0122] 可以看出,CNP取得比复杂得多的模型(例如MANN)更高的准确性。尽管CNP对于样本(shot)分类没有击败这些替代方法,但准确性值是相当的。至关重要的是,在测试时使用简单得多的架构并以 $\mathcal{O}(n + m)$ 而不是 $\mathcal{O}(nm)$ 的较低运行时间来达到这些值。因此,本文描述的实施例具有相当的准确性,并且计算效率大大提高。

[0123] 本文描述的实施例利用了一组新的模型,即条件神经过程。这些模型在测试时是灵活的而且具有从训练数据提取先验知识的能力。本文描述的实施例能够执行各种任务,包括回归、分类和图像补全。

[0124] 重要的是要注意,本文描述的特定CNP实现方式只是简单的概念验证,并且可以进行实质性扩展,例如,通过包含与现代深度学习进步相适应的更精细的架构。此外,通过共

享先验模型和后验模型或添加隐变量的层次结构以及其他选项,可以将模型本身更新为“更加贝叶斯”。

[0125] 总而言之,这项工作可以视为迈向学习高级别抽象的一步,学习高级别抽象是当代机器学习的巨大挑战之一。大多数常规深度学习模型学习的功能在训练的任何阶段都与特定的、受约束的统计上下文相关。在这个意义上,经训练的CNP更具一般性,因为它封装了一族函数的高级别统计信息。因此,它构成了可重用于多个任务的高级别抽象。这些模型可以帮助解决似乎与抽象有关的许多关键的机器学习问题,例如迁移学习、元学习和数据效率。

[0126] 鉴于上述讨论,通过使用观察的编码的表示的聚合来条件化解码器,本文描述的方法能够以计算效率高得多的方法基于目标输入进行准确的预测。实施例实现了较简单的架构,该架构相对于替代的更复杂的架构而言对实现计算系统提供了减少的存储器负担。另外,本文描述的实施例能够以仅少量的训练观察和仅少量的训练步骤来获得准确的预测。这进一步提高了效率,并提供了基于新的观察快速学习的能力。

[0127] 本说明书与系统和计算机程序组件有关地使用术语“被配置”。一个或多个计算机的系统被配置为执行特定操作或动作的意思是,系统在其上安装有软件、固件、硬件或其组合,在操作时,所述软件、固件、硬件或其组合使系统执行所述操作或动作。一个或多个计算机程序被配置为执行特定操作或动作的意思是,所述一个或多个程序包括指令,该指令在由数据处理装置执行时使装置执行所述操作或动作。

[0128] 本说明书中描述的主题和功能操作的实施例可以在数字电子电路、在有形地实现的计算机软件或固件、在计算机硬件(包括本说明书中公开的结构以及其结构等效物)或在它们中的一个或多个的组合中实现。本说明书中描述的主题的实施例可以被实现为一个或多个计算机程序,即编码在有形非暂态程序载体上以供数据处理装置执行或控制数据处理装置的操作的计算机程序指令的一个或多个模块。替代地或附加地,程序指令可以编码在人工生成的传播信号上,例如,机器生成的电、光或电磁信号,其被生成为编码信息以传输到合适的接收器装置以供数据处理设备执行。计算机存储介质可以是机器可读存储设备、机器可读存储基片、随机或顺序存取存储器或它们中一个或多个的组合。但是,计算机存储介质不是传播信号。

[0129] 术语“数据处理装置”涵盖用于处理数据的所有种类的装置、设备和机器,举例来说包括可编程处理器、计算机或多处理器或计算机。所述装置可以包括专用逻辑电路,例如FPGA(现场可编程门阵列)或ASIC(专用集成电路)。除了硬件之外,所述装置还可以包括代码,该代码为所针对的计算机程序创建执行环境,例如,构成处理器固件、协议栈、数据库管理系统、操作系统或它们中一种或多种的组的代码。

[0130] 计算机程序(其也被称为或被描述为程序、软件、软件应用、模块、软件模块、脚本或代码)可以以任何编程语言写成,包括编译或解释型语言、陈述性或过程语言,并且可以以任何形式部署计算机语言,包括以独立程序的形式或以模块、组件、子例程或其他适于用在计算环境中的单元。计算机程序可以但不必与文件系统中的文件对应。程序可以存储在保持其他程序或数据的文件的一部分中,例如,存储在标记语言文档中的一个或多个脚本中、在专用于所针对的程序的单个文件中、或在多个协作的文件中,例如,存储一个或多个模块、子程序或代码部分的文件中。计算机程序可以部署为在一个计算机上或多个计算机

上执行,所述多个计算机位于一个站点或分布在多个站点之间并通过通信网络连接。

[0131] 当在本说明书中使用时,“引擎”或“软件引擎”指提供与输入不同的输出的软件实现的输入/输出系统。引擎可以是编码的功能块,例如库、平台、软件开发套件(“SDK”)或对象。每个引擎可以被实现在包括一个或多个处理器和计算机可读介质的任何适当类型的计算设备上,例如,服务器、移动电话、平板计算机、笔记本计算机、音乐播放器、电子书阅读器、膝上型或台式计算机、PDA、智能电话或其他固定或便携式设备。另外,引擎中的两个或更多个可以实现在相同的计算设备上或在不同的计算设备上。

[0132] 本说明书中描述的过程和逻辑流可以由执行一个或多个计算机程序的一个或多个可编程计算机执行,从而通过对输入数据进行操作并生成输出来执行功能。过程和逻辑流也可以由专用逻辑电路(例如FPGA(现场可编程门阵列)或ASIC(专用集成电路))来执行,并且装置可以被实现为上述专用逻辑电路。例如,过程和逻辑流可以由图形处理单元(GPU)来执行,并且装置也可以被实现为图形处理单元。

[0133] 适于执行计算机程序的计算机举例来说包括、可以基于通用或专用微处理器或二者,或者任何其他种类的中央处理单元。通常,中央处理单元将从只读存储器或随机存取存储器或二者接收指令和数据。计算机的基本元件是用于运行或执行指令的中央处理单元和用于存储指令和数据的一个或多个存储器设备。一般来说,计算机还将包括用于存储数据的一个或多个大容量存储设备(例如、磁盘、磁光盘或光盘),或操作性地耦合到一个或多个所述大容量存储设备以从其接收数据或将数据传输到其处或者执行接收和传输二者。然而,计算机不是必需具有这样的设备。另外,计算机可以嵌入到另一设备中,例如,移动电话、个人数字助理(PDA)、移动音频或视频播放器、游戏控制台、全球定位系统(GPS)接收器或便携式存储设备(例如通用串行总线(USB)快闪驱动器),等等。

[0134] 适于存储计算机程序指令和数据的计算机可读介质包括所有形式的非易失性存储器、介质和存储设备,举例来说,包括半导体存储设备,例如EPROM、EEPROM和闪存设备;磁盘,例如内部硬盘或可移除盘;磁光盘;以及CD-ROM和DVD-ROM盘。处理器和存储器可以由专用逻辑电路来补充或可以结合到专用逻辑电路中。

[0135] 为了提供与用户的交互,本说明书中描述的主题的实施例可以实现在具有显示设备及键盘和指向设备的计算机上,显示设备例如用于向用户显示信息的CRT(阴极射线管)或LCD(液晶显示)监视器,此外,用户可以通过键盘和指向设备向计算机提供输入,指向设备例如为鼠标或轨迹球。也可以使用其他种类的设备来提供与用户的交互;例如,提供给用户的反馈可以是任何形式的感觉反馈,例如视觉反馈、听觉反馈或触觉反馈;并且可以以任何形式接收来自用户的输入,包括声学、语音或触觉输入。此外,计算机可以通过向由用户使用的设备发送文档和接收来该设备的文档来与用户交互;例如,通过响应于从网页浏览器接收的请求而将网页发送到用户的客户端设备上的浏览器来与用户交互。

[0136] 本说明书中描述的主题的实施例可以在计算系统中实现,该计算系统包括后端组件(例如作为数据服务器),或包括中间件组件(例如应用服务器),或包括前端组件(例如具有图形用户界面和网页浏览器的客户端计算机,用户可通过其与本说明书中描述的主题的实现方式进行交互),或者一个或多个这样的后端、中间件或前端组件的任意组合。可以通过任意形式或媒介的数字数据通信,例如通信网络,将系统的组件互连。通信网络的示例包括局域网(LAN)和广域网(WAN),例如互联网。

[0137] 计算系统可以包括客户端和服务端。一般来说,客户端和服务端彼此远离,并且通常通过通信网络交互。通过运行在各个计算机上并且彼此具有客户端-服务端关系的计算机程序而形成的客户端和服务端的关系。

[0138] 尽管本说明书包含许多具体的实现细节,但是这些细节不应当被解释为对任何发明或请求保护的范疋进行限制,而应该被解释为对于可能特定于特定发明的特定实施例的特征的描述。在本说明书中在不同实施例的上下文中描述的某些特征也可以以组合方式实现在单个实施例中。相反,在单个实施例的上下文中描述的各种特征也可以在多个独立的实施例中实现或在任何适当的子组合中实现。此外,尽管特征可以如上所述描述为在某些组合中起作用并且甚至最初也这样地要求保护,但在一些情况下,来自所要求保护的组合的一个或多个特征可以从组合中排除,并且所要求保护的组合可以指向子组合或子组合的变体。

[0139] 类似地,尽管在附图中以特定顺序描绘了操作,但是这不应当被理解为要求以示出的特定顺序或相继的顺序来执行这样的操作或者需要执行所有示意的操作来实现期望的结果。在某些情况下,多任务和并行处理可以是有利的。另外,在上述实施例中对各种系统模块和组件的划分不应当被理解为在所有实施例中都要求这样的划分,而是应当理解,所描述的程序组件和系统可一般地在单个软件产品中被集成在一起或者被封装成多个软件产品。

[0140] 已经描述了主题的特定实施例。其他实施例在下述权利要求的范围内。例如,记载在权利要求中的动作可以以不同的顺序执行而仍然达到期望的结果。作为一个示例,附图中描绘的过程不是必须要求所示出的特定顺序或者相继的顺序来实现期望的结果。在某些实现方式中,多任务和并行处理可能是有利的。

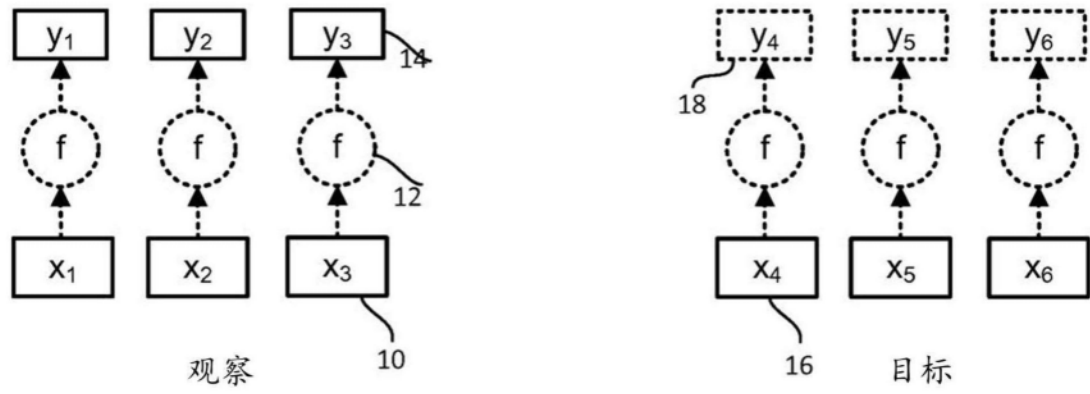


图1A

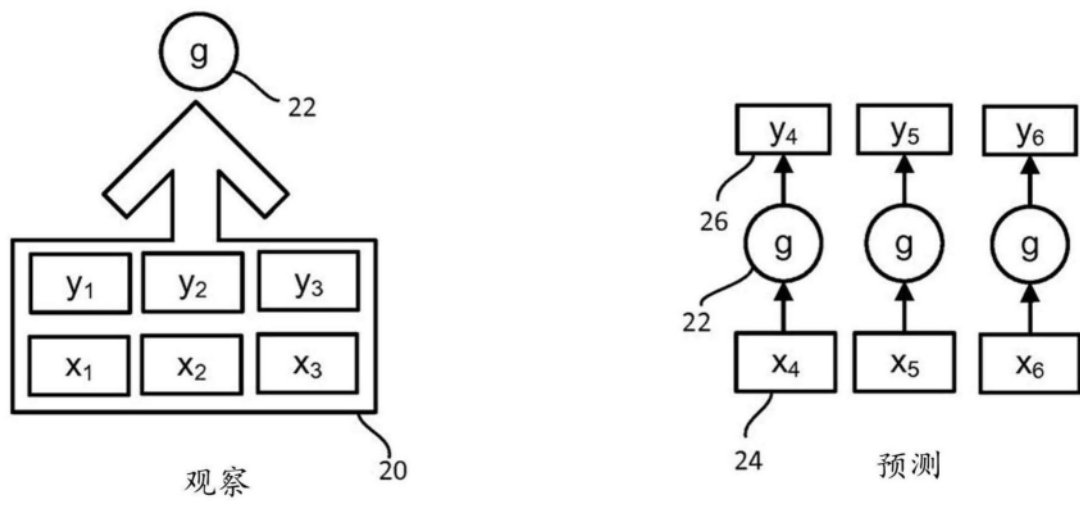


图1B

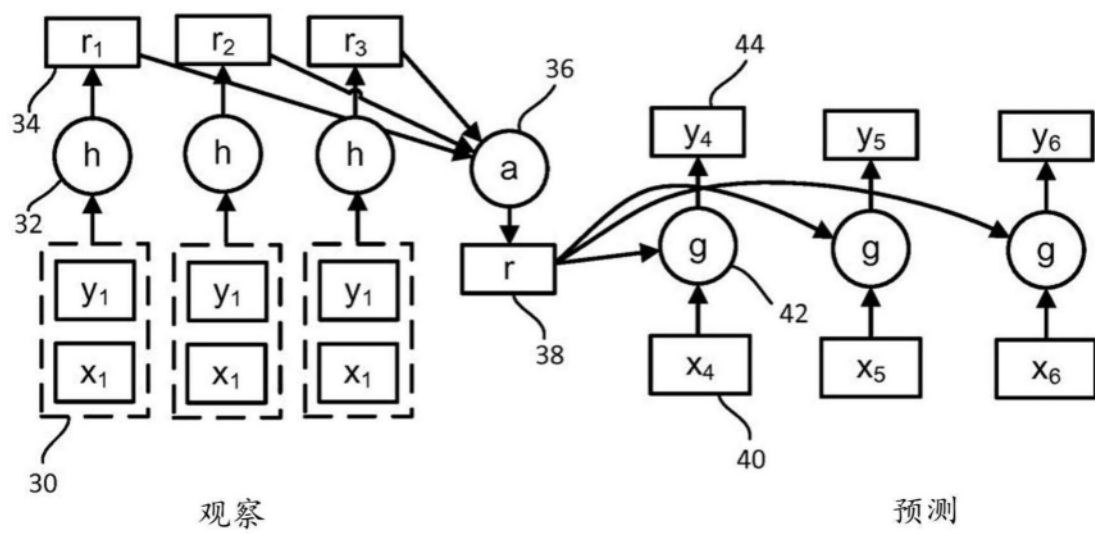


图1C

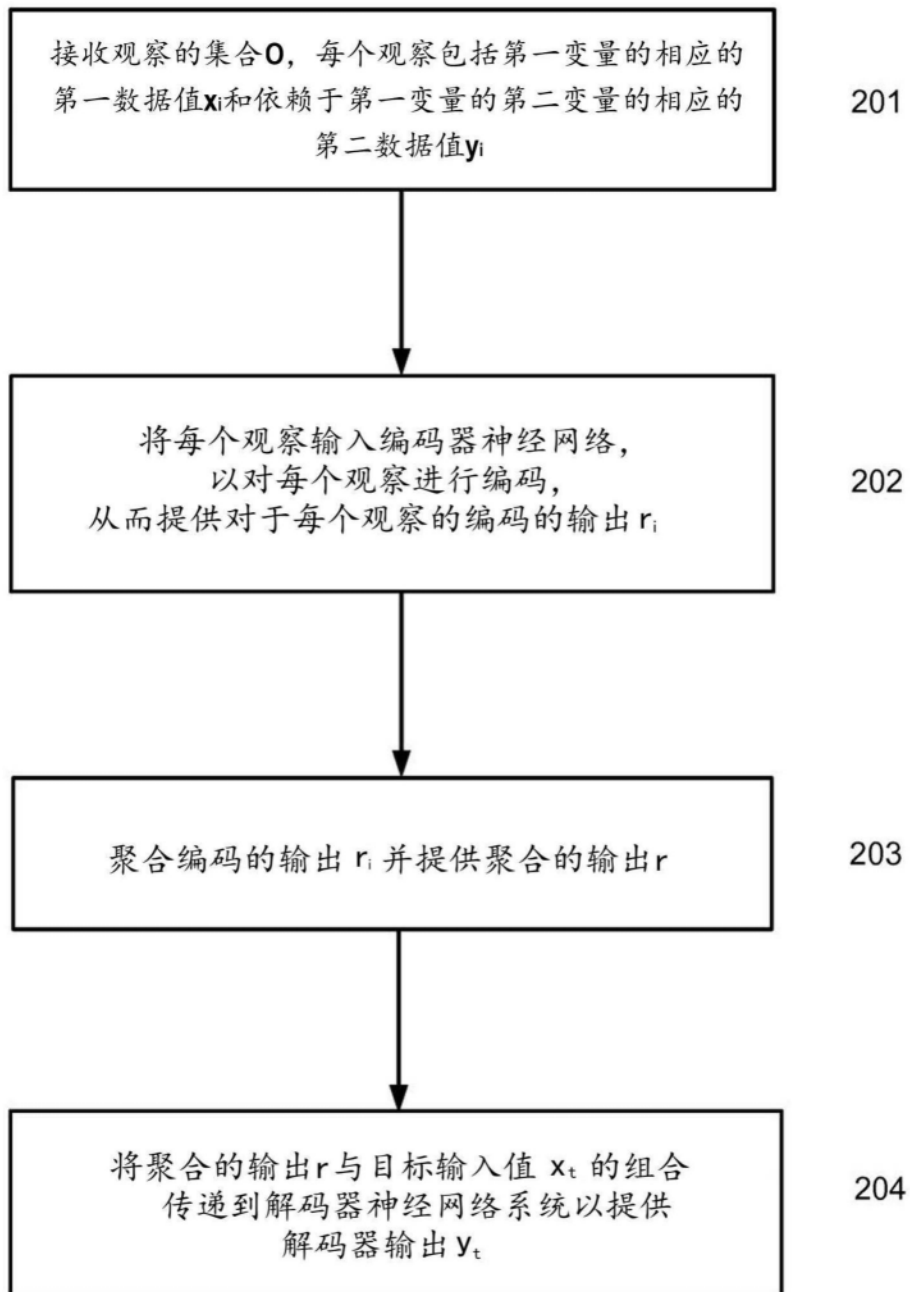


图2

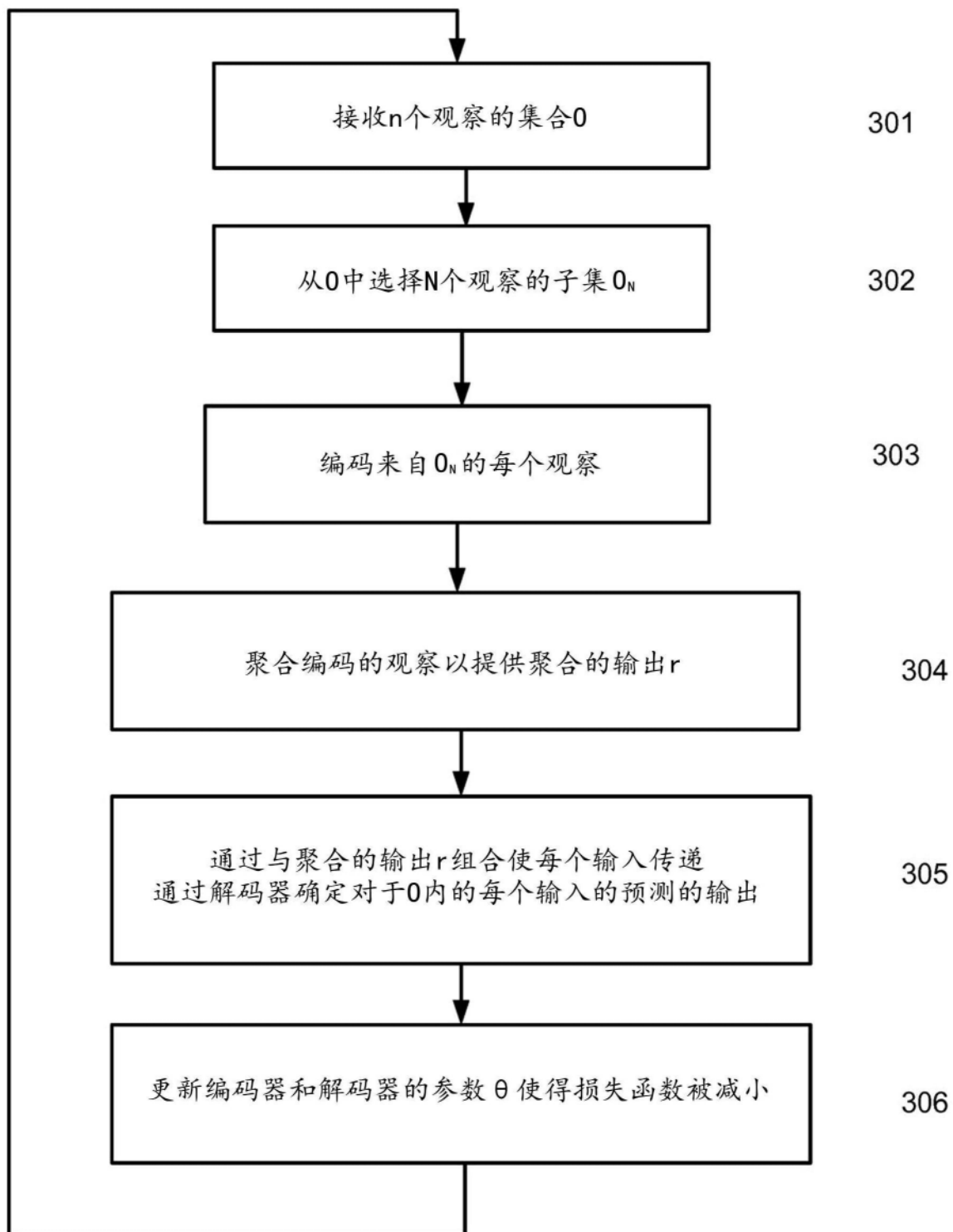


图3